AMÉRICO LOPES DE AZEVEDO

APOIO À DECISÃO NA NEGOCIAÇÃO DE ENCOMENDAS EM REDES DE EMPRESAS

Faculdade de Engenharia da Universidade do Porto

1999

AMÉRICO LOPES DE AZEVEDO

APOIO À DECISÃO NA NEGOCIAÇÃO DE ENCOMENDAS EM REDES DE EMPRESAS

Dissertação apresentada à Faculdade de Engenharia da Universidade do Porto
para a obtenção do grau de Doutor em Engenharia Electrotécnica e de Computadores
e realizada sob a orientação científica do Doutor Jorge Manuel Pinho de Sousa,
Professor Auxiliar da Faculdade de Engenharia da Universidade do Porto.

DEPARTAMENTO DE ENGENHARIA ELECTROTÉCNICA E DE COMPUTADORES
FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO
1999

681.3(043)/AREA/AIP
UNIVERSIDADE DO PORTO
Faculdade de Engenharia
BIBLIOTECA M
N.º 49055
CDU
Data 22/05/2000
cat

O trabalho de investigação apresentado nesta dissertação foi apoiado pelo programa PRODEP 2, medida 5 – Acção 5.2 – Formação Avançada no Ensino Superior – Concurso público 3/94

Às minhas meninas: Cristina, Inês e Beatriz.

Resumo

A confluência da globalização e dos avanços tecnológicos revolucionou neste final de século o processo de criação de valor, nomeadamente através da transição de uma economia industrial para uma economia baseada na informação. Neste contexto de grande complexidade, instabilidade e incerteza, as mudanças organizacionais têm sido encaradas como um dos principais meios para estruturar e explorar o novo mundo dos negócios.

Ao nível da produção, uma consequência imediata do estabelecimento de novos paradigmas organizacionais, nomeadamente sob a forma de redes de empresas e suportadas por cadeias logísticas complexas, é a alteração nos processos de planeamento, de um nível estritamente local, para um nível mais abrangente, onde os aspectos logísticos e de coordenação assumem grande importância. Por outro lado, como resultado da cada vez maior interacção com o cliente final, estas novas formas de organização pressupõem a existência de um conjunto de instrumentos de gestão para, de forma eficiente e em tempo útil, apoiar todo o processo de negociação e aceitação de encomendas (*order promising*) com a avaliação real da capacidade produtiva.

Neste contexto, foi identificado um conjunto de requisitos que uma avaliação exaustiva dos produtos disponíveis comercialmente mostrou não estarem adequadamente contemplados. Assim, desenvolveram-se um quadro conceptual e uma metodologia de planeamento, tendo em conta estes requisitos. Com base nesse quadro, foi projectado um Sistema de Apoio à Decisão para apoiar o processo de negociação e aceitação de encomendas em *tempo-real* e com avaliação de capacidades, em estruturas produtivas distribuídas (conceptualizadas como *Empresas Virtuais*) e tendo em conta os aspectos logísticos e de coordenação decorrentes de uma organização produtiva em rede. Este sistema foi desenvolvido recorrendo a tecnologia "estado-da-arte" de componentes distribuídos.

Com base numa arquitectura informática distribuída, de carácter inovador, procedeu-se à estruturação do problema associado à coordenação da *Empresa Virtual*, tendo sido desenvolvidos dois procedimentos de optimização do tipo meta-heurísticas, baseados em *simulated annealing* e em *algoritmos genéticos*. Os resultados computacionais, embora preliminares, demonstram as potencialidades das abordagens consideradas.

Finalmente, é apresentada uma abordagem alternativa ao problema, baseada no paradigma dos *Sistemas Multi-Agente*. Nesta linha, é proposta uma arquitectura para o sistema de informação, baseada em agentes inteligentes e que deverá permitir o desenvolvimento de metodologias cooperativas de planeamento.

Abstract

Globalisation together with technological advances have dramatically changed the process of value creation in this end of the century, through a fast move from an industrial economy to an information based economy. In this context of complexity, instability and uncertainty, organisational changes can be viewed as a key step for structuring and exploring the new business world.

At the production level, an immediate result of new organisational paradigms (such as enterprise networks supported by complex logistic chains), is the move in planning procedures from a strictly local level, towards a broader level, where logistic and co-ordination issues play an important role. On the other hand, due to the growing interaction with final customers, these new forms of organisation assume that a set of management tools is available, to provide efficient and on-time support for *order promising*.

In this context, we have identified a set of requirements that are not adequately satisfied by currently available commercial systems, as shown by a comprehensive survey we have conducted. Accordingly, we have designed a conceptual framework and a planning methodology, taking into account these requirements.

Based on this framework, we have developed a Decision Support System to help planners in a capacity checker, real-time *order promising* process, in distributed production networks (conceptualised as *Virtual Enterprises*), and taking into account the resulting logistic and co-ordination issues. This system was developed with the "state-of-the-art" technology of distributed components.

Furthermore, based on an innovative distributed architecture, we have structured the problem associated with the co-ordination of the *Virtual Enterprise*, and designed two meta-heuristic procedures (*simulated annealing* and *genetic algorithm*). Preliminary computational results have clearly shown the potential of these approaches.

Finally, we present an alternative approach to the problem, that is based on the paradigm of *Multi-Agent Systems*. In this direction, we propose an architecture for the information system, built around *intelligent agents*, that will hopefully allow the setting up of effective co-operative planning methodologies.

Résumé

La globalisation croissante avec les progrès technologiques ont fortement changé les processus de création de valeur en cette fin de siècle, par un mouvement rapide de l'économie industrielle vers une économie basée sur l'information. Dans ce contexte de complexité, instabilité et incertitude, les changements organisationnels sont sûrement une clé fondamentale pour structurer et exploiter le nouveau monde des négoces.

Au niveau de la production, comme résultat des nouveaux paradigmes organisationnels tels que les réseaux d'entreprises, supportés par des chaînes logistiques complexes, on assiste à des changements importants sur les procédures de gestion qui sont en train de passer d'un niveau strictement local à un niveau plus global où les aspects logistiques et de coordination sont très importants. D'autre part, comme résultat de l'interaction croissante avec le client final, ces nouvelles formes d'organisation sont basées sur l'existence d'un ensemble d'outils de gestion pour appuyer d'une façon efficiente, tout le procès de négociation et d'acceptation de commandes (*order promising*).

Dans ce contexte, on a identifié un ensemble de besoins auxquels les systèmes commerciaux actuels ne répondent pas d'une façon satisfaisante, comme le démontre une étude qu'on a menée encadrée par ce travail. Ces besoins ont été tenus en compte pour la conception et développement d'un cadre conceptuel et d'une méthodologie de gestion. Basé sur ce cadre, on a développé un Système d'Appui à la Décision pour aider les décideurs dans les processus de négociation avec des capacités en des réseaux de production distribués (formalisés comme des *Entreprises Virtuelles*) et tenant compte des questions logistiques et de coordination. Ce système a été développé avec la technologie « état-de-l'art » des composants distribués.

L'approche suivie dans ce travail de recherche conduit à une architecture informatique distribuée de caractère innovateur. De plus, on a procédé à la structuration du problème associé à la coordination de l'Entreprise Virtuelle ayant été développés deux logiciels d'optimisation du type meta-heuristique basés en *simulated annealing* et en *algorithmes génétiques*. Bien qu'ils soient préliminaires, les résultats obtenus démontrent les potentialités des approches considérées.

Finalement, on présente une autre approche à ce problème, qui est basée dans le paradigme des *Systèmes Multi-Agent*. De cette façon on propose une architecture pour le système d'information qui s'appuie sur les propriétés des agents intelligents et qui devra permettre le développement de méthodologies coopératives de planification.

Prefácio

O texto que se dá a conhecer nesta tese é a síntese de um trabalho de investigação que decorreu durante vários anos e que envolveu inúmeras pessoas, entidades e instituições. É para mim, sem dúvida alguma, um marco importante poder escrever este texto não só pelo significado que encerra em si, como fundamentalmente pela materialização dos variados conhecimentos que tive possibilidade de adquirir, complementar e de refinar.

Por várias vezes tentei com "duas palavras" explicar o assunto do meu trabalho de doutoramento. Sem sucesso, e com a finalidade de ultrapassar a incompreensão às minhas sucintas explicações, optei então por caracterizar o contexto na qual se situava o trabalho e só depois tentava descrever nas tais "duas palavras" o que de facto andava a fazer. Basicamente, começava por referir o seguinte: "... Os sistemas de informação empresariais de gestão, referidos usualmente por sistemas ERP, tendem a ser encarados como um dos principais meios que sustentam e incrementam a competitividade das empresas numa economia em constante mudança. A economia e a forma de fazer negócios mudou de facto. O cliente está na verdade mais próximo do fornecedor exigindo respostas rápidas e fiáveis aos seus pedidos – potenciais encomendas. Por outro lado, os meios de produção estão cada vez mais distribuídos geograficamente sendo, simultaneamente, mais heterogéneos nas partes que os constituem. Neste contexto, os fornecedores de soluções e sistemas ERP, clamam para si a fórmula mágica que permite às empresas serem eficientes, ágeis e competitivas. Contudo, a realidade é, na verdade, bastante diferente!

Fortemente baseados em metodologias de planeamento e modelos de produção desadequados às novas exigências dos mercados e das estruturas produtivas (por exemplo, através de redes de empresas), os sistemas ERP ainda estão longe de satisfazerem um requisito, nos dias de hoje, básico e simultaneamente fundamental: a necessidade de em tempo útil (no limite enquanto o cliente está ao telefone!) conseguirem obter uma solução global exequível e optimizada, quer do ponto de vista da produção quer do ponto de vista logístico, e simultaneamente determinarem uma data fiável de entrega (mediante a avaliação real da capacidade produtiva) para o produto pretendido pelo cliente". Em termos gerais, era desta forma que eu procurava explicar que o que eu andava a tentar fazer era arranjar uma solução que permitisse que, ao nível da pro-

dução, um decisor produzisse uma resposta optimizada, rápida e fiável a cada pedido dos clientes.

Apesar de formalmente esta dissertação apresentar na capa o nome de um autor, não quero deixar de exprimir o meu agradecimento a todas as pessoas, entidades e instituições que permitiram e contribuíram, de forma diversa, para a concretização deste trabalho de investigação. Em particular e em primeiro lugar, quero salientar o inestimável, valioso e constante apoio que o Prof. Doutor Jorge Pinho de Sousa, meu orientador científico, me proporcionou desde a primeira hora. Não posso deixar de referir a confiança e autonomia que em mim depositou, desde o início, e o permanente incentivo, ajuda e rigor que colocou sempre na realização deste trabalho.

Aos colegas da Unidade de Engenharia de Sistemas de Produção do INESC PORTO agradeço toda a atenção, cuidado e dedicação que me dispensaram. Em particular, quero agradecer aos colegas César Toscano e Rui Diogo pela constante disponibilidade demonstrada, pela paciência e pela ajuda sem fim nas implementações informáticas realizadas.

Não posso deixar a oportunidade para agradecer também a todos os colegas da FEUP que, de forma directa e indirecta, me permitiram ter a "concentração" indispensável à realização deste trabalho.

De forma muito especial, e apesar da ausência do meu pai Fernando, quero expressar profundamente a minha gratidão aos meus pais pela confiança que sempre me deram, pela formação que me proporcionaram e pelos ensinamentos que me transmitiram.

Finalmente, e reconhecendo a minha dificuldade em expressar literariamente aquilo que devo e sinto, não quero deixar de tentar de expressar também aqui, o conforto que advém de uma "família" que teimosamente suportava a minha constante indisponibilidade e falta de tempo, possivelmente com a esperança de que as "coisas" iriam em breve normalizar. Não posso pois deixar de salientar o decisivo apoio logístico que me proporcionaram no momento em que, e em quaisquer condições, esperariam receber!

Sumário

Sumário

Lista de Figuras

Lista de Tabelas

Lista de Algoritmos

Capítulo 1 - Introdução 3

1.1 Enquadramento e relevância do problema considerado 3
1.1.1 Novas exigências no processo de planeamento 4
1.1.2 Área de aplicação 6
1.2 Objectivos e motivação do trabalho realizado 7
1.3 Organização da dissertação 9

Capítulo 2 - Planeamento da Produção e Avaliação de Capacidades 15

2.1 Contexto 15
2.2 Tipos de sistemas de produção 18
2.3 Níveis de planeamento 22
2.4 A negociação de encomendas baseada em ATP 24
2.4.1 Considerações gerais 24
2.4.2 Plano de produção 27
2.4.3 Algoritmo para inserção de encomendas 34
2.4.4 '*Order Promising*' com avaliação de capacidades 38
2.5 Avaliação da capacidade produtiva 39
2.5.1 Planeamento de capacidades 39
2.5.2 Capacidade produtiva 42

2.5.3 Modelação da capacidade 45
2.5.4 Algoritmos de planeamento de capacidade 52
2.5.5 Requisitos para os modelos de capacidade 57
2.5.6 Exemplos de modelos de capacidades 58
2.6 Conclusões 65

Capítulo 3 - Novos Paradigmas Organizacionais 69

3.1 A emergência de novos paradigmas 69
3.1.1 Enquadramento 69
3.1.2 Programas e projectos relevantes 73
3.1.3 Empresa ágil 76
3.2 A *Empresa Virtual* 81
3.2.1 *Empresa Virtual* ou Estendida ("extended") ? 83
3.2.2 Características da *Empresa Virtual* 86
3.2.3 Ciclo de Vida 87
3.2.4 A *Empresa Virtual* e as tecnologias de informação 89
3.3 Sistemas de informação empresariais 90
3.3.1 Contexto 90
3.3.2 Os sistemas ERP, APS e SCM 91
3.3.3 O mercado de ERP 98
3.3.4 O mercado de SCM 98
3.3.5 Conclusão 99
3.4 Requisitos para os Sistemas de Informação 100
3.5 Conclusões 103

Capítulo 4 - Um Sistema de Apoio à Decisão para Aceitação de Encomendas 107

4.1 Enquadramento e domínio do problema 107
4.1.1 Domínio do problema 108
4.2 O sistema *Order Promise* 112
4.2.1 Enquadramento no projecto X-CITTIC 112
4.2.2 Arquitectura geral do sistema X-CITTIC 116
4.2.3 Modelo de objectos 119
4.2.4 Descrição funcional 126
4.2.5 Estrutura e arquitectura geral do sistema *Order Promise* ... 128
4.2.6 *Order Promise Client* (OPClient) 138

4.2.7 *Order Promise Server* 145
4.2.8 *Rough Planner* 151
4.3 Tecnologias e ambiente de implementação 167
4.3.1 Tecnologia de objectos distribuídos 167
4.3.2 Ambiente de implementação 173

Capítulo 5 - Modelos e Abordagens Heurísticas 177

5.1 Introdução; revisão bibliográfica 177
5.2 Estruturação geral do problema 182
5.2.1 Admissibilidade e avaliação das soluções 184
5.2.2 O problema de inserção de ordens 186
5.3 Modelos e casos particulares 188
5.3.1 Notação geral 189
5.3.2 Modelo 1 190
5.3.3 Modelo 2 193
5.4 Meta-heurísticas 196
5.4.1 Estratégia de pesquisa local 200
5.4.2 Geração de soluções iniciais 201
5.4.3 Avaliação das soluções 207
5.4.4 Exemplo ilustrativo 209
5.5 *Simulated Annealing* 213
5.5.1 Referências históricas e bibliografia 213
5.5.2 Estrutura geral do algoritmo 213
5.5.3 Decisões de carácter genérico 217
5.5.4 Decisões específicas ao problema 219
5.5.5 Estrutura de vizinhança 220
5.5.6 Avaliação das soluções 222
5.5.7 Experiências computacionais 224
5.6 Algoritmos Genéticos 242
5.6.1 Referências históricas e bibliográficas 242
5.6.2 Estrutura do algoritmo 243
5.6.3 Codificação 246
5.6.4 Geração da população inicial 248
5.6.5 Mecanismos de selecção 249
5.6.6 Operadores genéticos 252
5.6.7 Tipos de algoritmos genéticos 259
5.6.8 Experiências computacionais 261
5.7 Heurística multi-atributo 278

5.7.1 Considerações gerais 278
5.7.2 Uma abordagem multi-atributo 282
5.7.3 Pressupostos gerais 285
5.7.4 Formulação matemática 286
5.7.5 Restrições 288
5.7.6 Dados de entrada e resultados 289
5.7.7 Descrição geral da heurística 290
5.7.8 Exemplo numérico 291
5.8 Conclusões 294

Capítulo 6 - Uma Abordagem Baseada em Agentes Inteligentes 297

6.1 Agentes inteligentes 298
6.1.1 Conceitos fundamentais e enquadramento 298
6.1.2 *Knowledge Query Manipulation Language* (KQML) 304
6.1.3 *Knowledge Interchange Format* (KIF) 308
6.2 Uma ontologia para a Empresa Virtual 309
6.2.1 A noção de ontologia 310
6.2.2 Objectivos de uma ontologia 310
6.2.3 Tipos de ontologias 311
6.2.4 Ontologia da *Empresa Virtual* (*Virtual Enterprise Ontology*) 312
6.3 Uma arquitectura multi-agente 314
6.3.1 Breve revisão bibliográfica e contexto 314
6.3.2 Arquitectura e tipos de agentes 315
6.3.3 Criação da rede de agentes 318
6.3.4 Organização dos agentes 318
6.3.5 Criação, configuração e organização da EV 320
6.3.6 Operação da EV 324

Capítulo 7 - Conclusões e Desenvolvimentos Futuros 339

7.1 Principais resultados e conclusões 339
7.2 Desenvolvimentos futuros 344

Referências 347

Apêndice A - Problema Teste 361

A.1 Estrutura produtiva 361
A.2 Parametrização dos modelos de capacidade 362
A.2.1 Informação adicional (perfil de capacidade e/ou ATP)..... 362
A.3 Informação logística 366
A.4 Exemplos de respostas dos modelos de capacidades 367

Apêndice B - Extracto da Ontologia VEO (*Virtual Enterprise Ontology*) 373

B.1 Presentation 373
B.1.1 Terms (from Eo v1.1) 373
B.2 Virtual Enterprise Ontology Overview 374
B.2.1 Plan and Order Management 374
B.2.2 Activities and Resources 376
B.2.3 Networked/Extended Organisations 377

Apêndice C - Extracto das Interfaces IDL Definidas 381

C.1 Componente *Order Promise Client* 381
C.2 Componente *Order Promise Server* 382
C.3 Componente *Rough Planner* 384

Lista de Figuras

Figura 2.1 - Tendência dos sistemas de produção 16
Figura 2.2 - Evolução da interacção cliente/produtor 17
Figura 2.3 - Diferentes modos de produção 19
Figura 2.4 - Posicionamento do ponto de desacoplamento no processo produtivo 22
Figura 2.5 - Interacção entre o planeamento de capacidades e o planeamento da produção 40
Figura 2.6 - Modelo conceptual para o planeamento da produção 41
Figura 2.7 - A nova "visão" de capacidade produtiva 44
Figura 2.8 - Taxonomia dos modelos de capacidade 46
Figura 2.9 - Método híbrido de planeamento 51
Figura 2.10 - Interacção operações/recursos 52
Figura 2.11 - Planeamento progressivo: cenários possíveis 53
Figura 2.12 - Planeamento regressivo: cenários possíveis 54
Figura 2.13 - Representação esquemática do modelo de capacidade 59
Figura 2.14 - Resposta do modelo em função do tempo de processamento e da quantidade a produzir 62
Figura 2.15 - Resposta do modelo num cenário de produção de uma quantidade fixa 62
Figura 2.16 - Exemplo de resposta do modelo baseado num perfil de capacidades 63
Figura 2.17 - Exemplo de perfil de capacidade 64
Figura 3.1 - A globalização da economia e a consequente desintegração vertical 71
Figura 3.2 - A emergência da *criação de valor cooperativo* 72
Figura 3.3 - Matriz de 'Fisher' 79
Figura 3.4 - Coexistência dos conceitos 'agile' e 'lean' 80
Figura 3.5 - Cooperação entre entidades autónomas numa *Empresa Virtual* 85
Figura 3.6 - Intensidade e duração da cooperação 86

Figura 3.7 - Ciclo de vida de uma *Empresa Virtual* 88
Figura 3.8 - Funcionalidades dos sistemas ERP/APS/SCM 94
Figura 3.9 - Relações entre sistemas ERP e APS, a vários níveis de planeamento 95
Figura 3.10 - Arquitectura monolítica 97
Figura 3.11 - Tendência dos ERP: arquitecturas baseadas em componentes 97
Figura 4.1 - Processo de fabrico de semicondutores 110
Figura 4.2 - Modelo funcional preliminar do sistema X-CITTIC 113
Figura 4.3 - Módulos funcionais da solução X-CITTIC 114
Figura 4.4 - Posicionamento da solução X-CITTIC 115
Figura 4.5 - Arquitectura "four-tier" do sistema X-CITTIC 117
Figura 4.6 - Arquitectura básica do sistema X-CITTIC 118
Figura 4.7 - Estruturação da informação 120
Figura 4.8 - Modelação da estruturação da informação 121
Figura 4.9 - Diagrama de classes (modelo de objectos) global 122
Figura 4.10 - Diagrama de classes *local* (1ª parte) 123
Figura 4.11 - Diagrama de classes *local* (2ª parte) 124
Figura 4.12 - Modelo funcional preliminar do módulo *Order Promise* 127
Figura 4.13 - Diagrama de estados do módulo *Order Promise* 128
Figura 4.14 - Estrutura geral do módulo *Order Promise* 129
Figura 4.15 - Diagrama de *caso de uso* para o sistema *Order Promise* 130
Figura 4.16 - Representação gráfica (vista estática) do sistema *Order Promise* 131
Figura 4.17 - Diagrama de interacção (vista dinâmica) do sistema *Order Promise* 131
Figura 4.18 - Arquitectura global do sistema *Order Promise* 132
Figura 4.19 - Interacção com os componentes de gestão de informação 134
Figura 4.20 - Tipos de interfaces nos componentes de gestão de informação 134
Figura 4.21 - Diagrama de componentes 136
Figura 4.22 - Cenário de configuração de funcionamento do sistema *Order Promise* 137

Figura 4.23 - Interacção entre componentes 139
Figura 4.24 - Diagrama de classes do componente OPClient 140
Figura 4.25 - Diagrama de estados do componente OPClient 141
Figura 4.26 - Écran de configuração do OPClient 142
Figura 4.27 - Exemplo de entrada de pedidos 143
Figura 4.28 - Exemplo de oferta 145
Figura 4.29 - Diagrama de interacção entre o OPServer e outros componentes 146
Figura 4.30 - Interacção entre o OPServer e o GRC 147
Figura 4.31 - Diagrama de classes do componente OPServer 149
Figura 4.32 - Diagrama de estados do componente OPServer 150
Figura 4.33 - Exemplo de janelas do OPServer 151
Figura 4.34 - Interacção entre o componente de planeamento (RP) e os modelos locais de capacidade (RCM) 152
Figura 4.35 - Heterogeneidade de componentes RCM 153
Figura 4.36 - Interacção síncrona versus interacção assíncrona 157
Figura 4.37 - Protocolo de interacção entre o componente RP e os componentes RCM 159
Figura 4.38 - Diagrama de classes do componente RP 161
Figura 4.39 - Diagrama de interacção entre classes (RP) 162
Figura 4.40 - Diagrama de estados do RP 163
Figura 4.41 -Visualização da configuração da *Empresa Virtual* 164
Figura 4.42 - Monitorização do processo de negociação durante o planeamento 165
Figura 4.43 - Exemplo de uma janela de parametrização do algoritmo de planeamento 166
Figura 4.44 - Configuração da estratégia global de planeamento 167
Figura 4.45 - Conceito de *middleware* 168
Figura 4.46 - OMA (*Object Management Architecture*) 171
Figura 4.47 - Estrutura de um ORB CORBA (Cliente e Servidor) 172
Figura 5.1 - Representação esquemática da estrutura produtiva 183
Figura 5.2 - Soluções possíveis (*routings*) para o problema 185
Figura 5.3 - Interacção entre modelo global e modelos locais 188
Figura 5.4 - Representação de uma solução 201
Figura 5.5 - Ciclo de geração e avaliação de uma solução 208
Figura 5.6 - Configuração do processo produtivo 209

Figura 5.7 - Ofertas recebidas 211
Figura 5.8 - Rota da melhor alternativa 212
Figura 5.9 - Geração de uma solução vizinha 221
Figura 5.10 - Rota parcial associada a uma solução 224
Figura 5.11 - Variação da função objectivo ($T_0 = 1000$; $\alpha = 0.85$; $L = 400$) 227
Figura 5.12 - Custo médio da solução final (instância A) 229
Figura 5.13 - Tempo médio de processamento (instância A) 229
Figura 5.14 - Custo médio da solução final (instância B) 230
Figura 5.15 - Tempo médio de processamento (instância B) 231
Figura 5.16 - Custo médio da solução final (instância C) 232
Figura 5.17 - Tempo médio de processamento (instância C) 232
Figura 5.18 - Diagrama do tipo caixa das soluções iniciais (H-I, H-II e H-III) e soluções finais (F-I, F-II e F-III) 239
Figura 5.19 - Representação das soluções iniciais e soluções finais para cada corrida do algoritmo 239
Figura 5.20 - Diagrama (x,y) da solução final e da solução inicial 240
Figura 5.21 - Diagrama de caixa do número de iterações realizadas 241
Figura 5.22 - Diagrama (x,y) da solução final e do tempo de processamento 241
Figura 5.23 - Ciclo de funcionamento dos algoritmos genéticos 244
Figura 5.24 - Codificação de uma solução 247
Figura 5.25 - Exemplo do mecanismo de selecção tipo "roleta viciada" 250
Figura 5.26 - Operador de cruzamento simples e múltiplo 253
Figura 5.27 - Exemplo de aplicação do operador de cruzamento 255
Figura 5.28 - Exemplo de aplicação do operador de mutação 258
Figura 5.29 - Impacto do grau de substituição da população na solução final (instância A) 264
Figura 5.30 - Tempo de processamento em função do grau de substituição da população (instância A) 264
Figura 5.31 - Impacto do grau de substituição da população na solução final (instância B) 265
Figura 5.32 - Tempo de processamento (instância B) 266

Figura 5.33 - Impacto do grau de substituição da população na solução final (instância C) 267
Figura 5.34 - Tempo de processamento (instância C) 268
Figura 5.35 - Efeito da combinação de operadores de cruzamento 269
Figura 5.36 - Avaliação em função da dimensão da população e do número de gerações (instância A) 271
Figura 5.37 - Tempo de processamento em função da dimensão da população e do número de gerações (instância A) 271
Figura 5.38 - Avaliação em função da dimensão da população e do número de gerações (instância B) 272
Figura 5.39 - Tempo de processamento função da dimensão da população e do número de gerações (instância B) 273
Figura 5.40 - Avaliação função da dimensão da população e do número de gerações (instância C) 274
Figura 5.41 - Tempo de processamento função da dimensão da população e do número de gerações (instância C) 274
Figura 5.42 - Evolução do algoritmo 277
Figura 5.43 - Planeamento envolvendo todos os estágios 279
Figura 5.44 - Planeamento de produtos intermédios 279
Figura 5.45 - Planeamento parcial 280
Figura 5.46 - Rota associada ao planeamento de encomendas no caso geral 281
Figura 5.47 - *Order splitting* num estágio 281
Figura 5.48- Topologia genérica da estrutura produtiva 287
Figura 6.1 - Perspectivas principais dos "agentes inteligentes" (adaptado de (Oliveira, 1998)) 301
Figura 6.2 - Estrutura em camadas da linguagem KQML 305
Figura 6.3 - Cenários de interacção entre agentes usando *performativas* KQML (adaptado de (Finin et al., 1997)) 306
Figura 6.4 - Tipos de agentes 316
Figura 6.5 - Criação da rede de agentes 318
Figura 6.6 - Associação geográfica ou lógica dos agentes 319
Figura 6.7 - Sequência de mensagens trocadas nas fases de criação, configuração e organização da EV (cenário 1) 321

Figura 6.8 - Sequência de mensagens trocadas nas fases de criação, configuração e organização da EV (variante ao cenário 1) 322
Figura 6.9 - Sequência de mensagens trocadas nas fases de criação, configuração e organização da EV (cenário 2) 323
Figura 6.10 - Sequência de mensagens KQML no planeamento centralizado 325
Figura 6.11 - Soluções alternativas no planeamento cooperativo regressivo 328
Figura 6.12 - Soluções alternativas no planeamento cooperativo progressivo 328
Figura 6.13 - Interacções no planeamento cooperativo progressivo 333

Lista de Tabelas

Tabela 2.1 - Exemplo de plano de produção 31
Tabela 2.2 - Resultado do quadro de planeamento após a aceitação de uma encomenda (cenário 1) 32
Tabela 2.3 - Resultado do quadro de planeamento após a aceitação de uma encomenda (cenário 2) 33
Tabela 2.4 - Quadro (parcial) de planeamento 35
Tabela 2.5 - Quadro (parcial) de planeamento (cenário 1) 36
Tabela 2.6 - Quadro (parcial) de planeamento (cenário 2) 37
Tabela 2.7 - Quadro (parcial) de planeamento (cenário 3) 37
Tabela 2.8 - Quadro (parcial) de planeamento (cenário 4) 38
Tabela 2.9 - Parâmetros de configuração do modelo 64
Tabela 3.1 - "Agile" versus "Lean" 78
Tabela 4.1 - Principais classes de objectos 125
Tabela 4.2 - Estados do processo de avaliação da submissão de um pedido 144
Tabela 4.3 - Mensagens entre o módulo de *Order Promise* e o módulo de *Monitorização e Controlo Reactivo* 147
Tabela 4.4 - Mensagens de interacção entre o componente RP e os componentes RCM 158
Tabela 5.1 - Caracterização das instâncias (pedidos a planear) 226
Tabela 5.2 - Resumo dos resultados computacionais para a instância A 230
Tabela 5.3 - Resumo dos resultados computacionais para a instância B 231
Tabela 5.4 - Resumo dos resultados computacionais para a instância C 233
Tabela 5.5 - Parâmetros do algoritmo SA 234
Tabela 5.1 - Instâncias teste (pedidos a planear) 261
Tabela 5.2 - Resultados computacionais (instância A) 265
Tabela 5.3 - Resultados computacionais (instância B) 266
Tabela 5.4 - Resultados computacionais (instância C) 268

Tabela 5.5 - Resumo dos resultados computacionais na avaliação dos operadores de cruzamento 270
Tabela 5.6 - Resumo dos resultados computacionais (instância A) 272
Tabela 5.7 - Resumo dos resultados computacionais (instância B) 273
Tabela 5.8 - Resumo dos resultados computacionais (instância C) 275
Tabela 6.1 - Sumário das palavras-chave da linguagem KQML e respectivo significado 307
Tabela 6.2 - Exemplos de "performativas" 308
Tabela 6.3 - Tipos de *performativas* KQML envolvidas nas fases de criação, configuração e organização da EV 322

Lista de Algoritmos

Algoritmo 2.1 - Algoritmo de inserção de encomendas baseado em informação de ATP 34
Algoritmo 2.2 - Estrutura base do algoritmo *'order promise'* 38
Algoritmo 2.3 - Método de planeamento regressivo e a capacidade infinita 55
Algoritmo 2.4 - Método de planeamento regressivo e à capacidade finita 56
Algoritmo 5.1 - Heurística aleatória para geração de uma solução admissível (I) 204
Algoritmo 5.2 - Heurística para geração de uma solução admissível (II) 206
Algoritmo 5.3 - Heurística para geração de uma solução admissível (III) 207
Algoritmo 5.4 - *Simulated Annealing* 216
Algoritmo 5.5 - Heurística para geração de uma solução vizinha 221
Algoritmo 5.6 - Algoritmo genético genérico 244
Algoritmo 5.7 - Estrutura geral do operador de cruzamento (com dois pontos de cruzamento) 256
Algoritmo 5.8 - Estrutura geral do operador de mutação 257

Capítulo **1**

Introdução

1.1 Enquadramento e relevância do problema considerado

1.2 Objectivos e motivação do trabalho realizado

1.3 Organização da dissertação

1 INTRODUÇÃO

1.1 Enquadramento e relevância do problema considerado

A globalização constitui, sem dúvida, hoje em dia e nas suas diversas vertentes, objecto privilegiado de reflexão e de especulação. Uma ideia chave implícita no conceito de globalização é a de que tudo acontece rapidamente, em simultâneo e de forma cooperativa, independentemente do lugar do mundo em que tenha ocorrido.

A confluência da globalização e dos avanços tecnológicos revolucionou neste final de século o processo de criação de valor, nomeadamente através da transição de uma economia industrial para uma economia baseada na informação. Neste contexto de grande complexidade, instabilidade e incerteza, as mudanças organizacionais têm sido encaradas como um dos principais meios para estruturar e explorar o novo mundo dos negócios.

A história da organização e gestão empresariais mostra-nos, de forma inequívoca, uma constante procura de sistemas e modelos de gestão para conseguir vantagens competitivas reais e capazes de responderem às tendências e solicitações dos mercados. Neste fim de século, as novas realidades económicas tornam definitivamente insustentáveis os padrões organizacionais baseados na estabilidade e na previsibilidade.

A continuada proliferação de tecnologias de informação em todos os sectores de actividade económica, permitiu a transição de uma competitividade centrada em factores de natureza tangível para uma competitividade em que a ênfase é colocada nas questões organizacionais e de coordenação e cooperação.

A emergência de novos paradigmas organizacionais e num sentido lato e generalizado, a emergência da 'Organização Virtual', resultante por uma lado da globalização e por outro da tendência de desintegração vertical como consequência fundamental da cada vez maior concentração das empresas nas suas actividades principais, permitiram a constituição de redes de produção envolvendo fornecedores de matéria-prima e componentes intermédios, unidades de armazenamento, centros de distribuição e operadores logísticos de diferentes tipos.

De uma forma talvez simplificadora, pode afirmar-se que a evolução das redes de empresas tem seguido duas trajectórias principais:

1. verifica-se, por um lado, a intensificação da cooperação entre empresas de pequena e média dimensão, com limitada disponibilidade de recursos (financeiros, tecnológicos, produtivos, humanos, etc.) e com actividades complementares;
2. assiste-se à reestruturação das grandes organizações, numa cada vez maior desintegração vertical, no sentido de se conseguir maior adaptabilidade e agilidade às solicitações do mercado.

As novas formas de organização, nomeadamente a 'Empresa Virtual'[1], fortemente baseadas na cooperação e de duração muito variável, pressupõem a existência de um conjunto de instrumentos de apoio à gestão, com base em tecnologias de informação e comunicação, que suportem adequadamente o seu ciclo de vida, nomeadamente, as fases de criação, configuração, operação e reconfiguração. Mas o que se constata actualmente é que os requisitos destas novas formas de organização, quer ao nível funcional quer quanto à infra-estrutura tecnológica, ainda não são cabalmente satisfeitos pelas soluções disponíveis comercialmente, nomeadamente, pelos sistemas de informação empresarial ERP (*Enterprise Ressource Planning*) e SCM (*Supply Chain Management*). Existe, por isso, um espaço interessante de intervenção, na gestão das redes de empresas, que constitui o quadro geral em que se desenvolveu o trabalho apresentado nesta tese.

1.1.1 Novas exigências no processo de planeamento

Ao nível da produção, uma consequência imediata do estabelecimento de estruturas produtivas em rede, fortemente distribuídas e integradas em cadeias logísticas complexas, é a alteração nos processos de planeamento, de um âmbito estritamente local e centralizado, para um âmbito mais abrangente, onde os aspectos de mediação e de coordenação assumem grande importância. Assim, numa perspectiva essencialmente de produção, as tecnologias de informação devem permitir aos decisores gerir dinamicamente as

[1] A expressão 'Empresa Virtual', refere-se aqui a uma rede temporária de unidades produtivas e operadores logísticos, de carácter em geral temporário e estabelecida com vista a explorar, de forma cooperativa e coordenada, uma oportunidade de negócio associada à produção de um determinado tipo de bens.

capacidades dos diferentes intervenientes, apoiando, por um lado, todo o processo de negociação e aceitação de encomendas e por outro, o planeamento das unidades produtivas.

Existe portanto, ao nível do planeamento, uma forte necessidade de dispor de soluções que, de forma eficiente produzam planos realistas, e estejam adaptadas às novas características do mercado e das estruturas produtivas. Particularmente importantes são os instrumentos que possibilitem, de forma eficiente e em *tempo-real*, apoiar todo o processo de negociação e aceitação de encomendas. O planeamento e avaliação da capacidade produtiva, à escala de toda a cadeia de fornecimento, revela-se igualmente um instrumento decisivo para o sucesso da *Empresa Virtual*.

Os mecanismos de negociação e, num âmbito mais estrito, o planeamento colaborativo em *tempo-real*, a coordenação necessária entre entidades independentes e a disponibilização e gestão de informação global, constituem, na nossa opinião, pilares fundamentais para as funcionalidades que os sistemas de informação de apoio às novas estruturas organizacionais devem suportar. Igualmente importantes serão os aspectos de configuração e reconfiguração (da rede de empresas) e, em particular, a forma como serão conceptualizadas e modeladas, nos sistemas de informação, as entidades que compõem as organizações.

A orientação ao cliente e as limitações dos sistemas actuais

A produção orientada ao cliente, é cada vez mais, uma questão central das empresas industriais. A tendência continuada de redução dos lotes de fabrico e de redução do ciclo de vida associado aos produtos, a imprevisibilidade da procura e a grande variedade dos produtos, bem como a sua cada vez maior personalização, são factores que no limite levarão as empresas a aproximarem-se do paradigma da produção unitária. Naturalmente que esta tendência impõe grandes desafios às empresas industriais, ao nível da organização do processo produtivo e ao nível dos sistemas e processos de planeamento, de gestão e de controlo da produção.

Por outro lado, uma cada vez maior interacção entre cliente e fornecedor (produtor) e a tendência crescente para que um eventual *ponto de desacoplamento*[2] se desloque para montante, coloca igualmente fortes requisitos aos sistemas de planeamento. Cada vez mais, o planeamento da produção tem de ser baseado nas encomendas dos clientes e menos em previsões. Os sistemas actuais de planeamento (por exemplo, sistemas ERP e MRP-II) não se adequam cabalmente a esta nova realidade. Na verdade, não suportam o carácter distribuído do processo produtivo inerente às novas formas de organização, não admitem situações frequentes de replaneamento, decorrentes das necessidades de planear novas encomendas, e não fazem uma avaliação real da capacidade produtiva (naturalmente difícil de calcular, em sistemas complexos e distribuídos). Além disso, o processo de negociação/aceitação de novas encomendas (*order promising*) é baseado exclusivamente em informação de ATP (*Available-To-Promise*), calculada a partir de planos de produção rígidos, não permitindo, dessa forma, a inclusão de novas encomendas com verificação de capacidades.

1.1.2 Área de aplicação

O trabalho de investigação apresentado nesta dissertação foi desenvolvido no âmbito de um projecto europeu que teve como objectivo principal o desenvolvimento de uma metodologia e de um sistema de planeamento e controlo da produção e da logística associada, para redes empresariais complexas[3]. A indústria dos semicondutores constituiu a área de aplicação do projecto. Pelas suas características, trata-se de uma das indústrias mais avançadas tecnologicamente, apresentando uma das mais rápidas evoluções, e sendo de grande exigência em questões de planeamento e de gestão da produção.

As empresas de semicondutores, estando inseridas em mercados de produção globais, envolvendo geralmente diversas unidades de produção, de logística, de armazenamento e de serviços, distribuídas por diferentes países, assim como recorrendo permanentemente a entidades externas em regime de subcontratação, podem ser natu-

---

[2] O ponto de desacoplamento representa conceptualmente, ao nível da produção/planeamento, a charneira entre a produção/planeamento para *stock* (a montante) e a produção/planeamento por encomenda (a jusante).

[3] Projecto Esprit EP20544 – X-CITTIC – *A Planning and Control System for Semiconductor Virtual Enterprise*, financiado pela Comissão Europeia, e que decorreu entre 1996 e 1998.

ralmente conceptualizadas, no âmbito dos novos paradigmas organizacionais, como 'Organizações Virtuais'. Para além do conjunto dos aspectos estruturais referidos, as empresas de semicondutores da área de negócios orientada ao mercado de "Circuitos Integrados para Aplicação Específica" (ASIC's) fabricam produtos à medida do cliente, estando, por isso, sujeitas a uma imprevisibilidade constante e significativa da procura, para além de terem de suportar todas as dificuldades inerentes a ciclos de vida de produtos extremamente curtos.

Pelo facto desta indústria envolver elevadíssimos investimentos e estar inserida num mercado de grande imprevisibilidade, os processos de aceitação de encomendas (*order promising*) e de planeamento e gestão da produção desempenham um papel fundamental, segundo duas perspectivas. Por um lado, a viabilidade do investimento passa pela necessidade de se garantir elevadas taxas de utilização dos recursos produtivos, por outro lado, exige-se resposta rápida e fiável às solicitações do mercado, nomeadamente, ao nível de novos produtos.

1.2 Objectivos e motivação do trabalho realizado

Do exposto anteriormente, resulta naturalmente uma crescente necessidade de satisfazer requisitos de flexibilidade e de resposta rápida na negociação e aceitação de encomendas por um lado, e de apoiar a cooperação e a coordenação entre as entidades produtivas envolvidas, por outro. Na prática, torna-se assim inviável a utilização dos sistemas de informação tradicionais para planeamento e gestão da produção, na generalidade dos casos, rígidos, monolíticos, centralizados e baseados em metodologias e modelos de negócios, desactualizados e desadequados às novas realidades.

Objectivos principais

O objectivo principal do trabalho de investigação que se apresenta nesta dissertação, consistiu na concepção um Sistema de Apoio à Decisão (SAD) para apoiar o decisor, no processo de negociação e aceitação de encomendas em *tempo-real* e com avaliação de capacidades, em estruturas produtivas distribuídas (conceptualizadas como *Empresas Virtuais*). Para tal, foi necessário proceder-se à análise do processo tradicional de negociação e aceitação de encomendas, baseado unicamente em informação

de ATP (*Available-To-Promise*). Igualmente decisivo foi o estudo das características das estruturas produtivas organizadas em rede e a confrontação dos seus requisitos funcionais, ao nível dos instrumentos de gestão e numa perspectiva essencialmente produtiva e logística, com as funcionalidades oferecidas pelos sistemas de informação disponíveis comercialmente.

Outro dos objectivos deste trabalho consistiu na concepção e avaliação de técnicas avançadas de Optimização Combinatória (*meta-heurísticas*) baseadas nomeadamente em "*simulated annealing*" e *algoritmos genéticos*. Estas abordagens foram utilizadas na resolução do problema global de coordenação associado a um ambiente produtivo em rede.

Ao nível conceptual, um dos objectivos do trabalho consistiu na modelação da *Empresa Virtual* e, em particular, no desenvolvimento de uma primeira *ontologia* para este domínio.

Segundo uma perspectiva tecnológica, procurou-se investigar e demonstrar a aplicabilidade da tecnologia emergente dos *componentes distribuídos*, na concepção do SAD.

Motivação

A coexistência de um conjunto de condições institucionais favoráveis e de meios de desenvolvimento adequados, a par com a receptividade e a colaboração de empresas industriais, foi determinante para a realização do projecto de investigação descrito nesta dissertação.

A participação activa do autor em projectos envolvendo de uma forma directa empresas industriais na vanguarda da tecnologia e em sectores de extrema exigência ao nível dos processos de planeamento e gestão da produção, permitiram-lhe antever uma realidade muito para além daquela que é apresentada nos inúmeros canais de informação disponíveis e que, além disso, constituindo um permanente desafio, permitiu manter uma elevada motivação ao longo da realização deste trabalho de investigação.

1.3 Organização da dissertação

A dissertação está organizada em sete capítulos. No *capítulo 1*, que agora se encerra, foram apresentados o enquadramento e a relevância do tema tratado, e os objectivos fundamentais da investigação realizada.

O *capítulo 2* introduz alguns princípios gerais associados ao planeamento da produção. É analisado o processo de aceitação de novas encomendas (*order promising*) com base em informação de ATP (*available-to-promise*), subjacente aos processos e metodologias tradicionais de planeamento, nomeadamente no contexto do sistema MRP-II. A problemática da avaliação da capacidade produtiva, nas suas diversas vertentes, é igualmente apresentada, sendo referidas as principais vantagens e desvantagens dos diferentes métodos. O capítulo conclui com a apresentação de alguns modelos de capacidade, aptos a modelarem entidades de produção, com um nível de agregação muito elevado e segundo uma estratégia de negociação baseada em *propostas* e *ofertas*.

A vantagem competitiva das empresas passa cada vez mais pelo desenvolvimento de novas estruturas organizacionais e novas metodologias de gestão e planeamento, apoiadas fortemente por tecnologias de informação e de comunicação. O *capítulo 3* tem como principal finalidade apresentar o conceito de *Empresa Virtual* como o paradigma organizacional das empresas emergentes, e explorar a sua estreita relação com as tecnologias de informação e comunicação. Começa por fazer-se uma breve introdução ao aparecimento de novos paradigmas organizacionais, salientando, em particular, o conceito de "empresa ágil". Faz-se igualmente uma referência breve a projectos e programas de investigação relevantes nesta área. Após caracterizar o conceito de *Empresa Virtual* e de apresentar diferentes definições propostas na literatura, é identificado um conjunto de requisitos base para as tecnologias de informação e de comunicação, de suporte a este tipo de organização. A caracterização do mercado dos sistemas de informação empresarial é objecto de uma secção deste capítulo, onde se procura igualmente apresentar as principais tendências actuais. Conclui-se o capítulo com a identificação de um conjunto de requisitos para os sistemas de informação de suporte à *Empresa Virtual*.

No *capítulo 4* é apresentado um módulo de um sistema de informação para o planeamento e controlo da produção na indústria dos semicondutores, que permite apoiar a tomada de decisões no exigente e complexo processo de negociação e aceita-

ção de encomendas (*order promising*) que tem lugar em tempo-real e num ambiente eminentemente distribuído, através de uma rápida avaliação da capacidade produtiva (ao longo da rede de empresas) e tendo em conta os aspectos logísticos e de coordenação decorrentes de uma organização produtiva em rede.

Assim, e após caracterizar o domínio para o qual o sistema foi desenvolvido, começa por situar-se o Sistema de Apoio à Decisão desenvolvido, no âmbito do projecto europeu X-CITTIC. Depois de definida a funcionalidade genérica do sistema, são apresentados a sua arquitectura e o modelo de dados, sendo caracterizados os *componentes* que constituem essa arquitectura, e em particular, as interfaces estabelecidas entre os componentes. Atendendo ao facto de se terem usado tecnologias "estado-da-arte", nomeadamente tecnologia de componentes distribuídos sobre uma infra-estrutura compatível com a especificação CORBA, o capítulo conclui com uma referência breve às ferramentas utilizadas na implementação do sistema.

No *capítulo 5* são apresentados modelos e abordagens heurísticas para a resolução do problema de coordenação associado ao planeamento e aceitação de novas encomendas, tendo em conta aspectos produtivos e de logística, em redes de cooperação multi-estágio. Os algoritmos desenvolvidos, integrados no sistema informático apresentado no *capítulo 4*, permitem o planeamento de novas encomendas segundo uma abordagem inovadora e baseada em técnicas avançadas de pesquisa local.

Após ser apresentada uma revisão bibliográfica da investigação relevante nesta área, começa por estruturar-se o problema de planeamento, como um problema de Optimização Combinatória. A aplicação de duas *meta-heurísticas*, baseadas em *Simulated Annealing* e *Algoritmos Genéticos*, à resolução do problema é, de seguida, apresentada pormenorizadamente. Finalmente, na última secção do capítulo, é apresentada uma heurística construtiva específica, baseada numa análise "multi-atributo" do problema.

O *capítulo 6* apresenta uma outra abordagem do problema, baseada em métodos e conceitos desenvolvidos no âmbito da Inteligência Artificial Distribuída, nomeadamente através dos *Sistemas Multi-Agente*. É proposta uma arquitectura de um sistema informático distribuído, sendo igualmente identificados os cenários de interacção mais significativos, expressos através da linguagem de comunicação de agentes KQML. Pela sua importância, no âmbito dos Sistemas Multi-Agente, é igualmente apresentada uma ontologia da *Empresa Virtual* desenvolvida no âmbito deste trabalho.

Finalmente, no capítulo 7, é apresentado um conjunto de conclusões, revendo-se em síntese as principais contribuições conceptuais e algorítmicas para a resolução do problema de planeamento de encomendas em estruturas produtivas distribuídas. São igualmente perspectivados desenvolvimentos futuros para o trabalho de investigação realizado.

Capítulo 2

Planeamento da Produção e Avaliação de Capacidades

2.1 Contexto

2.2 Tipos de sistemas de produção

2.3 Níveis de planeamento

2.4 A negociação de encomendas baseada em ATP

2.5 Avaliação da capacidade produtiva

2.6 Conclusões

2 PLANEAMENTO DA PRODUÇÃO E AVALIAÇÃO DE CAPACIDADES

Neste capítulo, faz-se em primeiro lugar, uma breve apresentação das principais tendências na organização dos sistemas de produção. Seguidamente, é feita uma introdução aos vários tipos de sistemas e são caracterizados os diferentes níveis de planeamento. De seguida, é feita uma análise acerca do processo de aceitação de novas encomendas (*order promising*) a partir da informação obtida directamente do plano de produção actual. Em particular, é apresentado o processo de cálculo da informação de ATP (*available-to-promise*), subjacente às metodologias tradicionais de planeamento, tais como o MRP-II, informação na qual o processo de aceitação de encomendas se baseia.

Pela sua importância na obtenção de respostas fiáveis, aquando do processo de negociação de datas e de quantidades pretendidas pelos clientes, a problemática da avaliação da capacidade produtiva é igualmente objecto de análise no presente capítulo, sendo apresentadas as diferentes técnicas de avaliação utilizadas.

O capítulo conclui com a apresentação de alguns modelos de capacidades, com um nível de simplificação e agregação elevados, adequados a integrarem um sistema de planeamento distribuído e baseado em negociação, que será apresentado em pormenor no capítulo 4.

2.1 Contexto

A especificidade de cada ambiente de produção tem naturalmente de ser tomada em consideração, aquando do desenho e implementação de sistemas de planeamento e controlo de produção (Higgins et al., 1996).

Se, no passado, as ferramentas de planeamento tratavam essencialmente do planeamento de materiais, actualmente a atenção centra-se em torno do consumidor final, dando assim origem aos sistemas de produção orientados ao cliente.

A tendência continuada de redução de lotes de fabrico e simultaneamente a elevada personalização dos produtos, são factores que, no limite, conduzirão à produção "unitária" ou individualizada. A par desta realidade, está igualmente o facto de se as-

sistir, quer à fragmentação, quer à globalização dos mercados e simultaneamente se verificar uma tendência contínua de crescimento ao nível do consumo. A conjugação de todos estes factores promove a mudança inevitável do paradigma da produção em massa para um novo paradigma que se poderia talvez designar-se por *personalização em massa* (*mass customisation*) associado à produção unitária (ver Figura 2.1).

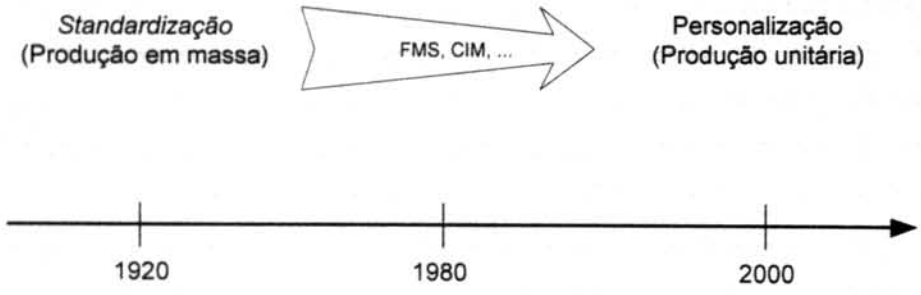


Figura 2.1 – Tendência dos sistemas de produção

Em síntese, pode afirmar-se que actualmente as empresas actuam num ambiente caracterizado por:

- redução drástica do ciclo de vida dos produtos;
- elevada diversidade e sofisticação (complexidade) dos produtos;
- elevada personalização dos produtos;
- procura imprevisível e de grande variabilidade;
- reduzidos prazos de entrega ao cliente.

De facto, a personalização das necessidades dos clientes é actualmente o fio condutor da actividade dos sistemas de produção. Esta tendência da produção orientada ao cliente, e consequentemente a maior interacção entre os sistemas de produção e o cliente, cria grandes desafios e a necessidade de soluções inovadoras, quer ao nível da organização do processo produtivo e das tecnologias utilizadas, quer ao nível dos sistemas e processos de planeamento e controlo da produção.

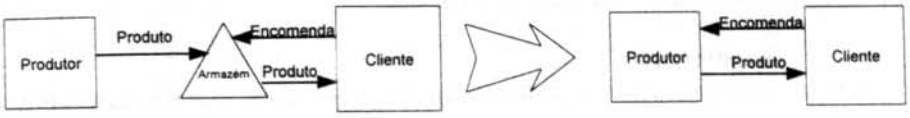


Figura 2.2 – Evolução da interacção cliente/produtor

No passado, a preocupação dos sistemas produtivos era essencialmente o abastecimento de armazéns de produtos finais. O cliente não exercia qualquer pressão significativa sobre o produtor. A tendência actual, emergente, é a integração do cliente final com o produtor, para satisfazer as novas exigências ao nível de prazos de entrega e especificidade dos produtos solicitados. A Figura 2.2 procura ilustrar a evolução verificada na interacção cliente / produtor (unidade de produção).

Em síntese, a tendência de orientação ao cliente dos sistemas de fabrico e consequentemente a produção unitária, impõem nas empresas a adopção de várias práticas, sumariadas nos pontos seguintes (Wortmann et al., 1997):

- obrigatoriedade de fazer "bem" à primeira, tanto o projecto como o fabrico;
- necessidade de intervenção e integração do cliente, em todas as fases do ciclo de vida do produto;
- recursos produtivos universais e flexíveis;
- descentralização dos meios de produção;
- cooperação intra-organização e inter-organizações;
- conceitos, estratégias e ferramentas para manipular a incerteza;
- descentralização da 'engenharia' e da gestão de informação;
- reutilização sistemática da experiência, informação e do conhecimento;
- grau elevado de engenharia concorrente;
- recurso intensivo a tecnologias e estratégias de prototipagem rápida;
- colaboradores de elevada qualidade, criatividade e produtividade intelectual.

2.2 Tipos de sistemas de produção

A classificação dos processos produtivos em diferentes tipos permite, naturalmente, estruturar e inferir muitos aspectos relativos às suas características funcionais e comportamentais. Basicamente, a organização dos sistemas de produção é dependente da natureza e volume de produtos que o sistema deve produzir (tipo de produto, características, sazonalidade do produto, etc.), assim como da tecnologia associada ao processo de fabrico (manuseamento do produto, grau de normalização possível, especificidades de materiais e processos, etc.).

A selecção de uma determinada configuração para o processo produtivo tem variadas implicações para a empresa, em termos da satisfação dos clientes, das características e capacidades da produção, do grau de investimento a realizar, dos custos de produção e exploração envolvidos, e do tipo de planeamento, controlo e gestão a adoptar.

Se o sistema é concebido para produzir um reduzido número de produtos em volumes muito elevados, então é de esperar uma forte sistematização das operações envolvidas, devendo ser dada especial atenção à sincronização das diferentes actividades, assim como ao nivelamento das cargas resultantes. Este contexto (cada vez menos frequente) será normalmente caracterizado por uma elevada especialização do equipamento produtivo, baixa qualificação dos operadores, elevada produtividade, reduzida complexidade ao nível da gestão fabril e, possivelmente, reduzida flexibilidade.

Se, pelo contrário, o sistema é concebido para responder a uma grande diversidade de produtos em volumes reduzidos, então verificar-se-á, relativamente à situação anterior, a inversão das características apresentadas, devendo o processo produtivo ser flexível e orientado para satisfazer encomendas com um elevado grau de especificação por parte do cliente.

Numa outra perspectiva, podem considerar-se dois modelos extremos de produção: produção contínua ou em fluxo, e produção discreta ou intermitente. Associados às especificidades dos produtos e dos mercados, e até do tipo de investimentos e de custos, considera-se habitualmente uma maior variedade de modelos intermédios, que graficamente estão esquematizados na Figura 2.3.

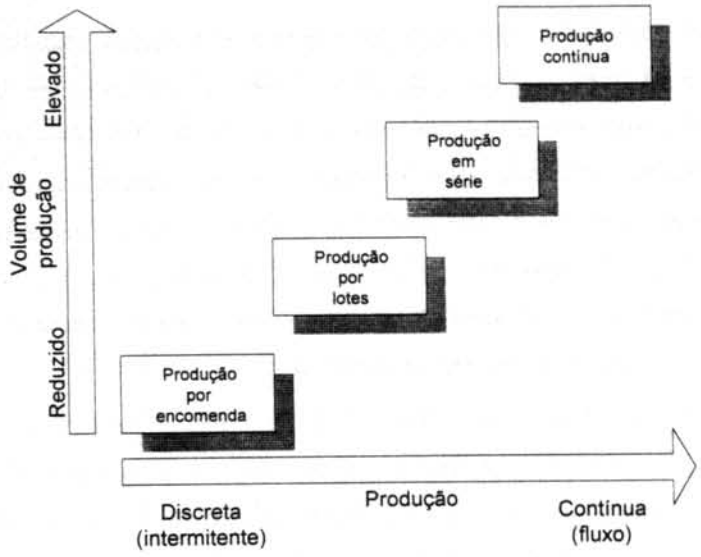


Figura 2.3 – Diferentes modos de produção

Geralmente, associado à evolução da produção discreta para a produção contínua, verifica-se um aumento da complexidade tecnológica do equipamento, com uma maior repetitividade das operações, um aumento do volume de produtos produzidos e uma diminuição significativa da sua variedade. A Figura 2.3 evidencia igualmente a relação existente entre os volumes de produção e a "continuidade" do processo.

A taxonomia dos sistemas de produção pode, por exemplo, ser estabelecida em função dos seguintes critérios (Roldão, 1995):

- quantidades fabricadas e repetitividade;
- grau de continuidade do processo;
- implantação física do processo produtivo;
- tipologia da estrutura dos produtos.
- relação com os clientes.

A classificação por quantidade e repetitividade considera no essencial três grandes tipos de produção: unitária, por lotes e produção em massa.

A classificação dos sistemas de produção segundo o grau de continuidade de processo, origina basicamente três ambientes produtivos: produção contínua (*flow*

shop), produção descontínua (*job shop*) e produção mista. A *produção contínua* (mais tradicional) está usualmente associada a elevados volumes de produção, com uma implantação física em linha, tornando linear o fluxo de produção. A *produção descontínua* caracteriza-se por processar volumes relativamente baixos de variados produtos, em implantações organizadas funcionalmente. O número de operações a realizar por tarefa, a sua afectação aos recursos produtivos e a sua sequência de processamento são variáveis mas conhecidas antecipadamente. No ambiente de *produção mista* coexistem os dois modos de produção anteriormente descritos.

Do ponto de vista de implantação física ("*layout*") considera-se em geral uma classificação que descreve no essencial a disposição e o agrupamento dos recursos produtivos. Podemos, nesta perspectiva, considerar basicamente três modelos organizacionais : modelo *funcional* ou *orientado por processo*, modelo em *linha* ou orientado por produto e modelo de *implantação celular*. Numa disposição funcional, os recursos que realizam o mesmo tipo de operações são agrupados fisicamente. A implantação em linha consiste em organizar consecutivamente todas as operações de produção relacionadas com um determinado produto. A implantação celular associada à tecnologia de grupo, reúne os recursos em pequenos grupos, por forma a obter um compromisso entre uma implantação funcional e uma implantação em linha ou por produto.

Um outro aspecto importante tem a ver com as diferentes estruturas dos produtos, que naturalmente condicionam a especificidade dos sistemas de produção. Identificam-se geralmente duas estruturas principais:

- estruturas convergentes, caracterizadas por incorporação de uma grande variedade de componentes numa reduzida gama de produtos finais;
- estruturas divergentes, onde um reduzido número de matérias-primas originam uma grande variedade de produtos finais.

Na classificação consoante a relação com o cliente, e apesar de se encontrarem modos de produção híbridos, é habitual considerar a seguinte tipologia:

- engenharia por encomenda;
- produção por encomenda;
- montagem por encomenda;
- produção para *stock*.

Ao longo deste texto serão considerados, para a tipologia apresentada, os acrónimos derivados das designações anglo-saxónicas referidas vulgarmente na literatura da área. Assim, designaremos por ETO ("*Engineer -To-Order*") as indústrias de engenharia por encomenda, por MTO ("*Make-To-Order*") as indústrias de produção por encomenda, por ATO ("*Assembly-To-Order*") as indústrias de montagem por encomenda e finalmente MTS ("*Make-To-Stock*") as indústrias de produção para stock.

Note-se que diferentes tipos de relações com o cliente originam diferentes tempos de ciclo ou de resposta na entrega do produto encomendado. Por outro lado, verifica-se igualmente um posicionamento diferente do *ponto de desacoplamento* no sistema de produção. O *ponto de desacoplamento* representa conceptualmente, ao nível da produção, a charneira entre a produção para *stock* (a montante) e a produção por encomenda (a jusante). No caso da produção para *stock*, o processo de planeamento considera essencialmente informação previsional da procura. A partir do *ponto de desacoplamento* para jusante, o processo de planeamento é baseado essencialmente em encomendas firmes.

Por outro lado, quanto maior a interacção do processo produtivo com o cliente (por exemplo, no caso MTO), maior é o deslocamento para montante do *ponto de desacoplamento*. Quanto menor a interacção (por exemplo, no caso MTS), maior é o deslocamento para jusante.

O posicionamento deste ponto estratégico no sistema de produção permite, em primeira instância, definir o valor do tempo mínimo (*lead-time*) para a satisfação de um novo pedido de um cliente. Quanto mais se encontrar posicionado para jusante mais rápida é a resposta aos cliente, mas em contrapartida maior é o risco de obsolescência dos produtos produzidos a partir de planos de produção baseados essencialmente em previsões. Existe, por isso, um compromisso no posicionamento deste ponto no sistema de produção, associado naturalmente a diferentes "atitudes" e organizações do processo produtivo.

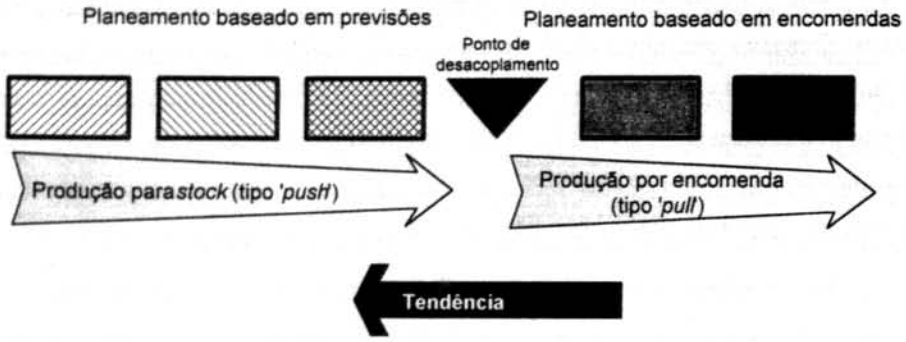


Figura 2.4 – Posicionamento do ponto de desacoplamento no processo produtivo

A Figura 2.4 ilustra a tendência de deslocamento do *ponto de desacoplamento* para montante, associada a uma maior interacção com o cliente.

2.3 Níveis de planeamento

O planeamento da produção envolve decisões complexas, com objectivos múltiplos e normalmente, contraditórios. Em última análise, o que se pretende é garantir que o fornecimento de produtos finais é feito nas quantidades adequadas, nas datas devidas, com a qualidade necessária e a um custo razoável. Naturalmente que, conforme a organização da empresa e o tipo de produtos e de mercado envolvidos, o processo de planeamento da produção resulta mais ou menos complexo.

Na literatura das Ciências Económicas e da Investigação Operacional, podem, entre outras, identificar-se duas abordagens fundamentais à modelação dos problemas de tomada de decisões no âmbito do planeamento da produção: abordagem monolítica e abordagem hierárquica (Bitran and Hax, 1977).

A abordagem monolítica procura descrever completamente o problema de tomada de decisões num único modelo. São consideradas de forma detalhada todas as variáveis de decisão, assim como as suas relações. O modelo resultante é em geral de grande dimensão e complexidade (tipicamente de Programação Inteira Mista). A abordagem hierárquica decompõe o problema num certo número de níveis de decisão.

As decisões de um determinado nível determinam o conjunto de soluções admissíveis (ou seja, constituem restrições) do nível imediatamente inferior.

A aproximação aos problemas de produção por um único modelo (abordagem monolítica) é, na prática, impossível, devido essencialmente à enorme complexidade que resulta de se terem de considerar objectivos múltiplos e vários níveis organizacionais. Além disso, a abordagem monolítica ao planeamento da produção dificilmente traduziria as características "intrinsecamente" hierárquicas da organização (Carravilla, 1996).

Note-se que a decomposição hierárquica assume diferentes significados consoante o contexto considerado. Contudo, subjacente a esta abordagem, está sempre a decomposição ou partição do problema global em sub-problemas que são resolvidos sequencialmente ou simultaneamente, de forma coordenada (Nagi, 1991). As vantagens fundamentais das abordagens hierárquicas são a redução de complexidade e a agregação progressiva de informação dos níveis inferiores para os superiores. Em cada nível hierárquico, as decisões são tomadas com a finalidade de optimizar o modelo considerado nesse nível e tendo em conta as decisões tomadas nos níveis superiores. No nível mais elevado, trabalha-se com entidades muito agregadas, reduzindo-se dessa forma a dimensão do problema.

Existem na literatura inúmeros trabalhos no campo do planeamento hierárquico da produção. A investigação nesta área iniciou-se com os trabalhos publicados por Bitran, Meal e Hax (Hax and Meal, 1975) e, desde então, surgiram inúmeros artigos com aplicações práticas desta abordagem. Em (Bukh, 1994) é feita uma apresentação extensa da bibliografia sobre planeamento hierárquico da produção.

Ainda assim, parece ser interessante apresentar o quadro conceptual tradicionalmente referido na literatura que, apesar de ser bastante genérico, se tem revelado útil num grande número de situações práticas. Assim, de acordo com o enquadramento proposto, podem classificar-se as decisões relativas à produção em três níveis hierárquicos distintos: planeamento estratégico, planeamento táctico e planeamento operacional.

O *planeamento estratégico* tem por objectivo principal situar o sistema de produção no seu ambiente económico global. A partir de estimativas da evolução da procura e considerando um horizonte de planeamento de longo prazo, procura adaptar-se a capacidade global do sistema produtivo às tendências do mercado. Entre as decisões

mais importantes a serem tomadas neste nível incluem-se: a fixação dos objectivos a atingir; a implantação de novos sistemas de produção assim como dos meios de distribuição; o "redesenho" dos sistemas produtivos já existentes; o desenvolvimento de novos produtos; a adopção de novos sistemas de transporte e movimentação, etc. As decisões estratégicas são naturalmente de vital importância na definição competitiva da empresa pelo facto de serem estruturantes e dos seus efeitos se fazerem sentir a longo prazo.

O *planeamento táctico* tem por objectivo gerir o sistema de produção num horizonte temporal de médio prazo. Procura-se, a este nível, afectar de forma eficaz os recursos disponíveis, a fim de satisfazer a procura e as necessidades tecnológicas. O horizonte de planeamento é dividido em diferentes períodos, com uma agregação significativa da informação disponível. Decisões típicas tomadas a este nível são, por exemplo, o recurso a horas extraordinárias e/ou a subcontratação, a alocação de capacidade agregada às famílias de produtos, a constituição de *stocks* sazonais para absorver variações da procura, ou a selecção de alternativas de transporte.

Por último, o *planeamento operacional* tem por objectivo a programação das operações ao nível dos recursos elementares do sistema de produção. Após a desagregação da informação proveniente do nível superior, procura-se sequenciar (escalonar) e lançar as diferentes actividades pelos recursos disponíveis. O horizonte temporal é curto (dependendo do tipo de sistema de produção e dos tempo de produção envolvidos) e normalmente são considerados, neste nível, decisões como o sequenciamento e a calendarização das operações, a definição do tamanho dos lotes e volumes de produção, e a gestão de *stocks*.

2.4 A negociação de encomendas baseada em ATP

2.4.1 Considerações gerais

Independentemente da forma de organização do processo produtivo, na gestão de produção, o Plano Director de Produção (PDP) revela-se um elemento fundamental na coordenação entre o cálculo das necessidades de materiais e dos recursos fabris a curto e a médio prazo, e na articulação com a função comercial nos processos de negociação de

novas encomendas. Se o PDP constitui uma referência para a função *produção*, disponibiliza igualmente a informação necessária ao processo de *negociação e aceitação de pedidos* (potenciais encomendas) provenientes dos clientes (na literatura anglo-saxónica este processo de negociação de encomendas é referido por "*order promising*").

Idealmente o PDP deveria ser a base para a negociação de encomendas, já que reflecte a resposta do sistema de produção à procura prevista (Higgins et al., 1996). A análise do desempenho do PDP assim como a avaliação das práticas correntes no planeamento e controlo da produção foram objecto de estudo do projecto IMPACS (projecto ESPRIT 2338, que decorreu de 1989 a 1992). O projecto concluiu que existia uma necessidade premente de melhoria nos processos tradicionais de planeamento e controlo de produção, tendo procurado resposta, entre outras, às seguintes questões (Higgins et al., 1996):

- Como pode ser efectivamente usado o PDP na gestão do "não previsível"?
- Como pode a metodologia de planeamento baseada no PDP ser melhorada?
- Como deve ser implementado um mecanismo adequado de *order promising*?

A prática corrente revela uma ténue (insuficiente) interacção entre o PDP e a gestão da procura. Esta situação leva a que, muitas das vezes, a negociação de encomendas seja feita sem uma avaliação adequada das capacidades, mas antes com base unicamente em taxas médias de produção. Como consequência, e em virtude da variabilidade do sistema produtivo, a qualidade de serviço é, em geral, reduzida, particularmente no que respeita ao cumprimento das datas e quantidades acordadas.

Tradicionalmente, a negociação de encomendas (quantidades e datas) apoiada pelos sistemas de informação de gestão, decorre segundo uma lógica de cálculo, rígida e com base no conceito de "Disponível-para-Venda" ou, usando a terminologia anglo-saxónica, quantidade ATP (acrónimo de *Available-to-Promise*). O conhecimento dos valores de ATP permite que o processo de negociação de quantidades e datas de entrega decorra dentro de limites temporais de disponibilidade impostos pelo próprio PDP. Isto é, a informação de ATP impõe a 'folga' possível na possibilidade de aceitação de uma nova encomenda, sem provocar qualquer alteração ao plano actual. Assim, no contexto presente, o ATP é usado na actividade de obtenção de datas de entrega a pedidos (potenciais encomendas) de clientes.

Na literatura (ver por exemplo (APICS, 1987) e (Vollmann et al., 1997)), são referidos com frequência dois tipos de ATP: o acumulativo, e o simples ou não acumulativo. Como o próprio nome indicia, o ATP acumulativo define num determinado instante de tempo a quantidade total de produto disponível para venda. O seu valor quantifica o somatório da quantidade disponível para venda desde o instante de tempo actual e o instante de tempo a que se refere. O ATP simples ou não acumulativo exprime o disponível para venda num determinado período futuro pressupondo que o ATP do período anterior terá sido totalmente consumido.

Na negociação de datas e quantidades de um novo pedido, baseada em informação de ATP, podem identificar-se dois cenários principais:

1. Existe, na data pretendida, ATP suficiente para satisfazer a quantidade pedida;
2. Não existe ATP suficiente para satisfação do pedido.

O primeiro cenário envolve basicamente a análise do quadro de planeamento no que respeita aos valores de ATP (simples ou acumulativo). Neste cenário existem algumas variantes, nomeadamente a possibilidade de satisfação da quantidade pedida na data pretendida, por obtenção de quantidades provenientes e disponíveis em períodos precedentes.

O segundo cenário, decorrente do facto de, para a data em questão, não existir suficiente valor de ATP, obriga, ou à construção de um novo plano que considere o novo pedido (que não é ainda uma encomenda!), solução em geral irrealizável devido ao elevado tempo envolvido, ou a um processo que em tempo útil procure a inserção do pedido no plano actual com verificação da capacidade real.

Para ambos os cenários serão apresentados nos pontos seguintes abordagens algorítmicas simples ilustradas com pequenos exemplos numéricos. Uma solução computacional mais elaborada, e adequada à complexidade associada ao segundo cenário, será apresentada com detalhe no capítulo 4.

No âmbito do presente trabalho, parece ser oportuno apresentar, ainda que de uma forma breve, a lógica associada ao cálculo do ATP, a partir de um Plano Director de Produção. Assim, com o propósito de uniformizar os conceitos fundamentais e introduzir a terminologia básica associada a esta problemática, descreve-se, de seguida, o procedimento usual associado à elaboração dum plano de produção, passível de utilização em ambientes produtivos tipos MTS ou MTO.

2.4.2 Plano de produção

O plano de produção estabelece a calendarização das quantidades a produzir de cada produto final[4], num determinado horizonte temporal. Geralmente este horizonte encontra-se segmentado de forma gradual em duas zonas principais: plano firme e plano previsional (embora naturalmente a fronteira entre ambas possa ser pouco precisa). Procura-se com a primeira zona 'congelar', tanto quanto possível, o plano, no sentido de garantir a sua estabilização. O plano firme envolve essencialmente informação proveniente de encomendas firmes, devendo ter uma duração que tenha em conta os tempos indispensáveis para desencadear os processos de produção e aprovisionamento.

Num sistema de produção para *stock* (MTS), o planeamento é baseado essencialmente em previsões de vendas. A elaboração de um plano de produção neste ambiente envolve basicamente a seguinte informação:

- Previsão total da procura para o horizonte de tempo que se pretende planear (*forecast*);
- nível desejado de *stock* do produto a planear no fim do horizonte de planeamento;
- nível actual de *stock* (quantidade disponível de produto, na data actual).

Com base nesta informação, pode calcular-se a produção total necessária para o horizonte de tempo considerado:

ProduçãoTotal = forecast + (nível_desejado_stock - nível_actual_stock)

Supondo uma estratégia de nivelamento da produção ao longo do horizonte de planeamento, calcula-se a produção a lançar por período efectuando o quociente entre a *ProduçãoTotal* e o número de períodos considerados.

Neste tipo de ambiente, a ausência de uma interacção regular e directa com o cliente, em virtude de este adquirir produtos a partir do *stock* existente, diminui o interesse em explorar o plano como base de negociação de potenciais novas encomendas.

Num ambiente de produção por encomenda (MTO), a produção não é orientada pelo nível de *stock* desejado, mas antes pelas encomendas existentes ou que se pre-

---

[4] Ou eventualmente para "componentes" muito próximos do produto final e que darão origem, por processos expeditos de montagem e acabamento, aos produtos finais.

vêem vir a existir. Neste caso, a negociação de novas encomendas pressupõe a análise rigorosa da informação existente no plano de produção. Das decisões tomadas podem resultar efeitos negativos, quer ao nível de instabilidade no plano de produção, quer quanto ao incumprimento de compromissos assumidos aquando de negociações prévias.

Neste contexto, um importante conceito associado a este tipo de ambientes produtivos é o *backlog*. O *backlog* está associado à "quantidade" de trabalho que a empresa tem em carteira. Pode mesmo ser quantificado em dias, semanas ou meses de trabalho. É como que a medida do comprimento da fila de espera para as encomendas em carteira. Há no entanto diferentes interpretações acerca deste conceito que convém aqui clarificar.

Alguns autores consideram o *backlog* igual ao somatório de todas as encomendas firmes de um determinado período. Ou seja, o *backlog* num determinado período é a carteira de encomendas unicamente para esse período (eventualmente acrescidas de encomendas atrasadas - *backorder*).

Uma segunda interpretação consiste em considerar que, num determinado período de tempo, o *backlog* é igual ao somatório de todas as encomendas firmes existentes na empresa (não confinadas unicamente ao período em questão - como na definição anterior). Esta visão de *backlog* é equivalente ao conceito de carteira total de encomendas.

A actualização do *backlog* dever-se-á verificar sempre que existirem novas encomendas ou sempre que se verificar a liquidação parcial ou total das existentes no período.

O procedimento usual para construir um plano de produção, segundo uma estratégia de nivelamento ao longo do horizonte de planeamento, é o seguinte:

- Determina-se a previsão total da procura para o horizonte de planeamento considerado (*forecast*);
- Determina-se, para o produto em questão, o valor actual do *backlog* e especifica-se o valor desejado no final do horizonte de planeamento;
- Calculam-se as necessidades globais da produção, de acordo com a expressão seguinte: $ProduçãoTotal = forecast + (backlog_actual - backlog_final_desejado)$;

- A produção a lançar por período calcula-se efectuando o quociente entre a *ProduçãoTotal* e o número de períodos considerados.

2.4.2.1 Cálculo do ATP

O conhecimento preciso da informação de ATP é fundamental para se conseguir obter o rigor desejado no processo de negociação de uma nova, potencial encomenda. Assim, após o enquadramento feito na secção anterior, vai-se agora analisar o processo de cálculo e de actualização do ATP numa situação genérica de um plano de produção.

Considera-se aqui um horizonte temporal discretizado e com uma granularidade tal que se adequa, quer ao tempo de ciclo de fabricação do produto considerado, quer ao '*lead-time*' acumulado, associado às matérias-primas/componentes necessárias para fabricação. Por outro lado, a definição das quantidades a produzir depende de variados factores, nomeadamente a estratégia de produção seguida pela empresa (quantidade mínima, quantidade económica, etc.). Considera-se, neste contexto, a estratégia, relativamente geral, de assegurar uma quantidade mínima desejável (nível de segurança).

Para cada plano de produção (associado a um produto), considera-se uma representação tabular com as seguintes entradas:

- previsões de vendas (PV) – representam, em cada período, a procura prevista;
- encomendas (*backlog*) – representam as encomendas firmes para o intervalo de tempo considerado (as encomendas firmes consomem parcialmente as previsões de vendas);
- PDP - quantidade de produto disponibilizado pela produção, no início do período considerado (proveniente de ordens de produção a lançar ou já lançadas em intervalos de tempo anteriores de acordo com o '*lead time*' do produto);
- disponível previsional (DP) – representa a disponibilidade de produto no fim de cada intervalo de tempo (subtraído do *stock* de segurança); sempre que desce a zero, implica necessidade de lançar a produção;

- *available-to-promise* (ATP) – quantidade disponível para venda;
- *stock* de segurança (SS) – define a quantidade mínima a considerar em *stock*; o objectivo de manter o DP acima de um determinado nível de segurança é expresso através deste valor.

O ATP exprime o que de facto existe disponível para venda, ou seja aquilo que pode ser oferecido ao cliente. O ATP simples (não acumulativo) é calculado para o primeiro período de tempo e cada vez que se verifica disponibilização adicional do produto (ou seja, sempre que o PDP for não nulo). Considera-se por isso constante o seu valor nos períodos de tempo compreendido entre os instantes previstos para a conclusão de ordens planeadas. No seu cálculo, são consideradas todas as encomendas firmes até ao instante em que se verifica nova disponibilidade de produto (PDP não nulo). O cálculo do ATP é baseado na expressão seguinte:

$$ATP_i = \begin{cases} SS + DP_0 + PDP_1 - \sum_{i=1}^{k:PDP_k \neq 0} backlog_i & se\ i = 1 \\ PDP_i - \sum_{i=i}^{k:PDP_k \neq 0} backlog_i & se\ i \neq 1 \end{cases}$$

em que o índice i representa um período de tempo no horizonte temporal considerado; DP_0 – representa o *stock* inicial.

Assim, para o primeiro intervalo de tempo (e até que se verifique PDP não nulo), o cálculo do ATP é baseado na diferença entre as existências actuais em *stock* e o total das encomendas firmes a satisfazer até ao período de tempo em que o valor do PDP é não nulo. Para os restantes intervalos de tempo, deve partir-se do princípio que todo o ATP anterior pode ter sido consumido, pelo que se considera unicamente as disponibilidades de produto proveniente da produção (PDP).

O ATP acumulativo, como o próprio nome indica, considera que, se a quantidade disponível de produto para venda num determinado período não for, de facto, vendida, transita para o período seguinte. Ou seja, sempre que PDP é não nulo e se procede ao cálculo do ATP, adiciona-se o valor de ATP do período imediatamente anterior.

Com o propósito de esclarecer os conceitos apresentados, apresenta-se seguidamente um exemplo numérico ilustrativo do cálculo de ATP. Tendo por base o exemplo apresentado, serão igualmente apresentados dois cenários de negociação baseados na informação de ATP. No texto, o ATP simples é referido por ATP_I e o ATP acumulativo por ATP_{II}.

2.4.2.2 Exemplo numérico

Considere-se, para um dado produto, uma situação em que o *stock* de segurança é nulo (SS=0), em que a quantidade mínima a lançar em produção é de 25 unidades, em que os períodos considerados são semanas e em que as quantidades associadas ao PDP se encontram disponíveis no início de período respectivo (chegada de produto). Para efeitos de definição da produção (PDP), isto é, de definição das quantidades a produzir, considerou-se o maior valor entre *previsões de vendas* e *backlog* (ver definições anteriores).

Período	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
Previsões de vendas (PV)	10	10	10	10	10	15	15	10	10	10
backlog	8	8	4	4						
PDP	25			25		25		25		25
Disponível previsional (DP)	20	10	0	15	5	15	0	15	5	20
ATP_I	10	10	10	21	21	25	25	25	25	25
ATP_{II}	10	10	10	31	31	56	56	81	81	106

DP[0] =5, SS=0

Tabela 2.1 – Exemplo de plano de produção

O valor do *stock* previsional (DP) foi calculado em cada período da seguinte forma:

$$DP[i] = DP[i-1] + PDP[i] - PV[i]$$

O ATP não acumulativo (ATP_I) foi calculado da seguinte forma, para os períodos em que o PDP é não nulo (não se alterando, para os períodos intermédios):

$$ATP_I[1] = DP[0] + PDP[1] - (backlog[1] + backlog[2] + backlog[3])$$

$ATP_I[4] = PDP[4] - (backlog[4] + backlog[5])$

$ATP_I[6] = PDP[6] - (backlog[6] + backlog[7])$

$ATP_I[8] = PDP[8] - (backlog[8] + backlog[9])$

$ATP_I[10] = PDP[10] - backlog[10]$

Por sua vez, o ATP acumulativo (ATP_{II}) foi calculado da seguinte forma:

$ATP_{II}[1] = ATP_I[1]$

tendo sido, para os restantes intervalos em que PDP é não nulo, utilizada a expressão seguinte:

$ATP_{II}[i] = ATP_{II}[i-1] + ATP_I[i]$

2.4.2.3 Cenário 1

Considere-se a situação representada no exemplo anterior e "simule-se" a chegada de um pedido, para o período 5, de 17 unidades. Será possível satisfazer este pedido?

Por consulta do quadro de planeamento, verifica-se estar disponível um ATP de 21 unidades para os períodos 4 a 5. Logo, é possível satisfazer o pedido não procedendo a qualquer alteração do plano de produção. O quadro seguinte reflecte as alterações entretanto verificadas unicamente nos valores do ATP (tipo I e tipo II). Refira-se a importância do ATP no processo de aceitação de um pedido; na verdade, o seu valor permite 'saber' até onde se pode ir sem ter de se alterar o plano actual.

Período	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
Previsões de vendas	10	10	10	10	10	15	15	10	10	10
backlog	8	8	4	4	17					
PDP	25			25		25		25		25
Disponível previsional	20	10	0	15	5	15	0	15	5	20
ATP_I	10	10	10	(21) 4	(21) 4	25	25	25	25	25
ATP_{II}	10	10	10	(31)14	(31)14	(56)39	(56)39	(81)64	(81)64	(106)89

DP[0] =5, SS=0

Tabela 2.2 – Resultado do quadro de planeamento após a aceitação de uma encomenda (cenário 1)

2.4.2.4 Cenário 2

Considerando os dados inicialmente apresentados, será possível aceitar um pedido de 28 unidades para o período 7?

A consulta do ATP para o período, mostra que existem disponíveis apenas 25 unidades (valor calculado no período 6). Contudo, os valores de ATP tipo II (acumulativo) mostram que se poderá dispor de um total de 56 unidades para o período considerado, se não se verificar o seu consumo nos períodos de tempo anteriores. A satisfação do pedido de 27 unidades poderá ser feita a partir das 25 unidades de ATP existentes no período 7 e de 3 unidades de períodos anteriores. No sentido de prevalecer um funcionamento o mais *'just-in-time'* possível, satisfazem-se as 3 unidades a partir do período imediatamente anterior. A Tabela 2.3 reflecte as alterações verificadas no ATP após a aceitação do pedido.

Período	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
Previsões de vendas	10	10	10	10	10	15	15	10	10	10
backlog	8	8	4	4			28			
PDP	25			25		25		25		25
Disponível previsional	20	10	0	15	5	15	0	15	5	20
ATP_I	10	10	10	(21)18	(21)18	(25)0	(25)0	25	25	25
ATP_{II}	10	10	10	(31)28	(31)28	(56)28	(56)28	(81)53	(81)53	(106)78

DP[0] =5, SS=0

Tabela 2.3 – Resultado do quadro de planeamento após a aceitação de uma encomenda (cenário 2)

Chama-se a atenção para o facto de não se ter procedido a qualquer alteração do plano de produção. A actualização do quadro de planeamento foi feita unicamente para os valores de ATP.

2.4.3 Algoritmo para inserção de encomendas

A informação de ATP permite com facilidade avaliar a possibilidade de satisfazer um novo pedido. Existem, contudo, diferentes possibilidades de inserir uma nova encomenda no plano actual. O Algoritmo 2.1, baseado na informação de ATP, procede à inserção de uma nova encomenda, garantindo igualmente a actualização do ATP resultante. Os parâmetros de entrada do algoritmo são: a quantidade (Q) e a data de entrega pretendida (dd).

Basicamente, o algoritmo começa por verificar a disponibilidade de ATP para o período que contém a data pedida. Se existir quantidade suficiente disponível, é feita imediatamente a inserção da encomenda, procedendo-se de imediato à actualização, quer do *backlog*, quer do ATP. Se no período solicitado não existe suficiente ATP simples mas existe suficiente ATP acumulado, o algoritmo procede a uma alocação distribuída da quantidade pedida, pelos períodos imediatamente precedentes, até se perfazer a totalidade do pedido. Esta situação, similar à apresentada no cenário 2 atrás apresentado, encontra-se implementada computacionalmente através da *recursividade*. A dispersão da satisfação de um pedido por diferentes períodos de tempo é facilmente limitada através do nível de recursividade permitido. A confirmação de encomenda é reflectida, quer no *backlog* do período, quer no ATP.

```
algoritmo insere_encomenda(Q,dd)

    pt = calcula_periodo_temporal(dd); //determina o pe-
    ríodo temporal onde a data pedida se insere

  se (ATP_I[pt] >= Q) então //Existe ATP suficiente?
    backlog[pt] = backlog[pt] + Q; //actualiza backlog
    actualiza_atp(Q, pt);//actualiza ATP
    Retorna SUCESSO;
  senão //tenta períodos anteriores
    se (atp_II[pt] >= Q) então
      insere_encomenda (Q-atp_I[pt], pred(pt));
      backlog[pt] = backlog[pt] + atp_I[pt];
      actualiza_atp(atp_I[pt], pt);
      retorna SUCESSO;
    senão
      retorna IMPOSSÍVEL;
    fimSe
  fimSe
fim
```

Algoritmo 2.1 – Algoritmo de inserção de encomendas baseado em informação de ATP

A lógica de funcionamento deste algoritmo é ilustrada através de um pequeno exemplo numérico, apresentado a seguir. Parte-se de um quadro de planeamento e de diferentes cenários possíveis para satisfação do pedido, e apresenta-se o resultado da inserção total ou parcial do pedido no plano. São considerados quatro cenários, representativos da maioria das situações possíveis:

1. satisfação imediata do pedido por disponibilidade de ATP no período solicitado;
2. satisfação parcial do pedido no período solicitado e quantidade restante no período imediatamente anterior;
3. satisfação parcial do pedido no período solicitado e quantidade restante em diversos períodos anteriores;
4. satisfação do pedido a partir unicamente de períodos que precedem o solicitado.

2.4.3.1 Exemplo numérico

O quadro (parcial) de planeamento considerado é o apresentado na Tabela 2.4. Para facilitar a leitura, são apresentados unicamente as linhas correspondentes à informação de encomendas firmes (*backlog*) e aos valores de ATP_I e ATP_{II} (os períodos de tempo 1, 5, 7, 10 e 13, assinalam os instantes em que o PDP é não nulo e consequentemente os instantes de tempo em que são calculados os valores de ATP_I e ATP_{II}).

Período	**1**	2	3	4	**5**	6	**7**	8	9	**10**	11	12	**13**	14
backlog	8	7	10	12	10	8	7	10	12	10	12	10	12	10
ATP_I	10	10	10	10	5	5	10	10	10	4	4	4	6	6
ATP_{II}	30	30	30	30	35	35	45	45	45	49	49	49	55	55

Tabela 2.4 – Quadro (parcial) de planeamento

2.4.3.2 Cenário 1

É possível satisfazer um pedido de 3 unidades para o período 11?

Este cenário exemplifica a situação de um pedido ser totalmente satisfeito no período de tempo pretendido. O ATP simples no período é de 4 unidades, logo o pedido pode ser satisfeito totalmente a partir da quantidade existente neste período. A aceitação da encomenda implica a actualização do *backlog* de 12 para 15 unidades assim como a actualização do ATP simples e acumulativo de acordo com o apresentado na Tabela 2.5.

Período	**1**	2	3	4	**5**	6	**7**	8	9	**10**	11	12	**13**	14
backlog	8	7	10	12	10	8	7	10	12	10	(12)15	10	12	10
ATP_I	10	10	10	10	5	5	10	10	10	(4)1	(4)1	(4)1	6	6
ATP_{II}	30	30	30	30	35	35	45	45	45	(49)46	(49)46	(49)46	(55)52	(55)52

Tabela 2.5 – Quadro (parcial) de planeamento (cenário 1)

2.4.3.3 Cenário 2

É possível satisfazer um pedido de 8 unidades para o período 11?

Este cenário exemplifica a situação de um pedido ser apenas parcialmente satisfeito no período de tempo pretendido. No período onde se insere a data pretendida, existem apenas 4 unidades disponíveis para venda, contudo o ATP acumulativo apresenta um valor de 49 o que significa que a encomenda pode ser satisfeita a partir de períodos precedentes. Do consumo das 4 unidades no período, resulta uma quantidade remanescente de 4 unidades. O período imediatamente anterior apresenta um ATP de 10 unidades pelo que a inserção da quantidade remanescente pode ser totalmente satisfeita. A Tabela 2.6 reflecte as alterações verificadas após a aceitação da encomenda.

Período	**1**	2	3	4	**5**	6	**7**	8	9	**10**	11	12	**13**	14
backlog	8	7	10	12	10	8	7	10	(12)16	10	(12)16	10	12	10
ATP_I	10	10	10	10	5	5	(10)6	(10)6	(10)6	(4)0	(4)0	(4)0	6	6
ATP_{II}	30	30	30	30	35	35	(45)41	(45)41	(45)41	(49)41	(49)41	(49)41	(55)47	(55)47

Tabela 2.6 – Quadro (parcial) de planeamento (cenário 2)

2.4.3.4 Cenário 3

É possível satisfazer um pedido de 16 unidades para o período 11?

Cenário similar ao anterior mas apresentando um nível de "dispersão" por períodos anteriores superior. Neste exemplo a quantidade pretendida pode ser satisfeita a partir de três períodos de tempo consecutivos. A Tabela 2.7 reflecte as alterações verificadas após a inserção da quantidade pretendida.

Período	**1**	2	3	4	**5**	6	**7**	8	9	**10**	11	12	**13**	14
backlog	8	7	10	12	10	(8)10	7	10	(12)22	10	(12)16	10	12	10
ATP_I	10	10	10	10	(5)3	(5)3	(10)0	(10)0	(10)0	(4)0	(4)0	(4)0	6	6
ATP_{II}	30	30	30	30	(35)33	(35)33	(45)33	(45)33	(45)33	(49)33	(49)33	(49)33	(55)39	(55)39

Tabela 2.7 – Quadro (parcial) de planeamento (cenário 3)

2.4.3.5 Cenário 4

É possível satisfazer um pedido de 7 unidades para o período 12 após se ter aceite o pedido apresentado no cenário 3?

A informação de 'partida' é o resultado obtido no cenário anterior após a aceitação da encomenda. Este cenário exemplifica uma situação de não existir qualquer quantidade de ATP no período solicitado mas existir disponibilidade de produto em períodos precedentes.

A análise do quadro de planeamento nessa situação (ver Tabela 2.7), mostra que não existe ATP nem no período considerado, nem no período imediatamente anterior. Contudo existe ATP acumulativo podendo o pedido por isso ser satisfeito. Desta situação, resulta a actualização do *backlog* e ATP de acordo com o apresentado na Tabela 2.8.

Periodo	**1**	2	3	4	**5**	6	**7**	8	9	**10**	11	12	**13**	14
backlog	8	7	10	(12)16	10	(10)13	7	10	22	10	16	10	12	10
ATP_I	(10)6	(10)6	(10)6	(10)6	(3)0	(3)0	0	0	0	0	0	0	6	6
ATP_{II}	(30) 26	(30) 26	(30) 26	(30) 26	(33) 26	(33) 26	(33) 26	(33) 26	(33) 26	(33) 26	(33) 26	(33) 26	(39) 32	(39) 32

Tabela 2.8 – Quadro (parcial) de planeamento (cenário 4)

2.4.4 '*Order Promising*' com avaliação de capacidades

O algoritmo apresentado anteriormente (Algoritmo 2.1) procura satisfazer um pedido mediante a disponibilidade de ATP. Contudo, o problema que se coloca agora é a avaliação da possibilidade de satisfazer o pedido numa situação em que não existe ATP, ou seja, numa situação não prevista pelo PDP. Neste caso, é necessário proceder-se ou à elaboração de um novo plano, que considere o pedido (potencial encomenda) que se procura satisfazer, ou à inserção da encomenda no plano existente, mas procedendo a uma avaliação da capacidade produtiva global. O Algoritmo 2.2 constitui o mecanismo base deste processo, aqui designado por *order promising* com avaliação de capacidades.

```
algoritmo order_promise
   recebe_pedido_cliente;
   resultado = insere_encomenda(Q, dd);
   se (resultado == SUCESSO)então
      envia_proposta_cliente;
   senão
      rough_planning(Q, dd);
      envia_proposta_cliente;
   fimSe
fim
```

Algoritmo 2.2 – Estrutura base do algoritmo *'order promise'*

Basicamente, começa por avaliar-se a possibilidade de satisfazer o pedido a partir do plano já existente. Assim, é invocado o método de *inserção* apresentado anteriormente e, em acaso de impossibilidade, é desencadeado um mecanismo de *avaliação em tempo-real da capacidade* do sistema produtivo, para verificar a possibilidade de inclusão do pedido em negociação. Este mecanismo de avaliação, a ser apresentado pormenorizadamente no capítulo 4, procura, de forma iterativa e por um processo de negociação e coordenação, obter uma solução para o pedido pendente.

2.5 Avaliação da capacidade produtiva

2.5.1 Planeamento de capacidades

Todos os sistemas de produção têm naturalmente capacidades limitadas. O crescente interesse nos métodos e abordagens associados ao planeamento de capacidades, decorre da inevitável interligação entre o processo de planeamento e o efeito do plano nos recursos envolvidos, normalmente escassos e caros. Uma questão fundamental no desenvolvimento de metodologias de planeamento ou de avaliação de planos é sem dúvida o nível de detalhe aplicável, assim como a granularidade do plano e o horizonte temporal considerados. A maioria dos sistemas de planeamento são estruturados em diferentes níveis hierárquicos, correspondendo igualmente a diferentes níveis de agregação, ao nível da informação envolvida.

