

Maria Antónia da Silva Lopes de Carravilla

Modelos e Algoritmos para o Planeamento Hierárquico da Produção

Aplicações a um Caso de Estudo

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

1996

O trabalho de investigação apresentado nesta dissertação foi parcialmente suportado pela bolsa BD/1648/91-IA do programa CIÊNCIA, e pela bolsa BD/3658/94 do programa PRAXIS XXI

A UM DEUS DESCONHECIDO

...Dirigiu-se para casa na escuridão, acendeu os candeeiros e também o lume do fogão. O relógio a que Elisabeth dera corda, continuava o seu tiquetaque, armazenando na mola a pressão das mãos dela, e as peúgas de lã que ela estendera a secar sobre o guarda-fogo ainda estavam húmidas. Eram partes vitais de Elisabeth que ainda não tinham morrido. Joseph ponderou lentamente sobre isto - a vida não se pode cortar bruscamente. Não se pode estar morto enquanto as coisas que nós alterarmos não morrem. Os nossos efeitos são a única evidência da nossa vida. Enquanto perdurar nem que seja uma recordação dolorosa, uma pessoa não pode ser amputada, morta. E pensou: "É um processo lento e demorado, isto de uma pessoa morrer. Mata-se uma vaca, e ela morre assim que se lhe comer a carne; mas a vida do homem morre como morre a vibração num charco tranquilo, em pequenas ondas, alastrando e crescendo até à quietude."...

...Tudo parece ter um ritmo repetido excepto a vida. Há um só nascer e um só morrer. Não há coisa nenhuma que se lhe assemelhe...

John Steinbeck

Resumo

Nesta dissertação descrevem-se os trabalhos de investigação realizados no âmbito dos problemas de Planeamento Hierárquico da Produção e baseados no projecto e implementação de um Sistema de Apoio à Decisão (SAD) para uma empresa da indústria do calçado. É feita uma descrição detalhada desse Caso de Estudo, começando por se descrever o processo de fabrico em causa, a hierarquia dos processos de tomada de decisões e o SAD. É apresentado o carácter hierárquico do processo de tomada de decisões, reflectido no SAD, seguindo-se uma descrição de cada um dos componentes do SAD.

Seguidamente, descrevem-se em pormenor os diversos níveis do processo de decisão considerados no SAD, apresentando-se para cada um desses níveis um modelo de Programação Matemática, heurísticas que foram implementadas no SAD e as reacções dos Agentes de Decisão aos resultados fornecidos por essas heurísticas.

Aborda-se também o carácter multiobjectivo de um dos níveis do processo de decisão considerados no SAD, sendo descrita uma heurística multiobjectivo desenvolvida especialmente para esse caso. São em seguida comparados os resultados dessa heurística com os resultados obtidos por métodos analíticos específicos da Decisão Multicritério, utilizando-se para esse efeito uma instância gerada aleatoriamente.

Considerou-se interessante explorar uma generalização do problema associado a um dos níveis do processo de decisão, apresentando-se algumas abordagens promissoras desenvolvidas para esse problema. Uma primeira abordagem baseia-se na exploração de propriedades estruturais do problema, com vista à obtenção de desigualdades válidas para um modelo de Programação Inteira proposto, que permite calcular minorantes (“lower bounds”) sobre

o valor da solução óptima. Estes minorantes, integrados em métodos de resolução do tipo “branch-and-bound”, aumentam a eficiência da resolução do problema de optimização.

A segunda abordagem genérica, baseia-se na utilização de meta-heurísticas “Pesquisa Local Aleatorizada”, Pesquisa Tabu e “Simulated Annealing”, que permitem a obtenção eficiente de soluções satisfatórias, ou seja de majorantes (“upper bounds”) sobre o valor da solução óptima.

Da comparação das diferentes abordagens e dos algoritmos resultantes, feita com instâncias do problema geradas aleatoriamente, conclui-se que, com os métodos apresentados, por si só ou incluídos em métodos de optimização mais sofisticados, se poderão obter boas soluções, em tempo reduzido, para classes mais gerais de problemas, ou para outros de acrescida complexidade mas de maior interesse prático.

Abstract

In this thesis, we describe the design and implementation of a Decision Support System (DSS) for the shoe industry and the related hierarchical production planning problems. A full description of the case study follows, including the production process, the hierarchy in the decision making process and the DSS.

The hierarchy implemented in the DSS reflects the hierarchy in the decision making process. We describe the two hierarchical levels existing in the DSS. For each level we present an Integer Programming model, the implemented algorithms, and the users feedback.

We did perform a Multicriteria analysis of the higher decision level in the DSS generating a representative sample of the non-dominated solutions followed by a decision-aid procedure and an interactive method, enabling the planner to incorporate his preferences in the decision process, instead of having to accept an unique “optimal” solution. We present a problem-oriented heuristic to tackle this problem and a comparison of the obtained results with those obtained using specific analytical multicriteria methods for a randomly generated instance of the problem.

We considered important to study a generalisation of the problem of the higher level of the DSS. A first approach exploits the properties of the Integer Programming model which enable to obtain Valid Inequalities. These inequalities are used in computing lower bounds for the optimal integer solution and in the development of a problem solving procedure.

A second approach uses Metaheuristics, such as “Generalized Local Search”, Taboo Search and Simulated Annealing, and good upper bounds on the optimal solution were obtained using these techniques.

Comparing the results obtained with some instances of the problem, we can conclude that this two approaches alone or embedded in more sophisticated optimization methods may provide fast and good solutions even for more general and complicated practical problems.

Prefácio

Esta dissertação teve a sua origem no projecto e implementação de um SAD para uma empresa da indústria do calçado e foi realizada sob a supervisão do Professor Jorge Pinho de Sousa, a quem agradeço o tempo e os conhecimentos que me ofereceu e ainda a paciência e o cuidado com que leu as múltiplas versões dos vários capítulos. O Jorge excedeu, em dedicação e apoio, tudo o que se pode esperar de um orientador e de um amigo.

Durante a preparação desta dissertação, tive a sorte de estar sempre rodeada de amigos a quem estou muito grata por criarem um ambiente único de trabalho, e por estarem sempre prontos para dar a “pequena ajuda” que se torna tão preciosa em momentos críticos. A Helena Correia, o António Miguel Gomes, o Luís Ferreira, o Pedro Borges, o Moutinho e mais recentemente a Teresa Costa, formam o ambiente no qual eu gostaria de continuar a trabalhar.

Ao António Marcelo devo um agradecimento especial, por ter permitido que eu me dedicasse a 100% ao desenvolvimento dos algoritmos para o SAD, libertando-me de outras tarefas bem menos gratificantes. Não esquecerei nunca que foi ao longo de longas tardes de discussão, que foram desenvolvidos os esqueletos dos algoritmos implementados no SAD.

Foi propositadamente que deixei o José Fernando Oliveira para último. É óbvio que só lhe tenho que agradecer as críticas (que por vezes tenho tanta dificuldade em aceitar) as discussões, a leitura cuidada dos capítulos da tese e o apoio dado nos momentos mais críticos. Enfim, só tenho que lhe agradecer por ser um Amigo.

Quero também agradecer ao Gabriel David por não me ter deixado dar as aulas de TS “para acabares depressa essa coisa!!”, e à Margarida Amorim por

ser sempre um refúgio seguro quando há algum problema com a matemática. Ao Prof. Manuel Matos, agradeço os comentários e sugestões dadas sobre aspectos da Decisão Multicritério.

Estou muito grata ao INESC por me ter proporcionado excelentes condições de trabalho, e à JNICT (programa CIENCIA e PRAXIS XXI) por me ter concedido uma bolsa ao longo de 4 anos.

Agradeço também a toda a minha família e em especial à minha Mãe, pela longa experiência de Vida que me transmitiu, e aos meus amigos, cujo entusiasmo e lealdade me são tão preciosos.

O Eduardo é uma parte inseparável de mim. É impossível agradecer-lhe.

Conteúdo

1	Introdução	1
1.1	Motivação	1
1.2	Objectivos da investigação	3
1.3	Estrutura da dissertação	4
1.4	Enquadramento	5
1.4.1	Planeamento Hierárquico	6
1.4.2	Carácter multicritério do problema	10
1.4.3	Sistema de Apoio à Decisão (SAD)	12
1.4.4	Modelos de Optimização Combinatória (OC)	14
2	Caso de Estudo	21
2.1	Introdução	21
2.2	Descrição do processo de fabrico	22
2.2.1	Corte	22
2.2.2	Costura	25
2.2.3	Montagem	26
2.2.4	Acabamentos/Embalagem	27
2.3	Processos de tomada de decisões	27
2.4	Dos processos de tomada de decisões ao SAD	29
2.5	Planeamento Hierárquico	31
2.5.1	Enquadramento hierárquico	31
2.5.2	Interpretação “hierárquica” do SAD em estudo	32
2.5.3	Plano Director da Produção (PDP)	36
2.5.4	Programação da Produção (PROG)	36
2.6	Sistema de Apoio à Decisão (SAD)	37

2.6.1	Gestão de dados	38
2.6.2	Modelos e Algoritmos	40
2.6.3	“Interface”	40
2.6.4	Exemplo de utilização do SAD	41
3	Plano Director da Produção	47
3.1	Introdução	47
3.2	Modelo	48
3.2.1	Funções objectivo	48
3.2.2	Restrições	49
3.2.3	Dimensão do modelo para problemas reais	50
3.3	Heurística para o Caso de Estudo (<i>H-PDP</i>)	51
3.3.1	Critérios para avaliação de soluções	51
3.3.2	Heurística <i>H-PDP</i> : descrição geral	52
3.3.3	Dados de entrada	53
3.3.4	Resultados	55
3.3.5	Descrição genérica do algoritmo	55
3.3.6	Exemplo ilustrativo	56
3.3.7	Aplicação da heurística <i>H-PDP</i> ao Caso de Estudo	57
4	PDP - uma Abordagem Multicritério	61
4.1	Introdução	61
4.2	Conceitos fundamentais	62
4.3	Objectivos do problema PDP	65
4.4	Métodos analíticos	66
4.4.1	Método paramétrico	67
4.4.2	Método das restrições	67
4.5	Heurística multiobjectivo	68
4.6	Testes computacionais	71
4.6.1	Geração de um problema teste	71
4.6.2	Resultados obtidos pela heurística <i>H-PDP</i>	71
4.6.3	Resultados obtidos pela heurística multiobjectivo	72
4.6.4	Resultados obtidos pelo método das restrições	74
4.6.5	Comparação de resultados	76