A Figura 2.5 apresenta a segmentação hierárquica do planeamento da produção, usualmente referida nas metodologias de planeamento tipo MRP-II e de acordo com o apresentado na secção 2.3. Esta figura permite evidenciar simultaneamente a interacção existente entre o planeamento de produção propriamente dito e o planeamento de capacidades.

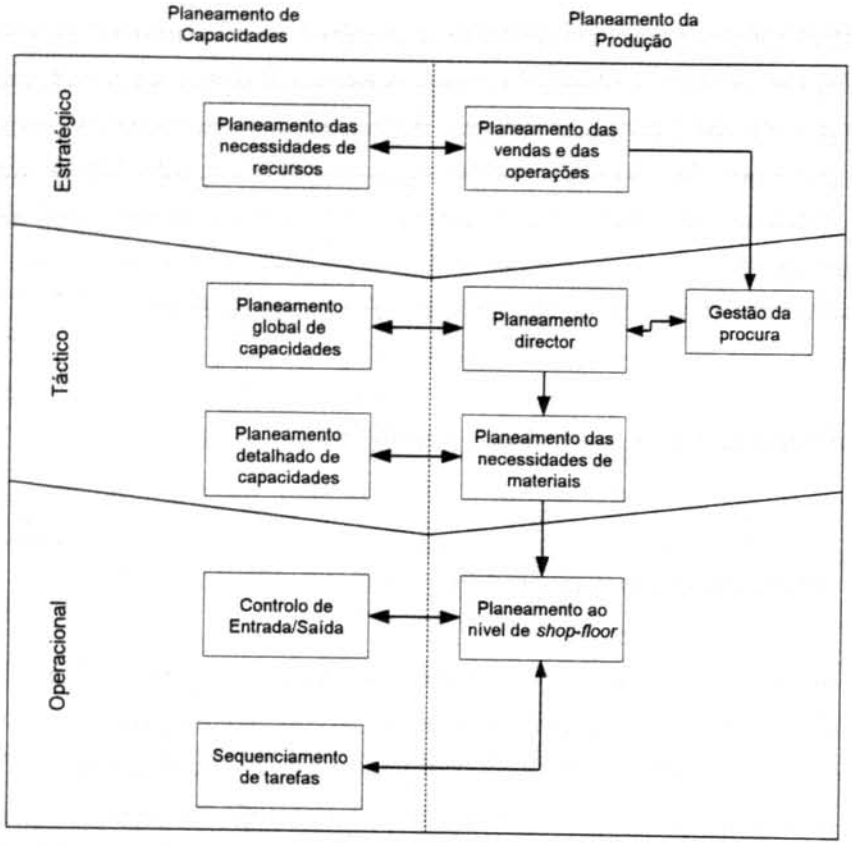


Figura 2.5 – Interacção entre o planeamento de capacidades e o planeamento da produção (adaptado de (Higgins et al., 1996))

O presente trabalho, situa-se essencialmente no nível táctico, pelo que o seu âmbito é o da avaliação de capacidades a esse nível.

Assim e por razões de sistematização, entende-se ser útil considerar um modelo genérico de planeamento da produção. Neste modelo conceptual, identificam-se basicamente quatro entidades directamente intervenientes:

- *encomendas* – conjunto de encomendas firmes e previsionais de produtos finais;
- *recursos* – conjunto de todos os meios produtivos disponíveis;
- *metodologia de planeamento* – define e implementa o método de afectação das tarefas aos recursos (plano);

- *modelo de capacidades* – avalia o efeito do plano no conjunto dos recursos disponíveis.

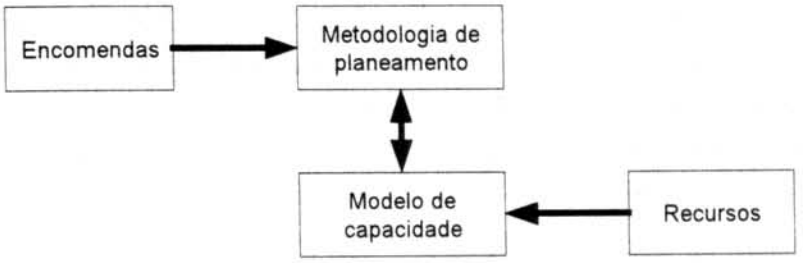


Figura 2.6 – Modelo conceptual para o planeamento da produção

A Figura 2.6 ilustra o modelo conceptual considerado, salientando a interligação entre as diferentes entidades. Este modelo conceptualiza a divisão de um processo de planeamento de produção segundo dois pontos de vista complementares:

- a afectação de tarefas, decorrentes da existência de encomendas, a recursos;
- a avaliação da exequibilidade dessa afectação, assim como o efeito resultante nos recursos envolvidos.

Estes dois pontos de vista têm correspondência directa com o algoritmo de planeamento e com o modelo de capacidades utilizados.

O fluxo bidireccional, ilustrado na Figura 2.6, associado à interacção entre o algoritmo de planeamento e o modelo de capacidades nem sempre é encontrado nos sistemas de planeamento. Assim, por exemplo, no caso do MRP-II, apesar de ser uma metodologia de planeamento em anel fechado, não se verifica de forma autónoma e implícita uma interacção entre o modelo de capacidades e o algoritmo de planeamento. Na verdade, a realimentação é conseguida unicamente com a interacção do planeador, nomeadamente, procedendo à mudança de quantidades, datas de início ou de conclusão associadas às encomendas.

As limitações identificadas nos sistemas tradicionais de planeamento, decorrentes de pressupostos cada vez menos válidos nos actuais contextos produtivos, têm conduzido nos últimos anos ao desenvolvimento de inúmera investigação neste domí-

nio (ver, por exemplo, (Artiba and Elmaghraby, 1997) e (Zweben and Fox, 1994)). Igualmente significativo tem sido o aparecimento de técnicas de optimização, baseadas em estratégias evolutivas e em algoritmos de pesquisa local, originando comercialmente diversos sistemas de planeamento "acoplados" de forma estreita aos sistemas tradicionais de planeamento. Estes novos sistemas, referidos muitas das vezes por "optimizadores", procuram "corrigir" ou "reparar" os planos de produção, gerados pelas metodologias tradicionais, de acordo com algoritmos de optimização e modelos de dados específicos.

Ao nível dos modelos de capacidades, verifica-se que a investigação efectuada tem sido pontual e direccionada para áreas muito específicas. Esta realidade parece decorrer do facto de que os algoritmos e metodologias de planeamento podem ter uma grande abrangência, em oposição aos modelos de capacidade que dependem fortemente das especificidades da unidade produtiva.

2.5.2 Capacidade produtiva

Genericamente, a exequibilidade de um plano de produção pode ser verificada pela confrontação entre as cargas resultantes nos recursos, com as capacidades disponíveis. Tradicionalmente, no âmbito do planeamento da produção, são considerados dois níveis de avaliação de capacidades, com horizonte, granularidade temporal e âmbito diferentes.

Um primeiro nível, denominado na literatura anglo-saxónica como *Rough-Cut Capacity Planning* (RCCP), avalia o plano de produção ao nível da sua exequibilidade quanto às quantidades globais envolvidas, confrontando as cargas globais com as capacidades globais disponíveis, isto é, faz uma estimativa grosseira e de âmbito global.

O segundo nível de avaliação de capacidades determina de forma precisa o efeito do plano em cada recurso da unidade produtiva. Este nível, denominado usualmente na literatura por *Capacity Requirements Planning* (CRP), avalia de forma faseada no tempo e com maior detalhe o efeito não só do plano global como também o resultante das ordens de produção existentes (por exemplo, ordens de produção lançadas mas ainda não executadas) e que consomem capacidades nos diferentes recursos envolvidos.

A apresentação detalhada das diferentes técnicas de avaliação de capacidade utilizadas em produtos comerciais, ao nível do RCCP e do CRP, pode ser encontrada, por exemplo, em (Vollmann et al., 1997) e (Wortmann et al., 1996).

2.5.2.1 Definição de capacidade

Uma questão fundamental quando se fala de planeamento de capacidades é precisamente a definição a adoptar para o conceito de capacidade. Tradicionalmente, a definição de capacidade está associada à quantidade de saída que pode ser obtida num processo, num determinado período de tempo (APICS, 1995), (Vollmann et al., 1997). Uma segunda definição, mais consentânea com a realidade, seria definir a capacidade produtiva como a aptidão de um sistema produzir uma determinada quantidade de produtos diferentes (*mix* de produtos) a uma determinada taxa de produção e num determinado período de tempo.

A definição tradicional de capacidade é adequada a ambientes produtivos repetitivos, com reduzida variabilidade de produtos e com tempos de produção reduzidos. Em situações de processos produtivos complexos, nomeadamente, com situações constantes de fluxos reentrantes, com simultaneidade de produtos diferentes e com longos tempos de produção, o conceito clássico de capacidade revela-se claramente insuficiente, sendo mais adequado considerar a segunda das definições apresentadas (X-CITTIC, 1997).

No âmbito do projecto MIMAC[5] foi estudado pormenorizadamente a temática da capacidade produtiva e, nomeadamente, foram identificados os factores que originam "perda" de capacidade, isto é, os factores que fazem com que a capacidade real disponível seja inferior à capacidade teórica possível. Além disso, este projecto, envolvendo diversas empresas industriais, permitiu constatar que o método mais frequente para avaliação da capacidade produtiva é baseado na medida do tempo médio de ciclo (*lead time*) e na sua relação não linear com a taxa de produção. Ou seja, verificou-se na prática que a definição clássica de capacidade é em certo sentido desadequada e irrea-

---

[5] O MIMAC - Measurement and Improvement of Manufacturing Capacity (JESSI T30C / ESPRIT 8003 / MST –III Theme 3.3) foi um projecto de cooperação envolvendo diversas entidades académicas e empresas industriais de semicondutores (europeias e norte-americanas). O objectivo principal do projecto era identificar e quantificar os efeitos e interacções dos principais factores que causam perda de eficiência produtiva em unidades fabris de semicondutores.

lista, constituindo a segunda "definição" apresentada, o fundamento do método habitual considerado em processos produtivos de grande complexidade (como, por exemplo, o processo de fabrico de semicondutores).

A Figura 2.7 ilustra a evolução observada nos últimos anos, do conceito de capacidade de um recurso ou processo produtivo. A relação não linear entre *o lead-time* ou nível de WIP e a taxa de utilização dos equipamentos produtivos ou taxa de produção é dependente em primeira instância do *mix* de produtos em produção. Em termos práticos, é desejável garantir um determinado valor máximo para o *lead-time*. A imposição deste valor máximo define o nível máximo de utilização para os equipamentos, isto é, restringe a capacidade produtiva disponível. Desta forma, a maneira usual de garantir os tempos de ciclo desejados é precisamente através do controlo do nível de utilização (ou nível de produção) dos equipamentos.

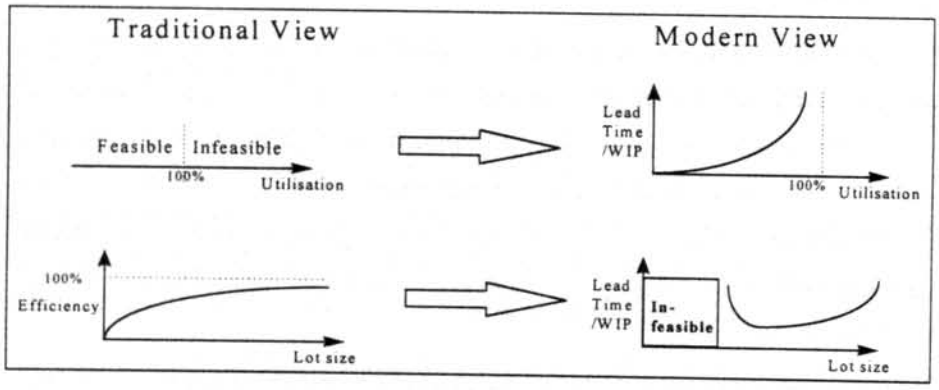


Figura 2.7 – A nova "visão" de capacidade produtiva (fonte (X-CITTIC, 1997))

Matematicamente de uma forma muito agregada, esta abordagem ao problema de avaliação de capacidade poderia ser expressa da seguinte forma:

Seja τ a taxa de produção de um processo produtivo e $C(\tau)$ o tempo de ciclo resultante. Então, τ^*, o valor máximo da taxa de produção que garante que o valor médio do tempo de ciclo não é superior a um limite especificado C^*, é dado por $\tau^* = \max_{\tau \le \mu} \{\tau : C(\tau) \le C^*\}$.

Em (Spence and Welter, 1987) e (Baudin, 1990) é analisada com pormenor a relação não linear entre o tempo de ciclo de produção, ou alternativamente entre o nível de WIP (*work in process*), e a taxa de utilização dos equipamentos produtivos.

Diferentes definições e perspectivas acerca do conceito de capacidade produtiva podem ser encontradas em (Elmaghraby, 1991). Neste artigo, o autor analisa com profundidade o conceito de capacidade produtiva e identifica diversos tipos de capacidade (nominal, operacional, etc) propondo abordagens diferenciadas em função do tipo. Ainda no mesmo artigo, são identificadas e exploradas as principais causas de inexactidão, associadas à quantificação da capacidade, nomeadamente:

- variabilidade, ao longo do tempo do número de produtos simultaneamente em produção, isto é, a capacidade varia com o *mix* de produtos; além disso, os "estrangulamentos" não são necessariamente os mesmos para todos os produtos;
- tempos de preparação (*set-up time*);
- variação da eficiência do processo, com o tipo de produto produzido;
- utilização de subcontratação em partes do processo;
- desperdícios e rejeitados;
- questões sociológicas, culturais e económicas.

2.5.3 Modelação da capacidade

Não se pretende, neste contexto, apresentar de forma pormenorizada as diferentes abordagens disponíveis para modelação de capacidades. Contudo, é naturalmente útil referir as técnicas mais utilizadas, assim como, salientar as suas principais vantagens e desvantagens. Para uma análise mais abrangente, sugere-se a leitura de (Wortmann et al., 1996), (Leachman, 1993) e (Leach and Ristic, 1997).

A modelação de capacidades pode envolver, desde soluções baseadas em regras simples e estáticas, considerando *lead-times* fixos e independentes da carga actual e do *mix* de produtos, até abordagens sofisticadas, baseadas em técnicas de simulação discreta, com modelação individual das diferentes entidades envolvidas.

De igual forma, estas diferentes possibilidades de modelação podem ver a entidade produtiva como um processo estanque, explorando unicamente a relação saída/entrada e não recorrendo, por isso, a informação interna ao próprio processo (*modelação exógena*); ou podem, por outro lado, recorrer a essa informação (*modelação endógena*), nomeadamente, às características dos recursos, ao estado de execução, etc.

Uma taxonomia para os modelos de capacidade, segundo as perspectivas apresentadas, encontra-se ilustrada na Figura 2.8. Referem-se de seguida alguns desses modelos, não de uma forma sistemática, mas procurando antes discutir algumas ideias fundamentais em torno desta problemática.

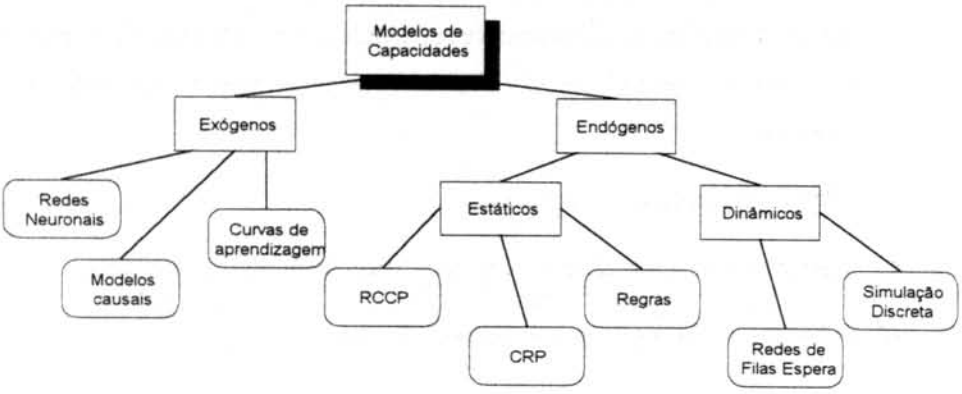


Figura 2.8 – Taxonomia dos modelos de capacidade

2.5.3.1 Modelos estáticos

Os modelos estáticos são dos modelos mais utilizados, pela sua simplicidade de implementação, de afinação e de utilização. São geralmente implementados recorrendo a folhas de cálculo e compreendem um conjunto limitado de regras, relativamente, por exemplo, a tempos e taxas de produção.

Este tipo de modelos exige geralmente poucos parâmetros de configuração, resultando desta constatação a sua fácil configuração, sendo igualmente muito rápidos na produção de resultados. As desvantagens principais, resultantes da evidente inaptidão para lidarem com efeitos dinâmicos inerentes ao funcionamento do processo produtivo, têm a ver com a exiguidade da análise quanto às consequências das decisões

relativamente aos *em-curso*, desempenho das entregas, níveis de *stocks* e níveis de utilização do equipamento. Por outro lado, não consideram a interacção existente entre o *lead-time* real e o nível de carga do processo produtivo.

2.5.3.2 Modelos causais

Os modelos causais são essencialmente modelos preditivos. Basicamente, procuram estimar as datas de conclusão associadas às ordens planeadas ou às taxas de produção possíveis.

No caso dos modelos exógenos, recorre-se frequentemente a métodos quantitativos de previsão, como o método de decomposição e o método de alisamento exponencial (métodos que, pela sua generalidade, são igualmente aplicáveis na gestão da procura). Os modelos causais recorrem extensivamente a informação histórica, procurando estabelecer relações matemáticas (relações causais) entre os factores seleccionados (por exemplo, WIP, níveis de *stock*, taxas de utilização, etc) e as variáveis que se procuram prever (por exemplo, taxas de produção, datas de conclusão, etc). Dado o seu carácter genérico, existe naturalmente, imensa bibliografia sobre estas técnicas, ver, por exemplo, (Chase and Aquilano, 1995), (Buffa and Sarin, 1987) e (Taha, 1992).

Os modelos causais são particularmente importantes na modelação de entidades não directamente controláveis, como por exemplo, na modelação de sistemas produtivos externos à própria organização (unidades de subcontratação), com o propósito de avaliação da capacidade produtiva, por exemplo. As limitações destes modelos resultam do facto de considerarem a informação a um nível muito agregado.

Uma aplicação de um modelo causal, baseado em redes neuronais, a unidades de produção de semicondutores é apresentado em (Westkamper and Schmidt, 1997). Os autores desenvolvem um modelo de capacidades que, a partir de informação histórica dos níveis de *stock* e dos valores reais de *lead-time* de encomendas, permite a determinação do *lead-time* para novos pedidos (potenciais encomendas) de clientes.

Um exemplo simples de um modelo causal consiste em relacionar o *lead-time* (LT) com o nível de WIP. Se se considerar, com base em informação histórica, que $LT=(WIP/100)^2$, então para se garantir um LT de 9 dias, o sistema produtivo não pode ser "carregado" com mais do que o equivalente a um WIP de 300 unidades.

2.5.3.3 Modelos de perfis de capacidade

São modelos endógenos que decompõem com grande detalhe os recursos envolvidos no sistema produtivo. Nestes modelos incluem-se as técnicas tradicionais de avaliação de capacidade RCCP e CRP.

Em (Vollmann et al., 1997) e (Wortmann et al., 1996) são apresentadas as principais técnicas utilizadas na implementação destes modelos. As suas principais limitações resultam do facto de recorrerem intensivamente a informação estática e considerada invariável no tempo (por exemplo, informação de *lead-times*, *mix* de produtos, etc).

2.5.3.4 Modelos baseados em curvas de aprendizagem

A teoria das "curvas de aprendizagem" apresenta uma vasta gama de aplicações, podendo ser utilizada para estimar o tempo e os custos de produção (Chase and Aquilano, 1995).

Basicamente, uma curva de aprendizagem estabelece uma relação matemática mais ou menos complexa, dependendo da técnica usada, entre o tempo necessário para produzir uma determinada quantidade de um produto e um conjunto de factores, tais como, taxas iniciais e nominais de produção, constante de tempo associada ao próprio modelo, etc. As curvas de aprendizagem baseiam-se em três pressupostos principais (Chase and Aquilano, 1995):

- o tempo necessário para produzir um produto será tanto menor quanto maior for a quantidade produzida;
- o tempo por unidade produzida decresce a uma dada taxa;
- a redução do tempo segue um padrão previsível.

Na subsecção 2.5.6.1 descreve-se um modelo de capacidade baseado em curvas de aprendizagem, desenvolvido no âmbito deste trabalho.

2.5.3.5 Modelos dinâmicos

Os modelos dinâmicos recorrem a informação actual do processo produtivo para estimarem variados factores que, nas abordagens anteriores, eram considerados invariáveis.

Assim, por exemplo, os tempos de produção, considerados nestes modelos, passam a depender do estado actual da carga em cada um dos recursos.

Dentro dos modelos dinâmicos podemos considerar, ainda, duas abordagens diferentes. A primeira engloba a denominada *simulação discreta por acontecimentos* (*event discret simulation*). Nesta abordagem, é desenvolvido especificamente um modelo "executável" para o problema em análise, procurando representar as entidades do mundo real (por exemplo, os recursos produtivos). O número de variáveis é geralmente elevado, conduzindo, por norma, a tempos de execução elevados.

A segunda abordagem no âmbito dos modelos dinâmicos, inclui a denominada *macro-simulação* (Leach and Ristic, 1997). Estes modelos implementam uma abordagem similar à empregue nos modelos de perfis de recursos, não considerando contudo, como constantes e invariáveis factores tais como, tempo de espera, *lead-times*, etc. Recorrem, geralmente, à teoria das filas de espera para modelarem individualmente cada um dos recursos, partindo, normalmente, do estado actual do processo produtivo. Esta abordagem, relativamente à *simulação discreta por acontecimentos*, apresenta vantagens significativas, quanto ao número de variáveis envolvidas assim como quanto ao tempo necessário para produzir uma resposta.

Um modelo baseado nesta abordagem é apresentado em (Leach and Ristic, 1997). Trata-se de um modelo para um processo de grande complexidade e com uma estrutura tipo *job-shop*, em que os tempos de ciclo dependem, de forma não linear, do nível de carga do processo produtivo (WIP) e do *mix* de produtos. Esta situação levou os autores a explorar um modelo de capacidade finita baseado, não no conceito tradicional de capacidade disponível num determinado período de tempo, mas antes no valor do tempo de ciclo máximo permitido.

Na avaliação do efeito de uma nova encomenda no processo produtivo, para cada recurso modelado e no período de tempo considerado, o sistema calcula o tempo de ciclo previsto e compara o valor obtido com o valor máximo permitido nesse período. Em caso desse limite ser ultrapassado, o sistema tenta planear a produção no período imediatamente precedente. Desta forma, e através da modelação dos recursos com filas de espera, com comportamentos similares aos representados pelas curvas *tempo de ciclo = f(WIP, mix)*, o modelo de capacidade estima os tempos de ciclo, o nível de WIP, assim como o nível de utilização dos recursos produtivos.

2.5.3.6 Outras abordagens

A tipologia apresentada para os modelos de capacidades, resulta de um trabalho de levantamento e síntese realizado pelo autor e constitui naturalmente uma simplificação que não cobre, exaustivamente, todas as abordagens possíveis na prática. Por exemplo, quando os modelos causais incluem, de forma individualizada, recursos e intervalos de tempo, transformam-se em modelos baseados em perfis de recursos. Da mesma forma que, se aos modelos baseados em perfis de recursos se adicionarem factores dinâmicos, tais como *lead-times* dependentes do nível de utilização dos recursos, passa-se a ter modelos dinâmicos tipo macro-simulação. Estes últimos, quanto mais detalhe considerarem (menor grau de agregação), mais se aproximam dos modelos de eventos discretos.

Em situações específicas, alguns investigadores têm desenvolvido modelos híbridos, procurando explorar a coexistência de diferentes técnicas, por forma a conseguir-se resultados simultaneamente mais precisos e em tempo útil.

Em (Hung and Leachman, 1996), é apresentada uma metodologia de planeamento de produção, para uma linha de fabrico de *wafers* (indústria de semicondutores), baseada num processo iterativo de optimização por Programação Linear (PL) e Simulação Discreta. Os autores começam por apresentar um *modelo de programação linear* do processo produtivo, com base em "perfis de capacidades". Consideram um horizonte de planeamento discretizado, definindo, para cada período de tempo, os níveis de procura, de capacidades e taxas de produção, e assumindo como constantes os tempos de produção, de espera e de transporte. Esta hipótese implica naturalmente, tempos fixos de chegada a cada ponto do processo produtivo. As dificuldades de modelação da relação não linear entre tempo de ciclo e taxa de utilização dos recursos produtivos, conduziu os autores a considerarem uma resolução do problema de PL com tempos constantes, mas de forma iterativa. Basicamente, o que é feito é usar os resultados da resolução do modelo de PL como entrada de um *modelo de simulação discreta* para prever os tempos de ciclo. Estes tempos são usados como constantes na próxima "corrida" do modelo de PL.

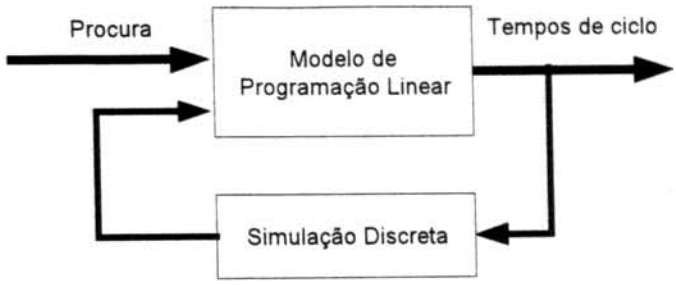


Figura 2.9 – Método híbrido de planeamento

A Figura 2.9 ilustra este processo iterativo. Nas experiências realizadas com dados reais, os autores conseguem ao fim de cinco iterações, um desvio médio de apenas 5% para os tempos de ciclo (comparação entre as estimativas dadas pelo método e os dados reais).

A aplicação de um sistema de simulação adaptativo à previsão dos *lead-times* de um processo produtivo é apresentada em (Westkamper and Schmidt, 1997). Os autores investigam a utilização de um modelo causal, baseado em redes neuronais artificiais, na determinação dos *lead-times* a partir de informação histórica dos níveis de WIP. Recorrem a técnicas de simulação discreta para treinarem a rede neuronal, após o que esta fica apta a ser usada como modelo de capacidades. Em regime estacionário, a estrutura interna e configurável da rede é constante. Este modelo de capacidades recebe como entrada um plano de produção para diferentes produtos e gera na saída os valores previsíveis dos *lead-times* associados às quantidades definidas no plano.

Finalmente, refira-se que em (Leachman, 1993) são analisadas várias técnicas, baseadas ainda em Programação Linear, para o planeamento da produção e modelação de capacidades de unidades de produção de semicondutores. O autor apresenta igualmente uma heurística para o problema global do planeamento multi-fábrica, baseada na decomposição do problema global num conjunto de problemas de dimensão inferior.

2.5.4 Algoritmos de planeamento de capacidade

Pela importância que os "modelos de capacidade" representam nos sistemas de planeamento, apresentam-se de seguida, ainda de que forma sucinta, os principais algoritmos usualmente utilizados na prática.

O planeamento de capacidade envolve no essencial dois tipos de entidades principais: operações produtivas e recursos produtivos. Com o propósito de se conceptualizar as interacções entre estas entidades, começa por apresentar-se, usando a notação UML[6], um modelo simplificado dessas interacções (Figura 2.10).

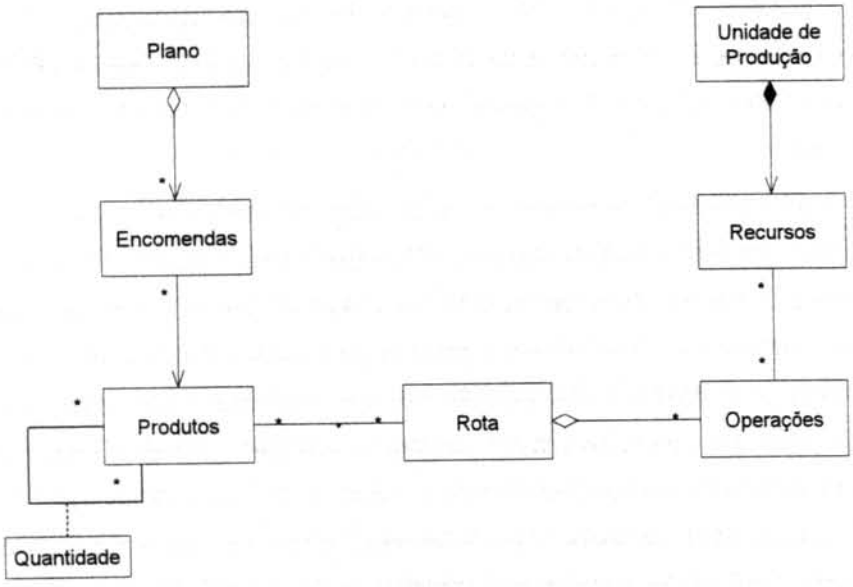


Figura 2.10 – Interacção operações/recursos

Os modelos de capacidade, como referido em 2.5.1, avaliam o efeito da execução de um *plano* sobre os recursos existentes na unidade de produção. No caso presente, considera-se como sendo um *plano*, uma agregação de um conjunto de *encomendas* (*ordens*) planeadas sobre um horizonte temporal discretizado e constituído por períodos de igual duração. As *encomendas* encontram-se relacionadas com *produtos*, definindo exigências ao nível de quantidade e tempo de produção necessário. Por sua vez,

---

[6] *Unified Modelling Language* (Booch et al., 1999).

associado a cada *produto*, encontra-se definida uma ou mais *rotas*, definindo um conjunto de *operações* a efectuar em *recursos* associados a uma determinadas unidades de produção.

O algoritmo de planeamento de capacidades, de acordo com uma dada estratégia, vai associar operações e recursos, definindo assim soluções e, calculando o valor de diversas medidas de desempenho que possibilitam, no fim, avaliar qualitativamente o plano.

É habitual encontrar-se duas metodologias principais associadas ao sentido em que é executado o planeamento, relativamente ao tempo: abordagem progressiva (*forward planning*) e abordagem regressiva (*backward planning*). Em cada uma destas abordagens, é ainda possível considerar a divisão entre planeamento à capacidade finita ou infinita.

A Figura 2.11 ilustra os dois cenários possíveis na abordagem progressiva, quando se procura planear uma determinada operação num recurso produtivo. Num caso a produção é concluída antes da *due-date*, tendo lugar uma "folga" positiva, e no outro, a data de conclusão ultrapassa a *due-date* originando uma "folga" negativa.

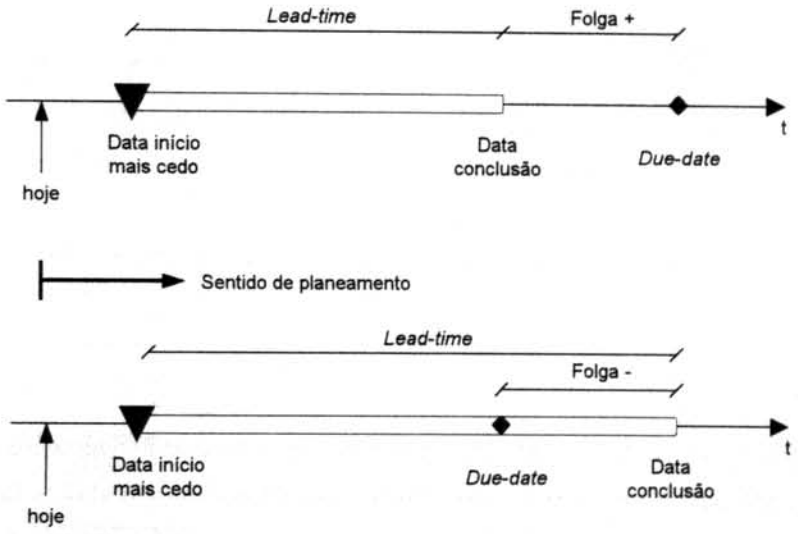


Figura 2.11 – Planeamento progressivo: cenários possíveis

Na abordagem progressiva, a realização do plano é sempre possível mas, naturalmente, e como ilustrado na parte inferior da figura, nem sempre se consegue a conclusão antes da data "devida" (*due-date*).

A abordagem regressiva planeia do fim (isto é, a partir da *due-date*) para o princípio, não garantindo, no entanto, que se obtenham soluções realizáveis ou admissíveis. Na verdade, situações existem em que as datas planeadas para o início das operações são anteriores à data actual, resultando, inevitavelmente, numa impossibilidade de realização do plano (a menos que se alterem a disponibilidade dos recursos envolvidos). A Figura 2.12 ilustra os cenários possíveis nesta abordagem.

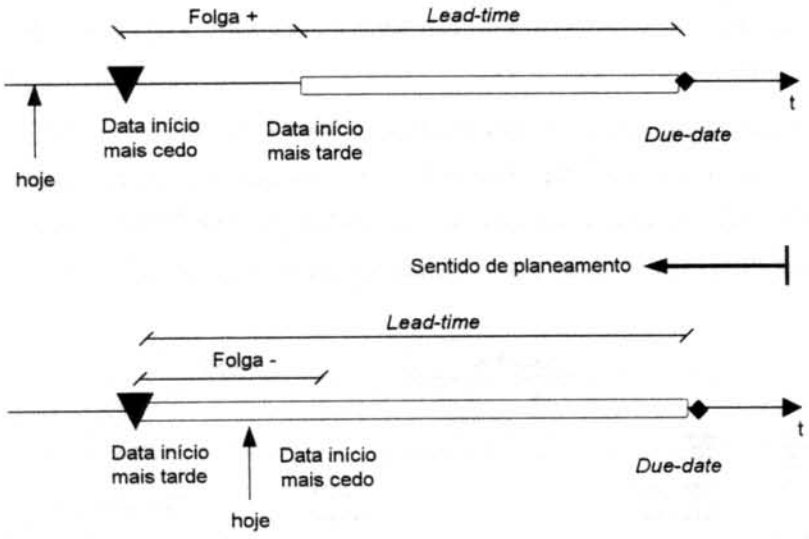


Figura 2.12 – Planeamento regressivo: cenários possíveis

Algoritmos

Apresentam-se de seguida dois algoritmos de planeamento regressivos (um considerando capacidade finita e outro infinita). Por razões de simplicidade e facilidade de leitura, a estrutura apresentada contempla unicamente o planeamento de uma ordem/produto, sendo, contudo, trivial a extensão para o caso de se pretender determinar, de uma forma sequencial, um plano composto por inúmeras ordens/produtos.

Notação utilizada

J_P	Conjunto de operações associadas à produção de um produto P;
R	Conjunto de recursos produtivos existentes;
E_D	Data de início mais cedo (possível) para o produto P;
D_D	*Due-date* associada ao produto P;
H	Duração de um período de planeamento;
T_{jp}, T_{jt}, T_{jw}	Tempo de processamento, tempo de transporte e tempo de espera associadas à operação j pertencente a J_P;
t	Contagem de tempo num determinado período de planeamento;
h	Período de planeamento;
r_j	Recurso seleccionado para executar a operação j;
c	Variável booleana (0 – falso, 1 – verdade);
j	Operação em planeamento.

```
algoritmo regressivo_cap_infinita

 t ← 0;
 h ← int(D_D / H); // período onde se insere a due-date
 j ← UltimaOperação(J_P);

 enquanto existir operações a planear
  t ← t+T_jp+T_jt+T_jw;

  se t>H então
    se h≤0 ou h<E_D então
      fim(INSUCESSO)
    senão
      h ← h-1;
      t ← 0;
    fimSe
  fimSe
 j ← Predecessor(j);
 fimEnquanto

 fim(SUCESSO)
```

Algoritmo 2.3 – Planeamento regressivo a capacidade infinita

Naturalmente que no algoritmo regressivo a capacidade infinita (Algoritmo 2.3), o planeamento é feito sem qualquer verificação de capacidade nos recursos. Contudo, verifica-se um "insucesso" no planeamento, sempre que a data de início for inferior à data actual ou inferior à data de início mais cedo possível para o produto em planeamento.

```
algoritmo regressivo_cap_finita

 t ← 0;
 h ← int(D_D / H); // período onde se insere a due-date
 j ← UltimaOperação(J_P);

 enquanto existir operações a planear
  r_j ← SeleccionaRecurso(j);

  repete
     c ← ExisteCapacidade(r_j,j,h);
     se c=0 então
        c ← ExisteRecursoAlternativo(r_j,j);
        se c=0 e (h≤0 ou h<E_D) então
           fim(INSUCESSO)
        senão
           h ← h-1;
           t ← 0;
        fimSe
     senão
        InsereOperação(r_j,j);
        t ← t+T_jp+T_jt+T_jw;
     fimSe
  enquanto c=0

  se t>H então
     se h≤0 ou h<E_D então
        fim(INSUCESSO)
     senão
        h ← h-1;
        t ← 0;
     fimSe
  fimSe
  j ← Predecessor(j);
fimEnquanto

fim(SUCESSO)
```

Algoritmo 2.4 – Método de planeamento regressivo e à capacidade finita

No algoritmo regressivo a capacidade finita (Algoritmo 2.4), ao planear cada operação num determinado recurso, é feita uma verificação de capacidade. Se para um recurso, num determinado período de tempo não existir capacidade suficiente, o algoritmo procura um recurso alternativo e, no caso de não existirem recursos alternativos, planeia a operação no período de tempo imediatamente anterior.

Neste algoritmo, a função *ExisteCapacidade(r_j, j, h)* avalia se é possível realizar uma determinada operação j, num determinado recurso r_j, num determinado período de tempo h. A estratégia de avaliação pode restringir-se apenas à comparação entre o tempo "produtivo" disponível no período, com o tempo necessário para processar a operação, ou pode "encapsular" estratégias mais complexas como, por exemplo, o recurso a filas de espera para modelação do recurso. Existindo capacidade suficiente, a operação em causa é alocada ao recurso seleccionado, no período de tempo h. A função *InsereOperação(r_j, j)* é responsável por fazer essa alocação, garantindo igualmente a actualização da carga resultante (isto é, consumindo capacidade). De forma idêntica ao caso anterior, verifica-se "insucesso" sempre que o tempo actual for inferior à data de início mais cedo definida.

2.5.5 Requisitos para os modelos de capacidade

Na prática, o planeamento da produção num ambiente que seja distribuído quer geograficamente quer em termos de controlo (redes de produção), impõe um conjunto de requisitos à avaliação da capacidade, geralmente não contemplados pelos sistemas tradicionais.

Por um lado, ao nível global e táctico, é necessário um mecanismo que assegure respostas fiáveis e atempadas aos pedidos dos clientes. Além disso, procura-se, a este nível, a optimização do fluxo produtivo e a coordenação das unidades envolvidas.

Por outro lado, e ao nível local, são necessários modelos de capacidades precisos e fiáveis, já que os compromissos com os clientes, estabelecidos ao nível global, são baseados nos resultados destes modelos. Além disso, a interdependência entre os diferentes nós da rede de produção requer que os planos locais apresentem um grau de precisão elevado, sob pena de comprometerem as decisões, quer a montante quer a jusante.

Um outro requisito, determinante no desenvolvimento destes modelos de capacidades, está associado à sua eficiência computacional. Deve-se naturalmente procurar um compromisso entre a qualidade das soluções e o tempo de execução dos algoritmos, que tenha em conta as exigências do processo de planeamento, em termos de tempo de resposta.

Outra característica não menos importante é a actualização dinâmica do próprio modelo. Isto é, o modelo deve ser dinâmico, no sentido de que os seus parâmetros possam ser modificados como consequência da ocorrência de alterações, quer dos recursos disponíveis (falhas de equipamento, etc), quer das condições actuais de fabrico, nomeadamente '*mix*' de produtos, nível de WIP, dependência do *lead-time* e do WIP com o grau de utilização do equipamento, etc.

Refira-se finalmente que, com o objectivo de implementarem um procedimento baseado no conceito de proposta/oferta, os modelos devem poder funcionar de uma forma "incremental", isto é, devem poder avaliar o efeito de um novo pedido (gerando a correspondente proposta) sem terem que reavaliar todos os pedidos anteriormente confirmados.

2.5.6 Exemplos de modelos de capacidades

Foram referidos anteriormente vários métodos e abordagens gerais ao problema da modelação de capacidades. Nesta secção, descrevem-se diferentes modelos de capacidades, configuráveis, que, em geral, apresentam um elevado grau de simplificação e agregação da informação e que estão aptos a modelar diferentes estratégias locais de produção (por exemplo, planeamento regressivo ou planeamento progressivo). Cada instância de um modelo particular procurará reflectir as características próprias de um dado nó da rede de produção.

Os modelos desenvolvidos são constituídos por quatro blocos principais (ver Figura 2.13). Cada um dos blocos é responsável pelo comportamento do modelo, segundo os eixos associados a cada *item* da resposta produzida, nomeadamente:

- determinação da quantidade de produção possível (prometida) para a *due-date* pedida (QTP);
- determinação da quantidade previsível para o estágio predecessor (InQ);

- determinação do *lead-time* médio esperado para a quantidade prometida (LT);
- cálculo do custo global associado (**$$$**).

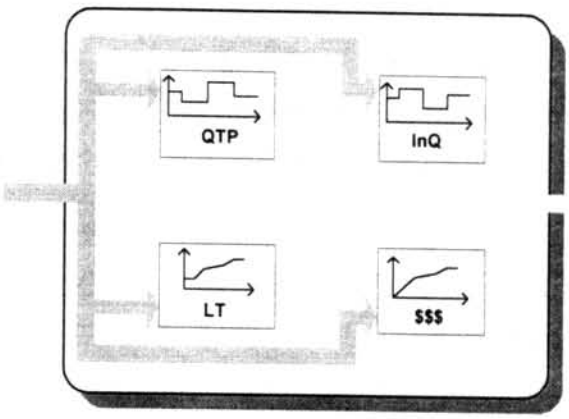


Figura 2.13 – Representação esquemática do modelo de capacidade

Basicamente, os blocos implementam funções de transferência excitadas, quer pelos dados de entrada fornecidos ao modelo, quer simultaneamente pelos resultados de saída de outros blocos. Existe, assim, interdependência entre as diferentes funções de transferência.

Note-se que os modelos consideram um horizonte temporal deslizante, de comprimento fixo e dividido em intervalos de tempo (períodos) iguais. Os modelos são configuráveis, de forma independente, por um conjunto de parâmetros, quer dependentes do tempo (esta característica permite modelar diferentes comportamentos do modelo em situações dependentes unicamente do instante de tempo em que são executados), quer modificáveis por factores externos, representativos do contexto que o modelo procura de forma agregada representar.

Descrevem-se, de seguida, os vários modelos desenvolvidos no âmbito deste trabalho e implementados computacionalmente.

2.5.6.1 Modelo baseado em curvas de aprendizagem

Neste modelo recorre-se a uma equação resultante da aplicação da teoria das curvas de aprendizagem (Chase and Aquilano, 1995), para estimar o tempo de produção associado ao fabrico de uma determinada quantidade de um dado produto.

De acordo com a taxonomia apresentada anteriormente na secção 2.5.3, este é um modelo causal, em que se relaciona o tempo de produção com a quantidade a produzir e a tendência monótona crescente das taxas de produção.

A equação utilizada para calcular o tempo estimado de produção de um lote de dimensão Q é:

$$t_e = \frac{Q + Y_f \cdot T\left(1 - e^{\frac{-t_e}{T}}\right)}{Y_i + Y_f}$$

onde

t_e	- tempo estimado para produção da quantidade Q
Q	- quantidade a produzir
Y_i	- taxa de produção inicial
Y_f	- ganho máximo na taxa de produção
T	- constante de tempo do modelo.

A equação é resolvida por um processo iterativo, utilizando para t_e, como estimativa inicial, o quociente entre Q e o valor máximo possível da taxa de produção $(Y_i + Y_f)$.

O cálculo do custo de produção baseia-se numa função não linear multivariável. O custo é constituído por uma componente fixa, e por uma componente que depende da quantidade a produzir e do tempo disponível para produção. Existe agravamento do custo sempre que se verifica necessidade de armazenamento local (custo de armazenamento), isto é, sempre que o tempo estimado para a produção é inferior ao tempo solicitado.

Em geral, utiliza-se a seguinte função para determinação do custo unitário global associado à produção de um dado produto:

$$Custo = \begin{cases} \left[k_2 \cdot e^{-k_3 Q} + k_1\right] \cdot e^{k_4 Q\left(\frac{t_e}{t_p}-1\right)} + \max\left\{t_p - t_e - k_6, 0\right\} \cdot k_5 & t_p \geq T_{\min} \\ \infty \quad t_p < T_{\min} \end{cases}$$

onde

t_e - tempo estimado para produção da quantidade Q

t_p - tempo solicitado para a produção

Q - quantidade a produzir

$T_{\min}$ - tempo mínimo possível para produção

$k_1 \cdots k_5$ - parâmetros do modelo.

Para além da componente fixa do custo, definida pelo parâmetro k_1, o efeito no custo, da quantidade a produzir apresenta um comportamento exponencial negativo, podendo ser parametrizado através dos parâmetros k_2 e k_3. A segunda componente exponencial modela o efeito de se desejar um agravamento do custo, no caso do tempo disponível para produção ser inferior ao tempo estimado para condições normais de produção. Na situação em que o tempo estimado é inferior ao tempo disponível, incorre-se em custos adicionais, se a folga verificada for superior ao valor definido pelo parâmetro k_6. Nesta situação, resultam custos de armazenamento com um peso definido pelo parâmetro k_5.

Refira-se ainda que neste modelo, foram implementadas duas funcionalidades adicionais, com a finalidade de representar situações mais realistas, nomeadamente considerando não linearidades associadas, quer aos tempos de processamento (por exemplo, através da definição de *lead-times* mínimos), quer às disponibilidades de produção no curto prazo (por exemplo, através da definição de níveis de *backlog*).

A Figura 2.14 apresenta um exemplo da resposta do modelo de capacidades para o custo unitário de produção em função do tempo disponível para produção (Duração) e do número de unidades a produzir (Quantidade), no caso da produção ocorrer sempre a partir do mesmo período de tempo. Na configuração apresentada, é nítida uma zona "plana" em que o custo é mínimo (situação óptima para afectar a produção). Verifica-se contudo o agravamento do custo, quer na situação do tempo disponível para produção diminuir, quer na situação em que este tempo aumenta (em que os custos de armazenamento se podem tornar significativos).

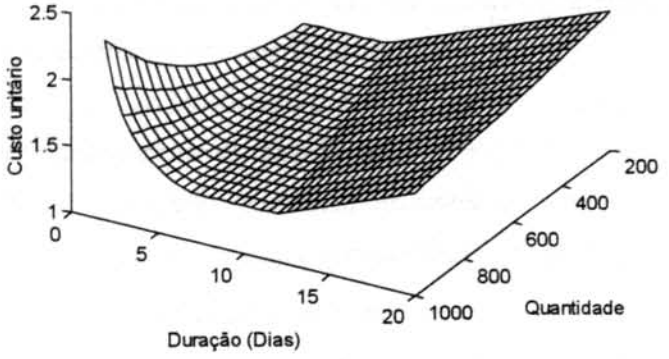


Figura 2.14 – Resposta do modelo em função do tempo de processamento e da quantidade a produzir

A Figura 2.15 apresenta a resposta do modelo, com os mesmos valores dos parâmetros, numa situação de se produzir uma quantidade de um determinado produto em diferentes alturas, e com diferentes tempos de produção disponíveis. Verifica-se, neste caso, o agravamento do custo quando a produção se inicia muito próxima da data actual.

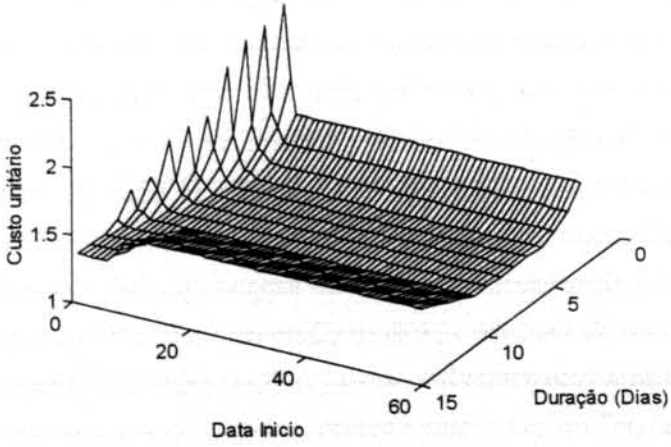


Figura 2.15 – Resposta do modelo num cenário de produção de uma quantidade fixa

2.5.6.2 Modelo baseado num perfil de capacidade

Neste modelo, define-se para o todo o horizonte temporal e com um nível de agregação elevado, o perfil de capacidade disponível.

O modelo implementa (em alternativa) o algoritmo de planeamento à capacidade finita nas duas estratégias possíveis: regressiva e progressiva. O cálculo do custo de produção é similar ao descrito no modelo anterior.

A Figura 2.16 apresenta um exemplo da resposta deste modelo numa situação em que o perfil de capacidade do sistema de produção, para o produto em planeamento, apresenta os valores representados na Figura 2.17.

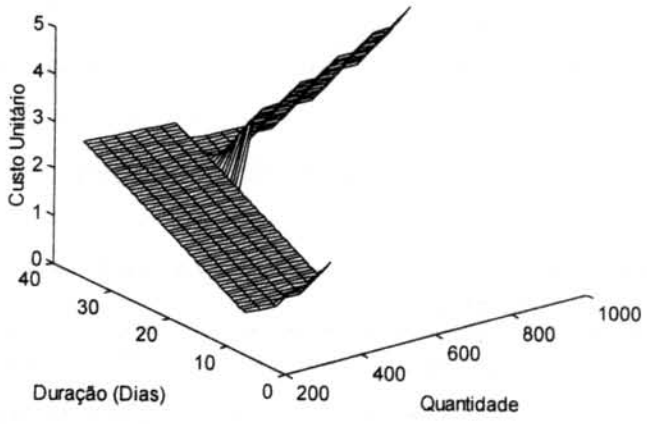


Figura 2.16 – Exemplo de resposta do modelo baseado num perfil de capacidades

Ao contrário do anterior, este modelo, permite definir com facilidade períodos de tempo em que se verifica impossibilidade de produção (por exemplo, situações de manutenção). Permite por isso modelar situações em certo sentido mais realistas, originando assim respostas "não lineares".

Para efeitos de configuração, o modelo necessita dos perfis de capacidade definidos para o sistema de produção, e de um conjunto de parâmetros apresentados na Tabela 2.9. Na Figura 2.17 é apresentado um exemplo de um perfil de capacidade para

um determinado produto e de acordo com um determinado estado do processo produtivo.

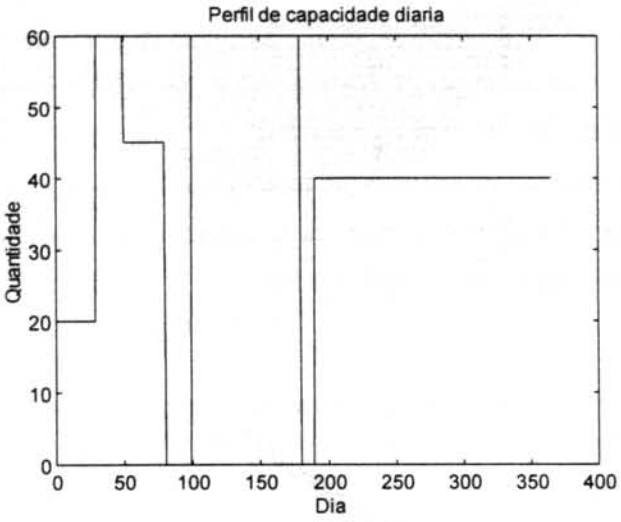


Figura 2.17 – Exemplo de perfil de capacidade

Parâmetro	Descrição
K_1	Define o limite da taxa de compressão para o tempo de processamento (por exemplo, k1=1.2 significa que o sistema suporta no limite uma redução de 20% no tempo normal de produção)
K_2	Custo fixo de produção (unitário)
K_3	Custo variável de produção (dependente do número de dias de produção)
K_4	Custo de armazenamento (por unidade e período)
K_5	Agravamento do custo produção quando necessário produzir em menos tempo (compressão)
K_6	Custo *standard* para produtos já planeados
LT_{min}	*Lead-time* mínimo para processamento
T	Horizonte de planeamento

Tabela 2.9 – Parâmetros de configuração do modelo

2.5.6.3 Modelo baseado num perfil de capacidade e em informação ATP

Este modelo é uma evolução do modelo anterior, baseando-se, por isso, no mesmo princípio de funcionamento, e incluindo informação adicional de ATP ao longo do horizonte de planeamento.

Basicamente, recorre-se ao mesmo algoritmo de planeamento (descrito na secção anterior), deduzindo-se para os valores das quantidades a planear, os valores possíveis de ATP no período em questão.

2.6 Conclusões

Não se procurou neste capítulo, apresentar uma revisão exaustiva das metodologias de planeamento da produção usualmente referidas na literatura. Optou-se, antes, por colocar a ênfase no processo de *order promising*, baseado na informação de ATP, pela sua importância que representa nos sistemas de produção MTO.

Constatou-se que o processo de *order promising* necessita de ser complementado por um mecanismo expedito de avaliação real de capacidades, por forma a assegurar a eficácia do processo de aceitação e negociação de encomendas, quando não existe quantidade suficiente disponível para satisfazer, na data pretendida, a quantidade solicitada pelo cliente. Além disso, o processo de negociação/aceitação tem de suportar o carácter distribuído subjacente aos sistemas de produção baseados em redes de empresas.

O desacoplamento, aqui considerado, entre as metodologias de planeamento e as metodologias de avaliação de capacidades é, em certo sentido, inovador. Esta visão conceptual permite, ao nível da implementação prática, desenvolver modelos de capacidades heterogéneos e interactuantes que suportem uma interface de certa forma mais natural e baseada em protocolos de negociação tipo proposta/oferta. A apresentação na parte final do capítulo, de alguns exemplos muito simples de modelos de capacidades é assim uma contribuição preliminar a esta problemática.

Note-se finalmente que, nos capítulos 4 e 5 desta dissertação, serão apresentadas metodologias de planeamento que recorrem intensivamente a esta abordagem desacoplada entre planeamento da produção e planeamento de capacidades.

Capítulo 3

Novos Paradigmas Organizacionais

3.1 A emergência de novos paradigmas

3.2 A empresa virtual

3.3 Sistemas de informação empresariais

3.4 Requisitos para os Sistemas de Informação

3.5 Conclusões

3 NOVOS PARADIGMAS ORGANIZACIONAIS

A história da organização e gestão empresariais, mostra de uma forma clara que as empresas têm sempre procedido a mudanças organizacionais por forma a responderem mais adequadamente às tendências e solicitações dos mercados, e a satisfazerem um conjunto cada vez mais exigente de requisitos.

Neste capítulo, temos como principal finalidade apresentar o conceito de *Empresa Virtual* como o novo e emergente paradigma organizacional das empresas, e a sua estreita relação com as tecnologias de informação e comunicação. Começa-se por fazer uma breve introdução ao aparecimento de novos paradigmas organizacionais, salientando, em particular, o conceito de "empresa ágil". Faz-se igualmente uma referência breve a projectos e programas de investigação relevantes nesta área. Após caracterizar o conceito de *Empresa Virtual* e de apresentar diferentes definições propostas na literatura, procura identificar-se um conjunto de requisitos base para as tecnologias de informação e de comunicação, para suporte a este tipo de organização.

A caracterização do mercado dos sistemas de informação empresarial será objecto de uma secção deste capítulo, onde se procura igualmente apresentar as principais tendências actuais. Conclui-se o capítulo, com a identificação de um conjunto de requisitos para os sistemas de informação de suporte à *Empresa Virtual*.

3.1 A emergência de novos paradigmas

3.1.1 Enquadramento

As questões organizacionais nunca foram objecto de tanta divulgação e especulação, como nos nossos dias. Se, no período posterior à II Guerra Mundial, a competitividade empresarial assentava em factores de produção fundamentalmente de natureza tangível, na última década do século, como resultado de uma continuada profusão de tecnologias por todos os sectores da actividade económica, os principais factores diferenciadores são

essencialmente intangíveis, passando cada vez mais por questões organizacionais e de coordenação entre as entidades envolvidas em complexos processos de cooperação.

As últimas três ou quatro décadas foram particularmente profícuas no aparecimento de novas estratégias de produção e organização. Na origem destas novas lógicas produtivas e organizacionais, estão as limitações de um quadro-tipo dominado pela doutrina da administração científica do trabalho (Frederick Taylor) e da estrutura organizacional (Henri Fayol e Max Weber), implementado em larga escala na primeira metade do século, com a preocupação única de se conseguir uma cada vez maior eficiência produtiva.

A segunda metade do século transmite-nos uma ideia clara de mudança, em que a nova lógica dominante é a competição e o desenvolvimento económico das nações industrializadas. De uma era, anos 50/60, em que o ênfase da competitividade se centrava nos custos de produção, passou-se para uma fase em que a saturação do mercado conduziu a uma inflexão para as questões de *marketing* e posteriormente para factores tais como a qualidade e a fiabilidade dos produtos. A década de 90 assistiu a uma nova mudança, na qual a vantagem competitiva é marcada pela ideia chave de *resposta rápida* às solicitações do mercado (*time to market*). Assim, emergiram novos conceitos, tais como: *Supply Chain Management*, *Lean Production*, *World Class Manufacturing* e *Agile Manufacturing* (Jagdev and Browne, 1998).

Para se manterem competitivas, as empresas tendem a evoluir progressivamente para formas de organização mais globais, com uma especial preocupação no que respeita à maior variabilidade da procura, à proliferação de novas tecnologias (que permitem reduzir substancialmente os tempos de desenvolvimento e produção de novos produtos, cada vez mais complexos), às novas exigências no tempo de resposta às solicitações dos clientes e ao aumento substancial da qualidade.

Este quadro geral conduziu naturalmente a novos paradigmas organizacionais, caracterizados por uma maior concentração das empresas nas suas competências e actividades principais - "core competencies" - e pelo estabelecimento de redes de cooperação com entidades externas[7], desde fornecedores a clientes, em que as activi-

---

[7] A necessidade de criar mais valor para o cliente, e a tentativa de partilhar, reduzir e exteriorizar as actividades marginais, isto é, aquelas que não implicam um papel fundamental na criação de valor para o cliente conduziram à implementação de políticas generalizadas de *outsourcing* (Rodrigues, 1998).

dades de coordenação e colaboração assumem naturalmente uma enorme importância, conduzindo consequentemente a desafios de grande complexidade.

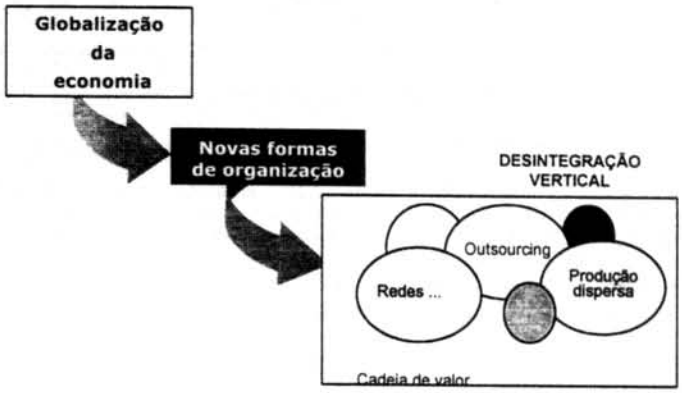


Figura 3.1 – A globalização da economia e a consequente desintegração vertical

Esta tendência de desintegração vertical ou de "desagregação" de competências, visando uma maior adaptação a um mercado cada vez maior e simultaneamente mais disperso, tem sido objecto de alargada investigação nas últimas duas décadas. A Figura 3.1 ilustra a tendência de desagregação vertical, como consequência da globalização da economia e da emergência de novas formas de organização. Como advogado pela teoria económica neo-liberal, a cooperação preferencialmente à competição, quer entre empresas, quer a um nível mais lato entre nações, torna-se mais vantajosa e mais promissora em qualquer cenário económico (Pontrandolfo and Okogbaa, 1999).

A emergência, neste fim de século, de novos paradigmas organizacionais dominados por características tais como a adaptabilidade, a flexibilidade, a criatividade, a agilidade, mostram bem o quanto a evolução das tecnologias da informação está relacionada com a globalização dos mercados e com a sua dinâmica de mudança.

A flexibilidade organizacional das estruturas empresariais, aliada a uma contínua evolução do seu nível tecnológico, serão cada vez mais no futuro, a fonte de vantagem competitiva por detrás das suas "core competencies". A empresa tenderá no futuro a ser heterogénea nas partes que a constituem, muito mais do que uniforme, rígida e hierárquica como até recentemente. Em (Warnecke, 1993) é identificada como carac-

terística comum a todas as revoluções industriais, a tendência de progressão de soluções centralizadas para soluções descentralizadas.

Dinâmica e complexidade são dois desafios que reflectem as características centrais dos futuros mercados (Eversheim et al., 1997). Esta constatação tem conduzido as empresas a procurarem e incrementarem a criação de valor mediante o estabelecimento de redes de cooperação não permanentes, isto é, temporárias, envolvendo entidades independentes e de forma a garantirem vantagens mútuas. A imparável dinâmica de competição e a complexidade dos sistemas de produção estão naturalmente associadas à emergência da criação de valor cooperativo (Figura 3.2).

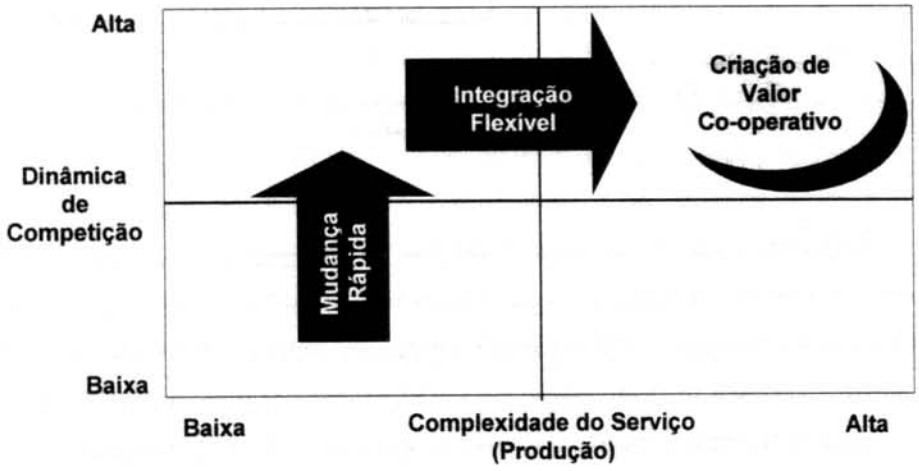


Figura 3.2 – A emergência da *criação de valor cooperativo* (adaptado de (Eversheim et al., 1997))

Por outro lado, o aumento da complexidade dos serviços (independentemente do tipo) e o recurso a redes de cooperação estabelecidas entre várias entidades, implica não só o desenvolvimento de sistemas organizacionais flexíveis e reconfiguráveis, como, fundamentalmente, um incremento significativo do esforço de coordenação.

Assim, novas ideias e conceitos têm sido nos últimos anos propostos no âmbito da organização dos sistemas de manufactura, com a finalidade de suportarem de forma efectiva os requisitos e tendências emergentes, nomeadamente a globalização da competitividade, a constante diminuição do ciclo de vida dos produtos, a contínua melhoria da qualidade e da fiabilidade, a concepção e a produção de produtos cada vez

mais complexos, etc. Exemplos de novas ideias e conceitos incluem: a empresa fractal, os sistemas de fabrico holónicos, os sistemas de fabrico biológicos e os sistemas de fabrico ágeis (Gindy, 1999). Para uma apresentação e comparação destes conceitos, sugere-se a leitura de Tharumarajah (Tharumarajah et al., 1996). Alguns autores consideram que a maior parte destes conceitos serão a base de novos paradigmas organizacionais permitindo suportar, de forma adequada, a cada vez maior imprevisibilidade nas condições de mercado.

Refira-se finalmente que diversos programas e projectos de investigação têm sido lançados nos últimos anos, com o objectivo de formalizar e de desenvolver metodologias de modelação adequadas aos novos paradigmas organizacionais, nomeadamente às organizações ágeis, virtuais ou a operarem em estruturas cooperativas complexas. A secção seguinte apresenta de forma sucinta alguns dos projectos mais significativos lançados nos últimos anos, nesta área.

3.1.2 Programas e projectos relevantes

Em 1990, o Japão tomou a iniciativa de lançar um programa de investigação de âmbito internacional, envolvendo os Estados-Unidos, a União Europeia, o Canadá e a Austrália. A iniciativa, denominada IMS – *Intelligent Manufacturing Systems,* encontrou-se em discussão durante três anos, seguindo-se uma fase piloto de dois anos em que foram desenvolvidos seis projectos experimentais de investigação, de âmbito internacional. Oficialmente, a iniciativa IMS começou em 1995, tem uma duração de 10 anos e envolve neste momento cerca de 250 empresas e cerca de 200 instituições de investigação. O objectivo principal é a promoção de projectos e programas de investigação que desenvolvam a futura geração das tecnologias de produção. Exemplos de programas desenvolvidos neste âmbito incluem o NGMS IMS – The Next Generation Manufacturing Systems e o Globeman 21 (*The Enterprise Integration for Global Manufacturing Towards 21st Century*).

O programa internacional NGMS IMS – *The Next Generation Manufacturing Systems* envolve os Estados-Unidos, o Japão, a Austrália e a União Europeia, e tem como objectivo principal desenvolver a denominada "*Next Generation Enterprise*" (NGE) através de linhas de investigação, com vista a desenvolver:

- uma visão unificada dos sistemas de manufactura da próxima geração;

- um quadro de referência para integração de sistemas de produção globalmente distribuídos e constituídos por unidades autónomas;
- bases de conhecimento (*on-line*) para re-utilização de informação;
- métodos de modelação e metodologias de simulação para empresas compostas por unidades autónomas e distribuídas.

O programa NGMS IMS procura integrar quatro perspectivas diferentes, relacionadas com sistemas avançados de produção, e com origem em três dos principais membros do consórcio. Assim, os Estados Unidos contribuem com a perspectiva de *Agile Manufacturing*, o Japão, com os conceitos subjacentes a dois programas: um de nível de *shop-floor* denominado *Autonomous & Distributed Manufacturing Systems* (ADMS) e outro de nível superior (táctico) denominado *Biological Manufacturing System* (BMS)[8]; a Alemanha contribui com a visão desenvolvida em *The Fractal Company* (Warnecke, 1993) em que a empresa é vista como um conjunto de identidades similares, com orientação, organização e optimização próprias, cooperativas e dinâmicas.

Outro projecto da iniciativa IMS é o Globeman 21 (*The Enterprise Integration for Global Manufacturing Towards 21st Century*). Basicamente, este projecto investigou a problemática envolvida no acesso e no controlo da informação em redes de empresas.

Uma iniciativa internacional de âmbito europeu foi promovida com o projecto EUREKA "Factory for the Future". Este projecto envolveu três países (Alemanha, França e Reino-Unido) e procurou desenvolver diversos modelos de referência, numa perspectiva vertical, para a denominada "fábrica de futuro" suportando os conceitos de *agilidade* e de *Empresa Virtual*.

Iniciativas de âmbito nacional foram igualmente lançadas por diversos países, nomeadamente o programa *Agile Manufacturing* nos Estados-Unidos, o *Innovative Manufacturing Initiative* pelo Reino Unido e o *Production 2000* na Alemanha.

---

[8] O programa BMS considera que algumas das características dos sistemas biológicos podem ser imitadas pelos sistemas de produção. A contribuição deste último programa para a visão global do programa NGMS IMS é essencialmente explorar os resultados mais recentes das áreas da biologia, bioquímica molecular, matemática aplicada e vida artificial, nomeadamente ao nível da integração de elementos funcionais e hierarquias, da disposição da informação genética, do uso de inteligência natural e das estruturas dinâmicas.

Subjacente a todos os programas e projectos de investigação referidos, encontra-se sempre uma organização empresarial, geralmente temporária, heterogénea e cooperativa. É por isso frequente o recurso a designações como *Empresa Virtual* para caracterizar o tipo de estruturas organizativas envolvidas, apesar de não ser ainda consensual o âmbito preciso desta expressão. Existem variados projectos que abordam diferentes aspectos associados às "empresas virtuais", nomeadamente desde aspectos conceptuais de criação, operação e extinção, passando pelo projecto de infra-estruturas tecnológicas de suporte às suas funcionalidades e pelo desenvolvimento de metodologias de planeamento colaborativo. Exemplos representativos são os projectos seguintes:

- NIIIP - *National Industrial Information Infrastructure Protocols* (EUA). Este projecto (levado a cabo por um consórcio de 18 instituições) tem como objectivo principal adoptar, desenvolver, demonstrar e disseminar as tecnologias de suporte a todo o ciclo de vida das empresas virtuais (NIIIP, 1995).

- X-CITTIC - *A Planning and Control System for Semiconductor Virtual Enterprise* (UE). Este projecto, com ênfase particular na indústria dos semicondutores, teve como objectivo principal o desenvolvimento de um sistema de informação que suportasse o planeamento e o controlo da produção numa rede de empresas[9].

- PRODNET II – *Production Planning and Management in an Extended Enterprise* (UE), projecto financiado pela Comissão Europeia, envolveu parceiros europeus e da América Latina; teve como objectivo principal o desenvolvimento de uma arquitectura de referência e infra-estrutura de suporte para empresas virtuais, com ênfase particular nas pequenas e médias empresas (Camarinha-Matos et al., 1997b) (Camarinha-Matos et al., 1997a).

Outros projectos na área incluem:

- MUSSELS – Multi Site & Multi Stage Enterprises Logistic Control System;

---

[9] Financiado pela Comissão Europeia (ESPRIT), decorreu no período de 1996 a 1998, e envolveu um consórcio formado por três instituições de I&D, três empresas industriais de produção de semicondutores e uma "software-house". Dado o envolvimento que o autor teve neste projecto, será feita, no capítulo 4, uma apresentação pormenorizada de alguns dos seus resultados.

- LOGSME – Logistics to Support Continuous Improvement for SMES in The Virtual Enterprise;
- VEGA – Virtual Enterprise Using Groupware Tools and Distributed Architecture;
- PLENT – Planning Small Medium Enterprises Networks.

3.1.3 Empresa ágil

Numa economia global, apoiada por tecnologias de informação e comunicação avançadas, competir à escala mundial significa satisfazer clientes cada vez mais exigentes e em mercados cada vez mais fragmentados. São estes mercados que exigem respostas inovadoras por parte dos sistemas de produção. A "massificação da personalização dos produtos" (*mass customisation*), a redução dos prazos de entrega, a redução do ciclo de vida, a entrega pontual, a qualidade e o preço tornam-se assim factores essenciais de sobrevivência e de sucesso das empresas.

Ao longo deste século, a base dos processos de fabrico progrediu desde um ambiente artesanal, através da produção em massa, até à denominada produção "magra" (*lean*) e flexível, e presentemente continua a sua evolução a um ritmo crescente dando origem ao que é já um novo paradigma organizacional: a "empresa ágil".

A expressão 'Agile Manufacturing' teve a sua origem nos Estados Unidos e resultou de um estudo efectuado por uma equipa de investigação envolvendo empresas industriais, agências governamentais e organismos públicos, acerca da competitividade das empresas americanas num mercado global (Nagel and Dove, 1991). Uma das conclusões deste estudo foi que a melhoria do desempenho dos sistemas de produção em série não traria competitividade para as empresas a operarem num mercado global. Em vez disso, seria necessário adoptar e estimular novas formas de trabalho nomeadamente através de "empresas ágeis". Segundo o mesmo relatório, a "agilidade" requer a integração de vários factores: tecnologias flexíveis de produção; uma força de trabalho competente; conhecimento; e estruturas organizacionais de gestão que promovam e estimulem iniciativas cooperativas entre as empresas (Jagdev and Browne, 1998). Um aspecto igualmente importante do *Agile manufacturing* é a capacidade para desenvolver relações estratégicas entre clientes e fornecedores.

Ser "ágil" significa literalmente mover-se com facilidade. Aplicado a um organização produtiva, ser ágil é responder rapidamente aos requisitos de um mercado em constante mudança.

A empresa, para ser ágil num contexto global, tem de ter a capacidade de rapidamente introduzir no mercado produtos inovadores; tem igualmente de ser capaz de dar uma resposta rápida ao mercado e aos clientes, quanto a novos produtos; e tem que operar com um sistema produtivo re-configurável e continuamente em mudança para se adaptar à própria dinâmica do mercado.

A *empresa ágil*, como novo paradigma organizacional, baseia-se em três características fundamentais: a dinâmica de mudança, a resposta rápida e a noção de qualidade total (Warnecke, 1993).

Uma empresa ágil encontra-se necessariamente em constante mudança, nunca atingindo um estado estável, e adaptando-se constantemente a um conjunto de circunstâncias imprevisíveis. A capacidade para responder rapidamente apenas pode ser conseguida através de processos de cooperação internos e externos à própria empresa. A cooperação pressupõe um elevado grau de confiança entre parceiros, e uma partilha igualmente segura do conhecimento e da informação. Por último, a qualidade não pode ser unicamente uma característica do "produto" em si, mas deve ser antes estendida a todo o relacionamento existente entre o cliente e o fornecedor.

Note-se que diferentes perspectivas e visões são propostas na literatura, acerca do conceito de "agilidade". Contudo, todas elas, de uma ou outra maneira, têm na base um outro conceito ou paradigma proposto no início da década de 90 por Womack, Jones e Roos com o livro "The Machine that Changed the World" (Womack et al., 1990). O conceito de "produção magra" (*lean production*) desenvolvido a partir de então, foi inspirado no modelo de produção desenvolvido pela Toyota. A evolução/extensão do conceito de "produção magra" para "empresa magra" defende a eliminação de desperdícios a todo o nível e em toda a cadeia de valor (Womack and Jones, 1994).

Naylor et al. (1999) separam, de uma forma clara, os conceitos de "ágil" e de *lean*, para os quais propõem as seguintes formulações:

"Agilidade pressupõe usar o conhecimento do mercado e recorrer a organizações virtuais para explorar as oportunidades de negócio em mercados voláteis."

"Leanness pressupõe o desenvolvimento de um sistema de valor fluido eliminando o desperdício, incluindo o tempo, e assegurando um nível estável de execução."

Basicamente, a agilidade enfatiza a ideia de adequação a mercados voláteis e instáveis, isto é, a mercados em constante mudança. Por seu lado, o conceito de *lean* enfatiza a ideia de estabilidade e previsibilidade. Apesar de apresentarem conceitos distintos, os autores defendem a sua integração/fusão na aplicação a cadeias logísticas complexas.

Christopher et al. (1999), com base num programa de investigação sobre *agilidade* e o seu nível de implementação em empresas europeias, caracterizaram a agilidade por confrontação directa com as ideias subjacentes a ambientes *lean*. A Tabela 3.1 apresenta e em termos comparativos resumidamente as principais características identificadas.

Agile	*Lean*
Orientação ao cliente ênfase nos clientes e no mercado	**Orientação ao produto e ao processo** ênfase na eliminação do desperdício nos produtos e nos processos
Colaboração dinâmica ênfase em organizações em rede e com reconfigurações permanentes	**Parcerias de longo prazo** ênfase em relações duradoiras e estáveis
"Capabilidade" ênfase na aptidão e na diferenciação	**Medidas de desempenho universais** ênfase em medidas de desempenho rígidas (produtividade, qualidade, etc)
Gestão autonomizada maximização da autonomia e da capacidade de resposta	**Gestão hierarquizada** normalização do trabalho para assegurar conformidade com níveis *standard* de qualidade e de produtividade
Resposta oportuna oportunidade de mercado: captura instantânea da procura, interpretação e resposta imediata	**Métodos multi-estágio e multi-função** ênfase na estabilidade da produção

Tabela 3.1 – "Agile" versus "Lean"

Numa perspectiva de cadeia de fornecimento, pode caracterizar-se uma estrutura *lean* como sendo um processo "fisicamente eficiente", isto é, optimizado para o fornecimento de "produtos funcionais" (normalizados) em que a procura é de certa

forma estável. Por outro lado, uma estrutura *agile* é capaz de responder adequadamente às solicitações do mercado para produtos inovadores, para os quais a procura é instável e imprevisível.

Assim, o conceito de *lean* está, em certo sentido, associado à eficiência de toda a estrutura produtiva e o conceito de *agile* à "responsividade" da mesma. Esta associação foi originalmente apresentada por Fisher (Fisher, 1997) sob a forma de uma matriz em que é ilustrada a estreita relação entre estruturas produtivas eficientes e produtos "funcionais" e estruturas produtivas "responsivas" e produtos inovadores (Figura 3.3).

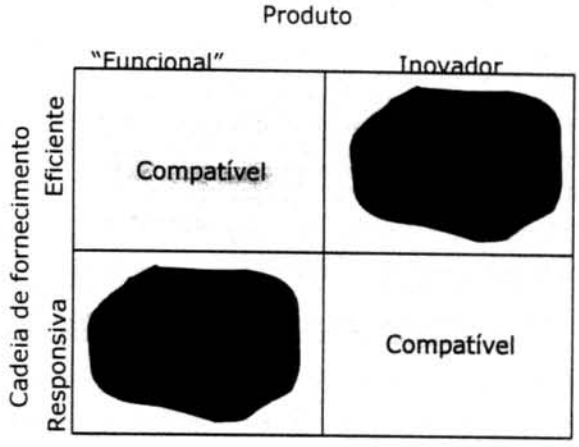


Figura 3.3 – Matriz de 'Fisher' (adaptado de (Christopher et al., 1999))

O autor defende que os conceitos de *lean* e *agile*, não são antagónicos mas antes e fundamentalmente complementares, podendo portanto coexistir, quer ao nível de uma entidade própria (autónoma), quer ao nível de uma organização complexa constituída por diversas entidades eventualmente independentes. A Figura 3.4 procura ilustrar a coexistência destes dois conceitos através de um ponto de interacção que coincide com a ideia de ponto de desacoplamento, no processo de produção e de planeamento anteriormente já referido.

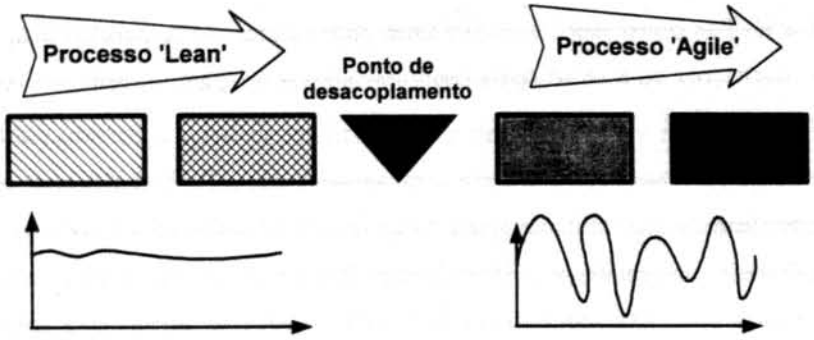


Figura 3.4 – Coexistência dos conceitos 'agile' e 'lean'

Uma abordagem à empresa ágil, numa perspectiva essencialmente intra-empresa e direccionada para a utilização e adaptação dos sistemas de informação a um ambiente "ágil", pode ser encontrada em (Maskell, 1994).

Refira-se, no entanto que o âmbito aqui considerado, para os conceitos associados à agilidade empresarial, pressupõe não só uma perspectiva intra-empresa, como fundamentalmente uma perspectiva inter-empresas, isto é, estendida a toda uma organização "virtual".

O pressuposto é de que a transição para a "agilidade" contempla a remodelação total da organização, quer do ponto de vista de produção, quer dos processos de gestão.

Responder rapidamente ao mercado com novos e mais complexos produtos é, naturalmente, uma vantagem competitiva das empresas. De acordo com este paradigma organizacional, a introdução de novos produtos assenta na formação de estruturas organizacionais envolvendo recursos de diferentes entidades numa única entidade de negócio e enquanto prevalecer a oportunidade de mercado. Inerente a esta forma de organização, encontra-se assim a noção de *Empresa Virtual*, conceito a ser caracterizado em secções seguintes deste capítulo. De facto, a noção de *Empresa Virtual* parece estar subjacente ao paradigma do *Agile Manufacturing*, constituindo o factor diferenciador fundamental relativamente ao paradigma do *Lean Production*.

3.2 A *Empresa Virtual*

Pela variedade de termos e conceitos apresentados em inúmeros trabalhos de investigação e em diversas publicações, torna-se evidente que não existe um consenso acerca desta temática. Diferentes termos, tais como *Virtual Enterprise*, *Extended Enterprise*, *Virtual Factory*, *Virtual Company*, *Virtual Organization*, etc, têm sido propostos para expressar, por vezes, as mesmas ideias no mesmo contexto, ou ideias diferentes em contextos igualmente diferentes.

O conceito do termo "virtual", no contexto que temos vindo a referir, pode ser explicado em parte recorrendo a uma analogia com a unidade de memória de um sistema computacional, como sugere Zimmermann (Zimmermann, 1988). Não sendo economicamente e tecnicamente viável ter memória principal para todas as exigências das aplicações informáticas, a solução consiste em recorrer a memória virtual através do "mapeamento" em memória secundária, que constitui, geralmente, um recurso ilimitado. Através de mecanismos de alocação dinâmica entre memória principal e secundária, é possível satisfazer plena e simultaneamente os pedidos das aplicações. Nesta analogia, identificam-se diversas similitudes, nomeadamente, a satisfação temporária de um ou mais pedidos, a disponibilidade quase ilimitada de recursos, a dinâmica de reconfiguração, etc.

A inexistência de uma plataforma comum, ao nível de termos, conceitos e relações, na temática das empresas virtuais, levou o autor a participar no desenvolvimento de uma *ontologia*[10] orientada a este domínio (Soares et al., 1996). Um exemplo da aplicação da ontologia desenvolvida, assim como a fundamentação e necessidade do seu desenvolvimento encontram-se referidos em (Soares et al., 1997).

O termo "virtual", num contexto de manufactura, tem as suas raízes no trabalho de Williamson, "Markets and Hierarhies", em 1975 acerca da economia nos custos de transacção (Sharp and Beach, 1998). Nesse trabalho, o autor advogava que as empresas industriais deveriam "exteriorizar" sempre que possível as suas actividades, quer produtivas, quer de serviços. Um impacto imediato destas ideias foi a crescente

---

[10] Entendemos uma *ontologia* como a conceptualização de um domínio tornada explícita através da definição rigorosa das entidades (conceitos, relações e axiomas) identificadas nesse domínio. A principal finalidade de uma ontologia é constituir um entendimento do domínio que, após formalização, possa ser utilizado em várias aplicações (Soares, 1998).

popularidade verificada na década de 80 com a implementação de estratégias de *outsourcing*.

Numa perspectiva orientada aos negócios e com abrangência mais lata, o conceito de *Virtual Corporation* foi apresentado por Davidow e Malone, em 1992 (Davidow and Malone, 1992).

Num sentido lato e generalista, a emergência da "Organização Virtual" resulta por um lado da globalização dos mercados e consequentemente da economia, e por outro da tendência de desintegração vertical ocorrida nas organizações como condição fundamental à sua cada vez maior concentração nas actividades principais (*core business*). Assim, em primeira instância, uma "Organização Virtual" pode ser definida como um agrupamento (com carácter temporário) de *agentes económicos*, internos ou externos à própria organização, com competências e actividades complementares, que interagem e cooperam espacial e temporalmente de forma coordenada, por forma a alcançarem um objectivo comum bem definido.

A evolução das organizações virtuais ou, num sentido mais restrito, a evolução das redes de empresas, tem seguido duas trajectórias principais. Por um lado verifica-se a intensificação da cooperação entre empresas de pequena e média dimensão e, noutra linha, verifica-se uma crescente reestruturação das grandes empresas, numa cada vez maior desintegração vertical, fomentando por isso uma cooperação, num certo sentido, verticalizada.

Como é natural, a obtenção de vantagens competitivas por parte de empresas pequenas e de média dimensão, quanto à possibilidade de entrarem em novos mercados e fornecerem novos produtos, é grandemente dificultada pela limitada disponibilidade de recursos. Uma forma de ultrapassar estas dificuldades passa então pela cooperação entre empresas, nomeadamente através do estabelecimento de contractos de cooperação e formação de alianças estratégicas. Estas soluções, clássicas, suportadas por disposições legais de certa forma rígidas, apresentam como vantagem fundamental o facto de limitarem os custos de transacção decorrentes da cooperação estabelecida. Contudo, uma desvantagem significativa resulta da consequente rigidez de um contrato impossibilitar uma resposta flexível, rápida e reactiva a alterações ou novas solicitações do mercado. As empresas virtuais ultrapassam estes obstáculos pelo facto de apresentarem uma organização fortemente baseada na cooperação entre as entidades

que a constituem e por, implicitamente, recorrerem a uma infra-estrutura tecnológica de informação e comunicação que limita significativamente os custos de transacção.

3.2.1 *Empresa Virtual* ou Estendida ("extended") ?

Procurando restringir o âmbito da "Organização Virtual" ao contexto que se tem vindo a apresentar, começa-se por referir as definições originais propostas no dicionário da APICS (American Production and Inventory Control Society) para *Virtual Corporation* e *Virtual Factory* (APICS, 1995):

Virtual Corporation

"The logical extension of outpartnering. With the virtual corporation, the capabilities and systems of the firm are merged with those suppliers, resulting in a new type of corporation where the boundaries between the suppliers' systems and those of the firm seem to disappear. The virtual corporation is dynamic in that the relationships and structures formed change according to the new needs of the customer."

Virtual Factory

"A changed transformation process most frequently found under the virtual corporation. It is a transformation process that involves merging the capabilities and capacities of the firm with those of its suppliers. Typically, the components provided by the supplier are those that are not related to a core competency of the firm, while the components managed by the firm are related to core competencies. One ability found in the virtual factory is that it can be restructured quickly in response to changing customer demands and needs."

Apesar da variedade de termos já referida, há hoje uma tendência para se concentrar fundamentalmente em dois deles:

- Empresa Virtual (Virtual Enterprise) - EV;
- Empresa "Estendida" (*Extended Enterprise*) - EE.

Se começa a haver concordância quanto ao emprego destes termos como designação dos novos paradigmas organizacionais, ainda não existe unanimidade quanto

ao âmbito preciso de cada um deles. Alguns autores defendem que o conceito de *Empresa Virtual* é mais abrangente e inclui o conceito de Empresa Estendida, outros defendem exactamente o oposto (Rolstadas, 1998).

A unanimidade é no entanto patente naquilo que parece ser a característica fundamental de uma *Empresa Virtual* ou Estendida, nomeadamente, quanto ao facto de ser um consórcio temporário, formado por entidades autónomas e dispersas geograficamente, e cooperando de forma coordenada através do recurso a tecnologias de informação e de comunicação. Esta visão tem sido defendida em diversas publicações, nomeadamente em (NIIIP, 1995), (Camarinha-Matos et al., 1997a) e (Azevedo and Sousa, 1997).

Segundo a literatura, a tendência actual é considerar a *Empresa Virtual* como uma rede de nós, nomeadamente, fornecedores, clientes, unidades de produção e de montagem, unidades logísticas e outros serviços especializados. Em (Azevedo and Sousa, 1999), considera-se que uma *Empresa Virtual* é baseada numa matriz alargada de sistemas e processos de negócios, composta por fornecedores, clientes, desenvolvedores, produtores, produtos, serviços e informação global. Subjacente a essa matriz está uma integração, segundo diferentes eixos, nomeadamente o geográfico ou físico, o funcional ou de processo e o da informação.

Por seu lado, Jagdev e Browne advogam que a diferença entre Empresa Estendida e *Empresa Virtual* é em certo sentido uma questão de semântica (Jagdev and Browne, 1998). Estes autores propõem as seguintes definições para Empresa Estendida e *Empresa Virtual*:

Extended Enterprise

"The formation of closer co-ordination in the design, development, costing and the co-ordination of the respective manufacturing schedules of co-operating independent manufacturing enterprises and related suppliers."

Virtual Enterprise

"Linking the operations of co-operating enterprises extremely closely so as to respond extremely quickly, and jointly, to the changing market needs."

Pelas definições apresentadas, é evidente que nestes tipos de empresas estão envolvidas unidades externas, dispersas geograficamente, e não detidas legalmente pela

Empresa Virtual/Estendida. Igualmente se conclui que a *Empresa Virtual* pode ser vista como uma aglomeração cooperante de *empresas ágeis*.

Por outro lado, parece também poder deduzir-se que o grau de integração e o âmbito de cooperação é mais amplo no caso da *Empresa Virtual*, pelo que as Empresas "Estendidas" podem ser consideradas como um caso especial da *Empresa Virtual*.

Camarinha-Matos et al. (1997a) consideram que o contexto de uma Empresa "Estendida" se adequa melhor a organizações nas quais uma empresa "principal", isto é, dominadora, "estende" os seus limites a todos ou a alguns dos seus fornecedores, enquanto a *Empresa Virtual* pode ser vista como um conceito mais geral, envolvendo outros tipos de organizações, nomeadamente uma estrutura mais democrática, na qual a cooperação é estabelecida de igual para igual.

A organização estrutural da *Empresa Virtual* encontra-se ilustrada na Figura 3.5. Neste exemplo, a *Empresa Virtual* é constituída por cinco entidades independentes. Algumas delas são apenas partes de outras empresas (por exemplo, a entidade P1 da empresa B). A figura ilustra igualmente a possibilidade de uma entidade poder participar simultaneamente em mais do que uma *Empresa Virtual*.

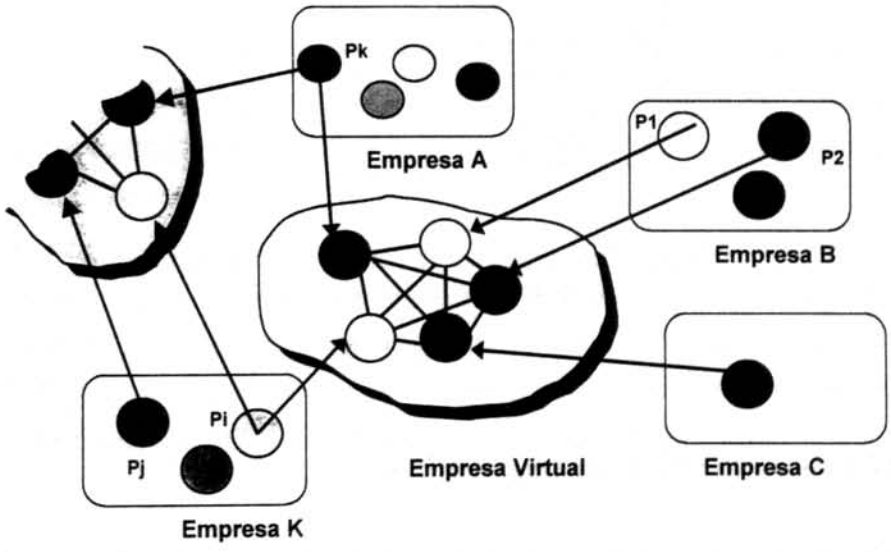


Figura 3.5 – Cooperação entre entidades autónomas numa *Empresa Virtual*

Note-se que, comparativamente a outras formas de organização tais como alianças estratégicas, *joint ventures* ou grupos de empresas, a *Empresa Virtual* diferencia-se quanto ao facto de combinar várias entidades, mantendo a flexibilidade e a autonomia de cada uma, mas exteriorizando-se como uma só entidade. Por outro lado, uma outra diferença significativa, resulta do facto de a duração da *Empresa Virtual* poder assumir valores muito diversos e, consequentemente, se estabelecerem ligações de tipo muito diverso entre as entidades que a compõem. A Figura 3.6 ilustra, no caso geral, o posicionamento da *Empresa Virtual* relativamente a outras formas de cooperação tradicionais, num plano em que um dos eixos representa a intensidade da cooperação entre entidades e o outro, a duração dessa cooperação.

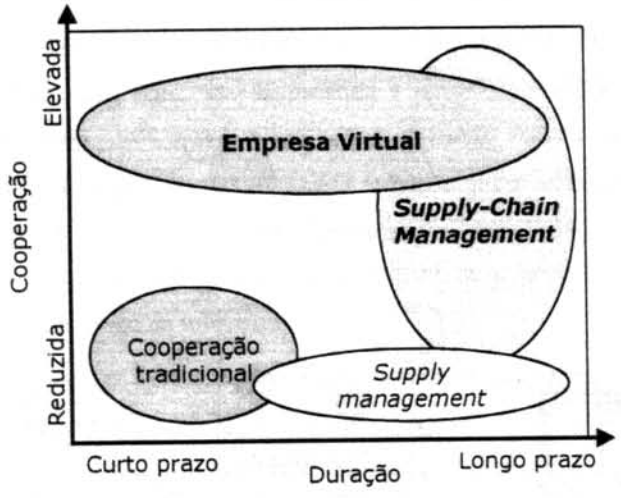


Figura 3.6 – Intensidade e duração da cooperação (adaptado de (Westkamper, 1998))

3.2.2 Características da *Empresa Virtual*

Comparativamente a empresas ou organizações tradicionais, podem identificar-se nas Empresas Virtuais as seguintes características principais:

- *Autonomia* – as entidades são, em geral, independentes, podendo nomeadamente integrar outras empresas virtuais, ou encontrar-se integradas em estruturas de cooperação com contratos de longo prazo.

- *Distribuição* – as entidades que compõem a *Empresa Virtual* encontram-se fisicamente dispersas, eventualmente muito afastadas (por exemplo, em diferentes continentes).

- *Configuração dinâmica* – constituindo em geral uma aliança temporária, a *Empresa Virtual* tem subjacente um ciclo de vida que, para além do início e fim da própria *Empresa Virtual*, compreende a entrada e saída de novas entidades, de acordo com as oportunidades do mercado em que opera, originando-se assim actividades de reconfiguração.

- *Adaptabilidade* – no seu conjunto, a *Empresa Virtual* reage rapidamente, isto é, adapta-se com agilidade às mudanças do mercado.

- *Redes de informação* – a emergência da *Empresa Virtual* como paradigma organizacional foi possível, em grande parte, ao desenvolvimento sem precedentes das tecnologias de informação e de comunicação.

3.2.3 Ciclo de Vida

A génese de uma empresa/organização virtual está na capacidade para criar e estabelecer cooperações temporárias e aproveitar oportunidades de negócio efémeras, tais que, uma empresa actuando de forma isolada, por incapacidade tecnológica ou limitação de recursos, não conseguiria aproveitar.

Resulta deste comportamento dinâmico, provocado pela necessidade de constante adaptação ao mercado, um ciclo de vida estruturado em cinco fases principais e ilustrado na Figura 3.7: criação e identificação de entidades, configuração e organização da empresa, operação, reconfiguração e dissolução (ver, por exemplo, (Katzy, 1998), (Zimmermann, 1988) e (Camarinha-Matos and Afsarmanesh, 1998)).

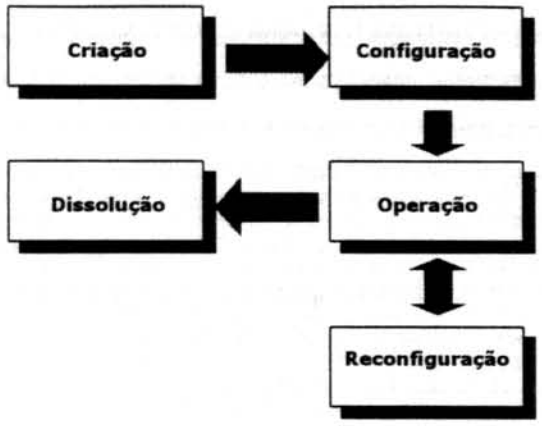


Figura 3.7 – Ciclo de vida de uma *Empresa Virtual*

Criação e identificação de entidades externas

Esta fase compreende o processo de selecção, por parte da empresa principal ou coordenadora (*core company*), de empresas com competências adequadas a uma *Empresa Virtual* para aproveitar a oportunidade de negócio identificada.