4.7	Apoio na tomada de decisão	76
5	Programação da Produção	79
5.1	Introdução	79
5.2	Modelo	81
5.2.1	Funções objectivo	81
5.2.2	Restrições	82
5.2.3	Dimensão do modelo para problemas reais	83
5.3	Heurística baseada em PL (<i>H-PROG1</i>)	84
5.3.1	<i>H-PROG1</i> , parte 1	85
5.3.2	Procedimento de arredondamento	86
5.3.3	<i>H-PROG1</i> , parte 2	86
5.3.4	Procedimento para reduzir as mudanças de “layout”	88
5.3.5	<i>H-PROG1</i> - conclusões	90
5.4	Heurística para o Caso de Estudo (<i>H-PROG</i>)	90
5.4.1	Critérios para avaliação das soluções	90
5.4.2	Heurística <i>H-PROG</i> : descrição geral	91
5.4.3	Dados de entrada	92
5.4.4	Resultados	94
5.4.5	Descrição genérica do algoritmo	94
5.4.6	Exemplo ilustrativo	95
5.4.7	Aplicação da heurística <i>H-PROG</i> ao Caso de Estudo	98
6	Um Modelo de Prog. Inteira e Desigualdades Válidas	101
6.1	Motivação	101
6.2	Identificação do problema	102
6.2.1	Modelo 0 (PDP)	102
6.2.2	Modelo 1 (encomendas unitárias e $r_e = 0$)	103
6.2.3	Modelo 2 (encomendas do mesmo tipo de produto)	104
6.2.4	Modelo 3 (encomendas de 1, 2 e 3 unidades e uma só linha)	105
6.3	Modelo PAP (modelo 3)	107
6.3.1	Propriedade 1 - Ordenação das encomendas do mesmo tamanho	107

6.3.2	Propriedade 2 - Valor das variáveis na relaxação linear	108
6.4	Pré-processamento	109
6.5	Desigualdades válidas	111
6.6	Algoritmo de planos de corte	113
6.6.1	Heurística de separação	113
6.7	Resultados computacionais	114
6.7.1	Conclusões	116
7	Meta-heurísticas	119
7.1	Notação	120
7.2	Pesquisa Local (LS - “Local Search”)	121
7.3	Geração de soluções iniciais	122
7.3.1	Heurística Básica para geração de soluções iniciais	122
7.3.2	Heurística Aleatória para geração de soluções iniciais	122
7.4	Melhoramento de soluções	123
7.5	Estruturas de vizinhança e subvizinhança	123
7.5.1	Vizinhança $\mathcal{N}_{Desloca(dist)}(x_n)$	124
7.5.2	Vizinhança $\mathcal{N}_{Troca(dist)}(x_n)$	125
7.5.3	Vizinhança $\mathcal{N}_{Mista(dist,p)}(x_n)$	126
7.6	Pesquisa Tabu (TS - “Taboo Search”)	127
7.7	“Simulated Annealing” (SA)	130
7.7.1	Parâmetros	131
7.8	Determinação de parâmetros	133
7.8.1	Primeiro conjunto de testes	136
7.8.2	Determinação dos valores dos parâmetros para o “Simulated Annealing”	147
7.8.3	Resumo dos resultados obtidos	156
7.9	Desempenho relativo das meta-heurísticas	157
7.9.1	Conclusões	161
7.9.2	Testes adicionais	161
7.10	Conclusões	164
8	Conclusões e Desenvolvimentos Futuros	167
	Referências	171

Apêndices	177
A Notação	179
A.1 Índices	179
A.2 Parâmetros	179
A.3 Variáveis de decisão	181
A.4 Grandezas auxiliares	181
B Funções usadas nas heurísticas PDP e PROG	185
B.1 ActualizaCapDisp()	186
B.2 ActualizaCargas(q, j, l, t)	187
B.3 ActualizaPrimPerDisp()	188
B.4 ActualizaUltimPerDisp()	189
B.5 CalculaFolgas()	190
B.6 CapProdLin(j, nlin, nper, t ₁)	191
B.7 CapProdTip(j, nper, t ₁)	192
B.8 CarregaLinhas(e, nlin, nper, t ₁ , caso)	193
B.9 ColocaMaisCedo(e)	194
B.10 ColocaMaisTarde(e)	195
B.11 ColocaPROG()	196
B.12 DeterminaLimites(e)	197
B.13 FazLayout()	198
B.14 Inicializa()	199
B.15 OrdPrefJL()	200
B.16 OrdPrefLJ()	201
B.17 RelaxaLimites(e)	202
B.18 RoundLayout()	203
C Exemplo PDP	205
C.1 Dados de entrada	205
C.2 Resolução	206
D Exemplo PROG	215
D.1 Dados de entrada	215
D.2 Resolução	215

E	Algumas Questões de Implementação	231
E.1	Plataformas utilizadas nos testes	231
E.2	Utilização de mmc_{lt}	232
E.3	Alteração dos valores de C_{jlt} , após a leitura de dados	232
E.4	Transição entre orçamentos	233
F	Problemas Teste	235
F.1	Problemas teste para o PDP	235
F.1.1	Processo de geração	235
F.1.2	Instância PDP1	238
F.2	Problemas teste para o modelo PAP	243
F.2.1	Processo de geração	243
F.2.2	Instâncias geradas	244
G	Demonstração da Propriedade 1 (secção 6.3.1)	251

Lista de Figuras

1.1	Modelo conceptual de um SAD	13
2.1	Diagrama do processo produtivo	23
2.2	Sequência de operações no sector do Corte	24
2.3	O SAD como parasita do Sistema Informático de Base (SIB)	30
2.4	Planeamento Hierárquico	31
2.5	Ligações entre níveis de planeamento e entre estes e o SIB	33
2.6	Plataforma do SAD	38
2.7	Exemplo de um ecran do SAD	42
2.8	Exemplo de um gráfico obtido como resultado no SAD	43
2.9	Tabela de encomendas e resultados do planeamento para encomenda número -39 (pedido número 53157)	44
2.10	Introdução de data de fim para encomenda número -39 (pedido número 53157) e resultados do novo planeamento	45
2.11	Alterações no plano com introdução de data de fim para encomenda número -39	45
3.1	Função objectivo <i>JIT</i>	49
3.2	Resultados do exemplo PDP	58
4.1	Valores de <i>JIT</i> e <i>PREF</i> , nas soluções da heurística multiobjectivo (<i>H-MO</i>)	73
4.2	Valores de <i>JIT</i> e <i>PREF</i> , nas soluções eficientes obtidas por <i>H-MO</i>	73
4.3	Valores de <i>JIT</i> e <i>PREF</i> , nas soluções eficientes obtidas pelo método das restrições	75

4.4	Comparação das soluções obtidas pelo método das restrições com as soluções obtidas por <i>H-MO</i> ($ESP = 14$)	76
4.5	Valores dos objectivos <i>JIT</i> e <i>PREF</i> , nas soluções eficientes obtidas por <i>H-MO</i> (<i>JIT</i> limitado a 400 000)	77
5.1	Procedimento de arredondamento	87
5.2	À procura do “buraco” mais fundo	89
5.3	Solução obtida: Períodos e linhas em que foram colocadas as OF e ocupação das linhas	97
5.4	Solução obtida: Número de máquinas existentes de cada tipo (K_m) e número de máquinas de cada tipo necessárias em cada período	98
5.5	Solução obtida: Número máximo de postos permitido em cada linha (D_l) e número de postos necessários em cada período e em cada linha	98
6.1	Solução óptima do exemplo	108
6.2	Solução da relaxação linear do exemplo	112
6.3	Algoritmo de planos de corte	114
6.4	Instância PAP1 - evolução do valor da função objectivo com introdução de desigualdades válidas	115
7.1	Esquema de arrefecimento	131
7.2	PAP1 - Determinação de $TipSolIni$, $TipViz$, $dist$ e $TamSubViz$	139
7.3	PAP2 - Determinação de $TipSolIni$, $TipViz$, $dist$ e $TamSubViz$	140
7.4	PAP3 - Determinação de $TipSolIni$, $TipViz$, $dist$ e $TamSubViz$	141
7.5	PAP4 - Determinação de $TipSolIni$, $TipViz$, $dist$ e $TamSubViz$	142
7.6	PAP5 - Determinação de $TipSolIni$, $TipViz$, $dist$ e $TamSubViz$	143
7.7	PAP1 - Determinação de p em $\mathcal{N}_{Mista(dist,p)}$	144
7.8	PAP2 - Determinação de p em $\mathcal{N}_{Mista(dist,p)}$	145
7.9	PAP3 - Determinação de p em $\mathcal{N}_{Mista(dist,p)}$	145
7.10	PAP4 - Determinação de p em $\mathcal{N}_{Mista(dist,p)}$	146
7.11	PAP5 - Determinação de p em $\mathcal{N}_{Mista(dist,p)}$	146
7.12	PAP1 - SA, determinação de T_0	148
7.13	PAP2 - SA, determinação de T_0	148

7.14	PAP3 - SA, determinação de T_0	149
7.15	PAP4 - SA, determinação de T_0	149
7.16	PAP5 - SA, determinação de T_0	150
7.17	PAP1 - SA, determinação de J	151
7.18	PAP2 - SA, determinação de J	151
7.19	PAP3 - SA, determinação de J	152
7.20	PAP4 - SA, determinação de J	152
7.21	PAP5 - SA, determinação de J	153
7.22	PAP1 - SA, determinação de L e de α	154
7.23	PAP2 - SA, determinação de L e de α	154
7.24	PAP3 - SA, determinação de L e de α	155
7.25	PAP4 - SA, determinação de L e de α	155
7.26	PAP5 - SA, determinação de L e de α	156
7.27	PAP1: Comparação de desempenho entre algoritmos	158
7.28	PAP2: Comparação de desempenho entre algoritmos	159
7.29	PAP3: Comparação de desempenho entre algoritmos	159
7.30	PAP4: Comparação de desempenho entre algoritmos	160
7.31	PAP5: Comparação de desempenho entre algoritmos	160
7.32	PAP1: Comparação de desempenho entre algoritmos	162
7.33	PAP2: Comparação de desempenho entre algoritmos	162
7.34	PAP3: Comparação de desempenho entre algoritmos	163
7.35	PAP4: Comparação de desempenho entre algoritmos	163
7.36	PAP5: Comparação de desempenho entre algoritmos	164
C.1	Períodos e linhas em que foram colocadas as encomendas e ocupação das linhas	214
D.1	Solução obtida: Períodos e linhas em que foram colocadas as OF e ocupação das linhas	228
D.2	Solução obtida: Número de máquinas existentes de cada tipo (K_m) e número de máquinas de cada tipo necessárias em cada período	228

D.3	Solução obtida: Número máximo de postos permitido em cada linha (D_l) e número de postos necessários em cada período e em cada linha	229
F.1	Função densidade de probabilidade da distribuição $\Gamma(m, n)$, para diversos valores dos parâmetros m e n	236
G.1	Função objectivo genérica	251

Lista de Tabelas

3.1	Matriz com capacidades de produção por tipo de produto j , para um período t	54
3.2	Matriz de índices de preferência de cada tipo de produto j para cada linha l , para um dado período t	54
3.3	Calendário de Produção	55
3.4	Índices de preferência, P_{jlt} e capacidades de produção, C_{jlt} , das linhas l por tipo de produto j , para todos os períodos t e máxima produção possível do produto tipo j em todas as linhas no período t , U_{jt}	57
3.5	Lista de encomendas: tipo de produto (τ_e), “release date” (r_e), data de entrega (d_e) e quantidade (q_e)	57
4.1	Características da instância PDP1	71
4.2	Solução de PDP1 usando a heurística H -PDP	72
4.3	Valores dos objectivos nas soluções eficientes obtidas por H -MO	74
4.4	Valores dos objectivos nas soluções eficientes obtidas pelo método das restrições	75
5.1	Matriz com capacidades de produção por tipo de produto j , para um período t	93
5.2	Matriz de índices de preferência de cada tipo de produto j para cada linha l , para um dado período t	94
5.3	Calendário de Produção	94
5.4	Lista de OF: tipo de produto (τ_e), “release date” (r_e), data de entrega (d_e) e quantidade (q_e)	95