Configuração e organização

Nesta fase, são estabelecidas as regras e as condições de cooperação, nomeadamente, atribuição de tarefas, afectação de recursos, definição de procedimentos operatórios, estabelecimento de infra-estruturas tecnológicas, etc.

Operação

Esta fase, envolve a execução e a coordenação das tarefas planeadas por todas as entidades participantes. É uma fase em que a negociação no planeamento de actividades assume particular importância e onde a monitorização das actividades se revela decisiva para garantir reacções adequadas a alterações ou situações imprevisíveis.

Reconfiguração

A *Empresa Virtual* é por natureza dinâmica, isto é, deve poder facilmente suportar alterações de estrutura. Esta fase compreende a entrada e a saída de entidades, com a finalidade de alterar a estrutura da própria *Empresa Virtual*, ou simplesmente para substituir algumas das entidades participantes.

Dissolução

Uma *Empresa Virtual* é criada para aproveitar uma oportunidade do mercado, tendo por isso um tempo de vida limitado. Uma vez atingidos os objectivos para os quais foi criada, procede-se, em geral à sua dissolução.

3.2.4 A *Empresa Virtual* e as tecnologias de informação

A emergência da *Empresa Virtual* como novo paradigma organizacional pressupõe o recurso a um conjunto de instrumentos de apoio à gestão, suportados por tecnologias de informação e comunicação adequadas, dotando a empresa de meios adequados à negociação, cooperação e coordenação inter-entidades.

As fases de configuração, reconfiguração e operação parecem ser as mais exigentes e para as quais há actualmente, menos soluções disponíveis. Em particular, a configuração e a reconfiguração necessitam de instrumentos para "mapear", de forma eficiente, os sistemas e processos físicos produtivos em agentes de informação, por forma a automatizarem os processos de negociação a decorrerem durante a fase de operação. Nesta fase, em que predomina a coordenação, é possível identificar duas funcionalidades principais: uma *pro-activa* e outra *reactiva*. À primeira são associados os processos de negociação, aceitação e planeamento de encomendas; a segunda trata essencialmente do controlo e monitorização do fluxo das encomendas ao longo da empresa e das acções correctivas a tomar em situações imprevistas.

Uma consequência imediata do estabelecimento de estruturas produtivas em rede integrando cadeias logísticas complexas, ao nível da produção, é a alteração nos processos de planeamento, de um nível estritamente local, para um nível mais abrangente, onde os aspectos logísticos e de coordenação assumem grande importância. Assim, na perspectiva da produção, deve ser possível gerir dinamicamente as capacidades dos diferentes intervenientes, apoiando, por um lado, todo o processo de aceitação e colocação de encomendas e por outro, o planeamento das unidades produtivas. Neste sentido, as tecnologias de informação devem disponibilizar aos decisores, um conjunto de soluções que permitam:

- a negociação das encomendas (*order promising*) em tempo real, através de uma rápida avaliação da capacidade produtiva (ao longo da rede de empresas);

- o planeamento optimizado e sincronizado das actividades das empresas;
- a monitorização do fluxo das ordens de produção e o apoio à tomada de decisões, em caso de ocorrências que perturbem esse fluxo, por forma a garantir o cumprimento dos prazos de entrega;
- um sistema de informação configurável e escalável, segundo a estrutura da rede empresarial.

Os mecanismos de negociação, na qual o planeamento se baseia, a coordenação necessária entre entidades independentes e a disponibilização e acesso a informação global actualizada são, seguramente, pilares fundamentais das funcionalidades que os sistemas de informação de apoio às empresas virtuais devem suportar.

3.3 Sistemas de informação empresariais

Nesta secção, faz-se uma breve apresentação das soluções informáticas hoje disponíveis comercialmente para suporte à gestão e operação das empresas industriais. Procurar-se-á, de forma sucinta e não exaustiva, caracterizar alguns produtos disponíveis comercialmente, para ilustrar os principais tipos de soluções existentes. Procurar-se-á igualmente identificar as principais tendências de evolução nesta área.

3.3.1 Contexto

O desenvolvimento de mercados globais, com a emergência de complexos fluxos produtivos e as novas exigências no tempo de resposta aos clientes, bem como a necessidade de se conhecer em tempo útil a informação logística associada ao produto e ao estado de execução de uma encomenda, originaram novos requisitos nos sistemas de informação empresariais.

Os sistemas de informação para coordenação, planeamento e controlo das actividades produtivas (fisicamente distribuídas) tornaram-se definitivamente um factor chave na gestão empresarial dos anos 90. A gestão global dos recursos fabris e da cadeia de fornecimento, assim como a análise dinâmica e em tempo-real dos processos produtivos com abrangência local e global, são questões que os sistemas de informa-

ção empresariais actuais procuram tratar com a adopção de novas tecnologias ao nível de infra-estruturas de comunicação, e com a adopção de metodologias e abordagens algorítmicas anteriormente não exploradas por manifesta limitação dos meios computacionais disponíveis.

3.3.2 Os sistemas ERP, APS e SCM

Desde meados da década de 90 que a comunidade empresarial tem sido "bombardeada" com a sigla ERP – acrónimo de *Enterprise Ressource Planning* – anunciada como a solução definitiva para os complexos problemas de gestão empresarial, em todos os seus níveis de abrangência. A sigla ERP foi introduzida em 1992 pelo Gartner Group, como um marco na passagem de uma gestão eminentemente centrada e orientada para os materiais, para uma gestão focada e orientada no serviço ao cliente (Marsh and Moulding, 1997). Procurou-se, com esta nova denominação, assinalar a chegada ao mercado de sistemas de informação empresariais com integração das áreas financeira, da produção e da distribuição. Refira-se que o final desta década tem sido particularmente fértil no aparecimento de novos conceitos e siglas neste domínio, destacando-se na linha dos sistemas ERP, uma ideia introduzida pela PeopleSoft (PeopleSoft, 1998) que nos parece exprimir, cabalmente o âmbito dos sistemas de informação da próxima geração, os sistemas ERO, acrónimo de *Enterprise Resource Optimization*.

Numa primeira análise, e como resultado do poderoso desenvolvimento das tecnologias de informação dos últimos anos, podem considerar-se os sistemas de informação empresariais ERP como a aglutinação, de forma integrada (isto é, sobre a mesma informação), da maioria das actividades empresariais. Mas uma análise mais pormenorizada permite, talvez mais correctamente, encarar os sistemas ERP como a evolução natural do tradicional sistema MRP-II orientado segundo uma perspectiva departamental, no sentido de um funcionamento orientado ao *processo de negócio*. Ou seja, ao nível da gestão e planeamento das actividades produtivas, parece não se terem verificado mudanças significativas nas metodologias e lógicas de planeamento, pelo que na realidade se mantêm a maior parte das limitações tradicionalmente identificadas nos sistemas correntes de gestão, tais como o MRP-II.

Na prática, um ERP integra, sobre uma estrutura de informação comum, todas as aplicações associadas às diferentes áreas funcionais de uma organização, nomeada-

mente financeira, comercial, produtiva, de aprovisionamentos, de recursos humanos, de qualidade, de manutenção, etc. Por outro lado, sendo um sistema de gestão de âmbito global, deverá integrar igualmente todos os níveis de planeamento tradicionais, nomeadamente, o estratégico, o táctico e o operacional.

Contudo, a necessidade de metodologias de planeamento e escalonamento mais poderosas, respectivamente nos níveis táctico e operacional, conduziram recentemente ao aparecimento de novos sistemas de informação, referidos na literatura como sistemas avançados de planeamento (APS – *Advanced Planning Systems*) (Dullin, 1998; Hess, 1998) que procuram colmatar as limitações que os ERP herdaram dos MRP-II tradicionais. Os sistemas APS são geralmente orientados a áreas específicas, procurando explorar a estrutura do problema que procuram resolver, e recorrem a técnicas avançadas de optimização, nomeadamente: programação linear, algoritmos evolutivos, programação por restrições e meta-heurísticas. Estes sistemas são muita das vezes usados conjuntamente com modelos de simulação (Lybrand and Logiplan, 1998).

Constata-se assim que os novos paradigmas organizacionais, atrás caracterizados (ver 3.2.4), colocam novas exigências de gestão que não foram até à data, satisfeitas pelos sistemas ERP disponíveis comercialmente. Esta constatação despoletou um promissor nicho de mercado na área das metodologias e abordagens informáticas para a gestão, optimização e coordenação da cadeia de fornecimento. É neste contexto que surgem mais recentemente, os sistemas empresariais SCM (*Supply-Chain Management*). Estes sistemas de informação suportam, de forma integrada, a coordenação e a optimização de toda a cadeia logística. Numa fase de execução, apoiam a tomada de decisões em cenários de desregulação de um determinado nó da cadeia de fornecimento.

Em (Pillep et al., 1999) são identificados três tipos de soluções SCM. A primeira baseia-se na utilização de módulos específicos que se utilizam como "*add-ons*" de sistemas ERP comerciais. Este tipo de solução resulta naturalmente da estratégia delineada por alguns fornecedores de sistemas ERP, que envolve a aquisição de pequenas empresas com *know-how* específico em determinados tipos de produtos (refira-se, por exemplo, o caso de ferramentas de optimização). O segundo tipo de solução SCM é constituído por módulos autónomos, isto é, com gestão própria de informação e independentes de outros sistemas de informação, e com funcionalidades muito orientadas a problemas específicos. Geralmente, este tipo de soluções não abrange a totalida-

de da cadeia de fornecimento. O terceiro tipo de solução compreende módulos profundamente integrados com ERP standards, recorrendo por isso à informação disponível nesses sistemas (ao contrário dos do primeiro tipo, estes módulos são "indissociáveis" dos sistemas ERP).

Ao nível do planeamento, são considerados três níveis distintos:

- *Configuração da estrutura da cadeia de fornecimento* - compreende a modelação da estrutura produtiva e logística, tendo um âmbito de longo prazo (tem, naturalmente, um carácter estratégico).
- *Planeamento da cadeia de fornecimento* - compreende o planeamento da procura, do nível de stocks, dos transportes e dos níveis de ATP.
- *Execução e controlo* - nível operacional envolvendo a execução das ordens de produção nas unidades produtivas, ordens de transporte, etc.

Assim, em primeira instância, os sistemas SCM podem ser vistos como optimizadores multi-empresa, ao nível do planeamento das actividades de produção e distribuição, e com controlo e monitorização do fluxo de produtos em toda a cadeia de fornecimento. A sua funcionalidade mais importante é talvez a definição dos níveis de produção de cada empresa nos vários estágios do processo de fabrico, entrando em linha de conta com custos de produção, transporte e armazenamento estendidos a toda a cadeia de fornecimento.

3.3.2.1 Funcionalidades

A Figura 3.8 ilustra as principais funcionalidades que se podem identificar nos sistemas ERP, SCM e APS, procurando igualmente realçar a interacção existente entre esses sistemas e a sua complementaridade.

Note-se que os fornecedores tradicionais de sistemas ERP, num esforço contínuo para ganharem a quota de mercado que as soluções APS e SCM procuram ocupar, anunciam constantemente novas funcionalidades para os seus produtos, nomeadamente ao nível da optimização da cadeia de fornecimento, ao nível da negociação/aceitação de encomendas e também ao nível do planeamento fabril.

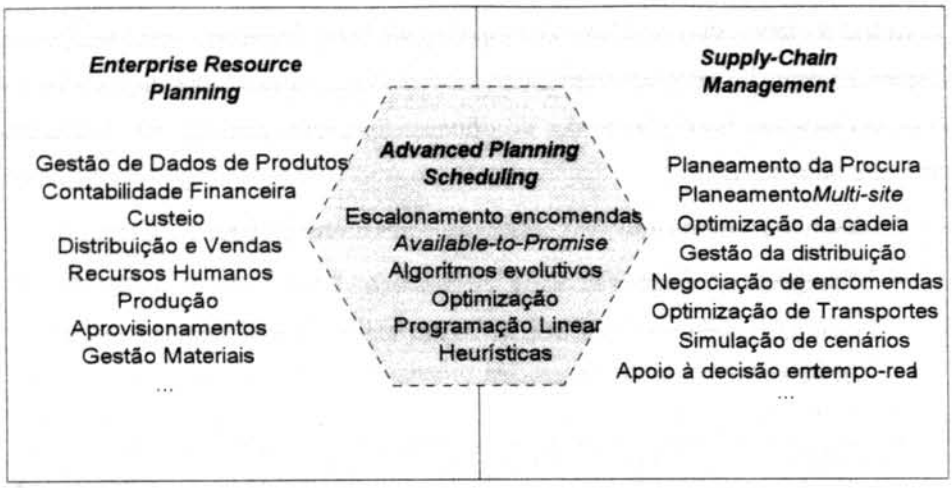


Figura 3.8 – Funcionalidades dos sistemas ERP/APS/SCM

Contudo, o ênfase dos ERP tem sido posto na empresa/organização no seu todo, mas numa perspectiva que se poderia considerar como "egocêntrica". Os sistemas SCM, por seu lado, estão mais orientados à gestão e coordenação do contexto em que a organização se insere, resultando por isso funcionalidades que são complementares das anteriores.

Da "fusão" dos três tipos de sistemas ou, em primeira instância, da estreita cooperação entre eles, resultaria um sistema de informação de elevado desempenho e de âmbito intra e inter-empresarial. Esta aglutinação de funcionalidades em torno dos ERP tem sido, ao nível de mercado, prosseguida com duas estratégias diferentes: aquisição por parte dos grandes fornecedores de sistemas ERP de empresas terceiras, com produtos nas áreas SCM e APS; ou desenvolvimento próprio de novos módulos ou reestruturação de outros já existentes, por forma a conseguir maior desempenho e novas funcionalidades.

Deve referir-se que a tendência de aglutinação de funcionalidades abrange todos os níveis de planeamento, desde a cadeia de fornecimento à linha produtiva numa unidade fabril. Contudo a especificidade de algumas áreas industriais "obriga" a que sejam necessários módulos externos ou mesmo a ligação a sistemas "legados", pelo que, simultaneamente, têm os ERP sido enriquecidos com interfaces que permitem a cooperação *on-line* entre produtos diferentes.

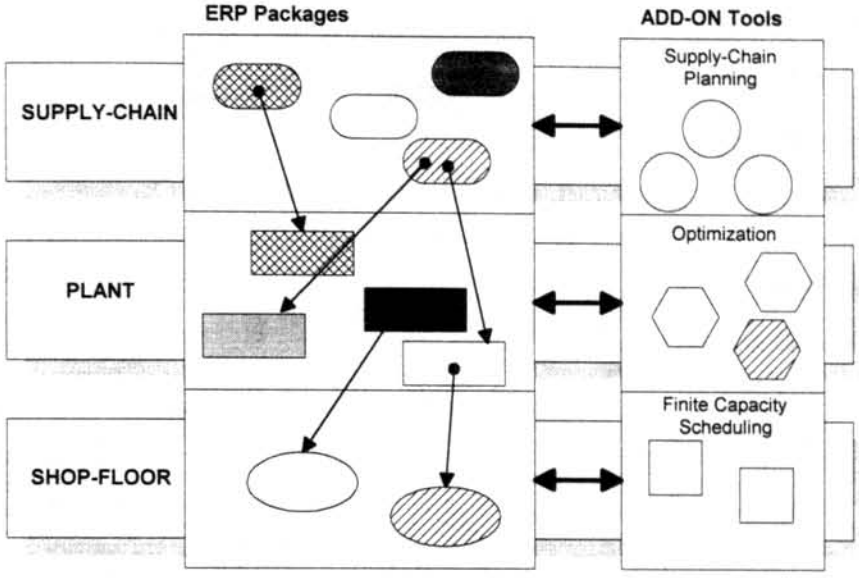


Figura 3.9 – Relações entre sistemas ERP e APS, a vários níveis de planeamento (adaptado de (Lybrand and Logiplan, 1998))

A Figura 3.9 ilustra as interacções, a diferentes níveis, entre os sistemas ERP e os módulos adicionais, tais como os sistemas APS.

Assim, cada vez mais se assiste ao enriquecimento dos sistemas empresariais pela adição de novas funcionalidades. Contudo, este aumento constante das funcionalidades origina, ao mesmo tempo, novas dificuldades de implementação, resultando num esforço crescente de configuração do sistema de informação. Como resposta a esta constatação, os fornecedores de soluções ERP têm definido metodologias de implementação (envolvendo a análise e redesenho dos processos) e apoiadas por modelos de referência de processos de negócio. Além disso, disponibilizam versões pré-configuradas e simultaneamente "enriquecidas" dos seus sistemas, para segmentos específicos (por exemplo, indústria automóvel, indústria química, etc).

3.3.2.2 Tendências de evolução

Actualmente, o desenvolvimento dos ERP encontra-se claramente direccionado para os processos logísticos e gestão da cadeia de fornecimento e para os processos de negócio, com um particular ênfase na orientação ao cliente. Da análise dos principais produtos

comerciais, identificam-se neste momento variadas tendências de desenvolvimento que se referem de seguida.

A adopção da Internet

Integração com a Internet, nomeadamente para os processo de entrada e seguimento de encomendas, divulgação de informação técnica de produtos, troca de informação, etc.

Apoio à negociação de encomendas

Ferramentas de apoio à tomada de decisões em tempo útil, nomeadamente, mecanismos de negociação de encomendas pela análise em "tempo-real" da capacidade dos recursos críticos. Procura-se, com esta funcionalidade, ultrapassar a limitação habitual dos sistemas de planeamento da produção, quanto ao facto de considerarem fixos os tempos de produção.

Configuradores de produto

Como resultado da cada vez mais frequente interacção com o cliente e da consequente personalização dos produtos encomendados, os fornecedores de ERP apostam fortemente no desenvolvimento dos denominados *configuradores de produto*, procurando assim quebrar a rígida estrutura de dados geralmente associada aos sistemas tradicionais de gestão de produção (por exemplo, o *Bill-of-Materials* do MRP-II).

A evolução para a tecnologia dos componentes

A evolução progressiva para uma economia global e a consequente emergência de novas formas de transacção entre os participantes de um negócio (como, por exemplo, o advento do comércio electrónico) , tem conduzido a novas arquitecturas tecnológicas para os sistemas de informação empresariais. Tal evolução passa nomeadamente pela adopção da *tecnologia orientada a objectos* e mais recentemente pela adopção de *tecnologia baseada em componentes,* com o recurso a infra-estruturas distribuídas baseadas em CORBA, DCOM e JAVA. Assiste-se, assim, a uma rápida transição das arquitecturas eminentemente monolíticas para arquitecturas baseadas em componentes (Mauth, 1998).

Uma arquitectura monolítica (Figura 3.10) assenta numa base de dados centralizada e numa aplicação constituída por diferentes módulos, correspondentes a um igual número de áreas funcionais da organização.

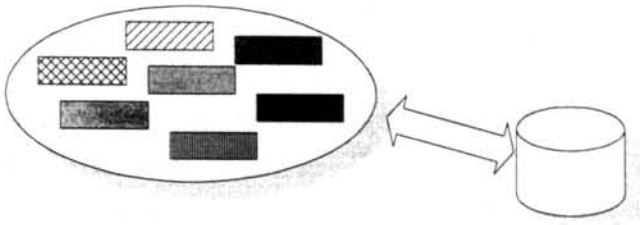


Figura 3.10 – Arquitectura monolítica

Uma arquitectura baseada em componentes permite "construir", à medida do problema, novos módulos funcionais, a partir de macro-objectos (componentes) que, suportados por uma infra-estrutura tecnológica distribuída, permitem uma adequação natural do sistema de informação à dispersão geográfica do próprio problema.

Para além de facilitar grandemente os processos de configuração e de inclusão de novas funcionalidades, a adopção da tecnologia de componentes, associada à definição de interfaces *standard*, possibilita a cooperação estreita com aplicações/módulos externos à própria solução. Tal permite a coexistência, no mesmo sistema de informação, de componentes provenientes de diferentes fornecedores e possibilita, em cada caso, a combinação dos que melhor se adequam ao problema em questão.

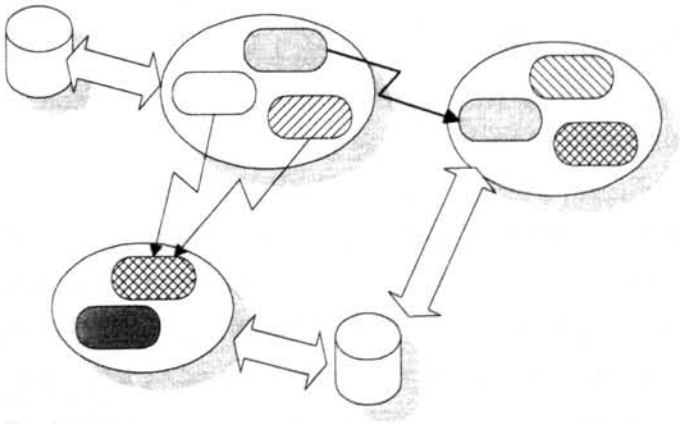


Figura 3.11 – Tendência dos ERP: arquitecturas baseadas em componentes

3.3.3 O mercado de ERP

O mercado de sistemas ERP tem apresentado nos últimos anos um crescimento significativo. Em 1997, cresceu 20%, representando cerca de 14.4 mil milhões de dólares (fonte IDC - International Data Corporation). As expectativas de crescimento continuam ainda muito elevadas, e segundo a AMR (Advanced Manufacturing Research), este mercado atingirá aproximadamente 52 mil milhõesde dólares, em 2002.

Por outro lado, as alianças estratégicas de alguns fornecedores com as principais empresas de consultadoria internacionais têm assegurado, a um reduzido número de fornecedores de soluções ERP, uma significativa quota de mercado. Em 1998, os principais fornecedores de soluções ERP incluíam a SAP (Alemanha), PeopleSoft (EUA), Baan (Holanda), J. D. Edwards (EUA) e Oracle (EUA), representando, por si só, 64% do mercado total. Outros fornecedores importantes incluem a Intentia International (Suécia), a JBA (Reino Unido) e a System Software Associates (EUA). O líder de mercado é a SAP com uma quota superior a 35% e superior ao conjunto dos cinco concorrentes mais directos (fonte IDC).

A análise das funcionalidades dos ERP mais importantes (realizada informalmente no âmbito deste trabalho) permite concluir que são mais os pontos em comum do que as diferenças entre eles. O factor diferenciador mais relevante é talvez o custo global da solução (*software*, *hardware*, formação e serviços de consultadoria) a par com a metodologia de implementação utilizada.

3.3.4 O mercado de SCM

Segundo a AMR, o mercado de soluções para a gestão da cadeia de fornecimento atingirá 13.6 mil milhões de dólares em 2002, com um crescimento anual de 50%. Este mercado encontra-se segmentado em dois níveis principais: planeamento (*supply-chain planning*) e execução (*supply-chain execution*). Se actualmente existe, ao nível do segmento de execução, grande fragmentação entre as quotas de mercado dos fornecedores, ao nível do segmento de planeamento, este é dominado por dois fornecedores: I2 Technologies e Manugistics. Outros fornecedores significativos de soluções SCM incluem a RedPepper, Paragon, Chesapeak e a SynQuest.

Todos os produtos disponíveis no mercado recorrem a núcleos avançados de planeamento e optimização, nomeadamente com o uso de *Algoritmos Genéticos* no caso dos produtos Rhythm (I2 Technologies) e SynQuest6 (SynQuest) e o uso de Programação Linear (CPLEX Solver) no caso do Paragon 6.0 (Paragon Management System) e do MIMI (Chesapeak Decision Sciences).

Os produtos SCM encontram-se geralmente organizados segundo um conjunto de componentes, compreendendo as seguintes áreas:

- Gestão e negociação de encomendas;
- Planeamento da procura;
- Planeamento e monitorização da produção;
- Planeamento e gestão dos transportes;
- Gestão da informação.

As arquitecturas destes produtos procuram naturalmente maximizar o desempenho computacional e a eficiência, com soluções que passam, por exemplo, por forçar que os algoritmos de planeamento e optimização residam permanentemente em memória, procurando assim minimizar as transacções decorrentes de acessos a informação residente na base de dados.

Porque ainda se trata de um mercado recente e por explorar, espera-se que nos próximos anos se venha a assistir a um significativo desenvolvimento das soluções SCM, nomeadamente ao nível de componentes para planeamento reactivo e em tempo-real, assim como para a implementação de mecanismos avançados e autónomos de negociação. Por outro lado, irá seguramente observar-se um abandono progressivo dos sistemas de gestão de informação centralizados, que irão ser substituídos por sistemas distribuídos.

3.3.5 Conclusão

Os sistemas de informação empresarias continuarão a ter nos próximos anos um significativo desenvolvimento, essencialmente impulsionado pelo uso intensivo da tecnologia distribuída de objectos. Este desenvolvimento dever-se-á em parte ao facto de, actualmente, as funcionalidades disponíveis não se adequarem cabalmente aos desafios que se

colocam às novas formas de organização e de colaboração empresarial e à necessidade de responder rapidamente e de forma fiável às solicitações dos clientes.

3.4 Requisitos para os Sistemas de Informação

É hoje crescente a necessidade de dispor de ferramentas para apoiar paradigmas organizacionais com elevados requisitos de flexibilidade, de cooperação e de coordenação, em detrimento de soluções informáticas orientadas às áreas funcionais da empresa. Por outro lado, constata-se que os sistemas de informação empresariais tradicionais são, na generalidade dos casos, rígidos, monolíticos, centralizados, fechados, dependentes de tecnologia "proprietária" e baseados em metodologias, tais como o MRP II no planeamento da produção, desactualizadas e desadequadas às novas realidades.

Parece também ser claro que as soluções avançadas dos sistemas SCM disponíveis comercialmente ainda não suportam adequadamente as exigências colocadas pela *Empresa Virtual*, quer ao nível funcional, quer ao nível da infra-estrutura tecnológica.

É neste contexto que a participação do autor num projecto[11] de grande dimensão, na indústria dos semicondutores, permitiu identificar, de uma forma clara, um conjunto de requisitos base para os "novos" sistemas de informação empresariais. As empresas de semicondutores envolvidas no projecto, estando inseridas em mercados de produção globais, envolviam diversas unidades de produção, de logística, de armazenamento e de serviços, distribuídas por diferentes países, assim como recorriam permanentemente a entidades externas em regime de subcontratação e *outsourcing*.

Para além dos aspectos estruturais referidos, as empresas tinham como área de negócio preferencial, o mercado dos "Circuitos Integrados para Aplicação Específica" (ASIC), isto é, fabricavam produtos à medida do cliente e estavam sujeitas por isso a uma imprevisibilidade constante e significativa da procura.

Pelas suas características, estas empresas podem ser conceptualizada como Empresas Virtuais, pelo que os requisitos identificados[12] no âmbito deste projecto

[11] Projecto X-CITTIC – *A Planning and Control System for Semiconductor Virtual Enterprise*, (1996/1998) financiado pela Comissão Europeia no âmbito do programa ESPRIT

[12] Refira-se que nas empresas participantes tiveram lugar reorganizações significativas ao nível dos sistemas de informação, nomeadamente com a instalação de um ERP "estado-da-arte", o SAP R/3, pelo que, em certo sentido, se podem considerar os requisitos dos utilizadores ainda "mais válidos".

constituem naturalmente um subconjunto dos requisitos gerais da *Empresa Virtual*. Refira-se que, neste contexto, o ênfase foi sempre colocado nas actividades relacionadas com a produção. Assim, como requisitos fundamentais foram identificados os seguintes (Billings and Sturm, 1996), (Frederix, 1996):

Cooperação

É necessário estabelecer um quadro de cooperação entre entidades, que permita a negociação, o estabelecimento de regras de negócio, etc.

Coordenação

Deve ser possível: "balancear" a capacidade global; minimizar o inventário ao longo de toda a cadeia de fornecimento; comprimir os tempos de resposta e o tempo total de processo.

Aceitação/Negociação de encomendas

São necessários mecanismos que permitam: determinar, com fiabilidade, datas de entrega; responder rapidamente aos clientes, através de procedimentos expeditos para avaliação dos seus pedidos; renegociar encomendas; interagir com os clientes; gerar avisos no caso de potenciais atrasos.

Planeamento e controlo da produção

As capacidades "reais" (envolvendo toda a rede de unidades produtoras) devem ser avaliadas de uma forma fiável e eficiente; deve ser possível analisar o efeito na capacidade do "mix" de produtos; o planeamento deve ser optimizado para toda a cadeia de fornecimento, com a finalidade de se conseguir reduzir o WIP, reduzir o *lead-time* e de aumentar a taxa de utilização; análises "what-if" e de cenários alternativos devem ser possíveis a nível de toda a rede.

Monitorização

São necessárias formas de seguimento da execução de encomendas, ao longo de toda a cadeia de fornecimento; são necessários mecanismos inteligentes de pré-avisos de ruptura ou incumprimento de datas.

A confrontação, segundo uma perspectiva "funcionalista", dos requisitos apresentados, com as funcionalidades dos sistemas de informação empresariais, no merca-

BIBLIOTECA F.E. PORTO

do, permite inferir que actualmente estes últimos não se adequam ou não apoiam adequadamente a gestão da *Empresa Virtual*.

Assim, considerando a autonomia de cada entidade na *Empresa Virtual*, por forma a suportar a sua acentuada distribuição e heterogeneidade e por forma a suportar mecanismos de negociação, cooperação e coordenação, propõem-se os seguintes *requisitos para os sistemas de informação para empresas virtuais* (Azevedo and Toscano, 1999):

- suporte à cooperação e à coordenação, a uma escala mundial (distribuição), quer em actividades de desenvolvimento, quer em actividades de planeamento colaborativo das actividades de produção;
- mecanismos eficientes de actualização, partilha e distribuição de informação em tempo-real, a todas as entidades envolvidas;
- suporte à escalabilidade - o sistema de informação deve estar apto a integrar novos componentes ou novos "objectos de negócio", sem afectar os outros componentes já existentes no sistema;
- o sistema de informação deve ser flexível e aberto, de tal modo que suporte a dinâmica de reconfiguração e reorganização da *Empresa Virtual*;
- facilidade de configuração e de arranque;
- funcionamento em tempo-real;
- utilização de tecnologia *standard* e evolutiva;
- arquitectura distribuída e baseada na tecnologia de objectos, por forma a permitir a distribuição de informação e de aplicações/funcionalidades;
- adequação a processos de negócio temporários;
- adequação a diferentes modelos de informação;
- suporte à tecnologia de componentes distribuídos (como, por exemplo, componentes para planeamento distribuído e baseado em negociação);

BIBLIOTECA F.E. PORTO

- adequação a um ambiente heterogéneo - a cooperação com entidades independentes e autónomas pressupõe a existência de diferentes sistemas de informação (diferentes aplicações e sistemas "legados") e diferentes infra-estruturas tecnológicas (sistemas operativos e sistemas de comunicação);
- elevado nível de segurança no acesso à informação.

3.5 Conclusões

A vantagem competitiva das empresas passa, cada vez mais, pelo desenvolvimento de novas estruturas organizacionais e novas metodologias de gestão e planeamento, apoiadas fortemente por tecnologias de informação e de comunicação.

A emergência da *Empresa Virtual*, como novo paradigma organizacional, impõe um conjunto de requisitos ao nível funcional e de infra-estrutura tecnológica, ainda não cabalmente suportados pelas soluções disponíveis comercialmente, nomeadamente, pelos sistemas de informação empresarial do tipo ERP e SCM.

Assim, existe, ao nível das metodologias de planeamento, uma enorme necessidade de dispor de soluções que, de forma eficiente, produzam planos realistas, e estejam adaptadas às novas características do mercado e das estruturas produtivas. Particularmente importantes são os instrumentos que possibilitem, de forma eficiente e em tempo-real, apoiar todo o processo de negociação e aceitação de encomendas (*order promising*). O planeamento e a avaliação da capacidade produtiva, à escala de toda a cadeia de fornecimento, revela-se igualmente um instrumento de gestão decisivo para o sucesso da *Empresa Virtual*.

Por outro lado, os aspectos de configuração e reconfiguração e, em particular, a forma como serão conceptualizadas e modeladas nos sistemas de informação, as entidades que compõem a *Empresa Virtual*, serão seguramente no futuro, pela actual falta de soluções satisfatórias, áreas de forte investigação e desenvolvimento.

O recurso intensivo a arquitecturas baseadas em tecnologias de componentes deverá permitir, a breve prazo, satisfazer tão exigentes desafios.

Capítulo **4**

Um Sistema de Apoio à Decisão para Aceitação de Encomendas

4.1 Enquadramento e domínio do problema

4.2 O sistema *Order Promise*

4.3 Tecnologias e ambiente de implementação

4 UM SISTEMA DE APOIO À DECISÃO PARA ACEITAÇÃO DE ENCOMENDAS

No capítulo anterior, identificou-se um conjunto de requisitos não contemplados adequadamente nos sistemas de informação empresariais actuais e que resultam naturalmente de novos paradigmas organizacionais, nomeadamente do aparecimento da *Empresa Virtual* (EV). Em particular, concluiu-se serem necessários, hoje em dia, instrumentos que permitam responder, em tempo útil, às solicitações dos clientes no processo de negociação e aceitação de encomendas (*Order Promising*), em ambientes eminentemente distribuídos.

Neste capítulo, é apresentado um Sistema de Apoio à Decisão (que é parte de um sistema de informação mais vasto para o planeamento e controlo da produção na indústria de semicondutores) com o qual se procurou responder a esses requisitos. Assim, e após caracterizar o domínio, começa por situar-se o Sistema de Apoio à Decisão no âmbito do projecto X-CITTIC. Será ainda feita uma referência à sua arquitectura geral e ao modelo de dados desenvolvido.

Após a definição da funcionalidade genérica do sistema, será apresentada a sua arquitectura, caracterizando-se os componentes que a constituem. Atendendo ao facto de se terem usado tecnologias "estado-da-arte", nomeadamente, tecnologia de componentes distribuídos sobre uma infra-estrutura compatível com a especificação CORBA, será feita, no final do capítulo, uma referência breve às ferramentas utilizadas na implementação do sistema.

4.1 Enquadramento e domínio do problema

No capítulo anterior, apresentou-se um conjunto de requisitos, identificados segundo duas perspectivas: uma essencialmente funcionalista e outra mais tecnológica ou de ênfase infraestrutural. Estes requisitos, que naturalmente orientaram o trabalho desenvolvido no âmbito desta dissertação, resultaram essencialmente da constatação das limitações actualmente existentes nos sistemas de informação empresariais (ERP), quando utilizados como instrumentos de apoio à gestão de estruturas produtivas distribuídas.

Neste capítulo, será feita uma descrição de um Sistema de Apoio à Decisão (SAD) para o processo de aceitação/negociação de encomendas (*order promising*) em estruturas de produção distribuídas. O trabalho apresentado foi, na sua quase totalidade, desenvolvido pelo autor no âmbito do projecto europeu X-CITTIC (*A Planning and Control System for Semiconductor Virtual Enterprises* – ESPRIT E-20544). Este projecto, com a duração de três anos, decorreu entre 1996 e 1998 e teve por objectivo principal o desenvolvimento de uma metodologia e de um sistema de planeamento e controlo da produção e da logística associada, em redes empresariais complexas. Foi realizado por um consórcio envolvendo três instituições académicas: Imperial College (Reino Unido), Fraunhofer Gesellschaft IPA (Alemanha) e INESC (Portugal); três empresas industriais: MITEL Semiconductors (ex GEC-Plessey, Reino Unido), Temic (Alemanha) e Alcatel (Bélgica); e um fornecedor de sistemas de informação: Nimble (Bélgica). A indústria dos semicondutores, constituiu a área de aplicação do projecto. Trata-se seguramente de uma das indústrias mais avançadas em termos tecnológicos, e uma das mais exigentes em termos de planeamento e de gestão da produção, para além de ser uma das que apresenta uma mais rápida evolução.

Como referido, uma das tarefas do projecto (da responsabilidade do autor desta dissertação) foi o desenvolvimento de um Sistema de Apoio à Decisão (SAD) que permitisse apoiar o processo de aceitação/negociação de encomendas (*order promising*) em tempo-real, através de uma rápida avaliação da capacidade produtiva (ao longo da rede de empresas) e incluindo os aspectos logísticos e de coordenação decorrentes de uma organização produtiva em rede. Com a finalidade de se clarificar o domínio, e consequentemente o contexto de utilização do sistema, começa por apresentar-se uma descrição breve do processo produtivo na indústria de semicondutores. Far-se-á igualmente referência a algumas das questões que, neste tipo de indústria em particular, tornam muito difícil o processo de planeamento.

4.1.1 Domínio do problema

O fabrico de semicondutores (circuitos integrados) assenta num processo produtivo complexo, baseado em equipamento tecnologicamente avançado e com requisitos extremos de precisão e repetitibilidade. O facto desta indústria, por um lado envolver elevadíssimos investimentos em infra-estruturas produtivas, e, por outro, ter um mercado

alvo de grande imprevisibilidade, ao nível da procura de produtos existentes e de novos produtos, faz com que os processos de planeamento, escalonamento e controlo da produção desempenhem um papel fundamental, quer na viabilidade do investimento, nomeadamente ao nível de se garantir elevadas taxas de utilização dos equipamentos, quer na resposta rápida e fiável às solicitações do mercado.

O processo de fabrico

O fabrico de circuitos integrados envolve centenas de operações, geralmente agrupadas em quatro estágios principais:

- *Fabrico de "wafers" (Wafer Fab)* – Esta fase do processo, a mais complexa e de maior investimento, envolve a impressão multi-camada de circuitos em "bolachas" de silício (*wafers*). A *wafer* é um disco de silício, tradicionalmente de 6 ou 8 polegadas de diâmetro, contendo um determinado número de circuitos integrados ainda não funcionais[13], que serão individualizados em fases posteriores do processo de fabrico.

- *Teste eléctrico (Probe)* – Estágio no qual é efectuado o teste individual de cada circuito (*die*) existente na *wafer*, sendo marcados os que não estiverem de acordo com as especificações.

- *Montagem (Assembly)* - Nesta fase, os circuitos existentes na *wafer* são individualizados. Depois de serem soldados os terminais para interligação ao exterior, são encapsulados num dos inúmeros invólucros possíveis (aos quais estão associados produtos finais diferentes).

- *Teste final (Final test)* – Envolve o teste funcional e eléctrico de cada um dos circuitos integrados.

É habitual considerar-se o agrupamento dos estágios referidos em dois grandes grupos: *front-end* e *back-end* (ver Figura 4.1). O primeiro engloba os estágios de fabrico de *wafers* e o respectivo teste, o segundo grupo engloba a montagem e o teste final. Como separação entre estes dois grupos existe geralmente um armazém "estratégico", referido na terminologia deste sector industrial por *die bank*. Este ponto de armazena-

---

[13] Nesta fase, os circuitos constituídos na *wafer* são designados por *die*. O número de *die* numa *wafer* é variável, dependendo não só da complexidade do próprio circuito e consequentemente da sua dimensão, como também do diâmetro da *wafer*.

mento funciona como um ponto de desacoplamento no processo de produção. Até ao *die bank*, o planeamento da produção é maioritariamente baseado em previsões da procura e a jusante deste, a produção é despoletada principalmente por encomendas firmes dos clientes (ver secção 2.2).

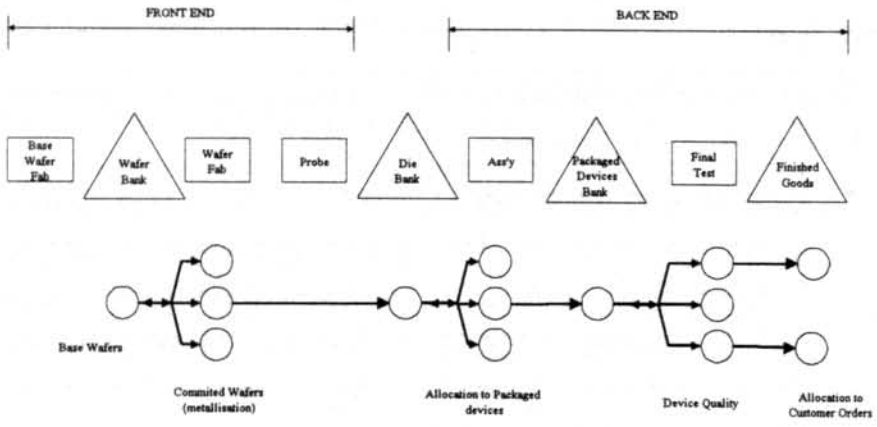


Figura 4.1 – Processo de fabrico de semicondutores

O tempo de ciclo típico para cada um dos estágios é de aproximadamente 40 dias para a *wafer fab*, 2 a 7 dias para o *probe*, 18 a 24 dias para o *assembly* e 3 a 7 dias para o *final test*.

A estrutura produtiva nesta indústria é do tipo *divergente*. No caso geral, uma *wafer* de silício bruta (*raw wafer*) pode ser a base de vários tipos de *wafers* diferentes. Por outro lado, uma mesma *wafer* pode originar diferentes produtos finais (por exemplo, com diferentes tipos de invólucro). Além disso, para certos tipos de produtos, e ao longo do processo produtivo, verifica-se a subdivisão noutros tipos de produtos, como resultado da separação em diferentes categorias de *performance*, nomeadamente, velocidade de funcionamento, temperatura de funcionamento, etc.

Para além do armazém referido (*die-bank*) , é habitual a existência no processo produtivo de dois outros: o armazém de *wafers* "difusas" (*mirror bank*) – contém *wafers* já com algumas operações efectuadas (as operações de base e comuns a diferentes tipos de *wafers*), prontas a serem processadas na *wafer fab*; e o armazém de produtos finais.

Os fabricantes de semicondutores apresentam geralmente uma organização produtiva constituída por diversas unidades fabris independentes, geograficamente distribuídas, e dedicadas a cada um dos estágios que constituem o processo produtivo. Geralmente, para cada estágio, existem diversas unidades de produção alternativas. Além disso, é habitual recorrer-se à subcontratação em qualquer das fases do processo produtivo e em particular no estágio de *assembly*.

Complexidade do processo de planeamento

O planeamento e o escalonamento da produção na indústria de semicondutores são considerados por vários autores como talvez os de maior complexidade (Atherton and Atherton, 1995), (Ovacik and Uzsoy, 1997).

Ao nível global, o planeamento da produção apresenta dificuldades acrescidas relativamente aos problemas habituais, em resultado dos seguintes factores:

- multiplicidade de unidades produtivas envolvidas no processo produtivo, distribuídas geograficamente;
- ciclo de vida dos produtos extremamente reduzido, restringindo significativamente o recurso a estratégias baseadas em *stocks* de segurança;
- grande variedade de produtos finais, de elevada complexidade e com níveis variados de procura;
- processo produtivo muito complexo, particularmente ao nível da *wafer fab*; neste estágio, o processo é do tipo *job shop* e envolve aproximadamente três centenas de passos principais (com diversas operações) e com inúmeros fluxos reentrantes;
- dificuldade em quantificar a capacidade produtiva, devido à grande variedade de equipamento e à interdependência entre os tempos de ciclo e o nível de carga dos recursos;
- dificuldade em estimar o rendimento produtivo (*yield*) de determinados produtos: pequenos distúrbios no equipamento podem originar perdas significativas de produção;
- capacidade dependente do nível de "mix" dos produtos.

Para uma revisão bibliográfica de modelos de planeamento e de escalonamento da produção utilizado neste contexto, sugere-se a leitura dos artigos de (Uzsoy et al., 1992) e de (Frederix, 1996).

4.2 O sistema *Order Promise*

4.2.1 Enquadramento no projecto X-CITTIC

Na sequência das limitações e dos problemas identificados no âmbito do projecto PASSE[14] e com base num conjunto de requisitos de empresas industriais de semicondutores (Billings and Sturm, 1996), o projecto X-CITTIC desenvolveu uma solução informática conceptualmente organizada em *módulos* funcionais, sendo por sua vez estes módulos constituídos por vários *componentes*[15]. A funcionalidade e o âmbito de um módulo são definidos pela interacção conjunta dos componentes que o constituem e pela interacção com o exterior. Além disso, um componente pode ser usado por mais do que um módulo, podendo apresentar um âmbito global ou local, consoante a sua visibilidade e acção sejam estendidas a toda a organização ou restringidas a uma determinada entidade em particular.

A identificação e análise de requisitos e a especificação preliminar do sistema, deu origem a um modelo funcional preliminar, representado parcialmente na Figura 4.2.

---

[14] *Planning, Scheduling, and Manufacturing Control Systems*, projecto europeu financiado pela Comissão Europeia (E-9245), decorreu no período de Dezembro de 1993 a Outubro de 1995. O projecto PASSE era em certo sentido uma "medida de acompanhamento" com origem num grupo de interesse em planeamento, escalonamento e controlo da produção e focado preferencialmente na área da microelectrónica.

[15] Em primeira instância, um *componente* é um bloco de *software* executável, com uma interface bem definida, e com gestão própria e independente de qualquer aplicação. Uma vantagem imediata da *tecnologia de componentes* é a possibilidade de um sistema poder ser constituído por "blocos" provenientes de diferentes fornecedores, podendo evoluir funcionalmente com a inclusão de novos componentes ou por substituição dos existentes, sem necessitar de se reconstruir todo o sistema. Os componentes podem, por isso, ser vistos como blocos construtivos para a nova geração de aplicações distribuídas. Uma boa referência acerca desta temática é o livro de Szyperski (Szyperski, 1997).

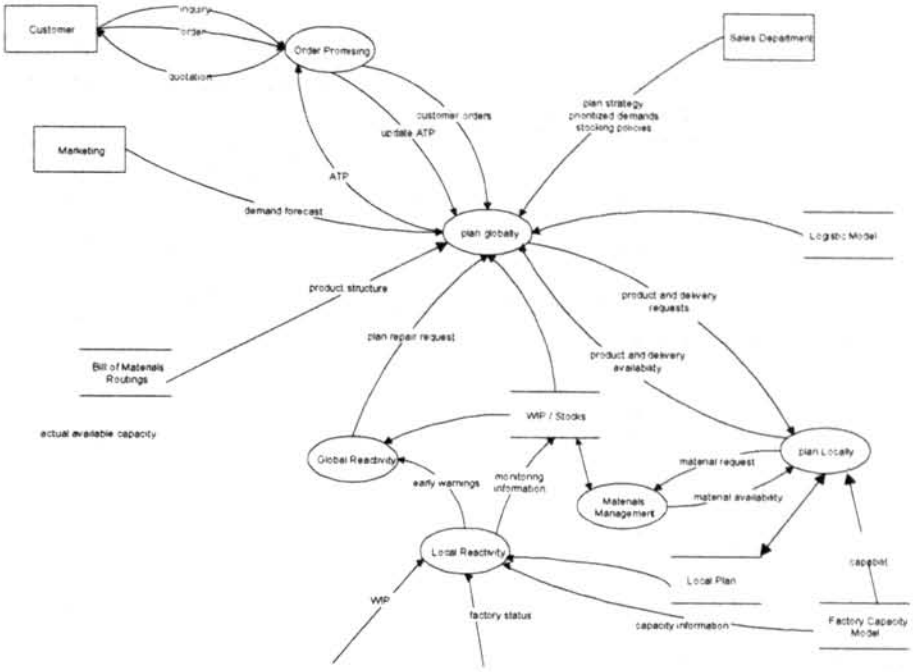


Figura 4.2 – Modelo funcional preliminar do sistema X-CITTIC

Este primeiro modelo permitiu criar uma base de trabalho que, numa fase posterior, deu origem a diferentes componentes que, interactuando de forma adequada, implementam as funcionalidades requeridas.

O desenvolvimento da solução X-CITTIC considerou dois níveis funcionais principais: um nível pro-activo e um nível reactivo.

No nível pro-activo, são disponibilizadas duas funcionalidades principais que correspondem a dois módulos do sistema: módulo de *Order Promise* e módulo de planeamento fino (*Fine Planner*) ou de pós-optimização. No nível reactivo, inclui-se o módulo de monitorização e de controlo reactivo. No apoio a estes módulos, e com âmbito local e global, encontra-se o módulo de gestão de informação. A Figura 4.3 apresenta uma representação esquemática da interligação entre os módulos que compreendem a solução X-CITTIC.

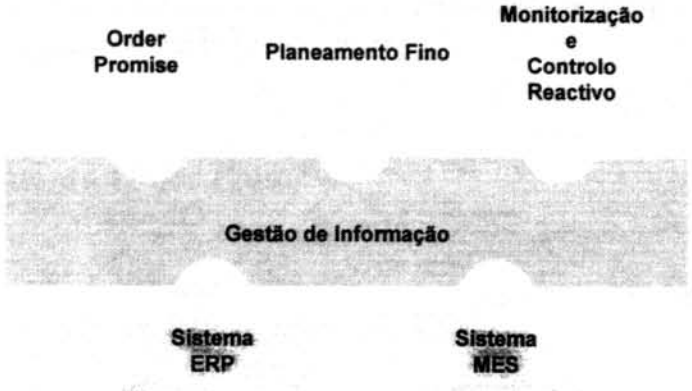


Figura 4.3 – Módulos funcionais da solução X-CITTIC

Módulo de Order Promise

Este módulo é responsável pela recepção dos pedidos provenientes dos clientes e pelo planeamento, em tempo útil, de novas encomendas, com a determinação de datas de entrega, através da avaliação das capacidades ao longo da estrutura produtiva. Este módulo é igualmente responsável por implementar o mecanismo de coordenação inerente ao planeamento de uma nova encomenda e por determinar o respectivo percurso, ao longo das unidades da rede. É constituído por diversos componentes de âmbito global e de âmbito local.

Módulo de "planeamento fino"

Após o processo de negociação, apoiado pelo módulo de *Order Promise*, e a confirmação das novas encomendas, este módulo é activado com a finalidade de proceder à pós-optimização das encomendas, ao nível global, e de gerar planos para as entidades envolvidas, de acordo com a rota planeada no módulo de *Order Promise*. É constituído por diversos componentes de âmbito global e de âmbito local.

Módulo de monitorização e controlo reactivo

É o módulo que procura garantir que a execução dos planos desenvolvidos nos módulos anteriores sejam executados de forma a respeitar os compromissos assumidos com os clientes. Numa analogia com a teoria dos sistema de controlo, este é o

elemento que fecha o anel de controlo, realimentando o controlador do sistema. É constituído por componentes globais e locais.

Módulo de gestão de informação

Toda a funcionalidade da solução X-CITTIC assenta num sistema de informação distribuído, configurável e escalável, de acordo com a estrutura produtiva em causa. Este módulo de gestão de informação, de âmbito local e global, é baseado num modelo de objectos de informação, comum a toda a empresa, e constitui o elemento principal na integração dos restantes módulos. É também o módulo que garante a interface com os sistemas de informação externos, nomeadamente, sistemas de informação ERP e sistemas de execução fabril (MES – *Manufacturing Execution Systems*).

Posicionamento dos módulos X-CITTIC

No quadro conceptual tradicional de hierarquização dos níveis de planeamento, a solução X-CITTIC posiciona-se no nível táctico e operacional (Figura 4.4). Em particular, o módulo de *Order Promise* situa-se no nível táctico, tendo como entrada principal pedidos de clientes, e gerando, na saída, encomendas.

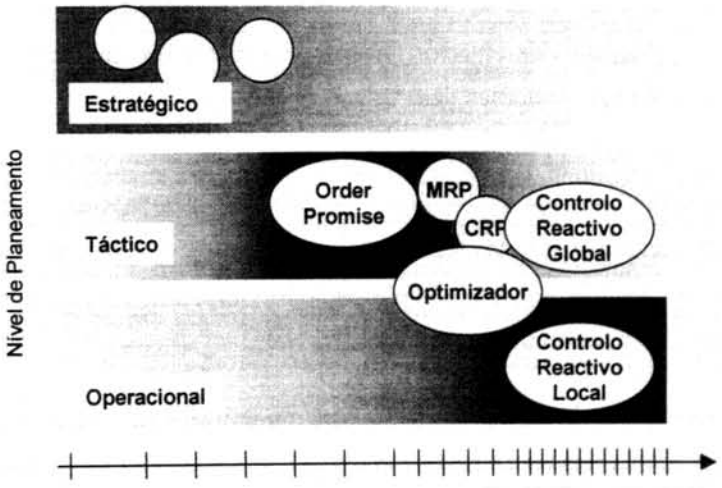


Figura 4.4 – Posicionamento da solução X-CITTIC

4.2.2 Arquitectura geral do sistema X-CITTIC

No contexto da *engenharia de software*, o sistema X-CITTIC pode ser visto como um conjunto de objectos distribuídos ou, a um outro nível de agregação, como um conjunto de componentes interactuando entre si, através de uma plataforma que proporciona transparência na localização, activação e comunicação entre objectos[16]. No caso presente, a plataforma utilizada (vulgarmente designada por *middleware*, no âmbito da engenharia de *software*) é baseada em tecnologia CORBA (acrónimo de *Common Object Request Broker Architecture*). Basicamente, o CORBA é a especificação de uma plataforma de suporte à construção de aplicações distribuídas, em ambientes heterogéneos. O elemento central que medeia o mecanismo de invocação entre objectos é o ORB (*Object Request Broker*). No âmbito do CORBA, o ORB é descrito como um "barramento" a que se ligam os componentes de *software* para interactuarem entre si.

No contexto da arquitectura, o ORB facilita, através dos seus serviços, a interligação entre os elementos constituintes da arquitectura, permitindo, nomeadamente, que um *objecto cliente* possa de forma transparente invocar um método num *objecto servidor*, que poderá estar na mesma máquina do objecto cliente ou noutra qualquer, remota. Assim, desta forma, o ORB constitui uma plataforma na qual as aplicações distribuídas de objectos são desenvolvidas, garantindo não só a portabilidade, como também a interoperabilidade dos objectos, independentemente do ambiente (linguagem de programação, sistema operativo, rede de comunicação, etc).

No sistema X-CITTIC, cada componente de *software* disponibiliza interfaces públicas bem definidas[17], através das quais, e mediante o ORB, ocorrem as interacções (invocação de métodos definidos nas interfaces). A arquitectura, baseada na *tecnologia de componentes* ao nível do projecto e implementação (Szyperski, 1997), segue um modelo "*four-tier*". A Figura 4.5 ilustra a arquitectura global do sistema.

---

[16] Não se pretende nesta secção aprofundar aspectos essencialmente tecnológicos, pelo que se referirá apenas o estritamente necessário à compreensão geral do sistema, deixando para secções posteriores aspectos não só tecnológicos como também de implementação.

[17] As interfaces são definidas através de uma linguagem de descrição de interfaces (IDL). A linguagem IDL é puramente declarativa e corresponde a um subconjunto da linguagem C++, com algumas adições, por forma a suportar os conceitos associados à distribuição.

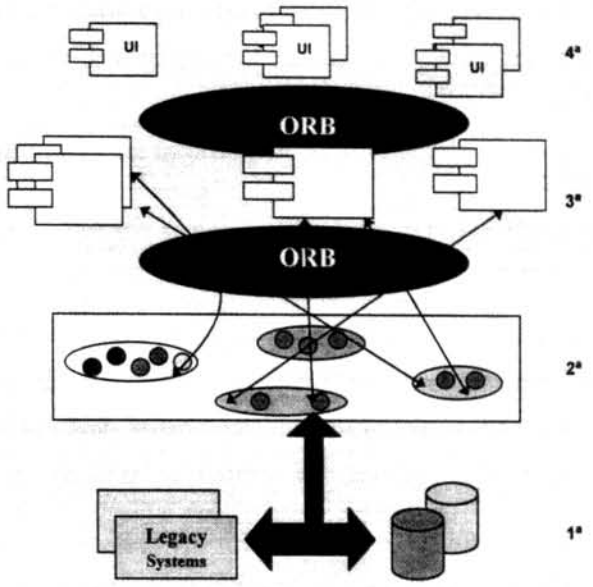


Figura 4.5 – Arquitectura "four-tier" do sistema X-CITTIC

A primeira camada é responsável pela gestão da informação estática e dinâmica, de forma distribuída, em diversas base de dados relacionais, algumas delas de âmbito estritamente local e outras de âmbito global. Esta camada implementa igualmente as interfaces ("*wrappers*") a sistemas de informação externos ao X-CITTIC, nomeadamente, sistemas "legados" (*legacy systems*) tais como ERP e MES.

A segunda camada implementa o modelo de classes de objectos a partir da informação relacional disponibilizada pela primeira camada. Nesta camada é feita, por isso, a conversão de objectos em informação relacional e de informação relacional em objectos. Além disso, esta camada constitui a base na qual os componentes de *software*, que residem na terceira camada, constroem os seus próprios modelos de objectos.

A terceira camada constitui um repositório distribuído de componentes de *software* executáveis, que implementam, quando interagem, as funcionalidades pretendidas para o sistema. Esta camada engloba componentes quer com âmbito local (por exemplo, modelos de capacidade de recursos produtivos), quer componentes globais (por exemplo, motores de planeamento).

A quarta e última camada implementa as interfaces para interacção com os utilizadores do sistema. Refira-se que esta camada não implementa, por si só, qualquer funcionalidade do sistema.

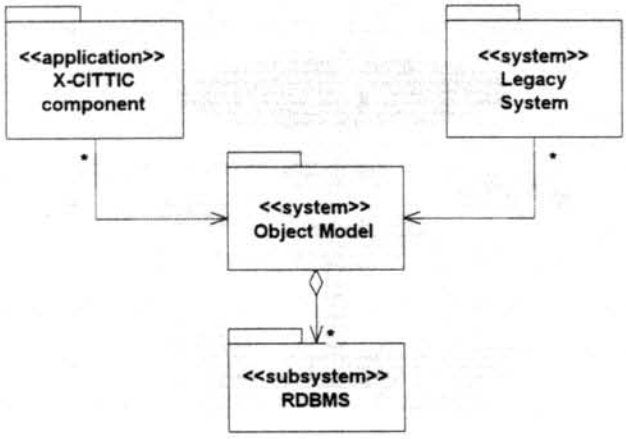


Figura 4.6 – Arquitectura básica do sistema X-CITTIC

Na Figura 4.6, é apresentada uma visão da arquitectura segundo uma perspectiva de projecto do sistema, e segundo a notação UML[18]. Esta figura procura salientar o *modelo de objectos* como o elemento central de todo o sistema. Para além de, por si só, descrever diversos aspectos do domínio do problema, o modelo de objectos é, ao nível da informação, o elemento fundamental na integração entre os diferentes componentes funcionais do sistema, nomeadamente entre os diversos componentes que constituem as aplicações X-CITTIC e os sistemas externos tais como os ERP e os MES.

---

[18] A necessidade de utilizar uma linguagem comum ao longo do processo de desenvolvimento do projecto, nomeadamente desde a identificação de requisitos, especificação e concepção de componentes até à sua integração, levou-nos a adoptar a notação UML (*Unified Modelling Language*) (Booch et al., 1999). A UML, sendo apenas uma notação para construção de modelos orientados por objectos, não constitui uma metodologia pelo que se adoptou na fase de projecto do X-CITTIC a metodologia OMT (*Object Modelling Technique*) na construção dos modelos de objectos, dinâmicos e funcionais do sistema (Rumbaugh et al., 1991).

4.2.3 Modelo de objectos

4.2.3.1 Estruturação da informação

A análise do domínio do problema permitiu, desde cedo, estruturar a informação segundo diferentes perspectivas, nomeadamente:

- *informação global e informação local* – algumas das classes identificadas no domínio são eminentemente globais, isto é, são "visíveis" em toda a *Empresa Virtual*; outras instanciarão objectos unicamente a nível local, isto é, estarão confinadas a um determinado nó (entidade) da estrutura produtiva (rede).

- *informação estática e informação dinâmica* – alguma da informação é, em certo sentido, constante ou estática, isto é, é independente do estado de execução do processo produtivo (por exemplo, sequência de operações, recursos, etc); por seu lado, a informação dinâmica muda (frequentemente) ou depende do estado de execução do processo produtivo (por exemplo, informação relativa a ordens de fabrico, lotes, etc).

O *modelo de objectos* que foi desenvolvido neste trabalho permite descrever a globalidade da empresa (ou rede) em termos das unidades produtivas que a constituem, que podem ou não, ser propriedade da empresa central, dos produtos que disponibiliza, das encomendas dos clientes, dos processos de fabrico utilizados, etc. Os componentes de *software* obtêm a informação de que necessitam através do *modelo de objectos*, garantindo este a consistência, a actualização e a integridade da informação.

Com a finalidade de se conseguir a maior normalização possível para a informação, durante as fases de análise e especificação, e em particular na construção do modelo de objectos para o sistema, recorreu-se, como base de trabalho, à especificação da SEMATECH[19].

O modelo de objectos foi desenvolvido segundo dois níveis de utilização: um local e outro global. O modelo de objectos de âmbito local instancia objectos unica-

---

[19] A SEMATECH (CIM Framework for Enterprise Modelling) é uma especificação de um ambiente integrado, flexível, modular e orientado a objecto, para o desenvolvimento de aplicações informáticas para o domínio da indústria dos semicondutores. A sua finalidade principal é definir uma arquitectura *standard* que permita a coexistência de soluções multi-fabricante num ambiente CIM (Sematech, 1996).

mente confinados a uma entidade local (uma determinada unidade produtiva, por exemplo). Este modelo é acedido unicamente por componentes de *software* igualmente de âmbito local.

O modelo de objectos de âmbito global é de certa forma mais estruturante e engloba objectos com "visibilidade" para toda a estrutura produtiva (por exemplo, instancia objectos tais como produtos finais, encomendas, etc). Este modelo é acedido por componentes globais. Num cenário normal de funcionamento do sistema existem, instanciados, um modelo global e tantos modelos locais quantas as entidades participantes na estrutura produtiva (rede).

A modelação da informação, com vista ao desenvolvimento posterior do modelo de objectos do sistema, foi feita segundo a estrutura ilustrada na Figura 4.7 e que se apresenta de seguida.

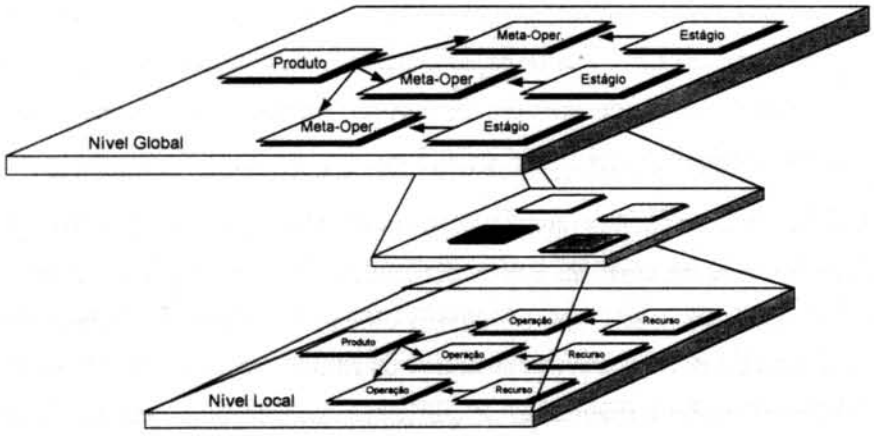


Figura 4.7 – Estruturação da informação

Considera-se que a produção de um determinado *produto* requer um determinado conjunto de *operações* a serem executadas num conjunto de *recursos*. As relações entre estas três entidades básicas (produto, operação, recurso) são do tipo "muitos-para-muitos".

Ao nível global, isto é, ao nível da *Empresa Virtual*, considera-se que a produção de um produto final requer um conjunto de operações, designadas por *meta-operações*, a realizar em cada um dos estágios produtivos que compõem a estrutura produtiva. Num mesmo estágio pode existir mais do que uma unidade produtiva. Assim, um

produto final tem sempre associado um conjunto de sub-produtos (produto local) em número igual ao número de estágios necessários para a sua produção.

Num determinado estágio e para cada unidade produtiva, a produção de um produto local (sub-produto) envolve um conjunto de operações a serem efectuadas num determinado conjunto de recursos. Também aqui é considerada uma relação de "muitos-para-muitos" entre operações, recursos e produtos locais.

A Figura 4.8 representa a forma como esta informação foi conceptualizada em termos de modelos de objectos.

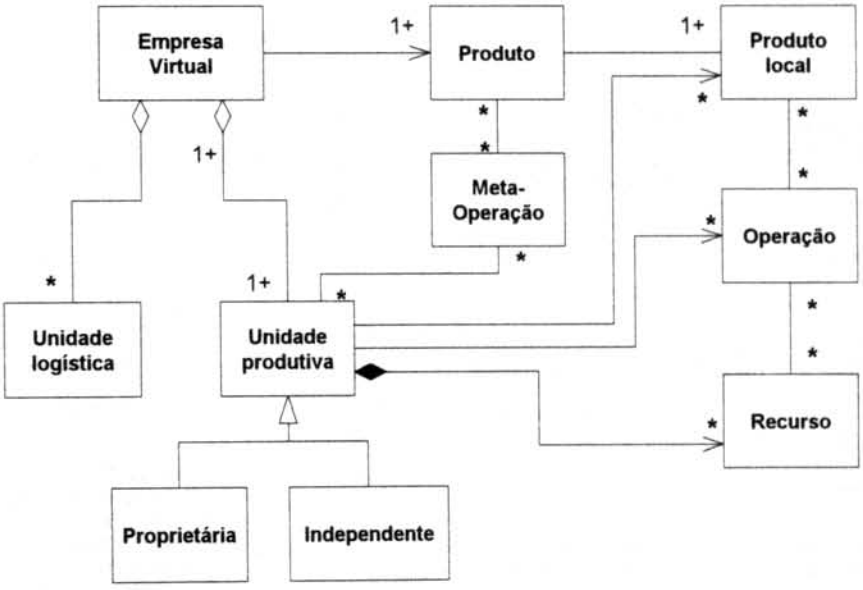


Figura 4.8 – Modelação da estruturação da informação

4.2.3.2 Diagrama de classes (global)

Por razões de facilidade de leitura, o diagrama de classes (modelo de objectos) é apresentado segundo os níveis, global e local, em que posteriormente foi implementado. Assim, a Figura 4.9 apresenta o diagrama de classes global, e a Figura 4.10 e a Figura 4.11 apresentam o diagrama de classes local.

O diagrama de classes é constituído por classes[20] do domínio na qual o sistema opera. Naturalmente que, neste contexto, se pretende apenas ilustrar o trabalho realizado, apresentando as suas linhas gerais e não entrando em questões de pormenor.

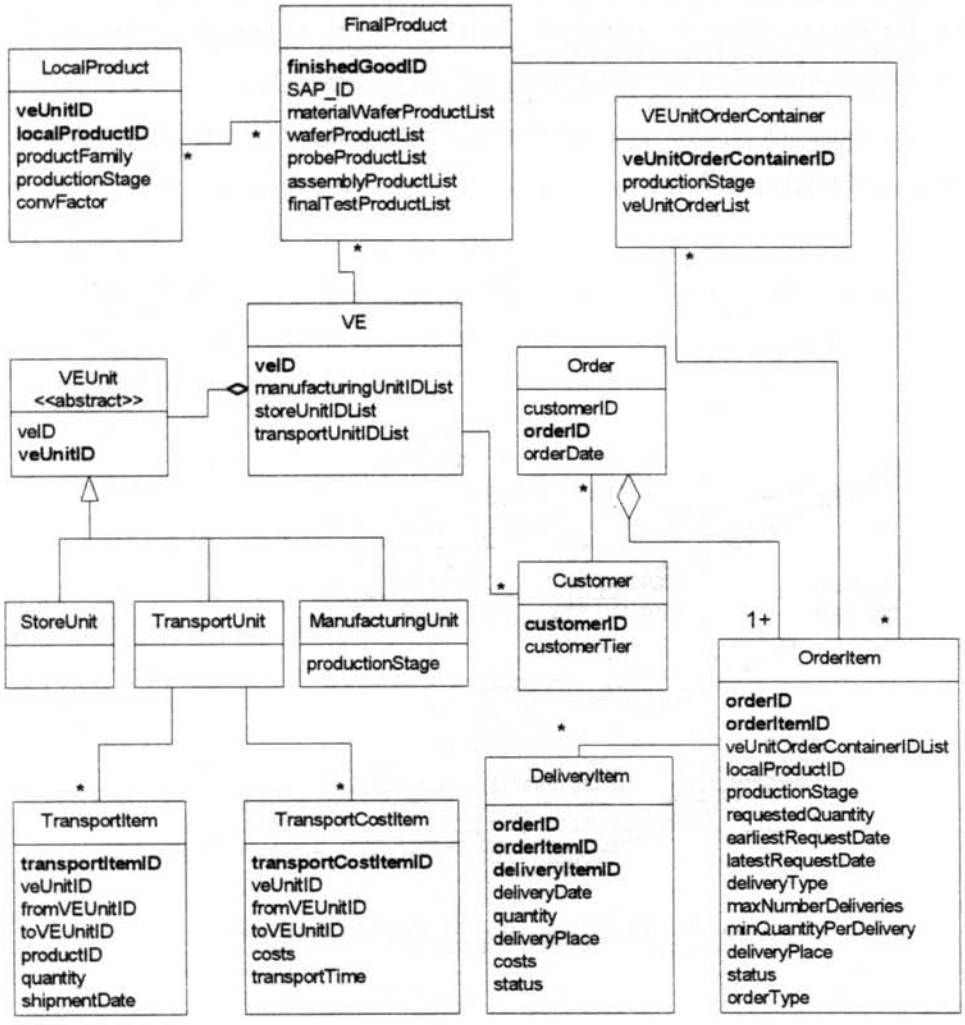


Figura 4.9 – Diagrama de classes (modelo de objectos) global

O diagrama de classes global (Figura 4.9) basicamente modela a *Empresa Virtual* (*EV*) como uma agregação de uma ou mais entidades (*VEUnit*) que podem ser: do

[20] Uma *classe* descreve um grupo de objectos com propriedades semelhantes (atributos), comportamento comum (operações), idêntico relacionamento com outros objectos e a mesma semântica.

tipo produtivo (*ManufacturingUnit*), do tipo de armazenamento (*StoreUnit*) e do tipo logístico (*TransportUnit*). São estabelecidas relações entre a *Empresa Virtual*, os clientes (*Customer*) e o conjunto de produtos finais disponíveis (*FinalProduct*). Por outro lado, para cada cliente são estabelecidas relações com encomendas (*Order*), constituídas por diversas linhas de encomenda (*OrderItem*).

Para cada linha de encomenda, são estabelecidas relações com a rota planeada para a sua execução (*VEUnitOrderContainer*) e com o conjunto de objectos representativo das quantidades planeadas (*DeliveryItem*).

4.2.3.3 Diagrama de classes (local)

O diagrama de classes *local* (Figura 4.10 e Figura 4.11) constitui a base de informação dos modelos que representam cada unidade produtiva existente na *Empresa Virtual*.

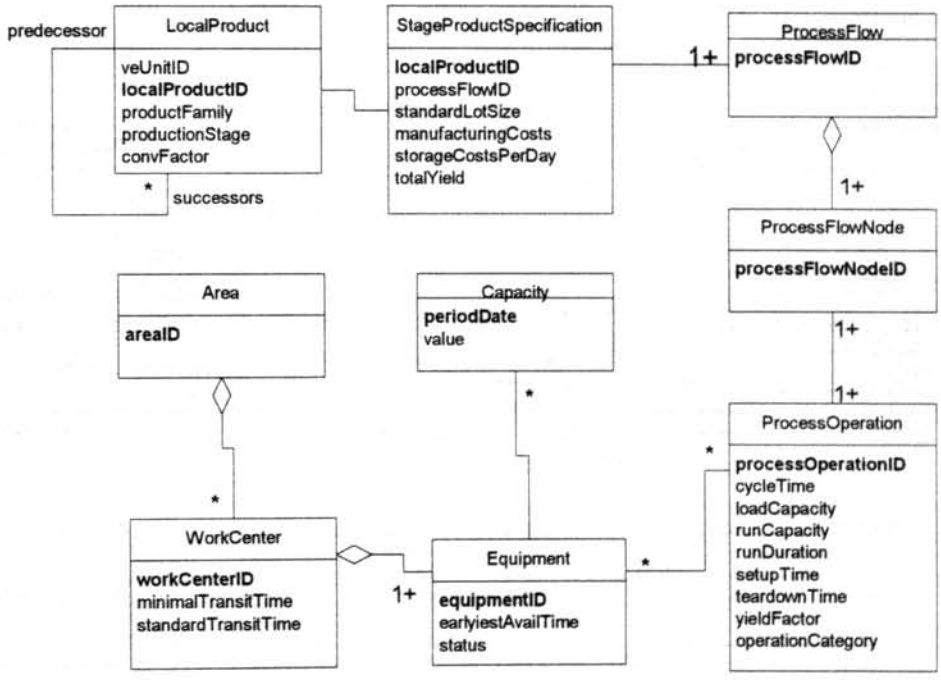


Figura 4.10 – Diagrama de classes *local* (1ª parte)

Para cada produto local (*LocalProduct*), são estabelecidas relações com uma especificação técnica do processo de fabrico (*StageProductSpecification*) que, entre outros

aspectos, define a rota produtiva (*ProcessFlow*) numa determinada unidade fabril. Além disso, o modelo considera igualmente os objectos instanciados como resultado do planeamento de encomendas, nomeadamente, encomendas locais (*VEUnitOrder*), lotes (*VEUnitLot*), etc.

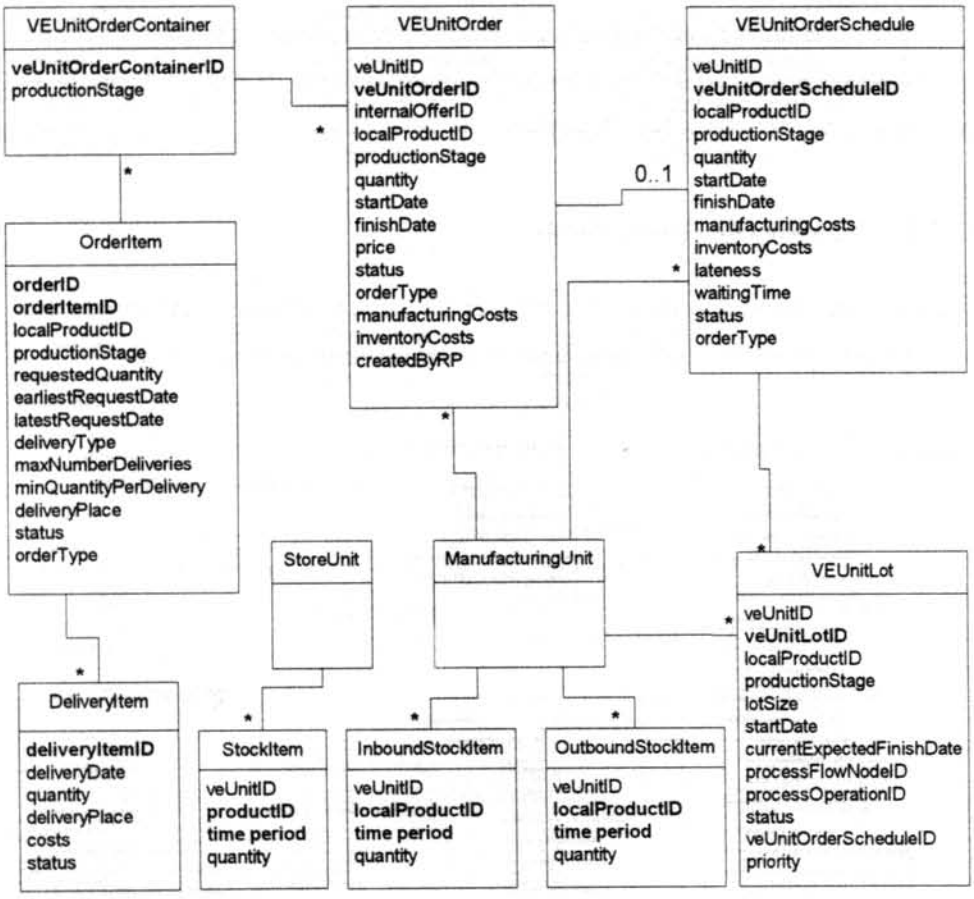


Figura 4.11 – Diagrama de classes *local* (2ª parte)

4.2.3.4 Objectos principais

A implementação do modelo de objectos é assegurada por dois componentes: o *Gestor de Informação Global* (GIM – *Global Information Manager*) e o *Gestor de Informação Local* (LIM – *Local Information Manager*). Estes componentes disponibilizam interfaces através das

quais os restantes componentes do sistema, locais e globais, acedem, criam, eliminam ou actualizam instâncias de objectos.

Apresenta-se na Tabela 4.1, a caracterização (descrição, âmbito e tipo de informação) dos objectos principais considerados nos diagramas de classes apresentados.

Classe	Descrição	Âmbito	Tipo
VE	define a *Empresa Virtual* e é uma agregação de uma ou mais entidades VEUnit	global	estática
VEUnit	define um nó genérico da *Empresa Virtual*	global	estática
FinalProduct	representa os produtos que podem ser encomendados pelos clientes	global	estática
LocalProduct	representa o sub-produto produzido em cada estágio produtivo	global e local	estática
Order	define as encomendas dos clientes	global	dinâmica
OrderItem	define uma linha de encomenda	global	dinâmica
DeliveryItem	define a entrega de uma quantidade planeada (total ou parcial) para satisfação de uma linha de encomenda	global	dinâmica
VEUnitOrderCont.	representa a rota de produção planeada	global	dinâmica
ProcessFlow	define a rota do processo de produção numa determinada unidade	local	estática
ProcessFlowNode	define sub-rotas de processo	local	estática
ProcessOperation	representa uma operação elementar	local	estática
Equipment	representa o recurso elementar onde são executadas uma ou mais operações de fabrico	local	estática
Capacity	para cada equipamento define, num determinado instante de tempo, a capacidade disponível	local	dinâmica
WorkCenter	representa um grupo de recursos funcionalmente idênticos	local	estática
VEUnitOrder	define as encomendas locais, planeadas a nível global	local	dinâmica
VEUnitLot	representa um lote de produtos, em curso de fabrico	local	dinâmica
VEUnitOrderSche.	representa objectos planeados numa fase de pós-optimização (planeamento fino)	local	dinâmica

Tabela 4.1 – Principais classes de objectos

Adicionalmente, aos diagramas de classes apresentados, cada componente de *software* implementa o seu próprio modelo de dados, de acordo com as suas especificidades.

4.2.4 Descrição funcional

O módulo de *Order Promise* permite, no processo de negociação e aceitação de encomendas, determinar datas de entrega para os produtos pedidos pelos clientes, de forma rápida e fiável e com base numa avaliação das capacidades produtivas da rede, no seu conjunto.

Em termos gerais, a actividade de *order promising* tem a finalidade de produzir uma "cotação", em resposta ao pedido de um cliente. Segundo (Higgins et al., 1996),

"*Order Promising stands for all activities related to the acceptance of an incoming customer order against a feasible date*".

A Ontologia da *Empresa Virtual*[21] (*Virtual Enterprise Ontology* (Soares et al., 1996)), desenvolvida no âmbito do projecto X-CITTIC e para a qual o autor contribuiu activamente, considera para *order promising* a seguinte definição:

ORDER PROMISING: an *ACTIVITY* whose *PURPOSE* is to produce a QUOTATION in response to an EXTERNAL CUSTOMER REQUEST.

QUOTATION: is a statement of price, description of PRODUCTS, DUE DATES, and other terms of sale, offered by a supplier to a prospective CUSTOMER.

DUE DATE: is the *CALENDAR DATE* when PRODUCTS are available for delivery.

O modelo funcional preliminar do módulo, apresentado na Figura 4.12, realça as suas funcionalidades principais:

- interacção com clientes na recepção de pedidos e na apresentação de cotações;
- gestão de informação de ATP (*available-to-promise*) para os produtos disponíveis na EV;

---

[21] No capítulo 6, é feita uma descrição sucinta do conteúdo e da organização da ontologia, encontrando-se um extracto em apêndice.

- planeamento agregado (*rough planning*) das encomendas, pela EV.

Basicamente este módulo do sistema X-CITTIC (que será neste trabalho designado como *Sistema Order Promise*) interactua com os clientes, determinando em tempo-real a possibilidade de satisfazer os seus pedidos, e originando um conjunto de informação (encomendas, planos globais) a ser posteriormente processada por um sistema de optimização global (planeamento fino).

Com a confirmação da encomenda, por parte do cliente, o sistema procede à actualização das capacidades em cada uma das entidades produtivas envolvidas, por forma a assegurar que pedidos futuros sejam avaliados de forma correcta. Além disso, determina a rota produtiva para a encomenda planeada, isto é, são geradas sub-encomendas para as entidades envolvidas na satisfação da encomenda.

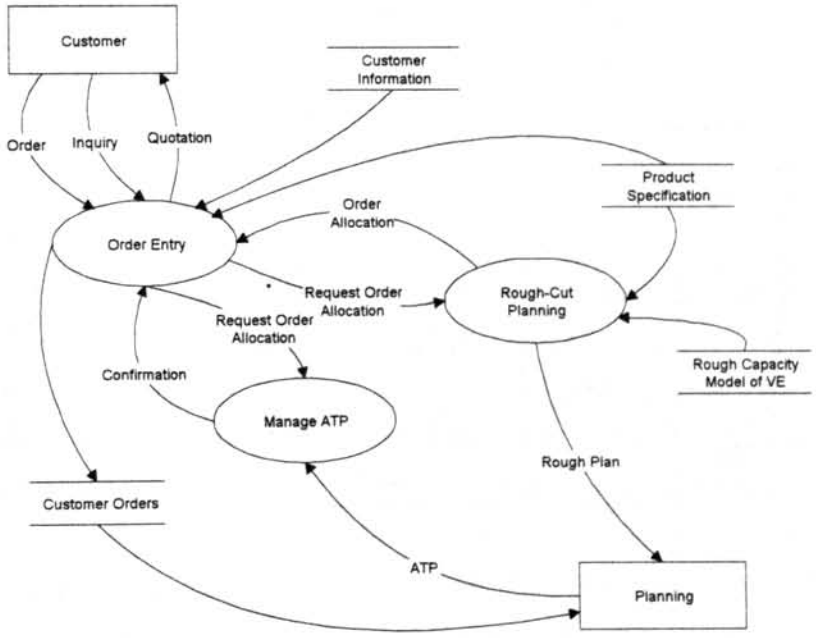


Figura 4.12 – Modelo funcional preliminar do módulo *Order Promise*

Como referido anteriormente (ver secção 2.4) e como resultado da experiência prática, foram identificados dois cenários principais, aquando do processamento do pedido de um cliente:

1. ou o pedido é satisfeito integralmente a partir da informação de ATP;
2. ou é desencadeado imediatamente o planeamento da produção do pedido, por insuficiência de quantidade disponível para venda, na data pedida.

O diagrama de estados[22] da Figura 4.13 ilustra, com um nível de abstracção elevado, os estados possíveis para o módulo, de acordo com os cenários apresentados.

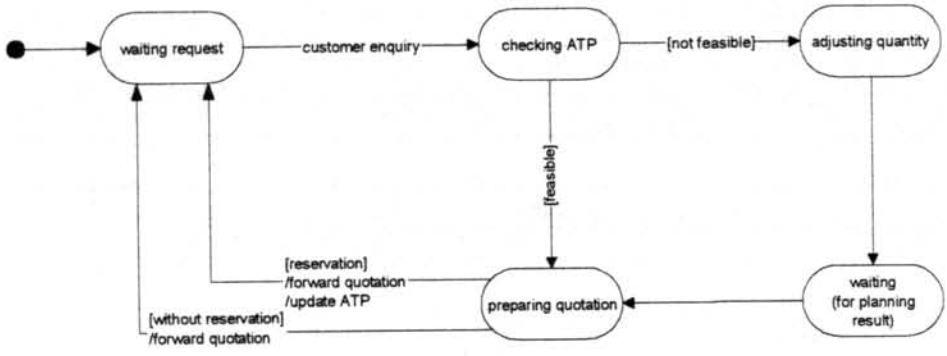


Figura 4.13 – Diagrama de estados do módulo *Order Promise*

4.2.5 Estrutura e arquitectura geral do sistema *Order Promise*

O requisito de ser capaz de receber pedidos em simultâneo de clientes distribuídos geograficamente, por um lado, e a necessidade de se conseguir o planeamento em tempo-real e envolvendo todas as entidades da *Empresa Virtual*, por outro, conduziram à estruturação global do sistema segundo três níveis principais (Figura 4.14):

1. *order taking* – este nível é responsável pela interacção com os clientes;
2. *order processing* – este nível processa o pedido do cliente, começando por verificar se existe disponível a quantidade requerida na data pedida; se não, acciona o processo de planeamento agregado; assegura a coordenação das

[22] Um *diagrama de estados* é um grafo cujo nós são *estados* de um sistema e cujos arcos dirigidos representam transições de estado, provocadas por acontecimentos (*eventos*). O diagrama especifica uma sequência de estados causada por uma sequência de eventos. Isto é, um diagrama de estados permite descrever o comportamento dinâmico do sistema.

entidades envolvidas na satisfação da encomenda e actua como mediador na fase de negociação (planeamento);

3. *capacity checking* – este nível envolve um conjunto de entidades que funcionam como agentes locais de planeamento (modelos locais de capacidade), associadas às diferentes unidades da rede.

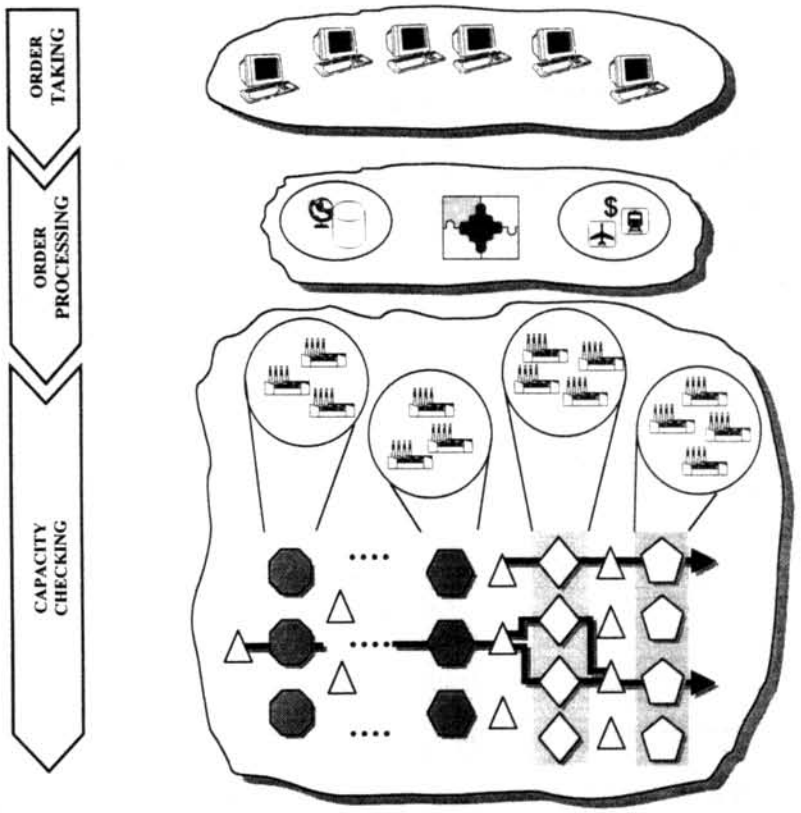


Figura 4.14 – Estrutura geral do módulo *Order Promise*

Refira-se que o desenvolvimento do sistema seguiu uma *abordagem iterativa e incremental* (Jacobson et al., 1999), permitindo uma crescente compreensão do problema, através de refinamentos e melhoramentos sucessivos. Assim, no projecto, diversos componentes foram desenvolvidos seguindo um ciclo contínuo de análise, "redese-

nho" e melhoria. Procurou-se igualmente descrever o sistema recorrendo a diagramas de *casos de uso*[23], seguindo a notação UML.

A Figura 4.15 apresenta um *caso de uso* para o sistema *Order Promise*. Identificam-se três tipos de *actores*: o cliente (externo ou interno à empresa), o "vendedor" (*sales person*) que "trata" o pedido do cliente e os sistemas de informação externos ao sistema (sistemas legados, tais como ERP e MES).

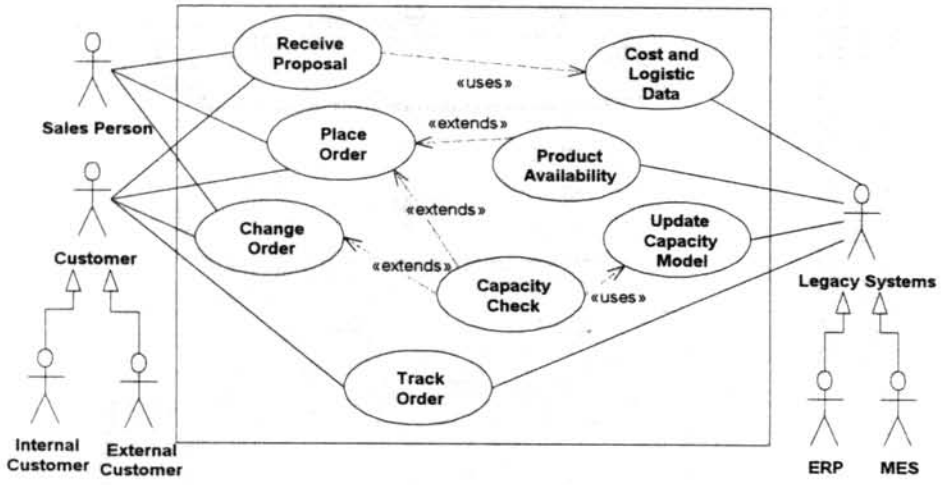


Figura 4.15 – Diagrama de *caso de uso* para o sistema *Order Promise*

Os subsistemas do Order Promise

O sistema *Order Promise* foi decomposto num conjunto de subsistemas, sendo cada subsistema, um agrupamento de diferentes componentes. Segundo a hierarquização atrás referida, consideraram-se três subsistemas de base: *Order Taker*, *Order Planner* e *Capacity Checker*. Cada um destes subsistemas acede à informação através de outros dois subsistemas (GIM – *Global Information Manager* e LIM – *Local Information Manager*).

---

[23] Os diagramas de casos de uso (*use case diagrams*) representam as interacções entre utilizadores (*actores*) e um sistema computacional, procurando captar, na perspectiva do utilizador, as funcionalidades do sistema. Os actores representam as entidades externas ao sistema, isto é, o seu contexto (Booch et al., 1999).

A Figura 4.16 apresenta graficamente, de acordo com a notação UML, o sistema *Order Promise* permitindo ter, com um nível de abstracção elevado, uma visão estática do sistema.

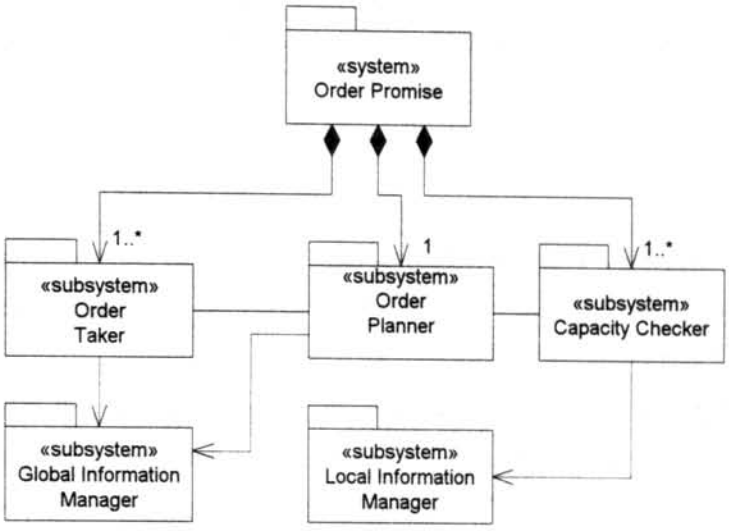


Figura 4.16 – Representação gráfica (vista estática) do sistema *Order Promise*

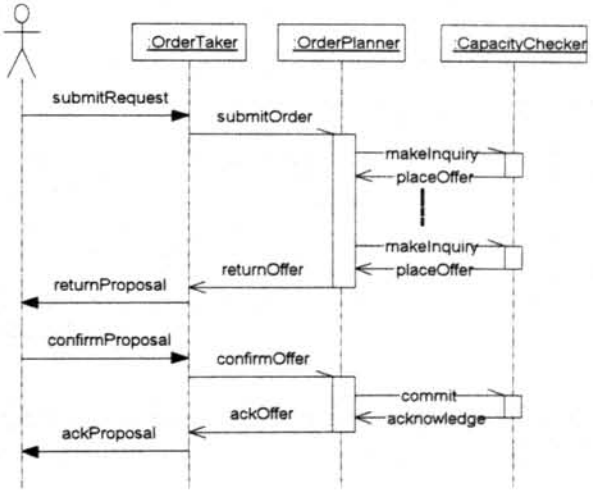


Figura 4.17 – Diagrama de interacção (vista dinâmica) do sistema *Order Promise*

Por outro lado, o diagrama de interacção, apresentado na Figura 4.17, permite, igualmente com um nível de abstracção elevado, ter uma visão dinâmica do sistema. Na verdade, o diagrama ilustra o processo iterativo da fase de negociação entre o coordenador global (*Order Planner*) e os modelos locais de capacidade (*Capacity Checkers*) representativos de cada unidade produtiva envolvida no planeamento de uma determinada encomenda.

Arquitectura geral

O sistema *Order Promise* envolve diversos componentes distintos, interactuando de forma integrada, e agrupados em subsistemas de acordo com a representação gráfica apresentada anteriormente na Figura 4.16.

A arquitectura geral do sistema, com a definição dos principais fluxos de informação entre os componentes, encontra-se representada na Figura 4.18. A comunicação entre componentes é feita através de uma infra-estrutura compatível com a especificação CORBA.

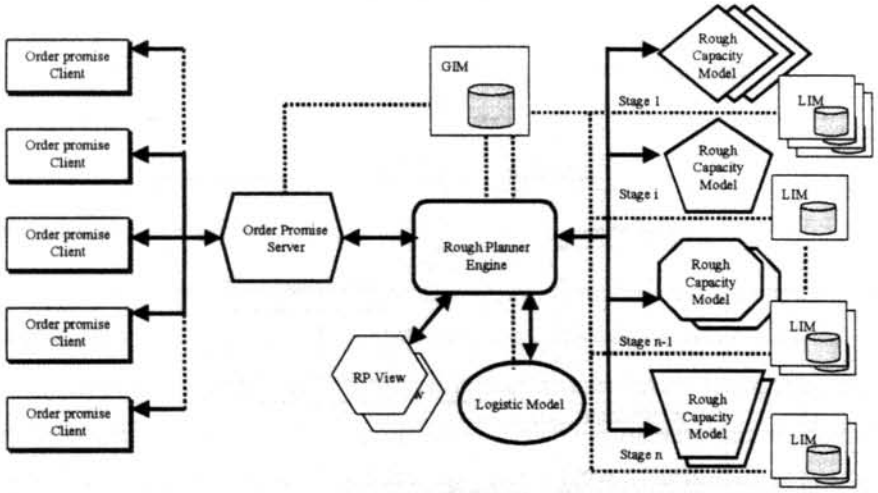


Figura 4.18 – Arquitectura global do sistema *Order Promise*

Os componentes considerados (dos quais alguns podem ter várias "instâncias") são os seguintes:

1. *Order Promise Client* (OPClient) – é responsável pela interacção com o utilizador, permite submeter ao sistema, um pedido de um cliente e visualizar as diferentes ofertas produzidas pelo sistema;
2. *Order Promise Server* (OPServer) – gere os pedidos submetidos pelos componentes OPClient e sequencia esses pedidos, de acordo com diferentes regras (FIFO, nível de prioridade, data pedida, etc);
3. *Rough Planner Engine* (RP) – constitui o motor de planeamento baseado em negociação; gera propostas, a submeter aos modelos de capacidade locais, existentes em cada nó da estrutura produtiva (rede); gere as ofertas recebidas dos modelos locais de capacidade; avalia as alternativas possíveis de produção;
4. *Rough Capacity Model* (RCM) – modelam cada entidade da estrutura produtiva (sob o ponto de vista de gestão de capacidades), avaliam as propostas geradas pelo RP e geram as correspondentes ofertas; são instanciados tantas vezes quantas as unidades produtivas existentes;
5. *Rough Planner View* (RPView) – permite a monitorização em tempo-real do processo de negociação entre o RP e cada um dos RCM; permite igualmente a parametrização do algoritmo de planeamento do RP e a visualização da configuração actual da *Empresa Virtual*;
6. *Logistic Model* (LM) – disponibiliza informação logística, nomeadamente, tempos, custos e disponibilidades de transporte, custos de armazenamento, etc;
7. *Global Information Manager* (GIM) – é responsável pela manutenção e gestão do modelo de objectos global;
8. *Local Information Manager* (LIM) – é responsável pela manutenção e gestão do modelo de objectos local; é instanciado tantas vezes, quantas o número de unidades produtivas existentes na *Empresa Virtual*.