5.5	Índices de preferência, P_{jlt} e capacidades de produção, C_{jlt} , das linhas l por tipo de produto j , para todos os períodos t e máxima produção possível do produto tipo j em todas as linhas no período t , U_{jt}	96
5.6	Gamas operatórias: Carga do tipo de produto j na máquina m (em minutos por unidade de produto)	96
5.7	Quantidade de máquinas existentes de cada tipo (K_m)	96
5.8	Número máximo de postos em cada linha	97
6.1	Capacidade disponível, C_t em cada período t	108
6.2	Encomendas e , com quantidades q_e e datas de entrega d_e	108
6.3	Resultado da relaxação linear	112
6.4	Características das instâncias PAP	115
6.5	Instâncias PAP, soluções antes e depois da introdução das desigualdades válidas	116
7.1	Resumo dos resultados obtidos para parâmetros meta-heurísticas	156
7.2	Testes realizados com Pesquisa Tabu	157
7.3	Testes realizados com “Simulated Annealing”	158
7.4	Comparação entre as melhores soluções obtidas com meta-heurísticas e as soluções óptimas	164
B.1	Matriz $LinOrd_{jtn}$	200
B.2	Matriz $TipProdOrd_{ltn}$	201
C.1	Características gerais da instância	205
C.2	Fábrica a que pertence cada uma das linhas l	205
C.3	Índices de preferência, P_{jlt} e capacidades de produção, C_{jlt} , das linhas l por tipo de produto j , para todos os períodos t e máxima produção possível do produto tipo j em todas as linhas no período t , U_{jt}	206
C.4	Lista de encomendas: tipo de produto (τ_e), “release date” (r_e), data de entrega (d_e) e quantidade (q_e)	206
C.5	Valores iniciais de $CapDispLin_{jlt}$ e $CapDisp_{jt}$	207
C.6	Primeiros e últimos períodos onde existe capacidade disponível	207

C.7	Matriz $LinOrd_{jtn}$	207
C.8	Valores da capacidade disponível, primeiro período disponível e último período disponível.	209
C.9	Valores da capacidade disponível, primeiro período disponível e último período disponível.	211
C.10	Valores da capacidade disponível, primeiro período disponível e último período disponível.	212
C.11	Valores da capacidade disponível, primeiro período disponível e último período disponível.	213
D.1	Características gerais da instância	215
D.2	Índices de preferência, P_{jlt} e capacidades de produção, C_{jlt} , das linhas l por tipo de produto j , para todos os períodos t e máxima produção possível do produto tipo j em todas as linhas no período t , U_{jt}	216
D.3	Gamas operatórias para cada tipo de produto: Carga do tipo de produto j na máquina m (em minutos por unidade)	216
D.4	Quantidade de máquinas existentes de cada tipo (K_m)	216
D.5	Número máximo de postos em cada linha	217
D.6	Lista de OF: tipo de produto (τ_e), “release date” (r_e), data de entrega (d_e) e quantidade (q_e)	217
D.7	Valores iniciais de $CapDispLin_{jlt}$ e $CapDisp_{jt}$	217
D.8	Primeiros e últimos períodos onde existe capacidade disponível	218
D.9	Matriz $TipProdOrd_{ltn}$, igual para todos os períodos t	218
D.10	$CapDisp$, $PrimPerDisp$, $UltimPerDisp$	220
D.11	$CapDisp$, $PrimPerDisp$, $UltimPerDisp$	221
D.12	$CapDisp$, $PrimPerDisp$, $UltimPerDisp$	222
D.13	$CapDisp$, $PrimPerDisp$, $UltimPerDisp$	223
D.14	$CapDisp$, $PrimPerDisp$, $UltimPerDisp$	224
D.15	Carga para cada linha l , tipo de produto j e período t	224
D.16	Número de máquinas necessárias em $t = 1$	225
D.17	Número de máquinas necessárias em $t = 2$	225
D.18	Número de máquinas necessárias em $t = 3$	226

D.19	Número de máquinas existentes de cada tipo (K_m) e número de máquinas de cada tipo, necessárias em cada período	227
D.20	Número máximo de postos permitido em cada linha (D_l) e número de postos necessários em cada período e em cada linha	227
E.1	Índices de preferência	234
E.2	Novos índices de preferência	234
F.1	Parâmetros introduzidos no gerador para gerar PDP1	238
F.2	Características da instância PDP1	238
F.3	Fábrica a que pertence cada uma das linhas l	238
F.4	Número de dias de cada período t	239
F.5	Índices de preferência e capacidades de produção diárias por tipo de produto j , no período 1 e no período 2	240
F.6	Índices de preferência e capacidades de produção diárias por tipo de produto j , nos períodos 3 e 4 e do período 5 ao período 13	241
F.7	Índices de preferência e capacidades de produção diárias por tipo de produto j , no período 14	242
F.8	Lista de encomendas	242
F.9	Parâmetros introduzidos no gerador para gerar instâncias PAP	244
F.10	Características das instâncias PAP	244
F.11	PAP1 - Capacidade disponível, C_t em cada período t	245
F.12	PAP1 - Encomendas e , com respectivos períodos de entrega propostos d_e	245
F.13	PAP2 - Capacidade disponível, C_t em cada período t	246
F.14	PAP2 - Encomendas e , com respectivos períodos de entrega propostos d_e	246
F.15	PAP3 - Capacidade disponível, C_t em cada período t	247
F.16	PAP3 - Encomendas e , com respectivos períodos de entrega propostos d_e	247
F.17	PAP4 - Capacidade disponível, C_t em cada período t	248
F.18	PAP4 - Encomendas e , com respectivos períodos de entrega propostos d_e	248

F.19 PAP5 - Capacidade disponível, C_t em cada período t	249
F.20 PAP5 - Encomendas e , com respectivos períodos de entrega propostos d_e	249

Lista de Algoritmos

3.1	Descrição genérica de $H-PDP$	56
4.1	Método paramétrico	67
4.2	Método das restrições	68
4.3	Heurística $H-JIT$	69
4.4	Heurística $H-PREF$	69
4.5	Heurística multiobjectivo ($H-MO$)	70
7.1	Algoritmo de Pesquisa Local	121
7.2	Heurística Básica para geração de uma solução inicial	122
7.3	Heurística Aleatória para geração de uma solução inicial	122
7.4	Heurística MelhoraSolução	123
7.5	Obtenção de $\bar{x} \in \mathcal{N}_{Desloca(dist)}(x_n)$	125
7.6	Obtenção de $\bar{x} \in \mathcal{N}_{Troca(dist)}(x_n)$	126
7.7	Procedimento seguido na vizinhança $\mathcal{N}_{Mista(dist,p)}(x_n)$	127
7.8	Algoritmo de Pesquisa Tabu	130
7.9	Algoritmo de “Simulated Annealing”	132
7.10	Estrutura geral do programa para fazer testes com as meta-heurísticas	134
B.1	Algoritmo ColocaMaisCedo	194
B.2	Algoritmo ColocaMaisTarde	195
B.3	Algoritmo ColocaPROG	196
B.4	Algoritmo RelaxaLimites	202

Capítulo 1

Introdução

1.1 Motivação

O trabalho apresentado nesta dissertação teve origem num projecto apoiado pelo PEDIP (Programa 5 - Missões de Produtividade / Acções de Demonstração), em que os principais parceiros foram o INESC - Instituto de Engenharia de Sistemas e de Computadores (grupo Produção e Logística, ao qual a autora pertence) e o grupo industrial Growela, do sector do calçado. O projecto PEDIP teve o seu início em Setembro de 1991 e terminou em Julho de 1993 com a realização de uma sessão pública de divulgação de resultados, tendo a actividade de investigação iniciada na altura sido prosseguida ao longo dos anos seguintes.

O INESC é uma associação privada sem fins lucrativos e de utilidade pública, abrangendo um vasto conjunto de actividades que vão da investigação científica fundamental à incubação de projectos empresariais, passando pelo desenvolvimento científico e tecnológico, transferência de tecnologia e formação tecnológica avançada em Tecnologias da Informação e Telecomunicações. Criado em 1980, o INESC constituiu-se como instituição pioneira no “interface” entre o sistema universitário público e o tecido empresarial.

O grupo Growela Portuguesa Calçado é constituído por três unidades fabris distintas, a Growela (empresa mãe), a Fundador e a Coca. O grupo pertence ao sector do calçado, que por sua vez pertence ao universo das

indústrias sazonais, de moda e de produção por encomenda. A Fundador e a Coca foram adquiridas recentemente pelo grupo, mas mantiveram a sua carteira de clientes e a sua especificidade quanto aos tipos de produtos a produzir.

Nas últimas décadas, os reduzidos custos da mão-de-obra nacional, associados à nossa proximidade dos fortes mercados da Europa Central constituíam uma inegável vantagem da indústria portuguesa. Mais recentemente, o aumento da penetração no mercado europeu de produtos provenientes do extremo oriente, a preços muito baixos, tem afectado a grande maioria das empresas. Por outro lado, a evolução verificada nos últimos anos em processos automáticos de planeamento, gestão e controlo da produção tem provocado alterações importantes no equilíbrio industrial.

De um modo geral, a atitude dos industriais no sentido de ultrapassar a crise, tem passado pela introdução de melhorias em produtividade e qualidade e pela redução e cumprimento dos prazos de entrega. Em termos comerciais, tem sido feita uma grande aposta no “design”, por forma a conseguir produtos mais atractivos, disponibilizar uma gama de produtos mais alargada e propor preços mais baixos.

Nas indústrias de moda o “cliente-tipo” está em mutação. Cada vez mais exige produtos exclusivos e de elevada qualidade, o que implica necessariamente que faça encomendas de pequenas quantidades e que os produtos tenham preços mais elevados. Para além disso, o cliente não deseja armazenar produtos e, para garantir a venda dos seus produtos, coloca as encomendas exigindo um prazo de entrega cada vez mais curto, por forma a poder prever o melhor possível o resultado do seu investimento.

As exigências de cumprimento de prazos de entrega cada vez mais curtos criou nos sectores comerciais das empresas a necessidade de dispor de ferramentas de apoio na confirmação de datas de entrega que efectuassem em tempo real cálculos sobre capacidades de produção. Por outro lado, e devido ao grande número de pequenas encomendas, nos sectores de planeamento dessas empresas, as tarefas de afectação de encomendas a linhas de produção e de determinação de datas de entrada em produção tornaram-se cada vez mais complexas, exigindo cada vez mais o apoio de um sistema informático.

Todos esses factores contribuíram para que, de uma forma natural, se gerassem contactos entre a Growela e o INESC, com o intuito de iniciar um projecto conjunto para o desenvolvimento de uma ferramenta informática que apoiasse o planeamento da produção das três unidades fabris do grupo. Dadas as características e a complexidade do problema, optou-se pelo desenvolvimento de um sistema informático com a filosofia e as características de um Sistema de Apoio à Decisão (SAD).