Os componentes de gestão de informação, GIM e LIM, asseguram não só a distribuição da informação como permitem integrar os diferentes componentes[24]. Os

[24] De acordo com a linguagem de modelação gráfica UML, um componente é representado de forma icónica por um rectângulo grande e por dois rectângulos pequenos colocados sobre o lado esquerdo. As interfaces existentes no componente são representadas por pequenos círculos ligados a qualquer um dos lados, esquerdo ou direito.

componentes globais acedem ao GIM e os componentes locais ao LIM respectivo (Figura 4.19).

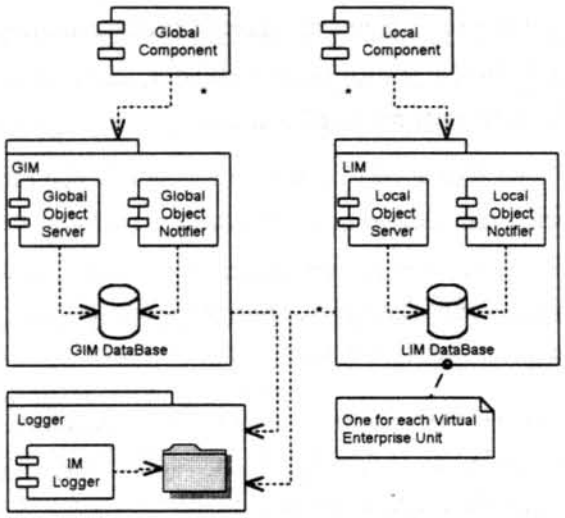


Figura 4.19 – Interacção com os componentes de gestão de informação

A interacção entre os componentes é feita através de três tipos de interface: uma *estática* (*static view*), uma *dinâmica* (*action view*) e outra por *notificação* (*notification view*) (Figura 4.20).

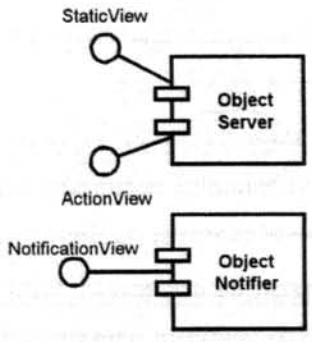


Figura 4.20 – Tipos de interfaces nos componentes de gestão de informação

Interface estática (Static View)

Esta interface, usada essencialmente para obtenção de informação estática, implementa um mecanismo típico de pedido/resposta de informação. Um componente faz, ao gestor de informação, um pedido de uma classe de objectos e obtém como resposta uma sequência de informação (objectos).

Interface dinâmica (Action View)

Através desta interface, os componentes actuam no modelo de objectos podendo instanciar, alterar ou remover objectos. Esta interface é tipicamente usada pelos componentes, no caso da gestão de informação tipo dinâmica.

Interface por notificação (Notification View)

Esta interface permite que um componente registe o seu "interesse" por uma determinada classe de objectos. Sempre que se verificam alterações, instanciações ou remoções de objectos, os componentes registados são notificados automaticamente, estando por isso sempre actualizados. Por exemplo, se for criada uma nova encomenda no GIM por um determinado componente, todos aqueles que necessitem de ter a informação acerca da nova encomenda recebem automaticamente o objecto respectivo.

Subjacente a cada acção decorrente das três interfaces descritas, nos componentes responsáveis pela gestão de informação, o GIM e cada um dos LIM, verifica-se uma interacção com a base de dados relacional, nomeadamente através de comandos SQL tipo *select*, *insert*, *delete*.

Diagrama de componentes

Para além dos componentes já descritos, o sistema *Order Promise* interactua com outros componentes integrados noutros módulos do sistema X-CITTIC, nomeadamente, o módulo de Controlo Reactivo. Qualquer um dos componentes, independentemente da sua organização lógica em subsistemas ou módulos, pode residir em qualquer ponto da rede de informação. A sua localização é feita, de forma transpa-

rente, através da infra-estrutura CORBA. A Figura 4.21 apresenta o diagrama de todos os componentes que constituem e/ou interactuam com o sistema *Order Promise*.

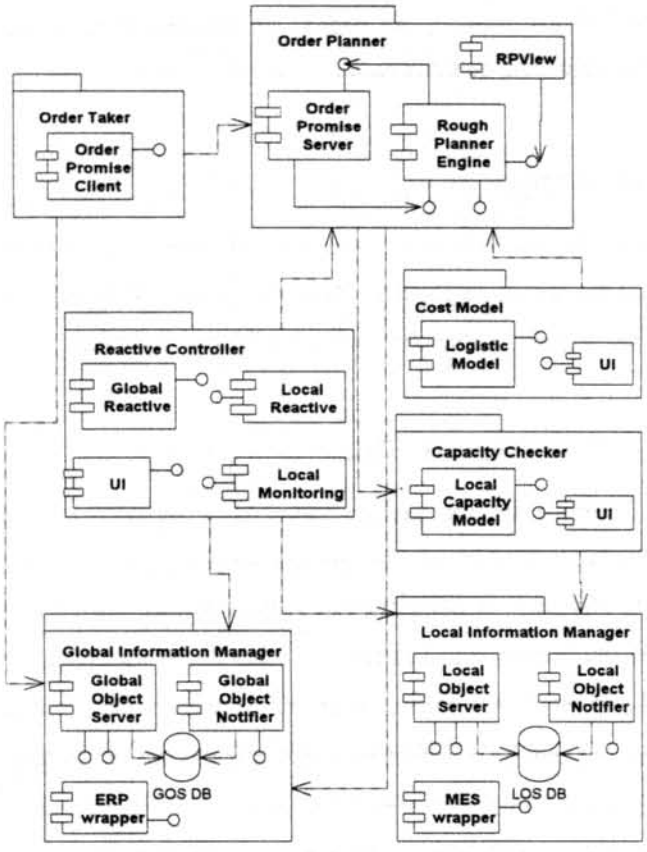


Figura 4.21 – Diagrama de componentes

Configuração do sistema

A organização lógica dos componentes nos módulos ou subsistemas apresentados na Figura 4.21 não significa que, ao nível de execução, quaisquer componentes precisem de se localizar no mesmo sistema computacional. Assim, a Figura 4.22 ilustra um cenário possível de execução, em que os componentes se encontram distribuídos por diferentes sistemas computacionais, geograficamente distribuídos. O caso particular apresentado diz respeito a uma situação real de funcionamento do sistema *Order Promise* numa empresa industrial de semicondutores, constituída por diferentes unidades produtivas (oito na configuração apresentada e distribuídas pelos quatro estágios principais do processo).

Existem basicamente seis *nós servidores*, cada um com diversos componentes, podendo existir tantos *componentes clientes* (tipo OPClient e RPView) quantos se queiram.

No cenário apresentado, optou-se por colocar no sistema computacional afecto a uma determinada unidade local (por exemplo *Wafer Fab Server*) todos os componentes de âmbito local e respeitantes a essa unidade, nomeadamente, o LOS (*Local Object Server*), o LON (*Local Object Notifier*), o LRC (*Local Reactive Controller*) e o RCM (*Rough Capacity Model*) respectivo. Os componentes de âmbito local foram distribuídos por dois sistemas computacionais principais (:globalPlanner e :global).

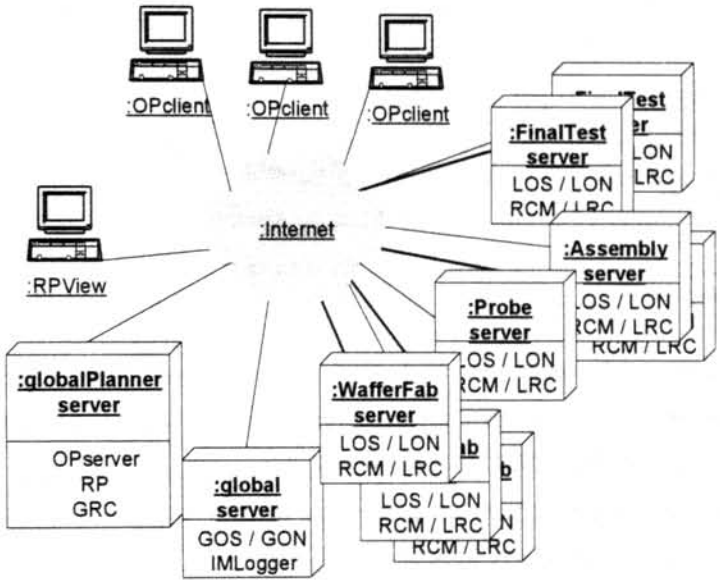


Figura 4.22 – Cenário de configuração de funcionamento do sistema *Order Promise*

4.2.6 *Order Promise Client* (OPClient)

4.2.6.1 Descrição funcional

Este componente é responsável pela interface com o utilizador, permitindo que este submeta *pedidos*, oriundos de clientes, e receba *ofertas*, geradas pelo componente de planeamento RP (*Rough Planner*).

Este componente foi desenvolvido de forma a permitir que um dado utilizador possa submeter diversos pedidos em simultâneo, para o mesmo cliente (por exemplo, na situação de uma encomenda ser constituída por diversas linhas de encomenda com produtos eventualmente diferentes), ou para clientes diferentes. Além disso, o componente foi desenvolvido por forma a poder ser instanciado tantas vezes quantas as necessárias, isto é, por forma a permitir a simultaneidade de utilizadores distribuídos geograficamente.

4.2.6.2 Interfaces

O componente interactua com outros dois componentes do sistema: o OPServer e o GIM. Para garantir essa interacção, foram especificadas duas interfaces IDL[25] assíncronas, através das quais é possível obter informação estática de clientes e de produtos finais (interface estática com o gestor de informação global GIM), e efectuar a submissão de pedidos e recepção de ofertas (interface com o OPServer). As interacções entre estes componentes estão representadas na Figura 4.23.

No Apêndice C são apresentados extractos das principais interfaces IDL definidas e implementadas nos componentes respectivos. O componente OPClient funciona, quer como componente cliente (invocação de métodos noutros componentes) quer como componente servidor, consoante o sentido da interacção de comunicação. Procurou-se desta forma tirar partido do maior desempenho de uma interface assíncrona.

---

[25] *Interface Definition Language* (IDL) – é uma linguagem neutra puramente declarativa (não especificando nada em relação à implementação) e independente da linguagem de programação e do sistema operativo. A IDL é usada para definir as *interfaces* de objectos CORBA por forma a garantir a interoperabilidade desejada entre eles.

Basicamente, o OPCLient interactua com o OPServer quando se inicia o processo de submissão de um novo pedido (potencial encomenda) de um cliente. Nesta situação, com a invocação do método *CreateNewOrder* procede-se à criação no GIM de um novo objecto (*Order*). Posteriormente, com a definição de uma linha de encomenda e com a correspondente invocação do método *CreateNewOrderItem* é instanciado no GIM, para o pedido em curso, o objecto *orderItem*. Após a definição de uma linha de encomenda, nomeadamente através da definição do tipo de produto, quantidade e data de entrega pretendida, encontram-se reunidas as condições para se iniciar o processo de planeamento associado. Assim, nesta situação, a invocação do método *RequestDeliveryItem*, no componente OPServer, dará origem a um pedido de planeamento (inserção da encomenda no plano actual) através da invocação do método *RoughPlanningRequest* no componente de planeamento *Rough Planner* (RP).

Após a recepção da avaliação, efectuada pelo RP, ao pedido submetido, a invocação dos métodos *RequestOrderItemConfirmation* ou *RequestOrderItemRejection* permitem, respectivamente, aceitar ou rejeitar os resultados gerados pelo componente de planeamento RP e difundidos através da invocação do método *RoughPlanningRequest*.

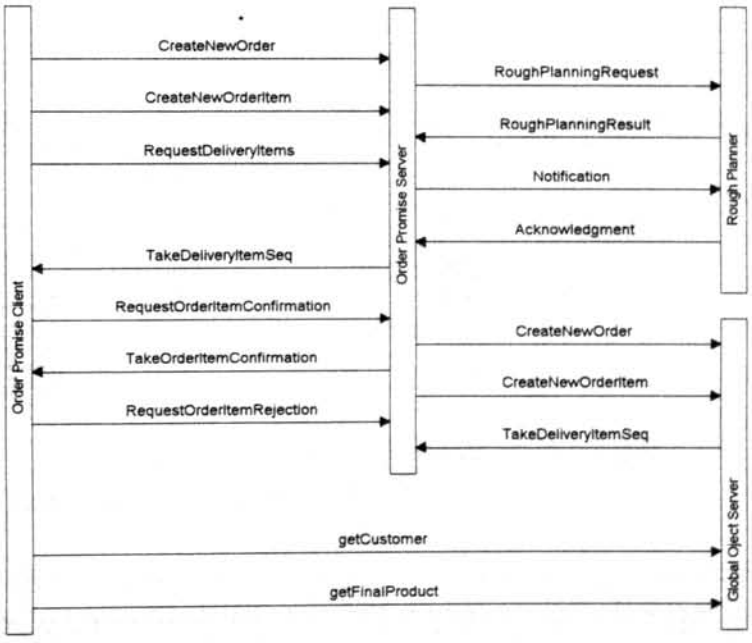


Figura 4.23 – Interacção entre componentes

4.2.6.3 Diagrama de classes

O diagrama de classes e o diagrama de estados estão representados respectivamente na Figura 4.24 e na Figura 4.25.

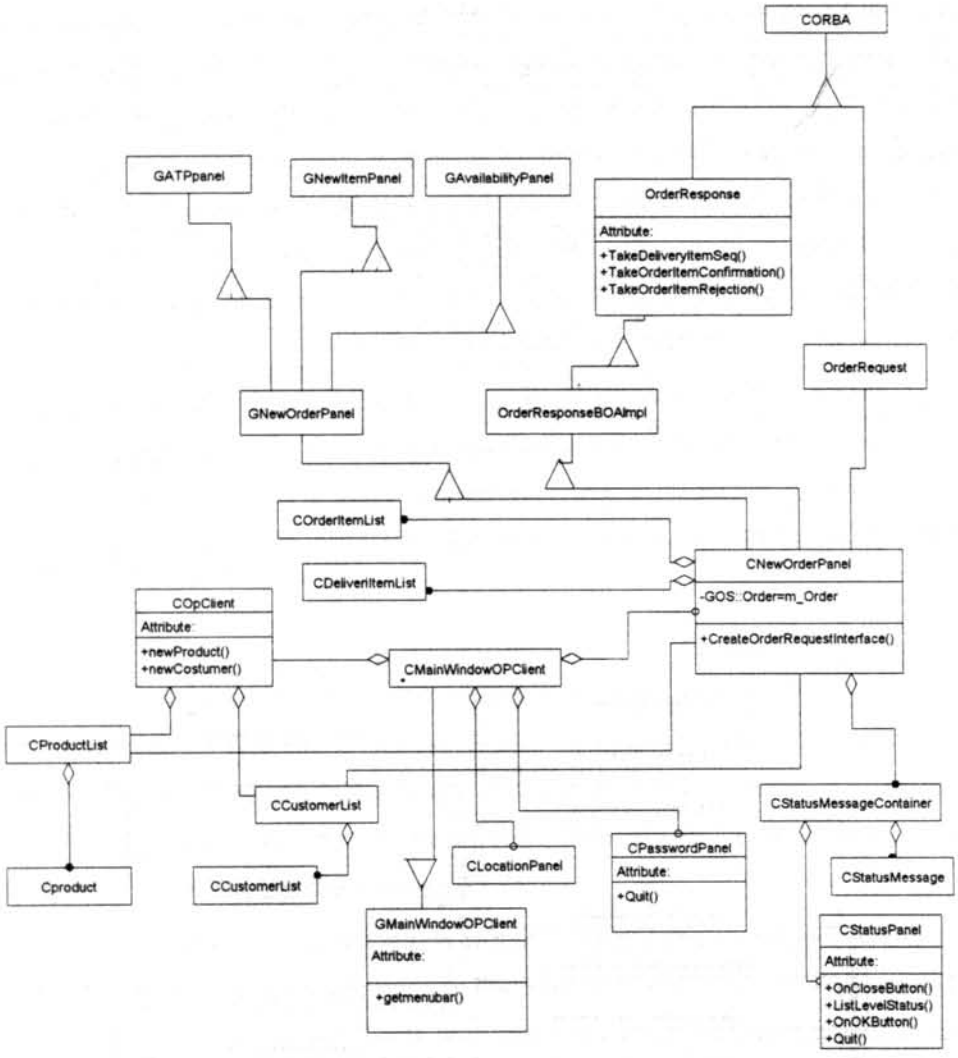


Figura 4.24 – Diagrama de classes do componente OPClient

No diagrama da Figura 4.24 estão fundamentalmente representadas as classes que suportam a interface com o utilizador, através de janelas de diálogo interactivas.

As classes **OrderResponse** e **OrderRequest** implementam as interfaces com os componentes externos.

No diagrama de estados do componente (Figura 4.25), identificam-se dois estados fundamentais: o estado de inicialização (*init*) e o estado de repouso (*idle*). No estado *init*, o componente estabelece uma conexão com o servidor de objectos global (GIM), com a finalidade de obter informação relativa ao conjunto de clientes existentes e às referências dos produtos finais disponíveis para venda. Esta informação é obtida através da interface estática (*static view*) do GIM.

Após a fase de inicialização, o componente passa para o estado de repouso (*idle*), em que aguarda a chegada de novos pedidos (*new inquire*) ou a confirmação ou rejeição de qualquer oferta anterior, pendente. A chegada de um novo pedido faz com que o componente passe para um estado de espera (*waiting*), aguardando uma resposta do componente de planeamento RP (esta resposta chega via um outro componente, o OPServer).

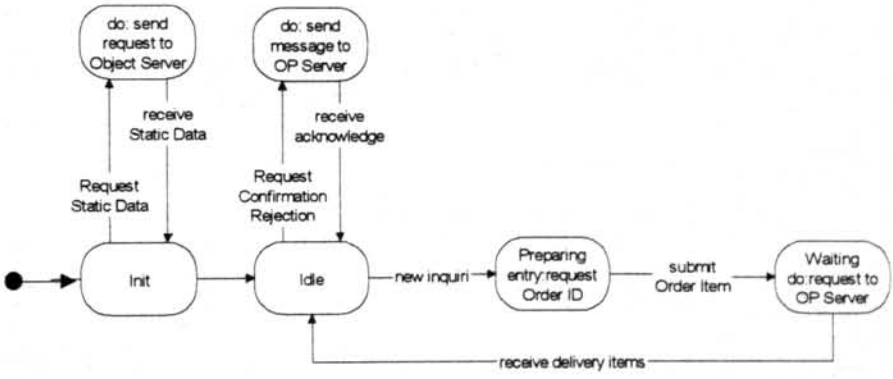


Figura 4.25 – Diagrama de estados do componente OPClient

4.2.6.4 Interacção com o utilizador

Com a finalidade de garantir a portabilidade dos componentes responsáveis pela interacção com o utilizador, entre diferentes plataformas computacionais, nomeadamente, diferentes sistemas operativos e diferentes sistemas gráficos de gestão de janelas, recorreu-se

à utilização de uma ferramenta de concepção de interfaces (ILOG Views). Conseguiu-se, além disso, uma interface com *um "look-and-feel"*, nomeadamente, Windows 95/98 e X Windows Systems, seleccionável pelo utilizador.

Procurou-se que o diálogo entre o utilizador e o SAD fosse tão amigável e intuitivo quanto possível. Assim, o utilizador pode interagir por meio de menus, de comandos ou teclas de função, e do rato.

No caso do componente OPClient, a única acção de configuração necessária consiste na indicação do nome ou endereço IP dos componentes com os quais interactua, nomeadamente o GIM e o OPServer. A Figura 4.26 ilustra a operação de configuração do componente, visualizando-se em tempo-real o estado da interligação.

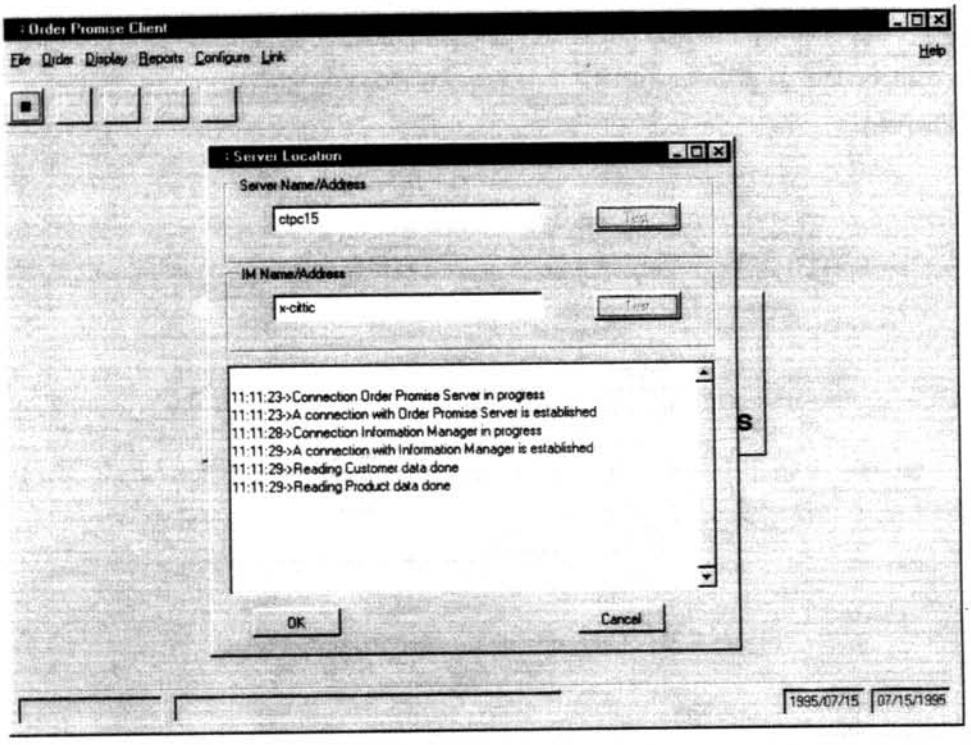


Figura 4.26 – Écran de configuração do OPClient

Exemplo de utilização

Descreve-se de seguida, uma sessão de trabalho típica de utilização do componente. Não se pretende naturalmente ser exaustivo, mas apenas ilustrar algumas das suas características principais.

A submissão de um novo pedido (potencial encomenda) de um cliente e a posterior visualização do seu planeamento são feitas através da mesma janela de interacção (ver Figura 4.27). A figura ilustra um cenário em que um pedido, com referência O#6 (submetido em 27/10/1998) é constituído por três linhas de encomenda. Na altura em que o pedido é "criado" o utilizador tem a possibilidade de impor um prazo de validade para o mesmo, isto é, o sistema permite que pedidos submetidos e avaliados, mas ainda não confirmados pelo cliente, sejam reservados.

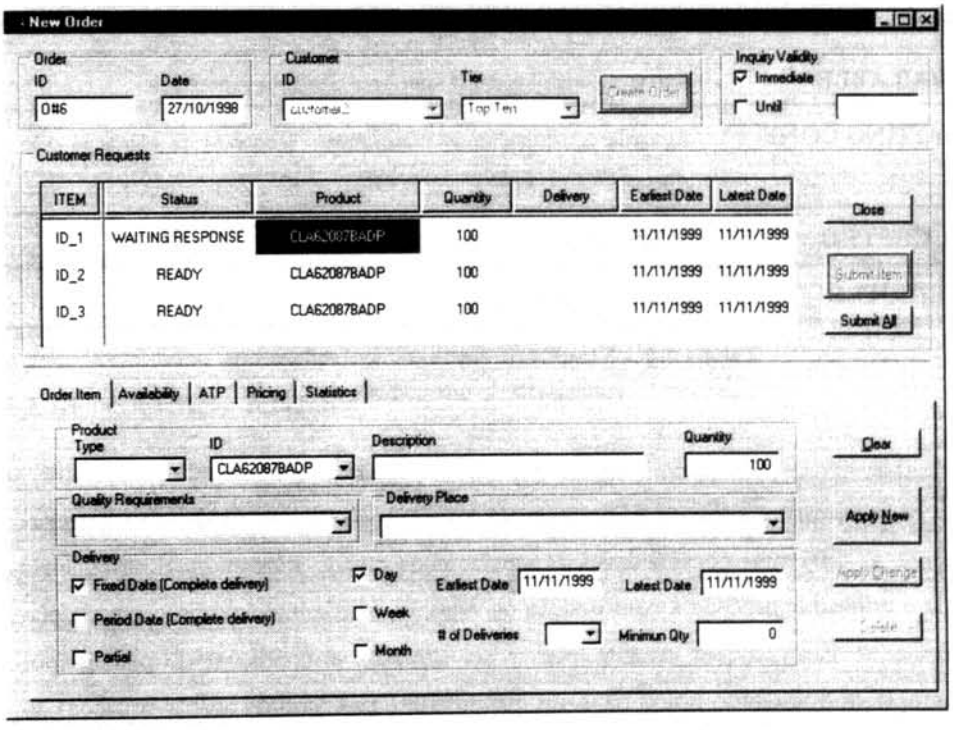


Figura 4.27 – Exemplo de entrada de pedidos

Para cada uma das linhas de encomenda, o utilizador selecciona o produto pedido, a quantidade total pedida e a forma desejada para a data de entrega: entrega numa data fixa da quantidade total pedida; entrega num dado intervalo de tempo, da

quantidade total; ou entregas parciais, de acordo com um número fixado de entregas e quantidades determinadas. Em qualquer instante, é possível avaliar imediatamente a possibilidade de satisfação de um pedido individual (linha de encomenda) ou do conjunto de pedidos já constituídos.

Durante o processo de avaliação, e para cada linha de encomenda, o estado da negociação é apresentado em tempo-real na coluna *Status*. Os estados possíveis são os apresentados na Tabela 4.2.

Estado	Descrição
READY	Pedido apto a ser submetido ao componente de planeamento
SUBMITED	Pedido submetido ao componente de planeamento via OPServer
WAITING RESP.	Aguarda resposta do planeamento
AVAILABLE	Existe a possibilidade de satisfazer o pedido submetido
WAITING CONF.	Aguarda confirmação do planeamento (aquando da aceitação pelo cliente)
ACCEPTED	Pedido aceite (encomenda firme)
REJECTED	Impossibilidade de satisfação do pedido submetido

Tabela 4.2 – Estados do processo de avaliação da submissão de um pedido

Finalmente, a Figura 4.28 apresenta, para o mesmo cenário descrito anteriormente, as diferentes possibilidades de satisfação do pedido submetido. É nesta janela que o utilizador procede à confirmação ou rejeição do resultado do processo de planeamento. Em qualquer instante, podem ser alteradas as condições do pedido (por exemplo, introduzindo novas datas ou quantidades), por forma a avaliar situações alternativas.

Note-se que o utilizador pode manter, em simultâneo, diferentes sessões de submissão de pedidos, eventualmente de clientes diferentes.

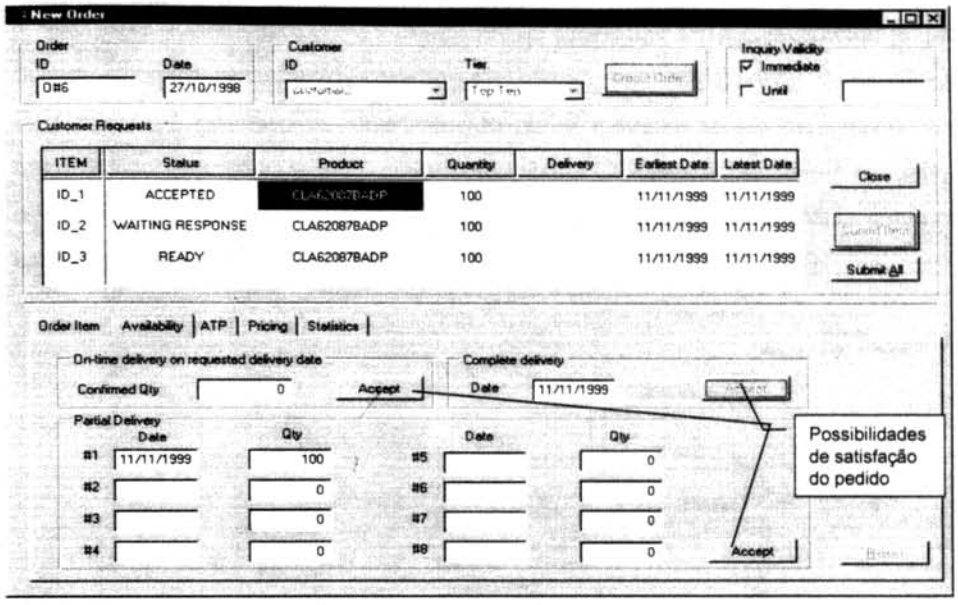


Figura 4.28 – Exemplo de oferta

4.2.7 *Order Promise Server*

4.2.7.1 Descrição funcional

Este componente é responsável pela gestão dos pedidos submetidos pelos diversos componentes OPClient. Basicamente, o componente sequencia os pedidos, de acordo com uma estratégia definida pelo utilizador e, para cada pedido, verifica se é necessário activar o componente de planeamento (RP).

A activação do componente RP acontece sempre que não existe quantidade ATP para o intervalo de tempo definido no pedido em processamento.

O diagrama de interacção apresentado na Figura 4.29 ilustra a sequência de eventos que ocorre em duas situações frequentes na prática: a submissão de um pedido através do componente OPClient e a confirmação da oferta gerada ao pedido submetido.

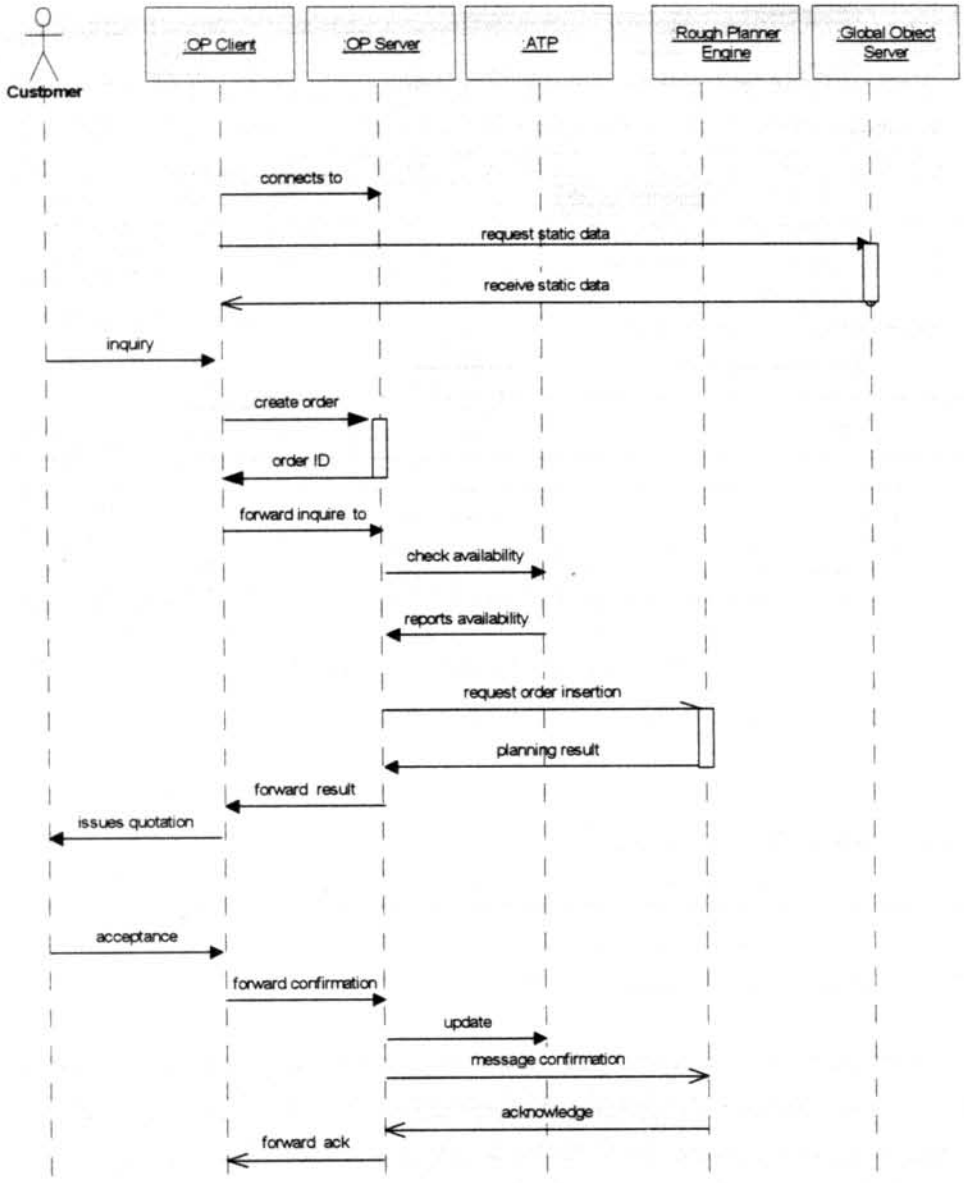


Figura 4.29 – Diagrama de interacção entre o OPServer e outros componentes

O OPServer interactua também com outros componentes do sistema X-CITTIC, nomeadamente, com o componente global de controlo reactivo e de monitorização (GRC – *Global Reactive Controller*) e com um componente, denominado *DemandPlan*, responsável pela interface ao sistema ERP (no caso presente SAP R/3).

A interacção com o GRC é no sentido de se efectuar um replaneamento global, quando ocorrem situações inesperadas na execução dos planos actuais de produção. Basicamente, o processo de replaneamento global é similar ao planeamento de um novo pedido submetido através de um componente OPClient. O diagrama apresentado na Figura 4.30 ilustra o protocolo de mensagens, segundo a sequência temporal habitual entre o OPServer e o GRC.

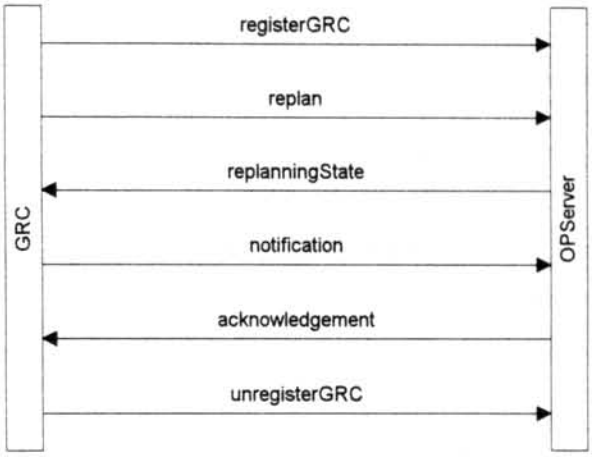


Figura 4.30 – Interacção entre o OPServer e o GRC

Mensagem	Descrição
registerGRC	permite o registo do componente no OPserver (o GRC envia ao OP-Server o seu identificador)
replan	pedido de replaneamento global de uma encomenda já anteriormente planeada
replanningState	resultado do replaneamento (sucesso ou insucesso) efectuado pelo RP
notification	o GRC aceita ou rejeita o resultado do planeamento
acknowledgement	confirmação de que o replaneamento foi efectuado com sucesso
unregisterRCM	termina a ligação com o OPServer

Tabela 4.3 – Mensagens entre o módulo de *Order Promise* e o módulo de *Monitorização e Controlo Reactivo*

A Tabela 4.3 apresenta a descrição das mensagens possíveis entre os dois componentes.

No OPServer, a interface com o utilizador serve essencialmente para monitorização, possibilitando a visualização de todos os pedidos pendentes, assim como o instante em que foram submetidos. Existe igualmente a possibilidade de se visualizar as interacções com o componente de planeamento RP, assim como o estado das ligações entre componentes, através de um sinóptico animado.

4.2.7.2 Interfaces

No OPServer, foram definidas e implementadas interfaces com os seguintes componentes: OPClient, RP, GRC, GIM e DemandPlan. No caso do OPClient e do DemandPlan, estas interfaces são instanciadas tantas vezes quantas as necessárias, sendo, nos restantes casos, instanciadas uma única vez.

A título ilustrativo, são apresentadas no Apêndice C, extractos das principais interfaces IDL desenvolvidas no componente.

A interface ao sistema ERP, via o componente DemandPlan, permite que o sistema Order Promise seja inicializado com a informação relativa a encomendas firmes já existentes. Garante-se desta forma que os planeamentos subsequentes sejam feitos em condições de "carga" realista. A interface IDL correspondente é apresentada igualmente no Apêndice C.

4.2.7.3 Diagrama de classes

O diagrama de classes e o diagrama de estados estão representados respectivamente na Figura 4.31 e na Figura 4.32.

A classe fundamental do componente é **CopServer**. Esta classe agrega classes secundárias associadas à interface com o utilizador, nomeadamente a classe **CMainWindowOPServer**, e classes responsáveis pela manutenção interna da informação estática. A classe **OpClient** é instanciada neste componente tantas vezes quantos os componentes clientes instanciados, isto é, são criadas interfaces independentes para cada um dos componentes externos ao OPServer. Consegue-se, desta forma, garantir a existência de um ambiente multi-cliente.

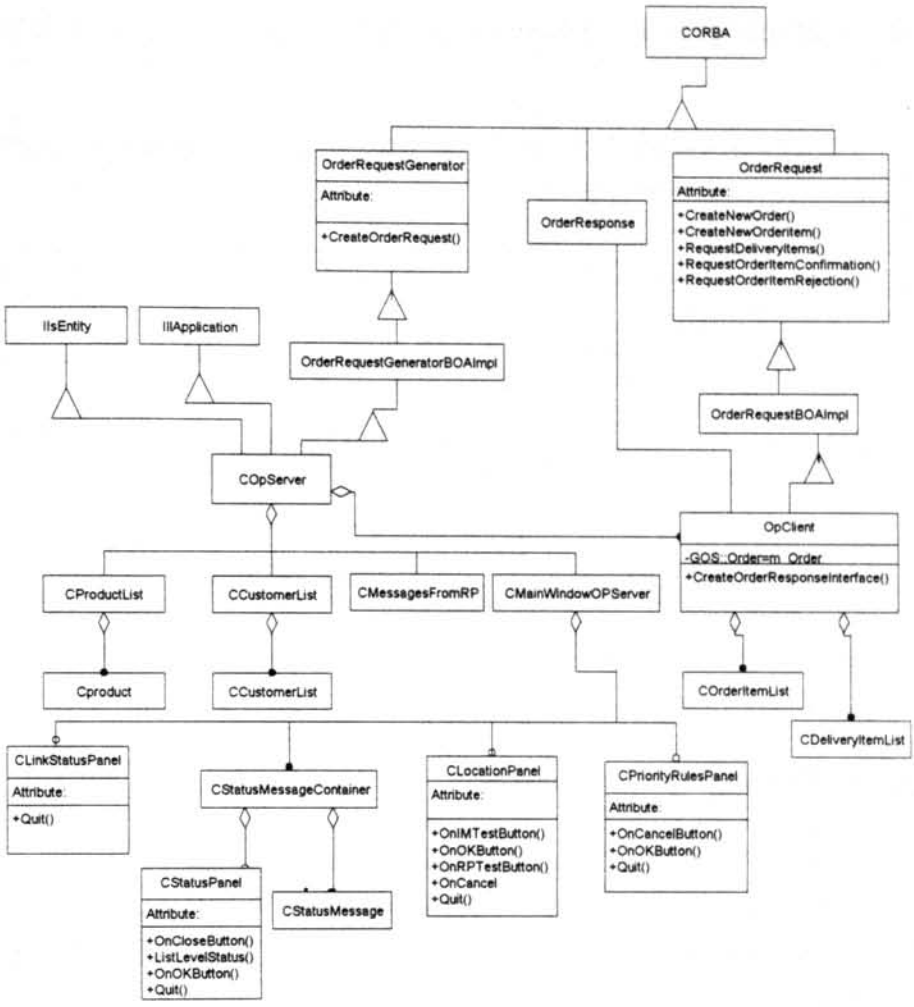


Figura 4.31 – Diagrama de classes do componente OPServer

Modelo dinâmico

Após a fase de inicialização, o componente fica num estado de repouso (*idle*). Deste estado, passa a um estado de espera (*waiting*), quando recebe um novo pedido de um qualquer componente OPClient e caso não exista quantidade ATP para o intervalo de tempo especificado no pedido. A saída deste estado ocorre quando é recebida uma resposta (oferta) do componente de planeamento (RP).

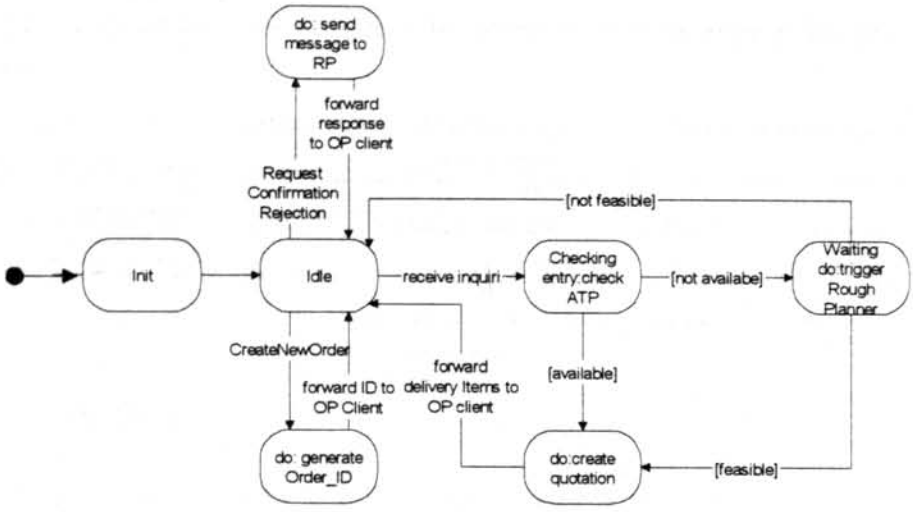


Figura 4.32 – Diagrama de estados do componente OPServer

4.2.7.4 Interacção com o utilizador

Neste componente, a interacção com o utilizador serve essencialmente a monitorização dos pedidos pendentes e submetidos via componentes OPClient. Além disso, o utilizador tem a possibilidade de definir a estratégia de sequenciamento da fila de pedidos pendentes.

Após a operação de configuração do componente, em que são definidos os nomes ou endereços IP dos componentes com que interactua, o utilizador tem a possibilidade de visualizar a actividade do sistema, segundo diferentes "filtros de informação".

A Figura 4.33 apresenta um exemplo desta visualização. Na janela denominada *System Activity*, são apresentadas em tempo-real todas as acções decorrentes da criação de um novo pedido e resultantes da interacção com os diversos componentes envolvidos.

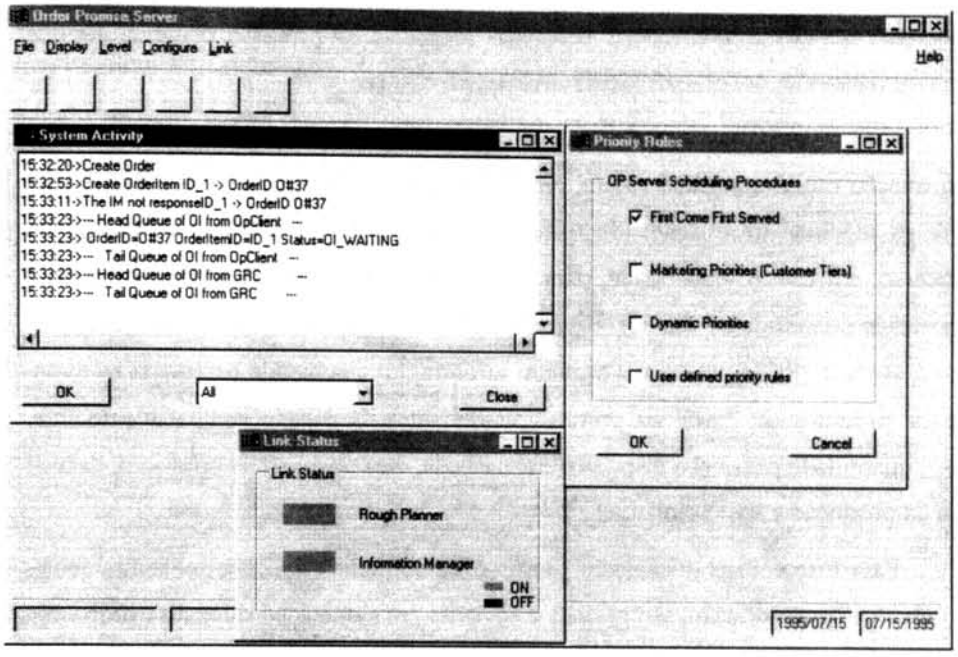


Figura 4.33 – Exemplo de janelas do OPServer

4.2.8 Rough Planner

4.2.8.1 Descrição funcional

O componente *Rough Planner* (RP) é responsável pelo processo de planeamento, ao longo da *Empresa Virtual*, de um pedido submetido através dos componentes OPClient e OPServer. Com vista a satisfazer o pedido em planeamento (e de acordo com o cenário 2 apresentado na secção 4.2.4), o RP interactua com os diversos modelos locais de capacidade (componentes RCM), desencadeando um processo de negociação, por forma a determinar uma rota através da *Empresa Virtual*, assim como as datas de início e de fim da produção em cada uma das unidades envolvidas nessa rota.

O RP, na sua função (central) de coordenador do planeamento, começa por gerar *propostas* de produção para cada um dos modelos locais de capacidade, recebendo destes, *ofertas* que resultam da avaliação feita por esses modelos da possibilidade e das

condições de execução local das propostas recebidas. O processo continua até que uma ou mais soluções globais satisfatórias sejam obtidas.

Este processo de negociação, ilustrado na Figura 4.34, envolve, naturalmente, informação muito agregada. Assim, uma *proposta* (enviada pelo RP a um RCM) consiste na seguinte informação: identificação do subproduto (associado ao estágio do processo, em causa), quantidade, data de início possível, data de entrega desejada e prioridade (associada ao cliente). Uma *oferta*, gerada por um determinado modelo local de capacidade (RCM), contém a seguinte informação: quantidade necessária na entrada (de produtos ou "matérias primas" provenientes do estágio imediatamente anterior), quantidade possível a disponibilizar na saída, data de início da produção, data de fim da produção e custo global.

Para tornar mais eficiente o processo de comunicação entre os componentes envolvidos na negociação, são geradas e enviadas em simultâneo diferentes propostas, para cada um dos modelos de capacidade. Além disso, a implementação das interfaces é assíncrona, permitindo que a negociação decorra, em paralelo, com todos os modelos de capacidade envolvidos.

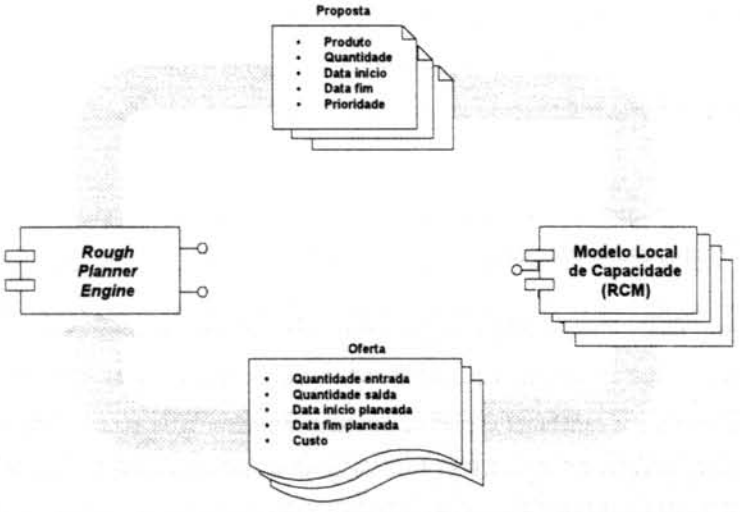


Figura 4.34 – Interacção entre o componente de planeamento (RP) e os modelos locais de capacidade (RCM)

Note-se que a adopção do conceito de *componente* no desenvolvimento do sistema, permitiu que a definição de interfaces fosse, em certo sentido, universal, conseguindo-se desta forma que, ao mesmo tempo, se tenha em funcionamento vários RCM com características eventualmente muito diferentes.

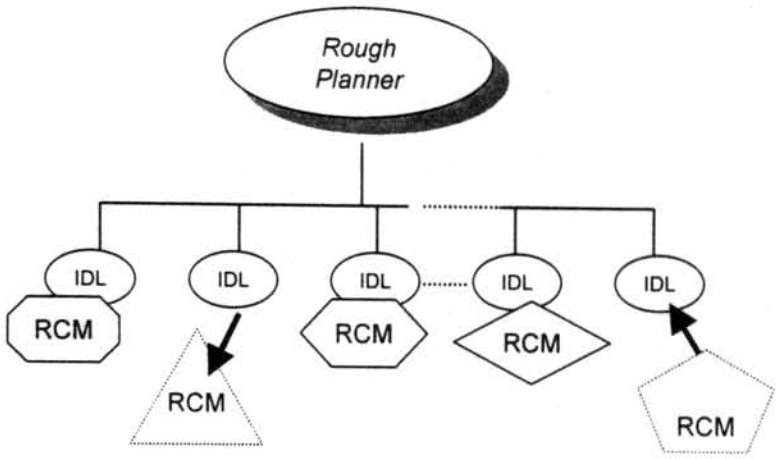


Figura 4.35 – Heterogeneidade de componentes RCM

A Figura 4.35 ilustra a capacidade de *plug-and-play* do sistema, nomeadamente quanto ao facto de poderem coexistir em qualquer instante, sem necessidade de inicializar o sistema, diferentes modelos RCM, e igualmente quanto à possibilidade de se "ligarem" ou "desligarem", durante o funcionamento, novos componentes.

Para além da interacção com os componentes RCM, o componente RP interactua também com o componente responsável pela gestão dos aspectos logísticos da *Empresa Virtual* (componente LM). Isto é, ao gerar as propostas para cada um dos estágios produtivos, o RP considera as restrições logísticas existentes, tais como o tempo de transporte entre estágios, a disponibilidade de transporte e o respectivo custo.

Por outro lado, a monitorização do processo de negociação, nomeadamente a visualização das propostas geradas e ofertas recebidas, assim como a parametrização do componente, é feita por um componente externo, denominado RPView, que permite a sua realização remotamente.

Refira-se ainda que, dado o papel central desempenhado pelo "motor de planeamento" do RP, foram, no âmbito deste trabalho, desenvolvidos vários algoritmos

de planeamento para a geração de propostas e a avaliação de ofertas. Estes algoritmos têm em conta o carácter eminentemente combinatório dos problemas em causa e baseiam-se em técnicas de pesquisa local (*simulated annealing* e algoritmos genéticos). A aplicação e desenvolvimento destes algoritmos ao problema em estudo, serão objecto de descrição pormenorizada no capítulo 5.

4.2.8.2 Modelação conceptual do componente

A modelação e a implementação deste componente foram baseadas em diversos padrões de projecto (*design patterns*[26]), procurando dessa forma capitalizar-se as melhores práticas adoptadas em situações conceptualmente similares.

O componente foi internamente organizado segundo os sub-módulos principais descritos de seguida.

Algoritmo de optimização

Este sub-módulo é responsável pelo processo de planeamento que, como referido, se pode considerar a parte "central" de todo o sistema. Foi definida uma interface genérica e comum a todos os algoritmos, por forma a permitir o recurso a estratégias diferentes, através da selecção pelo utilizador, entre vários, de um dado algoritmo de optimização, e sem necessidade de "re-arrancar" com o componente. Assim, definiram-se classes que encapsulam diferentes estratégias de planeamento, a partir do mesmo tipo de interface comum. O desenvolvimento e implementação foram baseados no padrão de projecto *Strategie* descrito em (Gamma et al., 1995). Basicamente, este padrão de projecto define uma família de algoritmos, encapsula cada um deles, tornando-os intermutáveis, e possibilitando, dessa forma, que sejam vistos do exterior

[26] O recurso a *design patterns*, no desenvolvimento de *software*, tem emergido como uma das mais promissoras abordagens para a melhoria da qualidade do *software*. Basicamente, *design patterns* descrevem soluções simples e "elegantes" para problemas específicos de *software* orientado a objectos (Gamma et al., 1995). A motivação principal da utilização de *design patterns*, no âmbito da engenharia do *software*, tem sido a cada vez mais necessária reutilização de elementos, característica das áreas da engenharia tradicionais. O livro *Design patterns: Elements of Reusable Object-Oriented Software* (Gamma et al., 1995), é talvez a principal e mais importante referência nesta matéria. Nesta obra, os autores apresentam vinte e três *design patterns* estruturados de acordo com os seguintes elementos principais: nome do padrão, descrição do problema a que se destinam, a solução para o problema e respectiva aplicação e por fim as consequências da aplicação da solução.

como um único algoritmo (ou estratégia), com comportamentos diferentes e específicos.

Controlador

Dada a complexidade do RP, utilizou-se uma estratégia de controlo e coordenação, baseada numa máquina de estados, por forma a garantir a robustez comportamental do componente, ao longo de todo o processo de planeamento. O desenvolvimento e implementação foram baseados no padrão de projecto *State*. Este padrão de projecto permite que um objecto altere o seu comportamento, quando se verificam mudanças internas do seu estado. Isto é, consegue-se que o comportamento de um objecto (por exemplo, quando se invoca um determinado método) dependa do seu estado interno. A ideia chave neste padrão de projecto, é a definição de uma classe abstracta que declara uma interface comum a todas as classes que representam os diferentes estados operacionais (ou seja, as classes que implementam o comportamento específico desejado para cada estado).

Gestão de propostas

Este sub-módulo é responsável pela conversão de uma proposta de solução, gerada pelo algoritmo de optimização, no conjunto de sub-propostas a enviar a cada modelo local de capacidade (RCM). Além disso, e devido à característica assíncrona do processo de comunicação entre o RP e cada um dos RCM, este sub-módulo efectua a gestão de *timeout* entre o envio de uma proposta e a respectiva recepção da oferta. Outra tarefa deste sub-módulo, consiste na agregação de propostas a enviar a cada um dos RCM, procurando assim que na mesma transacção sejam enviadas diversas propostas, com a finalidade de reduzir o *overhead* da comunicação.

Gestão de ofertas

Este sub-módulo é responsável pela recepção das ofertas geradas pelos modelos de capacidade e pela "associação" dessas ofertas à proposta correspondente. Esta associação é feita pela recombinação das ofertas recebidas (relativas a uma dada proposta de solução gerada pelo algoritmo de optimização), por forma a avaliar o custo

global de uma solução (através da integração dos custos individuais de produção, custos de transporte, custos de armazenamento, etc).

Com a finalidade de aumentar, de uma forma significativa, o desempenho do processo de negociação, implementou-se localmente no RP, uma base de dados de ofertas. Assim, é possível verificar se já existe registada alguma oferta para a proposta em negociação. Caso exista, o sub-módulo de gestão de ofertas automaticamente informa de tal facto o sub-módulo de gestão de propostas, evitando que dessa forma seja enviada, através da rede de comunicação, uma proposta para a qual se dispõe já de uma oferta. Esta característica do sistema é particularmente importante atendendo a que, sendo o processo de negociação evolutivo e verificando-se grandes semelhanças entre soluções consecutivas, se consegue uma redução significativa do tempo de negociação. Além disso, numa situação real de comunicação via WAN, procura-se assim reduzir a possibilidade do sistema de comunicações ser um "estrangulamento" do processo de negociação.

Gestão de RCM

Este sub-módulo é responsável pelo registo de cada modelo local de capacidade (RCM). O processo de negociação decorre unicamente entre as entidades que compõem a *Empresa Virtual*, com a configuração da rede produtiva estabelecida na fase de registo de cada uma das unidades (locais) que a constituem. Assim, quando fica activo um determinado RCM, o RP é informado das suas características principais, nomeadamente, qual o estágio produtivo em que a unidade associada a esse RCM se insere. Em qualquer instante, pode ocorrer a entrada ou saída de novas entidades na *Empresa Virtual*, sem que seja necessário proceder a qualquer re-configuração explícita do processo de negociação.

Note-se que todas as interacções entre o RP e os RCM são realizadas de forma assíncrona, conseguindo-se assim o maior desempenho possível. Desta forma, o processo de negociação decorre em paralelo.

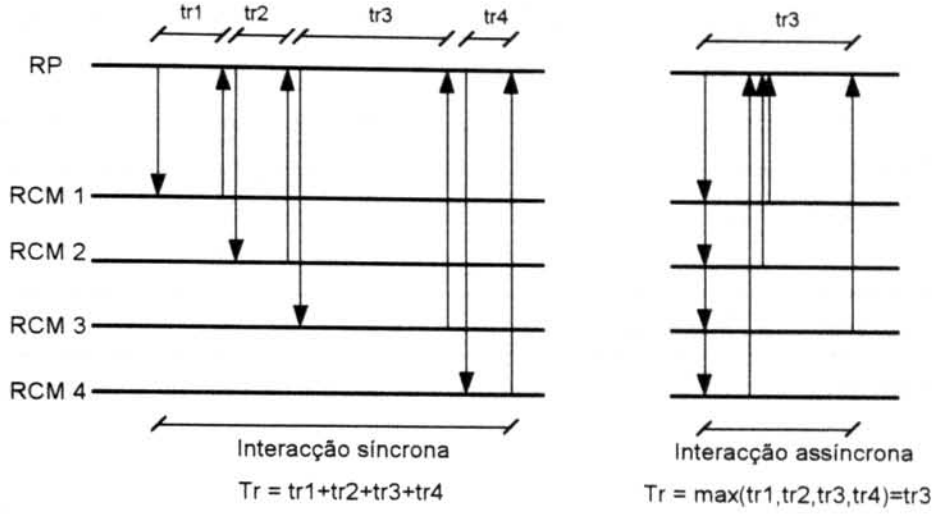


Figura 4.36 – Interacção síncrona versus interacção assíncrona

A Figura 4.36 ilustra as duas possibilidades de interacção entre os componentes RP e os RCM, num cenário constituído unicamente por quatro modelos de capacidade. No caso síncrono, o tempo de negociação, para uma única proposta, é igual ao somatório dos tempos de resposta individuais de cada modelo de capacidade. No caso assíncrono, para as mesmas condições, o tempo total é igual ao maior tempo de resposta individual (para o cenário apresentado é igual ao tempo de resposta do modelo RCM 3).

No caso geral, pode-se estabelecer o seguinte resultado que define um majorante para o tempo de resposta total associado à negociação de uma única proposta: independentemente do número de modelos locais de capacidade existentes, o tempo total de avaliação de uma proposta global (solução) é igual ao tempo de resposta do modelo de capacidade mais demorado.

4.2.8.3 Interfaces

No RP foram definidas e implementadas interfaces para os seguintes componentes: RPView, RCM, OPServer, GIM e Logistic Model (a título ilustrativo, são apresentadas no Apêndice C extractos das interfaces desenvolvidas para o componente).

Para os dois primeiros, essas interfaces podem ser instanciadas tantas vezes quanto as necessárias, sendo para os restantes, instanciadas uma única vez.

A interacção entre o RP e os RCM foi implementada segundo um protocolo de mensagens assíncronas, ilustrada no diagrama de interacção da Figura 4.37 e descritas na Tabela 4.4.

Mensagem	Descrição
registerRCM	um determinado modelo local de capacidade informa o RP que está pronto a participar na rede de produção (indicando o estágio onde deve ser inserido)
insertInquirySeq	o RP envia um conjunto de propostas para avaliação local
insertOfferSeq	o RCM responde às propostas recebidas com uma sequência de ofertas
insertNotification	o RP aceita ou rejeita ofertas recebidas
insertAcknowledge	confirmação do RCM, aquando de uma aceitação por parte do RP
unregisterRCM	o RCM informa o RP que passou a estar indisponível para participar no processo de planeamento

Tabela 4.4 – Mensagens de interacção entre o componente RP e os componentes RCM

Basicamente, e após a fase de registo no RP por parte de cada RCM (invocação do método *registerRCM*), a interacção entre o RP e os RCM é feita da forma descrita a seguir. O RP envia, a cada RCM envolvido no processo de negociação, um conjunto de propostas para avaliação local (*insertInquirySeq*); cada RCM, ao qual foram enviadas propostas, responde, após a avaliação local, com uma sequência de ofertas (*insertOfferSeq*). A consolidação e avaliação das ofertas recebidas pelo RP, dará origem a acções de aceitação ou rejeição de ofertas. No caso particular de uma aceitação (*insertNotification*), o RCM envolvido procede a uma re-avaliação da proposta associada à oferta gerada e em caso afirmativo procede à respecticva confirmação (*insertAcknowledge*).

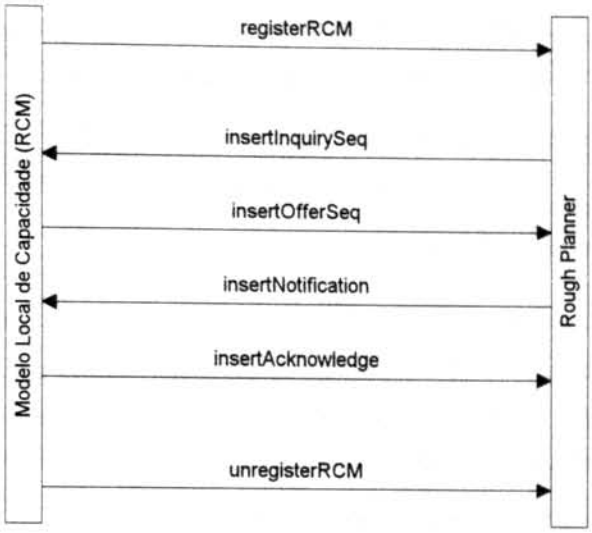


Figura 4.37 – Protocolo de interacção entre o componente RP e os componentes RCM

A interacção entre o RP e o componente responsável pelo modelo logístico LM, foi feita através de uma interface síncrona no sentido do RP para o LM e numa interface assíncrona no sentido contrário.

A interface síncrona é usada pelo RP para inicialmente obter informação logística essencialmente estática e respeitante à configuração actual da *Empresa Virtual* (tempos de transporte, custos de transporte e custos de armazenamento entre todas as entidades registadas).

A interface assíncrona implementa basicamente um mecanismo de notificação, informando o RP de alterações relativas a informação logística, garantindo assim que o RP se encontra sempre actualizado.

Esta interface permite por exemplo obter, durante o processo de negociação, resposta às seguintes questões:

- Qual o menor tempo de transporte entre a unidade J e a unidade K para o período X?
- Qual o menor tempo de transporte entre o estágio A e o estágio B para o período X?
- Qual o custo de transporte entre as unidades F e D?

- Qual o custo de armazenamento para uma quantidade Y de produto Z entre os estágios S e H para o período compreendido entre K e W?

Um extracto desta interface IDL é apresentado no Apêndice C.

4.2.8.4 Diagrama de classes

Na Figura 4.38, apresenta-se, numa versão simplificada e a título ilustrativo, o diagrama de classes do RP. Para além de um conjunto de classes responsáveis pelas interfaces com outros componentes, no diagrama salientam-se as classes fundamentais do componente: **RP-Server**, **Algorithm**, **RP_State** e **RCM_Manager**.

A interface da classe **Algorithm** reflecte o nível de abstracção considerado no desenvolvimento do componente. Basicamente, o processamento implementado por esta classe considera duas fases fundamentais. A primeira fase decorre sem a intervenção de quaisquer componentes externos, e corresponde à elaboração de sub-propostas com base no pedido recebido e de acordo com a estratégia do próprio algoritmo de optimização (que como referido, pode ser escolhido entre várias alternativas). Na segunda fase, o algoritmo processa a informação recebida dos componentes RCM das unidades envolvidas no processo de produção. O ciclo repete-se, até se verificar uma determinada condição de paragem (por exemplo, quando se atinge um determinado número de iterações, parametrizável pelo utilizador)[27].

Na classe **Algorithm**, o método **startOptimization** procede à inicialização do algoritmo, de acordo com parâmetros cujos os valores são definidos pelo utilizador, através do componente RPView. De seguida, e iteradamente, é invocado o método **stepForSendingRequests,** no qual são enviadas propostas aos componentes RCM. O método **stepAfterRequestsArrived** é invocado quando todas as ofertas correspondentes às propostas enviadas, são recebidas. A classe **RP_State** implementa a máquina de estados do componente.

---

[27] Os algoritmos desenvolvidos para o componente RP e implementados nesta classe serão pormenorizadamente apresentados no capítulo 5.

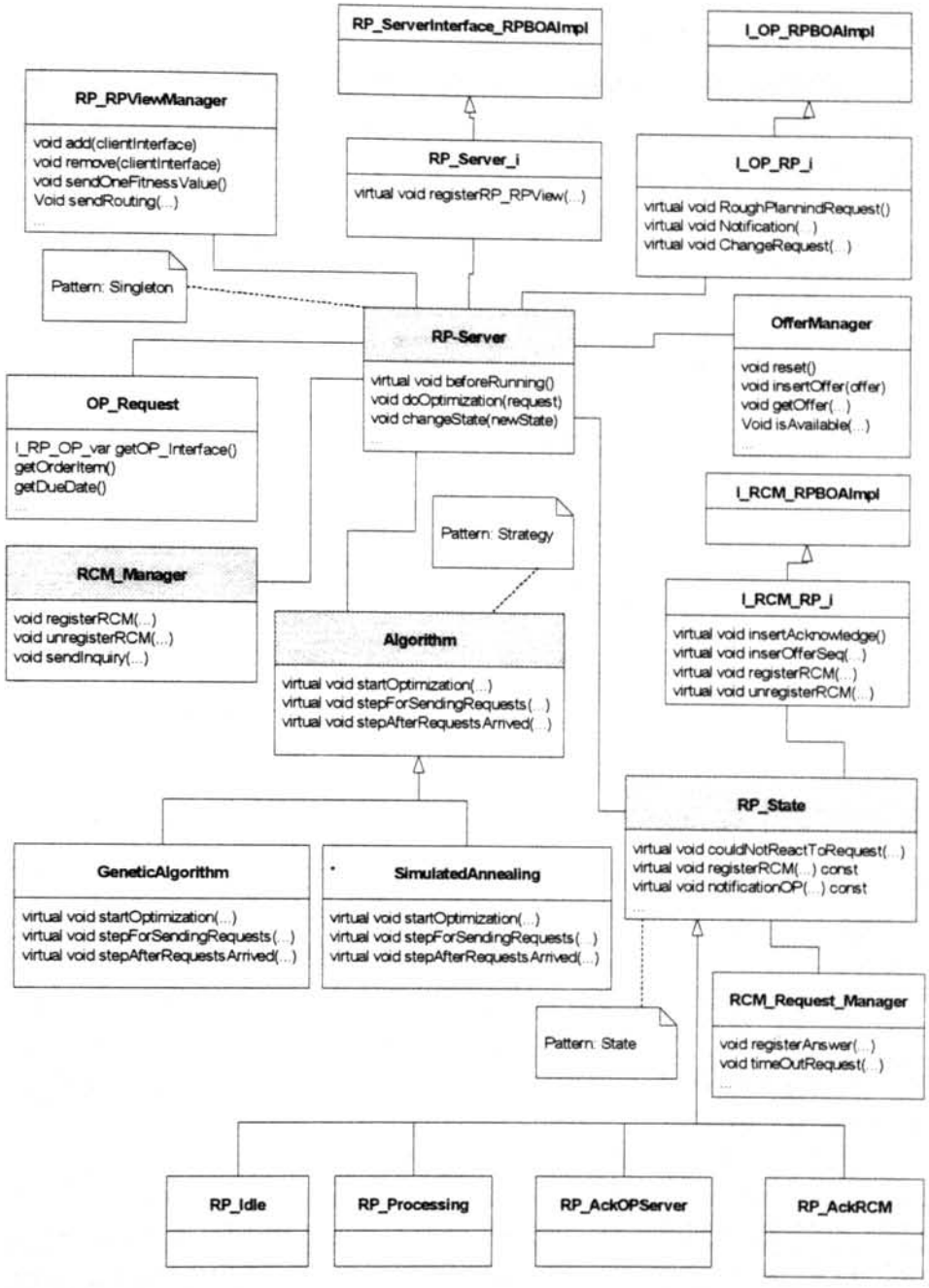


Figura 4.38 – Diagrama de classes do componente RP

Modelo dinâmico

Na Figura 4.39, encontra-se esquematicamente representada a interacção entre as classes principais, a partir do instante em que se inicia o processo de planeamento e durante uma iteração completa.

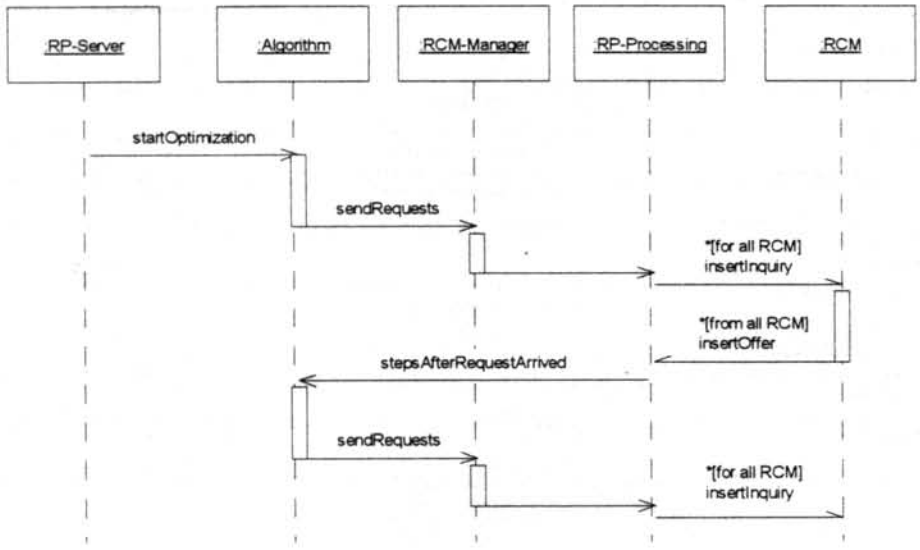


Figura 4.39 – Diagrama de interacção entre classes (RP)

A Figura 4.40 apresenta o diagrama de estados do componente RP. Após a fase de inicialização (período em que o RP estabelece ligação ao componente global de gestão de informação), o componente fica num estado de repouso (*Idle*). A transição deste estado para o estado de processamento (*Processing*) acontece quando é recebido um pedido do OPServer.

Após ter terminado o processo de negociação com os componentes RCM, isto é, após se terem recebido e "consolidado" todas as ofertas (e depois de ser enviado o resultado do planeamento ao componente que submeteu o pedido), passa-se a um estado (*AckOPServer*) em que se aguarda uma resposta de aceitação ou rejeição. No caso de aceitação, passa-se para um estado (*AckRCM*) em que, por sua vez, se procede junto de cada um dos modelos locais de capacidade envolvidos na satisfação do pedido, a uma confirmação das consições negociadas.

A passagem para o estado inicial de repouso, acontece numa qualquer situação de rejeição, ou quando é recebida a confirmação por parte dos componentes RCM, no caso de o RP se encontrar no estado *AckRCM*.

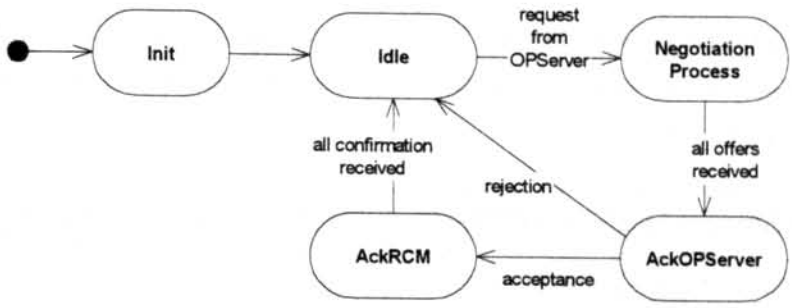


Figura 4.40 – Diagrama de estados do RP

4.2.8.5 Gestão do tempo

A característica distribuída do sistema, nomeadamente a distribuição dos diferentes componentes de capacidade RCM por diferentes locais (a uma escala que pode, mesmo, ser planetária), impõe procedimentos especiais para gestão do tempo, em virtude de eventuais diferenças horárias. A solução encontrada para este problema consiste usar sempre, em todas as transacções entre componentes, o tempo no formato UTC (*Universal Coordinated Time*), sendo todo o processo de negociação feito com base no tempo universal.

Naturalmente que, na apresentação de informação ao utilizador, é feita a respectiva conversão para o tempo local. Com esta finalidade, foram desenvolvidas duas classes de objectos, denominadas *DateTime* e *Duration*. A primeira é responsável pelas conversões tempo local / tempo UTC e tempo UTC / tempo local e pela gestão do tempo de acordo com a granularidade pretendida no planeamento. A classe *Duration* disponibiliza diversos métodos adequados à gestão de instantes de tempo.

4.2.8.6 Interacção com o utilizador

A interacção com o utilizador é feita através do componente RPView. Este componente pode ser instanciado tantas vezes quantas as necessárias, permitindo que a interacção com o RP possa ser feita remotamente.

Através deste componente, o utilizador tem a possibilidade de visualizar a configuração actual da *Empresa Virtual*, nomeadamente a disposição das unidades ao longo dos diferentes estágios produtivos.

A Figura 4.41 apresenta uma janela com a configuração da rede produtiva e a informação resultante do processo de negociação entre o componente RP e uma determinada unidade produtiva.

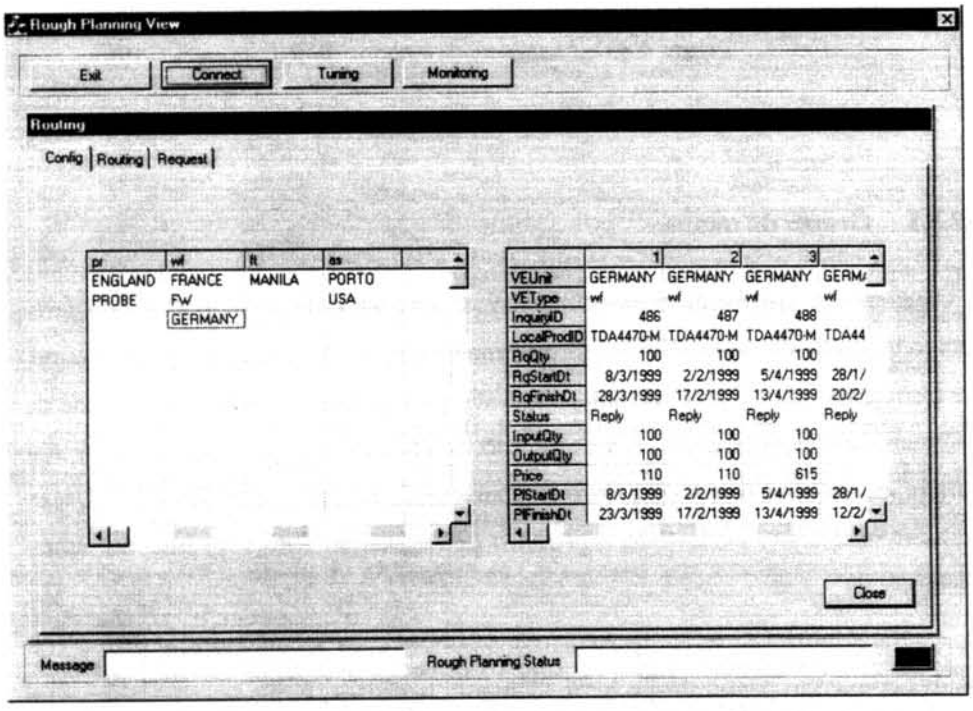


Figura 4.41 –Visualização da configuração da *Empresa Virtual*

No cenário representado nesta figura, a rede produtiva é constituída por oito unidades, distribuídas por quatro estágios. A selecção de uma qualquer entidade no quadro da esquerda, permite visualizar no quadro da direita todas as transacções que ocorreram durante o processo de negociação, envolvendo a unidade seleccionada.

O utilizador tem igualmente a possibilidade de visualizar a rota planeada para uma determinada encomenda (ver Figura 4.42) assim como, as sub-ordens alocadas a cada estágio da rede produtiva.

A evolução do processo de negociação, através nomeadamente da evolução do custo global associado a uma determinada solução, é representada graficamente e em tempo-real num gráfico de tendência, na mesma janela de visualização da rota planeada (ver Figura 4.42).

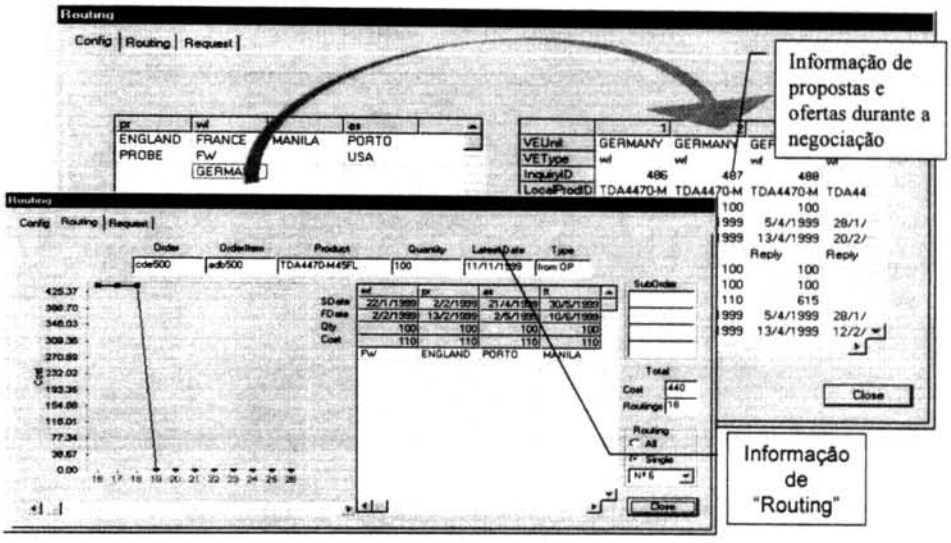


Figura 4.42 – Monitorização do processo de negociação durante o planeamento

Por sua vez, a Figura 4.43 apresenta uma janela para configuração do algoritmo de planeamento integrado no componente RP. Procurou-se aqui, tal como nas outras partes do sistema, desenvolver uma interface de fácil utilização e o mais intuitiva possível. Com essa finalidade, optou-se pela utilização de "barras horizontais" (*sliders*) para introdução dos valores dos parâmetros de decisão mais importantes e sempre que possível através de "escalas normalizadas" (percentagens), não se exigindo assim que o utilizador conheça os pormenores técnicos envolvidos.

O utilizador pode, com o componente RP em funcionamento, visualizar ou alterar ("*on-line*") qualquer um dos parâmetros do algoritmo. Além disso, existe a possibilidade de parametrizar o algoritmo, estágio a estágio, isto é, podem definir-se, para

além da estratégia global de planeamento, estratégias de planeamento diferentes para cada um dos estágios.

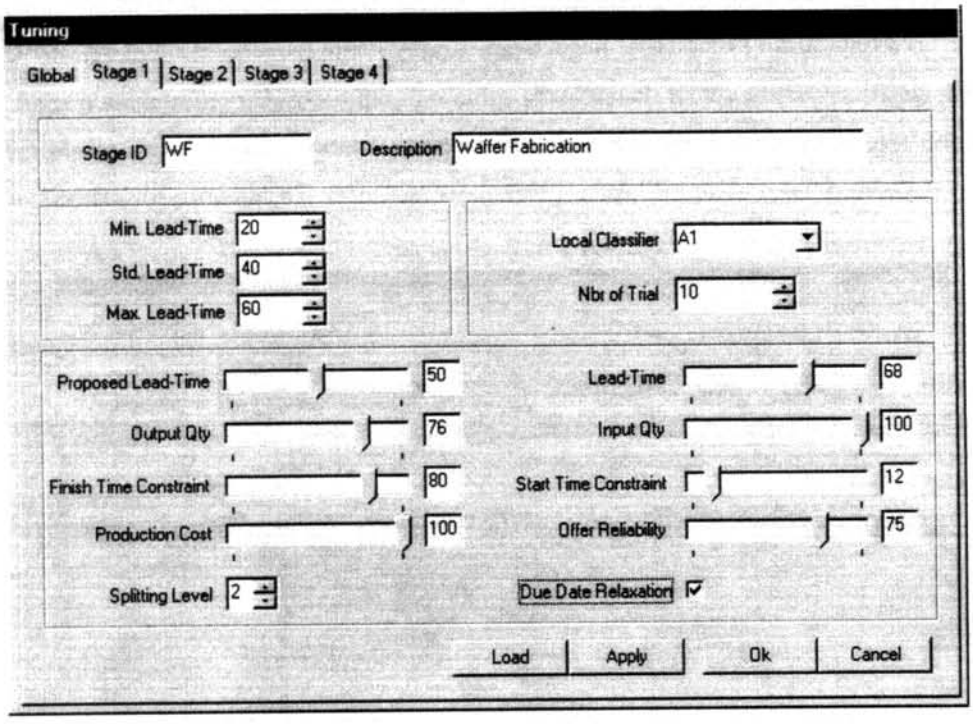


Figura 4.43 – Exemplo de uma janela de parametrização do algoritmo de planeamento

Finalmente, na Figura 4.44, apresenta-se a janela em que se define a estratégia global de planeamento e alguns parâmetros do algoritmo, de carácter geral.

O utilizador tem aqui a possibilidade de definir o número de iterações ou ciclos de planeamento, podendo igualmente definir o número de propostas a submeter a cada estágio, durante cada iteração. Por último, pode definir o número de soluções possíveis para cada pedido submetido, isto é, tem a possibilidade de definir o número de soluções alternativas a considerar.

Ao nível da estratégia global de planeamento, pode privilegiar ou penalizar soluções de acordo com quatro critérios: produção *just-in-time*, custos logísticos, custos globais de produção e *lead-time* global.

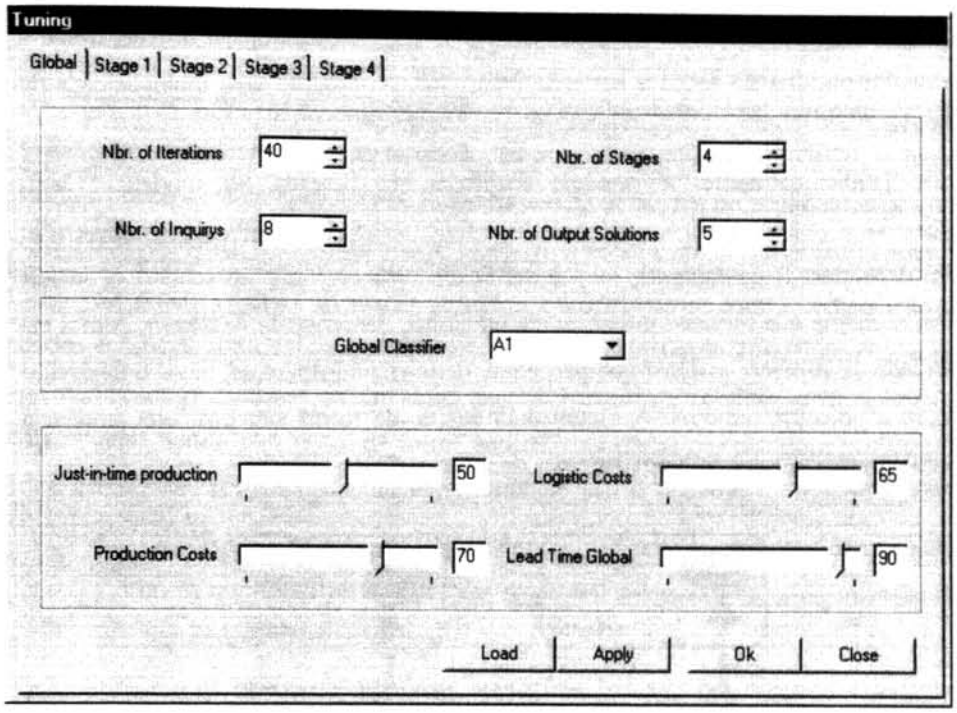


Figura 4.44 – Configuração da estratégia global de planeamento

4.3 Tecnologias e ambiente de implementação

4.3.1 Tecnologia de objectos distribuídos

Entre os requisitos identificados no âmbito deste trabalho e as soluções propostas para os sistemas de informação a operar no contexto da *Empresa Virtual*, considerou-se como particularmente crítica, a necessidade de recorrer a uma plataforma distribuída e baseada na tecnologia de objectos, por forma a permitir a distribuição de funcionalidades, aplicações e informação.

Com a enorme expansão das redes de computadores, o conceito de distribuição por vários recursos, do processamento, das aplicações e das funcionalidades tem-se tornado cada vez mais importante. Ao longo dos últimos anos, diversos métodos

emergiram com a finalidade de permitirem esta distribuição, nomeadamente, desde a elementar partilha de dados a sistemas avançados, suportando uma multiplicidade de serviços.

Tradicionalmente, a *tecnologia distribuída* era baseada no modelo "Cliente/Servidor", onde *clientes* e *servidores* tinham funcionalidades e âmbitos de intervenção bem definidos. Recentemente, tem-se verificado uma evolução no sentido de incluir entre o cliente e o servidor uma camada funcional, denominada *middleware*, com a capacidade de fornecer *serviços*[28] que permitam, de forma transparente, fiável e flexível, o acesso a objectos remotos. A Figura 4.45 ilustra, de forma sintética, esta tendência recente para a arquitectura dos sistemas de informação.

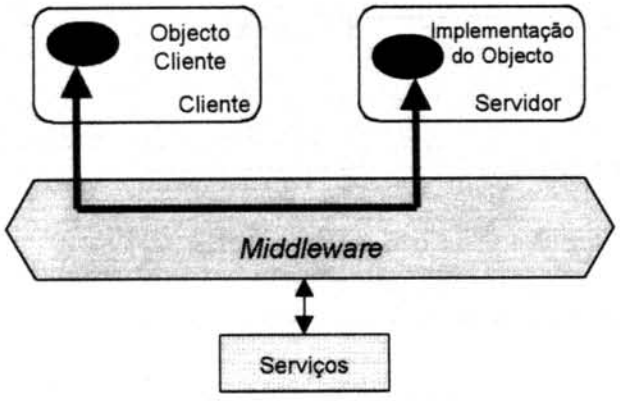


Figura 4.45 – Conceito de *middleware*

Em termos gerais, a camada *middleware* permite que um *cliente* interactue com um *servidor*, com um nível de abstracção tal, que não necessite de directamente especificar o nome ou endereço do recurso computacional onde se encontra o *servidor*.

[28] Define-se *serviço* como sendo um conjunto de funcionalidades bem definidas, acessíveis aos clientes, através de interfaces "publicadas" no sistema.

4.3.1.1 Plataformas distribuídas

As plataformas[29] de suporte à construção de aplicações distribuídas têm naturalmente sofrido uma enorme evolução ao longo do tempo. Assim, se na década de 70, se baseavam no conceito de mensagens, e ao longo da década de 80, em procedimentos de invocação remota (RPC), mais recentemente passaram a assentar em objectos distribuídos.

As potencialidades da noção de *objecto*, nomeadamente com a combinação de dados e funcionalidades/comportamento, e a estrita separação entre interface e implementação, tornaram-se atractivas no desenvolvimento e construção de aplicações distribuídas, sendo por isso, e de forma natural, exploradas nas plataformas distribuídas. Entre estas, as mais significativas actualmente, são as seguintes (Burghart, 1998):

- CORBA (*Common Object Request Brocker Architecture*) – resultou de um esforço de normalização de múltiplas empresas, agregadas numa organização designada por *Object Management Group* (OMG), para criar um ambiente genérico de desenvolvimento de aplicações, baseado em objectos e independente das linguagens de programação, sistemas operativos e tecnologias de comunicação.

- DCOM (*Distributed Component Object Model*) – definido pela Microsoft, sendo uma extensão da sua arquitectura de integração COM (*Component Object Model*).

- Java RMI (*Java Remote Method Invocation*) – é uma plataforma, desenvolvida pela Sun Microsystems, que possui uma arquitectura especificamente orientada à linguagem Java, não suportando, por isso, outras linguagens.

- DCE (*Distributed Computing Environment*) – é uma plataforma, formalizada pela OSF – Open Software Foundation (agora denominada Open Group), baseada em RPC (*Remote Procedure Call*) e em métodos procedimentais, não suportando por isso ambientes orientados a objectos.

Apesar das plataformas CORBA, DCOM e Java RMI partilharem os mesmos conceitos básicos, diferem nalgumas questões de pormenor (Orfali et al., 1996a). Ao

---

[29] Uma plataforma pode ser vista como um conjunto de rotinas, regras e protocolos, tipicamente organizados sob a forma de uma biblioteca, fornecendo um conjunto de funções de suporte às aplicações distribuídas (Brás, 1996).

contrário do DCOM e do Java RMI, o CORBA não é, por si só, um produto funcional, é antes uma especificação. Assim, comercialmente existem diversas implementações compatíveis com CORBA, nomeadamente, Iona Orbix, IBM SOM, Inprise Visibroker, etc.

No caso presente, a escolha de uma plataforma distribuída foi feita atendendo, não apenas aos requisitos de heterogeneidade do ambiente de funcionamento, como também à necessidade de interoperabilidade das aplicações a desenvolver com diferentes sistemas legados, nomeadamente, através da definição de interfaces abstractas adequadas. A escolha de uma plataforma compatível com CORBA é justificada não só por se tratar de uma arquitectura aberta e orientada a objectos, como essencialmente pela sua independência, quer da linguagem de programação quer do sistema operativo.

A plataforma DCE foi excluída porque não suporta um ambiente de computação orientado aos objectos e porque é uma plataforma especificamente orientada à linguagem C. A plataforma Java RMI, apesar de ser independente do sistema operativo, não foi considerada por ser exclusivamente específica à linguagem Java, e além disso "proprietária". A plataforma DCOM, apesar de ser independente da linguagem, foi igualmente excluída, por ser "proprietária" e por suportar unicamente os sistemas operativos da Microsoft.

Para uma apresentação destas diferentes plataformas, sugere-se a leitura de (Orfali et al., 1996b) e (Orfali et al., 1996a), sendo as diferentes plataformas comparadas em (Tallman and kain, 1998) e (Brando, 1995). Em (Wallnau et al., 1997) é analisado o impacto das tecnologias de objectos distribuídos, numa perspectiva de desenvolvimento de *software*.

4.3.1.2 CORBA

O consórcio internacional formado em 1989 e designado por OMG, definiu, no âmbito de um ambicioso projecto de normalização, uma arquitectura de referência designada por OMA (*Object Management Architecture*). Esta arquitectura define os blocos e os serviços que devem existir numa plataforma de apoio ao desenvolvimento de aplicações distribuídas. Assim, são definidos, com um elevado nível de abstracção, os vários elementos necessários para a coexistência e manutenção de objectos distribuídos, nomeada-

mente, o modo como os objectos são definidos, criados e partilhados, assim como a forma de serem invocados.

A Figura 4.46 ilustra os diferentes elementos constituintes da arquitectura OMA, onde o ORB (Object Request Broker) desempenha o papel central, constituindo o "bus" a que se ligam os componentes de *software* para interactuarem entre si.

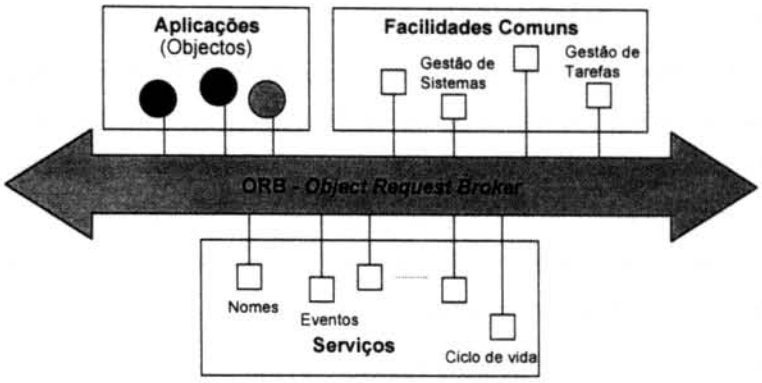


Figura 4.46 – OMA (*Object Management Architecture*)

De facto, o ORB é responsável pelo mecanismo de *invocação* entre objectos, permitindo, através dos serviços que disponibiliza, a interligação entre diferentes componentes existentes num sistema. O ORB é como que um mediador de interacções entre os componentes (clientes) que necessitem de serviços e os componentes (servidores) capazes de proporcionarem esses serviços. No âmbito da arquitectura OMA, foi especificado o CORBA.

O CORBA é uma especificação que define comportamentos e protocolos para reger a interoperabilidade dos objectos, num ambiente distribuído. As *interfaces* são definidas através de uma linguagem própria para descrição de interfaces (IDL) que suporta ligações às linguagens de programação mais utilizadas. A IDL é puramente declarativa e suporta herança múltipla. Na *interface* de um objecto, definem-se os atributos, os tipos de eventos possíveis, os serviços disponibilizados, os parâmetros de entrada e as características dos resultados possíveis.

Para além da linguagem de definição de interfaces e de diversos serviços (tais como gestão do ciclo de vida de componentes, gestão de nomes, gestão de eventos -

síncronos e assíncronos, serviços de armazenamento persistente, etc), as facilidades mais importantes definidas no CORBA são as seguintes:

- *Invocação estática e dinâmica de objectos* – a invocação estática (*IDL Stubs*) é semelhante à invocação de um procedimento remoto (RPC) através de uma rotina de adaptação previamente criada aquando da compilação, permitindo que um objecto cliente invoque um método num objecto remoto (basicamente, é gerado localmente no cliente um *proxy* para cada objecto remoto); a invocação dinâmica (*DII*) baseia-se na construção da mensagem de invocação, após a consulta de um inventário de interfaces, no momento da invocação, sendo adequada em situações em que as interfaces não são conhecidas no momento da compilação, permitindo que objectos clientes interactuem com implementações de objectos que não são conhecidos na altura em que a compilação ocorreu.

- *Repositório de interfaces* (*Interface Repository*) – é um serviço de registo de *interfaces*, que permite conhecer dinamicamente os métodos que essas interfaces suportam, assim como os respectivos parâmetros.

A Figura 4.47 representa a estrutura de um ORB, do lado do cliente e do lado do servidor (compatível com a especificação CORBA 2.2). Estão representados na figura, alguns dos elementos já referidos (onde a invocação feito pelo *cliente* se realiza ao mais alto nível, isto é, sem necessidade de especificar o *servidor*).

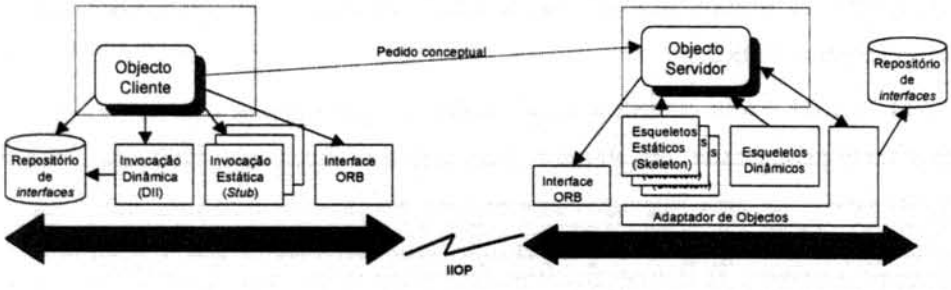


Figura 4.47 – Estrutura de um ORB CORBA (Cliente e Servidor)

A separação clara entre interfaces e implementações, do lado do cliente e do lado do servidor, permite que o ORB possa invocar objectos implementados noutras

linguagens, eventualmente até em linguagens não orientadas a objectos. É assim possível e fácil considerar e fazer a integração com sistemas legados.