O desenvolvimento do SAD foi realizado ao longo do projecto PEDIP e envolveu a colaboração de vários elementos do grupo de Produção e Logística do INESC. Essa colaboração estendeu-se pelo levantamento das necessidades da empresa, pela definição dos problemas e sua estruturação, pela coordenação do desenvolvimento do produto, pela definição das bases de dados, pela geração de interfaces e pelo desenvolvimento de algoritmos para o planeamento da produção.

1.2 Objectivos da investigação

O primeiro objectivo do trabalho de investigação descrito nesta dissertação, foi colaborar na concepção e no desenvolvimento do SAD a que se fez referência na secção anterior. Para tal, foi necessário um conhecimento aprofundado dos processos de tomada de decisões e, como segundo objectivo da investigação, proceder a uma estruturação hierárquica desses processos. Uma tal estruturação implicou a definição dos vários horizontes de decisão e das interligações existentes entre os níveis de decisão.

Como terceiro objectivo deste trabalho foi necessário definir, estruturar e modelizar os problemas levantados para cada um dos níveis de decisão, desenvolver algoritmos eficientes para a sua resolução e incorporá-los no SAD.

O facto de que os problemas levantados para cada nível têm associados critérios múltiplos e habitualmente conflituosos, levou ao estudo do carácter Multiobjectivo de um problema associado a um dos níveis de decisão, e à implementação de uma heurística multiobjectivo com a qual fosse possível gerar um conjunto de soluções eficientes para o problema.

O estudo e a avaliação das potencialidades de algumas das técnicas mais

promissoras da Investigação Operacional actual, tais como as abordagens de Programação Inteira e Desigualdades Válidas por um lado, e as Meta-heurísticas por outro, constituíram o último objectivo do trabalho de investigação realizado no âmbito desta dissertação.

1.3 Estrutura da dissertação

Neste capítulo introdutório, começou-se por apresentar a motivação e os objectivos para o trabalho realizado, e a seguir será enquadrado o problema a tratar, fazendo-se uma breve descrição dos diversos conceitos aprofundados posteriormente ao longo da dissertação, tais como Planeamento Hierárquico, tomada de Decisões Multiobjectivo, Sistemas de Apoio à Decisão (SAD) e modelos de Optimização Combinatória específicos para problemas de planeamento da produção. Para referências complementares, apresenta-se bibliografia associada a cada um destes temas.

No Capítulo 2, é feita uma descrição detalhada do Caso de Estudo. Começa-se por descrever o processo de fabrico de calçado, a hierarquia dos processos de tomada de decisões e o SAD. É apresentado o carácter hierárquico do processo de tomada de decisões, reflectido no SAD, seguindo-se uma descrição de cada um dos componentes do SAD.

No Capítulo 3, descreve-se em pormenor o nível mais alto do processo de decisão considerado no SAD e designado por Plano Director da Produção (PDP). Começa-se por apresentar o modelo de Programação Matemática desenvolvido para o PDP, e em seguida uma heurística original desenvolvida para o Caso de Estudo (*H-PDP*). Apresenta-se também um exemplo de aplicação dessa heurística. Por último é feita uma avaliação, a partir das reacções dos Agentes de Decisão, aos resultados fornecidos pela heurística *H-PDP*, para problemas reais do Caso de Estudo.

O carácter multiobjectivo do problema PDP é tratado no Capítulo 4, sendo descrita uma heurística multiobjectivo desenvolvida especialmente para este caso. São em seguida comparados os resultados da heurística com os resultados obtidos por métodos analíticos específicos da Decisão Multi-critério, utilizando-se para esse efeito uma instância gerada aleatoriamente.

No Capítulo 5 é descrito em pormenor o segundo nível de planeamento,

designado por Programação (PROG). Começa-se por apresentar o modelo de Programação Matemática desenvolvido para o PROG, e em seguida faz-se uma descrição de uma primeira heurística desenvolvida para esse problema, baseada em programação linear, e a que se chamou *H-PROG1*. Descreve-se a seguir uma segunda heurística desenvolvida para o PROG, a que se chamou *H-PROG*. Apresenta-se também um exemplo ilustrativo resolvido por esta última heurística, assim como as opiniões dos Agentes de Decisão, face aos resultados fornecidos pela heurística para problemas reais do Caso de Estudo.

Considerou-se interessante explorar uma generalização do problema PDP, a que se chamou Planeamento Agregado da Produção (PAP), e o objectivo dos Capítulos 6 e 7 é apresentar algumas abordagens promissoras desenvolvidas para este problema, comparando os resultados respectivos. Uma primeira abordagem baseia-se na exploração de propriedades estruturais do problema, com vista à obtenção de desigualdades válidas para o modelo proposto, que permitam calcular minorantes (“lower bounds”) sobre o valor da solução óptima. Estes minorantes, integrados em métodos de resolução do tipo “branch-and-bound”, aumentam a eficiência da resolução do problema de optimização. A segunda abordagem genérica, baseia-se na utilização de meta-heurísticas baseadas na aleatorização de procedimentos de Pesquisa Local (“Pesquisa Local Aleatorizada”, Pesquisa Tabu, “Simulated Annealing”), que permitem a obtenção eficiente de soluções satisfatórias, ou seja de majorantes (“upper bounds”) sobre o valor da solução óptima. A comparação das diferentes abordagens e dos algoritmos resultantes foi feita com instâncias do problema PAP geradas aleatoriamente.

No Capítulo 8 faz-se um resumo dos diversos pontos abordados na dissertação, sintetizam-se as principais conclusões do trabalho realizado, enunciam-se problemas deixados em aberto e perspectivam-se trabalhos futuros.