Na Figura 4.47, representa-se ainda o *adaptador de objectos* (OA) que permite, do lado do servidor, a instanciação e a invocação de objectos, disponibilizando os meios através dos quais os vários tipos de objectos têm acesso aos serviços do ORB. Além de tratar os pedidos de serviços por parte dos clientes, direccionando-os para a rotina de adaptação respectiva (*skeleton*), é responsável por criar objectos e gerir as referências para esses objectos. O CORBA especifica vários tipos de OA, mas todos os ORB comerciais proporcionam um OA genérico denominado *Basic Object Adaptor* (BOA).

Ainda na Figura 4.47, está representada a interoperabilidade entre ORB que se encontra definida através de um protocolo, denominado de GIOP – *General Inter-ORB protocol*. Este protocolo define e especifica um conjunto de formatos de mensagens, uma representação comum dos dados, e a respectiva sintaxe. No caso específico do mecanismo de transporte baseado no TCP/IP, foi especificado o protocolo designado por *Internet Inter-ORB protocol* (IIOP).

Para uma apresentação completa da plataforma CORBA, e em particular para uma clarificação de aspectos de implementação e desenvolvimento de aplicações baseadas em CORBA, sugere-se a leitura de (Orfali et al., 1997), (Siegel, 1996) e (Otte et al., 1996).

4.3.2 Ambiente de implementação

No projecto e desenvolvimento do sistema *Order Promise*, apresentado neste capítulo, recorreu-se às seguintes ferramentas e ambiente de desenvolvimento:

Sistema operativo

- Windows NT Server 4.0 e NT Workstation 4.0

Linguagem de programação

- C++ em Visual C++ 4.2

Object Request Broker

- ORBIX V2.2 (Iona Technologies)

Protocolo de comunicação

- TCP/IP

Sistema de base de dados e interfaces

- Oracle Workgroup Server V7.3
- ILOG DB Link V2.0

Desenvolvimento de interfaces gráficas

- ILOG Views V2.41

Apesar do sistema operativo e do ambiente de desenvolvimento C++ utilizados, serem "proprietários", todos os componentes foram implementados por forma a garantir a sua portabilidade entre diferentes ambientes, nomeadamente através do recurso a interfaces independentes da plataforma, tais como o ILOG Views para interfaces com o utilizador e o ILOG DB Link para interfaces com Base de Dados.

Capítulo 5

Modelos e Abordagens Heurísticas

5.1 Introdução; revisão bibliográfica

5.2 Estruturação geral do problema

5.3 Modelos e casos particulares

5.4 Meta-heurísticas

5.5 Simulated Annealing

5.6 Algoritmos Genéticos

5.7 Heurística multi-atributo

5.8 Conclusões

5 MODELOS E ABORDAGENS HEURÍSTICAS

A actual tendência de evolução para formas de organização mais globais, nomeadamente através do estabelecimento de redes de cooperação com todas as entidades intervenientes na cadeia de fornecimento, levanta complexas questões de coordenação. No capítulo anterior, apresentou-se um sistema informático, cuja arquitectura permite apoiar, no contexto e domínio apresentados, a tomada de decisões no processo de negociação e aceitação de encomendas em ambientes eminentemente distribuídos.

Neste capítulo, são apresentados modelos e abordagens heurísticas para a resolução do problema específico de coordenação que ocorre no processo de planeamento e aceitação de novas encomendas, em redes de cooperação multi-estágio, quando se tem em conta não só a produção mas também as questões logísticas.

Os algoritmos apresentados neste capítulo, e integrados no sistema informático apresentado no capítulo anterior, permitem o planeamento de novas encomendas segundo uma abordagem inovadora, baseada em técnicas de *pesquisa local* no espaço das soluções, através de *meta-heurísticas*.

Após ser apresentada uma revisão bibliográfica da investigação mais relevante nesta área, é feita uma estruturação do problema em causa, enquanto problema de optimização de carácter combinatório. A aplicação de duas *meta-heurísticas*, baseadas em *Simulated Annealing* e *Algoritmos Genéticos*, à resolução do problema, será apresentada pormenorizadamente em secções seguintes. Finalmente, é apresentada uma heurística construtiva específica, baseada numa análise "multi-atributo" do problema, e com a qual se procura dar resposta a algumas questões de indiscutível relevância prática.

5.1 Introdução; revisão bibliográfica

As organizações em rede têm, nos últimos anos, proliferado por variadas áreas de negócios (com particular ênfase no sector industrial), suscitando por isso um interesse crescente pela comunidade científica (Ching et al., 1993).

A necessidade cada vez maior de se conseguir resposta adequada às solicitações de um mercado em contínua mudança e a constante procura de estruturas produtivas

cada vez mais eficazes, têm conduzido as empresas a explorarem novos modelos organizacionais, nomeadamente através da constituição de redes envolvendo fornecedores de matéria-prima e de componentes intermédios, unidades de montagem, unidades de armazenamento e centros de distribuição.

No passado, as organizações encaravam a tomada de decisões segundo uma perspectiva direccionada para dentro da empresa e, portanto, desacoplada da sua envolvente, nomeadamente, fornecedores e clientes. Basicamente, as entidades que participavam na cadeia de fornecimento eram geridas de forma independente, não existindo consequentemente um enfoque directo nas actividades de coordenação inter-entidades. A continuidade do processo produtivo era garantida, tradicionalmente, com o recurso a múltiplos pontos intermédios de armazenamento.

As exigências decorrentes da globalização dos mercados, não permitem, hoje em dia, que a tomada de decisões tenha lugar de forma desacoplada e com âmbito estritamente local, pelo que actualmente as actividades de coordenação entre entidades envolvidas numa cadeia de fornecimento são nucleares para se alcançarem os elevados níveis de desempenho definidos a nível global.

A existência de múltiplas entidades cooperantes, com interesses complementares (mas muitas das vezes, parcialmente conflituosos), a incompletude e incerteza da própria informação e a insuficiência e limitação de recursos, levanta questões complexas em relação à integração das acções a tomar, pelo que a coordenação e mediação se revela fundamental na resolução de conflitos inter-entidades. A modelação da cadeia de fornecimento segundo uma perspectiva global de estreita interacção entre as suas entidades constituintes, e nomeadamente através da constituição de estruturas produtivas em rede, originou novos problemas e desafios, pelo que só recentemente tem sido objecto de investigação por parte da comunidade científica (Thomas and Griffin, 1996).

Genericamente, pode afirmar-se que a variabilidade e a complexidade resultantes das actividades das empresas de âmbito internacional (empresas multinacionais) num cenário de produção global, e a operarem segundo estruturas em rede[30] originam

[30] Podem identificar-se basicamente dois tipos de organização: operação autónoma através de diferentes unidades fabris distribuídas geograficamente e sem dependência directa ao nível dos processos produtivos, ou divisão dos processos de produção por diferentes unidades fabris com relações directa de dependência entre elas. A estas duas configurações extremas correspondem naturalmente diferentes problemas de coordenação.

dois tipos de problemas de tomada de decisões: a configuração da rede de recursos em que as empresas operam, por uma lado, e a sua coordenação, por outro.

As questões de configuração são essencialmente estratégicas, envolvendo basicamente a identificação de oportunidades, a disponibilidade de recursos complementares, assim como a sua organização e adequação a diferentes ambientes. A coordenação, tendo um enfoque mais táctico ao nível da tomada de decisões, envolve essencialmente a gestão e o planeamento das redes. A coordenação é vista igualmente como um instrumento fundamental no aumento do desempenho global das redes estabelecidas.

Diferentes perspectivas podem contudo ser consideradas na coordenação de recursos dispersos (geograficamente distribuídos), de acordo com o seu posicionamento ou interacção na cadeia de valor. Assim, a coordenação pode envolver actividades ao nível da investigação e desenvolvimento (I&D), da concepção de produtos e processos, do fornecimento de materiais, da produção, da distribuição, etc.

No âmbito do presente trabalho, considera-se a coordenação apenas na perspectiva da produção e da logística, e com um âmbito global. Assim, procura-se garantir com a coordenação que, nas decisões de produção ao nível de cada entidade envolvida, sejam considerados, para além das restrições de capacidade produtiva, objectivos globais envolvendo obrigatoriamente questões logísticas fundamentais, tais como transportes e armazenamentos intermédios.

Uma revisão da investigação feita nesta área pode ser encontrada em (Pontrandolfo and Okogbaa, 1999) e em (Thomas and Griffin, 1996). Segundo Pontrandolfo, constata-se haver muito trabalho de investigação na vertente da configuração e organização, em detrimento das questões relacionadas com a coordenação, e em particular com a coordenação das actividades produtivas. Estes autores identificam a necessidade de se desenvolver uma abordagem sistemática ao problema da coordenação, nomeadamente através de um quadro de referência que permita sistematizar e analisar as questões decorrentes das redes de recursos heterogéneos e dinâmicos. Este quadro de referência possibilitaria não só o desenvolvimento de ferramentas de decisão adequadas, como permitiria igualmente sustentar a teoria económica "neo-liberal" na sua ideia chave de que é preferível, numa economia global, a cooperação em detrimento da competição.

No mesmo artigo os autores propõem um quadro conceptual suportando a construção de modelos logísticos de empresas a operarem em rede. São identificadas quatro entidades base, organizadas em quatro estágios consecutivos: fornecedores (matéria-prima e componentes intermédios), unidades logísticas de entrada, unidades de montagem final e unidades logísticas de saída. No modelo logístico apresentado, é considerada a existência de uma unidade de coordenação central, responsável pela mediação das relações entre estágios. As interacções entre as entidades envolvidas são modeladas através de um conjunto de variáveis logísticas, agrupadas em sete subconjuntos (nível global, nível local, internas ao estágio, externas ao estágio, a montante do estágio, a jusante do estágio e externas ao modelo - variáveis exógenas).

É importante realçar que as organizações em rede diferem, em muitos aspectos, das organizações tradicionais, nomeadamente das organizadas hierarquicamente (Ching et al., 1993). Um dos factores diferenciadores mais significativos é o da coordenação entre as entidades participantes na organização. Um outro factor, igualmente importante, tem a ver com a relação estabelecida entre cada uma das entidades. No caso das organizações em rede, existem ligações dinâmicas e simultaneamente flexíveis, contrastando com as organizações hierárquicas em que as ligações entre entidades são rígidas e simultaneamente inflexíveis.

Para uma revisão da literatura sobre as organizações em rede, segundo uma perspectiva organizacional, sugere-se a leitura de (Ching et al., 1993). Nesse trabalho, é apresentado um modelo formal, com o propósito de caracterizar, do ponto de vista organizacional, as estruturas em rede, numa perspectiva centrada em torno da coordenação a partir de uma entidade nuclear ("core enterprise"). Os autores introduzem, no modelo desenvolvido, o conceito inovador de *reputação* como factor fulcral na actividade de coordenação. O factor *reputação* encapsula diversos atributos que caracterizam as entidades participantes na rede em actividades desenvolvidas anteriormente. É como que uma medida de avaliação global da fiabilidade da participação de uma dada entidade na rede. A actualização deste factor permite construir o que os autores chamam "perfil de desempenho" da entidade, contribuindo significativamente para a sua qualificação em actividades futuras. O mecanismo de coordenação desenvolvido baseia-se num modelo centrado em torno da empresa nuclear e envolve, num determinado instante de tempo, as entidades com os níveis de qualificação adequados (por exemplo, com determinado nível de reputação, com determinada capacidade, etc).

Em (Bhatnagar et al., 1993) é apresentada uma extensa revisão da literatura sobre modelos para coordenação multi-fábrica, de organizações verticais, com um conjunto de questões a considerar ao nível da coordenação. Os autores concluem que a grande maioria dos modelos analisados considera a coordenação a um nível estritamente funcional, e restringida a estágios únicos. No fabrico de produtos complexos, em que os processos produtivos são divididos por diversas unidades, o problema consiste na coordenação dos diversos planos de produção associados às diferentes unidades. Nesta situação, os autores argumentam que não existe uma visão unificada quanto à problemática da coordenação. São distinguidos dois grandes níveis de coordenação: um, geral (relativo à coordenação das decisões de diferentes funções, nomeadamente, fornecimento, produção, distribuição e inventário) e um, ao nível de cada entidade (unidade produtiva). Ainda segundo estes autores, um enquadramento adequado à problemática da coordenação multi-fábrica deve ter em conta um conjunto de questões fundamentais, tais como a "instabilidade" associada aos planos de produção e a sua propagação, a complexidade associada às questões de *lotsizing*, e a problemática associada aos níveis de segurança dos *stocks*. Para uma revisão muito actual da literatura acerca de modelos e formulações associados às questões de *lotsizing* recomenda-se o artigo de Drexl e Kimms (Drexl and Kimms, 1997).

Thomas e Griffin, num artigo recente (Thomas and Griffin, 1996), apresentam uma revisão da investigação centrada na área da coordenação da cadeia de fornecimento (*supply chain*) e, em particular, na modelação das entidades que a constituem. Os autores definem três categorias de coordenação ao nível operacional: coordenação entre comprador-vendedor, coordenação entre produção-distribuição e coordenação entre armazenamento-distribuição. Para cada uma das categorias, são apresentados modelos e formulações que procuram fundamentalmente resolver questões associadas à determinação de quantidades de produção e/ou escolha de modos de transporte. A um nível essencialmente estratégico, os autores apresentam uma revisão da investigação baseada em modelos de programação linear inteira e em modelos estocásticos. Concluem o artigo com uma enumeração de tópicos de investigação em aberto, parecendo clara a necessidade de desenvolver modelos e metodologias que possibilitem a decomposição eficiente das complexas estruturas em rede que caracterizam as cadeias de fornecimento, por forma a permitir, com fiabilidade, responder rapidamente aos pedidos dos clientes.

Uma revisão bibliográfica muito completa sobre planeamento da produção em estruturas multi-estágio é apresentada num artigo de Simpson e Erenguc (1996). Seguindo uma classificação de acordo com a complexidade dos problemas, os autores apresentam diversas abordagens e metodologias, e sugerem uma nova taxonomia para as heurísticas de planeamento de processos multi-estágios, nomeadamente através da sua segmentação em dois grupos: técnicas de passagem simples e técnicas de passagem múltiplas. Apesar de identificarem algumas semelhanças entre os problemas de planeamento de produção de sistemas multi-estágio e os sistemas de distribuição multi-estágio, e de considerarem que da sua fusão resultam a classe de problemas relacionados com a gestão da cadeia de fornecimento, os autores concluem que nenhuma das técnicas heurísticas apresentadas satisfaz os requisitos e a complexidade desta nova classe de problemas.

Finalmente refira-se uma abordagem para a resolução de problemas multi-estágio, multi-produto e envolvendo várias unidades de produção, apresentada em (Roux et al., 1999). O modelo desenvolvido pelos autores faz a integração dos problemas de planeamento e de escalonamento, através da adopção de uma decomposição a dois níveis (planeamento e escalonamento) e com a resolução alternada dos dois problemas. Basicamente, a resolução do problema de planeamento determina a quantidade a produzir. Este problema é modelado como um problema do tipo *lotsizing* multi-nível e com restrições de capacidade (MLCLP). O problema de escalonamento, sendo independente entre unidades de produção, é resolvido em paralelo e determina, para cada unidade produtiva, a sequência a considerar na produção.

5.2 Estruturação geral do problema

Os sistemas de produção multi-estágio envolvem a transformação de produtos e materiais, em diferentes unidades produtivas, inseridas numa cadeia logística organizada em diferentes estágios. Do ponto de vista global, uma cadeia logística pode ser vista como uma rede constituída por diversos nós (unidades fabris de produção e/ou montagem e unidades de armazenamento e/ou distribuição), interligados entre si e originando diferentes percursos (*'routings'*) no fabrico dos produtos (ver Figura 5.1). A existência de percursos alternativos depende, não só da tecnologia a utilizar, como também do tipo de produtos em causa, sendo as decisões quanto aos percursos a utilizar em cada caso,

também função da capacidade e disponibilidade dos recursos e, naturalmente, dos custos envolvidos.

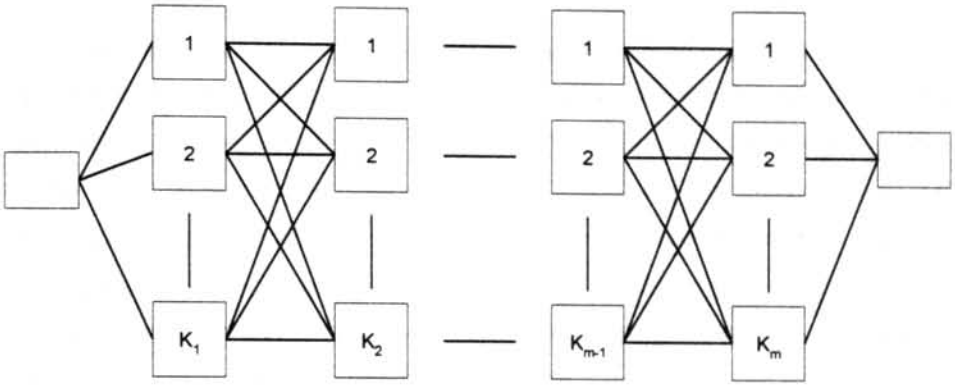


Figura 5.1 – Representação esquemática da estrutura produtiva

Neste contexto, considera-se que a organização do processo produtivo (segundo uma estrutura de precedência linear) segue um modelo produtivo do tipo *flow-shop* com um conjunto de características adicionais, nomeadamente a existência em cada estágio de unidades de processamento funcionalmente idênticas (em paralelo). No âmbito mais restrito dos problemas de escalonamento e de sequenciamento, este tipo de problemas é referido na literatura como *flexible flow shop* (FFS) (Blazewicz et al., 1994; Grasso and Fichera, 1993; Nowicki and Smutnicki, 1998). A sua resolução envolve, não só decisões de sequenciamento, mas também de afectação de recursos (tendo em conta as unidades em paralelo, alternativas).

O modelo FFS é constituído por vários estágios, organizados sequencialmente, podendo cada estágio ser constituído por múltiplas unidades de processamento. O fluxo de trabalho associado a uma determinada tarefa é unidireccional e geralmente envolve todos os estágios. Na literatura, é habitual formular os problemas FFS do seguinte modo:

Pretende-se processar um conjunto de n tarefas (j=1,...,n). Cada tarefa é constituída por m operações, a serem realizadas sequencialmente em máquinas, cada uma de um de m estágios (J=1,..., m). Em cada estágio, existem várias máquinas, alternativas. Para cada uma das tarefas são conhecidos os tempos de processamento de cada

uma das operações (em cada uma das máquinas), supostos determinísticos. Geralmente, consideram-se incluídos nos tempos de processamento os tempos de *setup* (tidos como independentes).

Uma solução do problema (plano) é normalmente avaliada segundo o seu "makespan" (isto é, pela data de conclusão da última operação). Mas naturalmente que, na prática, outros objectivos podem ser estabelecidos, a partir, por exemplo, da definição, para cada tarefa, de uma data desejável de entrega (*due-date*) ou, ainda, níveis de prioridade que exprimem a urgência relativa das tarefas. Outros pressupostos usualmente considerados são a permanente disponibilidade das máquinas para processamento, a imposição de que, em cada momento, uma operação só pode ser processada por uma máquina e cada máquina só pode processar uma operação, e ainda que as operações, uma vez iniciadas, não podem ser interrompidas (*preemption* não permitida).

5.2.1 Admissibilidade e avaliação das soluções

Uma solução admissível está associada a um plano (calendário) de execução das operações, que respeite as restrições do problema. Este plano corresponderá, naturalmente, a um dado sequenciamento ou escalonamento das operações.

Por outro lado, cada solução deverá poder ser avaliada segundo uma dada "função objectivo" (ou critério) e eventualmente segundo vários critérios.

Na prática, a dificuldade em resolver este tipo de problemas de forma eficiente é devida, no essencial, a dois motivos:

1. na selecção de soluções, utilizam-se critérios (múltiplos) que explicitam objectivos, muitas das vezes antagónicos;

2. o número de soluções cresce "exponencialmente" com a dimensão do problema, dado o seu carácter combinatório.

Como referido, a minimização do tempo total de fabrico (*makespan*) é talvez o critério de avaliação mais utilizado, em termos práticos. Contudo, outros critérios, directamente relacionados com as tarefas, podem igualmente ser considerados, nomeadamente, a minimização da soma dos atrasos ou do atraso máximo, como forma de medir a conformidade das tarefas de uma sequência com determinadas datas desejadas de entrega (*due dates*). Nalgumas situações, é ainda interessante considerar os atrasos

com "pesos" associados às tarefas (medindo, de alguma forma, a sua importância relativa).

Note-se que, no caso presente, se considerou, não o problema geral de escalonar um dado conjunto de encomendas na rede produtiva, mas antes, o problema (que corresponde melhor aos requisitos práticos) de planear, num dado momento, um pedido acabado de chegar ao sistema, e que naturalmente já vai encontrar este parcialmente carregado. Vai-se então procurar determinar o percurso óptimo, ao longo das unidades de produção, para satisfazer uma encomenda potencial.

Basicamente, para um pedido de um determinado produto, numa dada quantidade e com uma data de entrega, procura-se determinar, de forma optimizada, as janelas temporais (data início/data fim) e as quantidades a produzir por cada uma das unidades directamente envolvidas na satisfação do pedido. Em geral, a resolução do problema origina uma solução envolvendo todos os estágios produtivos e apenas uma unidade produtiva por estágio. Contudo, diversas situações particulares podem originar apenas o envolvimento de alguns dos estágios (planeamento parcial) ou a afectação no mesmo estágio de mais do que uma unidade produtiva. A Figura 5.2 apresenta graficamente os diferentes cenários possíveis, para um processo de produção constituído por quatro estágios e com um número variável de unidades por estágio.

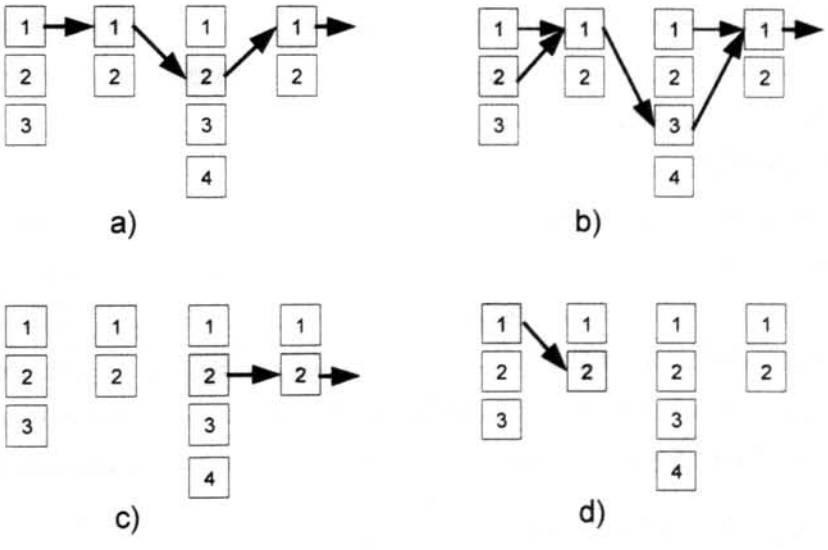


Figura 5.2 – Soluções possíveis (*routings*) para o problema

No problema em análise, podem considerar-se vários objectivos em simultâneo. Por um lado, procura-se minimizar os custos de fabrico associados às unidades produtivas envolvidas na satisfação da encomenda; por outro lado, procura-se igualmente minimizar os custos logísticos associados, nomeadamente custos de transporte entre unidades, assim como os custos de armazenamento. Adicionalmente, pretende-se que a data de entrega seja o mais próxima possível da data pedida, assim como se procura garantir que, para um mesmo estágio produtivo, se minimiza o número de unidades envolvidas.

5.2.2 O problema de inserção de ordens

Relativamente aos problemas de planeamento tradicionais envolvendo unicamente uma unidade produtiva, o caso presente é caracterizado por diversas dificuldades adicionais. Em primeiro lugar, o problema envolve várias unidades independentes de produção, e eventualmente, com diferentes estratégias de produção, não coordenadas. Por outro lado, não se conhece à partida o plano actual de produção, isto é, o processo de planeamento não parte de um plano director de produção global (*master production schedule*) para, de forma global e optimizada, gerar planos de produção locais. Na verdade, o problema consiste, em cada momento, no planeamento coordenado de uma única encomenda (ordem), por inserção no plano actual. Numa fase posterior, a pós-optimização do plano de produção resultante, nomeadamente através da re-afectação de recursos, poderá ser realizada com o recurso a metodologias de planeamento mais tradicionais e largamente exploradas na literatura.

Outra dificuldade adicional, que contribui significativamente para o aumento da complexidade do problema, tem a ver com o facto de não serem aqui definidos tempos de processamento (*lead-times*) determinísticos, associados às diferentes operações[31]. No caso presente, são considerados, em cada instante, o estado actual de cada unidade de produção, isto é, o planeamento é feito com verificação da capacidade real disponível. A abordagem seguida procura por isso efectuar o planeamento, não com base

[31] Os sistemas de planeamento tradicionais (tipo MRP-II) consideram os tempos de processamento um atributo do próprio componente a produzir. Assumem, por isso, que os *lead-time* são constantes e independentes do estado do sistema. Assim, na sua essência, estes sistemas estão a admitir permanentemente uma capacidade infinita, e consequentemente irrealista.

numa situação estática, mas antes considerando o estado dinâmico das entidades intervenientes, assim como o efeito da variabilidade da capacidade disponível com o "mix" de produtos, níveis de *WIP*, etc.

A inserção de uma ordem no plano (o mesmo é dizer, o planeamento da ordem) envolve três actividades fundamentais:

1. Divisão da ordem em subordens, de acordo com os estágios do processo de produção associado ao produto a que a ordem diz respeito;
2. Escalonamento das subordens, considerando restrições de âmbito estritamente local (capacidades de produção, por exemplo) e global (restrições de âmbito logístico);
3. Alocação das subordens às entidades da rede que participam na produção da ordem.

Tendo em vista a estruturação do problema com a finalidade de se poderem aplicar algoritmos de optimização, foram desenvolvidos dois modelos de programação matemática envolvendo objectivos múltiplos, modelos estes que serão apresentados na secção seguinte.

Nas formulações associadas a estes modelos, considerou-se dois níveis hierárquicos: um global, responsável pela determinação da solução "óptima" do problema; e um local, responsável pela actualização das capacidades e disponibilidades dos recursos existentes em cada unidade e pela avaliação da proposta de solução gerada no nível superior. Durante o processo de determinação de uma solução, o modelo recorre a um conjunto de parâmetros (custos, tempos de produção, etc.) considerados invariáveis. Após a determinação de uma solução, e verificando-se a confirmação de encomenda, cada modelo local procede à actualização dos parâmetros que eventualmente sejam susceptíveis de alteração (como, por exemplo, a capacidade disponível). A Figura 5.3 procura representar a interacção entre os níveis hierárquicos definidos.

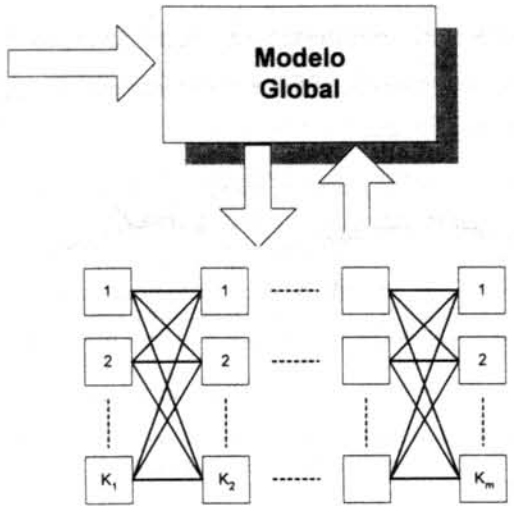


Figura 5.3 – Interacção entre modelo global e modelos locais

O modelo global procura resolver o problema de coordenação entre as entidades existentes na rede de produção e directamente envolvidas na satisfação da ordem. Os modelos locais modelam, num determinado instante de tempo, as entidades pertencentes à rede de produção.

As formulações apresentadas de seguida dizem respeito unicamente ao modelo global. Considera-se que os modelos locais, num determinado instante, se encontram num dado estado, caracterizado por um conjunto de parâmetros invariáveis enquanto decorre o processo de planeamento da ordem.

5.3 Modelos e casos particulares

Genericamente, o problema em análise consiste em planear encomendas num ambiente envolvendo diferentes unidades produtivas, inseridas numa cadeia logística organizada em diferentes estágios.

Nos modelos de programação matemática aqui considerados, pressupõe-se que uma solução envolve todos os estágios do processo produtivo. Não são exploradas situações de planeamento parcial, isto é, envolvendo apenas alguns dos estágios (ver Figura 5.2). Além disso, considera-se sempre que a quantidade a negociar por estágio é

constante para todos os estágios, não admitindo portanto a existência de eventuais *stocks* intermédios ou pontos de desacoplamento no próprio processo produtivo[32].

5.3.1 Notação geral

Índices

t	$(t = 1,...,T)$	período (T – horizonte discretizado de planeamento)
i	$(i = 1,...,I)$	estágio de produção
j,l		unidade de produção
k		modo de processamento

Parâmetros

q	quantidade pedida pelo cliente
dd	data de entrega pedida (*due date*)
ed	menor data de entrega possível (*earliest finish date*)
ld	última data de entrega possível (*latest finish date*)
C_{ijt}	capacidade de produção no período t para a unidade j do estágio i
a_{ij}	carga unitária da unidade j do estágio i (isto é, carga associada à produção de uma unidade de produto)
d_{ijl}^{move}	tempo de transporte entre as unidades j e l de estágios consecutivos (i e $i+1$)
c_{ijt}^{prod}	custo de produção no período t para a unidade j do estágio i
c_{ijtk}^{fab}	custo de produção no período t no modo k para a unidade j do estágio i
c_{ijl}^{move}	custo de transporte entre as unidades j e l de estágios consecutivos (i e $i+1$)
c_{ij}^{hold}	custo de armazenamento por unidade de tempo à entrada da unidade j no estágio i

[32] Caso do *die-bank* no caso do fabrico de semicondutores (ver secção 4.1.1).

d_{ijt}^{min}	tempo de processamento mínimo na unidade *j* do estágio *i* para processar a respectiva operação, na condição de se iniciar no período t
d_{ijt}^{max}	tempo de processamento máximo na unidade *j* do estágio *i* para processar a respectiva operação, na condição de se iniciar no período t
J_i	conjunto das unidades produtivas do estágio i
K_{ij}	conjunto dos modos de processamento para a unidade j do estágio i
W_1, W_2	factores de penalização

5.3.2 Modelo 1

5.3.2.1 Considerações gerais

Neste modelo, consideram-se os seguintes pressupostos:

- Para cada unidade produtiva e em cada período, é conhecido o 'tempo mínimo/máximo de resposta', ou seja, é conhecido, no caso da produção se iniciar nesse período, o intervalo de tempo possível para realizar a produção.
- Considera-se que o custo de produção é constante durante todos os períodos de processamento, dependendo contudo do instante em que a produção se inicia e da duração (tempo de processamento).
- Para cada estágio, as unidades que o constituem têm diferentes modos de execução, originando diferentes tempos de processamento.
- Não é permitida a participação, por estágio, de mais do que uma unidade de produção, na satisfação dum dado pedido.
- A quantidade a produzir é constante e previamente definida.
- Não existem *stocks* intermédios (entre estágios).

- Cada unidade de produção pode receber *'inputs'* de qualquer das unidades precedentes em que se verifique produção.
- O transporte entre estágios inicia-se imediatamente após o fim de processamento; considera-se que o tempo de transporte entre unidades é invariável no tempo.

5.3.2.2 Variáveis de decisão

$$\mu_{ijtk} = \begin{cases} 1 & \text{se a produção na unidade } j, \text{ estágio } i, \text{ se inicia no período } t, \text{ modo } k \\ 0 & \text{se não} \end{cases}$$

Variáveis auxiliares

d_{ij}^{proc}	tempo de processamento (duração) na unidade j do estágio i
t_{ij}^{start}	início da produção na unidade j do estágio i
t_{ij}^{finish}	fim da produção na unidade j do estágio i

5.3.2.3 Restrições

$$\forall i, \forall j \in J_i \quad d_{ij}^{proc} - \sum_t \sum_k d_{ijt}^{\max} \cdot \mu_{ijtk} \leq 0 \tag{1}$$

$$\forall i, \forall j \in J_i \quad \sum_t \sum_k d_{ijt}^{\min} \cdot \mu_{ijtk} - d_{ij}^{proc} \leq 0 \tag{2}$$

$$\forall i, \forall j \in J_i \quad d_{ij}^{proc} = \sum_t \sum_k k \cdot \mu_{ijtk} \tag{3}$$

$$\forall i, \forall j \in J_i \quad t_{ij}^{start} = \sum_t \sum_k t \cdot \mu_{ijtk} \tag{4}$$

$$\forall i, \forall j \in J_i \quad t_{ij}^{finish} = \sum_t \sum_k (t+k) \cdot \mu_{ijtk} \tag{5}$$

$$\forall i \quad \sum_{j \in J_i} \sum_{t} \sum_{k} \mu_{ijtk} \leq 1 \tag{6}$$

$$i = 1,\ldots, I-1, \forall j \in J_i, \forall l \in J_{i+1}$$

$$t_{ij}^{finish} + d_{ijl}^{move} \cdot \sum_{t} \sum_{k} \mu_{ijtk} - t_{i+1,l}^{start} \leq 0 \tag{7}$$

$$i = I, \forall j \in J_i \quad t_{ij}^{finish} \leq ld \tag{8}$$

$$i = I, \forall j \in J_i \quad t_{ij}^{finish} \geq ed \tag{9}$$

$$\forall i, \forall j \in J_i, \forall t, \forall k \quad \mu_{ijtk} \in \{0,1\} \tag{10}$$

As restrições (1) e (2) garantem que o tempo de processamento da operação, a iniciar-se num determinado período, se encontra no intervalo definido pelos parâmetros externos respectivos.

As restrições (3), (4) e (5) definem as variáveis auxiliares para o início, o fim e o tempo do processamento (duração).

A restrição (6) garante que em cada estágio, uma única unidade produtiva participa no processamento do pedido.

A restrição (7) assegura a satisfação das relações de precedência entre estágios.

As restrições (8) e (9) asseguram que a data de conclusão se encontra dentro do intervalo definido entre os limites *earliest finish date* e *latest finish date.*

5.3.2.4 Funções Objectivo

O problema é aqui encarado como tendo dois objectivos. Procura-se minimizar os custos de produção, assim como os desvios relativamente à data de entrega pedida.

Minimizar custos de produção

$$\min \sum_{i} \sum_{j \in J_i} \sum_{t} \sum_{k} c_{ijtk}^{fab} \cdot \mu_{ijtk}$$

Minimizar desvios relativamente à data de entrega

$$\min \quad \sum_{j \in J_l} \left(\max\left\{ W_1 \cdot \left(dd - t_{lj}^{finish} \right), 0 \right\} + \max\left\{ W_2 \cdot \left(t_{lj}^{finish} - dd \right), 0 \right\} \right)$$

5.3.3 Modelo 2

5.3.3.1 Considerações gerais

Neste segundo modelo, consideraram-se os seguintes pressupostos:

- Para cada unidade produtiva, é conhecida a capacidade em cada período, assim como o custo unitário de produção.
- Não existem *stocks* intermédios (entre estágios).
- Por estágio, pode participar mais do que uma unidade produtiva, na satisfação do pedido.
- Cada unidade de produção pode receber *'inputs'* de qualquer das unidades precedentes em que se verifique produção.
- Nenhuma operação pode ser iniciada numa unidade, antes de todas as que a precedem estarem concluídas.
- O transporte entre unidades inicia-se imediatamente após o fim do processamento. Considera-se que o tempo de transporte entre unidades é invariável no tempo.

5.3.3.2 Variáveis de decisão

x_{ijt}	quantidade a produzir na unidade *j* do estágio *i*, no período *t*
t_{ij}^{start}	início da produção na unidade *j*, do estágio *i*
t_{ij}^{finish}	fim da produção na unidade *j* dos estágio *i*

Variáveis auxiliares

$$\delta_{ijt} = \begin{cases} 1 \text{ se existe produção no período } t \text{ na unidade } j, \text{ estágio } i \\ 0 \text{ se não} \end{cases}$$

$$\gamma_{ij} = \begin{cases} 1 \text{ se existe produção na unidade } j, \text{ estágio } i \\ 0 \text{ se não} \end{cases}$$

5.3.3.3 Restrições

$$i = I, \forall j \in J_i \qquad t_{ij}^{finish} \leq ld \tag{1}$$

$$i = I, \forall j \in J_i \qquad t_{ij}^{finish} \leq ed \tag{2}$$

$$\forall i, \forall j \in J_i, \forall t \qquad t_{ij}^{finish} - t \cdot \delta_{ijt} \geq 0 \tag{3}$$

$$\forall i, \forall j \in J_i, \forall t \qquad t_{ij}^{finish} - t \cdot \delta_{ijt} \leq 0 \tag{4}$$

$$\forall i, \forall j \in J_i \qquad \sum_t \delta_{ijt} - \left(t_{ij}^{finish} - t_{ij}^{start}\right) \geq 0 \tag{5}$$

$$\begin{aligned} &i = 1..I-1, \forall j \in J_i, \forall l \in J_{i+1} \\ &\quad t_{ij}^{finish} + d_{ijl}^{move} \cdot \gamma_{ij} \leq t_{i+1,l}^{start} + \left(1 - \gamma_{i+1,l}\right) \cdot M \end{aligned} \tag{6}$$

$$\forall i \qquad \sum_{j \in J_i} \sum_t x_{ijt} \geq q \tag{7}$$

$$\forall i, \forall j \in J_i \qquad a_{ij} \cdot \sum_t x_{ijt} - \sum_t C_{ijt} \cdot \delta_{ijt} \leq 0 \tag{8}$$

$$\forall i, \forall j \in J_i, \forall t \qquad x_{ijt} \leq C_{ijt} \cdot \delta_{ijt} \tag{9}$$

$$\forall i, \forall j \in J_i \quad \sum_t x_{ijt} \leq M \cdot \gamma_{ij} \tag{10}$$

$$\forall i, \forall j \in J_i, \forall t \quad x_{ijt} \geq 0 \tag{11}$$

$$\forall i, \forall j \in J_i, \forall t \quad \delta_{ijt} \in \{0,1\} \tag{12}$$

$$\forall i, \forall j \in J_i \quad \gamma_{ij} \in \{0,1\} \tag{13}$$

As restrições (1) e (2) asseguram que a data de conclusão se encontra dentro do intervalo definido entre os limites *earliest finish date* e *latest finish date*.

As restrições (3), (4) e (5) garantem que a produção fica definida em períodos de tempo adjacentes.

A restrição (6) assegura a satisfação das relações de precedência entre estágios.

A restrição (7) garante que, em cada estágio, se produz a quantidade pedida.

A restrição (8) assegura que a quantidade produzida respeita a capacidade disponível.

5.3.3.4 Funções Objectivo

Neste caso, consideram-se, em simultâneo, três objectivos. Procura-se minimizar os custos de produção na satisfação do pedido. Os custos logísticos associados, nomeadamente custos de transporte e de armazenamento intermédio, devem igualmente ser minimizados. Procura-se igualmente penalizar os desvios da data de entrega, relativamente à data pedida.

Minimizar custos de produção

$$\min \sum_i \sum_{j \in J_i} \sum_t c_{ijt}^{prod} \cdot x_{ijt}$$

Minimizar custos de transporte

$$\min \sum_{i=1..I-1} \sum_{j \in J_i} \sum_{l \in J_{i+1}} c_{ijl}^{move} \cdot \gamma_{ij} \cdot \gamma_{i+1,l}$$

Minimizar custos de armazenamento entre estágios

$$\min \sum_{i=1..I-1} \sum_{l \in J_{i+1}} \sum_{j \in J_i} c_{ij}^{hold} \cdot \left(t_{i+1,j}^{start} + \left(1 + \gamma_{i+1,l}\right) \cdot M - \left(t_{ij}^{finish} + d_{ijl}^{move} \cdot \gamma_{ij} \right) \right)$$

Minimizar desvios relativamente à data de entrega

$$\min \sum_{j \in J_I} \left(\max\left\{ W_1 \cdot \left(dd - t_{Ij}^{finish} \right), 0 \right\} + \max\left\{ W_2 \cdot \left(t_{Ij}^{finish} - dd \right), 0 \right\} \right)$$

5.4 Meta-heurísticas

Os problemas de optimização considerados neste trabalho (e em particular o problema de coordenação atrás descrito) podem enquadrar-se na classe muito geral dos problemas de Optimização Combinatória (OC). Um problema diz-se de OC se o conjunto de soluções admissíveis é discreto (conjunto finito ou infinito mas enumerável de alternativas), como resultado, por exemplo, da afectação ou permutação de entidades (donde o carácter "combinatório" do problema).

Na sua forma mais básica, um problema de optimização envolve a minimização ou maximização de uma função objectivo (que mede a vantagem ou desvantagem atribuída pelo decisor a cada solução do problema), satisfazendo um conjunto de restrições dependentes directamente da natureza do problema tratado.

Exemplos clássicos de problemas de OC, de formulação "relativamente" simples, tais como o Problema do Caixeiro Viajante (*travelling salesman problem*), são paradigmáticos da classe dos problemas NP-difíceis (*NP-hard*), cuja dificuldade é "justificada" pela teoria da Complexidade Computacional (Garey and Johnson, 1979), (Nemhauser and Wolsey, 1988). A resolução destes problemas obriga a um compro-

misso entre tempo de resolução e qualidade das soluções obtidas, isto é, entre eficiência e eficácia do algoritmo, sendo que a obtenção de soluções óptimas está em geral fora de questão (já que para tal seria necessário recorrer a algoritmos "exponenciais" baseados, por exemplo, em processos de enumeração implícita). Deste compromisso resultam diferentes abordagens algorítmicas, nomeadamente, algoritmos de optimização ou exactos, e algoritmos aproximados ou heurísticos. Para uma classificação das diferentes abordagens algorítmicas a problemas de Optimização Combinatória sugere-se, por exemplo, a leitura de (Muller-Merbach, 1981).

Naturalmente que a elevada dimensão dos problemas reais, a sua complexidade assim como a sua natureza combinatória conduziram ao desenvolvimento de um grande número de modelos e técnicas heurísticas para a sua resolução. Nos últimos anos, têm sido usadas com sucesso em muitos problemas de Optimização Combinatória, técnicas baseadas em procedimentos gerais de carácter heurístico, que consistem em processos recursivos ou iterativos de pesquisa local, com algum grau de aleatorização. Porque são métodos heurísticos de pesquisa genérica e aplicáveis a grandes conjuntos de problemas de optimização, recebem a designação de *meta-heurísticas*. Estes métodos pressupõem sempre a existência de um outro algoritmo ou heurística que, a um nível inferior e dependente do problema concreto que se tenta resolver, gera soluções admissíveis para o problema e/ou determina o valor da função objectivo de uma dada solução.

As meta-heurísticas (*simulated annealing*, *taboo search*, etc) podem ser considerados métodos de pesquisa local em que se pretende, de forma iterativa e seguindo uma determinada estratégia de pesquisa, minimizar uma função de custo sobre um espaço de soluções. No caso geral, o processo inicia-se com uma solução admissível (arbitrária ou não), e em cada iteração, uma nova solução é escolhida na *vizinhança* da solução actual. A *vizinhança* de uma solução é um subespaço do espaço de soluções admissíveis, alcançável a partir dessa solução, por uma transformação elementar. A sequência de soluções de um problema, resultante do processo de avançar de solução em solução, constitui o que se convencionou designar por *trajectória* no espaço de soluções.

Estas técnicas de pesquisa apresentam um conjunto de características que as tornam particularmente interessantes na resolução de problemas de Optimização Combinatória (Viana, 1997) (Dias, 1995):

1. *flexibilidade* – permitem a adaptação com facilidade a diferentes estruturas do problema;

2. *robustez* – os resultados obtidos são, em geral, pouco sensíveis a variações nas características do problema;

3. *aleatoriedade* – apresentam um comportamento não determinístico, ou seja, utilizam procedimentos que envolvem alguma aleatoriedade (permitindo, assim, diversificar o processo de pesquisa);

4. *adaptabilidade* – permitem a modificação de parâmetros segundo a evolução do processo de pesquisa;

5. *paralelismo* – o conceito de conjunto de soluções, presente nalgumas das técnicas apresenta um paralelismo implícito, que pode facilmente ser explorado sob o ponto de vista computacional.

A eficiência e a eficácia das meta-heurísticas apresentam, no caso geral, uma forte dependência dos seus parâmetros, obrigando consequentemente a um trabalho de afinação, a partir da análise dos resultados obtidos através de experiências que podem envolver um elevado esforço de concepção e realização (Oliveira, 1995). Igualmente importante é a forma como o problema é estruturado, nomeadamente ao nível da caracterização e representação das soluções, assim como os procedimentos utilizados na geração das soluções e na definição da estratégia de obtenção de soluções vizinhas.

Nos últimos anos têm sido obtidos excelentes resultados em inúmeros problemas práticos, de Optimização Combinatória, pelo recurso às seguintes meta-heurísticas (normalmente "inspiradas" em domínios do conhecimento que nada têm a ver com a Optimização):

- "*simulated annealing*" – procura "reproduzir" no âmbito da optimização combinatória, fenómenos de termodinâmica e metalurgia, nomeadamente, quanto à evolução dos sistemas físicos para estruturas de energia mínima;

- *algoritmos genéticos* – inspiram-se na teoria da evolução e selecção natural das espécies e trabalham com "populações" de soluções;

- *pesquisa tabu* – inspira-se nos processos de aprendizagem, tendo subjacente o conceito de memória e explorando, em cada iteração, informação relativa a iterações anteriores;
- *redes neuronais* – resultam da analogia com o funcionamento neuronal do cérebro humano.

Uma descrição pormenorizada destas técnicas pode ser encontrada, por exempo, em (Reeves, 1995) e em (Pirlot, 1994). Refira-se que têm recentemente sido desenvolvidas muitas outras meta-heurísticas, por exemplo a partir de "hibridização" dos algoritmos referidos.

Uma estratégia de pesquisa baseada em meta-heurísticas é caracterizada, no caso geral, pela sucessão de forma progressiva de uma fase inicial de *diversificação* para uma fase posterior de *intensificação* (nalgumas versões, explora-se a alternância dos dois tipos de fases). A fase de *diversificação* explora, com um maior grau de aleatoriedade, o espaço de soluções, permitindo, em maior escala, que soluções piores do que a actual sejam consideradas, na perspectiva de posteriormente se virem a descobrir novas soluções melhores. Procura-se, assim, evitar cair em *mínimos locais*. A fase de *intensificação* é caracterizada pela crescente relutância à aceitação de soluções piores do que as obtidas até um dado momento, possibilitando assim que a pesquisa seja confinada a uma região do espaço de soluções potencialmente mais promissora. Assim, à medida que o processo de pesquisa avança no espaço de soluções, o algoritmo torna-se mais selectivo.

Nas secções seguintes, serão apresentados métodos de resolução do problema de planeamento de encomendas, baseados nas meta-heurísticas de "*simulated annealing*" e em *algoritmos genéticos*. Havendo um conjunto de aspectos comuns às duas abordagens, ir-se-á, de seguida descrever alguns desses aspectos comuns, nomeadamente, no que respeita à geração de soluções iniciais.

Com a finalidade de uniformizar e formalizar alguns dos conceitos geralmente considerados nas abordagens heurísticas aos problemas de Optimização Combinatória, começa por fazer-se uma breve descrição da estratégia de pesquisa local utilizada na resolução destes problemas.

5.4.1 Estratégia de pesquisa local

No caso geral, pode-se encarar um problema de Optimização Combinatória como a minimização de uma função custo F sobre um espaço de soluções finito S. Assim, considerando um espaço de soluções admissíveis S e uma função objectivo $F: S \rightarrow \Re$, pretende-se encontrar uma solução $x^* \in S$, tal que $\forall y \in S, \quad F(x^*) \leq F(y)$.

Um algoritmo de pesquisa local percorre uma "trajectória" em S, baseada no movimento de uma solução actual para outra (vizinha da primeira) e de acordo com determinadas regras (não necessariamente determinísticas). Pressupõe por isso a definição de subespaços de soluções sobre S, isto é, pressupõe a definição da estrutura da vizinhança em S, de tal modo que para cada $x \in S$ esteja associado um subconjunto $N(x) \subseteq S$ designado por *vizinhança* de x. Isto é, uma vizinhança é constituída pelo conjunto de soluções que se podem atingir a partir de x, realizando movimentos elementares específicos. As soluções vizinhas da solução x são todos os elementos existentes na sua vizinhança $N(x)$. Por convenção, nenhuma solução é vizinha de si própria, isto é, $\forall x \in S, \quad x \notin N(x)$.

É habitual definir-se igualmente o conceito de subvizinhança de x, como qualquer subconjunto $SN_i(x)$ de dimensão i, tal que $SN_i(x) \subseteq N(x) \wedge |SN_i(x)| = i$ (normalmente $SN_i(x)$ é gerado, escolhendo aleatoriamente i elementos de $N(x)$).

No caso geral, o processo de optimização é dificultado pela existência de funções objectivo não lineares e com múltiplos óptimos (mínimos) locais. Neste contexto, uma solução x^* é considerada um mínimo local se na sua vizinhança não existir uma solução melhor, isto é, se $\forall y \in N(x^*), \quad F(x^*) \leq F(y)$.

Os métodos de pesquisa local começam por seleccionar uma solução inicial pertencente ao espaço de soluções e em cada iteração n do processo de pesquisa escolhem uma nova solução x_{n+1} pertencente à vizinhança da solução actual x_n, isto é, $x_{n+1} \in N(x_n)$.

Uma estratégia de selecção de uma nova solução x_{n+1} na vizinhança de x_n consiste em escolher o "melhor" elemento da vizinhança. Numa variante desta estratégia, escolhe-se o primeiro melhor vizinho, isto é, são geradas sucessivamente soluções vizinhas e escolhe-se a primeira que apresentar um valor da função objectivo melhor do que o valor da função objectivo da solução actual. Métodos de pesquisa

local baseados nestas estratégias de selecção apresentam como desvantagem o facto de terminarem obrigatoriamente no primeiro mínimo local que encontram. É precisamente esta limitação que as meta-heurísticas procuram ultrapassar.

5.4.2 Geração de soluções iniciais

Uma solução do problema em estudo (inserção de uma ordem no plano actual) pode, de uma forma sintética, ser encarada como um conjunto de janelas temporais, não sobrepostas, uma por cada estágio do processo de produção, e com a restrição adicional de respeitarem a sequência de precedência "linear" do processo produtivo. Cada janela temporal é delimitada pelas datas possíveis de início e de fim, associadas ao processamento nesse estágio, para a satisfação da encomenda submetida. Considera-se aqui um horizonte temporal discretizado, constituído por intervalos iguais e de resolução correspondente a um dia (períodos), e com um comprimento imposto pela data de entrega associada ao pedido (encomenda).

BIBLIOTECA F.E. PORTO

Do ponto de vista computacional, uma solução é representada como um vector de números inteiros, de comprimento igual a duas vezes o número de estágios. Uma janela temporal é representada por dois elementos consecutivos do vector. Cada elemento inteiro do vector representa o número de períodos (dias) a partir da data actual (tomada como referência, ou seja, zero). A Figura 5.4 representa de forma gráfica a relação entre a representação vectorial de uma solução e o horizonte temporal discretizado. Note-se que, como referido, assim definida, uma solução é uma representação sucinta da realidade (por exemplo, "ainda" não contém informação quanto às unidades reais que vão estar alocadas à produção).

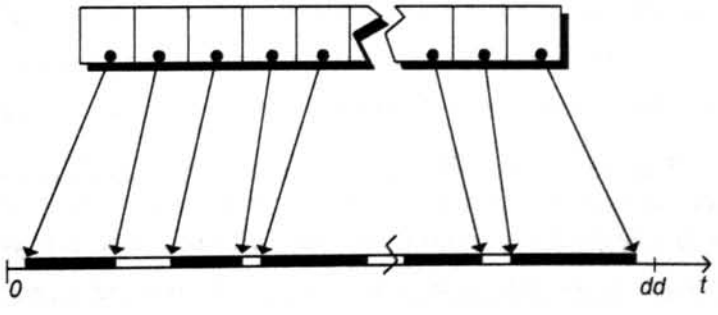


Figura 5.4 – Representação de uma solução

O exemplo seguinte ilustra numericamente uma possível solução inicial para um processo constituído por 4 estágios.

índice	0	1	2	3	4	5	6	7
solução[]	5	21	24	33	35	40	42	50
estágio	1		2		3		4	

As datas de início e de fim associadas a cada estágio são obtidas directamente a partir do vector solução[], como se exemplifica a seguir para um *estágio i*:

Data início = solução[i]

Data fim = solução[i+1]

Na determinação da solução, para além de se considerarem as restrições associadas a cada estágio (por exemplo, capacidades e quantidades disponíveis, custos, etc.), são consideradas as restrições logísticas na interligação entre estágios, nomeadamente o tempo e a disponibilidade de transporte. Implícita a cada solução estão ainda associadas as unidades de produção envolvidas na satisfação da encomenda. A determinação, para cada estágio, das unidades que participam na satisfação da encomenda constitui um problema adicional, por estágio, de alocação de unidades de produção (no caso naturalmente de existirem unidades alternativas no mesmo estágio).

BIBLIOTECA FE PORTO

No caso geral, uma solução admissível para o problema obedece às seguintes restrições fundamentais:

- para qualquer um dos estágios, a data de conclusão é posterior à data de início;
- são satisfeitas as relações de precedência entre estágios, de acordo com o processo produtivo, isto é, para estágios consecutivos, a data de conclusão do estágio precedente é anterior à data de início do estágio subsequente;
- é respeitado o tempo de transporte admissível entre estágios consecutivos;
- a data de conclusão do processamento no último estágio envolvido na satisfação da encomenda, deve ser anterior ao limite superior para a data de entrega e posterior ao seu limite inferior.

Estas restrições podem ser representadas pelas inequações seguintes.

Notação

$i \quad (i = 1,\ldots, I)$	estágio de produção
t_i^{start}	inicio da produção no estágio i
t_i^{finish}	fim da produção no estágio i
t_i^{move}	tempo de transporte entre os estágios i e $i+1$ consecutivos
ed	menor data de entrega possível (*earliest finish date*)
ld	maior data de entrega possível (*latest finish date*)
dd	data de entrega pedida (*due date*)

Restrições

$$\forall i \quad t_i^{start} \le t_i^{finish} \tag{1}$$

$$i = 1,\ldots, I-1 \quad t_i^{finish} + t_{i,i+1}^{move} \le t_{i+1}^{start} \tag{2}$$

$$t_I^{finish} \le ld \tag{3}$$

$$t_I^{finish} \ge ed \tag{4}$$

Descrevem-se, de seguida, três heurísticas para geração de soluções iniciais. A primeira heurística considera apenas algumas das restrições apresentadas. São geradas um conjunto de janelas temporais (datas de início e fim), uma por cada estágio, de forma a não se intersectarem, e com a restrição adicional de respeitarem a sequência de precedência linear do processo produtivo. A segunda heurística, para além de considerar todas as restrições apresentadas, utiliza na geração dos tempos de processa-

mento, afectos a cada estágio, valores médios estimados. A terceira heurística é similar à segunda, mas permite gerar soluções iniciais com datas de conclusão diferentes da *due-date*.

5.4.2.1 Geração de soluções iniciais - Heurística I

Esta heurística gera de forma aleatória uma janela temporal para cada um dos estágios do processo produtivo. Garante unicamente as restrições de precedência entre datas de início e de fim de processamento e entre estágios consecutivos. Basicamente, por cada estágio é necessário gerar duas datas, correspondendo ao início e ao fim de processamento da encomenda, nesse estágio. O conjunto de datas ordenadas por ordem crescente constitui uma solução admissível para o problema.

A estrutura básica da heurística implementada está descrita no Algoritmo 5.1.

```
algoritmo gera_solução_I

   solução ← vector_inteiros(2*numEstágios);

   para  i=1 até  2*numEstágios
      solução[i-1] ← NumeroAleatório(0, dueDate);
   fimPara
   ordena_vector (solução);
   retorna solução;
fim
```

Algoritmo 5.1 – Heurística aleatória para geração de uma solução admissível (H-I)

Após a "normalização" da *due date* da encomenda para o esquema discretizado considerado, a heurística começa por gerar um conjunto de números aleatórios inteiros, num total de duas vezes o número de estágios existentes, e compreendidos entre a data actual e a data de entrega associada à encomenda (*due date*). No fim, procede-se à ordenação crescente dos números inteiros gerados, sendo o vector resultante, uma solução admissível.

5.4.2.2 Geração de soluções iniciais - Heurística II

Esta heurística considera todas as restrições associadas a uma solução admissível e enumeradas anteriormente. Pressupõe além disso que o tempo de processamento num determinado estágio apresenta o comportamento de uma variável *gaussiana* (distribuição normal N[μ,σ]) em que o valor médio (μ) e o valor de variância (σ) são função do tipo de produto (p) e da quantidade envolvida (q). Para efeitos de implementação computacional, a distribuição normal foi obtida a partir de uma distribuição uniforme entre 0 e 1, aplicando o método de Box-Muller (Guimarães and Cabral, 1997), sendo implementada a partir da seguinte expressão:

$$P_{proc} = \mu[estagio, p, q] + \sigma[estagio, p, q] \cdot \sqrt{-2 \cdot \ln(u1)} \cdot \cos(2\pi \cdot u_2) \quad (5)$$

em que u_1 e u_2 são variáveis aleatórias com distribuição uniforme entre 0 e 1.

Com a implementação desta heurística para geração de uma solução inicial, procura-se garantir que os tempos de processamento se aproximam dos valores obtidos em condições normais de funcionamento. Para um determinado estágio `i`, a função P_{roc}`(i, μ[p,q], σ[p,q])` retorna o tempo de processamento associado ao processamento de uma quantidade `q` de um produto `p`.

A estrutura básica da heurística implementada está descrita no Algoritmo 5.2. A heurística começa por determinar a janela temporal associada ao último estágio, seguindo uma estratégia de geração regressiva de datas, estágio a estágio, até ao inicial. Na geração das janelas temporais, é considerado igualmente o tempo mínimo de transporte ($t_{move}[i]$) entre um estágio i e o estágio subsequente.

```
algoritmo gera_solução_II

 solução ← vector_inteiros(2*numEstágios);

 i ← 2*(ultimoEstagio-1);
 solução[i+1] ← DueDate;
 solução[i] ← solução[i+1]-P_roc(i/2+1, μ[p,q], σ[p,q]);
 i ← i-2;

 enquanto i≥0
  solução[i+1] ← solução[i+2]-t_move[i];
  solução[i] ← solução[i+1]
               -P_roc(i/2+1, μ[p,q], σ[p,q]);
  i ← i-2;
 fimEnquanto

 retorna solução;
fim
```

Algoritmo 5.2 – Heurística para geração de uma solução admissível (H-II)

5.4.2.3 Geração de soluções iniciais - Heurística III

Esta heurística é semelhante à anterior, no que respeita ao facto de recorrer a informação estatística na geração das datas de início e de fim, a afectar a cada estágio. A diferença fundamental reside na geração da data de conclusão associada à solução. Em vez de se considerar sempre a *due-date* como a data de conclusão associada à solução gerada, existe a possibilidade de se gerarem diferentes datas de conclusão, inferiores ou superiores e com diferentes intervalos de variação, de uma forma caracterizada probabilisticamente.

Esta heurística procura garantir diferentes pontos de partida, para a corrida do algoritmo de pesquisa, no que respeita à exploração de soluções com data de conclusão diferente da pedida (*due-date*). Esta situação é particularmente interessante em situações em que o último estágio não respeita a *due-date*, por falta de capacidade disponível.

algoritmo gera_solução_III

```
solução ← vector_inteiros(2*numEstágios);
i ← 2*(ultimoEstagio-1);
p=numAleatório(0,1);
se p<0.5
 então
   dataFim=DueDate;
 senão
   se p<0.85
     entao
       dataFim=numAleatório(0.9*DueDate,latestDate);
     senão
       se p<0.95
         entao
           dataFim=numAleatório(0.7*DueDate,
                        0.9*DueDate);
         senão
           dataFim=numAleatório(1,0.7*DueDate);
       fimSe
   fimSe
fimSe
solução[i+1] ← dataFim;
solução[i] ← solução[i+1]-Proc(i/2+1, μ[p,q], σ[p,q]);
i ← i-2;
enquanto i≥0
 solução[i+1] ← solução[i+2]-tmove[i];
 solução[i] ← solução[i+1]-Proc(i/2+1, μ[p,q], σ[p,q]);
 i ← i-2;
fimEnquanto
retorna solução;
fim
```

Algoritmo 5.3 – Heurística para geração de uma solução admissível (H-III)

5.4.3 Avaliação das soluções

Após a geração de uma solução, é necessário efectuar a respectiva avaliação, tendo em conta o conjunto de objectivos considerados. No caso presente, a aplicação de técnicas de optimização ou de pesquisa a um problema eminentemente distribuído, requereu a necessidade de se efectuar uma decomposição algorítmica, em diferentes níveis. Consideraram-se, no caso presente, três níveis principais.

O nível superior é responsável pela geração de soluções admissíveis e pela estratégia de pesquisa, ao longo desse espaço de soluções. O nível intermédio é responsável pela decomposição das soluções, geradas no nível superior, em tantas subsoluções quanto os estágios envolvidos. Neste nível, é implementado o mecanismo iterativo de negociação (envio de propostas e recepção de ofertas) com os modelos de capacidade locais. É feita igualmente a integração das ofertas recebidas e a consequente avaliação global. O nível inferior, composto por tantas instâncias dos modelos de capacidade quanto o número de entidades envolvidas, avalia as propostas submetidas, gerando as ofertas correspondentes.

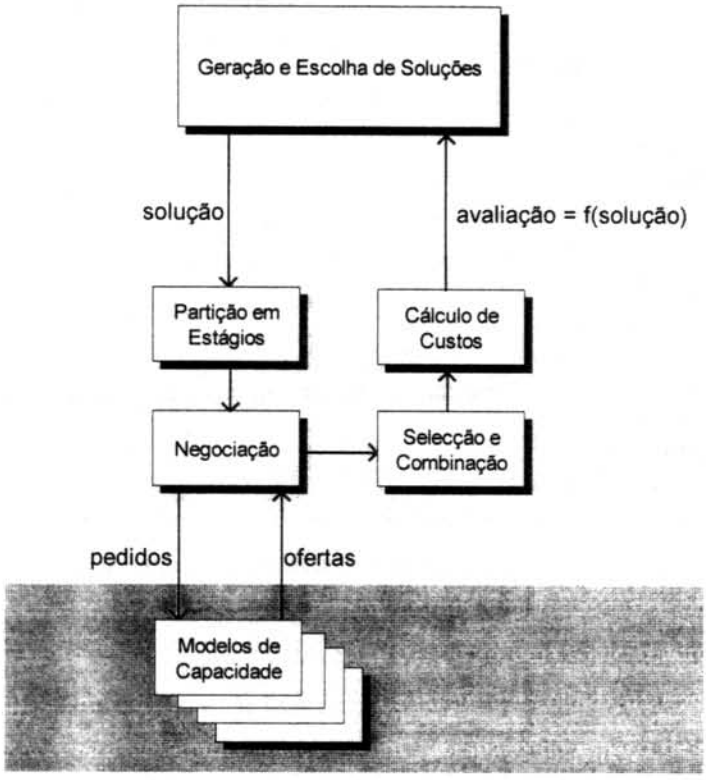


Figura 5.5 – Ciclo de geração e avaliação de uma solução

A Figura 5.5 apresenta a estrutura algorítmica, implementada para a geração e a avaliação de uma solução.

5.4.4 Exemplo ilustrativo

O exemplo seguinte ilustra numericamente o processo de avaliação de uma solução. Supõe-se um total de quatro estágios, com onze unidades de produção distribuídas da seguinte forma: quatro no estágio 1, duas nos estágios 2 e 4, e três no estágio 3 (ver Figura 5.6). U_{ij} – representa a unidade j pertencente ao estágio i.

Na avaliação da solução, são considerados unicamente os custos de produção e os custos de transporte e de armazenamento entre estágios. Não se considera custos de penalização decorrentes do não cumprimento da *due-date*.

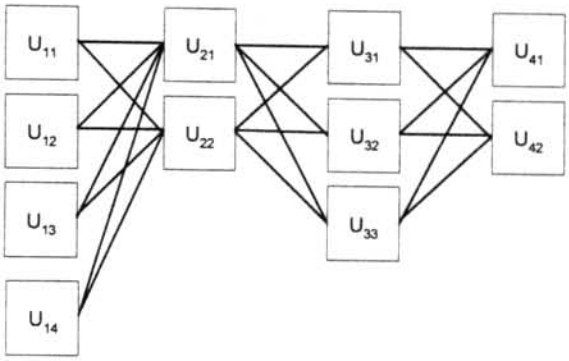


Figura 5.6 – Configuração do processo produtivo

Pretende-se avaliar a seguinte solução (conjunto das datas de início e de fim para cada um dos quatro estágios):

10	30	44	64	68	82	85	105

Consideram-se os valores de tempo de transporte, de custo de transporte e de custos de armazenamento apresentados nas tabelas seguintes.

5.4.4.1 Tempos e custos de transporte; custos de armazenamento

Esta matriz representa numa forma tabular, a informação logística considerada na função de avaliação (tempo de transporte e custo associado unicamente entre estágios consecutivos).

Cada elemento da matriz contém o tempo e o custo de transporte (em unidades de tempo e de custo), para a quantidade associada à encomenda em negociação.

	U_{21}	U_{22}	U_{31}	U_{32}	U_{33}	U_{41}	U_{42}
U_{11}	2,2	2,1	-	-	-	-	-
U_{12}	2,2	2,2	-	-	-	-	-
U_{13}	3,3	3,2	-	-	-	-	-
U_{14}	2,2	2,3	-	-	-	-	-
U_{21}	-	-	2,2	2,2	2,3	-	-
U_{22}	-	-	2,1	2,3	2,3	-	-
U_{31}	-	-	-	-	-	1,2	1,2
U_{32}	-	-	-	-	-	1,3	1,2
U_{33}	-	-	-	-	-	2,2	2,3

Consideram-se os seguintes custos de armazenamento, calculados para a quantidade definida na encomenda em negociação.

U_{11}	U_{12}	U_{13}	U_{14}	U_{21}	U_{22}	U_{31}	U_{32}	U_{33}	U_{41}	U_{42}
0.3	0.4	0.5	0.5	0.3	0.4	0.1	0.4	0.3	0.2	0.2

5.4.4.2 Avaliação das alternativas possíveis

Na avaliação das alternativas possíveis para a solução, consideraram-se o conjunto de ofertas recebidas dos modelos locais de acordo com o representado na Figura 5.7 (cada barra horizontal corresponde a uma oferta, oriunda de uma unidade; cada coluna corresponde a um estágio). Nesta figura é apresentada para cada unidade a data início, a data fim e o custo de produção respectivo (por exemplo, no segundo estágio, a unidade U_{21} gerou uma oferta com um custo total de 8 e com uma data de início de 50 e data de fim de 64).

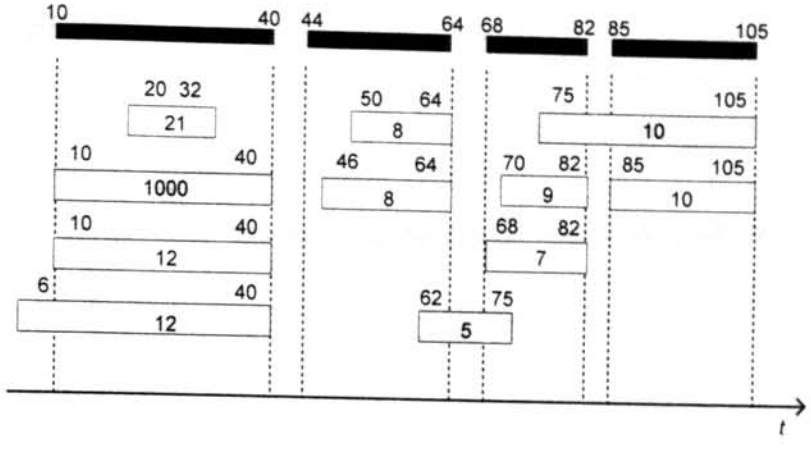


Figura 5.7 – Ofertas recebidas

As ofertas que violam qualquer das datas propostas na solução ou que apresentem custo muito elevado (representando a impossibilidade de realização) são eliminadas, não sendo por isso consideradas na avaliação. No caso presente, a eliminação de quatro das ofertas recebidas (representadas na figura a sombreado) reduz o número de alternativas, de 48 rotas possíveis para apenas 8. A aplicação directa das expressões de custos definidas anteriormente ao conjunto de ofertas geradas pelos modelos locais origina, para as alternativas possíveis, os seguintes valores de custo global:

Alternativa	Custo Produção	Custo Armazen.	Custo Transporte	Custo Global
1	48	6.2	6	60.2
2	46	6.2	7	59.2
3	45	5.4	4	54.4
4	43	5.2	7	55.2
5	39	4.9	7	50.9
6	37	4.9	8	49.9
7	36	3.3	5	44.3
8	34	3.1	8	45.1

A melhor solução é a alternativa 7 (representada graficamente na Figura 5.8) que apresenta um custo global inferior, apesar de não apresentar o custo de produção mais baixo.

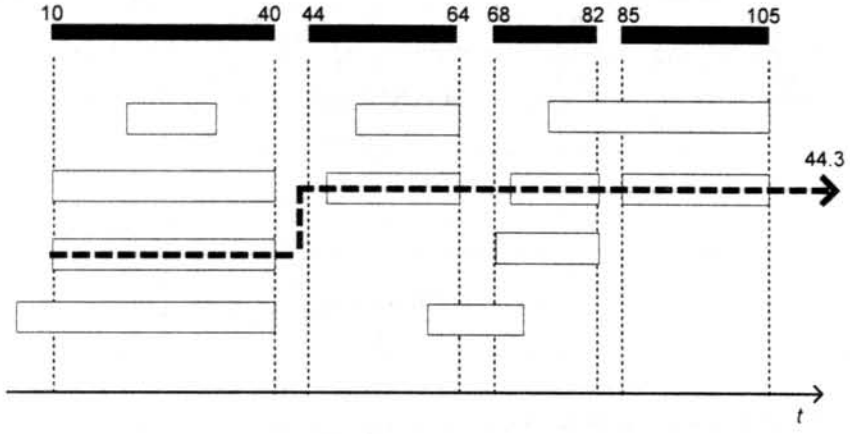


Figura 5.8 – Rota da melhor alternativa

Nas secções seguintes, serão apresentados e avaliados diferentes algoritmos para o problema de planeamento de encomendas em estruturas produtivas distribuídas (concretamente, para o problema de *inserção de uma encomenda*). Os algoritmos desenvolvidos, baseados em "*simulated annealing*" e *algoritmos genéticos*, de acordo com as considerações apresentadas em 5.2.2 apresentam um âmbito global (nível superior do processo de decisão), determinando não só uma solução para o problema, como simultaneamente garantindo a coordenação das entidades envolvidas. Assim, nas próximas secções serão feitas referências mais pormenorizadas a estas meta-heurísticas e em particular à sua aplicabilidade no problema de planeamento anteriormente descrito.

5.5 *Simulated Annealing*

5.5.1 Referências históricas e bibliografia

O algoritmo de *Simulated Annealing* (SA) inspira-se em fenómenos da termodinâmica e da metalurgia, nomeadamente, no processo físico segundo o qual, num sólido aquecido e seguido de um arrefecimento controlado da temperatura, se verifica uma evolução para estruturas de energia mínima. O *Simulated Annealing* foi pela primeira vez proposto por Kirkpatrick, Gelatt e Vecchi em 1982 (Kirkpatrick et al., 1983) com base no algoritmo apresentado por Metropolis em 1953 (Metropolis et al., 1953) que, inspirando-se no processo físico da cristalização dos sólidos, definiu um conjunto de procedimentos para a resolução de problemas de Optimização Combinatória.

Os princípios e fundamentação teórica deste algoritmo encontram-se descritos em inúmeros trabalhos, dos quais se salientam (Laarhoven and Aarts, 1987), (Reeves, 1995) e (Pirlot, 1994). A aplicação prática do *Simulated Annealing* cresceu, nos últimos anos, de uma forma extraordinária. Alguns exemplos práticos da sua aplicação em diferentes áreas encontram-se em (Rayward-Smith, 1995), (McMullen and Strong, 1999), (Sadeh et al., 1995), (Souilah, 1993), (Vilarinho, 1997), (Carravilla, 1996), (Oliveira, 1995) e (Melício et al., 1999).

5.5.2 Estrutura geral do algoritmo

Em termos gerais, o que distingue um método de pesquisa baseado no algoritmo de *Simulated Annealing* (ou, em termos gerais, qualquer outra meta-heurística) de um método tradicional de Pesquisa Local, é a inclusão de mecanismos "aleatórios" que procuram evitar que a pesquisa caia "irrecuperavelmente" em mínimos locais.

A implementação prática do método consiste em partir de uma solução admissível inicial para o problema e repetidamente, até se satisfazer um dado critério de paragem, seleccionar uma nova solução, que pertença à vizinhança da solução actual. A diferença fundamental deste método de pesquisa, relativamente a métodos tradicionais de Pesquisa Local nos quais se passa iterativamente de uma solução para outra vizinha segundo a direcção que optimiza o objectivo (soluções sucessivamente melhores), re-

side no facto de se permitir que soluções piores do que a actual sejam consideradas, tendo em vista explorar outras zonas do espaço de soluções.

Deste modo, uma solução vizinha em análise, com melhor valor da função objectivo, é aceite como solução actual para a próxima iteração. Por outro lado, uma solução pior, relativamente à solução actual, é aceite com uma probabilidade dependente da diferença do valor da função objectivo entre as duas soluções e do valor de um parâmetro de controlo (temperatura), que vai diminuindo gradualmente ao longo do tempo, permitindo assim regular a quantidade de soluções piores aceites na fase de pesquisa (diversificação). Assim, considerando s_1 como a solução candidata a substituir a solução actual s_0, e $F(s_1)$ e $F(s_0)$ os respectivos valores da função objectivo, a probabilidade de aceitação da solução s_1, numa determinada iteração do algoritmo, é dada por:

$$P_{aceitação} = \begin{cases} 1 & se \quad F(s_1) \leq F(s_0) \\ e^{\frac{-(F(s_1)-F(s_0))}{T}} & se \quad F(s_1) > F(s_0) \end{cases}$$

A diminuição do parâmetro T (temperatura) provoca, à medida que o processo de pesquisa avança, a diminuição da probabilidade de aceitação de piores soluções (fase de intensificação), tornando, por isso, o processo de pesquisa mais selectivo.

A abordagem algorítmica proposta é a versão homogénea[33] da pesquisa por *simulated annealing*, isto é, a versão em que o parâmetro de temperatura é diminuído ao fim de um certo número (fixo) de iterações. Além disso, na versão proposta, é executado um número determinado de corridas, partindo sempre de diferentes pontos iniciais (soluções iniciais). O modo como uma solução é gerada a partir de outra é implicitamente regulado pela estrutura de vizinhança definida para o problema.

O Algoritmo 5.4 apresenta uma descrição genérica da estrutura de um algoritmo baseado no *Simulated Annealing*.

No algoritmo proposto é feita a distinção entre *solução actual*, *melhor solução na corrida* e *melhor solução*.

---

[33] De acordo com a teoria subjacente a este algoritmo, esta versão (homogénea) pode ser descrita como uma sequência de cadeias de Markov homogéneas, cada uma gerada a uma temperatura fixa (Laarhoven and Aarts, 1987).

Notação utilizada

S_0	Solução actual
S_1	Solução candidata a substituir a solução actual
S_t	Melhor solução encontrada na 'corrida' actual
S^*	Melhor solução encontrada desde o início
$f(S)$	Valor da função objectivo para a solução S
M	Número total de 'corridas'
N	Número máximo de soluções geradas em cada patamar de temperatura
Q	Número máximo de soluções consecutivas geradas num patamar em que não se verifica melhoramento da solução
T	Temperatura actual
T_0	Temperatura inicial
α	Taxa de redução da temperatura em patamares sucessivos ($0 < \alpha < 1$)
i	Contador do número de 'corridas'
j	Contador do número de soluções consecutivas em que não se verificou melhoria
l	Contador do número de soluções geradas num patamar

```
algoritmo simulated_annealing

 S* ← +∞;
 para i=1 até M
  S0 ← SoluçãoInicial();
  T ← T0;
  St ← S0;
  j ← 0;
  enquanto j≤Q
     l ← 0;
     enquanto l≤N e j≤Q
        S1 ← Vizinho(S0);
        l ← l+1;
        j ← j+1;
        Δf ← f(S1)-f(S0);
        se Δf<0 então
           S0 ← S1;
           se f(S0)<f(St) então
              St ← S0;
              j ← 0;
           fimSe
        senão
           U ← NumeroAleatório(0,1);
           se U≤e^(-Δf/T) então
              S0 ← S1;
           fimSe
        fimSe
     fimEnquanto
     T ← T·α;
  FimEnquanto
  se f(S*)>f(St) então
     S* ← St;
  fimSe
 fimPara
fim
```

Algoritmo 5.4 – *Simulated Annealing*

A implementação de um algoritmo de pesquisa baseado no *simulated annealing* pressupõe a tomada de decisões de carácter genérico e específicas do problema. As decisões de carácter genérico envolvem basicamente a definição do valor da temperatura inicial, o esquema de arrefecimento, o valor inferior de temperatura, a definição da probabilidade de aceitação das soluções piores, e o critério de paragem do algoritmo. As decisões de carácter específico ao problema envolvem a forma como as soluções são representadas e geradas, envolvem igualmente a definição da estrutura de *vizinhança* e a forma como é avaliada a solução, em cada iteração.

5.5.3 Decisões de carácter genérico

A implementação do algoritmo de *simulated annealing* pressupõe a definição dos seguintes parâmetros de controle, de carácter geral:

- determinação da temperatura inicial;
- selecção de uma função de redução da temperatura (estratégia de arrefecimento);
- definição do número de iterações à mesma temperatura;
- critério de paragem.

Determinação da temperatura inicial

Por forma a garantir um comportamento inicial de diversificação na pesquisa do espaço de soluções, a temperatura inicial T_0 deve ter um valor tal que a probabilidade de aceitação de piores soluções seja relativamente elevada.

O valor deste parâmetro é fortemente dependente da natureza do problema pelo que a sua afinação deve ser feita experimentalmente. No caso presente, optou-se por seguir uma regra empírica proposta em (Kirkpatrick et al., 1983) na qual se sugere que o valor da probabilidade de aceitação de soluções inferiores seja aproximadamente 80%. Desta forma, o valor da temperatura inicial é calculado a partir da expressão $T_0 = -\Delta F_{\max} / \ln(0.8)$ em que $\Delta F_{\max}$ é um majorante para a variação da função objectivo.

Função de redução da temperatura

Diversas funções têm sido propostas na literatura para controlar a forma como o arrefecimento se processa:

- função aritmética $T_n = T_{n-1} - K$
- função geométrica $T_n = \alpha \cdot T_{n-1}$
- função inversa $T_n = \dfrac{K}{1+n}$
- função logarítmica $T_n = \dfrac{K}{1+\log(n)}$

De entre as várias funções, a mais utilizada é o esquema de arrefecimento geométrico, isto é, após uma série de N iterações, a temperatura é diminuída pela multi-

plicação por um valor real constante (factor de arrefecimento). Após variadas experiências optou-se por escolher o esquema de arrefecimento geométrico.

Tipicamente, e devido ao desempenho relativamente robusto do algoritmo, considera-se para o factor de arrefecimento α, valores compreendidos entre 0.85 e 0.95.

Número de iterações à mesma temperatura

Este parâmetro determina o número de soluções geradas à mesma temperatura. Várias funções têm sido propostas na literatura, nomeadamente:

- função constante $L_n = K$
- função aritmética $L_n = L_{n-1} + K$
- função geométrica $L_n = \frac{L_{n-1}}{\alpha} \quad \wedge \quad 0 < \alpha < 1$
- função exponencial $L_n = L_{n-1}^{1/\alpha} \quad \wedge \quad 0 < \alpha < 1$
- função logarítmica $L_n = \frac{K}{\log(L_n)}$

A versão clássica do algoritmo usa o esquema constante e independente da temperatura. Na implementação efectuada, optou-se igualmente por considerar constante o número de iterações realizadas à mesma temperatura.

Critério de paragem

Na analogia do algoritmo com o processo físico de arrefecimento dos sólidos, o critério de paragem específica o estado em que se considera que se atingiu o equilíbrio térmico (sistema "cristalizado"). Na prática, são utilizados diversos critérios de paragem, nomeadamente:

- terminar quando não houver melhoria no valor da função objectivo, após um número pré-determinado de iterações (múltiplo do número de iterações à mesma temperatura);
- terminar após um número pré-determinado de iterações;
- terminar quando se atingir um determinado valor mínimo da temperatura;
- terminar quando se atingir um valor mínimo da razão entre o número de soluções aceites e o número de soluções geradas.

No caso presente, o critério de paragem considerado é o seguinte: o algoritmo pára ao fim de um número determinado de M corridas. Contudo, verifica-se a interrupção de uma corrida sempre que, após Q iterações nessa corrida, não se verifique melhoria da função objectivo.

5.5.4 Decisões específicas ao problema

As decisões específicas ao problema dizem basicamente respeito ao espaço de soluções do problema. Assim, incluem a definição da forma como é gerada a solução inicial, a definição da estrutura da vizinhança e da forma como se procede à geração das soluções vizinhas e a definição da função de avaliação, ou função objectivo.

A solução inicial, em cada corrida do algoritmo, é gerada a partir de uma das três heurísticas apresentadas anteriormente (ver secção 5.4.2). Gerada a solução inicial, ela vai ser modificada movendo-se as datas de início e de fim (associadas aos estágios) para a frente ou para trás. A vizinhança de uma solução é assim definida como todos os conjuntos de janelas temporais que se obtêm por alteração das datas de início e de fim, associadas a um dos estágios. Uma solução vizinha é gerada escolhendo aleatoriamente um estágio e movimentando a data de início e/ou de fim, de um número de períodos também aleatório.

A avaliação das soluções, consiste na determinação do custo global associado ao processamento da encomenda. Neste custo, consideram-se não só os custos individuais de processamento, pelas unidades envolvidas, como também os custos de transporte, custos eventuais de armazenamento e custos associados a eventuais desvios da data possível de entrega relativamente à *due-date*.

A aplicação do *simulated annealing* no caso presente, obrigou a uma decomposição algorítmica em três níveis principais, de acordo com o processo descrito em 5.2.2. O primeiro nível inclui basicamente duas heurísticas: uma responsável pela geração de uma solução inicial e outra responsável pela geração de uma solução vizinha da solução actual.

O segundo nível é responsável pela implementação do mecanismo de avaliação de uma solução gerada no nível anterior. Após a partição da solução a avaliar pelo número de estágios do processo produtivo, inicia-se um processo iterativo de negociação, envolvendo os modelos de capacidade local. Neste nível, simultaneamente para

cada estágio, procede-se à selecção ou à agregação de unidades de produção nesse estágio que permitam satisfazer a correspondente solução parcial. A determinação do valor da função objectivo para a solução actual é feita recorrendo ao cálculo combinado dos custos individuais de processamento, transporte, etc.

O terceiro nível, constituído por diversas instâncias de modelos de capacidades, gera para cada pedido recebido a oferta correspondente (datas possíveis de realização, custo associado, etc.). Após a recepção das ofertas, e da sua combinação, procede-se ao cálculo dos custos associados.

5.5.5 Estrutura de vizinhança

A partir de uma solução inicial, a evolução pelo espaço de soluções do problema é imposta pela estrutura de vizinhança utilizada. Assim, para cada solução actual, a sua vizinhança é composta pelas soluções que se podem atingir, efectuando movimentos elementares específicos.

No caso presente, uma solução vizinha obtém-se seleccionando aleatoriamente um estágio, com igual probabilidade para todos os estágios, e gerando as novas datas de início e de fim associadas a esse estágio. As datas geradas são delimitadas por duas datas (limite inferior e limite superior) calculadas a partir da data fim do estágio precedente, acrescida do tempo mínimo de transporte entre estágios, e a data início do estágio subsequente, diminuída igualmente da correspondente duração do transporte entre estágios. A Figura 5.9 ilustra graficamente a evolução da solução, na situação de ser seleccionado um estágio *i*. A geração aleatória das novas datas para o estágio alvo pode originar, não só movimentações para a esquerda ou para a direita, como pode também levar a um aumento ou a uma diminuição do tempo permitido para processamento no estágio (ver Algoritmo 5.5).

Para os estágios extremos, primeiro e último, na geração das novas datas consideram-se como limites para a data início do primeiro estágio, a data actual, e para a data fim do último estágio a data mais tarde permitida (*latest date*).

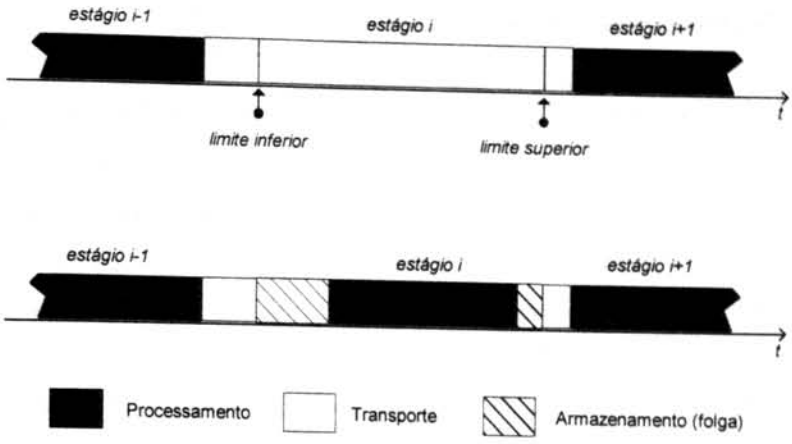


Figura 5.9 – Geração de uma solução vizinha

```
algoritmo vizinho

 u ← numAleatório(1,num_estágios);
 i ← estágioAlvo(u);
 se i=primeiroEstágio então
  t1 ← numeroAleatório(1,solução[i+2]-t_move[i]);
  t2 ← numeroAleatório(1,solução[i+2]-t_move[i]);
 fimSe
 se i=ultimoEstágio então
  t1 ← numeroAleatório(solução[i-1]+t_move[i-1],ld);
  t2 ← numeroAleatório(solução[i-1]+t_move[i-1],ld);
 senão
  t1 ← numeroAleatório(solução[i-1]
            +t_move[i-1],solução[i+2]-t_move[i]);
  t2 ← numeroAleatório(solução[i-1]
            +t_move[i-1],solução[i+2]-t_move[i]);
 fimSe

 se t1<t2 então
  solução[i] ← t1;
  solução[i+1] ←t2;
 senão
  solução[i] ← t2;
  solução[i+1] ←t1;
 fimSe

 retorna solução;
fim
```

Algoritmo 5.5 – Heurística para geração de uma solução vizinha

5.5.6 Avaliação das soluções

Dada a abordagem hierárquica adoptada, a avaliação de cada uma das soluções geradas (com base numa dada função objectivo) não é, em geral, um processo simples.

Note-se que uma solução inicial ou uma solução vizinha, geradas de acordo com as heurísticas anteriormente apresentadas e respeitando as restrições definidas, por si só, não são necessariamente soluções possíveis do problema. A segmentação da resolução do problema em três níveis (ver Figura 5.5) introduz um desacoplamento natural entre o processo que gera e determina soluções candidatas e o processo que verifica a exequibilidade das mesmas, de acordo com o contexto produtivo actual.

Para cada solução gerada (candidata), é feita a correspondente avaliação, submetendo-a primeiro a um processo de negociação envolvendo unidades autónomas (modelos locais de capacidade). A análise das ofertas recebidas de cada unidade permite avaliar diferentes alternativas, seleccionando-se aquela que apresenta, na globalidade, um custo menor. Assim, com base nas ofertas recebidas (custo de produção, quantidade, e período de produção) é calculado o custo associado à solução em análise, combinando quatro custos parcelares: custos de produção, custos de transporte entre estágios, custos de armazenamento e custos de penalização por afastamentos relativamente à *due date*. De seguida, apresenta-se, para cada uma das parcelas de custo referidas, a expressão utilizada na avaliação das ofertas.

Notação

$i \quad (i = 1,\ldots, I)$	estágio de produção
C_{ij}^{prod}	custo de produção para a unidade *j* do estágio *i*
C_{ij}^{hold}	custo de armazenamento para a unidade *j* do estágio *i*
C_{ijl}^{move}	custo de transporte entre a unidade *j* do estágio *i* e a unidade l do estágio *i+1*
t_{ij}^{start}	início da produção na unidade *j* do estágio *i*
t_{ij}^{finish}	fim da produção na unidade *j* do estágio *i*
t_{ijl}^{move}	tempo de transporte entre a unidade *j* do estágios *i* e a unidade *l* do estágio *i+1*

γ_{ij} variável binária que indica se a unidade j do estágio i participa no processamento

dd data de entrega pedida (*due date*)

custos de produção

$$\sum_{i} \sum_{j \in J_i} c_{ij}^{prod} \cdot \gamma_{ij} \tag{6}$$

custos de transporte

$$\sum_{i=1..I-1} \sum_{j \in J_i} \sum_{l \in J_{i+1}} c_{ijl}^{move} \cdot \gamma_{ij} \cdot \gamma_{i+1,l} \tag{7}$$

custos de armazenamento entre estágios

$$\begin{aligned} &\sum_{i=1..I-1} \sum_{l \in J_{i+1}} \sum_{j \in J_i} c_{ij}^{hold} \cdot \gamma_{ij} \cdot \gamma_{i+1,l} \cdot \left(t_{i+1,j}^{start} - \left(t_{ij}^{finish} + t_{ijl}^{move} \right) \right) \\ &+ \sum_{j \in J_I} c_{Ij}^{hold} \cdot \gamma_{Ij} \cdot \max\left\{ \left(dd - t_{Ij}^{finish} \right), 0 \right\} \end{aligned} \tag{8}$$

desvios relativamente à data de entrega

$$\min \quad \sum_{j \in J_I} \left(\max\left\{ k_2 \cdot \left(t_{Ij}^{finish} - dd \right), 0 \right\} \right) \tag{9}$$

Naturalmente que o processo de avaliação pressupõe, para a solução em análise, a definição de uma rota de processamento envolvendo, para cada estágio, as unidades que permitem obter o custo global mais baixo. A Figura 5.10 apresenta a rota parcial resultante da resolução do problema de afectação, associado a cada estágio, na situação de existirem ofertas alternativas.

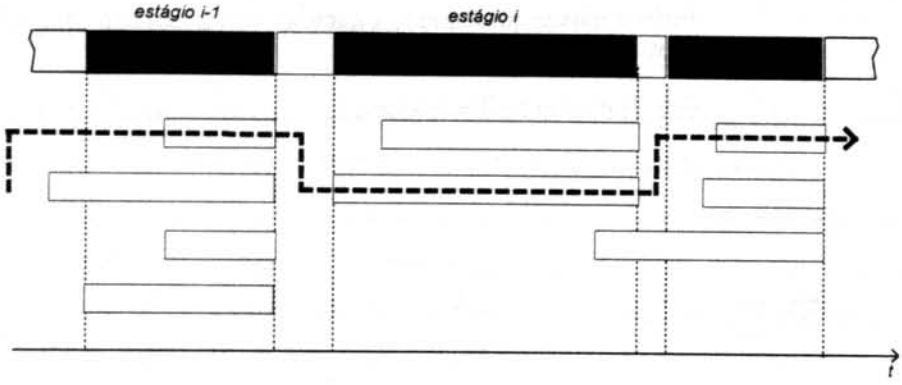


Figura 5.10 – Rota parcial associada a uma solução

5.5.7 Experiências computacionais

5.5.7.1 Enquadramento e instâncias teste

No contexto deste trabalho, o principal objectivo da realização de testes computacionais, foi o de avaliar, em primeiro lugar, a aplicabilidade e as potencialidades das metaheurísticas, na resolução do problema de planeamento de uma nova encomenda. Não se procurou, portanto, estudar exaustivamente o comportamento e a sensibilidade dos algoritmos relativamente aos seus parâmetros de controlo.

Antes de se apresentarem os testes realizados e os resultados computacionais obtidos, caracteriza-se uma instância genérica do problema, que é, basicamente, constituída por duas partes:

- *modelo da rede produtiva* – define todas as entidades produtivas consideradas na rede e as suas relações; caracteriza os modelos de capacidade associados a cada entidade; define as interacções admissíveis entre unidades (através de restrições logísticas, por exemplo, tempo e custo de transporte); identifica os diferentes estágios produtivos;

- *pedido a planear* – especifica completamente o pedido do cliente, nomeadamente, o tipo de produto, a quantidade e a data de entrega pretendida, bem como, eventualmente, alguma folga ("tolerância") relativamente a essa data.

Na implementação actual, considerou-se que em cada pedido se definem os custos de penalização por eventuais desvios entre a data de "entrega" planeada e a data pretendida (atrasos ou antecipações).

Refira-se que algumas decisões de implementação, nomeadamente as de carácter genérico, basearam-se em experiências descritas por outros autores e em testes computacionais relativamente exaustivos, que apesar de não se relatarem neste texto, foram usados para limitar a gama de variações dos parâmetros dos algoritmos.

Fez-se, numa fase preliminar deste trabalho, um conjunto alargado de experiências, sobre diferentes instâncias, que permitiram ganhar sensibilidade quanto ao comportamento e desempenho do algoritmo. Assim, experimentaram-se diferentes configurações da rede produtiva, quer ao nível do número de estágios envolvidos, quer ao nível do número de unidades produtivas e respectiva distribuição pelos estágios; experimentaram-se igualmente diferentes cenários relativos à maior ou menor relevância da componente logística face à componente produtiva e exploraram-se também cenários relativos ao tipo de pedido a planear, nomeadamente quanto a diferentes combinações data pretendida / quantidade pretendida.

Nesta secção, apresentam-se resultados computacionais sobre diferentes instâncias que têm por base a mesma estrutura produtiva (descrita pormenorizadamente no Apêndice A). Note-se que estes resultados não são substancialmente diferentes, em termos de qualidade, dos observados para outras estruturas.

A estrutura produtiva considerada é representativa, por um lado, da complexidade geral do problema em análise e, por outro lado, apresenta uma configuração idêntica à dos casos piloto (situações reais) a que se teve acesso no âmbito do projecto X-CITTIC. Assim, no problema teste, são considerados quatro estágios (procurando reflectir os estágios usualmente considerados na indústria dos semicondutores), envolvendo um total de onze entidades de produção. A cada unidade de produção, associou-se um modelo de capacidades, similar aos modelos descritos no capítulo 2, e parametrizado por forma a reflectir as características particulares dessa unidade.

Os testes foram realizados sobre três instâncias, caracterizadas na Tabela 5.1 (no que respeita à componente *pedido a planear*).

Instância	*due-date*	quantidade	folga	penalização
A	80	400	5	1.0
B	50	400	2	2.0
C	120	400	2	2.0

Tabela 5.1 – Caracterização das instâncias (pedidos a planear)

Como referido, as três instâncias consideram o mesmo tipo de produto e a mesma estrutura produtiva (ver Apêndice A).

5.5.7.2 Parametrização do algoritmo

Tendo em conta o carácter aleatório intrínseco do *simulated annealing*, e com vista a "absorver" nos resultados a variabilidade que daí resulta, todos os testes foram baseados em 30 corridas do algoritmo, partindo sempre de diferentes soluções iniciais (obtidas por heurísticas construtivas já de si com comportamento aleatório).

Os testes foram realizados com os seguintes parâmetros genéricos cujos valores, como referido, foram afinados numa fase prévia, através de um conjunto alargado de experiências.

Temperatura inicial

O valor T_0 da temperatura inicial foi determinado de modo a garantir que o valor da probabilidade de aceitação de soluções piores, fosse da ordem de 80%. Assim, considerou-se $T_0 = 1000$.

Número de iterações à mesma temperatura

Considerou-se constante o número de iterações à mesma temperatura, tendo sido usado o valor $L = 400$.

Esquema de arrefecimento

Utilizou-se um esquema de arrefecimento geométrico, com um factor de arrefecimento $\alpha = 0.85$.

Critério de paragem

Termina-se o processo de optimização quando não há melhoria no valor da função objectivo após $2 \cdot L$ iterações (neste caso, 800).

A Figura 5.11 apresenta o valor da função objectivo ao longo da pesquisa para uma corrida típica do algoritmo.

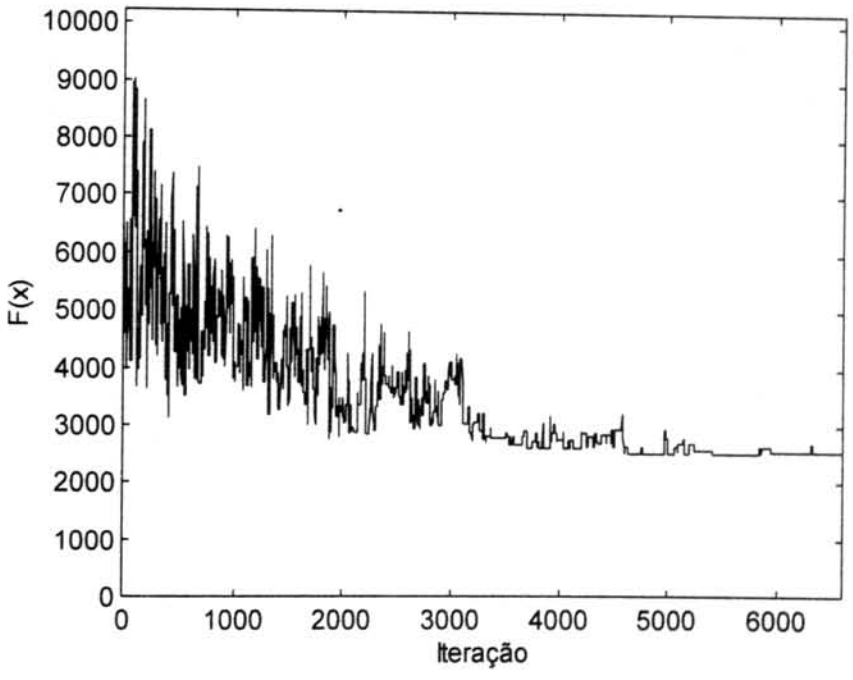


Figura 5.11 – Variação da função objectivo (T_0 =1000; $\alpha = 0.85$; $L = 400$)

5.5.7.3 Geração de soluções iniciais e dimensão da subvizinhança

Relativamente à geração de soluções iniciais foram consideradas as três heurísticas apresentadas em 5.4.2 e aqui referenciadas por H-I, H-II e H-III. Foram estudados os seguintes valores para a dimensão da subvizinhança: 1, 5, 10, 15 e 20.

Apresentam-se, a seguir, os resultados computacionais obtidos num conjunto de experiências, efectuadas com a finalidade de determinar o impacto das heurísticas referidas e a dimensão a adoptar para a subvizinhança (isto é, o número de soluções vizinhas a pesquisar em cada iteração).

Para cada uma das instâncias, são apresentados os valores médios do custo global da solução obtida, partindo de diferentes estratégias de geração de soluções iniciais (heurísticas H-I, H-II e H-III) e utilizando diferentes dimensões para a subvizinhança. São apresentados igualmente os tempos médios de processamento, envolvidos numa corrida do algoritmo.

Sob a forma de tabela, apresenta-se também, para cada uma das instâncias consideradas, os valores médio, mínimo e máximo para a solução inicial e para a solução final (em função da heurística de geração de soluções iniciais e da dimensão da subvizinhança). Estes valores referem-se, naturalmente, aos conjuntos de 30 corridas, realizadas para cada caso (isto é, instância e valores dos parâmetros).

Instância A

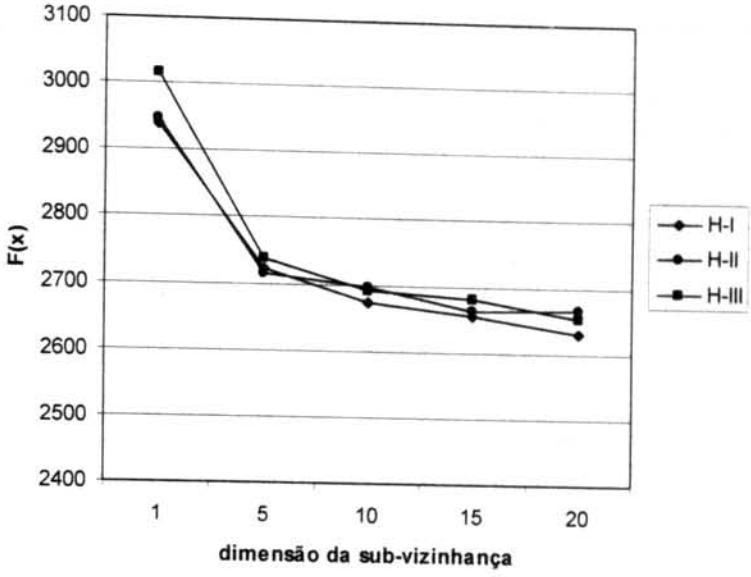


Figura 5.12 – Custo médio da solução final (instância A)

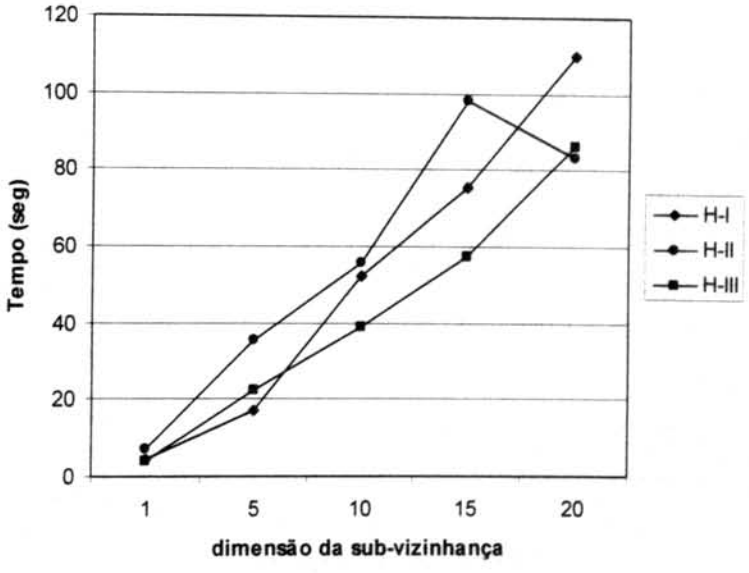


Figura 5.13 – Tempo médio de processamento (instância A)

Heurística		Solução inicial	Solução Final (subvizinhanças de diferente dimensão)				
			1	5	10	15	20
H-I	média	10635.9	2939.2	2723.2	2672.8	2655.5	2629.4
	mín.	5004.1	2664.4	2527.5	2555.5	2555.5	2527.5
	máx.	16311.3	3177.7	2868.4	2744.4	2748.4	2720.4
H-II	média	5142.9	2946.6	2713.5	2696.0	2663.8	2665.3
	mín.	4814.1	2756.4	2608.4	2572.4	2588.4	2558.1
	máx.	5638.1	3120.4	2784.4	2776.4	2732.4	2780.4
H-III	média	5626.4	3016.8	2737.5	2691.5	2680.8	2653.8
	mín.	2976.4	2647.5	2586.1	2618.1	2572.4	2548.4
	máx.	8345.0	3382.0	2840.4	2748.4	2800.4	2760.4

Tabela 5.2 – Resumo dos resultados computacionais para a instância A

Instância B

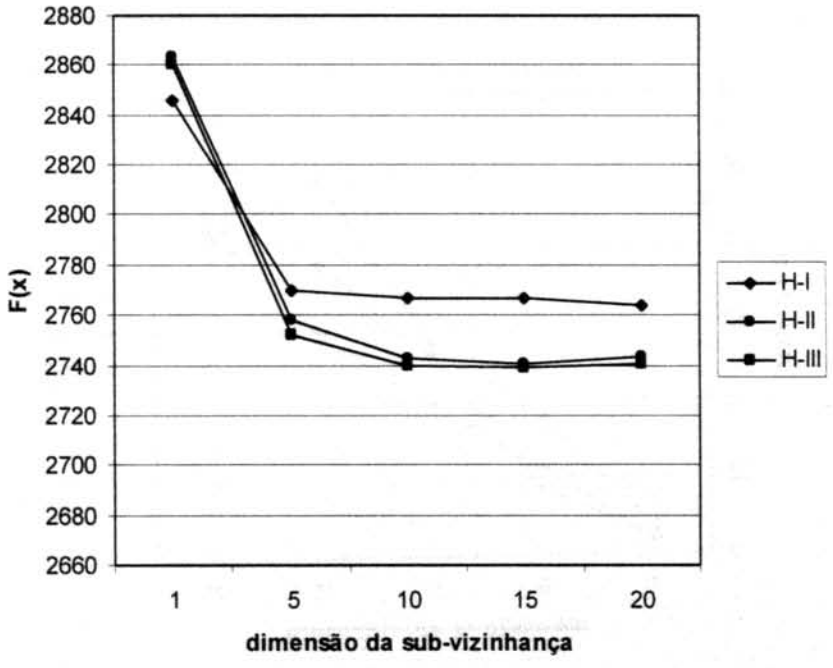


Figura 5.14 – Custo médio da solução final (instância B)

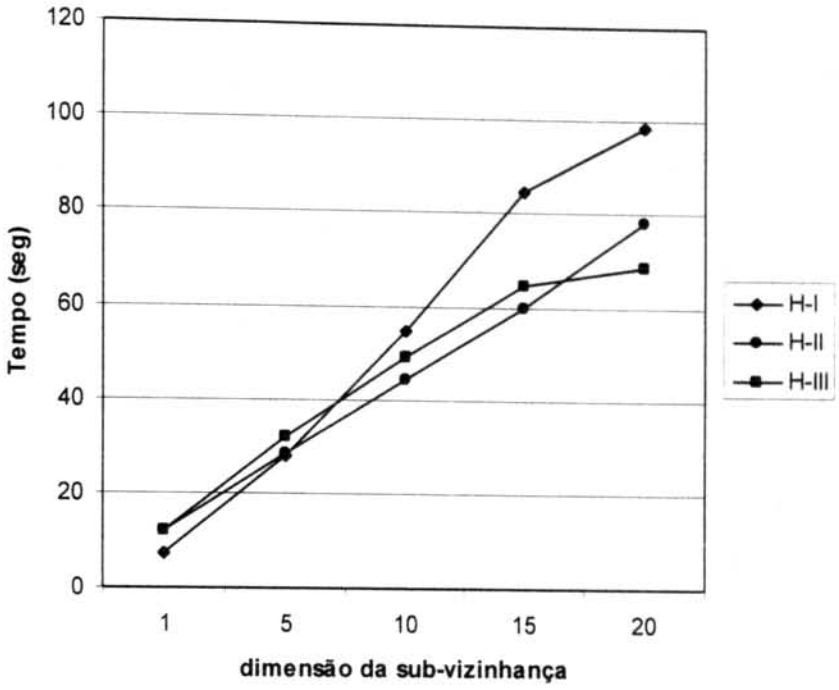


Figura 5.15 – Tempo médio de processamento (instância B)

Heurística		Solução inicial	Solução Final (subvizinhanças de diferente dimensão)				
			1	5	10	15	20
H-I	média	5782.8	2846.0	2769.7	2766.8	2766.8	2763.5
	mín.	3482.1	2705.1	2661.2	2661.2	2661.2	2661.2
	máx.	8237.7	2941.7	2857.1	2875.5	2870.1	2870.1
H-II	média	5290.5	2862.9	2757.6	2742.6	2740.3	2743.4
	mín.	5001.6	2703.5	2677.2	2661.2	2703.5	2703.5
	máx.	7313.2	2956.9	2808.1	2799.5	2791.5	2799.5
H-III	média	5631.8	2860.0	2752.4	2739.3	2739.1	2740.4
	mín.	4957.7	2661.2	2661.2	2703.5	2703.5	2703.5
	máx.	7313.2	2954.8	2808.1	2799.5	2799.5	2799.5

Tabela 5.3 – Resumo dos resultados computacionais para a instância B

Instância C

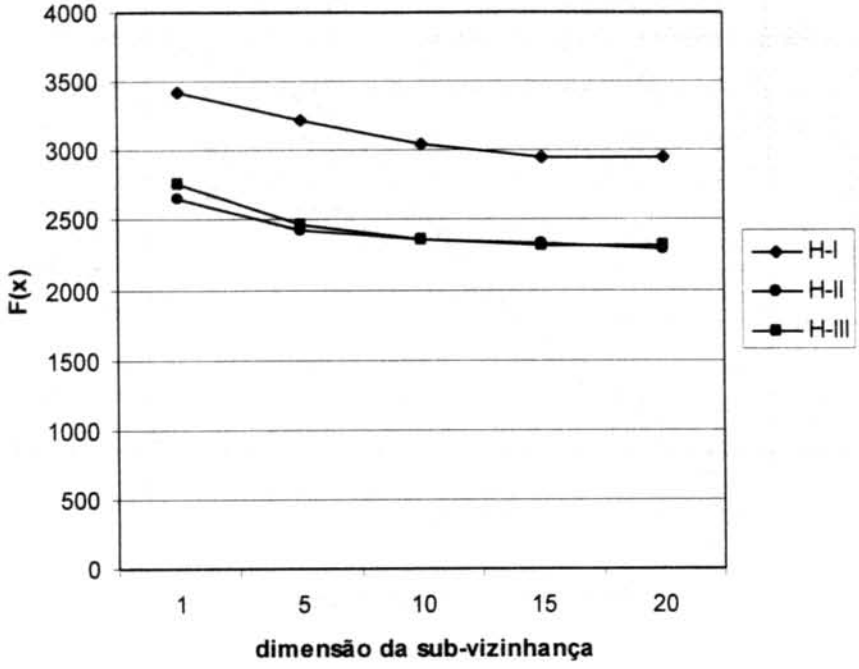


Figura 5.16 – Custo médio da solução final (instância C)

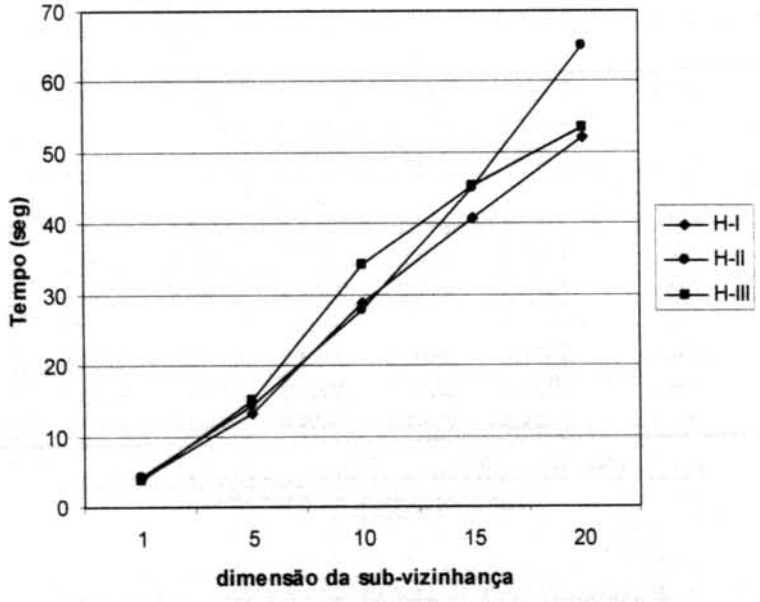


Figura 5.17 – Tempo médio de processamento (instância C)

Heurística		Solução inicial	Solução Final (subvizinhanças de diferente dimensão)				
			1	5	10	15	20
H-I	média	22031.5	3422.4	3212.6	3046.1	2954.4	2952.7
	mín.	5950.2	2407.5	2330.1	2290.1	2310.1	2418.0
	máx.	34619.7	3841.9	4203.9	4322.8	3193.4	3268.7
H-II	média	4494.1	2659.8	2426.8	2355.2	2323.3	2291.9
	mín.	4258.1	2320.9	2296.9	2254.1	2254.1	2210.1
	máx.	4980.9	2947.7	2572.9	2448.9	2408.9	2388.9
H-III	média	7742.1	2757.4	2462.8	2357.7	2317.3	2314.4
	mín.	4232.9	2350.1	2252.9	2225.4	2217.4	2218.1
	máx.	45868.9	3129.5	2650.1	2571.5	2476.9	2430.1

Tabela 5.4 – Resumo dos resultados computacionais para a instância C

Dos resultados obtidos, podem tirar-se as seguintes conclusões:

1. O algoritmo converge rapidamente (isto é, ao fim de cerca de metade das iterações realizadas, a solução que se obtém é, em geral, muito próxima da solução final).
2. Os resultados são, naturalmente, melhores à medida que aumenta a dimensão da subvizinhança, com a contrapartida óbvia de um aumento do tempo de processamento. Neste compromisso, verifica-se uma melhoria significativa quando se passa de uma subvizinhança de dimensão 1 para uma subvizinhança de dimensão 5, com um aumento pouco importante no tempo de processamento.
3. Em geral, conseguem-se melhores resultados partindo-se de soluções iniciais geradas a partir das heurísticas H-II e H-III do que com a heurística H-I. Em certo sentido, esta conclusão era previsível, particularmente para situações em que as datas pretendidas estão muito afastadas da data actual. Nestas situações, as unidades produtivas, apresentando naturalmente uma maior capacidade disponível, permitem uma estimação mais fiável dos tempos de produção, com base em informação histórica. As heurísticas H-II e H-III exploram precisamente esta constatação, resultando naturalmente num maior desempenho do algoritmo, relativamente à situação em

que se recorre à heurística H-I. Contudo, experiências efectuadas com outras instâncias, caracterizadas por datas pretendidas próximas da data actual, demonstraram o melhor desempenho da heurística H-I face às heurísticas H-II e H-III. Inclusive, verificou-se ser impossível gerar soluções iniciais com as heurísticas H-II e H-III, quando a data pretendida era inferior ao tempo total médio de produção (baseado em informação histórica) associado ao produto em planeamento.

4. Na geração das soluções iniciais com a heurística H-I, verifica-se uma grande dispersão do valor da função objectivo, devido ao facto do processo de geração ser totalmente aleatório.
5. Independentemente da heurística utilizada para geração de soluções iniciais, obtêm-se sempre soluções finais de muito boa qualidade, o que permite concluir que o algoritmo (meta-heurística) é robusto.

As conclusões obtidas vão de encontro às expectativas que motivaram a aplicação do *simulated annealing* ao problema que se tem vindo a analisar. Por outro lado, o conjunto de testes efectuados permitiu igualmente estabelecer, no caso geral e para além dos parâmetros genéricos considerados, que uma dimensão da ordem de 5 para a subvizinhança é adequada.

5.5.7.4 Estrutura das soluções

Apresenta-se de seguida os resultados referentes a 30 corridas do algoritmo para a instância A, utilizando para parâmetros do algoritmo os valores constantes da Tabela 5.5 afinados através das experiências apresentadas (limita-se a apresentação dos resultados à instância A, em virtude de estes serem semelhantes aos obtidos para as instâncias B e C).

T_0	α	L	Critério Paragem	Dimensão subvizinhança
1000	0.85	400	2 temperaturas	5

Tabela 5.5 – Parâmetros do algoritmo SA

Para cada uma das heurísticas H-I, H-II e H-III, apresenta-se a melhor solução final obtida (no conjunto das 30 corridas do algoritmo) assim como a correspondente solução inicial. Além disso, é apresentado o custo global associado à solução final, segundo as suas quatro componentes: custo total de produção, custo total de armazenamento entre estágios, custo total de transporte entre unidades produtivas e penalização por não cumprimento da data de entrega.

Geração da solução inicial com H-I

A solução inicial, referente à corrida do algoritmo em que se obteve a melhor solução final, estabeleceu as seguintes datas de início e de fim de produção para cada um dos estágios:

Estágio	1		2		3		4	
Datas produção	1	27	42	53	58	62	65	72

O valor da função objectivo para esta solução é de 11733.7. A solução final, após 1815 iterações, apresentou um custo global de 2527.48, com as seguintes datas de início e de fim de produção (indicam-se também as unidades de produção envolvidas em cada estágio):

Estágio	1		2		3		4	
Datas produção	42	49	51	61	63	69	71	80
Unidade produção	12		22		31		42	

Para esta solução, a segmentação dos custos foi a seguinte:

Custo total	**Custo produção**	**Custo armazen.**	**Custo transporte**	**Custo penalização**
2527.48	2367.48	0	160	0

Da análise da solução final obtida podem tirar-se as seguintes conclusões:

- respeitou-se a data pretendida, não se verificando por isso qualquer custo de penalização adicional;
- não se verifica qualquer armazenamento entre estágios, isto é, o intervalo entre a data de conclusão da produção numa unidade e a data de início de produção da unidade seguinte é exactamente igual ao tempo de transporte entre as unidades;
- apesar de se partir de uma solução inicial em certo sentido "espalhada" ao longo do período de planeamento (no caso presente desde o dia 1 ao dia 72), obteve-se uma solução final mais "concentrada" e em torno da data fim pretendida (início da produção no dia 42 e fim no dia 80).

Geração da solução inicial com H-II

Utilizando-se a heurística H-II, obteve-se uma solução inicial, referente à corrida do algoritmo em que se obteve a melhor solução final, com as seguintes datas de início e de fim de produção para cada um dos estágios:

Estágio	1		2		3		4	
Datas produção	15	25	31	43	45	64	68	80

O valor da função objectivo para esta solução é de 4936.87. A solução final, após 1901 iterações, apresentou um custo global de 2608.38 e estabeleceu as seguintes datas de início e de fim de produção:

Estágio	1		2		3		4	
Datas produção	21	31	35	51	52	69	71	80
Unidade produção	14		21		32		42	

Para esta solução, a segmentação dos custos foi a seguinte:

Custo total	Custo produção	Custo armazen.	Custo transporte	Custo penalização
2608.38	2328.38	0	280	0

Analisando os resultados, verifica-se que se obteve um custo global ligeiramente superior ao da situação anterior, apesar do custo de produção (englobando os quatro estágios) ser ligeiramente inferior. A razão desta situação deve-se ao facto dos custos de transporte, nesta situação, serem ligeiramente superiores.

Como no caso anterior, conseguiu-se com esta solução, para além do cumprimento da data pretendida, um "sincronismo" entre unidades, isto é, não se verifica qualquer armazenamento temporário entre estágios.

Geração da solução inicial com H-III

Utilizando-se a heurística H-III, obteve-se uma solução inicial, referente à corrida do algoritmo em que se obteve a melhor solução final, com as seguintes datas de início e de fim de produção para cada um dos estágios:

Estágio	1		2		3		4	
Datas produção	16	25	31	44	46	62	66	76

O valor da função objectivo para esta solução é de 6440.87. A solução final, após 1901 iterações, apresentou um custo global de 2586.13 e estabeleceu as seguintes datas de início e de fim de produção:

Estágio	1		2		3		4	
Datas produção	38	44	46	56	58	67	69	80
Unidade produção	12		22		31		42	

Para esta solução, a segmentação dos custos foi a seguinte:

Custo total	Custo produção	Custo armazen.	Custo transporte	Custo penalização
2586.13	2426.13	0	160	0

A solução final obtida neste caso é muito idêntica à obtida a partir da solução inicial gerada pela heurística H-I, verificando-se que as unidades produtivas envolvidas em cada estágio são exactamente as mesmas.

5.5.7.5 Análise estatística dos resultados

Os resultados apresentados, conjuntamente com os obtidos em muitos outros testes computacionais, permitem concluir que a qualidade das soluções finais obtidas não depende, de forma significativa, das soluções iniciais, isto é, a meta-heurística desenvolvida tem um comportamento muito robusto.

Apresenta-se de seguida, sobre a forma gráfica, informação estatística sobre os testes com a instância A, e com a qual se procura ilustrar a constatação anterior.

Assim, na Figura 5.18, representa-se graficamente, para 30 corridas do algoritmo, a mediana, o primeiro quartil, o terceiro quartil e os valores máximo e mínimo dos dados relativos à solução inicial (H-I, H-II, H-III) e à respectiva solução final (F-I, F-II, F-III), para cada uma das heurísticas de geração (estes diagramas designam-se, em inglês, por "*box plots*").

O diagrama evidencia uma clara dispersão das soluções iniciais geradas a partir da heurística H-I. Igualmente notória é a robustez do algoritmo relativamente à solução inicial (na representação da solução final, verifica-se, para as três situações representadas, praticamente a sobreposição do valor máximo, mínimo, mediana, primeiro quartil e terceiro quartil).

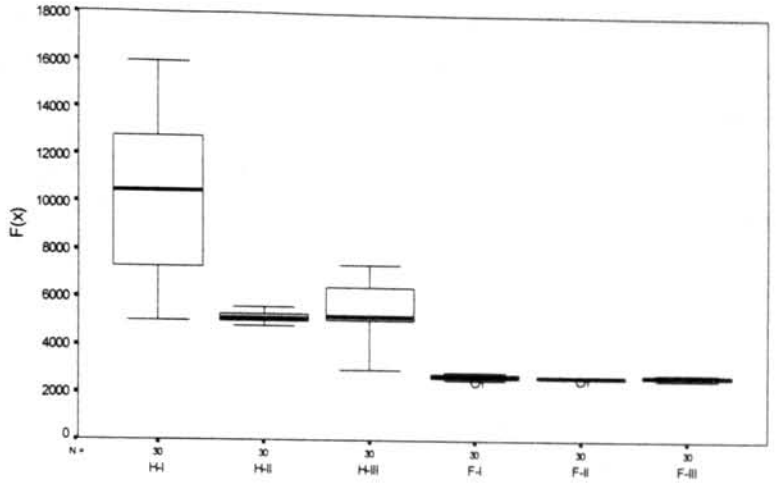


Figura 5.18 – Diagrama do tipo caixa das soluções iniciais (H-I, H-II e H-III) e soluções finais (F-I, F-II e F-III)

A Figura 5.19 representa graficamente a solução inicial e a solução final, para cada uma das 30 corridas do algoritmo, a partir das soluções iniciais obtidas com cada uma das heurísticas.

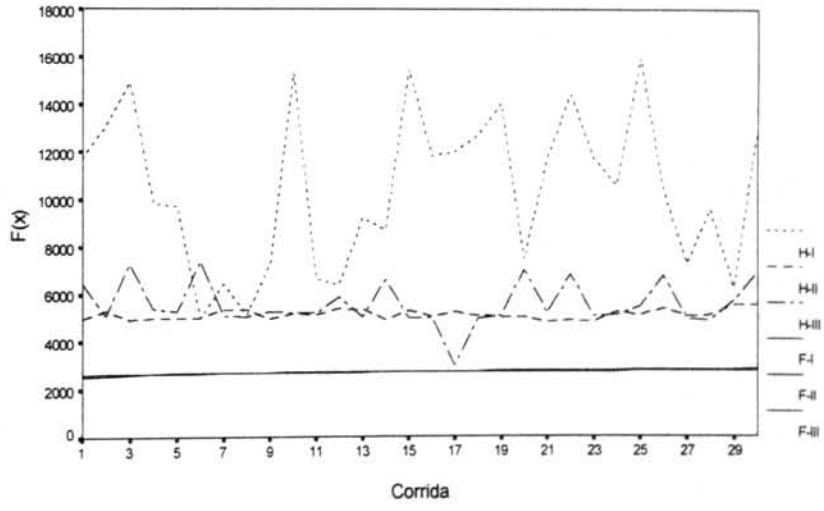


Figura 5.19 – Representação das soluções iniciais e soluções finais para cada corrida do algoritmo

A figura evidencia, uma vez mais, a robustez do algoritmo, pela "sobreposição" notória das soluções finais ao longo das corridas efectuadas, apesar da elevada variabilidade das soluções iniciais.

O diagrama (x,y) da Figura 5.20 evidencia, mais uma vez, que não existe qualquer relação entre a qualidade da solução inicial e a respectiva solução final.

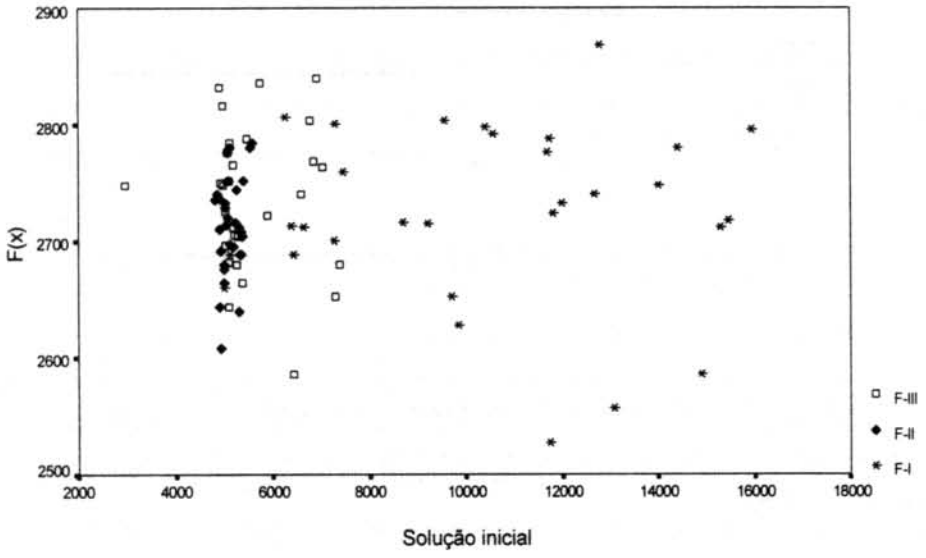


Figura 5.20 – Diagrama (x,y) da solução final e da solução inicial

Relativamente ao número de iterações realizadas, a Figura 5.21 permite concluir que, para a heurística H-I, o número de iterações apresenta uma variabilidade inferior ao verificado com as heurísticas H-II e H-III.

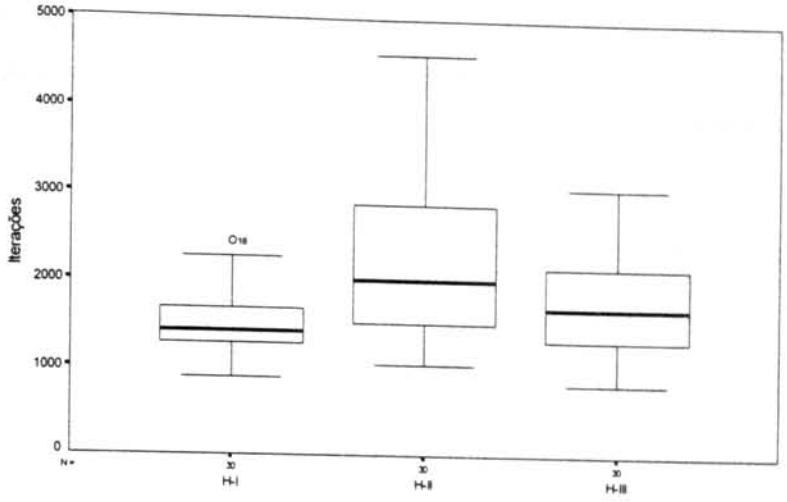


Figura 5.21 – Diagrama de caixa do número de iterações realizadas

Finalmente, o diagrama da Figura 5.22 evidencia que não existe uma relação directa entre a qualidade da solução obtida e o tempo de processamento envolvido.

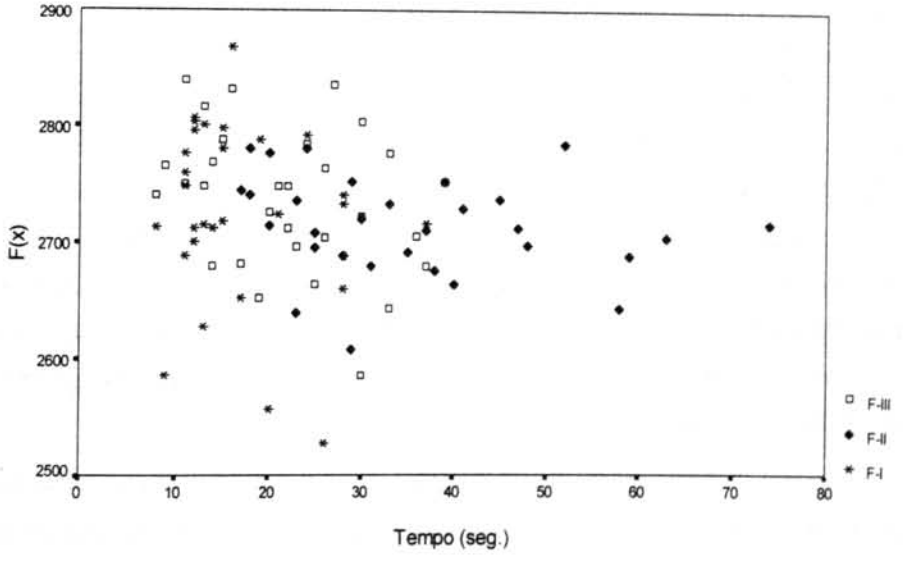


Figura 5.22 – Diagrama (x,y) da solução final e do tempo de processamento

Na verdade, verifica-se que o tempo de processamento do algoritmo se situou maioritariamente no intervalo compreendido entre os 10 e os 40 segundos, coexistindo neste intervalo as melhores e as piores soluções.

5.6 Algoritmos Genéticos

5.6.1 Referências históricas e bibliográficas

Os Algoritmos Genéticos (AG) inspiram-se nos processos genéticos dos organismos biológicos e na evolução das espécies, tendo a teoria que lhes está subjacente tido origem com os trabalhos pioneiros de John Holland em 1962. No início da década de 70, é dado um avanço decisivo neste domínio, com o desenvolvimento do Teorema dos Esquemas ou como também é conhecido como o Teorema de Holland. Contudo, foi com dois trabalhos fundamentais, publicados em 1975, que se verificou um grande impulso na investigação em torno dos AG: John Holland com o livro *Adaptation in Natural and Artificial Systems* (Holland, 1975) e De Jong com a dissertação *An Analysis of the Behavior of a Class of Genetic Adpative Systems* (Jong, 1975). Holland apresenta e desenvolve a teoria que mimetiza os sistemas naturais evolutivos com o propósito de modelar o comportamento de sistemas adaptativos complexos. De Jong combinou a teoria dos esquemas, desenvolvida por Holland, com a implementação e desenvolvimento de "operadores genéticos", que permitiram a formação de uma base sólida para a investigação desenvolvida a partir de então. Os AG podem, por isso, ser descritos como algoritmos de optimização inspirados nos mecanismos de selecção e na evolução genética das espécies (Goldberg, 1989). O princípio e fundamentação teórica dos AG encontram-se descritos em inúmeros trabalhos, dos quais se salienta (Davis, 1987; Davis, 1991; Goldberg, 1989; Pirlot, 1994; Reeves, 1995).

Em termos gerais, os algoritmos "inspirados" na teoria de selecção natural das espécies são designados por *Algoritmos Evolutivos* (Back, 1992). Nestes algoritmos, para além dos AG incluem-se entre outros, os "sistemas classificadores", as "estratégias evolutivas" e o "paradigma da programação genética". Estes métodos diferem essencialmente nas estruturas dos elementos que constituem a população, e consequentemente nos operadores específicos de recombinação e de mutação. Particularmente

interessantes são os métodos baseados em "estratégias evolutivas". Estes métodos caracterizam-se pela capacidade de aprendizagem, ao nível dos seus parâmetros estratégicos, mediante a auto-adaptação durante o funcionamento do algoritmo (ver (Back, 1992)).

O "paradigma da programação genética" foi sugerido pela primeira vez por Koza em 1989 (Koza, 1989). Nesta abordagem, existe uma população de programas que evolui por forma a obter-se um programa que obedeça a determinadas especificações. Cada programa é representado por uma estrutura em árvore, cada nó da árvore representa uma função, e cada folha representa uma constante ou variável. Os operadores genéticos são então redefinidos para actuarem sobre uma estrutura hierárquica e não sobre uma estrutura linear (Ranito, 1993).

Refira-se que a gama de aplicação dos AG cresceu rapidamente nos últimos anos, fruto de uma investigação intensa, quer ao nível da sua fundamentação teórica (Gordon and Whitley, 1993; Mahfoud, 1995a; Muhlenbein and Schlierkamp-Voosen, 1995; Park and Carter, 1994) quer essencialmente ao nível da sua aplicação a problemas reais e de âmbito industrial. Para uma revisão das aplicações dos algoritmos evolutivos a problemas reais, organizada segundo as diferentes áreas de aplicação, ver, por exemplo, (Back et al., 1993).

Deve finalmente notar-se que grande parte do sucesso dos AG tem tido lugar na resolução de problemas de Optimização Combinatória, de grande complexidade e dimensão.

5.6.2 Estrutura do algoritmo

No caso geral, um AG é um processo iterativo, sendo cada ciclo constituído por quatro estágios principais:

1. Geração de uma população de soluções;
2. Avaliação de cada elemento da população;
3. Selecção dos melhores elementos;
4. Manipulação genética da população.

A Figura 5.23 ilustra o ciclo de funcionamento dos algoritmos genéticos. Em cada ciclo, é produzida uma nova geração de soluções para o problema de optimização.



Figura 5.23 – Ciclo de funcionamento dos algoritmos genéticos

A estrutura geral de um AG, em notação algorítmica e de forma procedimental, pode ser descrita de acordo com o Algoritmo 5.6.

```
algoritmo GA
   n = 0;
   Pn = gera_população_inicial;
   avalia Pn;
   repete
    n = n + 1;
    Pn = selecçãoPn-1;
    manipulação Pn;
    avaliação Pn
   enquanto não_se_verifica_condição_paragem
fim
```

Algoritmo 5.6 – Algoritmo genético genérico

Basicamente, estes algoritmos modificam repetidamente uma *população* de *elementos*, através da aplicação de mecanismos de *selecção* e de recombinação, pela aplicação de *operadores* genéticos. Cada elemento da população representa (de uma forma mais ou menos sintética) uma solução potencial para o problema de optimização, sendo quantitativamente avaliado por uma função, directamente relacionada com a função objectivo que se procura optimizar. Mediante a aplicação de mecanismos de selecção, a população evolui num sentido que tende a favorecer os elementos com melhor

valor da função objectivo, isto é, os elementos potencialmente mais promissores são preferencialmente reproduzidos, quando uma nova população é gerada. A recombinação entre elementos de uma população (através do operador de *cruzamento*) permite que seja passada informação "genética" dos elementos "pais" para os "descendentes", aquando da geração de novos elementos, para uma nova população. A "novidade" numa nova geração é conseguida, por exemplo, através do operador *mutação*.

A experiência parece demonstrar que a troca de informação útil permite criar em geral informação ainda mais útil. É da adopção deste "princípio" que advém o "poder" dos algoritmos genéticos, nomeadamente através da combinação de "bocados de soluções" que revelaram bom desempenho em gerações anteriores, com boas probabilidades de se virem assim a gerar melhores soluções.

Dada a analogia natural entre os AG e os sistemas biológicos, é habitual considerar-se nos AG, a terminologia usualmente empregue nestes últimos. Assim, cada elemento da população (geração) é designado por *cromossoma*. Um *cromossoma* é formado por *genes* que tomam valores designados por *alelos*. A posição de um gene num cromossoma é o seu *locus*.

Para além de aspectos relativos aos parâmetros do algoritmo e da estruturação da informação e respectivo modelo de programação, a implementação computacional dos algoritmos genéticos envolve os pontos fundamentais seguintes:

1. *codificação* – define a representação de um elemento (cromossoma) da população;
2. *geração da população inicial* – processo pelo qual é definido o conjunto de elementos que constitui a população inicial;
3. *selecção* – tal como no processo da selecção natural das espécies (função da sua maior ou menor capacidade de sobrevivência), procede-se à escolha dos elementos que se vão "reproduzir";
4. *manipulação genética* – permite a introdução de novas características na população, por aplicação de "operadores genéticos" aos elementos seleccionados tais como o operador de *cruzamento* e o operador de *mutação*;
5. *avaliação* – atribui a cada elemento da população, um valor que mede qualitativamente esse elemento relativamente à função objectivo que se pretende optimizar; a avaliação da qualidade de um cromossoma relativamente a to-

da a população, e resultante da aplicação de uma função de *adaptação* (que é frequentemente uma transformação linear da função objectivo), é fundamental para o processo de selecção dos elementos sujeitos a "reprodução".

5.6.3 Codificação

Os AG assentam num sistema de codificação que, em geral, a cada solução do problema faz corresponder um *cromossoma* com o qual se procura representar, de forma concisa e eficiente, as principais características da solução.

Geralmente, os cromossomas são sequências finitas de caracteres, pertencentes a um alfabeto igualmente finito, isto é, um cromossoma é uma concatenação de elementos do alfabeto. Holland (Holland, 1975) demonstra que a representação óptima para um cromossoma, é conseguida com um alfabeto de cardinalidade mínima, em virtude de ser maior a quantidade de informação passível de ser utilizada de forma eficiente pelo AG. Desta constatação, resulta que, tradicionalmente, uma parte significativa das aplicações de AG recorressem a codificações binárias, explorando a ideia fundamental de que uma representação simples pode caracterizar estruturas muito complexas (embora actualmente outras codificações sejam usadas frequentemente).

Goldberg concluiu (Goldberg, 1989) que uma codificação eficiente obriga a um compromisso entre a cardinalidade do alfabeto e a complexidade da codificação, entendida aqui como o facto de uma pequena alteração induzir uma mudança significativa na solução que representa.

Naturalmente que a codificação está directamente ligada a questões de eficiência no que respeita à manipulação da informação que é processada pelo algoritmo. Mas é igualmente importante, do ponto de vista de eficiência computacional, considerar a complexidade resultante das operações de codificação/descodificação, obrigatoriamente invocadas no decorrer do algoritmo sempre que é feito o mapeamento bidireccional, entre uma solução real do problema e o respectivo cromossoma.

Apesar de, tradicionalmente, uma grande parte das aplicações com algoritmos genéticos recorrerem a codificações binárias, têm sido, como referido, desenvolvidas diversas aplicações que utilizam codificações "mais naturais", isto é, mais próximas da forma como são representadas, noutros contextos, as soluções do problema. Se a codificação, nestas situações, pode resultar facilitada, implica, por outro lado e em geral,

a necessidade de implementar operadores genéticos especiais que se adequem a essa representação. A teoria que fundamenta a convergência dos algoritmos genéticos com codificações alternativas (não binárias) é apresentada em (Goldberg, 1991).

No caso concreto do problema de optimização em análise, optou-se por uma codificação baseada no alfabeto de números inteiros. Do ponto de vista do algoritmo genético, um cromossoma, solução do problema, é um vector de números inteiros de comprimento igual a duas vezes o número de estágios do processo produtivo. Cada gene representa uma data no horizonte de planeamento. Dois genes consecutivos, e com o *locus* do primeiro par, constituem a janela temporal (data início, data fim) que delimita o intervalo para a produção, a decorrer no estágio a que os genes estão associados.

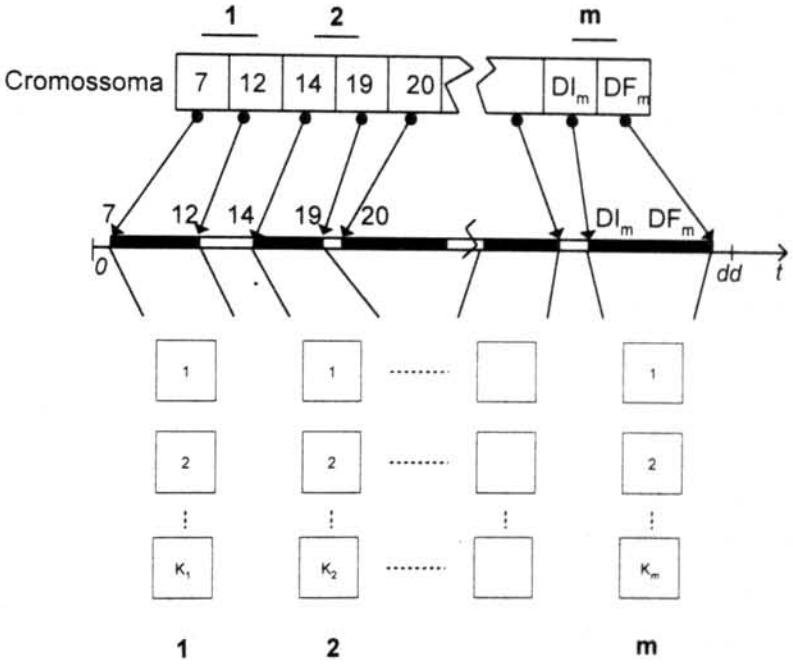


Figura 5.24 – Codificação de uma solução

A Figura 5.24 ilustra a codificação para uma configuração envolvendo *m* estágios. O primeiro par de genes, no exemplo apresentado, define a janela temporal para o primeiro estágio. Assim, a produção, neste estágio, estará compreendida entre o dia

7 e o dia 12. No segundo estágio, eventualmente por restrições logísticas associadas ao transporte entre estágios, a produção iniciar-se-á no dia 14, etc. Por forma a respeitar a data de entrega associada ao planeamento do pedido a que a solução diz respeito, o valor do último gene será geralmente menor ou igual a esta data, isto é, $DF_m \leq dd$.

Note-se que, ao utilizar este tipo de codificação, não estamos realmente a representar completamente uma solução. De facto, trata-se de uma representação sintética (não completa) com a qual, por exemplo, não se identifica a unidade de produção em cada estágio. Esta opção tem naturalmente a ver com a abordagem hierárquica adoptada: o AG vai apenas "resolver" o problema ao nível mais elevado.

5.6.4 Geração da população inicial

Um AG funciona, em cada momento, sobre uma população de cromossomas que evolui, de geração em geração, por aplicação de mecanismos de selecção e recombinação. Inicialmente é necessário proceder à criação de uma primeira população, segundo um qualquer processo aleatório, ou recorrendo a heurísticas que permitam introduzir na população inicial boas soluções, procurando dessa forma, acelerar o processo de pesquisa (Reeves, 1995).

No caso presente, utilizaram-se, em alternativa, as três heurísticas apresentadas em 5.4.2. Em particular, a utilização da heurística básica de geração de soluções iniciais de forma completamente aleatória (H-I) permitiu avaliar, de uma forma mais independente, o desempenho dos operadores genéticos, na medida em que a evolução de uma população inicial até à solução final depende assim, unicamente dos mecanismos de selecção e de recombinação.

5.6.4.1 Tamanho da população

O tamanho a considerar para a população é um factor que condiciona directamente o desempenho do algoritmo. Uma população demasiado reduzida, constituindo uma amostragem deficiente do espaço de soluções, pode conduzir prematuramente o algoritmo para uma solução de qualidade inferior (afastada significativamente da solução óptima), isto é, verifica-se a convergência rápida para óptimos locais. Populações demasiadamente grandes, evitando de certa forma o problema de uma convergência prematu-

ra, conduzem naturalmente a problemas de desempenho, ao nível do tempo de processamento.

Em (Mahfoud, 1995b) é feito um estudo experimental acerca do efeito da dimensão da população no desempenho de algoritmos genéticos.

5.6.5 Mecanismos de selecção

Nos AG, o mecanismo de selecção sobre os elementos da população actual, para efeitos de recombinação, desempenha um papel fundamental no processo de pesquisa do espaço de soluções. Deve ser conduzido no sentido de privilegiar as melhores soluções, por um lado, e na manutenção de uma diversidade significativa de cromossomas, por outro. Esta é naturalmente, uma questão mais geral que tem a ver com o compromisso entre conseguir elevada rapidez no processo de convergência e elevada probabilidade em encontrar uma solução de valor muito próximo do óptimo global.

Basicamente, através do processo de selecção, pretende-se garantir maior probabilidade de "sobrevivência" aos elementos com maior "adaptação" ao meio. Far-se-á de seguida uma breve referência aos principais mecanismos de selecção implementados na prática e extensivamente utilizados em diferentes configurações de AG. Para além da bibliografia de base acerca dos AG, o artigo de Back e Hoffmeister (Back and Hoffmeister, 1991) apresenta formalmente diversos mecanismos de selecção.

No que se segue, é usada a seguinte notação:

Seja S o espaço de soluções admissíveis do problema, e P_n a população de M elementos da geração n, tal que $P_n = \left(x_1^n, \cdots, x_M^n\right)$ e $P_n \subset S$, e em que x_i^n é um elemento da população na geração n. A função de adaptação (*"fitness"*) é representada por $F : S \rightarrow \Re$, sendo $\overline{F}$ e $F_{\max}$ respectivamente o valor médio e o valor máximo da função de adaptação para uma determinada população.

5.6.5.1 Selecção por "roleta viciada"

Este processo estocástico de selecção, dos mais utilizados nas implementações tradicionais, é baseado no conceito de selecção proporcional. Este processo é referido usualmente por "roleta viciada", pela analogia que se verifica com uma roleta em que cada ranhura ou "fatia" apresentasse uma dimensão proporcional ao valor da função de

adaptação associado a cada elemento da população, e em que o número de ranhuras ou fatias fosse igual à dimensão da população.

De cada vez que se pretende escolher um elemento da população, faz-se rodar a roleta, sendo estão seleccionado o elemento correspondente à ranhura em que a roleta parou. Assim, elementos com maior valor da função de adaptação têm maiores probabilidades de serem escolhidos, em virtude da maior dimensão da ranhura respectiva. A probabilidade de selecção associada a cada elemento é, assim, dada por:

$$p(x_i^n) = \frac{F(x_i^n)}{\sum_{j=1}^{M} F(x_i^n)}$$

A Figura 5.25 ilustra este mecanismo de selecção.

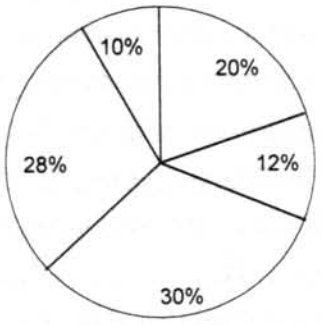


Figura 5.25 – Exemplo do mecanismo de selecção tipo "roleta viciada"

Identificam-se neste mecanismo de selecção, devido à aleatoriedade do processo, dois problemas principais:

- Existe a possibilidade de um elemento pior da população ser seleccionado um número de vezes superior a outros elementos com maior valor da função de adaptação. Além disso, pode igualmente acontecer que o melhor elemento da população não seja sequer seleccionado (estes problemas são referidos na literatura por "erros estocásticos");

- Pode verificar-se um problema de convergência prematura para um óptimo local, devido à "persistente" selecção de alguns elementos da população (os que apresentam maior valor da função de adaptação), provocando assim, uma diminuição da "diversidade genética" da população (Lutton and Martinez, 1993).

Com a finalidade de "regular" essa diversidade genética, especialmente nas gerações iniciais, e procurando evitar a convergência prematura da pesquisa, é feita a seguinte transformação linear da função de avaliação original:

$$F'(x_i^n) = k_1 \cdot F(x_i^n) + k_2$$

em que k_1 e k_2 são constantes, calculadas em cada geração, por forma a que $\overline{F'} = \overline{F} \quad \wedge \quad F'_{max} = C \cdot \overline{F}$, em que C é uma constante tipicamente com valores compreendidos entre 1.2 e 2. As constantes k_1 e k_2 são calculadas a partir das expressões seguintes:

$$k_1 = \frac{(C-1)\cdot \overline{F}}{F_{max} - \overline{F}} \quad \wedge \quad k_2 = \frac{\overline{F} \cdot (F_{max} - C \cdot \overline{F})}{F_{max} - \overline{F}}$$

A ideia base desta transformação linear, $F \rightarrow F'$, é a de assegurar que um elemento com um valor da função de adaptação igual ao valor médio do valor da adaptação da população, contribua para a geração seguinte com um descendente. Por outro lado, com a definição do valor máximo F'_{max} pretende-se controlar o número de descendentes obtidos a partir do melhor elemento da população.

5.6.5.2 "Torneio estocástico"

Este processo de selecção é baseado na "roleta viciada". Basicamente, para cada elemento novo a gerar são seleccionados, através do processo de roleta viciada, dois elementos da população. Destes dois elementos é escolhido o melhor (isto é, aquele que apresenta um maior valor da função de adaptação) com uma determinada probabilidade parametrizável.

5.6.5.3 Selecção proporcional ou determinística

Neste processo de selecção, o número de descendentes, associado a cada elemento da população, é determinado com base no número esperado de descendentes. Para cada

elemento, este número é calculado a partir da parte inteira do produto $M \cdot p(x_i^n)$ em que $p(x_i^n)$ representa a probabilidade associada ao elemento x_i^n de acordo com o exposto na secção anterior e M é igual à dimensão da população. Assim, a um elemento com um valor da função de adaptação superior à média é atribuído mais do que um descendente. Os elementos que faltem para totalizar o número de membros da população, são geralmente seleccionados a partir da implementação de uma "roleta viciada" com pesos proporcionais às partes fraccionárias dos quocientes dos valores esperados.

5.6.5.4 Selecção por posição (*rank*)

Nesta técnica de selecção, recorre-se, para calcular o número de descendentes, às posições de todos os elementos da população, numa ordenação de acordo com o valor da sua adaptação. São utilizadas diversas "funções de atribuição", destacando-se a função linear proposta por Baker (Baker, 1985) em que o valor esperado do número de descendentes para o elemento x_i^n é dado por:

$$e(x_i^n) = \frac{1}{M}\left(\lambda_{\max} - (\lambda_{\max} - \lambda_{\min}) \cdot \frac{rank(x_i^n) - 1}{M - 1}\right)$$

em que $\lambda_{\max}$ é uma constante tal que $1 \le \lambda_{\max} \le 2$ e $\lambda_{\min} = 2 - \lambda_{\max}$. A *posição* do elemento x_i^n é dada pela função $rank(x_i^n)$, tal que:

$$\forall i \in \{1,\ldots,M\} : rank(x_i^n) = i \quad \Leftrightarrow \quad \forall j \in \{1,\ldots,M-1\} : F(x_j^n) \le F(x_{j+1}^n).$$

5.6.6 Operadores genéticos

Os elementos seleccionados para reprodução são recombinados, dando origem a descendentes que irão compor a população da geração seguinte. Este processo de recombinação é feito tradicionalmente através de dois operadores genéticos: o operador de *cruzamento* (*crossover*) e o operador de *mutação* (*mutation*) que actuam sobre o conjunto de elementos seleccionados para reprodução, de acordo com probabilidades previamente definidas.

5.6.6.1 Operador de cruzamento

O operador de cruzamento actua sobre um par de cromossomas do grupo reprodutor ("pais"), originando, geralmente, dois "descendentes" formados cada um com genes dos dois pais, procurando-se assim que os novos elementos reúnam as boas características dos que lhe deram origem.

Na versão clássica deste operador, existe um único ponto de cruzamento, isto é, é escolhido aleatoriamente, de acordo com uma distribuição uniforme, a posição de um gene: o primeiro filho é constituído pela primeira parte do primeiro pai e pela segunda parte do segundo; por sua vez, o segundo filho é "gerado" de uma forma complementar (ver Figura 5.26). Variantes deste operador incluem o operador de cruzamento duplo (dois pontos de cruzamento) e o operador de cruzamento uniforme (equivalente no limite a ter um número de pontos de cruzamento igual ao número de genes do cromossoma).

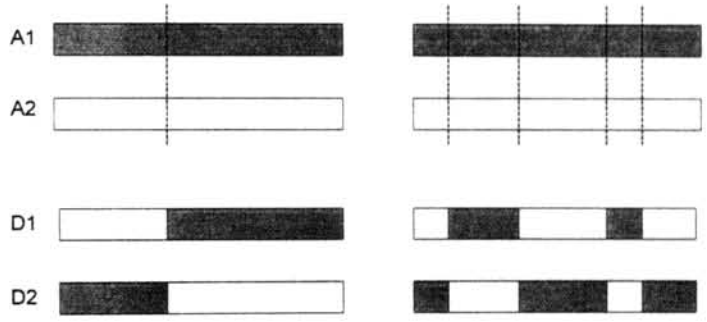


Figura 5.26 – Operador de cruzamento simples e múltiplo

A prática parece demonstrar que o desempenho dos algoritmos diminui à medida que o número de pontos de cruzamento aumenta, pelo que os operadores de cruzamento simples e duplo são geralmente os mais utilizados.

A frequência de aplicação do operador de cruzamento depende do valor definido para um parâmetro do algoritmo, previamente definido, que é a *probabilidade de cruzamento*. Elementos que não sofram qualquer cruzamento passam integralmente para a nova população.

Em princípio, o operador de cruzamento acelera a convergência da população para soluções com elevado valor da *função de adaptação* (Lutton and Martinez, 1993).

Valores demasiadamente elevados para a probabilidade de cruzamento podem originar a *convergência prematura*, assim como podem provocar um efeito de disrupção, isto é, não se propagam para as gerações seguintes esquemas[34] intactos (Ranito, 1993). Existe, por isso, um compromisso na escolha do valor dessa probabilidade. Os valores típicos para a probabilidade de cruzamento situam-se no intervalo entre 0.5 e 0.9.

Tradicionalmente, a aplicação do operador de cruzamento garante a admissibilidade das soluções geradas, isto é, os descendentes respeitam o conjunto de restrições do problema.

No âmbito deste trabalho, implementaram-se quatro operadores de cruzamento. Dois dos operadores possuem dois pontos de cruzamento, sendo num caso ambos os pontos fixados previamente, e no outro, ambos os pontos variáveis (aleatoriamente). O terceiro operador considerado apresenta um único ponto de cruzamento escolhido aleatoriamente. O quarto operador implementado (operador misto) é baseado na combinação "controlada" de dois operadores: o operador de cruzamento duplo (variável) e o operador com um único ponto de cruzamento.

Os operadores com dois pontos de cruzamento, implementados neste trabalho, encontram-se descritos no Algoritmo 5.7. Este algoritmo recebe como argumentos dois cromossomas seleccionados para recombinação (P1 e P2) e gera dois cromossomas (D1 e D2).

A Figura 5.27 apresenta um exemplo de aplicação do operador com dois pontos de cruzamento numa situação em que os cromossomas possuem 8 genes, ou seja, representam soluções para um problema com quatro estágios. Na figura estão representadas todos os resultados possíveis, para além da possibilidade de os descendentes serem uma cópia integral dos pais (P1 e P2).

---

[34] No âmbito dos algoritmos genéticos, cada cromossoma pode ser interpretado como uma representação de um grande número de *padrões* ou *esquemas*. Um *esquema* é um padrão de semelhança que descreve um conjunto de cadeias com similaridades em determinadas posições. Num alfabeto binário um esquema poderia ser , por exemplo, **11001, em que o carácter * pode representar qualquer carácter do alfabeto.

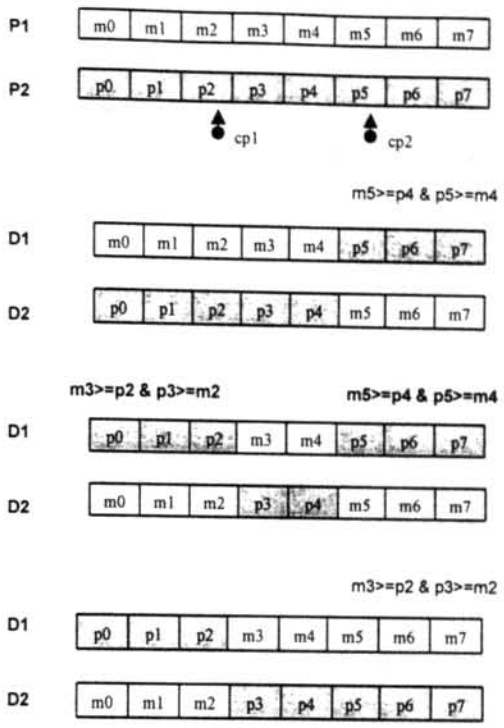


Figura 5.27 – Exemplo de aplicação do operador de cruzamento

O algoritmo apresenta basicamente três fases. Na primeira fase, e por análise dos valores dos genes correspondentes ao primeiro ponto de cruzamento, verifica-se a possibilidade de se permutar os genes entre P1 e P2, desde o primeiro até ao primeiro ponto de cruzamento. Na segunda fase, o algoritmo limita-se a copiar, entre pontos de cruzamento, os valores dos genes de P1 e P2 directamente para D1 e D2. A terceira e última fase é similar à primeira mas refere-se naturalmente aos genes situados a partir do segundo ponto de cruzamento, inclusive.

```
algoritmo cruzamento

//P1, P2 cromossomas a recombinar
//D1, D2 descendentes resultantes
//cp1, cp2 ponto de cruzamento inferior e superior

se (P1[cp1]<=P2[cp1+1] e P2[cp1]<=P1[cp1+1])
  então
    para i=0 até cp1
      D1[i]=P2[i];
```

```
            D2[i]=P1[i];
      senão
         para i=0 até cp1
            D1[i]=P1[i];
            D2[i]=P2[i];

   para i=cp1+1 até cp2
            D1[i]=P1[i];
            D2[i]=P2[i];

   se (P1[cp2]>=P2[cp2-1] e P2[cp2]>=P1[cp2-1])
      então
         para i=cp2 até numEstagios*2-1
            D1[i]=P2[i];
            D2[i]=P1[i];
      senão
         para i=cp2 até numEstagios*2-1
            D1[i]=P1[i];
            D2[i]=P2[i];
   fim.
```

Algoritmo 5.7 – Estrutura geral do operador de cruzamento (com dois pontos de cruzamento)

5.6.6.2 Operador de mutação

Com este operador a ideia é criar, embora com reduzida probabilidade, alguma "perturbação" na população gerada. Procura-se, desta forma, introduzir informação adicional às soluções em análise, com a finalidade de "diversificar a pesquisa" e de se explorarem outras zonas do espaço de soluções, contribuindo assim para garantir alguma diversidade na população da geração seguinte.

A probabilidade de mutação deve ser baixa, com a finalidade de se evitar que o algoritmo apresente um comportamento próximo do de uma pesquisa puramente aleatória. Tradicionalmente, a probabilidade de mutação é considerada constante ao longo de todo o processo; contudo, diferentes estratégias têm sido estudadas a fim de se acentuar a característica de *diversificação* na pesquisa do espaço de soluções numa fase inicial do algoritmo, passando gradualmente para uma fase de *intensificação* (Lutton and Martinez, 1993).

```
algoritmo mutação

//C - cromossona sujeito a mutação
//primeiro gene
p=numAleatório(0,1);
se p<=0.5
   então
      C[0]=C[0]+numAleatório(0,(C[1]-C[0]));
   senão
      C[0]=C[0]-numAleatório(0,C[0]);

i=1;
Repete //do segundo ao penúltimo gene
   T_transp=t_move(int(i/2+1));
   se (C[i+1]-T_transp-C[i])>1)
      então
         nova_data=(C[i+1]- T_transp-C[i])/2 + C[i];
      senão
         p=numAleatório(0,1);
         se p<=0.5
            então
               nova_data=(C[i+1]- T_transp-C[i])/2 + C[i]
                     + numAleatório(0,(C[i+2]-C[i+1]));
            senão
               nova_data=(C[i+1]- T_transp-C[i])/2 + C[i]
                     - numAleatório(0,(C[i]-C[i-1]));
   C[i]= nova_data;
   C[i+1]= nova_data+ T_transp;
   i=i+2;
enquanto(i<=2*(numEstagios-1))

//último gene
k=2*numEstagios-1;
p=numAleatório(0,1);
se p<=0.5
   então
      C[k]=c[k]+numAleatório(0,(due_date-C[k]));
   senão
      C[k]=c[k]+numAleatório(0,(C[k]-C[k-1]));
fim.
```

Algoritmo 5.8 – Estrutura geral do operador de mutação

O operador de mutação implementado neste trabalho está descrito no Algoritmo 5.8. Note-se que a aplicação deste operador garante a admissibilidade das soluções geradas.

Basicamente, o operador actua na data de conclusão associada a cada estágio, deslocando-a para trás ou para a frente, de forma aleatória, e respeitando a restrição do tempo de transporte entre estágios consecutivos. A Figura 5.28 ilustra a aplicação do operador, num cromossoma composto por oito genes e correspondente a um sistema com quatro estágios. A data de início do primeiro estágio e a data de fim do último estágio são limitadas, respectivamente, à data actual e à *due-date*.

Foi também implementada uma variante deste operador, em que apenas ocorre a alteração da data de conclusão ou da data de início, de um estágio seleccionado aleatoriamente.

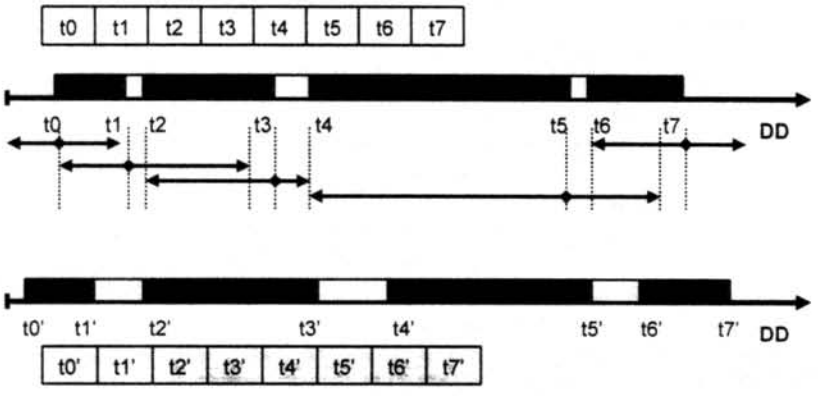


Figura 5.28 – Exemplo de aplicação do operador de mutação

5.6.6.3 Avaliação das soluções

A função objectivo considerada para a avaliação individual de cada cromossoma da população, é a que foi apresentada nas secções anteriores, nomeadamente, em 5.5.6. A *função de adaptação* ("*fitness*"), que mede a qualidade de cada cromossoma relativamente a toda a população, consiste, no caso geral, numa normalização linear da função objectivo.

A função de adaptação é fundamental para o processo de selecção. No caso geral, a taxa de convergência do algoritmo é dependente da forma como é feita a avaliação de cada elemento da população. Em particular, no caso de populações com um

reduzido número de elementos, esta dependência pode tornar-se muito significativa. Assim, se se considerar directamente a função objectivo na avaliação das soluções, a questão põe-se quando a geração da população inicial origina um conjunto de elementos que globalmente se possa considerar de fraca qualidade. Apesar da diferença entre os elementos poder ser pouco significativa, pode verificar-se a predominância dos que são (ligeiramente) melhores, nas gerações seguintes (em resultado dos mecanismos de selecção), podendo conduzir o algoritmo, prematuramente, para uma solução que é, em termos globais, de fraca qualidade.

Uma forma de limitar este efeito de convergência prematura é através da utilização de uma *função de adaptação* ("fitness"), com a finalidade de "normalizar" a função de avaliação, por forma a diminuir a diferença entre o valor da função de avaliação do melhor e do pior elemento da população, quando o intervalo entre esses valores é "excessivo". Na situação oposta, em que os elementos apresentam valores da função de avaliação muito próximos, a função de adaptação deve aumentar a diferença entre esses elementos, permitindo, dessa forma, uma maior discriminação entre esses elementos.

Note-se que na resolução do problema com algoritmos genéticos, considerou-se igualmente uma abordagem hierárquica, com a segmentação do problema em diferentes níveis (ver Figura 5.5).

O critério de paragem utilizado foi o de terminar o algoritmo ao fim de um determinado número de gerações.

5.6.7 Tipos de algoritmos genéticos

Para além das diferentes técnicas de selecção apresentadas anteriormente, a escolha da estratégia de substituição constitui uma decisão fundamental, quando se verifica uma tendência para a convergência prematura.

A estratégia de substituição da população actual, total ou parcial, conduz naturalmente a algoritmos com desempenhos que podem ser significativamente diferentes. Descreve-se de seguida as implementações efectuadas neste trabalho, nomeadamente o tipo *simples*, o tipo *steady-state* e o tipo *incremental*.

Algoritmo Simples

Esta versão implementa a configuração clássica do algoritmo descrita em (Goldberg, 1989). Em cada geração, é criada uma população completamente nova, a partir da aplicação dos mecanismos de selecção e de recombinação aos elementos da geração precedente. Não existe, por isso, na nova população, qualquer sobreposição com a população anterior. Na implementação efectuada, existe a possibilidade de activar, no mecanismo de selecção, um comportamento "elitista" em que o(s) melhor(es) membro(s) da população passa(m) obrigatoriamente para a geração seguinte, não sendo sujeitos aos operadores de cruzamento ou de mutação. A opção de preservar o melhor elemento da população (elitismo) garante, por si só, uma convergência monótona do algoritmo.

Algoritmo Steady-State

Considerou-se, aqui, uma implementação similar à versão descrita em (Jong, 1975). Esta versão do algoritmo baseia-se numa estratégia de substituição parcial da população, em que o nível de sobreposição entre gerações sucessivas é parametrizável. Assim, para fazer parte da nova população, são directamente escolhidos um determinado número de elementos da geração anterior. Os restantes elementos, necessários para completarem a população, são obtidos pelos processos tradicionais de recombinação.

Algoritmo Incremental

Esta versão gera unicamente um ou dois descendentes "novos" em cada nova população. Na geração destes descendentes, a inserir na nova população, diversos mecanismos de substituição podem ser utilizados, nomeadamente: substituição aleatória, substituição do pior elemento, substituição do melhor elemento, etc.

5.6.8 Experiências computacionais

5.6.8.1 Instâncias teste

Apresentam-se nesta secção, resultados computacionais sobre três instâncias, caracterizadas genericamente na subsecção 5.5.7.1 e que têm por base a estrutura produtiva descrita no Apêndice A. As características destas instâncias, no que se refere à componente pedido a planear, encontram-se na Tabela 5.1.

Instância	*due-date*	quantidade	folga	penalização
A	80	400	2	2.0
B	50	400	2	2.0
C	120	400	2	2.0

Tabela 5.1 – Instâncias teste (pedidos a planear)

Estas instâncias serviram para comparar directamente o desempenho das duas meta-heurísticas implementadás no âmbito deste trabalho.

5.6.8.2 Parametrização do algoritmo

Tal como no *simulated annealing*, e pelas razões então apresentadas, os testes basearam-se em 30 corridas do algoritmo, partindo sempre de diferentes populações iniciais (obtidas por heurísticas construtivas, já de si com comportamento aleatório).

A escolha dos parâmetros do algoritmo foi efectuada após um conjunto alargado de experiências, com diferentes combinações de valores e sobre diferentes instâncias, que permitiram, numa fase preliminar, ganhar sensibilidade quanto ao comportamento e desempenho do algoritmo.

Geração da população inicial

Numa fase preliminar de testes, utilizou-se para geração dos elementos da população inicial, as heurísticas de geração de soluções iniciais apresentadas em 5.4.2 e aqui referenciadas por H-I, H-II e H-III. O conjunto de experiências realizadas com estas heurísticas, no caso do *simulated annealing*, permitiu concluir que a heurística H-I dá origem a soluções iniciais em geral de menor qualidade, mas simultaneamente mais variadas. Por outro lado, as heurísticas H-II e H-III dão origem a soluções em geral qualitativamente melhores, mas menos variadas.

Como referido, as meta-heurísticas caracterizam-se por uma fase inicial de diversificação que explora, com um maior grau de aleatoriedade, o espaço de soluções. A população inicial deverá ser, por isso, tão representativa quanto possível do espaço de soluções, procurando ter-se em simultâneo, soluções variadas e com grande qualidade, com a finalidade de evitar a convergência prematura. Optou-se, neste sentido, por desenvolver uma heurística para geração dos elementos da população inicial, baseada numa combinação das três heurísticas básicas já referidas. Assim, consegue-se que na mesma população, se tenha elementos representativos de todo o espaço de soluções.

No caso presente, e após testes intensivos, implementou-se uma heurística para geração da população inicial "configurada" para gerar uma população em que 50% dos elementos são obtidos a partir da heurística H-I, 35% a partir da heurística H-II e os restantes 15% a partir da heurística H-III.

Processo de selecção

O processo de selecção dos elementos reprodutores é de importância crucial na convergência do algoritmo. De todos os processos implementados, e anteriormente descritos, a "roleta viciada" foi a que permitiu obter melhores resultados.

Estratégia de substituição das populações

A escolha da estratégia de substituição das populações é uma decisão fundamental, sobretudo quando se verifica uma tendência clara para a convergência pre-

matura. No caso presente, experimentaram-se diferentes estratégias, de acordo com os tipos de algoritmos descritos em 5.6.7. A fraca qualidade dos resultados, obtidos em testes preliminares, com a implementação do algoritmo incremental, face às implementações do algoritmo simples e do algoritmo *steady-state*, levou a que se considerassem apenas as duas seguintes possibilidades:

- geração total de uma nova população (isto é, sem sobreposição com a população anterior) com manutenção do melhor elemento da população anterior (elitismo);

- substituição parcial da população, com o nível de sobreposição entre gerações sucessivas parametrizável; neste processo são substituídos os piores $n*g$ elementos, sendo n, o número de elementos da população, e g, o grau de substituição considerado.

Apresentam-se, a seguir, os resultados computacionais obtidos num conjunto de experiências, efectuadas com a finalidade de determinar o impacto das estratégias de substituição das populações. Nestas experiências, considerou-se uma população com 50 elementos e 400 gerações, tendo-se utilizado um operador de cruzamento com dois pontos de cruzamento fixos e um operador de mutação múltiplo.

Para cada uma das instâncias, são apresentados, sob a forma gráfica, os valores médios do custo global da solução final obtida, utilizando diferentes graus de substituição da população para o algoritmo *steady-state*, nomeadamente 20%, 40%, 60% e 80% e na situação de substituição total da população (grau de substituição de 100%), para o algoritmo simples com a manutenção do melhor elemento. São apresentados igualmente os tempos médios de processamento envolvidos numa corrida do algoritmo.

Instância A

Para a instância A, como se pode observar na Figura 5.29, a estratégia de substituição total (grau de substituição de 100%) com manutenção do melhor elemento (elitismo) produz resultados significativamente melhores do que a estratégia de substituição parcial em regime estacionário (algoritmo *steady-state*). O tempo médio de processamento envolvido e representado graficamente na Figura 5.30 é, como previsto, maior à medida que o grau de substituição da população aumenta.

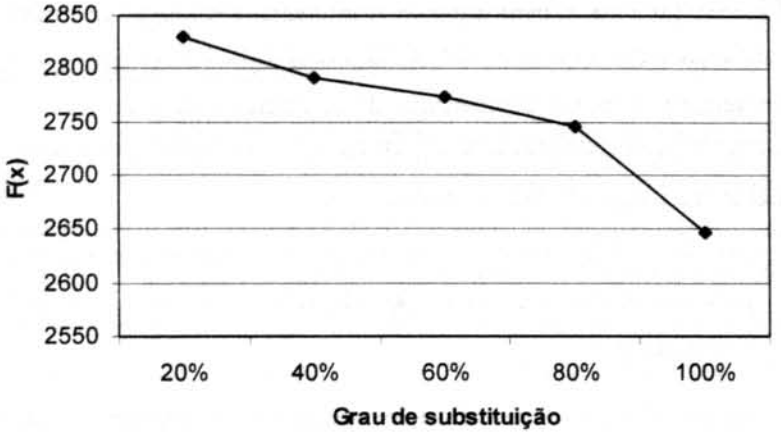


Figura 5.29 – Impacto do grau de substituição da população na solução final (instância A)

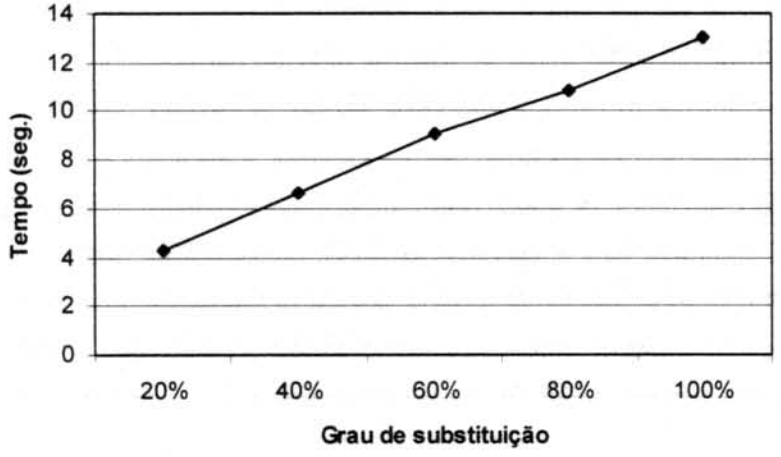


Figura 5.30 – Tempo de processamento em função do grau de substituição da população (instância A)

A Tabela 5.2 resume os resultados obtidos no conjunto de testes efectuados com a instância A. Para além de se verificar que em termos médios as soluções são melhores para a estratégia de substituição total (com elitismo), consegue-se igualmente

garantir, na mesma situação, uma menor variabilidade para a qualidade das soluções finais.

F(x)	20%	40%	60%	80%	100%
média	2829.4	2791.5	2774.1	2745.8	2646.9
desvio padrão	62.1	56.0	58.4	55.5	34.4
mín.	2618.1	2604.4	2602.4	2660.4	2548.4
máx.	2904.4	2864.4	2856.4	2834.8	2708.4

Tabela 5.2 – Resultados computacionais (instância A)

Instância B

No conjunto de experiências com a instância B, obtiveram-se igualmente melhores resultados com um grau de substituição da população mais elevado, tendo-se obtido igualmente melhores resultados para a estratégia de substituição total com elitismo. A Figura 5.31 ilustra a evolução verificada no valor médio do custo global das soluções obtidas para o conjunto de experiências realizadas.

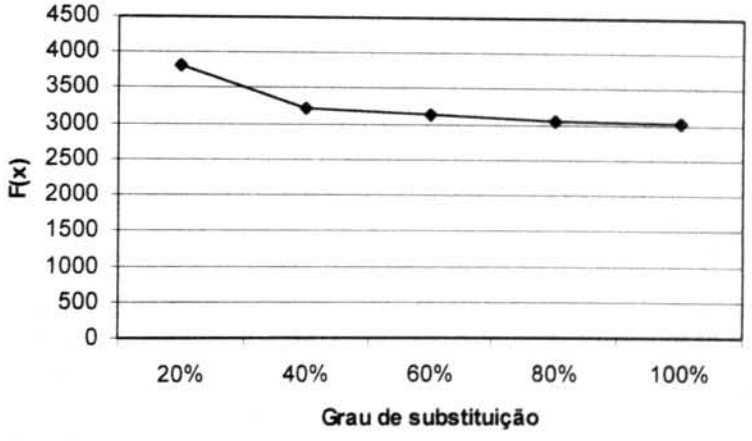


Figura 5.31 – Impacto do grau de substituição da população na solução final (instância B)

A análise do gráfico mostra que a melhoria não é tão significativa como a verificada para a instância A. Na verdade, a instância B, relativamente à instância A, apresenta uma *due-date* significativamente inferior, o que condiciona, por si só, as alternati-

vas possíveis para o planeamento. Nesta situação, verificam-se maiores dificuldades, quer na obtenção das soluções iniciais, quer na qualidade das próprias soluções, com uma muito maior variabilidade dessa qualidade.

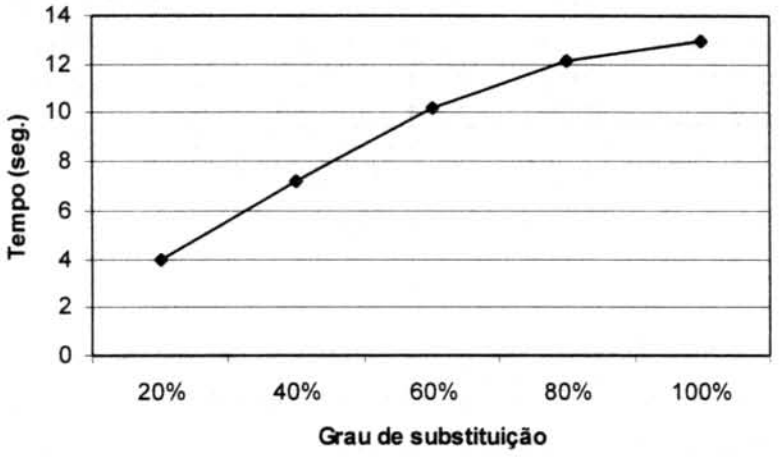


Figura 5.32 – Tempo de processamento (instância B)

O tempo médio de processamento envolvido, representado graficamente na Figura 5.30, apresenta de igual forma ao sucedido com a instância A um crescimento monótono, com o grau de substituição da população.

A Tabela 5.3 resume os resultados obtidos no conjunto de testes efectuados com a instância B.

F(x)	20%	40%	60%	80%	100%
média	3819.7	3203.1	3115.4	3033.1	3012.1
desvio padrão	918.1	247.5	187.1	116.4	344.2
mín.	2987.5	2810.6	2793.1	2750.9	2703.5
máx.	6838.9	3651.3	3440.1	3263.3	3917.8

Tabela 5.3 – Resultados computacionais (instância B)

Instância C

Os resultados obtidos com a instância C confirmam, uma vez mais, que se obtêm melhores resultados com uma estratégia de substituição total da população (mas mantendo o melhor elemento).

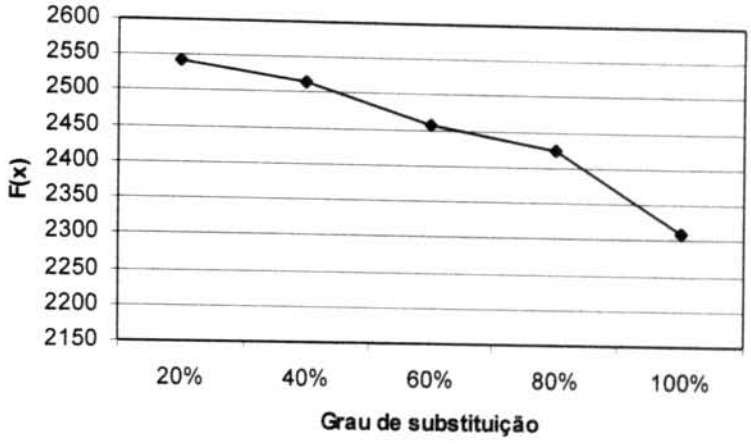


Figura 5.33 – Impacto do grau de substituição da população na solução final (instância C)

A análise do gráfico da Figura 5.33 evidencia, novamente, uma melhoria significativa com a estratégia de substituição total da população, em virtude de esta instância apresentar uma *due-date* significativamente superior e, em consequência, permitir mais alternativas de planeamento

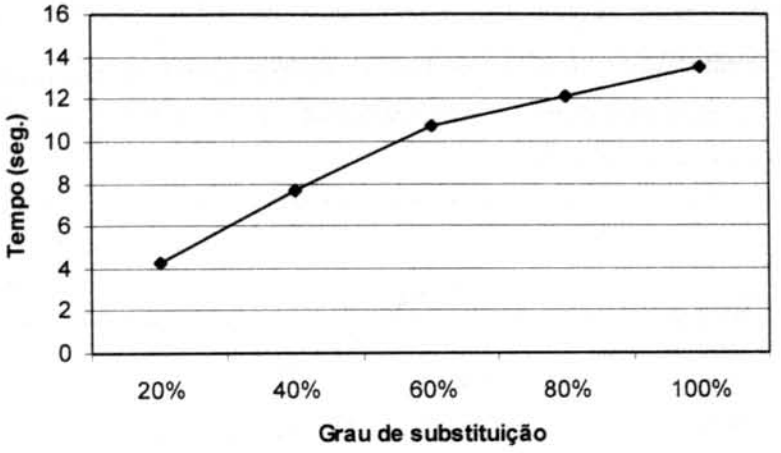


Figura 5.34 – Tempo de processamento (instância C)

F(x)	20%	40%	60%	80%	100%
média	2543.4	2516.3	2458.6	2425.6	2309.9
desvio padrão	136.0	121.3	88.5	102.0	55.0
mín.	2268.9	2280.9	2290.1	2210.1	2210.1
máx.	2780.7	2748.9	2600.4	2580.9	2418.1

Tabela 5.4 – Resultados computacionais (instância C)

Em face dos resultados obtidos para as três instâncias, seleccionou-se para o algoritmo, uma estratégia de substituição total da população, com preservação do melhor elemento.

Operador cruzamento

Dos operadores de cruzamento experimentados, o operador duplo com pontos de cruzamento fixos e o operador simples (com um único ponto de cruzamento mas variável) revelaram-se pouco eficazes, podendo, em certo sentido, serem considerados como casos particulares do operador misto implementado e descrito na subsecção 5.6.6.1.

Fez-se um conjunto de experiências para o operador misto, com diferentes graus de combinação entre o operador duplo e o operador simples (ambos com

pontos de cruzamento seleccionados aleatoriamente). O gráfico apresentado na Figura 5.35 representa, para diferentes graus de combinação dos dois operadores, o valor médio do custo global associado às soluções finais obtidas (limita-se a apresentação dos resultados à instância A, já que para as instâncias B e C, os resultados foram semelhantes). Assim, um grau de combinação de 70%-30% significa que, no processo de recombinação de elementos, em 70% dos casos é utilizado o operador de cruzamento duplo e nos restantes 30% o operador de cruzamento simples.

No mesmo gráfico apresenta-se igualmente o resultado obtido com o operador de cruzamento com dois pontos de cruzamento fixos (representado no gráfico por Fixo).

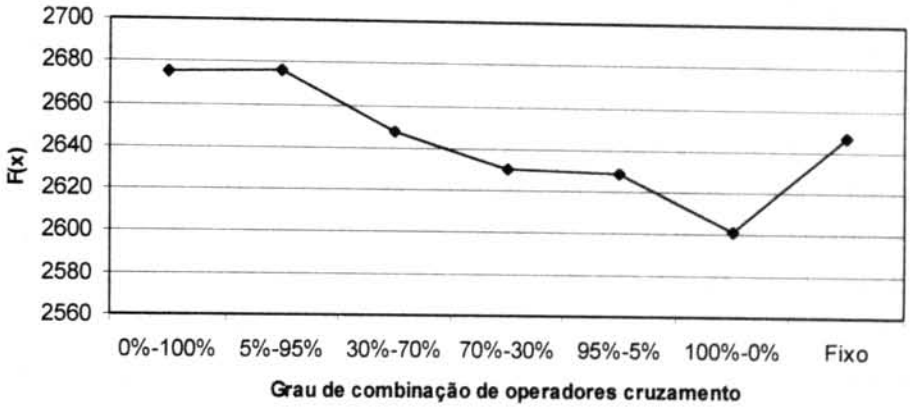


Figura 5.35 – Efeito da combinação de operadores de cruzamento

A análise do gráfico evidencia a melhoria dos resultados com o aumento da "influência" do operador de cruzamento duplo face ao operador de cruzamento simples.

Os resultados apresentados, conjuntamente com inúmeros outros testes realizados, permitem concluir que a qualidade das soluções finais obtidas é em geral melhor quando o operador de cruzamento é baseado em dois pontos de cruzamento definidos aleatoriamente.

A Tabela 5.5 resume os resultados obtidos para a instância A.

F(x)	0%-100%	5%-95%	30%-70%	70%-30%	95%-5%	100%-0%	Fixo
média	2675.4	2675.6	2647.6	2630.1	2628.8	2601.3	2646.9
desvio padrão	46.4	22.6	34.8	46.3	34.2	33.1	34.4
mín.	2588.4	2632.4	2568.4	2548.4	2556.4	2550.1	2548.4
máx.	2774.0	2710.1	2712.4	2692.4	2672.4	2660.4	2708.4

Tabela 5.5 – Resumo dos resultados computacionais na avaliação dos operadores de cruzamento

Dimensão da população e número de gerações

Para efeitos da determinação da dimensão da população e do número de gerações a considerar, e partindo dos resultados dos testes apresentados anteriormente, efectuou-se um novo conjunto de experiências, utilizando novamente todas as instâncias (A, B e C) com 30 populações iniciais. Para gerar a população inicial, utilizou-se a heurística mista e com a seguinte combinação: 50% de H-I, 35% de H-II e 15% de H-III; para processo de selecção, usou-se o método da "roleta viciada"; e para estratégia de substituição das populações, considerou-se substituição total com manutenção do melhor elemento. Além disso, utilizou-se o operador de cruzamento com dois pontos de cruzamento aleatórios, e probabilidade de cruzamento de 0.8. O operador de mutação considerado foi o múltiplo (alteração de todos os genes) com probabilidade de 0.05.

Fizeram-se testes com os seguintes valores para a dimensão da população: 10, 25, 50 e 100. Para o número de gerações foram experimentados os seguintes valores: 50, 100, 200, 400 e 800. Para cada uma das instâncias, são apresentados os valores médios do custo global da solução obtida, partindo de diferentes dimensões da população e utilizando diferentes números de gerações. São apresentados igualmente os tempos médios de processamento envolvidos numa corrida do algoritmo.

Sob a forma de tabela, apresenta-se também, para cada uma das instâncias consideradas, informação estatística acerca da solução final (em função da dimensão da população e do número de gerações consideradas).

Instância A

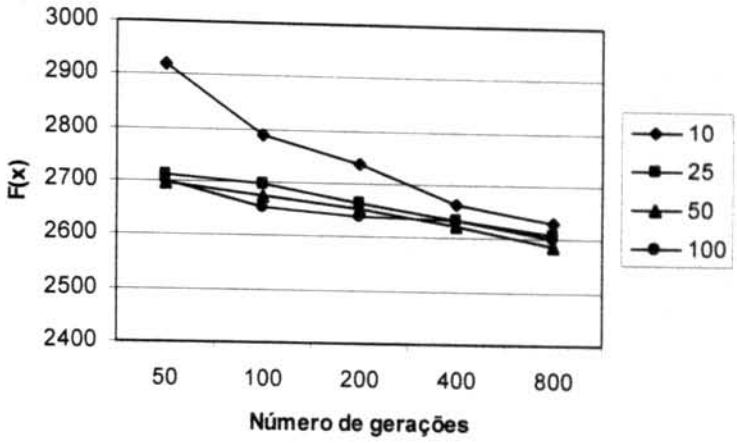


Figura 5.36 – Avaliação em função da dimensão da população e do número de gerações (instância A)

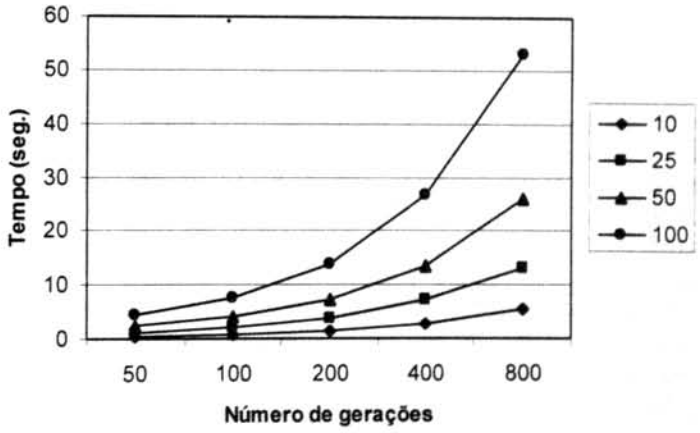


Figura 5.37 – Tempo de processamento em função da dimensão da população e do número de gerações (instância A)

Dim. Popul.	F(x)	Número de gerações 50	100	200	400	800
	média	2919.8	2787.8	2737.2	2664.0	2629.9
10	desvio padrão	184.7	83.9	85.5	54.6	42.0
	mín.	2595.5	2624.4	2572.4	2534.7	2568.4
	máx.	3219.6	2968.4	2940.4	2772.4	2730.1
	média	2712.4	2698.2	2664.4	2634.2	2608.9
25	desvio padrão	54.1	44.2	51.0	43.4	33.6
	mín.	2608.4	2624.4	2578.1	2556.4	2548.4
	máx.	2810.1	2790.1	2740.4	2692.4	2668.4
	média	2696.3	2673.6	2653.0	2624.0	2588.1
50	desvio padrão	43.8	43.4	42.2	36.6	29.6
	mín.	2580.4	2572.4	2556.4	2547.3	2530.1
	máx.	2774.8	2732.4	2704.4	2664.4	2633.4
	média	2701.2	2653.0	2637.2	2634.9	2599.8
100	desvio padrão	40.2	34.4	32.8	32.8	24.7
	mín.	2604.4	2584.4	2564.4	2548.4	2527.5
	máx.	2768.4	2712.4	2692.4	2678.1	2636.4

Tabela 5.6 – Resumo dos resultados computacionais (instância A)

Instância B

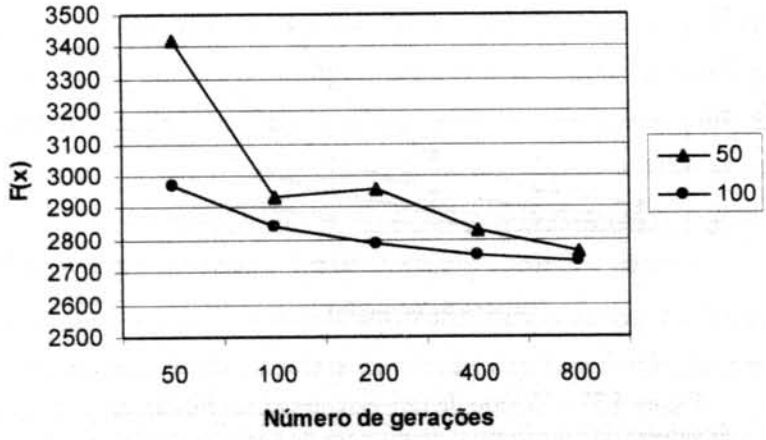


Figura 5.38 – Avaliação em função da dimensão da população e do número de gerações (instância B)

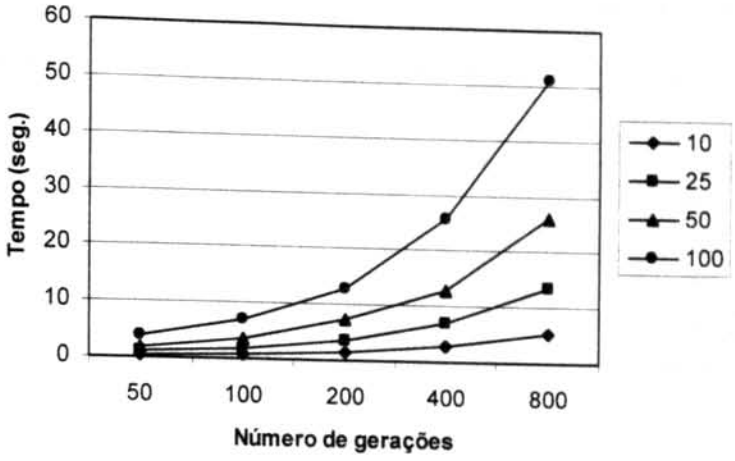


Figura 5.39 – Tempo de processamento função da dimensão da população e do número de gerações (instância B)

Dim. Popul.	F(x)	Número de gerações				
		50	100	200	400	800
10	média	---	---	---	---	---
	desvio padrão	---	---	---	---	---
	mín.	2753.1	2741.1	2829.1	2795.3	2717.8
	máx.	---	---	---	---	---
25	média	---	---	---	3210.1	3298.5
	desvio padrão	---	---	386.5	482.5	402.8
	mín.	2827.5	2753.1	2703.5	2696.4	2703.5
	máx.	---	---	3826.9	4123.5	3703.5
50	média	3420.3	2930.1	2956.0	2762.5	2829.0
	desvio padrão	1147.1	190.7	287.7	52.7	249.4
	mín.	2805.8	2719.3	2717.8	2661.2	2661.2
	máx.	7604.5	3445.3	3655.5	2855.5	3551.5
100	média	2971.4	2838.6	2785.8	2750.8	2729.7
	desvio padrão	142.2	89.1	46.8	41.3	23.6
	mín.	2751.5	2699.6	2677.2	2661.2	2696.4
	máx.	3239.5	3041.8	2870.1	2805.7	2774.9

Tabela 5.7 – Resumo dos resultados computacionais (instância B)

Instância C

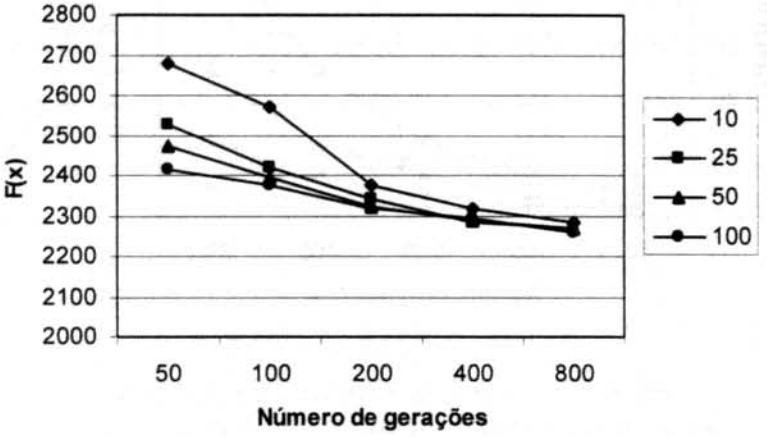


Figura 5.40 – Avaliação função da dimensão da população e do número de gerações (instância C)

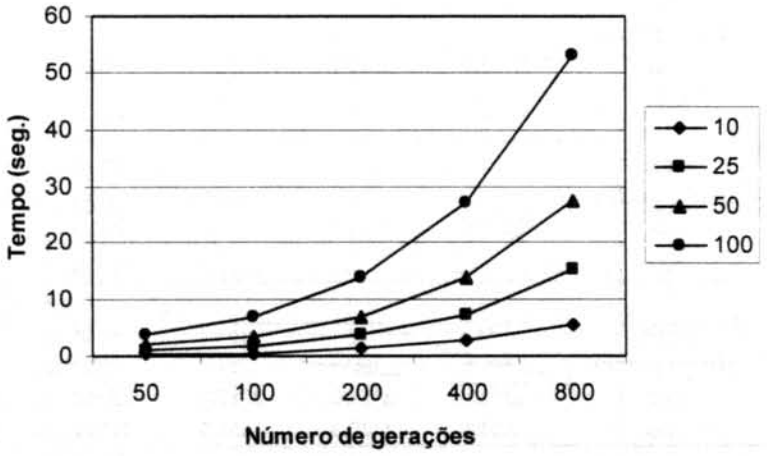


Figura 5.41 – Tempo de processamento função da dimensão da população e do número de gerações (instância C)

Dim. Popul.	F(x)	Número de gerações				
		50	100	200	400	800
10	média	2678.7	2569.7	2374.5	2318.0	2284.6
	desvio padrão	238.5	172.1	67.0	51.1	41.1
	mín.	2310.1	2326.1	2262.1	2210.1	2210.1
	máx.	3083.4	2910.1	2523.7	2428.1	2358.1
25	média	2528.4	2395.6	2343.2	2285.1	2269.5
	desvio padrão	133.0	80.6	73.6	59.8	34.5
	mín.	2282.1	2238.0	2218.1	2210.1	2210.1
	máx.	2799.6	2541.8	2486.8	2438.1	2330.1
50	média	2472.0	2421.2	2322.8	2288.2	2268.7
	desvio padrão	90.8	90.9	45.6	33.6	31.6
	mín.	2274.1	2237.4	2218.1	2210.1	2210.1
	máx.	2594.9	2560.4	2416.9	2338.1	2310.1
100	média	2417.1	2375.0	2316.5	2292.1	2259.2
	desvio padrão	74.7	69.0	46.1	37.3	27.9
	mín.	2254.1	2250.1	2238.0	2226.1	2210.1
	máx.	2540.5	2499.7	2391.6	2354.1	2298.1

Tabela 5.8 – Resumo dos resultados computacionais (instância C)

Analisando este conjunto de resultados, podem tirar-se as seguintes conclusões:

1. Os resultados são, naturalmente, melhores à medida que aumenta, quer a dimensão da população, quer o número de gerações, com a contrapartida óbvia de um aumento do tempo de processamento. Neste "compromisso", verifica-se uma melhoria significativa quando de passa de uma dimensão da população de 10 para uma dimensão de 25.
2. À medida que o número de gerações aumenta, verifica-se que os valores das soluções finais são muito próximas qualquer que seja a dimensão da população.
3. Para dimensões da população de 10 e de 25, nem sempre se conseguiram obter soluções finais para a instância B (caracterizada por uma data pretendida, próxima da data actual).
4. Não se verificam melhorias significativas para populações com dimensão superior a 50, e com pelo menos 200 gerações.

As conclusões obtidas, indo de encontro às expectativas, permitiram igualmente definir, no caso geral e para além dos outros parâmetros considerados, que uma dimensão da população de 50 e um número de gerações de 400 são adequados.

5.6.8.3 Análise da solução final

Apresenta-se de seguida a melhor solução final obtida para a instância A, utilizando para parâmetros do algoritmo os valores definidos anteriormente (limita-se a apresentação dos resultados à instância A, pelas razões anteriormente apresentadas).

A solução final, após 400 gerações, apresentou um custo global de 2548.38, com as seguintes datas de início e de fim de produção (indicam-se também as unidades de produção envolvidas em cada estágio):

Estágio	1		2		3		4	
Datas produção	22	32	36	50	51	69	71	80
Unidade prod.	14		21		32		42	

Para esta solução a segmentação dos custos foi a seguinte:

Custo total	**Custo produção**	**Custo armazen.**	**Custo transporte**	**Custo penalização**
2548.38	2268.38	0	280	0

Da análise da solução final obtida, podem tirar-se as seguintes conclusões:

- foi respeitada a data pretendida, não se verificando por isso qualquer custo adicional de penalização;
- não se verifica qualquer armazenamento entre estágios, isto é, a folga entre a data de conclusão da produção numa unidade produtiva e a data de início de produção da unidade seguinte é exactamente igual ao tempo de transporte entre as unidades.

O gráfico da Figura 5.42, correspondente à solução apresentada, representa a evolução da melhor solução de cada população ao longo das várias gerações.

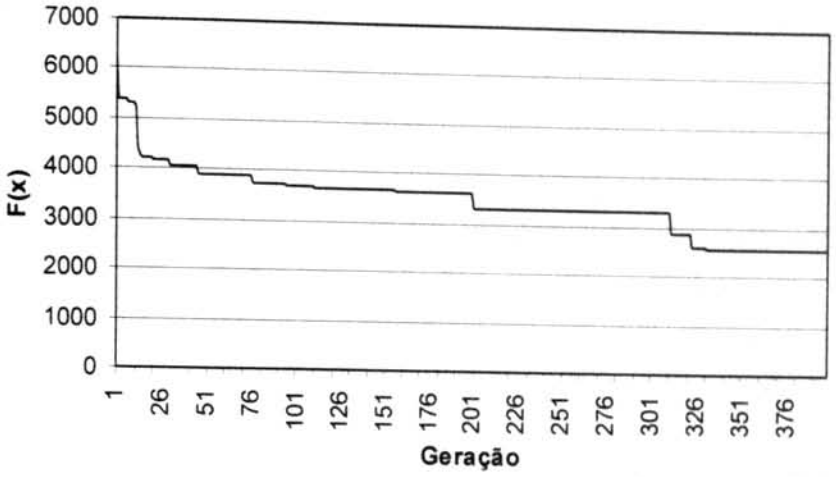


Figura 5.42 – Evolução do algoritmo

Uma outra solução, envolvendo para os três primeiros estágios outras unidades produtivas, e com custo global ligeiramente superior, foi a seguinte:

Estágio	1		2		3		4	
Datas produção	37	44	46	56	58	67	69	80
Unidade prod.	12		22		31		42	

Para esta solução a segmentação dos custos foi a seguinte:

Custo total	**Custo produção**	**Custo armazen.**	**Custo transporte**	**Custo penalização**
2586.13	2426.13	0	160	0

Relativamente à solução anterior, esta solução alternativa consegue "concentrar" mais o planeamento em torno da data de fim pretendida, envolvendo um aumento do custo de produção de cerca de 10%. Apesar do agravamento do custo de produção, não se verifica, contudo, um agravamento significativo do custo global, em virtude de, nesta alternativa, os custos de transporte serem significativamente inferiores.

5.7 Heurística multi-atributo

5.7.1 Considerações gerais

No contexto do problema que se tem vindo a estudar, existem numerosas alternativas no planeamento de um pedido, nomeadamente, quanto a diferentes datas de processamento, quantidades a produzir ao longo do tempo e/ou distribuição da produção no mesmo estágio, por diferentes unidades produtivas.

A excessiva dimensão de um modelo de programação linear inteira, por um lado, e o elevado número de iterações necessárias com métodos de pesquisa, como o *simulated annealing* ou os algoritmos genéticos, por outro, motivaram o desenvolvimento de uma estratégia de planeamento, baseada numa heurística construtiva específica, aqui designada por "heurística multi-atributo".

As abordagens com meta-heurísticas, consideradas nas secções anteriores, procuram encontrar uma solução para o problema, pressupondo sempre que todos os estágios são envolvidos. Não são tratadas situações de planeamento parcial, isto é, envolvendo apenas alguns dos estágios.

A Figura 5.2 apresenta, de forma simplista, diferentes cenários possíveis para uma solução do problema de planeamento de uma encomenda. Com a finalidade de contemplar este tipo de situações, e procurando ultrapassar as limitações referidas, desenvolveu-se uma heurística construtiva que, explorando a estrutura do problema, procura inserir a encomenda no actual plano de produção, a partir da data de conclusão pretendida.

Foram identificados três cenários principais: planeamento total, planeamento parcial de produtos intermédios e planeamento parcial de produtos finais.

5.7.1.1 Cenário 1: planeamento total

Nesta situação, a quantidade de produto pedida (pelo cliente) é a correspondente ao resultado do processamento do último estágio e envolve todos os estágios produtivos, eventualmente com diferentes quantidades (efeito de absorção de *stocks* intermédios). A Figura 5.43 ilustra este cenário, para a situação de quatro estágios de produção, com um ponto de desacoplamento intermédio (p_{iO} e q_{iO} representam, respectivamente, o produto e a quantidade na saída do estágio i; P_O e Q_O, representam, respectivamente, o produto final e a quantidade pedida).

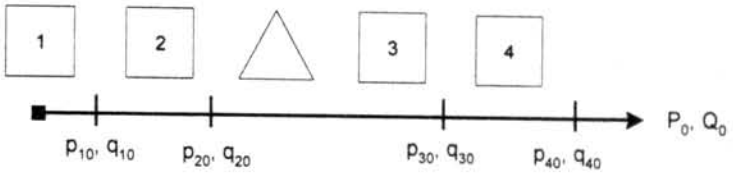


Figura 5.43 – Planeamento envolvendo todos os estágios

5.7.1.2 Cenário 2: planeamento parcial de produtos intermédios

Nesta situação, o estágio final alvo não corresponde ao último estágio do processo produtivo (ver Figura 5.44). Este cenário está associado ao planeamento de um produto intermédio (situação similar ao planeamento para *stock* intermédio como o *die-bank*, no caso da indústria de semicondutores – que corresponde a produção baseada em previsões).

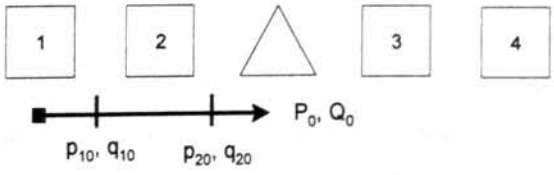


Figura 5.44 – Planeamento de produtos intermédios

5.7.1.3 Cenário 3: planeamento parcial de produtos finais

Este cenário é representativo da situação de satisfação de uma encomenda a partir de um ponto estratégico do processo produtivo (situação da satisfação de uma encomenda a partir do *die-bank*, no caso dos semicondutores).

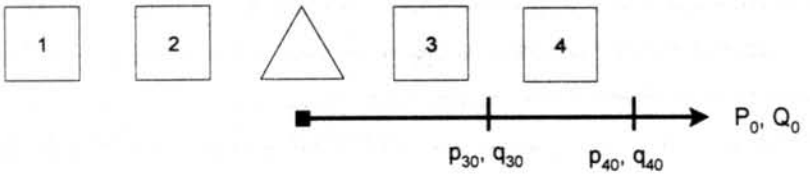


Figura 5.45 – Planeamento parcial

5.7.1.4 Cenários híbridos

Situações particulares dos cenários anteriores resultam quando o planeamento envolve unicamente alguns dos estágios intermédios (considerar, por exemplo que no cenário 2, o estágio inicial de planeamento não é o primeiro do processo produtivo).

No caso geral, é possível ter uma situação de planeamento de encomendas em que a rota associada envolve todos os estágios e em e que se verifica, para o mesmo estágio, a participação de diferentes unidades.

A Figura 5.46 ilustra um cenário em que existem quatro estágios constituídos por diversas unidades produtivas e respectivos pontos de armazenamento. Encontra-se representada a rota associada ao planeamento de uma encomenda. Este cenário evidencia a complexidade inerente aos processos de planeamento e de coordenação, que têm lugar num ambiente de produção multi-estágio e simultaneamente distribuído.

Neste exemplo, verifica-se a coexistência de situações similares às apresentadas nos cenário 1 e 3.

Naturalmente que o tipo de produto impõe o âmbito do planeamento. No limite, o pedido de um produto final pode envolver todos os estágios produtivos. O estágio mais a montante, e envolvido na satisfação do pedido, é imposto pelas necessidades líquidas na entrada do estágio imediatamente a seguir.

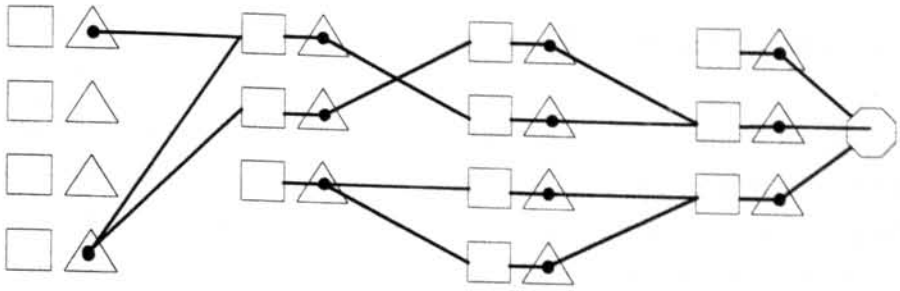


Figura 5.46 – Rota associada ao planeamento de encomendas no caso geral

Uma outra questão, que a heurística tem em conta, é o eventual envolvimento, no mesmo estágio, de mais do que uma unidade de produção (*order splitting*), com vista a satisfazer as necessidades globais do estágio. Em termos reais, esta possibilidade de "paralelismo" ao nível de alguns estágios de produção, assume particular importância nas situações envolvendo unidades subcontratadas. É claro que esta opção conduz, de forma natural, ao aumento da capacidade disponível nesse estágio. A Figura 5.47 apresenta uma situação em que a satisfação das necessidades de entrada e de saída de um estágio é conseguida com o envolvimento de duas unidades independentes.

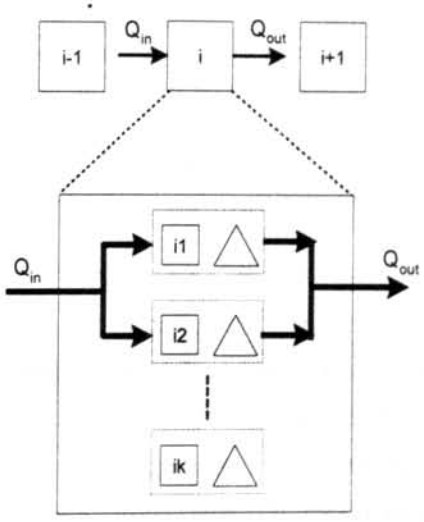


Figura 5.47 – *Order splitting* num estágio

5.7.2 Uma abordagem multi-atributo

A abordagem heurística que de seguida se apresenta, é baseada na análise "multi--atributo" das ofertas enviadas ao *rough planner* pelos modelos locais de capacidade. O conjunto de atributos considerados, procura reflectir a estratégia do decisor, no processo de negociação subjacente ao planeamento/coordenação de um pedido (potencial encomenda). Nomeadamente, são considerados factores tais como a violação de datas de conclusão, a não satisfação de quantidades pedidas, a minimização do WIP e do *lead-time* total, a maximização da "fiabilidade" das entidades participantes na satisfação do pedido, etc.

A heurística segue uma estratégia de planeamento "para trás" (*backward planning*), partindo sempre do estágio alvo ao qual está associado o produto pedido. Iterativamente, planeia estágio a estágio, até atingir o primeiro estágio do processo ou até um estágio intermédio em que o envolvimento das unidades produtivas pertencentes a esse estágio não implique quaisquer necessidades líquidas de produtos provenientes do estágio predecessor. Se for impossível satisfazer a quantidade pedida em qualquer um dos estágios ("*bottleneck*"), a heurística procura escalonar o pedido, respeitando a data de conclusão pedida, mas considerando uma quantidade inferior (fase de renegociação).

Considerou-se a mesma estrutura hierárquica descrita anteriormente: a um nível superior são geradas propostas e avaliadas ofertas considerando sempre um elevado nível de agregação de informação; a um nível inferior são avaliadas, pelos modelos locais de capacidade, as propostas geradas no nível superior.

5.7.2.1 Atributos

Foram definidos dois conjuntos de atributos, com âmbitos diferentes. O primeiro conjunto, de âmbito global, permite a avaliação das soluções alternativas associadas ao planeamento da encomenda, ao longo de toda a cadeia produtiva. O segundo conjunto de atributos permite a avaliação ao nível do estágio, estando por isso associado às ofertas recebidas das unidades pertencentes a esse estágio. O decisor tem a possibilidade de efectuar o ajuste individual da importância relativa de cada um dos atributos, quer a nível global, quer a nível de cada um dos estágios, e permitindo assim uma maior ou menor contribuição de cada um deles para a avaliação da alternativa.

Atributos de âmbito global

São considerados os atributos seguintes:

1. *Produção just-in-time* – mede o desvio da data de conclusão planeada relativamente à data pedida. Permite ao decisor expressar a importância de a data de conclusão planeada para a encomenda ser o mais próxima possível da data pedida (*due-date*).
2. *Custos de produção* – representam a agregação dos custos de produção (fixos e variáveis) associados a cada entidade envolvida na satisfação da encomenda. Permite ao decisor fazer reflectir a importância dos custos globais de produção na avaliação de soluções alternativas.
3. *Custos logísticos* – representam a agregação dos custos de transporte e de armazenamento entre estágios. Permite a avaliação da importância dos custos logísticos.
4. *Lead-time global* – é uma medida do tempo total planeado para a permanência do produto no processo produtivo. Permite ao decisor privilegiar soluções planeadas o mais tarde possível.
5. *Fiabilidade* – mede a qualidade da alternativa em avaliação, no que respeita ao cumprimento da data de conclusão planeada; é igual à probabilidade da data de entrega (*delivery date*) ser a mesma da data planeada (*promised date*).

O atributo *fiabilidade* depende directamente de cada uma das unidades produtivas participantes na alternativa em avaliação. Diferentes alternativas podem apresentar diferentes valores de "fiabilidade", em virtude de eventualmente terem associadas diferentes rotas produtivas envolvendo diferentes unidades produtivas, que naturalmente introduzem maior ou menor variabilidade no tempo global do processo.

Para cada uma das alternativas, este atributo é calculado da seguinte forma. Considere-se um sistema produtivo com n estágios e em cada estágio i, n_i entidades. Seja λ_{ij} a fiabilidade da entidade j pertencente ao estágio i (isto é, a fiabilidade de uma entidade é entendida como que a probabilidade dessa entidade cumprir com as datas planeadas consideradas nas ofertas por si geradas), e γ_{ij} uma variável binária que representa a participação ou não da entidade na alternativa em avaliação. A fiabilidade para cada um dos estágios pode então ser calculada da seguinte forma:

$$\lambda_i = \prod_{j=1}^{n_i} \lambda_{ij} \cdot \gamma_{ij}$$

Para cada alternativa, a fiabilidade global (λ_{global}) é então calculada a partir de:

$$\lambda_{global} = \prod_{i=1}^{n} \lambda_i = \prod_{i=1}^{n} \prod_{j=1}^{n_j} \lambda_{ij} \cdot \gamma_{ij}$$

Atributos de âmbito do estágio

Em cada estágio, e para cada uma das ofertas recebidas, são considerados os atributos seguintes:

1. *Quantidade de saída* – mede a qualidade da oferta no que respeita à proximidade entre a quantidade disponível na saída da data de conclusão e a quantidade pedida.
2. *Quantidade de entrada* – mede a dependência da quantidade disponível na saída relativamente ao estágio precedente.
3. *Data de conclusão* – exprime o desvio entre a data de conclusão pedida e a data de conclusão da oferta, isto é, a data de conclusão proposta pela unidade local.
4. *Data início* – exprime o desvio entre a data de início sugerida e a data início da oferta em avaliação.
5. *Lead-time* – mede, para cada oferta, o tempo total de produção.
6. *Custo de produção* – reflecte os custos totais, para produzir a quantidade proposta, no tempo proposto.
7. *Fiabilidade* – exprime a probabilidade da entidade cumprir a oferta proposta.
8. *Lead-Time proposto* – mede o desvio entre o *lead-time* sugerido e o *lead-time* proposto na oferta.

Assim considerado, o problema de planeamento de uma encomenda, quer a nível global quer a nível do estágio, pode ser encarado como um problema de Decisão Multi-Atributo (MADM, da terminologia anglo-saxónica, *Multi-Attribute Decision Making*), em que se procura tomar decisões perante múltiplas alternativas eventualmente conflituosas. Para uma descrição das principais metodologias de ajuda à decisão multi-critério (multi-atributo e multi-objectivo) ver, por exemplo, o trabalho de Hwang e Yoon (Hwang and Yoon, 1981).

O objectivo para cada estágio é determinar, considerando todos os critérios, qual é a oferta ou conjunto de ofertas que satisfazem os requisitos do pedido associado a esse estágio e apresentam o melhor desempenho. Descreve-se de seguida, de forma geral, o problema MADM.

Uma alternativa A_i é um conjunto de atributos X_{ij} $(j = 1,...,n)$, que medem o desempenho da alternativa relativamente a um conjunto de critérios definidos $A_i = \{X_{i1}, X_{i2}, \cdots, X_{in-1}, X_{in}\}$. O conjunto de todas as alternativas define a matriz de decisão A, onde cada alternativa é uma linha da matriz e cada coluna, vector $Xj = \{X_{1j}, X_{2j}, \cdots, X_{m-1j}, X_{mj}\}^T$, representa o valor de cada alternativa relativamente a um atributo j. Cada elemento X_{ij} da matriz de decisão A representa para a alternativa A_i o valor do atributo j.

Uma vez definida a matriz de decisão A, é necessário proceder à sua normalização, mediante uma transformação linear, obtendo-se assim a matriz de "comparação" A'. Esta transformação tem a finalidade de permitir uma comparação directa inter-atributos.

A matriz de comparação A' pode ser obtida a partir da matriz A, aplicando, por exemplo, as transformações lineares seguintes:

$$X'_{ij} = \frac{X_{ij}}{\max_i X_{ij}}$$

Para critérios de benefício, isto é, para parâmetros tais que quanto maior o valor maior a preferência;

$$X'_{ij} = \frac{\min_i X_{ij}}{X_{ij}}$$

Para critérios de custo, isto é, quanto menor o valor maior a preferência.

Sendo $W = \{w_1, w_2, \cdots, w_{n-1}, w_n\}^T$ o conjunto de pesos associados a cada atributo, definidos pelo decisor, então a melhor alternativa A^*, do conjunto de soluções alternativas em avaliação, é seleccionada de tal forma que $A^* = \{A_i \mid \max_i (A' * W^T)\}$.

5.7.3 Pressupostos gerais

A heurística aqui apresentada considera os seguintes pressupostos (menos restritivos do que nos casos tratados nas secções anteriores):

- A quantidade a produzir em cada estágio não é necessariamente igual (dependendo do *stock* disponível no período considerado).
- Cada unidade produtiva mantém localmente informação de ATP (ver capítulo 2). Este conceito é materializado pela disponibilidade de produto (*stock*) no fim da data pedida (*finish date*).
- Existem *stocks* intermédios (entre estágios). Estes stocks são a consolidação dos *stocks* existentes em cada uma das unidades pertencentes a um estágio.
- Por estágio, pode participar mais do que uma unidade produtiva, na satisfação do pedido.
- Nenhuma operação pode ser iniciada numa unidade, antes de todas as que a precedem estarem concluídas.
- O transporte inicia-se imediatamente após o fim de processamento. Considera-se que o tempo de transporte entre unidades é constante, no tempo.

5.7.4 Formulação matemática

Apresenta-se de seguida a formulação matemática para o conjunto de restrições consideradas pela heurística. A Figura 5.48 ilustra a topologia considerada no desenvolvimento da heurística, salientando-se a existência de múltiplas unidades produtivas no mesmo estágio.

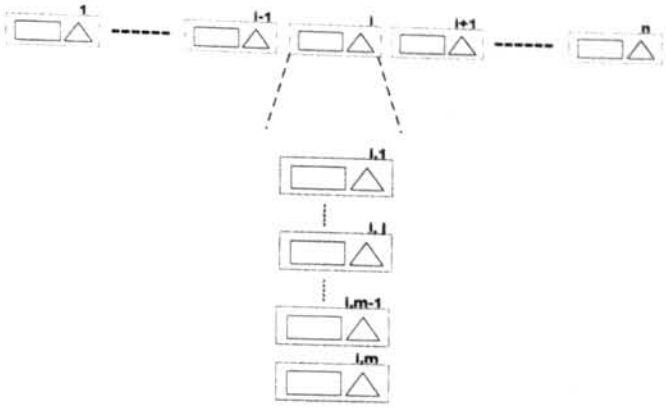


Figura 5.48– Topologia genérica da estrutura produtiva

Parâmetros

q	quantidade pedida
dd	data de entrega pedida (*due date*)
ed	menor data de entrega possível (*earliest finish date*)
ld	maior data de entrega possível (*latest finish date*)
t_{ij}^{move}	tempo de transporte entre as unidades *i* e *j* de estágios consecutivos
J_i	conjunto das unidades produtivas pertencentes ao estágio i
S_i	número de unidades que podem participar no estágio i
I	número de estágios

Variáveis de decisão

x_{ij}	quantidade de produto na saída na unidade *j* do estágio *i*
y_{ij}	quantidade de produto na entrada na unidade *j* do estágio *i*
t_{ij}^{start}	ínicio da produção na unidade *j* dos estágio *i*
t_{ij}^{finish}	fim da produção na unidade *j* dos estágio *i*

variáveis auxiliares

X_i	quantidade total na saída do estágio *i*
Y_i	quantidade total na entrada do estágio *i*

T_i^{start} início da produção no estágio i

T_i^{finish} fim da produção no estágio i

$$\delta_{ij} = \begin{cases} 1 & \text{se existe produção na unidade } j, \text{ estágio } i \\ 0 & \text{se não} \end{cases}$$

5.7.5 Restrições

$$i = I, \forall j \in J_i \quad \sum_J x_{ij} \geq q \tag{0}$$

$$\forall j \in J_i \quad X_i = \sum_J x_{ij} \cdot \delta_{ij} \tag{1}$$

$$\forall j \in J_i \quad Y_i = \sum_J y_{ij} \cdot \delta_{ij} \tag{2}$$

$$i = 1..I-1 \quad X_i \geq Y_{i+1} \tag{3}$$

$$\forall j \in J_i \quad \sum_J \delta_{ij} \leq S_i \tag{4}$$

$$\forall j \in J_i \quad T_i^{finish} \geq t_{ij}^{finish} \tag{5}$$

$$\forall j \in J_i \quad T_i^{start} \leq t_{ij}^{start} \tag{6}$$

$$i = I, \forall j \in J_i \quad t_{ij}^{finish} \leq ld \tag{7}$$

$$i = I, \forall j \in J_i \quad t_{ij}^{finish} \leq ed \tag{8}$$

$$i = 1..I-1, \forall j \in J_i \quad T_i^{finish} + t_{ij}^{move} \leq T_{i+1}^{start} \tag{9}$$

$$\forall i, \forall j \in J_i \quad x_{ij} \leq M \cdot \delta_{ij} \tag{10}$$

$$\forall i, \forall j \in J_i \quad x_{ij}, y_{ij} \geq 0 \tag{11}$$

$$\forall i, \forall j \in J_i \quad \delta_{ij} \in \{0,1\} \tag{12}$$

A restrição (0) garante que na saída de cada estágio se obtém a quantidade pedida.

As restrições (1) e (2) definem as variáveis auxiliares, respectivamente para a quantidade total na saída e na entrada de cada estágio.

A restrição (3) garante que, na saída de cada estágio, se dispõe da quantidade necessária na entrada do estágio seguinte.

A restrição (4) restringe o número de unidades produtivas que podem participar nesse estágio (grau de '*splitting*').

As restrições (5) e (6) definem os tempos de início e de fim de um processamento num estágio, como sendo iguais, respectivamente, ao menor e ao maior dos tempos de início e de fim das unidades envolvidas e pertencentes a esse estágio.

As restrições (7) e (8) asseguram que a data de conclusão se encontra dentro do intervalo definido entre os limites *earliest finish date* e *lateste finish date.*

A restrição (9) assegura a satisfação das relações de precedência entre estágios.

5.7.6 Dados de entrada e resultados

Dados de entrada

- Pedido a planear: produto, data de entrega, quantidade.
- Pesos a atribuir aos atributos de âmbito global e local (estágio).

Resultados

Conjunto de soluções alternativas definidas por:

- intervalos (para cada estágio) em que vai ter lugar a produção (data início, data fim);
- quantidade total a produzir por estágio;
- quantidade a produzir em cada unidade envolvida na produção;
- intervalo (data início, data fim) em que ocorre a produção em cada unidade envolvida;
- custo global.

5.7.7 Descrição geral da heurística

Esta heurística é constituída pelas seguintes fases:

Fase 1

Com base no pedido recebido, determina-se, a partir da identificação do produto, qual o estágio final de produção correspondente a esse pedido.

Fase 2

Para cada um dos estágios, e numa lógica de planeamento 'para-trás', procede-se à formação da proposta base, a enviar às unidades produtivas respectivas. Começa-se por calcular a quantidade de saída, com base na quantidade de entrada do estágio seguinte (para o estágio final, é a quantidade pedida pelo cliente). A data de fim de produção é calculada com base na data de início do estágio seguinte e no tempo de transporte entre estágios (para o estágio final, é a data de entrega pedida). A data de início é gerada aleatoriamente, a partir da data fim e das constantes de parametrização dos *lead-times* associados ao estágio. A proposta base associada ao estágio é enviada a todas as unidades pertencentes a esse estágio.

Fase 3

Após a recepção de ofertas (vindas das unidades "locais"), procede-se à sua ordenação, através de uma avaliação multi-atributo, constituindo-se uma "pilha de ofertas". Até perfazer a quantidade que se pretende produzir no estágio, são seleccionadas ofertas da "pilha". No caso de existirem diferenças (data e quantidade) entre a proposta efectuada e as ofertas seleccionadas, procede-se à reformulação da proposta, isto é, procede-se à actualização da quantidade a pedir a cada unidade envolvida, assim como é actualizado o intervalo de produção no estágio (mecanismo de "contra-oferta"). A proposta reformulada é enviada novamente às unidades pertencentes ao estágio, repetindo-se esta fase, enquanto se verificarem alterações entre quantidades 'pedidas' e quantidades 'oferecidas'.

Fase 4

Determina-se qual o estágio seguinte a planear, com base nas necessidades de entrada do estágio acabado de planear. Se a quantidade de entrada for nula, passa-se à fase seguinte, senão retorna-se à fase 2.

Fase 5

Procede-se à construção global da solução, com base nas soluções parcelares obtidas nas fases precedentes. Senão for possível satisfazer a quantidade pedida ou a data fim, são relaxados os valores respectivos, procurando-se satisfazer uma menor quantidade, ou mesmo planeando para depois da data de entrega pedida. Se o critério de paragem não foi atingido (por exemplo, número de soluções alternativas a definir) retorna-se à fase inicial.

Fase 6

Procede-se à ordenação das soluções globais obtidas, recorrendo ao avaliador multi-atributo global.

5.7.8 Exemplo numérico

O exemplo seguinte ilustra numericamente o funcionamento da heurística, para uma única "corrida", isto é, para a geração de apenas uma solução.

Considera-se uma estrutura produtiva com a mesma configuração e informação logística da apresentada em 5.4.4. Pretende-se verificar a possibilidade de se aceitar um pedido de 100 unidades de um produto final, para o período 105.

A partir da referência do produto, é determinado o estágio alvo (isto é, o último estágio, envolvido no fabrico do produto). No caso presente, começa-se então pelo último estágio da estrutura produtiva (estágio 4). De seguida, é gerado, a partir da heurística de geração de pedidos (de acordo com o descrito na *Fase 2* da heurística), um pedido ao estágio (data início, data fim e quantidade) e recebe-se um conjunto de ofertas produzidas por cada modelo local de capacidade.

Pedido: (64, 105, 100)

Conjunto de ofertas recebidas:

Id	Data início	Data fim	Quant. Saída	Quant. Entrada	Custo
U_{41}	75	105	100	100	10
U_{42}	85	105	100	100	10

Com o pedido e o conjunto de ofertas, é construída a matriz de decisão a partir do cálculo dos oito atributos anteriormente definidos. Assim, resulta para o estágio quatro e seguindo a mesma ordem com que foram definidos os atributos aquando da sua descrição, a seguinte matriz de decisão e de comparação. Note-se que, dos oito atributos considerados, alguns são de benefício (critérios 1, 4 e 7) e os restantes (2, 3, 5, 6 e 8) são de custo, procurando-se na avaliação das alternativas, respectivamente, a sua maximização ou minimização.

Matriz de decisão:

$$A = \begin{bmatrix} 1 & 1 & 0 & 1.172 & 30 & 0.1 & 0.9 & 0.732 \\ 1 & 1 & 0 & 1.328 & 20 & 0.1 & 0.6 & 0.488 \end{bmatrix}$$

Matriz de comparação (após normalização dos atributos)

$$A' = \begin{bmatrix} 1 & 1 & 1 & 0.883 & 0.667 & 1 & 1 & 0.667 \\ 1 & 1 & 1 & 1 & 1 & 1 & 0.667 & 1 \end{bmatrix}$$

Considerando para o estágio, o conjunto de pesos associados aos diferentes atributos, W = {0.25, 0.2, .25, 0.05, 0.09, 0.1, 0.05, 0.01}, a melhor alternativa, isto é, a alternativa que apresenta maior "pontuação" após o cálculo de $A' \cdot W^T$ é a oferta proveniente da unidade U_{42}. A solução obtida para este estágio envolve unicamente uma unidade, porque se consegue satisfazer a quantidade pedida a partir da melhor alternativa.

O processo de planeamento continua agora para o estágio anterior (estágio 3) e com a mesma quantidade a planear:

Pedido: (58, 83, 100)

Conjunto de ofertas recebidas:

Id	Data início	Data fim	Quant. Saída	Quant. Entrada	Custo
U_{31}	69	83	60	0	9
U_{32}	67	83	80	0	7
U_{33}	62	75	50	20	5

Matriz de decisão:

$$A = \begin{bmatrix} 0.6 & 0 & 0 & 1.19 & 14 & 0.15 & 0.9 & 0.56 \\ 0.8 & 0 & 0 & 1.155 & 16 & 0.09 & 0.6 & 0.64 \\ 0.5 & 0.4 & 0.32 & 1.069 & 13 & 0.1 & 0.9 & 0.52 \end{bmatrix}$$

Matriz de comparação:

$$A' = \begin{bmatrix} 0.75 & 1 & 1 & 1 & 0.929 & 0.6 & 1 & 0.929 \\ 1 & 1 & 1 & 0.971 & 0.813 & 1 & 0.667 & 0.813 \\ 0.625 & 0 & 0 & 0.898 & 1 & 0.9 & 1 & 1 \end{bmatrix}$$

Se o vector de pesos para este estágio for W = {0.25, 0.15, 0.25, 0.05, 0.09, 0.15, 0.05, 0.01}, resulta a seguinte pontuação para cada uma das ofertas: 0.6204, 0.7132 e 0.4862 respectivamente. Destas alternativas, escolhem-se as duas melhores ofertas, provenientes de U_{32} e U_{31}, por forma a garantir a disponibilização na saída da quantidade necessária. Neste caso, há ainda que proceder a uma renegociação das quantidades envolvidas, com a finalidade de garantir exactamente a quantidade pedida pelo cliente.

A solução final envolve três unidades de produção, distribuídas por dois estágios (situação similar ao cenário apresentado em 5.7.1.3). Isto é, a solução final envolve as unidades U_{31}, U_{32} e U_{42}.

5.8 Conclusões

Dos resultados computacionais apresentados, embora preliminares, pode-se concluir que as abordagens heurísticas desenvolvidas e implementadas neste trabalho, são bastante promissoras, dado que permitem obter em tempo útil, soluções de boa qualidade. Por outro lado, dada a excessiva dimensão de um eventual modelo de programação matemática a resolução do problema através de ferramentas clássicas de optimização afigura-se praticamente impossível (do ponto de vista de processamento). Tal verifica-se não só pela complexidade do problema, como fundamentalmente pela constante mudança dos dados e da estrutura do próprio problema, que resultam das mudanças que ocorrem na existentes na estrutura produtiva.

As soluções obtidas com as duas meta-heurísticas implementadas, *simulated annealing* e algoritmos genéticos, apresentam uma qualidade semelhante; no entanto os tempos envolvidos são significativamente maiores no *simulated annealing*. Além disso, um outro aspecto a considerar tem a ver com o facto da abordagem com algoritmos genéticos permitir reduzir substancialmente o *overhead* associado ao processo de negociação. Isto é, em cada transacção entre o componente de planeamento e os modelos de capacidade são enviadas simultaneamente todas as propostas respeitantes a uma população (no limite são geradas tantas propostas quanto o número de elementos da população).

Por fim, refira-se que a heurística multi-atributo apresentada parece conduzir a resultados práticos interessantes, abrindo novas perspectivas na abordagem do problema que, naturalmente, interessará explorar no futuro.

Capítulo 6

Uma Abordagem Baseada em Agentes Inteligentes

6.1 Agentes inteligentes

6.2 Uma ontologia para a Empresa Virtual

6.3 Uma arquitectura multi-agente

6 UMA ABORDAGEM BASEADA EM AGENTES INTELIGENTES

O Sistema de Apoio à Decisão que constitui a parte central do trabalho apresentado nesta dissertação (ver capítulo 4), apesar de ser baseado em componentes distribuídos e permitir, por isso, tirar partido de todas as vantagens dessa tecnologia, obriga a um esforço considerável para configuração de cada uma das entidades locais que constituem a *Empresa Virtual*. A integração ou, num sentido mais estrito, a cooperação entre entidades, pressupõe sempre a existência de um modelo de dados comum e hierarquizado segundo dois níveis complementares: um global e um local[35].

Uma abordagem de mais alto nível, conceptualizada segundo "agentes inteligentes", consiste em considerar as diferentes entidades como sendo autónomas, inteligentes e desacopladas, relativamente à informação de outras entidades. A ideia central desta abordagem é conseguir que essas entidades tenham capacidade de avaliar a sua própria "habilitação" para desempenhar determinadas actividades, através de um sistema de informação adaptativo e auto-reconfigurável. Subjacente a esta abordagem com agentes inteligentes, e no sentido de se tirar partido dos métodos e conceitos desenvolvidos no âmbito dos paradigmas em que se inserem, está a *ontologia* do domínio do problema, no caso presente designada por *Virtual Enterprise Ontology*.

É nesta perspectiva que o presente capítulo propõe uma abordagem ao problema de planeamento de encomendas em redes de produção distribuídas, segundo o paradigma dos Sistemas Multi-Agente. Assim, após uma breve caracterização dos Sistemas Multi-Agente e, em particular, da linguagem de comunicação entre agentes, são apresentadas as linhas principais da aplicação destes paradigmas ao problema em análise. São ainda descritos sucintamente, alguns cenários de comunicação entre entidades pertencentes à *Empresa Virtual*, durante as fases de configuração e de negociação (que não sendo o objecto central deste trabalho, assumem, na prática uma enorme importância).

---

[35] Ao nível da gestão de informação, o modelo é comum, sendo contudo instanciado tantas vezes quantos os modelos de capacidades considerados.

6.1 Agentes inteligentes

6.1.1 Conceitos fundamentais e enquadramento

No contexto que temos vindo a apresentar, em que se pretende fazer uma gestão eficaz de um ambiente produtivo intrinsecamente cooperativo e distribuído, enquadram-se, de forma natural, as propriedades consideradas fundamentais na caracterização dos "agentes inteligentes", tais como estas são usualmente apresentadas na literatura, isto é: autonomia, pro-actividade, sociabilidade e reactividade (adiante estas propriedades serão descritas sucintamente). Neste sentido, propõe-se, neste capítulo, uma abordagem ao problema em estudo, baseada em agentes inteligentes, e tendo por base os conceitos e metodologias desenvolvidos no âmbito da Inteligência Artificial Distribuída (IAD) .

Inteligência Artificial Distribuída (IAD)

O propósito da IAD é desenvolver e aplicar conceitos e métodos que permitam a construção de sistemas compostos de múltiplas entidades, com graus de autonomia variáveis, espacialmente distribuídas e que interactuam entre si de forma eficiente e "inteligente", com vista à resolução de um problema global. Subjacente à IAD está, em certo sentido, a ideia de *comportamento inteligente colectivo e cooperativo*, constituindo não só uma extensão da Inteligência Artificial (IA) clássica, como também um enriquecimento da IA, com a emergência de novas propriedades e comportamentos (Labidi and Lejouad, 1993).

Os conceitos e métodos desenvolvidos no âmbito da IAD aplicam-se, por natureza, a ambientes com as seguintes características:

- distribuição espacial e funcional;
- conjunção de actividades complementares;
- existência de um ou mais objectivos globais;
- comportamento dinâmico, exigindo uma permanente adaptabilidade;
- heterogeneidade;

- existência de recursos (alternativos), em qualidade superior ao estritamente necessário.

Confrontando as características enunciadas com as do problema tratado neste trabalho, parece adequada a aplicação da IAD.

Tradicionalmente, a resolução de problemas pela IAD subdivide-se em duas grandes áreas (Mota, 1996): Métodos de Resolução Distribuída de Problemas (DPS – *Distributed Problem Solving*) e Sistemas Multi-Agente (MAS – *Multi-Agent Systems*).

Nos métodos DPS, a coordenação da actividade dos *agentes* (entidades do sistema) é centralizada. O planeamento das acções a desenvolver é efectuado inicialmente, por decomposição do problema em subproblemas (existem regras já definidas, relativamente ao comportamento dos agentes múltiplos). Em certo sentido, assume-se um ambiente estático em termos de contexto do problema.

Nos sistemas Multi-Agente, a coordenação da actividade dos agentes é descentralizada, participando estes, de uma forma mais autónoma, na tomada de decisões respeitantes às suas próprias acções. Não existe, para a resolução do problema, um planeamento inicial. O planeamento é distribuído por todos os agentes, e realizado no decorrer da execução, através de processos de negociação e coordenação. Igualmente importante é o facto de, numa abordagem MAS, os agentes terem "utilidade" por si próprios, isto é, continuam "utilizáveis" mesmo quando retirados da comunidade em que geralmente se integram.

Sistemas Multi-Agente

É consensual hoje em dia que os sistemas multi-agente, pelas suas capacidades de interacção, adaptabilidade e autonomia, são adequados para a cooperação entre entidades (ou sistemas "elementares") heterogéneas, em que a resolução de um problema implica uma interacção entre as unidades que constituem uma dada comunidade. Por outro lado, são adequados para ambientes dinâmicos em que ocorram frequentes alterações estruturais.

Basicamente, um sistema MAS compreende uma colecção de *agentes* interactuantes. A designação de *agente* é utilizada muita das vezes de uma forma imprecisa ou mesmo incorrecta, originando consequentemente inúmeras definições, "orientadas" pelas aplicações específicas em causa (Oliveira, 1998). Uma revisão da literatura na

área revela uma variedade significativa de definições e perspectivas em torno do conceito de *agente*. Por exemplo, um agente pode ser definido como "uma entidade que pode perceber o seu ambiente através de sensores e agir sobre este ambiente através de actuadores" (Russel and Norvig, 1995), colocando-se assim a ênfase nas capacidades de percepção e de acção em relação ao ambiente.

Segundo Huhns e Singh, agentes são componentes (*software*) activos, persistentes, comunicativos e de acção racional (isto é, o agente, com base no conhecimento que possui e com base nas suas capacidades de percepção, executa a acção que dele se espera) (Huhns and Singh, 1998).

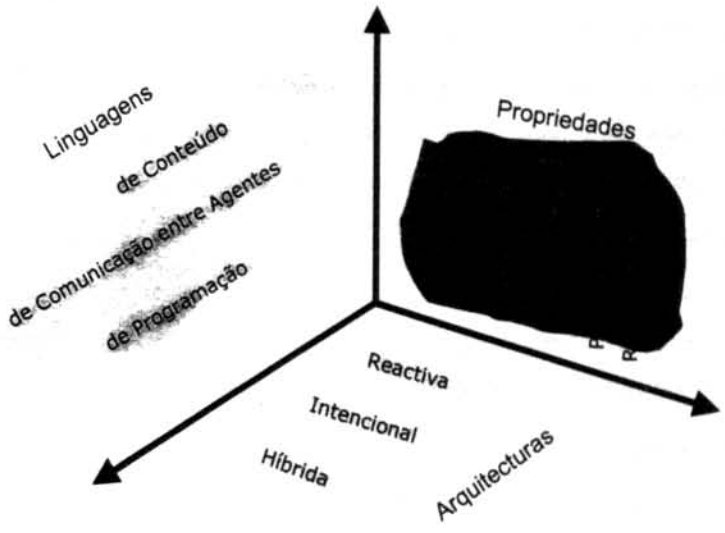
Jennings e Wooldridge classificam os agentes segundo dois tipos de utilização: uma fraca e simultaneamente geral e outra mais forte e específica (Wooldridge and Jennings, 1995). No primeiro tipo, em certo sentido mais consensual, define-se um agente como um sistema ou processo computacional com as seguintes propriedades (já atrás referidas):

1. *autonomia* – os agentes operam sem intervenção externa, sendo responsáveis pelas suas acções e pelo seu estado interno;
2. *sociabilidade* – os agentes interagem entre si através de uma linguagem de comunicação comum;
3. *reactividade* – os agentes têm percepção sobre o ambiente em que operam (são sensores desse ambiente) e respondem às mudanças nele verificadas;
4. *pró-actividade* – os agentes possuem capacidades, não apenas de reacção, mas de iniciativa nas acções que executam.

No segundo tipo, os autores caracterizam os agentes através de conceitos usualmente aplicáveis aos seres humanos, nomeadamente: conceitos de carácter racional do tipo conhecimento, crença, intenção e obrigação; ou ainda de carácter emocional, tais como credibilidade e personalidade.

No âmbito deste trabalho, considera-se agentes como entidades autónomas com capacidades próprias de execução de tarefas e que podem beneficiar de actividades complementares e concorrentes, executadas por outros agentes, quando não estão aptos a atingir determinados objectivos isoladamente. Além disso, considera-se aqui que os agentes devem ser "vistos" segundo três perspectivas indissociáveis (ver Figura

6.1): as suas propriedades, as arquitecturas que os integram e as linguagens que usam para, por exemplo, comunicar com os outros agentes.



BIBLIOTECA F.E. PORTO

Figura 6.1 – Perspectivas principais dos "agentes inteligentes" (adaptado de (Oliveira, 1998))

Dados os objectivos deste trabalho, não se vai naturalmente apresentar aqui o "estado da arte" dos sistemas multi-agente ou dos agentes inteligentes (segundo as dimensões referidas na Figura 6.1, ou segundo qualquer outra classificação). De entre a vasta bibliografia nesta área, sugere-se a leitura do "estado da arte", conciso mas completo, apresentado em (Oliveira, 1998).

A interoperabilidade nos sistemas multi-agente

A interoperabilidade nos sistemas MAS levanta questões cruciais, no sentido em que se tem de "compreender" e integrar o comportamento e as intenções dos agentes envolvidos. No caso geral, uma abordagem MAS pressupõe o uso de uma metodologia que permita a resolução de vários problemas, nomeadamente: a formalização da especificação do domínio de aplicação; os protocolos de comunicação inter-agentes; a coordenação inter-agentes; a eficiência computacional; as ferramentas e técnicas de implementação e os métodos de verificação (no sentido de permitirem, por

exemplo, monitorar, diagnosticar e corrigir sistemas multi-agente) (Oliveira et al., 1999).



Numa comunidade de agentes autónomos, para que se consiga garantir um nível adequado de cooperação, os agentes devem possuir capacidade de comunicação, e para tal deverão estar dotados dos seguintes componentes (Finin et al., 1997):

- *linguagem de conteúdo* (ou *linguagem de expressão de interacção*) – permite o entendimento comum do conhecimento partilhado entre os agentes;
- *linguagem de comunicação* – meio através do qual são comunicadas as atitudes e acções, expressas através da linguagem de conteúdo; determina quais os tipos de interacções permitidas, descrevendo igualmente a linguagem de conteúdo utilizada e a ontologia considerada;
- *protocolo de transporte* – mecanismo de comunicação (baixo-nível) das mensagens inter-agentes (por exemplo, TCP, SMTP, CORBA e HTTP).

Na abordagem multi-agente proposta neste trabalho, adoptou-se como linguagem de comunicação entre agentes, a linguagem KQML (*Knowledge Query Manipulation Language*). O KQML possui uma grande variedade de "actos de comunicação" que, entre outras características interessantes, permitem assegurar e contribuir para a "sociabilidade" de um agente inteligente.

A linguagem de conteúdo considerada, denominada AVEL *(Abstract Virtual Enterprise Language*), tem por base a ontologia desenvolvida para a *Empresa Virtual* (ver secções seguintes) e além disso é baseada em KIF (*Knowledge Interchange Format*).

"Speech Acts"

As linguagens de comunicação entre agentes, e em particular a linguagem KQML, inspiram-se na teoria dos "actos da fala" (*Speech Acts*) desenvolvida por (Austin, 1962) e (Searle, 1969) que preconiza que "falar é actuar". Segundo esta teoria, o "acto da fala" pode assumir três formas distintas:

- *acto da locução* – corresponde à expressão mental e física associada à comunicação usando uma gramática e um vocabulário;

- *acto ilocucional* – consiste na transmissão da intenção do orador para o ouvinte (por exemplo, quando se enuncia "São dez horas?", pode-se inferir que se trata de uma afirmação ou de uma interrogação, contudo, é o acto ilucucional que permite compreender que se trata de uma interrogação);

- *acto perlocucional* – "captura" todos os efeitos (directos ou indirectos) que o acto da fala pode induzir nas atitudes mentais do ouvinte, isto é, tem que ver com as acções do acto da fala que não se encontram inscritas directamente no enunciado, mas que dependem da situação de comunicação.

A teoria identifica diversos verbos, denominados pelos autores de "*performatives*", com força ilocucional, tais como, desejar, pedir, confirmar, declarar, prometer, etc. Seguindo esta teoria essencialmente no que respeita ao acto ilocucional da fala, o qual constitui o fundamento teórico das linguagens de comunicações de agentes, a linguagem KQML propõe diversas *performativas* com o propósito de criar "actos comunicacionais" entre agentes (por exemplo, durante processos de negociação).

Os "actos de fala" são geralmente agrupados nas seguintes categorias (Singh, 1998), (Oliveira, 1998), cujas designações não se traduziram, para evitar deturpar o seu significado:

- *assertives* – relativa a asserção e informação (por exemplo, "a encomenda está atrasada");

- *directives* – relativa a dar uma ordem ou pedir algo (por exemplo, "verifique a capacidade de produção");

- *commissives* – relativa a prometer algo (por exemplo, "eu cumpro com a data de entrega");

- *permissives* – relativa a dar permissão ou autorização (por exemplo, ("pode iniciar a produção");

- *prohibitives* – relativa a interditar algo (por exemplo, "não pode iniciar a produção");

- *declaratives* – relativa a declarações (por exemplo, "esta encomenda é prioritária");

- *expressives* – relativa a expressão de emoções e avaliações (por exemplo, "desejava cumprir com a data de entrega");
- *interrogative* – relativa a uma interrogação (por exemplo, "pode iniciar a produção?").

De seguida, referem-se resumidamente, algumas das características e princípios fundamentais da linguagem KQML e da especificação KIF. Pela sua importância na contribuição para a formalização da especificação do domínio de aplicação, apresenta-se também a ontologia desenvolvida para a EV.

6.1.2 *Knowledge Query Manipulation Language* (KQML)

A linguagem KQML é uma linguagem e um protocolo de alto nível para troca de informação e conhecimento entre agentes (Finin et al., 1997).

O KQML coloca a ênfase nas regras que regem o discurso entre os agentes comunicantes, incluindo, por isso, como saber com quem comunicar, como encontrar o receptor, e como iniciar e manter a comunicação. A linguagem pressupõe que a arquitectura multi-agente a que se destina, possui uma classe especial de agentes que organizam o processo de comunicação, designados por agentes "facilitadores" (*facilitators*). Estes agentes, entre outras funcionalidades, disponibilizam serviços de procura de recursos e de encaminhamento de mensagens; mantêm um registo de descrições de serviços e necessidades de outros agentes; promovem a decomposição de pedidos complexos em sub-pedidos a serem enviados a vários agentes, etc.

O KQML suporta protocolos de interacção síncronos (uma pergunta de um agente emissor origina uma resposta do agente receptor) e assíncronos (uma pergunta pode originar diversas respostas irregularmente espaçadas no tempo).

A linguagem encontra-se conceptualmente estruturada em três camadas (ver Figura 6.2). A primeira camada, denominada *camada de conteúdo*, contém a mensagem propriamente dita (conteúdo) numa representação da *linguagem de conhecimento* utilizada, por exemplo, numa linguagem baseada em KIF. A segunda camada, denominada *camada de mensagem*, constitui a essência da linguagem. Define os tipos de interacções, a linguagem usada na expressão de conteúdo, a ontologia utilizada, o protocolo de entrega, etc. A última camada, denominada *camada de transporte*, diz respeito à mecânica

da comunicação, tal como, a definição do emissor e do receptor, a identificação do destino e o protocolo de comunicação (por exemplo, TCP, SMTP, CORBA, etc.).

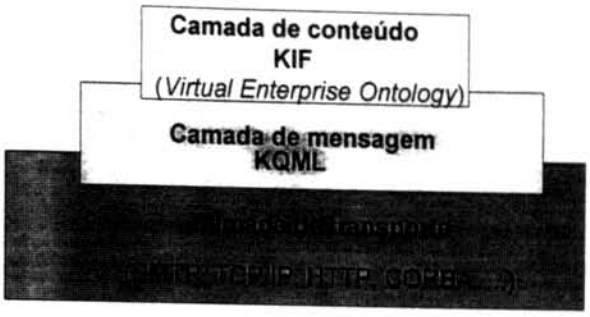


Figura 6.2 – Estrutura em camadas da linguagem KQML

O KQML é composto por várias primitivas de alto nível, designadas na literatura por "performativas" ou "actos de comunicação", agrupadas em três categorias:

- *"performativas" de conversação ou de discurso* – envolvem todos os actos de comunicação, que do ponto de vista linguístico estão mais próximos dos "actos de fala", e que dizem respeito à troca de informação e de conhecimento entre agentes (exemplos: com carácter informativo, *tell*, *achieve*, *untell*, *unachieve*, etc.; com carácter de pedido, *ask-if*, *ask-one*, *ask-all*, etc.);

- *"performativas" de controlo e intervenção* – agrupam actos de comunicação associados essencialmente ao controlo do fluxo normal de uma conversação entre agentes (exemplos: para sinalização de erros ou impossibilidades, *error*, *sorry*, *discard*, *next*, *ready*, etc.);

- *"performativas" de gestão e organização* – são basicamente orientadas aos agentes organizadores ("facilitators") de um sistema multi-agente, com o propósito de facilitarem o processo de gestão da rede de agentes (exemplos: *register*, *forward*, *broadcast*, *recruit-one*, *broker-one*, etc.).

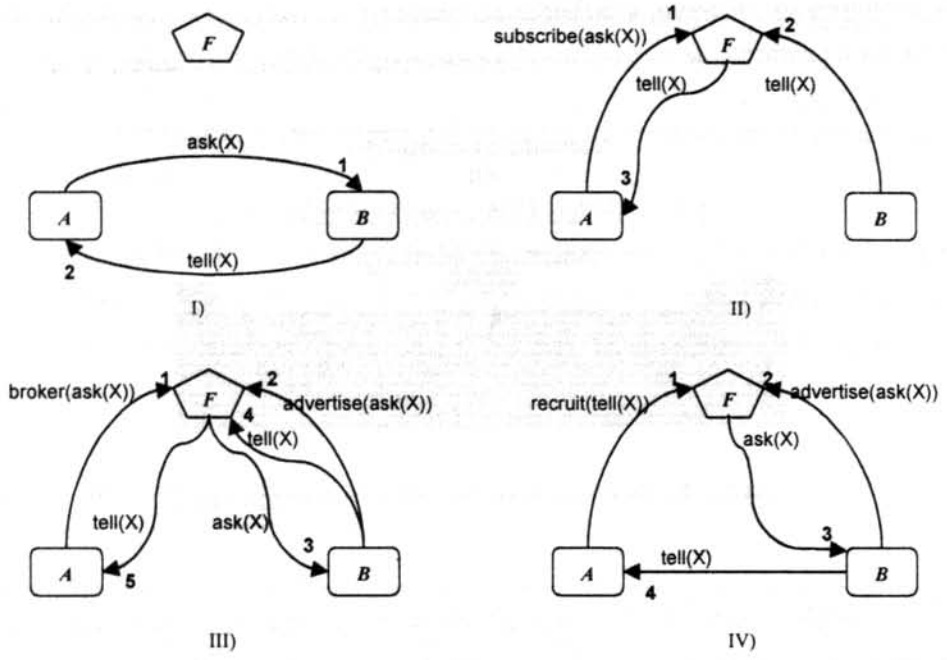


Figura 6.3 – Cenários de interacção entre agentes usando *performativas* KQML (adaptado de (Finin et al., 1997))

A Figura 6.3 apresenta alguns cenários possíveis na interacção entre agentes, usando *performativas* KQML.

O caso I) ilustra uma interacção ponto-a-ponto entre dois agentes (A e B). Nesta situação, o agente A pergunta directamente ao agente B acerca de X. Esta interacção pressupõe que o agente A conhece o agente B e que B contém, na sua base de conhecimento, informação relativa a X.

O cenário II) ilustra uma situação em que o agente A recorre ao agente facilitador F, "subscrevendo" um pedido para que este o mantenha informado acerca de X. Sempre que F é informado acerca de X, por exemplo através do agente B, o agente A é automaticamente informado.

No caso apresentado em III), o agente A solicita ao agente F que encontre um agente com a capacidade de processar a *performativa ask(X)*. Por sua vez o agente B informa o agente F que se encontra apto a aceitar *performativas* do tipo *ask(X)*. O agente F funciona como um intermediário, encaminhando pedidos para o agente B e respostas para o agente A.

Por fim, no caso IV), a performativa *recruit* solicita a um agente facilitador que encontre um agente adequado para processar uma determinada performativa (especificada num campo específico da mensagem). Nesta situação a resposta é enviada directamente para o agente que iniciou a interacção, neste caso, o agente A.

Uma descrição completa e pormenorizada de todas as "performativas" da linguagem KQML encontra-se em (Labrou and Finin, 1997).

Arquétipo de uma expressão KQML

Uma expressão completa KQML inclui além da "performativa" representativa do acto de comunicação, uma lista de pares atributo-valor que definem entre outros aspectos, o emissor, a linguagem, a ontologia, etc. A Tabela 6.1 apresenta, de uma forma sucinta, cada um dos parâmetros que compõem uma expressão KQML.

Atributo	Descrição
"performativa"	acto de comunicação desejado para a mensagem
:sender	o agente remetente da "performativa"
:receiver	o agente destinatário da "performativa"
:from	define a origem da "performativa" em *:content* quando é usada a "performativa" *forward*
:to	define o destino da "performativa" em *:content* quando é usada a "performativa" *forward*
:in-reply-to	define a "etiqueta" de resposta a uma mensagem anterior (valor igual ao contido em :reply-with da mensagem anterior)
:reply-with	valor esperado da "etiqueta" à mensagem actual
:language	nome da linguagem na qual está expresso o conteúdo da comunicação
:ontology	nome da ontologia que os agentes receptores devem usar para a interpretação da mensagem
:content	informação acerca do conteúdo da comunicação

Tabela 6.1 – Sumário das palavras-chave da linguagem KQML e respectivo significado

Apresenta-se na Tabela 6.2 alguns exemplos de "performativas" e respectivo significado, associadas ao "acto de fala" correspondente.

performativa	***acto de fala***	**Significado**	***:content***
ask-if	*interrogative*	o emissor quer saber se o *:content* se encontra na base de conhecimento do receptor	<expressão>
ask-one	*interrogative*	o emissor quer conhecer uma instância de *:content* que se encontra no receptor	<expressão>
tell	*assertive*	o emissor informa o receptor que a <expressão> em *:content* existe na sua base de conhecimento	<expressão>
untell	*assertive*	o emissor informa o receptor que a <expressão> em *:content* não existe na sua base de conhecimento	<expressão>
insert	*declarative*	o emissor pede para o receptor adicionar à sua base de conhecimento a <expressão> em *:content*	<expressão>
achieve	*directive*	o emissor pede ao receptor para tentar executar (tornar verdadeiro) a <expressão> em *:content*	<expressão>
advertise	*commissive*	o emissor faz saber ao receptor que está apto a processar as mensagens presentes em *:content*	<performativa>
subscribe	*interrogative*	o emissor pede para ser actualizado sempre que o receptor responda à performativa presente em *:content*	<performativa>
broadcast	*directive*	o emissor pede que a peformativa em *:content* seja enviada a todos os agentes registados no receptor	<performativa>
recommend-one	*interrogative*	o emissor pede ao receptor que sugira um agente que possa processar o *:content*	<performativa>
broker-all	interrogative	o emissor pede que encontrem todos os agentes com capacidade de processar o *:content*	<performativa>

Tabela 6.2 – Exemplos de "performativas"

6.1.3 *Knowledge Interchange Format* (KIF)

Quando os agentes comunicam, devem partilhar uma *ontologia* que descreva, de forma estruturada, os conceitos, as entidades e as relações existentes num determinado domí-

nio. A linguagem KQML pressupõe que o conteúdo da mensagem, com significado no domínio em que os agentes actuam, se encontre formalizado numa linguagem de representação de conhecimento.

O KIF (Genesereth and Fikes, 1992) é uma linguagem formal de representação de conhecimento desenvolvida especificamente para poder ser usada como linguagem de conteúdo, numa linguagem de comunicação entre agentes tal como o KQML.

A linguagem inclui uma semântica declarativa, isto é, o significado das expressões pode ser compreendido sem o recurso a um interpretador para manipular essas expressões; é logicamente compreensiva, isto é, tem capacidade de exprimir frases arbitrárias em lógica de primeira ordem (cálculo de predicados); permite a definição de termos e a representação do conhecimento acerca do próprio conhecimento.

O KIF fornece basicamente quatro tipo de expressões: termos, frases, regras e definições. *Termos* são usados para representar as entidades na realidade que se está a descrever. *Frases* permitem descrever factos dessa realidade. *Regras* permitem definir passos admissíveis num processo de inferência. *Definições* são usadas para definir constantes. Além disso, são disponibilizados variados operadores lógicos para apoiar a codificação de informação lógica.

Pormenores técnicos acerca da linguagem KIF assim como uma descrição detalhada da mesma podem ser encontrados em (Genesereth and Fikes, 1992).

6.2 Uma ontologia para a Empresa Virtual

A necessidade de desenvolver uma *ontologia*, no âmbito do trabalho apresentado nesta dissertação, resultou, em primeiro lugar, das dificuldades surgidas pela profusão de conceitos e termos em jogo e pela confrontação de diferentes perspectivas, no caso específico do projecto de um sistema de planeamento e controlo da produção para a *Empresa Virtual*.

A nossa experiência mostrou o que outros tinham igualmente já referido, quanto ao facto de que a confusão terminológica dá origem à confusão conceptual e vice-versa (Bradshaw et al., 1992). Esta constatação é particularmente mais notória nas fases preliminares do processo de concepção e projecto de um sistema. Uma forma de minimizar estas dificuldades consiste no estabelecimento de um núcleo terminológico

e conceptual de base, que assuma um papel mediador entre os actores envolvidos. Esta foi a principal razão para o desenvolvimento da Ontologia para a *Empresa Virtual – Virtual Enterprise Ontology*.

6.2.1 A noção de ontologia

No âmbito da representação do conhecimento de um determinado domínio, o termo *ontologia* refere-se a um conjunto de descrições formais das entidades que constituem o domínio, bem como das suas propriedades e relações. Em termos gerais, a principal finalidade de uma ontologia é constituir um entendimento partilhado do domínio que, depois de formalizado, seja utilizável em várias aplicações que vão desde uma plataforma de comunicação entre os intervenientes no desenvolvimento do sistema até à comunicação entre agentes de *software* (Soares, 1998).

Constituindo actualmente as ontologias uma área de investigação muito activa, em particular no campo da Inteligência Artificial, e apesar do seu propósito ser "normalizar", é difícil apresentar uma definição abrangente e única. Segundo (Gruber, 1993) "*Uma ontologia é a especificação da conceptualização*". Em (Guarino and Giaretta, 1995) é feita uma revisão das várias interpretações da noção de ontologia. Uma definição adequada parece ser a proposta por (Gruber, 1994): "*Ontologias são acordos sobre conceptualizações partilhadas. Conceptualizações partilhadas incluem quadros conceptuais para a modelação do conhecimento de um domínio, protocolos de conteúdo específico entre agentes inter-operantes, e acordos sobre a representação de teorias particulares de domínio. No contexto da partilha de conhecimento, as ontologias são especificadas sob a forma de definições do vocabulário representacional.*"

6.2.2 Objectivos de uma ontologia

No âmbito da modelação empresarial e dos sistemas multi-agente, identificam-se três categorias principais de situações nas quais as ontologias podem naturalmente ser utilizadas (Uschold and Gruninger, 1996):

- comunicação entre pessoas e organizações, através da constituição de uma base partilhada para entendimento do domínio;
- inter-operabilidade entre sistemas;

- engenharia de sistemas: a ontologia apoia o processo de projecto e desenvolvimento de *software*; fomenta a reutilização (a representação formal pode ser reutilizada e/ou partilhada por vários sistemas ou componentes de *software*); melhora a fiabilidade (uma representação formal é mais consistente) e contribui para a especificação com a identificação de requisitos.

Naturalmente que os objectivos que orientam o seu desenvolvimento e as suas aplicações, condicionam o "propósito" (*purpose*) da ontologia.

6.2.3 Tipos de ontologias

Da mesma forma que a definição de ontologia não é consensual, também a classificação das ontologias é feita de diferentes formas, de acordo com os diferentes pontos de vista e perspectivas. Considera-se aqui a taxonomia proposta por (Bradshaw et al., 1992):

- *ontologias genéricas* – definem termos suficientemente genéricos e frequentemente usados como base para a definição de outros termos; constituem, em certo sentido, meta-ontologias;
- *ontologias de domínio* – são baseadas nas ontologias genéricas, especializando, para determinadas áreas, os termos e conceitos aí definidos;
- *ontologias de modelação e representação* – facultam as primitivas que são usadas nas ontologias genéricas e de domínio;
- *ontologias de aplicação* – contêm as definições necessárias à modelação do conhecimento de uma área específica e tipicamente envolvem conceitos derivados de ontologias genéricas e de domínios.

O grau de formalização utilizado dá origem a uma outra classificação para as ontologias, nomeadamente, *ontologia informal* (quando expressa em linguagem natural), *ontologia semi-informal* (quando expressa numa forma restrita e controlada da linguagem natural), *ontologia semi-formal* (quando expressa numa linguagem formal, artificialmente definida para o efeito) e, finalmente, *ontologia formal* se expressa numa linguagem lógica, com uma semântica formalmente definida e com teoremas e provas de propriedades (Soares, 1998).

6.2.4 Ontologia da *Empresa Virtual* (*Virtual Enterprise Ontology*)

Descreve-se nesta secção, e de uma forma sintética, o núcleo conceptual desenvolvido na ontologia semi-informal designada por "*Virtual Enterprise Ontology*" (VEO).

Refira-se que o processo de desenvolvimento de uma ontologia envolve, pela sua natureza, inúmeras iterações, exigindo um refinamento contínuo. Assim sendo, não se pode nunca considerar que uma ontologia esteja definitivamente concluída.

6.2.4.1 Fontes de conhecimento

A VEO foi desenvolvida a partir da *Enterprise Ontology* (EO) que resultou do *Enterprise Project*, liderado pela Universidade de Edimburgo (Artificial Intelligence Applications Institute - AIAI). Este projecto teve como objectivo global desenvolver um enquadramento para integrar métodos e ferramentas, adequados para a modelação empresarial e a gestão da mudança (Uschold et al., 1996).

Porque em certo sentido se pode considerar a VEO como uma extensão da EO (que constitui o seu núcleo), nomeadamente no suporte ao paradigma das "organizações virtuais", apresenta-se aqui a estrutura base da EO.

A conceptualização subjacente à EO é baseada num conjunto de conceitos, relações e axiomas que decorrem naturalmente da vida de uma empresa, tais como actividades, processos, planos, recursos, capacidades, organização, estratégia, marketing e tempo.

A EO encontra-se estruturada nas seguintes secções:

- *meta-ontologia* – define alguns conceitos básicos fundamentais, tais como *entidade*, *relação*, *papel*, *actor*, etc.;
- *actividade, plano, capacidade e recurso* – define termos e conceitos relacionados com os processos e com o planeamento, tais como *actividade*, *plano*, *capacidade*, *qualificação*, *autoridade*, etc.;
- *organização* – define os conceitos centrais ligados à organização, nomeadamente, *unidade organizacional*, *entidade legal* e *ligação de gestão*;

- *estratégia* – define o conceito base de *propósito* ("purpose"); as unidades organizacionais são responsáveis por *propósitos* quando geram outras unidades; o conceito de *estratégia* é definido como um *plano* para atingir um *propósito estratégico*;

- *marketing* – define termos e conceitos de base, associados à *venda*; a *venda* é um acordo acerca do *preço de venda* entre duas entidades legais, sobre um *produto*;

- *tempo* – define conceitos base associados ao tempo e a calendários.

A EO foi, pelos seus autores, formalizada e codificada na linguagem Ontolingua[36], podendo ser reutilizada em diversas aplicações.

Uma outra fonte para a VEO foi a investigação levada a cabo no âmbito do projecto TOVE (Toronto Virtual Enterprise Project) (Fox and Gruninger, 1994) e na ontologia de escalonamento denominada OZONE (Smith and Becker, 1996).

Uma fonte adicional de informação para este trabalho foi a especificação SEMATECH, desenvolvida no âmbito da indústria de semicondutores (Sematech, 1996). Apesar de ser orientada a aspectos fundamentalmente operacionais, identificaram-se mesmo assim alguns conceitos que, após as necessárias adaptações, puderam ser incluídos na VEO.

6.2.4.2 Estrutura e conteúdo da VEO

A ontologia VEO, em cujo desenvolvimento o autor teve uma participação significativa, encontra-se estruturada segundo as secções seguintes:

1. *Enterprise Ontology* – onde se incluem os termos, conceitos e relações derivados da EO e tomados como base na VEO;
2. *Gestão de planos e de encomendas* – onde são identificados conceitos tais como *plano de produção, plano global, plano local, planeamento "grosseiro"* e *planeamento fino*; a gestão de encomendas é igualmente objecto desta secção, são caracterizados termos tais como *cotação, data pedida, data de entrega*, etc.;

---

[36] A Ontolingua é uma linguagem para codificar formalmente ontologias, sendo essencialmente baseada na sintaxe e semântica da linguagem KIF, com funcionalidades adicionais particularmente adequadas à representação de ontologias.

3. *Actividades e recursos* – onde se apresenta o conceito central de actividade e as suas relações com outras entidades como recurso; o conceito de capacidade é igualmente caracterizado;
4. Organização em rede/extendida – para definição de conceitos como *Empresa Virtual*, *unidade de empresa virtual* e *unidade organizacional*.

Refira-se que, dado o contexto em que este trabalho foi realizado, a ontologia foi desenvolvida em língua inglesa. Naturalmente que, pretendendo-se um elevado rigor e precisão, há subtilezas linguísticas de difícil ou mesmo impossível tradução.

No Apêndice B, é apresentada parte da VEO, com extractos de cada uma das secções aqui referidas.

6.3 Uma arquitectura multi-agente

6.3.1 Breve revisão bibliográfica e contexto

Encontram-se na literatura inúmeras referências a sistemas baseados em agentes, para a resolução de problemas complexos, nomeadamente, problemas de planeamento, problemas de optimização, etc (Fischer, 1998), (Marik et al., 1998). Um resumo de aplicações a problemas essencialmente industriais pode ser encontrado em (Parunak, 1998). Em (Oliveira, 1999), o autor apresenta um conjunto alargado de aplicações de sistemas multi-agente, nomeadamente nas áreas de "sofbots" (agentes de software em ambientes virtuais, isto é, não físicos), sistemas inteligentes de manufactura, organizações virtuais, comércio electrónico, gestão de tráfego, gestão de redes e entretenimento.

A maioria das abordagens multi-agente descritas na literatura é, em certo sentido, orientada a problemas muito específicos, como por exemplo, problemas de escalonamento de tarefas em sistemas FMS. Muito reduzido parece ser o número de aplicações a problemas de carácter mais táctico e menos estruturados, como por exemplo, os problemas do planeamento da produção em redes de empresas.

Trabalhos de investigação mais directamente relacionados com o objecto desta dissertação são, por exemplo, os desenvolvidos no Enterprise Integration Laboratory da Universidade de Toronto. Contudo, nesses trabalhos, a arquitectura proposta para

a gestão integrada da cadeia de fornecimento, baseada em agentes inteligentes (Fox et al., 1993), não se adequa, na nossa opinião, às exigências das novas organizações. Os autores apresentam uma arquitectura constituída por diversos agentes inteligentes que modelam as diferentes funcionalidades tradicionalmente encontradas nas empresas industriais (por exemplo, planeamento, distribuição, gestão de transportes, gestão de recursos, etc.). Esta abordagem é, em certo sentido, limitada, pois a opção de modelar funcionalidades em vez de modelar entidades (tais como fábricas, unidades logísticas, unidades de armazenamento, etc.) não é suficientemente flexível para incorporar novos processos de negócio que resultam, frequentemente, da dinâmica própria das redes de empresas.

A arquitectura aqui proposta opta por modelar, mediante agentes inteligentes, as entidades constituintes de uma organização (entidades operacionais), conseguindo-se pela sua interacção e cooperação implementar então as funcionalidades da rede de empresas em análise.

6.3.2 Arquitectura e tipos de agentes

A solução proposta neste trabalho é baseada num conjunto de agentes distribuídos espacialmente, comunicando e cooperando entre si. Inspirada nas arquitecturas que utilizam KQML, a arquitectura multi-agente proposta pressupõe dois grupos principais de agentes:

- *agentes organizacionais* – são agentes de serviços ou de gestão, isto é, que têm por função principal disponibilizar serviços de encaminhamento e de organização de mensagens entre agentes de características funcionalmente similares; os agentes organizacionais, por si só, não desempenham directamente qualquer função operacional no âmbito da *Empresa Virtual* (EV);

- *agentes funcionais* – são agentes operacionais, isto é, desempenham um papel activo no âmbito da EV; encapsulam as diferentes entidades que compõem a EV (por exemplo, as unidades de produção e logísticas).

A *Empresa Virtual* é modelada directamente por diversos agentes funcionais, de diferentes tipos ou funções (constituindo uma "comunidade" de agentes), e por um

ou mais agentes organizacionais que constituem uma infra-estrutura de suporte aos agentes funcionais.

A Figura 6.4 apresenta a classificação dos agentes considerados na arquitectura multi-agente proposta neste trabalho. Descreve-se de seguida cada um destes tipos de agentes.

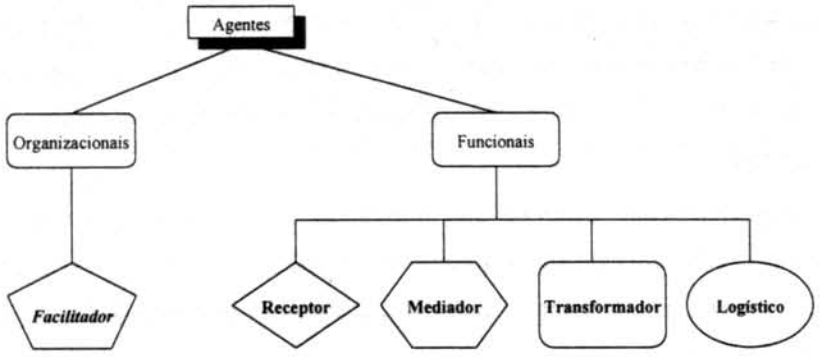


Figura 6.4 – Tipos de agentes

Agente Receptor (AR)

Este agente é responsável pela recepção de pedidos provenientes dos clientes, representando um ponto de entrada na *Empresa Virtual*. Podem existir tantos agentes receptores quanto os pontos de entrada desejados, ou seja, é possível submeter os pedidos dos clientes a partir de diferentes pontos, geograficamente distribuídos.

Agente Mediador (AM)

Este agente tem a função principal de coordenar toda a EV. Representa, do ponto de vista exterior, a organização à qual os clientes submetem pedidos e da qual recebem ofertas. Os pedidos dos clientes são submetidos ao agente mediador através dos agentes receptores. Numa determinada EV, só pode existir um agente mediador.

Em termos conceptuais, o agente mediador é a entidade responsável por toda a EV, desde a sua fase de criação, de formação/configuração e de operação até à fase de dissolução. É igualmente responsável por iniciar o processo de planeamento, de cada

pedido submetido, e por avaliar as ofertas globais produzidas pelo conjunto de agentes transformadores (AT) envolvidos.

Agente Transformador (AT)

O objectivo deste agente é modelar cada unidade produtiva inserida numa EV. Basicamente, este agente está habilitado a executar uma determinada actividade, proposta pelo agente mediador, de acordo com a sua competência específica.

Os agentes AT recebem propostas do agente mediador e, mediante cooperação com outros agentes AT que lhe são adjacentes e com agentes AL (agentes logísticos), geram ofertas.

Numa EV, existem tantos agentes AT quantas as unidades de produção.

Agente Logístico (AL)

Este agente é responsável pelas actividades logísticas envolvidas na EV. Do ponto de vista funcional, está associado à gestão do movimento de produtos e materiais entre unidades produtivas. No sistema proposto, podem existir mais do que um agente AL, organizados quer geograficamente (por exemplo, agentes responsáveis pelos serviços de transporte numa determinada região), quer do ponto de vista lógico (por exemplo, agentes representativos de um tipo específico de transporte). Em princípio, cada agente AL corresponderá a um operador logístico.

Agente Facilitador (AF)

Este tipo de agente é responsável pela gestão dos agentes funcionais. Basicamente, os AF disponibilizam serviços de procura de agentes de um determinado tipo ou especialidade, mantêm igualmente um registo de agentes AT ou AL e disponibilizam informação acerca de agentes AT ou AL a agentes AM de diferentes organizações.

6.3.3 Criação da rede de agentes

A interacção e a cooperação de agentes pressupõem o estabelecimento de uma *rede de agentes*, mediante o *registo* de cada agente funcional num ou mais agentes organizacionais. As acções de registo permitem criar a rede de agentes por forma a que, posteriormente, estes possam ser organizados e ou configurados de acordo com a especificidade da EV a criar.

A Figura 6.5 representa um diagrama genérico do processo de criação da rede de agentes. As *performativas* KQML envolvidas são: para a entrada na rede, *register;* e para a saída, *unregister*. Estas acções podem ocorrer em qualquer estado de operação da rede de agentes, isto é, mesmo numa fase em que estejam constituídas uma ou mais EV, podem verificar-se entradas ou saídas de agentes na rede.

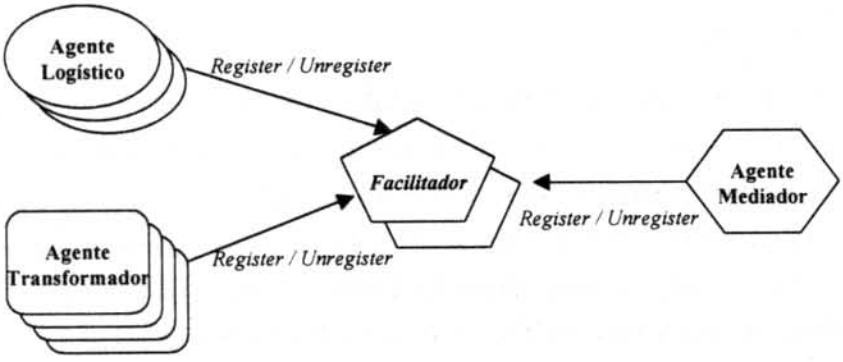


Figura 6.5 – Criação da rede de agentes

6.3.4 Organização dos agentes

O ambiente distribuído inerente à EV, por um lado, e a especificidade e a organização de algumas das entidades operacionais que a constituem, por outro, sugere que os agentes se organizem segundo dois tipos de associações: uma geográfica e outra lógica (Azevedo and Oliveira, 1999). Assim, numa EV podem ter-se diferentes regiões geográficas, envolvendo diversos agentes AT e eventualmente um ou mais AL. Da mesma forma, por razões de eficiência, diversos agentes AT e/ou agentes AL poderão estar organizados segundo uma ou mais associações lógicas (por exemplo, uma associação de

agentes da mesma especialidade, isto é, com as mesmas aptidões a executarem determinado tipo de tarefas).

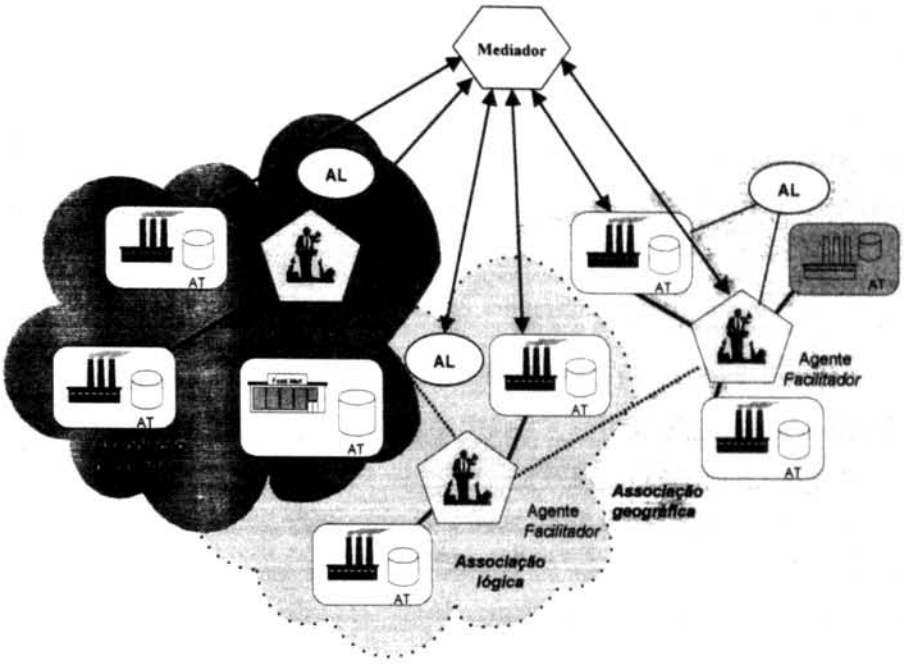


Figura 6.6 – Associação geográfica ou lógica dos agentes

Os agentes AF têm um papel importante nestes processos de associação, no sentido em que são as entidades responsáveis pela sua organização e gestão. Por exemplo, um agente AF inserido numa associação lógica pode representar um determinado estágio do processo produtivo (por exemplo, o fabrico de *wafers* no caso dos semicondutores).

A Figura 6.6 ilustra a organização dos diferentes tipos de agentes, segundo os dois tipos de associação possíveis. São evidenciadas igualmente na figura, algumas das interligações (lógicas) entre os agentes. Para facilitar a leitura, não estão representados na figura, os agentes AR.

6.3.5 Criação, configuração e organização da EV

De acordo com o exposto no capítulo 3, as fases de criação, configuração e organização da EV envolvem, fundamentalmente, a procura e a selecção das entidades com as quais será constituída a EV.

Procurando tirar partido das potencialidades dos agentes inteligentes, e em particular da linguagem de comunicação considerada, nas fases de criação, configuração e organização da EV, é fundamental que se consiga uma forma de explicitar e dar a conhecer informação sobre os diferentes agentes, nomeadamente:

- os "interesses" particulares dos agentes AT e AL;
- os "interesses" do agente mediador (AM);
- a disponibilidade, de um ou mais agentes, de integrar uma EV para produzir um determinado tipo de produtos;
- anúncio das competências dos agentes.

Os agentes organizacionais (agentes AF, distribuídos espacialmente e organizados segundo associações lógicas e/ou geográficas) têm um papel muito importante nestas fases. Toda a informação que os agentes funcionais pretendem ser divulgada ou difundida, deve ser enviada aos agentes AF. Em termos gerais, estes agentes mantêm internamente informação, estruturada de acordo com a sua organização geográfica ou lógica, acerca dos agentes funcionais.

Para suporte às fases de criação, configuração e organização da EV, são propostos dois cenários principais:

1. os agentes AT e AL anunciam os seus interesses, isto é, apresentam como que uma "declaração de intenções" aos agentes facilitadores, relativa ao tipo de pedidos (mensagens) que estão interessados em receber;
2. os agentes AT e AL "subscrevem" o seu interesse em determinado tipo de produtos ou família de produtos.

Cenário 1

Basicamente, neste cenário, tem lugar uma "declaração de intenções" por parte dos agentes com interesse em participarem em futuras EV. Assim, numa primeira fa-

se, os agentes do tipo AT e/ou AL anunciam os seus interesses, num ou mais agentes AF. Nesta fase, o conteúdo das mensagens, associada à "declaração de intenções", envolve, para além da especificação das mensagens em que está interessado em receber, um ou mais dos seguintes argumentos:

<skill-template> - o agente informa o AF acerca das suas capacidades específicas (competências) ou aptidões para executar determinado tipo de actividades;

<product-family-template> - o agente declara o seu interesse num determinado tipo ou família de produtos;

<product-template> - o agente declara o seu interesse num produto específico;

<order-template> - o agente declara o seu interesse num determinado tipo de ordens definidas no domínio em que se insere.

Após esta fase de "declaração de intenções", estão criadas as condições para que se possa constituir uma EV, por iniciativa de um agente mediador. Nessa situação, o agente mediador procede à procura de parceiros através dos agentes AF, recrutando aqueles que obedecem às especificações por si definidas. Em qualquer instante, podem ocorrer alterações na configuração da EV, quer por entrada de novas entidades, quer por saída (fim da relação de cooperação).

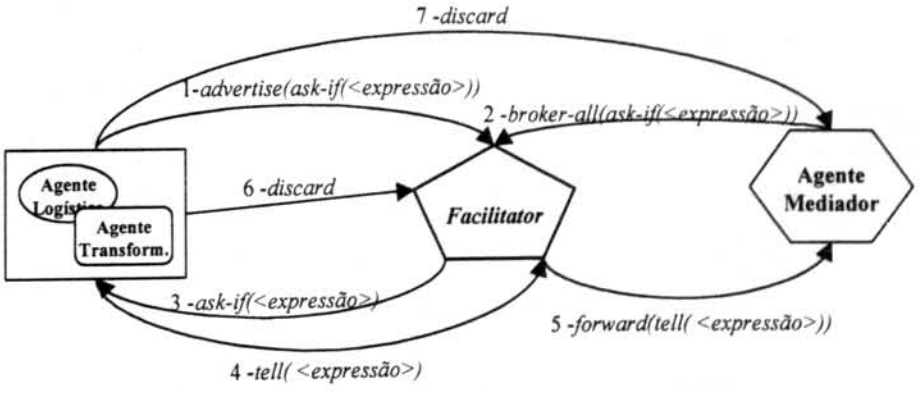


Figura 6.7 – Sequência de mensagens trocadas nas fases de criação, configuração e organização da EV (cenário 1)

Fase	Descrição	*performativas*
1	"declaração de intenções"	1
2	procura de parceiros	2, 3
3	constituição da EV	4,5
4	fim da relação de interesse	6, 7

Tabela 6.3 – Tipos de *performativas* KQML envolvidas nas fases de criação, configuração e organização da EV

Na Figura 6.7 representa-se um diagrama da sequência de mensagens trocadas e das *performativas* KQML envolvidas nas fases de criação, configuração e organização da EV. A Tabela 6.3 sintetiza, de acordo com o representado na figura, as fases principais e as *performativas* envolvidas nessas fases.

Numa variante deste cenário, apresentada na Figura 6.8, o agente mediador recorre à performativa *recruit-all* para anunciar a agentes AF o interesse em "recrutar" agentes AT e AL com determinadas competências. Nesta situação, os agentes AT e AL interessados respondem directamente ao agente AM, isto é, na resposta não se verifica a intervenção dos agentes AF.

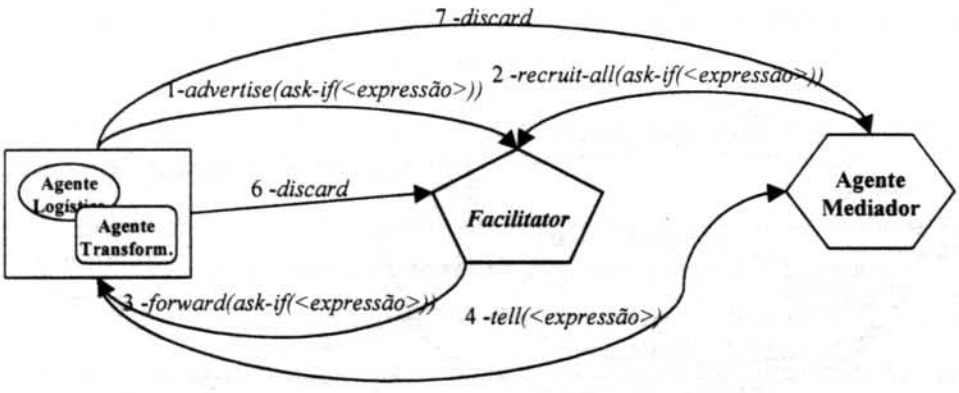


Figura 6.8 – Sequência de mensagens trocadas nas fases de criação, configuração e organização da EV (variante ao cenário 1)

Cenário 2

Neste cenário, a criação da EV é conseguida através de um mecanismo de subscrição. Numa primeira fase, o agente mediador informa os agentes facilitadores acerca do seu interesse em receber subscrições, de agentes AT e/ou AL, para mensagens relativas a um determinado assunto (definido numa *expressão*). Estas mensagens podem ser expressas através da performativa *ask-all* (nesta performativa, o emissor solicita toda a informação satisfazendo uma dada *expressão*; por exemplo, esta *expressão* pode ser o identificador relativo a propostas de produção, mostrando assim o emissor, o seu interesse em ser informado acerca de determinado tipo de propostas de produção).

Numa fase posterior, os agentes AT e/ou AL interessados em "arranjar" trabalho, solicitam aos agentes facilitadores que os informem acerca de agentes mediadores que precisem de agentes AT e/ou AL. Após esta fase, os agentes AT e/ou AL "subscrevem" directamente os seus interesses nos agentes mediadores, passando a receber destes directamente mensagens respeitantes aos interesses subscritos.

A Figura 6.9 apresenta, para este cenário, a sequência de mensagens envolvidas e respectivas performativas. Para facilitar a leitura, não se apresentam algumas mensagens adicionais tais como as relativas ao estabelecimento do fim das relações de interesses, que entretanto se possam ter estabelecido.

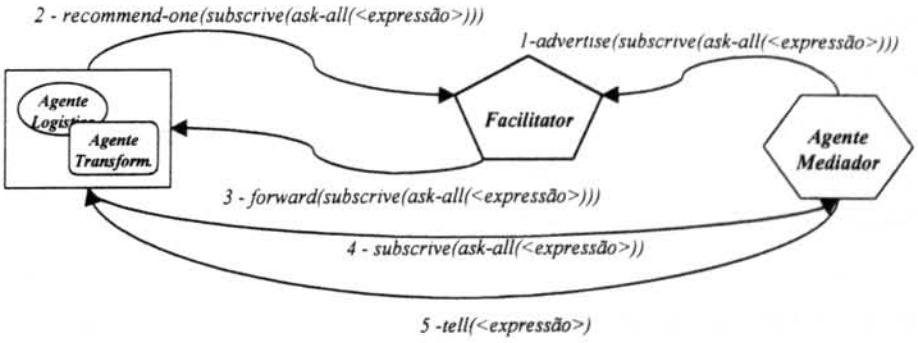


Figura 6.9 – Sequência de mensagens trocadas nas fases de criação, configuração e organização da EV (cenário 2)

Independentemente do cenário considerado para as fases associadas à criação da EV, o agente mediador é sempre responsável pela definição da rede de actividades

associada à execução de uma determinada encomenda. Basicamente, este agente anuncia aos agentes *facilitadores* as tarefas que há a realizar. Por sua vez, os agentes facilitadores, com base nas declarações de interesses registadas, encaminham as tarefas para possíveis agentes interessados na execução dessas tarefas. Os agentes AT e AL interessados respondem ao anúncio de uma ou mais tarefas, cabendo ao agente mediador a escolha, entre os candidatos, dos mais promissores. Com base nesta selecção de agentes, é constituída uma espécie de rede contratual, à qual são posteriormente submetidas, na fase de operação da EV, propostas de execução de tarefas. A atribuição de tarefas (nomeadamente o seu planeamento e afectação) é feita com base num processo de negociação entre agentes.

6.3.6 Operação da EV

A fase de operação da EV envolve essencialmente dois tipos de tarefas: planeamento de pedidos/encomendas de clientes (submetidos através dos agentes AR) e execução das encomendas ao longo da rede de produção.

Neste trabalho, considerou-se que o planeamento de pedidos/encomendas pode ser feito segundo duas estratégias de negociação distintas. Na primeira, procura-se uma solução que respeite a data de entrega pretendida (*due-date*) e que satisfaça a quantidade pedida, com um custo mínimo. Na segunda estratégia, procura-se determinar a data mais "conveniente" para satisfazer uma determinada quantidade pedida (naturalmente com custos mínimos).

Independentemente da estratégia de negociação, a metodologia de planeamento pode decorrer segundo duas abordagens diferentes: uma centralizada no agente mediador, e outra distribuída ou cooperativa.

6.3.6.1 Planeamento centralizado

A metodologia de planeamento centralizado é similar à descrita no capítulo anterior. Existe uma entidade, a um nível hierárquico superior, responsável pela coordenação das diversas entidades operacionais que constituem a EV.

Nesta abordagem, o agente mediador interactua permanentemente com os agentes AT e/ou AL, até obter uma solução de planeamento considerada satisfatória.

A coordenação é assegurada pelo agente AM e nesta fase não se verifica qualquer cooperação entre agentes AT e/ou AL.

O processo de planeamento é iterativo e a interacção entre agentes, durante o processo de negociação, envolve essencialmente duas *performativas* KQML: *ask-if* para submissão de sub-propostas do agente AM aos agentes AT e AL; e *tell* para envio de ofertas ao agente AM. A Figura 6.10 representa a sequência de mensagens que tem lugar neste processo de planeamento.

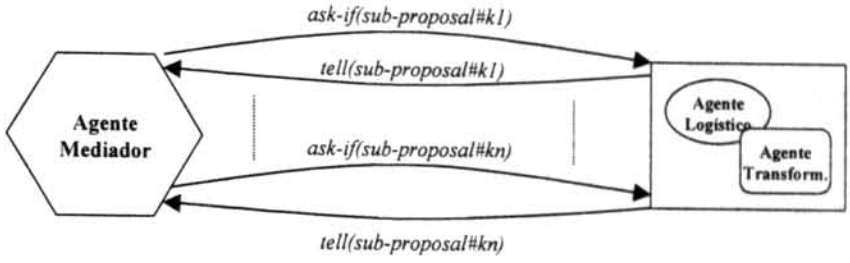


Figura 6.10 – Sequência de mensagens KQML no planeamento centralizado

6.3.6.2 Planeamento cooperativo

Para Empresas Virtuais, uma abordagem ao planeamento baseada na cooperação é, na nossa opinião, mais adequada e promissora. Contrariamente ao planeamento centralizado, em certo sentido hierarquizado e apriorístico, o planeamento cooperativo explora as características inerentes a um sistema multi-agente, nomeadamente quanto à capacidade de cooperação dos agentes na prossecução de um objectivo global, quanto à conjugação de competências diversas e quanto à adequação a alterações dinâmicas do ambiente.

Nesta metodologia de planeamento, o agente mediador anuncia à rede de agentes, entretanto estabelecida contratualmente, um conjunto de tarefas sob a forma de "propostas" que, após negociação exclusivamente entre agentes AT e AL (sem a intervenção do agente mediador), dará origem a uma ou mais "ofertas".

A interacção necessária entre os agentes AT e AL pode ser encarada como uma "negociação de contratos" entre os agentes que necessitam de ajuda na execução de

tarefas e aqueles que têm disponibilidade e competência para fornecerem tal ajuda. Esta negociação, conducente à cooperação inter-agentes, tem como vantagem principal o facto do planeamento levar em conta a situação em que o sistema de produção se encontra no momento da decisão, e não ser feito "à priori" (como no caso do planeamento centralizado). Igualmente importante é o facto de muitas das "restrições" associadas ao problema de planeamento serem resolvidas a nível local, isto é, ao nível de cada agente, reduzindo significativamente a dimensão do problema global e, consequentemente, o esforço total de cálculo.

Do ponto de vista computacional, esta metodologia é naturalmente escalável (Azevedo and Oliveira, 1999). A resolução do problema de planeamento global é, nesta abordagem, conseguida de forma distribuída e descentralizada, e resulta da integração do conhecimento, competências, objectivos e planos dos diferentes agentes. Estes participam mais autonomamente no processo de tomada de decisões relativas às suas próprias acções. No planeamento centralizado, descrito anteriormente, a resolução do problema é distribuída; contudo a coordenação da actividade dos agentes é centralizada, isto é, o planeamento das acções a desenvolver por cada um dos agentes, resulta da decomposição inicial do problema em subproblemas e da atribuição destes aos vários agentes.

Consoante a estratégia de negociação em curso, nomeadamente, se se atribui prioridade à data de entrega ou prioridade à quantidade (melhor data possível), consideram-se respectivamente dois tipos de planeamento cooperativo: um *regressivo* e outro *progressivo*.

Note-se que o processo de planeamento, independentemente de ser regressivo ou progressivo, envolve a geração inicial, pelo agente mediador, de uma proposta de execução, composta por uma lista de sub-propostas, uma por cada actividade (estágio) considerada. A informação de base de uma sub-proposta engloba a identificação do produto, a quantidade, a data de início e de fim da produção e uma prioridade associada (por exemplo, dependente da "importância" do cliente). Para além desta informação, uma sub-proposta contém ainda listas de actividades predecessoras e sucessoras (associadas a outros estágios do processo produtivo) e informação relativa à estratégia de planeamento a seguir.

Se para uma sub-proposta, associada a uma determinada actividade, a lista de predecessores se encontrar vazia e a estratégia de planeamento for progressiva, então

o ponto de partida do planeamento cooperativo tem lugar no estágio ao qual a actividade pertence. De igual forma, fica definido o ponto de partida do planeamento se a lista de sucessores de uma determinada proposta estiver vazia e a estratégia de planeamento for regressiva.

Planeamento cooperativo regressivo

O planeamento cooperativo regressivo procura garantir o cumprimento da data de entrega (*due-date*) associada ao pedido em análise.

Após o agente mediador anunciar à rede de agentes, eventualmente através dos agentes facilitadores, o conjunto de sub-propostas por si geradas, inicia-se o processo de planeamento cooperativo. Cada um dos agentes a quem as sub-propostas são enviadas, sabe quem são os agentes com os quais pode cooperar (o conteúdo das listas de predecessores e sucessores definem dinamicamente a sua vizinhança). Uma vez definido o ponto de partida do planeamento, os agentes adjacentes negoceiam propostas/ofertas de execução de tarefas, tendo em consideração restrições de âmbito externo, por exemplo de carácter logístico (restrições estas que são geridas por agentes facilitadores).

À medida que se vão estabelecendo consensos entre agentes adjacentes, são envolvidos no processo novos agentes (predecessores), e simultaneamente vai sendo definida uma rota produtiva para a execução da tarefa em planeamento. Este processo de planeamento cooperativo continua até ser atingido o início do processo, definido pelo agente mediador na rede de actividades. Atingindo-se o início do processo, são enviadas ao agente mediador as ofertas de solução global obtidas. Após a escolha de uma solução, entre as várias alternativas recebidas, o agente mediador procede à confirmação dessa solução com todos os agentes envolvidos.

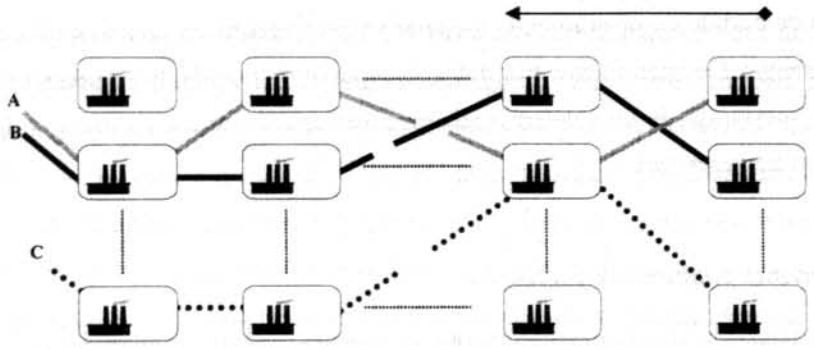


Figura 6.11 – Soluções alternativas no planeamento cooperativo regressivo

Planeamento cooperativo progressivo

Relativamente à situação descrita anteriormente, esta metodologia de planeamento difere, unicamente quanto ao sentido de progressão. O objectivo final é a obtenção da data de conclusão mais cedo, em que se satisfaz a quantidade pedida e com custo mínimo.

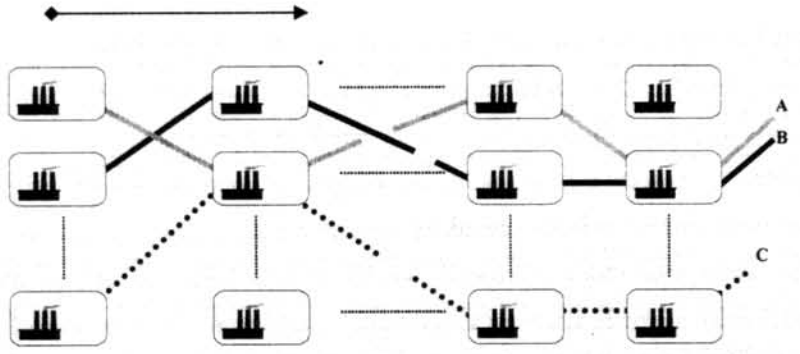


Figura 6.12 – Soluções alternativas no planeamento cooperativo progressivo

O processo de planeamento vai progredindo, constituindo-se, em simultâneo, diferentes rotas alternativas. Assim, uma unidade produtiva pode estar envolvida, durante a fase de planeamento, em várias rotas.

Apresentam-se, de seguida, as diferentes fases que compõem o processamento cooperativo progressivo (a situação de planeamento cooperativo regressivo é similar),

considerando que a EV já se encontra criada e configurada. Em cada um das fases, apresenta-se a sequência de "performativas" envolvidas.

Fase 1

O agente mediador, após definir a rede de actividades associada ao fabrico de um produto, envia em simultâneo para os agentes que compõem a EV, sub-propostas de acordo com a organização definida para a rede produtiva (por exemplo, sob a forma de uma sequência de estágios consecutivos). Nesta fase, o agente mediador (AM) recorre à performativa *tell* para informar o receptor acerca de determinada sub-proposta. A tabela seguinte apresenta a sequência de performativas verificada entre o agente mediador (AM) e cada um dos agentes AT da EV. Considera-se a seguinte notação: AT#j.i representa o agente AT número i pertencente ao estágio j.

seq.	emissor	performativa	receptor
1	AM	tell(sub-proposta#1)	AT#1.1
1	AM	tell(sub-proposta#1)	AT#1.2
1	...	...	...
1	AM	tell(sub-proposta#1)	AT#1.n
1	...	...	...
1	AM	tell(sub-proposta#u)	AT#u.1
1	AM	tell(sub-proposta#u)	AT#u.2
1	...	...	...
1	AM	tell(sub-proposta#u)	AT#u.m

Fase 2

Após a determinação do ponto de partida do planeamento, que pode envolver diversos agentes alternativos (por exemplo, pertencentes ao mesmo estágio), cada um destes agentes avalia localmente a proposta recebida e gera a correspondente oferta. De seguida, essa oferta é propagada para cada um dos agentes adjacentes, pertencentes ao estágio seguinte (isto é, verifica-se a propagação de "restrições" no sentido do pla-

neamento). Com as ofertas recebidas dos agentes predecessores e com a sub-proposta inicialmente recebida do agente mediador, cada um dos agentes vai gerar as respectivas ofertas retornando o resultado aos agentes predecessores envolvidos (processo de negociação entre agentes adjacentes).

A tabela seguinte exemplifica a sequência de performativas que tem lugar entre os agentes adjacentes de estágios consecutivos (isto é, entre cada um dos agentes de um estágio e todos os agentes do estágio seguinte). Nesta fase, as performativas envolvidas são: *ask-if* para a "propagação" das ofertas aos agentes receptores e *tell* para o retorno das respectivas respostas. Considera-se a seguinte notação: AT#s.i representa o agente AT número i pertencente ao estágio s (supõe-se que este estágio tem n agentes tipo AT) e AT#r.j representa o agente AT número j pertencente ao estágio sucessor r (supõe-se que este estágio tem w agentes tipo AT).

seq.	emissor	performativa	receptor
2	AT#s.1	ask-if(oferta#1)	AT#r.1
2	...	...	...
2	AT#s.1	ask-if (oferta#1)	AT#r.w
2	AT#s.2	ask-if(oferta #2)	AT#r.1
2	...	...	...
2	AT#s.n	ask-if (oferta #n)	AT#r.1
2	...	...	...
3	AT#r.1	tell(oferta #1.1)	AT#s.1
3	AT#r.1	tell(oferta #1.2)	AT#s.2
3	...	...	...
3	AT#r.1	tell(oferta #1.n)	AT#s.n
3	...	...	...
3	AT#r.w	tell(oferta #w.n)	AT#s.n

Fase 3

Nesta fase, os agentes avaliam as ofertas recebidas dos agentes sucessores seleccionando a mais favorável. Desta selecção, resulta também a selecção do agente, do estágio sucessor, que continuará com o processo, isto é, vai sendo definida uma rota de execução da encomenda em planeamento (a performativa envolvida nesta fase é *insert*). Note-se que este processo de selecção, dependendo do número de agentes sucessores seleccionados, vai estabelecendo, em simultâneo, diversas rotas de produção, alternativas.

Os agentes seleccionados (pertencentes ao estágio sucessor), iniciam com os agentes do estágio sucessor seguinte o processo de negociação, repetindo o procedimento descrito na fase 2. Este processo repete-se, até se atingir o estágio final.

seq.	**emissor**	**performativa**	**receptor**
4	AT#s.1	insert(solução-intermédia#1)	AT#r.j
4	...	...	...
4	AT#s.y	insert(solução-intermédia#y)	AT#r.t
5	AT#r.j	ask-if(oferta#1)	AT#r+1.1
5	...	...	...
5	AT#r.j	ask-if (oferta#1)	AT#r+1.p
5	AT#r.t	ask-if(oferta #x)	AT#r+1.1
5	...	...	...
5	AT#r.t	ask-if (oferta #x)	AT#r+1.p
5	...	...	...
6	AT#r+1.1	tell(oferta #1.1)	AT#r.j
6	...	...	...
6	AT#r+1.1	tell(oferta #1.y)	AT#r.t
6	...	...	...
6	AT#r+1.2	tell(oferta #2.1)	AT#r.j
6	...	...	...

6	AT#r+1.p	tell(oferta #p.y)	AT#r.t

Fase 4

Atingido o último estágio do planeamento, são avaliadas as melhores alternativas pelos agentes pertencentes a esse estágio. Assim, obtém-se, por cada agente seleccionado nesse estágio, uma solução para o planeamento da encomenda. As soluções obtidas são enviadas ao agente mediador que, após seleccionar uma delas segundo uma avaliação de carácter global, procede à confirmação definitiva a cada um dos agentes envolvidos na solução escolhida (nesta fase recorre-se à performativa *achieve*). Na tabela seguinte apresenta-se as sequências de performativas envolvidas nesta fase do processo, considera-se que AT#u.i representa o agente AT número i pertencente ao último estágio de planeamento.

seq.	**emissor**	**performativa**	**receptor**
7	AT#u.1	ask-if(solução-final#1)	AM
7	...	...	...
7	AT#u.g	ask-if(solução-final#g)	AM
8	MA	achieve(ordem#1)	AT# 1.x
8	...	...	...
8	MA	achieve(ordem#k)	AT#k.h
8	MA	achieve(ordem#k+1)	AT#k+1.b
8	...	...	...
8	MA	achieve(ordem#u)	AT#u.q

A Figura 6.13 ilustra as interacções principais existentes no processo de planeamento cooperativo progressivo ao longo das fases descritas e de acordo com as sequências apresentadas nas tabelas anteriores.

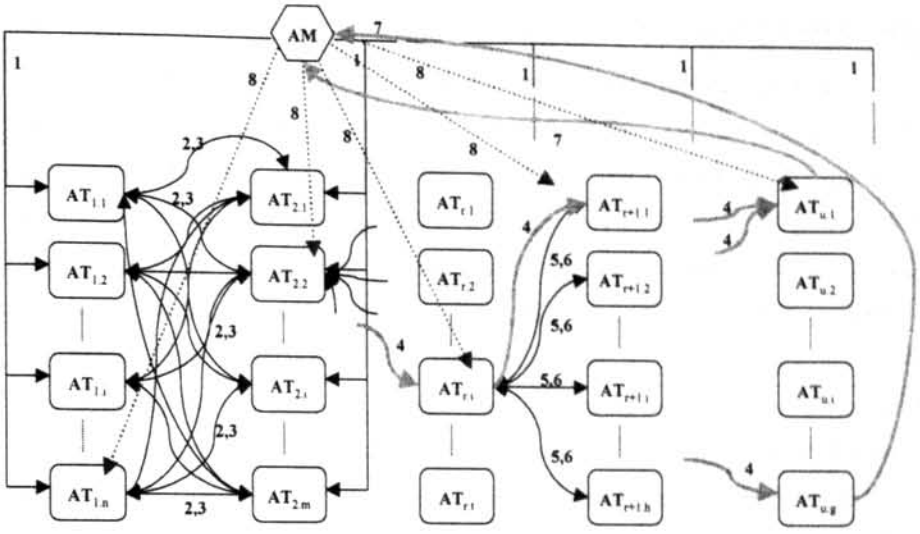


Figura 6.13 – Interacções no planeamento cooperativo progressivo

6.3.6.3 Potencialidades do planeamento cooperativo

Planeamento parcial

Para além das características já referidas anteriormente, o planeamento cooperativo permite tratar, de uma forma natural, situações de planeamento parcial, tal como as apresentadas no capítulo 5 (subsecções 5.7.1.1 a 5.7.1.3). Permite igualmente tratar situações em que a quantidade a produzir em cada estágio não é necessariamente igual, dependendo, por exemplo, dos níveis de *stock* existentes em cada unidade produtiva da rede.

Na verdade, o facto do planeamento decorrer através de um processo de negociação entre agentes adjacentes, e ir progredindo de acordo com os consensos estabelecidos, leva a que as decisões (por exemplo respeitantes a datas e quantidades) entre dois ou mais agentes se propaguem aos agentes predecessores, no caso do planeamento regressivo, ou aos agentes sucessores, no caso do planeamento progressivo.

Complexidade do processo produtivo

A metodologia de planeamento proposta suporta igualmente, de uma forma natural, situações em que o sistema produtivo apresenta configurações mais complexas do que um modelo do tipo *flow-shop* ou uma estrutura de precedência linear. Na verdade, o agente mediador é responsável pela definição da rede de actividades e consequentemente, durante a fase de operação da EV, pela geração das sub-propostas enviadas aos agentes AT e/ou AL. Como em cada sub-proposta são definidas todas as relações de adjacência(isto é, as possibilidades de relacionamento entre unidades), o processo de negociação entre agentes contempla, naturalmente, e de acordo com o definido pelo agente mediador, situações de configurações variáveis em complexidade tais como, por exemplo, situações de paralelismo entre unidades ou de simultaneidade de execução de actividades.

Empresa Virtual principal e secundária

Na abordagem multi-agente proposta, considerou-se que na EV existe um único agente mediador (assumindo o papel de interface entre a EV e o exterior)[37]. Contudo, qualquer agente AT e/ou AL pode, ele mesmo, ser um agente mediador de uma outra EV (EV secundária, eventualmente com uma constituição e organização mais simples) integrada na primeira (EV principal). Esta situação é particularmente interessante quando uma entidade recorre a uma rede de subcontratação para aumento da capacidade de produção.

A possibilidade de formação de uma ou mais EV secundárias, pressupõe que o agente ao qual está associada cada uma das EV secundárias tem capacidade de ser ele próprio um agente mediador. Naturalmente que esta extensão do conceito de *Empresa*

---

[37] Potencialmente, qualquer agente pode ser mediador, isto é, pode ter a iniciativa de formar uma EV. Ou seja, a rede de agentes pode dar origem a diferentes EV, eventualmente até em simultâneo. No entanto, em cada EV só pode existir uma gente mediador, mas qualquer agente pode participar, ou pertencer, a mais do que uma EV em simultâneo.

Virtual, pela flexibilidade que introduz na modelação de redes empresariais cada mais complexas, tem um enorme potencial, que se espera vir a explorar no futuro.

Capítulo 7

Conclusões e Desenvolvimentos Futuros

7.1 Principais resultados e conclusões

7.2 Desenvolvimentos futuros

7 CONCLUSÕES E DESENVOLVIMENTOS FUTUROS

7.1 Principais resultados e conclusões

A emergência de novas formas organizacionais, tais como a *Empresa Virtual*, colocando enormes desafios e requisitos em termos dos sistemas de informação de gestão e controlo das operações, permitiu-nos identificar um espaço interessante de intervenção, que constituiu o quadro geral em que se desenvolveu o trabalho de investigação apresentado nesta dissertação.

Numa perspectiva essencialmente de produção, e como resultado do novo contexto em que as empresas industriais operam e da cada vez maior orientação ao cliente final, existe uma forte necessidade de dispor de soluções inovadoras, ao nível das metodologias de planeamento, bem como das ferramentas informáticas que as suportam. Em particular, revela-se crítica para o decisor, a existência de instrumentos que possibilitem, de forma eficiente e em tempo útil, apoiar todo o processo de negociação e aceitação de encomendas.

Os sistemas de informação empresariais actualmente disponíveis comercialmente, não suportam adequadamente o carácter distribuído do processo produtivo, inerente às novas formas de organização, e não satisfazem os requisitos de resposta rápida e fiável (isto é, com a avaliação real da capacidade), decorrentes das necessidades de planear novas encomendas. Na base do trabalho de investigação que deu origem a esta dissertação, esteve esta constatação. Trata-se de um trabalho com uma forte componente inovadora que, pela importância e relevância do problema tratado, justificará, no futuro, uma continuada atenção e novos projectos de investigação.

Globalmente, os objectivos deste trabalho, expostos de forma sucinta no primeiro capítulo da dissertação e detalhados ao longo dos capítulos seguintes, foram, na nossa opinião, plenamente alcançados.

'Order promising' e planeamento agregado da produção

Numa primeira parte desta dissertação, procurou-se enquadrar e valorizar devidamente o processo de aceitação/negociação de encomendas em tempo-real e com avaliação de capacidades, em estruturas produtivas distribuídas.

No *capítulo 2* foram caracterizadas, de forma sucinta, as principais tendências neste domínio e apresentada, ainda que simplificadamente, a tipologia dos sistemas de produção e dos diferentes níveis de planeamento tradicionalmente considerados. Foi igualmente objecto de análise, o processo de aceitação de novas encomendas (*order promising*), realizado a partir de informação de ATP obtida a partir dos tradicionais quadros de planeamento. As limitações deste processo conduziu-nos à necessidade de propor um mecanismo que permita que esta negociação seja feita com uma avaliação real das capacidades, tendo em conta o carácter distribuído do processo produtivo. Assim, foram apresentadas as abordagens geralmente consideradas na avaliação da capacidade produtiva e descreveram-se novos modelos de capacidades, simples e com um nível de agregação elevado, que podem ser facilmente integrados num sistema de planeamento distribuído, baseado em negociação.

Empresa Virtual

A vantagem competitiva das empresas passa cada vez mais pelo desenvolvimento de novas estruturas organizacionais suportadas por sistemas de informação adequados. Ao longo do *capítulo 3,* foram apresentados e caracterizados alguns dos novos paradigmas organizacionais, com particular ênfase para a *Empresa Virtual* e os seus requisitos ao nível dos sistemas de informação. Conclui-se que a emergência da *Empresa Virtual*, como novo paradigma organizacional, impõe um conjunto de requisitos ao nível funcional e de infra-estrutura tecnológica ainda não cabalmente suportados pelas soluções disponíveis comercialmente, nomeadamente pelos sistemas de informação empresarial ERP e SCM.

Assim, identificou-se um conjunto de requisitos para as soluções informáticas, por forma a que permitam: estabelecer um quadro de cooperação entre entidades heterogéneas, independentes e autónomas; suportar a negociação de encomendas em tempo-real, através de uma rápida avaliação da capacidade produtiva (ao longo da rede de empresas); planear, de uma forma optimizada e sincronizada, as actividades das empresas; monitorizar a execução das encomendas, ao longo de toda a cadeia de for-

necimento; incorporar mecanismos inteligentes para pré-aviso da ocorrência de problemas de incumprimento de datas e que sejam suficientemente flexíveis para integrarem novos componentes ou novos "objectos de negócio" de tal modo que suportem a dinâmica de reconfiguração e reorganização da *Empresa Virtual*.

Sistema de Apoio à Decisão

A concepção e a implementação de um Sistema de Apoio à Decisão (SAD) para apoiar o processo de negociação e aceitação de encomendas em *tempo-real* e com avaliação de capacidades num ambiente eminentemente distribuído, constituíram o objectivo principal do trabalho de investigação apresentado nesta dissertação. Ao longo do *capítulo 4*, descreveu-se este sistema informático que se pode considerar inovador segundo duas perspectivas: uma conceptual e outra essencialmente tecnológica.

Ao nível conceptual, o carácter inovador reside na metodologia de planeamento desenvolvida que se baseia num processo dinâmico de negociação entre uma entidade coordenadora e diversas entidades locais (uma por cada unidade produtiva inserida na rede). Assim, a um nível superior são geradas propostas e avaliadas ofertas, considerando sempre um elevado nível de agregação de informação; a um nível inferior são avaliadas, pelos modelos locais de capacidade, as propostas geradas no nível superior. Uma das características principais da solução é a sua natural adaptação ao carácter dinâmico e simultaneamente heterogéneo da *Empresa Virtual*, nomeadamente, pela possibilidade de coexistirem diferentes modelos de capacidade (isto é, coexistirem diferentes estratégias locais de planeamento). Além disso, outra das características da solução adoptada reside na forma como foi desenvolvido o "motor" de planeamento e coordenação da rede de unidades produtivas. Existe a possibilidade, durante o funcionamento do sistema, de desactivar o algoritmo de optimização que está a ser utilizado, para activar um outro, alternativo.

Ao nível tecnológico, a solução é inovadora porque é totalmente baseada em componentes distribuídos, sobre uma infra-estrutura compatível com a especificação CORBA, e suportada por um sistema de gestão de informação de base de dados igualmente distribuído.

Modelos e algoritmos

Tendo em vista a estruturação do problema com a finalidade de se poderem vir a aplicar procedimentos de optimização, foram desenvolvidos dois modelos de programação matemática envolvendo objectivos múltiplos.

Os problemas considerados neste contexto enquadram-se na classe de problemas de Optimização Combinatória (OC) que apresentam, em geral, uma grande complexidade e enormes dificuldades de resolução. Assim, ao longo do *capítulo 5*, foram apresentadas várias abordagens heurísticas para resolução do problema de coordenação associado ao processo de negociação e que permitem o planeamento de novas encomendas. Os algoritmos desenvolvidos, que foram integrados e testados no SAD, baseiam-se em técnicas de pesquisa local e são do tipo *meta-heurísticas*. Concretamente, foram desenvolvidas duas *meta-heurísticas*, baseadas em *Simulated Annealing* e *Algoritmos Genéticos*.

No caso presente, a aplicação do *simulated annealing* obrigou a uma decomposição algorítmica em três níveis principais. O primeiro nível é, no essencial, responsável pela geração de soluções iniciais e pela estratégia de pesquisa ao longo do espaço de soluções (com base no conceito de vizinhança de uma solução). O segundo nível é responsável pela implementação do mecanismo de avaliação de uma solução. Após a decomposição da solução a avaliar, pelo número de estágios do processo produtivo, inicia-se um processo iterativo de negociação, envolvendo os modelos de capacidade local. Neste nível, procede-se à selecção ou à agregação de unidades de produção de cada estágio, para satisfazer a solução parcial correspondente. A determinação do valor da função objectivo para a solução actual é feita recorrendo ao cálculo combinado dos custos individuais de processamento, transporte, etc. O terceiro nível, constituído por diversas instâncias de modelos de capacidades, gera para cada pedido recebido a oferta correspondente (datas possíveis de realização, custo associado, etc.). Após a recepção das ofertas, e da sua combinação, procede-se ao cálculo dos custos associados.

No caso dos *algoritmos genéticos*, cada *gene* de um *cromossoma* representa uma data no horizonte de planeamento. Dois *genes* consecutivos constituem a janela temporal (data de início, data de fim) que delimita o período da produção a decorrer no estágio a que os genes estão associados.

Os resultados computacionais obtidos, embora preliminares, permitem concluir que as *meta-heurísticas* desenvolvidas são relativamente robustas. Contudo, apresentam como principal limitação o facto de assumirem que todos os estágios estão necessariamente envolvidos no processo de produção. Não são, assim, tratadas situações de planeamento parcial, isto é, envolvendo apenas alguns dos estágios.

Estas dificuldades, a par de problemas de eficiência, motivaram o desenvolvimento de uma heurística construtiva específica baseada na análise "multi-atributo" das ofertas recebidas dos modelos locais de capacidade. O conjunto de atributos considerados procura reflectir a estratégia do decisor, no processo de negociação associado ao planeamento/coordenação de um pedido (encomenda potencial). Nomeadamente, são considerados factores tais como penalização por violação de datas de conclusão, não satisfação de quantidades pedidas, minimização do WIP e do *lead-time* total, maximização da "fiabiliade" das entidades participantes na satisfação do pedido, etc.

Abordagem multi-agente

No capítulo 6, foi proposta uma arquitectura de sistema de informação para apoio ao planeamento, baseada no paradigma dos *Sistemas Multi-Agente*. A razão principal para esta proposta reside na necessidade de apoiar, na sua totalidade, o ciclo de vida da *Empresa Virtual*, em particular nas fases de criação e organização, e de se explorar novas potencialidades ao nível do planeamento, através de mecanismos baseados fortemente na cooperação entre agentes inteligentes associados a entidades "adjacentes".

Na base desta abordagem com agentes inteligentes, está a ontologia do domínio do problema (*Virtual Enterprise Ontology*), desenvolvida inicialmente com o objectivo de apoiar as fases de especificação e análise de um sistema de planeamento e controlo da produção para a *Empresa Virtual*.

A arquitectura proposta, inspirada nas arquitecturas subjacentes ao uso da linguagem e protocolo de comunicação inter-agentes KQML (adoptada aqui como linguagem de comunicação entre agentes), considera dois tipos de agentes inteligentes: *agentes organizacionais* e *agentes funcionais*. No grupo dos agentes funcionais, foram definidos os seguintes agentes específicos: agente receptor, agente empresa, agente logístico e agente mediador.

A *Empresa Virtual* é modelada directamente por diversos agentes funcionais, de diferentes tipos ou funções, e por um ou mais agentes organizacionais para apoio à infra-estrutura de suporte à comunidade de agentes funcionais.

7.2 Desenvolvimentos futuros

Dada a natureza dos problemas analisados nesta dissertação, há naturalmente ainda muito trabalho a realizar no sentido de apoiar cabalmente os processos de planeamento da produção em empresas distribuídas, organizadas em rede ou conceptualizadas como "organizações virtuais". Por exemplo, os desenvolvimentos conceptuais e algorítmicos apresentados ao longo da dissertação, necessitam de ser validados em cenários distintos daqueles que foram considerados.

Assim, antes de se proceder ao desenvolvimento de novas funcionalidades, entende-se ser prioritário promover o desenvolvimento de um quadro conceptual de referência que suporte, de uma forma mais abrangente e genérica, o paradigma organizacional da *Empresa Virtual*, numa perspectiva quer vertical quer horizontal.

A *Virtual Enterprise Ontology* foi aqui assumida como uma primeira proposta de núcleo conceptual para o estabelecimento de uma plataforma ontológica de suporte a novos paradigmas organizacionais, como o referido. Assim, é necessária não só a sua extensão e refinamento como fundamentalmente a sua formalização.

Os aspectos de configuração e reconfiguração associados ao ciclo de vida das organizações virtuais serão seguramente, pela actual falta de soluções adequadas e pela importância que representam na dinâmica do mercado, áreas de grande investigação e desenvolvimento futuros. Igualmente fundamental será, a integração explícita dos aspectos humanos e sócio-organizacionais nos complexos processos de tomada de decisões.

No limite, pode antever-se um núcleo funcional que, conjuntamente com uma infra-estrutura de comunicação global e mecanismos de configuração funcional e informacional, permita "criar" de uma forma o mais transparente e automática possível, mais um nó numa complexa rede de entidades heterogéneas, com especificidades, competências e interesses diversos. O paradigma dos agentes inteligentes e os con-

ceitos que lhe estão associados parecem, neste contexto, ter um grande potencial no sentido de se conseguir sistemas cada vez mais auto-configuráveis e auto-adaptativos.

Finalmente, refira-se que, na gestão de operações da *Empresa Virtual*, há ainda muito a fazer quer ao nível das metodologias de planeamento, quer ao nível de novas formas de avaliação de capacidades. Em particular, afigura-se como muito promissor, ao nível do planeamento, o recurso a técnicas e estratégias baseadas na colaboração e cooperação das diferentes entidades envolvidas.

Referências

APICS (1987). "Production and Inventory Control Handbook." , J. H. Greene, ed., McGraw-Hill Book Company.

APICS (1995). APICS Dictionary, APICS.

Artiba, A. and Elmaghraby, S.E. (1997). "The Planning and Scheduling of Production Systems – Methodologies and Applications". Chapman & Hall.

Atherton, L. F., and Atherton, R. W. (1995). Wafer Fabrication: Factory Performance and Analysis, Kluwer Academic Publishers.

Austin, J. L. (1962). How To Do things With Words, Clarendon Press.

Azevedo, A. L., and Oliveira, R. T. (1999). "A Scalable Approach for Order Negotiation and Planning." The 4th International Symposium on Logistics (ISL'99) Logistics In The Information Age, Florence - Italy.

Azevedo, A. L., and Sousa, J. P. (1999). "A Component-based Approach to Support Order Planning In a Distributed Manufacturing Enterprise." CAPE'99 - The 15th International Conference on Computer-Aided Production Engineering, University of Durham – UK.

Azevedo, A. L., and Toscano, C. (1999). "An Information System for Distributed Manufacturing Enterprises." CEIS 99 - First International Conference On Enterprise Information Systems, Setúbal, Portugal, 15-22.

Azevedo, A., and Sousa, J. P. (1997). "On the Design of an Order Promise System for Virtual Enterprises." MCPL'97- Conference on Management and Control of Production and Logistics, Campinas -SP- Brazil, 71-76.

Back, T. (1992). "Optimization by Means of Genetic Algorithms." Department of Computer Science - University of Dortmund, Dortmund - Germany.

Back, T., and Hoffmeister, F. (1991). "Extended Selection Mechanisms in Genetic Algorithms." Fourth International Conference on Genetic Algorithms and their Applications, University of California, San Diego, USA, 92-99.

Back, T., Hoffmeister, F., and Schwefel, H.-P. (1993). "Applications of Evolutionary Algorithms." SYS-2/92, Department of Computer Science - University of Dortmund, Dortmund - Germany.

Baker, J. E. (1985). "Adaptive selection methods for genetic algorithms." First International Conference on Genetic Algorithms and their Applications, Hillsdale, New Jersey, 101-111.

Baudin, M. (1990). Manufacturing Systems Analysis With Application to Production Scheduling, Yourdon Press.

Bhatnagar, R., Chandra, P., and Goyal, S. K. (1993). "Models for multi-plant coordination." European Journal of Operations Research (67), 141-160.

Billings, M., and Sturm, J. (1996). "Requirements Specification for X-CITTIC Phase I." Deliverable 1.4 - Esprit Project 20544, TEMIC.

Bitran, G. R., and Hax, A. C. (1977). "On the design of hierarchical production planning systems." Decision Sciences(8), 28-55.

Blazewicz, J., Ecker, K. H., Schmidt, G., and Weglarz, J. (1994). Scheduling in Computer and Manufacturing Systems, Springer-Verlag.

Booch, G., Rumbaugh, J., and Jacobson, I. (1999). The Unified Modeling Language User Guide, Addison Wesley.

Bradshaw, J. M., Boose, J. H., and Shema, D. B. (1992). " Steps Toward Sharable Ontologies for Design Rationale." AAAI-92 Design Rationale Capture and Use Workshop, San Jose, CA.

Brando, T. J. (1995). "Comparing DCE and CORBA." MP 95B-93, The MITRE Corporation.

Brás, H. C. M. (1996). "Plataformas de Suporte ao Desenvolvimento de Aplicações de Gestão Fabril Distribuídas" Tese de Mestrado, Universidade do Porto, Porto.

Buffa, E. S., and Sarin, R. K. (1987). Modern Production / Operations Management, John Wiley & Sons.

Bukh, N. D. (1994). "A bibliography of hierarchical production planning." Technical Report, Institute of Management - University of Aarhus.

Burghart, T. (1998). "Distributed Computing Overview." , QUOIN.

Camarinha-Matos, L. M., Afsarmanesh, H., Garita, C., and Lima, C. (1997a). "Towards an Architecture for Virtual Enterprises." 2nd World Congress on Intelligent Manufacturing Processes & Systems, Budapest.

Camarinha-Matos, L.M., and Afsarmanesh, H. (1998). "Virtual Enterprises:Life cycle supporting tools and technologies." Handbook of Life Cycle Engineering: Concepts, Tools and Techniques, A. M. Gutiérrez, J. M. Sánchez, and A. Kusiak, eds., Chapmam & Hall.

Camarinha-Matos, L.M., Lima, C.p., and Osorio, L. (1997b). "The PRODNET platform for production planning and management in virtual enterprises." ICE'97 - International Conference On Concurrent Enterprising, Nottingham - UK.

Carravilla, M.A.S.L. (1996). "Modelos e Algoritmos para o Planeamento Hierárquico da Produção - Aplicações a um Caso de Estudo". Tese de Doutoramento, Universidade do Porto, Porto.

Chase, R.B., and Aquilano, N.J. (1995). Gestão da Produção e das Operações: Perspectiva do Ciclo de Vida, Monitor.

Ching, C., Holsapple, C.W., and Whinston, A. B. (1993). "Modeling Network Organisations: A Basis for Exploring Support Coordination Possibilities." Journal of Organizational Computing, 3(3), 279-300.

Christopher, M., Harrison, A., and Hoek, R. v. (1999). "Creating The Agile Supply Chain: Issues and Challenges." Logistics In The Information Age, Florence, Italy.

Davidow, W. H., and Malone, M. S. (1992). The Virtual Corporation: Structuring and Revitalizing The Corporation for 21st Century, Harper Business, New York.

Davis, L. (1987). Genetic Algorithms and Simulated Annealing, Pitman.

Davis, L. (1991). Handbook of Genetic Algorithms, Van Nostrand Reinhold.

Dias, M. T. G. (1995). "Aplicação de Algoritmos Genéticos ao Problema da Geração de Serviços de Tripulações". Tese de Mestrado, Universidade do Porto, Porto.

Drexl, A., and Kimms, A. (1997). "Lot sizing and scheduling - Survey and extensions." European Journal of Operational Research(99), 221-235.

Dullin, E. (1998). "Powering the Supply-Chain with Enterprise and Plantic-Centric Scheduling." APICS - The Performance Advantage.

Elmaghraby, S. E. (1991). "Manufacturing Capacity and Its Measurement: A critical Evaluation." Computers Operations Research, 18(7), 615-627.

Eversheim, W., Klocke, F., Pfeifer, T., and Weck, M. (1997). Manufacturing Excellence in Global Markets, Chapman & Hall.

Finin, T., Labrou, Y., and Mayfield, J. (1997). "KQML as an agent communication language." Software Agents, AAAI Press/The MIT Press, 291-316.

Fischer, K. (1998). "An Agent-based Approach to Holonic Manufacturing Systems." BASYS'98 - 3rd IEEE/IFIP International Conference on Information Technology for Balanced Automation Systems, Prague, Czech Republic.

Fisher, M. L. (1997). "What Is the Right Supply Chain for Your Product?" Harvard Business Review, 105-116.

Fox, M. S., Chionglo, J. F., and Barbuceanu, M. (1993). "The Integrated Supply Chain Management System." , Department of Industrial Engineering, University of Toronto.

Fox, M., and Gruninger, M. (1994). "Ontologies for Enterprise Integration." Proc. 2th Conferenve On Cooperative Information Systems, Toronto.

Frederix, F. (1996). "Definition of the Problem." WP1-Task 1.1, Alcatel-Mietec.

Frederix, F. (1996). "Planning and Scheduling Multi-Site Semiconductor Production Chains: A survey of needs, current pratices and integration issues." IT and Manufacturing Partnerships, 107-116.

Gamma, E., Helm, R., Johnson, R., and Vlissides, J. (1995). Design Patterns - Elements of Reusable Object-Oriented Software, Addison-Wesley.

Garey, M. R., and Johnson, D. S. (1979). Computers and Intractability: A Guide to the Theory of NP-Completeness, , San Francisco.

Genesereth, M. R., and Fikes, R. E. (1992). "Knowledge Interchange Format Version 3.0 Reference Manual." Report Logic -92-1, Computer Science Department - Stanford University, Stanford, California.

Gindy, N. N. Z. (1999). "Responsive manufacturing in UK Aerospace Industry." CAPE´99 - The 15th International Conference on Computer-Aided Production Engineering, University of Durham, 557-565.

Goldberg, D. E. (1989). Genetic Algorithms in Search, Optimization, and machine Learning, Addison-Wesley.

Goldberg, D. E. (1991). "Real-coded Genetic Algorithms, Virtual Alphabets, and Blocking." Complex Systems(5), 139-167.

Gordon, V. S., and Whitley, D. (1993). "Serial and Parallel Genetic Algorithms as Function Optimizers." ICGA-93: The 5th International Conference on Genetic Algorithms, Urbana-Champaign, 177-183.

Grasso, V., and Fichera, S. (1993). "Scheduling and Work Flow Controlling in Flexible Flow Lines." Third International FAIM Conference - Flexible Automation and Integrated Manufacturing, University of Limerick - Ireland, 879-890.

Gruber, T. (1994). "Toward Principles for the Design of Ontologies Used for Knowledge Sharing." International Journal of Human Computer Studies(special issue on Formal Ontology, Conceptual Analysis and Knowledge Representation, Guarino and Poli Eds.).

Gruber, T. R. (1993). "A Translation Approach to Portable Ontology Specifications." Knowledge Acquisition(2), 199-220.

Guarino, N., and Giaretta, P. (1995). Ontologies and Knowledge Bases: Towards a Terminological Clarification, Towards Very Large Knowledge Bases, IOS Press.

Guimarães, R. C., and Cabral, J. A. S. (1997). Estatística, McGraw-Hill.

Hax, A. C., and Meal, H. C. (1975). "Hierarchical integration of production planning and scheduling." Logistics, 1, 53-69.

Hess, U. (1998). "The Care and Feeding of Real-Time Advanced Planning and Scheduling." APICS - The Performance Advantage.

Higgins, P., Roy, P. L., and Tierney, L. (1996). Manufacturing Planning and Control - Beyond MRP II, Chapman & Hall.

Holland, J. H. (1975). Adaptation in Natural and Artificial Systems, University of Michigan Press, Ann Arbor, Michigan.

Huhns, M., and Singh, M. P. (1998). Readings in Agents, Morgan Kaufman.

Hung, Y.-F., and Leachman, R. C. (1996). "A Production Planning Methodology for Semiconductor Manufacturing Based on Iterative Simulation and Linear Programming Calculations." IEEE Transactions On Semiconductor Manufacturing, 9(2).

Hwang, C.-L., and Yoon, K. (1981). Multiple Attribute Decision Making Methods and Applications - A State-of-the-Art Survey, Springer-Verlag.

Jacobson, I., Booch, G., and Rumbaugh, J. (1999). The Unified Software Development Process, Addison-Wesley.

Jagdev, H. S., and Browne, J. (1998). "The extended enterprise - a context for manufacturing." Production planning & Control, 9(3), 216-229.

Jong, D. (1975). "An Analysis of the Behavior of a Class of Genetic Adpative Systems," Ph.D Thesis, University of Michigan, Ann Arbor.

Katzy, B. (1998). "The Virtual Enterprise." Handbook of Life Cycle Engineering: Concepts, Tools and Techniques, A. M. Gutiérrez, J. M. Sánchez, and A. Kusiak, eds., Chapmam & Hall.

Kirkpatrick, S., Gelatt, C. D., and Vecchi, M. P. (1983). "Optimization by Simulated Annealing." Science(220), 671-680.

Koza, J. R. (1989). "Hierarchical genetic algorithms operating on populations of computer programs." Eleventh international joint conference on artificial intelligence, 768-774.

Laarhoven, P. J. M. v., and Aarts, E. H. L. (1987). Simulated Annealing: Theory and Applications, Kluwer Academic Publishers, Amsterdam.

Labidi, S., and Lejouad, W. (1993). "De l'intelligence artificielle distribuée aux systèmes multi-agents." N°2004, INRIA, Sophia Antipolis.

Labrou, Y., and Finin, T. (1997). "A proposal for a new KQML specification." , Computer Science and Electrical Engineering Department (CSEE) University of Maryland Baltimore County (UMBC).

Leach, N., and Ristic, M. (1997). "Capacity Models for Scheduling The Virtual Enterprise." MCPL'97- Conference on Management and Control of Production and Logistics, Campinas -SP- Brazil, 313-318.

Leachman, R. C. (1993). "Modeling Techniques for Automated Production Planning in the Semiconductor Industry." Optimization in Industry.

Lutton, E., and Martinez, P. (1993). "A Genetic Algorithm for the Detection of 2D Geometric Primitives in Images." 2110, INRIA - Institut National De Recherche En Informatique Et En Automatique.

Lybrand, C., and Logiplan. (1998). "Standard Software for Industry & Commerce." , Coopers & Lybrand - Logiplan.

Mahfoud, S. W. (1995a). "Niching Methods for Genetic Algorithms," PhD Thesis, University of Illinois at Urbana-Champaign, Urbana.

Mahfoud, S. W. (1995b). "Population Size and Genetic Drift in Fitness Sharing." Foundations of genetic algorithms, L. D. Whitley and M. D. Vose, eds., Morgan Kaufmann, San Francisco, 185-223.

Marik, V., Pechoucek, M., and Roche, C. (1998). "PVS 98 Agent Models and their Application in Production Planning." BASYS'98 - 3rd IEEE/IFIP International Conference on Information Technology for Balanced Automation Systems.

Marsh, K., and Moulding, R. (1997). "Enterprise Resource Planning - Myth or Magic?" , JBA.

Maskell, B. H. (1994). Software and the Agile Manufacturer:computer systems and world class manufacturing, Productivity Press.

Mauth, R. (1998). "Mission-Critical Components." Byte.

McMullen, P., and Strong, R. (1999). "Determination of lockbox collection points via simulated annealing." Journal of the Operational Research Society(50), 44-51.

Melício, F., Caldeira, P., and Rosa, A. (1999). "Solving the Timetabling Problem With Simulated Annealing." ICEIS 99 - First International Conference On Enterprise Information Systems, Setúbal, Portugal, 272-284.

Metropolis, N., Rosenbluth, A., Rosenbluth, M., Teller, A., and Teller, E. (1953). "Equation of state calculations by fast computing machines." Journal of Chemical Physics(21), 1087-1092.

Mota, F. A. C. e. S. (1996). "Ambiente de Desenvolvimento de Sistemas Multiagente para Controlo e Supervisão de Processos em Aplicações Distribuídas". Tese de Doutoramento, Universidade do Porto, Porto.

Muhlenbein, H., and Schlierkamp-Voosen, D. (1995). "Analysis of Selection, Mutation and Recombination in Genetic Algorithms." Evolution as a Computational Process, Lecture Notes in Computer Science, W. Banzhaf and F. H. Eeckman, eds., Springer, Berlin, 188-214.

Muller-Merbach, H. (1981). "Heuristics and their design: a survey." European Journal of Operational Research(8), 1-23.

Nagel, R., and Dove, R. (1991). "21st Century Manufacturing Enterprise Strategy." , Iacocca Institute, Lehigh University.

Nagi, R. (1991). "Design and Operation of Hierarchical Production Management Systems," Ph.D Thesis, University of Maryland.

Naylor, B., Mason-Jones, R., Towill, D. R., and Childerhouse, P. (1999). "Integrating Lean Manufacture And Agile Supply." Logistics In The Information Age, Florence, Italy, 185-190.

Nemhauser, G. L., and Wolsey, L. A. (1988). Integer and Combinatorial Optimization, John Wiley & Sons, Inc.

NIIIP (1995). "The NIIIP Reference Architecture: Concepts and Guidelines." , http://www.niiip.org.

Nowicki, E., and Smutnicki, C. (1998). "The flow shop with parallel machines: A tabu search approach." European Journal of Operational Research(106), 226-253.

Oliveira, E. (1999). "Applications of Intelligent Agent-Based Systems." 4° SBAI - Simpósio Brasileiro de Automação Inteligente, São Paulo, Brasil.

Oliveira, E., Fischer, K., and Stepankova, O. (1999). "Multi-agent systems: which research for which applications." Robotics and Autonomous Systems(27), 91-106.

Oliveira, J. F. C. (1995). "Problemas de Posicionamento de Figuras Irregulares - Uma perspectiva de Optimização," Tese de Doutoramento, Universidade do Porto, Porto.

Oliveira, R. F. T (1998). "Gestion des Réseaux avec Connaissance des Besoins: Utilisation des Agents Logiciel," Tese de Doutoramento, Ecole Nationale Supérieure des Télécommunications, Paris

Orfali, R., Harkey, D., and Edwards, J. (1996a). The Essential Client/Server Survival Guide, John Wiley & Sons, Inc.

Orfali, R., Harkey, D., and Edwards, J. (1996b). The Essential Distributed Objects Survival Guide, John Wiley & Sons, Inc.

Orfali, R., Harkey, D., and Edwards, J. (1997). Instant CORBA, John Wiley & Sons, Inc.

Otte, R., Patrick, P., and Roy, M. (1996). Understanding CORBA (The Common Object Request Broker Architecture), Prentice-Hall.

Ovacik, I. M., and Uzsoy, R. (1997). Decomposition Methods for Complex Factory Scheduling Problems, Kluwer Academic Publishers.

Park, K., and Carter, B. (1994). "On the effectiveness of genetic search in combinatorial optimization." BU-CS-94-010, Computer Science Department, Boston University, Boston.

Parunak, H. V. D. (1998). "Practical and Industrial Applications of Agent-Based Systems." Industrial Technology Institute.

PeopleSoft. (1998). "A New Approach to ERP for High Technology Business.", PeopleSoft, Inc.

Pillep, R., Phillipson, C., and Stick, V. (1999). "Supply Chain Software - Market Overview, Benefits, Innovative Solutions For Small And Medium Enterprises." CAPE'99 - 15th International Conference On Computer-Aided Production Engineering, Durham - UK.

Pirlot, M. (1994). "General Local Search Heuristhics in Combinatorial Optimisation: a tutorial." Belgian Journal of Operations Research, Statistics and Computer Science, 32(1-2).

Pontrandolfo, P., and Okogbaa, O. G. (1999). "Global manufacturing: a review and a framework for planning in a global corporation." International Journal of Production Research, 37(1), 1-19.

Pontrandolfo, P., and Okogbaa, O. G. (1999). "Global manufacturing: a review and a framework for planning in a global corporation." International Journal of Production Research, 37(1), 1-19.

Ranito, J. V. T. M. (1993). "Algoritmos genéticos na optimização de perguntas relacionais". Tese de mestrado, Universidade do Porto, Porto.

Rayward-Smith, V. J. (1995). "Applications of Modern Heuristic Methods." , Alfred Waller Limited in association with UNICOM.

Reeves, C. R. (1995). "Modern Heuristic Techniques for Combinatorial Problems." , A. T. i. C. Science, ed., McGrawHill.

Rodrigues, R. F. (1998). Organizações, Mudança e Capacidade de Gestão, Principia, Publicações Universitárias e Científicas, Cascais.

Roldão, V. S. (1995). Planeamento e Programação da Produção, Monitor.

Rolstadas, A. (1998). "Editorial - Virtual and extended enterprise definition." Production planning & Control, 9(3), 215.

Roux, W., Dauzèrre-Pèrés, S., and Lasserre, J. B. (1999). "Planning and scheduling in a multi-site environment." Production Planning & Control, 10(1), 19-28.

Rumbaugh, J., Blaha, M., Premerlani, W., Eddy, F., and Lorensen, W. (1991). Object-Oriented Modeling and Design, Prentice-Hall International, Inc.

Russel, S., and Norvig, P. (1995). Artificial Intelligence - A Modern Approach, Prentice Hall.

Sadeh, N. M., Nakakuki, Y., and Thangiah, S. R. (1995). "Learning To Recognize (Un)Promising Simulated Annealing Runs: Efficient Search Procedures For Job Shop Scheduling And Vehicle Routing." Metaheuristics Internacional Conference.

Searle, J. R. (1969). Speech Acts, Cambridge University Press, Cambridge, UK.

Sematech. (1996). "CIM Application Framework Specification 1.3." , SEMATECH, Austin – Texas, USA.

Sharp, J. A., and Beach, R. (1998). "Towards Globalisation and the Virtual Enterprise." PROLAMAT'98 - IFIP WG5.2/5.3 International Conference, Trento - Italy.

Siegel, J. (1996). CORBA Fundamentals and Programming, John Wiley & Sons, Inc.

Simpson, N. C., and Erenguc, S. S. (1996). "Multiple-stage production planning research: history and opportunities." International Journal of Operations & Production Management, 16(6), 25-40.

Singh, M. P. (1998). "Agent Communication Languages: Rethinking the Principles." IEEE Computer, 40-47.

Smith, S. F., and Becker, M. A. (1996). "An Ontology for Constructing Scheduling Systems." , The Robotics Institute - Carnegie Mellon University.

Soares, A. L., Sousa, J. P., Azevedo, A. L., and Bastos, J. A. (1996). "Virtual Enterprise Ontology - Version 1.0.", INESC-UESP, Porto.

Soares, A. L., Sousa, J. P., Azevedo, A. L., and Bastos, J. A. (1997). "Using an Informal Ontology in the Development of a Planning and Control System - The Case of the Virtual Enterprise." ISIP'97 International Conference on Integrated And Sustainable Industrial Production, Lisbon - Portugal.

Soares, A. M. L. (1998). "Modelação para Optimização Técnico-Organizacional de Sistemas Integrados de Fabrico," Tese de Doutoramento, Faculdade de Engenharia, Universidade do Porto, Porto.

Souilah, A. (1993). "Simulated annealing for manufacturing systems layout design." 1909, INRIA.

Spence, A. M., and Welter, D. J. (1987). "Capacity Planning of a Photolithography Work Cell in a Wafer Manufacuturing Line." IEEE International Conference On Robotics And Automation, Raleigh, North Carolina - USA.

Szyperski, C. (1997). Component Software, Addison Wesley.

Taha, H. A. (1992). Operations Research - An Introduction, Maxwell Macmillan.

Tallman, O., and kain, J. B. (1998). "COM versus CORBA: A Decision Framework.", QUOIN Inc., Cambridge - Massachusetts.

Tharumarajah, A., Wells, A. J., and Nemes, L. (1996). "Comparison of the bionic, fractal and holonic manufacturing system concpts." International Journal of Computer Integrated Manufacturing, 9(3), 217-226.

Thomas, D. J., and Griffin, P. M. (1996). "Coordinated supply chain management." European Journal of Operational Research(94), 1-15.

Uschold, M., and Gruninger, M. (1996). "Ontologies: Principles, Methods and Applications." Knowledge Engineering Review, 2(11).

Uschold, M., King, M., Moralee, S., and Zorgios, Y. (1996). "The Enterprise Ontology (Version 1.1)." Technical Report, AIAI, University of Edinburgh.

Uzsoy, R., Lee, C.-Y., and Martin-Vega, L. A. (1992). "A review of Production Planning and Scheduling Models in the Semiconductor Industry Part I." IIE Transactions, 24(4), 47-60.

Viana, A. M. (1997). "Aplicação de Meta-heurísticas Multiobjectivo ao Problema de Sequenciamento de Actividades com Restrições de Recursos". Tese de Mestrado, Faculdade de Engenharia, Universidade do Porto, Porto.

Vilarinho, P. M. (1997). "Concepção e Desenvolvimento de um Sistema de Apoio ao Projecto de Implantações Fabris". Tese de doutoramento, Faculdade de Engenharia, Universidade do Porto, Porto.

Vollmann, T. E., Berry, W. L., and Whybark, D. C. (1997). Manufacturing Planning and Control Systems, Irwin / McGraw-Hill.

Wallnau, K., Weiderman, N., and Northrop, L. (1997). "Distributed Object Technology With CORBA and JAVA: Key Concepts and Implications." CMU/SEI-97-TR-004, Software Engineering Institute - Carnegie Mellon University, Pittsburgh – Pennsylvania, USA.

Warnecke, H. J. (1993). The Fractal Company, Springer-Verlag.

Westkamper, E. (1998). "Manufacturing in Networks - Competitive Advantages for Virtual Enterprises." PROLAMAT'98 - IFIP WG5.2/5.3 International Conference, Trento - Italy.

Westkamper, E., and Schmidt, T. (1997). "Concept of a Learning Simulation System." IFAC/IFIP Conference On Management And Control Of Production and Logistics, Campinas, SP, Brasil.

Womack, J. P., and Jones, D. T. (1994). "From Lean Production to The Lean Enterprise." Harvard Business Review, 93-103.

Womack, J. P., Jones, D. T., and Roos, D. (1990). The Machine That Changed the World, Rawson Associates, New York.

Wooldridge, M., and Jennings, N. R. (1995). "Intelligent Agents. Theory and Practice." submitted to Knowledge Engineering Review.

Wortmann, J. C., Euwe, M. J., Taal, M., and Wiers, V. C. S. (1996). "A review of capacity planning techniques within standard software packages." Production Planning & Control, 7(2), 117-128.

Wortmann, J. C., Muntslag, D. R., and Timmermans, P. J. M. (1997). "Customer-driven Manufacturing." , Chapman & Hall, 464.

X-CITTIC. (1997). "Keeping abreast with the State-of-the-Art." Deliverable 5.2.1-X-CITTIC E20544, INESC.

Zimmermann, F.O. (1988). "Structural and Managerial Aspects of Virtual Enterprises", http://www.teco.uni-karlsruhe.de/IT-VISION/vu-e-teco.htm.

Zweben, M. and Fox, M.S. (1994). "Intelligent Scheduling", Morgan Kaufmann Publishers.

Apêndice **A**

Problema Teste

A.1 Estrutura produtiva

A.2 Parametrização dos modelos de capacidade

A.3 Informação logística

A.4 Exemplos de respostas

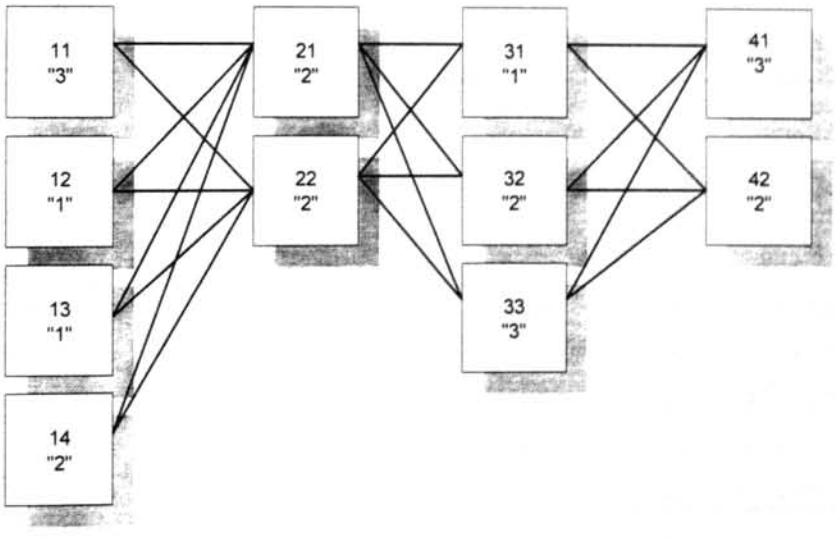
A PROBLEMA TESTE

Neste apêndice, apresenta-se uma instância do problema teste usado no capítulo 5, com a informação respeitante à estrutura produtiva, nomeadamente, número de estágios envolvidos, número de unidades produtivas e distribuição das unidades pelos estágios. Naturalmente que a instância apresentada está descrita parcialmente, já que não se apresenta a parte referente aos pedidos a planear (ver subsecção 5.5.7.1).

Cada unidade produtiva é modelada por um de três tipos de modelo de capacidades, de acordo com o que foi apresentado no capítulo 2. Além disso, a instância compreende igualmente a informação logística, nomeadamente, tempo de transporte, custo de transporte e custo de armazenamento entre unidades adjacentes.

A.1 Estrutura produtiva

A estrutura produtiva compreende um total de onze unidades de produção, distribuídas por quatro estágios, de acordo com a disposição apresentada na figura seguinte.



Na figura, em cada quadrado, que representa uma unidade de produção, encontra-se inscrito o identificador da unidade (por exemplo, 32 representa, no estágio

3, a segunda unidade de produção) e o tipo de modelo de capacidade considerado (por exemplo, "2" significa um modelo do tipo 2).

A.2 Parametrização dos modelos de capacidade

No quadro seguinte, para cada modelo de capacidade são apresentados os parâmetros base de configuração, de acordo com a descrição apresentada no capítulo 2.

estágio	ID	tipo	Yi	Yf	T	K1	K2	K3	K4	K5	K6	mLT
1	11	3	30	0	365	1.6	0.8	0.05	0.01	0.3	0.7	1
1	12	1	60	20	5	0.8	0.4	0.1e-3	2.2e-4	0.05	2	1
1	13	1	80	20	7	1.0	0.3	0.1e-3	2.0e-4	0.03	5	1
1	14	2	0	0	365	1.0	1.0	0.01	0.1	0.2	0	1
2	21	2	40	30	365	1.0	0.4	0.12	0.1	0.5	0.5	1
2	22	2	60	0	365	1.0	0.4	0.2	0.1	0.3	0.5	3
3	31	1	100	20	10	1.0	0.4	0.1e-3	2.2e-4	0.05	5	1
3	32	2	0	0	365	1.0	5.0	0.1	0.1	0.2	0	1
3	33	3	40	0	365	1.3	1.0	0.01	0.1	0.2	0.5	1
4	41	3	0	0	365	1.2	1.0	0.01	0.1	0.2	0.5	1
4	42	2	30	0	365	1.3	0.5	0.05	0.07	0.2	0	3

A.2.1 Informação adicional (perfil de capacidade e/ou ATP)

Modelo ID=11

Dia	Cap./dia	ATP
1 a 9	12	40
10 a 19	34	80
20 a 49	12	80
50 a 69	0	100
70 a 99	50	110
100 a 249	50	150
260 a 299	0	150
300 a 365	80	230

Modelo ID=12, ID=13 e ID=31

Dia	Cap.
1	56
2	70
3	40
4	10
5	56
6	70
7	25
8	49
9	67
10	74

Modelo ID=14

Dia	Cap.
1 a 19	20
20 a 31	40
32 a 89	0
90 a 365	20

Modelo ID=21

Dia	Cap.
1 a 29	20
30 a 79	30
80 a 89	50
90 a 119	0
120 a 179	40
180 a 365	30

Modelo ID=22

Dia	Cap.
1 a 29	20
30 a 79	40
80 a 89	100
90 a 119	0
120 a 179	40
180 a 365	30

Modelo ID=32

Dia	Cap.
1 a 19	30
20 a 49	15
50 a 79	35
80 a 94	0
95 a 365	20

Modelo ID=33

Dia	Cap.	ATP
1 a 9	12	10
10 a 19	34	10
20 a 49	12	20
50 a 89	0	40
90 a 102	50	100
103 a 195	0	150
196 a 204	0	150
205 a 299	15	200
300 a 365	80	230

Modelo ID=41

Dia	Cap.	ATP
1 a 9	12	10
10 a 19	34	10
20 a 49	12	10
50 a 89	0	10
90 a 102	50	100
103 a 195	0	150
196 a 204	0	150
205 a 299	15	200
300 a 365	80	230

Modelo ID=42

Dia	Cap.
1 a 29	20
30 a 49	60
50 a 79	45
80 a 89	0
90 a 179	60
180 a 189	0
190 a 365	40

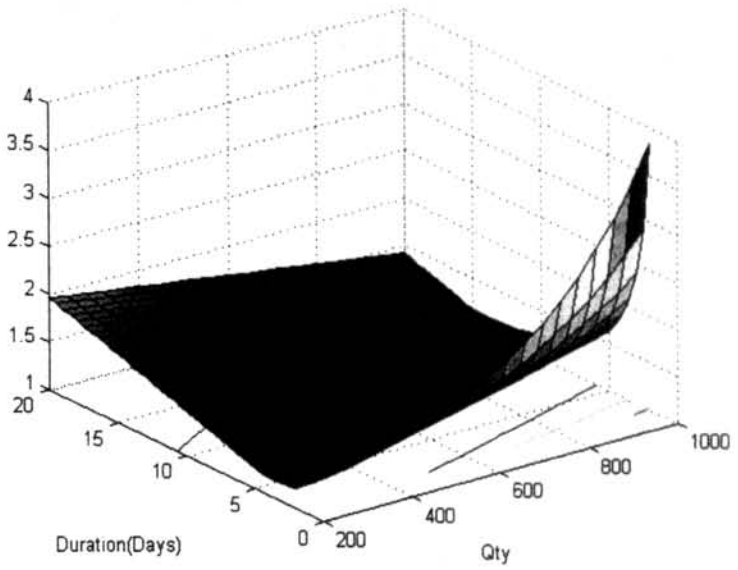
A.3 Informação logística

A informação logística apresentada no quadro seguinte foi naturalmente definida para a estrutura produtiva apresentada anteriormente. A unidade do tempo de transporte é o *dia* e os custos de transporte e de armazenamento são por unidade de produto.

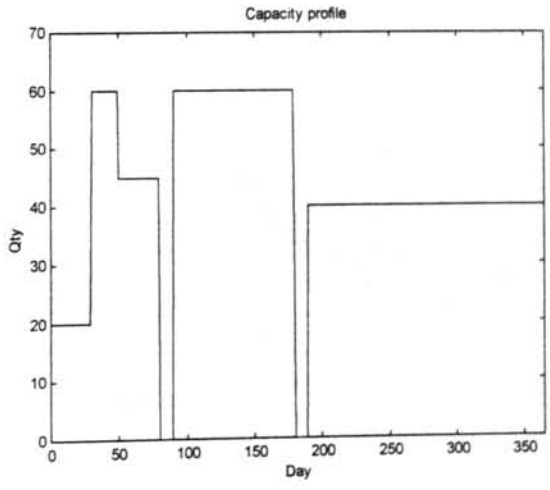
estágio	de	para	tempo transp.	custo transp.	custo armaz.
1	11	21	2	0.5	1.4
1	11	22	1	1.0	1.5
1	12	21	1	0.6	1.1
1	12	22	2	0.3	1.1
1	13	21	1	0.5	1.1
1	13	22	1	1.0	1.2
1	14	21	4	0.1	1.1
1	14	22	6	0.1	1.1
2	21	31	1	0.2	1.1
2	21	32	1	0.2	1.1
2	21	33	1	0.3	1.2
2	22	31	2	0.0	1.3
2	22	32	1	0.0	1.2
2	22	33	1	0.4	1.1
3	31	41	4	0.1	1.1
3	31	42	2	0.1	0.3
3	32	41	1	0.4	1.1
3	32	42	2	0.4	1.1
3	33	41	3	0.4	1.1
3	33	42	2	0.2	1.1

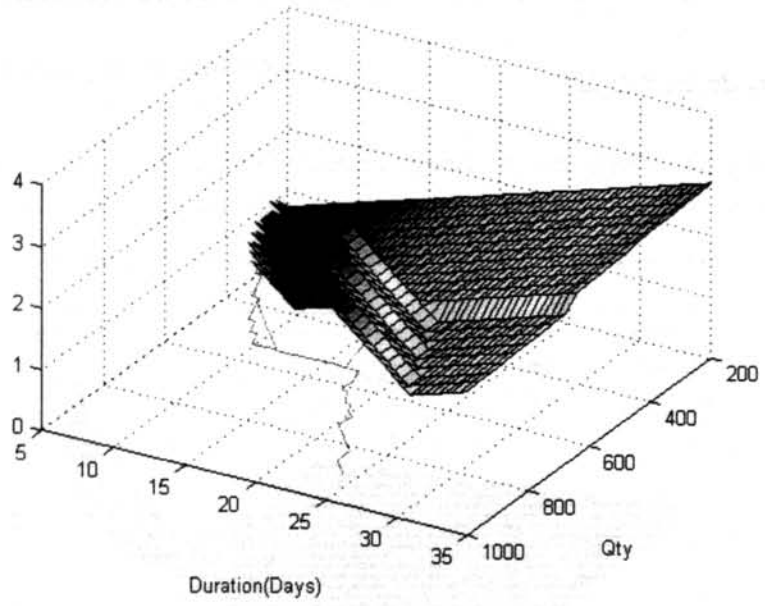
A.4 Exemplos de respostas dos modelos de capacidades

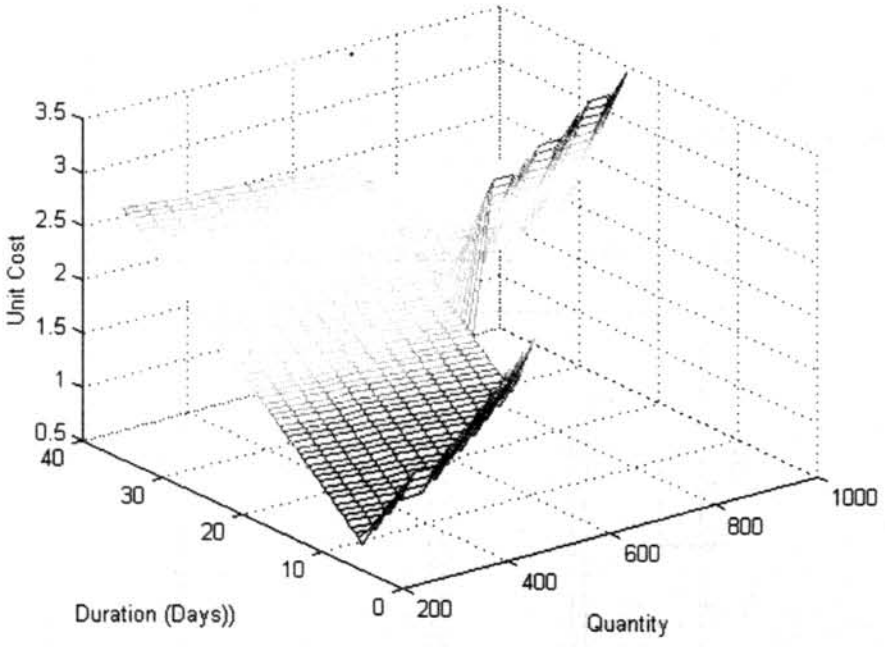
Resposta do modelo 12



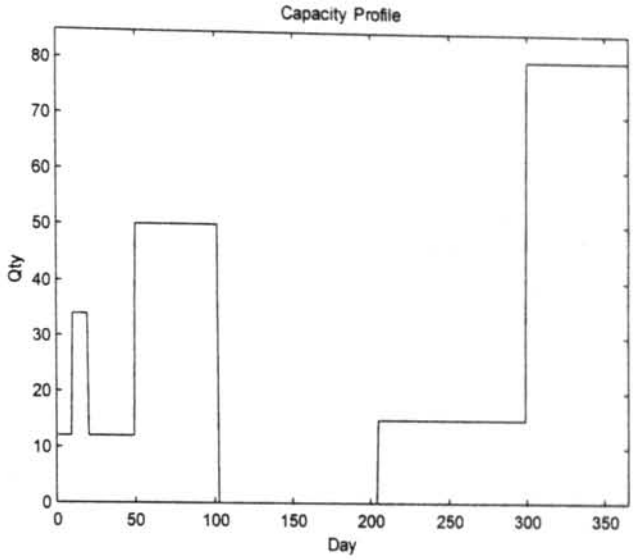
Resposta do modelo 42

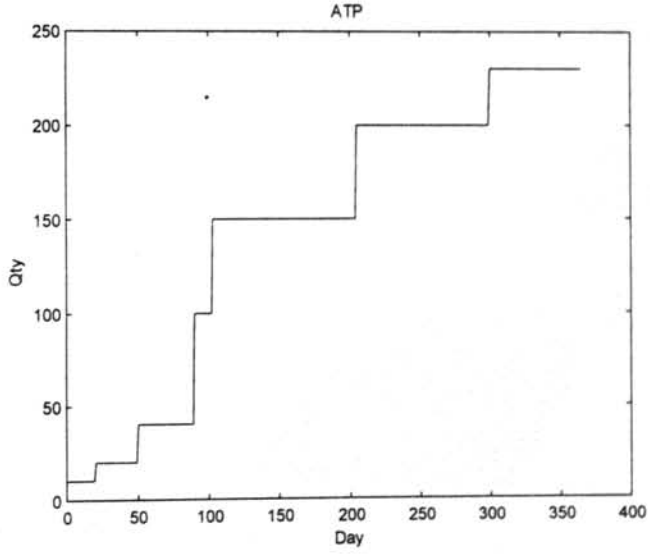


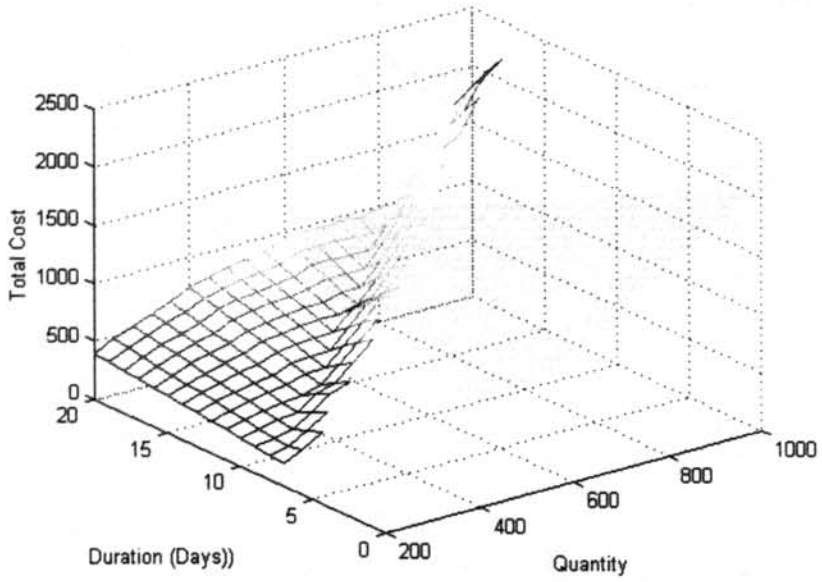

4
3
2
1
0
5
10
15
20
25
30
35
Duration(Days)
200
400
600
800
1000
Qty

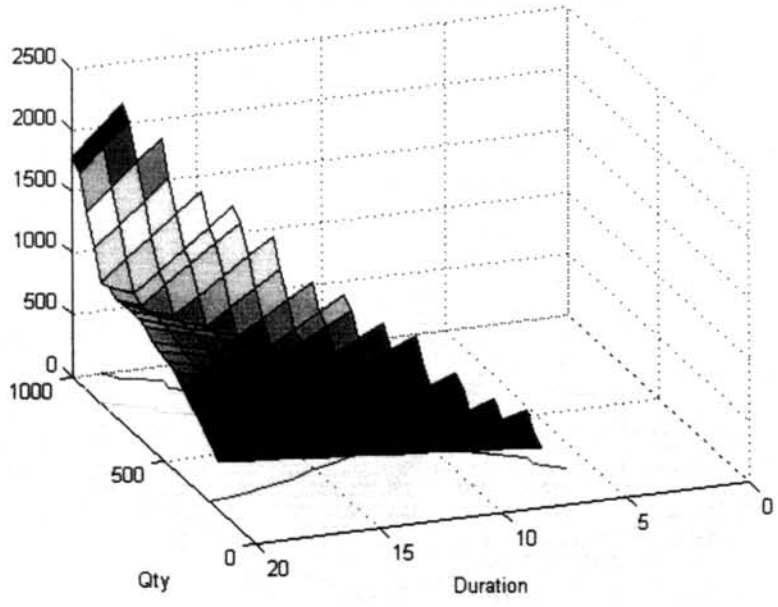

3.5
3
2.5
2
1.5
1
0.5
Unit Cost
40
30
20
10
0
Duration (Days))
200
400
600
800
1000
Quantity

Resposta do modelo 33






Total Cost
2500
2000
1500
1000
500
0
20
15
10
5
0
Duration (Days))
200
400
600
800
1000
Quantity


2500
2000
1500
1000
500
0
1000
500
0
Qty
20
15
10
5
0
Duration

Apêndice **B**

Extracto da Ontologia VEO (*Virtual Enterprise Ontology*)

B.1 Presentation

B.2 Virtual Enterprise Ontology Overview

B EXTRACTO DA ONTOLOGIA VEO (*VIRTUAL ENTERPRISE ONTOLOGY*)

Neste apêndice apresenta-se um extracto da ontologia *Virtual Enterprise Ontology* –Version 1.0.

B.1 Presentation

In the main sections presenting the ontology, each term is introduced with a definition. Within each section, we proceed by defining the terms that we regard as most basic first, and then define other terms using these basic ones.

The definitions are written in carefully chosen English with other Ontology terms in UPPER case. A term is defined using a base word, however for convenience of exposition, we use grammatical variations also in upper case as if they were themselves officially defined (e.g. ACHIEVE, ACHIEVEMENT).

Occasionally, we will use a word informally that is also used as an official term in the Ontology. The general rule is that official terms that appear in lower case, and all other words, should be interpreted in their dictionary sense in the light of their context. The definition of each term is intended to be necessary and sufficient as far as this is possible in natural language. However, in many cases it is felt essential to provide clarification or additional information. This is done as notes following the definition.

Besides the conventions described above, adopted from the Eo, the following ones were adopted for the VEo: terms in upper case are terms defined in the VEo; terms in upper case and italics are terms defined in the Eo; blocks of text in italics are transcribed from the Eo.

B.1.1 Terms (from Eo v1.1)

The central purpose of the Enterprise Ontology is to achieve effective sharing of meaning. The Enterprise Ontology consists mainly of:

Defined Terms: Terms explicitly included in the Enterprise Ontology. In addition to the natural language definition provided here, there is also a formally coded definition.

Related Terms: To better understand the Enterprise Ontology, it is helpful to know how its terms and concepts relate to the terms and concepts widely used in other contexts (e.g. other ontologies). Where possible, we specify the relationship between these terms and those in the Enterprise Ontology. These related terms fall into three categories:

1. Synonyms: Terms recognised as widely used in enterprises that are not defined in the Ontology, but which are considered the same or very close in meaning to defined terms.

2. Borderline Terms: Terms for which we make an attempt to show how they might be defined using Ontology terms. However, because they are deemed insufficiently important for sharing, are not formally included in the Ontology.

3. Other Commonly Used Terms: A list of commonly used terms that were not defined.

B.2 Virtual Enterprise Ontology Overview

The sections in this ontology are as follows:

- Enterprise Ontology Overview
- Plan and Order Management
- Activities and Resources
- Networked/Extended Organisations

B.2.1 Plan and Order Management

The Plan Management is responsible for the ACTIVITY SPECIFICATION with the INTENDED PURPOSE of producing the PRODUCTION PLANS for the several UNITS of the VE. Along with this definitions, GLOBAL and LOCAL PLANNING are addressed with the definitions of ROUGH and FINE PRODUCTION PLAN which are differentiated by the granularity of the information processed.

The Order Management addresses the concepts related to ORDERS within the VE. The specification of an ORDER ITEM, as an individual item in an ORDER is presented. Also the differentiation between INTERNAL CUSTOMER and EXTERNAL CUSTOMER (to the VE) is pointed out. Important concepts in the VE are SUPPLIER and CUSTOMER. This two types of ENTITIES are interchangeable over time, and their ROLEs are dependent on the transactions that they guarantee. Other important aspects regarding the Order Management are the ENQUIRY, QUOTATION, DUE DATE and DELIVERY DATE, which are ACTIVITIES and STATEMENTS necessary to support the RELATIONSHIPS between the ENTITIES of the VE.

B.2.1.1 Table 1: Plans management

***PLAN**: an ACTIVITY SPECIFICATION with an INTENDED PURPOSE. Notes: see notes under ACTIVITY SPECIFICATION.*
***PLANNING**: an ACTIVITY whose INTENDED PURPOSE is to produce a PLAN.*
PRODUCTION PLAN: is a *PLAN* whose *INTENDED PURPOSE* is to specify when and where PRODUCTION ACTIVITIES will be performed. A

PRODUCTION PLAN produces an assignment of *ACTIVITIES* to VE UNITS and *TIME INTERVALS*
ROUGH PRODUCTION PLAN: is a PRODUCTION PLAN based on aggregate information and setting *TIME INTERVALS* for production in the VE UNITS considered to be feasible candidates
FINE PRODUCTION PLAN: is a PRODUCTION PLAN based on more detailed information, and setting *CALENDAR DATE's* for production in specific VE UNITS
GLOBAL PLANNING: an *ACTIVITY* whose *INTENDED PURPOSE* is to produce PRODUCTION PLAN's
LOCAL PLANNING: an *ACTIVITY* whose *INTENDED PURPOSE* is to produce PRODUCTION *PLAN's* for a single VE UNIT.

B.2.1.2 Table 2: Orders management

ORDER: it is a CUSTOMER request for PRODUCT, issued to a SUPPLIER.. An ORDER contains one or more ORDER ITEMS (an ORDER contains general information such as the CUSTOMER identifier, number of items, shipping address, etc.)
ORDER ITEM: an item within an ORDER. This includes reference to the PRODUCT SPECIFICATION, the quantity of PRODUCT to produce, and the CUSTOMER's requested DELIVERY DATE.
CUSTOMER ORDER: is an ORDER in which the CUSTOMER is an EXTERNAL CUSTOMER (a CUSTOMER ORDER can be in one of the following states: under negotiation, confirmed, planned, being executed, completed.)
PRODUCTION ORDER: is an ORDER in which the CUSTOMER is an INTERNAL CUSTOMER (a PRODUCTION ORDER can be in one of the following states: under planning, planned, being executed, on schedule, late, stopped, cancelled and completed.)
PRODUCT REQUEST: is a statement defining a CUSTOMER's needs, in terms of PRODUCT types, quantities and DUE DATES.
CUSTOMER: the requirer of a PRODUCT. He expresses is requirements in terms of ORDERS. A CUSTOMER can be an INTERNAL CUSTOMER or an EXTERNAL CUSTOMER depending on its commitment to the VIRTUAL ENTERPRISE.
SUPPLIER: a provider of goods or services.
PRODUCT: is the result of a manufacturing *ACTIVITIES* chain. A manufacturing enterprise *PURPOSE* is to make PRODUCTS.
ENQUIRY: is an *ACTIVITY* that an EXTERNAL CUSTOMER *EXECUTES* with the *PURPOSE* of obtaining information about PRODUCTS, prices and DELIVERY DATES. This process eventually leads to placing a COSTUMER ORDER.
QUOTATION: is a statement of price, description of PRODUCTS, DUE DATES, and other terms of sale, offered by a supplier to a prospective CUSTOMER.
DUE DATE: is the *CALENDAR DATE* when PRODUCTS are available for delivery.

DELIVERY DATE: is the *CALENDAR DATE* when PRODUCTS are delivered to the *CUSTOMER*.
ORDER PROMISING: an *ACTIVITY* whose *PURPOSE* is to produce a QUOTATION in response to an EXTERNAL CUSTOMER REQUEST.
AVAILABLE-TO-PROMISE: is a quantity of PRODUCT available to allocate in a given *TIME POINT* to satisfy a CUSTOMER REQUEST. This quantity is calculated on the basis of the company's inventory and actual and planned production.

B.2.2 Activities and Resources

This section presents the central concepts of ACTIVITY, and of relationships between ACTIVITIES and other entities such as RESOURCE. An important related concept presented is CAPACITY, which is a quantity, in some unit measure, that is available for allocation to ACTIVITIES over time.

ACTIVITIES are performed over a particular TIME INTERVAL and are subject to constraints. Constraints restrict the ways in which ACTIVITIES can use RESOURCES. A RESOURCE provides CAPACITY for use in performing ACTIVITIES. The CAPACITY of a RESOURCE constraint the number of ACTIVITIES that it can simultaneously support. The performance of any given ACTIVITY requires some amount of CAPACITY of one or more RESOURCES.

B.2.2.1 Table 3: Activities and Resources

ACTIVITY: *something done over a particular TIME INTERVAL. The following may pertain to an ACTIVITY: has PRE-CONDITION(S);has EFFECT(S);is performed by one or more DOERS; is decomposed into more detailed SUB-ACTIVITIES;entails use and/or consumption of RESOURCES; has AUTHORITY requirements; is associated with an [ACTIVITY] OWNER; has a measured efficienc.*
ACTIVITY SPECIFICATION: *a characterisation of something to do; a specification of activity.*
EXECUTE: *a Relationship between one or more Potential Actors and an ACTIVITY SPECIFICATION whereby the one or more Potential Actors perform the specified ACTIVITIES.*
RESOURCE: *an Entity that can be used or consumed during the performance of the ACTIVITY or the ACTIVITIES.*
RESOURCE ALLOCATION: *the allocation of RESOURCES to ACTIVITIES.*
CAPACITY: is an *ATTRIBUTE* of a *RESOURCE* representing what is available for allocation to *ACTIVITIES* over time. The CAPACITY of a *RESOURCE* constrains the number of *ACTIVITIES* that it can simultaneously support.

B.2.3 Networked/Extended Organisations

A VIRTUAL ENTERPRISE is a set of VE UNITS. These are separately managed *ORGANISATIONAL UNITS* (plants, subcontractors, transport companies, warehouses, "customers", ...) that have some kind of relationship to the business. As opposed to "node" (in a network) that has a "mathematical" connotation, the term "unit" better reflects the manufacturing terminology. For each of these units, we define the level-of-control (loc), in some rough, partially subjective way, say in scale 0-1. This means that our core company has a loc of 1, a given plant may have a loc of 0.7, a given subcontractor a loc of 0.3, a given "customer" might have a loc of 0.1 or close to 0. Then we have, for the different units, some kind of "measure" of "ownership" towards the main (core) management entity (of course, this is not directly related to real ownership). This setting should allow us to define the VE in a kind of "fuzzy" way, not imposing a strict definition of the borders (type of "you are inside the VE, or you are out in the world...).

B.2.3.1 Table 4: Networked/Extendend organisations

VIRTUAL ENTERPRISE: is a set of inter-related VE UNITS that are totally or partially committed to some common *PURPOSE* (the relation between VE UNITS can be one of *OWNERSHIP* or a looser one of belonging to a common supplier chain)
VE UNIT: An *ORGANISATIONAL UNIT* belonging to the set that composes a VIRTUAL ENTERPRISE. A VE UNIT may be a manufacturing/processing plant, a subcontractor, a transport provider, a supplier or a warehouse (the term *OU* is here used in its broader sense i.e., as an UNIT providing the a complete set of products or services, e.g. a plant, a manufacturing unit, a warehouse, etc.)

Apêndice C

Extracto das interfaces IDL definidas

C.1 Componente *Order Promise Client*

C.2 Componente *Order Promise Server*

C.3 Componente *Rough Planner*

C EXTRACTO DAS INTERFACES IDL DEFINIDAS

Neste apêndice apresenta-se um extracto das interfaces IDL definidas e implementadas nos componentes descritos no Capítulo 4.

C.1 Componente *Order Promise Client*

```
INTERFACE:      OrderRequest
SERVER:         OPServer
CLIENT:         OPClient

module OPS_I {
   interface OrderRequest {
      boolean CreateNewOrder(inout GOS::Order order);
      boolean CreateNewOrderItem(inout GOS::OrderItem orderItem);
      boolean RequestDeliveryItems(inout GOS::OrderItem orderItem);
      boolean RequestOrderItemConfirmation(inout GOS::OrderItem
      orderItem);
      boolean RequestOrderItemRejection(inout GOS::OrderItem orderI-
      tem);
   };
};

INTERFACE:      OrderResponse
SERVER:         OPClient
CLIENT:         OPServer

module OPC_I {
   interface OrderResponse {
      oneway void TakeDeliveryItemSeq(in GOS::DeliveryItemSeq deliver-
      yItemSeq);
      oneway void TakeOrderItemConfirmation(in GOS::OrderItem orderI-
      tem, in boolean status);
      oneway void TakeOrderItemRejection(in GOS::OrderItem orderItem,
      in boolean status);
   };
};

INTERFACE:      OpStaticView
SERVER:         GIM
CLIENT:         OPClient

interface StaticView {
   // Operations to request static data from the GOS
   //...
   GOS::CustomerSeq getCustomer();
   GOS::FinalProductSeq getFinalProduct();
```

```
   //...
};

interface OpStaticView : StaticView {
   // Operations to request static data from the GOS
   // This object is instantiated, on request, by the GIM.
};
```

C.2 Componente *Order Promise Server*

```
INTERFACE:      I_OP_RP
SERVER:         Rough Planner
CLIENT:         OPServer
INTERFACE:      I_RP_OP
SERVER:         OPServer
CLIENT:         Rough Planner

module OP_RP {
   enum NotificationStatus {acceptance, rejection};
   enum EntryMode {internal, external, demandplan};

   interface I_OP_RP { // Messages from OPServer to RP
      oneway void RoughPlanningRequest(in GOS::OrderItem OItem, in
      EntryMode source, in GOS::StringSeq veUnitIDSeq,in string obj-
      Ref);

      oneway void Notification(in GOS::OrderItem OItem, in Notifica-
      tionStatus status, in EntryMode source, in string objRef);

      oneway void ChangeRequest(in GOS::OrderItem OItemOld, in
      GOS::OrderItem OItemNew, in string objRef);
   };

   interface I_RP_OP {     // Messages from RP to OP
      oneway void RoughPlanningResult(in GOS::OrderItem OItem, in
      GOS::DeliveryItemSeq Optsolution, in NotificationStatus status,
      in EntryMode source);

      oneway void Acknowledgment(in GOS::OrderItem OItem, in Notifica-
      tionStatus status, in EntryMode source);
   };
};


INTERFACE:      OpActionView
SERVER:         Global Object Server
CLIENT:         OPServer

interface OpActionView {

   // Operations to Insert/Update/Remove dynamic/static data on the
      GOS.
```

```
   void insertCustomer(in GOS::Customer customer);

   void insertOrder(inout GOS::Order order);
   void updateOrder(in GOS::Order order);
   void removeOrder(in string orderID);

   void insertOrderItem(in GOS::OrderItem orderItem);
   void updateOrderItem(in GOS::OrderItem orderItem);
   void updateOrderItemStatus(in string orderID,
      in string orderItemID,in GOS::OrderItemStatus status);
   void removeOrderItem(in string orderID, in string orderItemID);
   void removeDeliveryItem(in string orderID, in string orderItemID,
      in string deliveryItemID);
    void removeDemandPlanOrder(in string orderItemID);
  };
```

```
INTERFACE:      I_GRC_OP
SERVER:         OPServer
CLIENT:         Global Reactive
INTERFACE:      I_OP_GRC
SERVER:         Global Reactive
CLIENT:         OPServer
```

```
module GRC_OP {    // Module between the RP and the GRC module

   enum NotificationStatus{acceptance, rejection};
   enum ReplanningResult{success, failure};

   struct GRCReplanningRequest {
      string orderID;
      string orderItemID;
      GOS::StringSeq veuoIdSeq;
   };

   struct RPReplanningResult {
      string orderID;
      string orderItemID;
      ReplanningResult res;
   };

   struct RPNotification {
      string orderID;
      string orderItemID;
      NotificationStatus state;
   };

   struct RPAcknowledgement {
      string orderID;
      string orderItemID;
      NotificationStatus state;
   };

   interface I_GRC_OP {// Messages from GRC to OP Server
```

```
    // connection methods
    oneway void registerGRC(in string objRef);
    oneway void unregisterGRC();

    // replanning methods
    oneway void replan(in GRCReplanningRequest rpReq);
    oneway void notification(in RPNotification notification);
  };

  interface I_OP_GRC { // Messages from OP Server to GRC

    oneway void replanningState(in RPReplanningResult res);
    oneway void acknowledgement(in RPAcknowledgement ack);
  };
};
```

```
INTERFACE:    DemandPlan
SERVER:       OPServer
CLIENT:       DemandPlan

module DEMAND_PLAN_I {

  interface DemandPlan {

    void init()
      raises (OperationFailure);
    void close();

    void insertOrder(in Order newOrder)
      raises (OperationFailure, OrderNotInserted);
    void updateOrder(in string ID, in Order newOrder)
      raises (OperationFailure, OrderNotInserted);
    void deleteOrder(in string ID)
      raises (OperationFailure);
    void deleteAllOrders()
      raises (OperationFailure);

    Order getOrder(in string ID)
      raises (OperationFailure);
    OrderSeq getOrderSeq(in GOS::Date firstDay,in GOS::Date lastDay)
      raises (OperationFailure);
  };
};
```

C.3 Componente *Rough Planner*

```
INTERFACE:    I_RPView_RP
SERVER:       Rough Planner
CLIENT:       RPView
```

```
interface I_RPView_RP {  // Messages from RPView to RP

   oneway void registerRP_Client(in I_RP_RPView clientInterface);
   oneway void resetRP();
   oneway void sendTuning(in GlobalStage globalStage);
   boolean getTuning(inout GlobalStage globalStage);
};

INTERFACE:      I_RP_RPView
SERVER:         RPView
CLIENT:         Rough Planner

interface I_RP_RPView {  // Messages from RP to RPView

   oneway void sendOneCostValue(in float cost);
   oneway void initRouting();
   oneway void sendRoutingElement(in VEDataOffer acceptedOffer);
   oneway void sendNegotiationElement(in VEInqOffer transaction);
   oneway void sendAdded_RCM(in string RCM, in string type);
   oneway void sendRemoved_RCM(in string RCM, in string type);
   oneway void sendRPStatus(in string status, in long type);
   oneway void sendRPInputData(in GOS::OrderItem OItem,in ReqType
      source);
   oneway void initRPView();
};


INTERFACE:      I_RP_RCM
SERVER:         Rough Capacity Model
CLIENT:         Rough Planner

interface I_RP_RCM {// Messages from RP to RCM

   oneway void insertInquirySeq(in long inquirySeqID, in IntInquiry-
      Seq order);
   oneway void insertNotification(in VEIntNotification order);
};


INTERFACE:      I_RCM_RP
SERVER:         Rough Planner
CLIENT:         Rough Capacity Model

interface I_RCM_RP {// Messages from RCM to RP

   oneway void insertAcknowledge(in string veUnitID, in VEIntAcknow-
       ledge ack);
   oneway void insertOfferSeq(in string veUnitID, in long inquirySe-
       qID, in VEIntOfferSeq offers);
   oneway void registerRCM(in string veUnitID, in VeUnitType veU-
      nitType,in string objRef);
   oneway void unregisterRCM(in string veUnitID);
};
```

```
INTERFACE:      I_LM_RP
SERVER:         Rough Planner
CLIENT:         Logistic Model
INTERFACE:      I_RP_LM
SERVER:         Logistic Model
CLIENT:         Rough Planner

module LM_RP {     // Module between LM component and the RP component

   enum notificationStatus {acceptance, rejection};
   enum TypeLogisticData {transportation, storage};
   enum TypePointNetwork {unit, stage};
   enum LogisticCriterion {time, cost };

   struct reqType {
      TypeLogisticData req;
      LogisticCriterion logisticCriterion;
      TypePointNetwork typeFrom; //stage or unitID specification
      TypePointNetwork typeTo; //stage or unitID specification
      string FromUnit; //from stage X or from Unit x
      string ToUnit; // to stage Y or to Unit y
      GOS::ProductionStage FromStage; //from stage X or from Unit x
      GOS::ProductionStage ToStage; // to stage Y or to Unit y
      GOS::Date StartPeriod;
      GOS::Date EndPeriod;
      string ProductID;
      long Quantity;
   };

   struct logisticData {
      TypeLogisticData req;
      LogisticCriterion logisticCriterion;
      TypePointNetwork typeFrom;//stage or unitID specification
      TypePointNetwork typeTo;//stage or unitID specification
      string FromUnit;
      string ToUnit;
      GOS::ProductionStage FromStage;//from stage X or from Unit x
      GOS::ProductionStage ToStage; //to stage Y or to Unit y
      GOS::Date StartDate;
      long TransportTime;
      float costs;
      notificationStatus statusResponse;
   };

   interface I_LM_RP { // Messages from LM to RP

     void requestLogisticData(in reqType reqData, out logisticData
       outData);
   };

   interface I_RP_LM { // Messages from RP to LM

     oneway void notificationFromCostModel(in notificationStatus sta-
       tus);
   };
};
```

BIBLIOTECA F.E. PORTO

FACULDADE DE ENGENHARIA
UNIVERSIDADE DO PORTO
BIBLIOTECA
0000049055