1.4 Enquadramento

No âmbito do projecto PEDIP que deu origem a este trabalho, e com vista a determinar, num nível geral, as funcionalidades que deveriam estar contidas na ferramenta informática a desenvolver, foi necessário efectuar um

levantamento, fundamentado ao longo de múltiplas visitas e reuniões com responsáveis de sector, aos principais problemas existentes no planeamento da produção da empresa.

Com este levantamento, concluiu-se que o planeamento era realizado em vários níveis, em instantes diferentes e com níveis diferentes de agregação da informação, existindo, para cada um deles, vários Agentes de Decisão (AD). Esta conclusão levou à opção por uma abordagem hierárquica na estruturação e resolução dos problemas de decisão em causa. A multiplicidade de objectivos em cada nível (e o carácter não quantitativo de alguns deles) conduz na prática a que cada AD lhes atribua, em cada momento, uma diferente importância relativa. Essa situação e a complexidade que lhe está associada levou-nos a optar por um Sistema de Apoio à Decisão (SAD).

Após a identificação dos problemas de decisão a considerar nos diferentes níveis, modelizaram-se os processos de decisão tendo em consideração as interacções entre os níveis, os objectivos associados a cada nível e o tipo de agregação de informação. Seguidamente, desenvolveram-se algoritmos para cada um dos níveis de planeamento. Esses algoritmos são parametrizáveis, permitindo obter diferentes soluções para cada instância do problema e permitindo ao AD escolher em cada altura a solução que lhe pareça mais interessante.

Para apoiar a resolução do problema, é utilizado um SAD, com “interfaces” gráficas que facilitam a comparação de cenários alternativos. O SAD serve de estrutura base e de ligação entre os diversos níveis de planeamento, e entre estes e o AD.

Apresenta-se a seguir uma breve descrição dos conceitos mais importantes que serão utilizados ao longo desta dissertação, nomeadamente os que estão associados ao Planeamento Hierárquico, à tomada de Decisões Multicritério, aos Sistemas de Apoio à Decisão e aos modelos de Optimização Combinatória para o Planeamento da Produção.

1.4.1 Planeamento Hierárquico

O planeamento da produção envolve decisões complexas, com objectivos múltiplos e muitas vezes conflituosos, tomadas a diferentes níveis dentro da

hierarquia da empresa e com implicações em diferentes horizontes de planeamento. A aproximação aos problemas de planeamento da produção por um único modelo nunca foi possível na prática, devido ao facto de as decisões envolverem objectivos múltiplos e dado que esses modelos exigiriam a existência e manutenção de grandes quantidades de informação. Além disso, um modelo monolítico dificilmente traduziria as características hierárquicas da organização e impediria o envolvimento, desejável, dos gestores nos vários níveis do processo de decisão.

Com o planeamento hierárquico, pretende-se ultrapassar esses problemas, reduzindo a sua complexidade, através da divisão do problema global em problemas parciais entre os quais existam ligações uni ou bidireccionais. É então possível reduzir a quantidade de dados necessária em cada nível, recorrendo à agregação de variáveis (isto é, à redução “explícita” das diferentes linhas de acção disponíveis). Em [Boskma, 1982], referem-se três tipos de agregação possíveis para as variáveis:

- agregação de produtos;
- agregação de recursos;
- agregação do tempo.

O desenvolvimento de sistemas hierárquicos de planeamento da produção iniciou-se com os trabalhos publicados por Bitran, Meal e Hax nos anos 70, [Hax and Meal, 1975] [Bitran and Hax, 1977].

No artigo [Axsäter, 1981], é focado o problema de agregação e desagregação de variáveis no planeamento hierárquico, determinando-se as condições em que um plano agregado pode ser desagregado de forma consistente. Em [Erschler et al., 1986], Erschler, Fontan e Merce focam mais uma vez o problema da consistência no processo de desagregação no planeamento hierárquico. Em [Ari and Axsäter, 1988], é focado de novo o problema da desagregação de variáveis num sistema de planeamento hierárquico, com dois níveis e procura estocástica independente.

Nos trabalhos publicados por Bitran, Hax e Haas no início da década de 80, [Bitran et al., 1981] e [Bitran et al., 1982], é desenvolvida uma metodologia para tratamento de problemas de planeamento hierárquico da

produção, respectivamente para sistemas com um só nível e sistemas com dois níveis. Estes trabalhos baseiam-se na exploração da estrutura dos produtos, agregando “itens de produtos” em “tipos de produtos” e por sua vez “tipos de produtos” em “famílias de produtos”, que já tinha sido abordada em [Hax and Meal, 1975] e em [Bitran and Hax, 1977]. O problema de planeamento e escalonamento da produção é dividido numa hierarquia de sub-problemas, que são resolvidos sequencialmente em cada período de planeamento.

Graves apresentou em [Graves, 1982] um procedimento para decomposição de um modelo de programação mista associado a um problema global de planeamento da produção. O procedimento consiste em decompor o problema de planeamento da produção em dois subproblemas. O mecanismo de interligação entre os problemas, baseia-se na consistência entre os respectivos níveis de inventário, e é implementado com recurso a multiplicadores de Lagrange.

Em 1982, foi publicado o artigo [Gelders and Van Wassenhove, 1982] onde são discutidos os problemas de projecto de sistemas hierárquicos de planeamento da produção com dois níveis hierárquicos, a médio e a curto prazo e suas aplicações industriais. Nesse trabalho é referido o problema das relações entre os diversos níveis [Gelders and Van Wassenhove, 1982, pág.27]: “decisões tomadas nos níveis superiores impõem restrições às acções dos níveis inferiores, enquanto que as decisões dos níveis inferiores fornecem ‘feedback’ para avaliar as acções dos níveis superiores”. Os autores apontam para a necessidade de integrar os diferentes níveis, com vista a tornar os planos realizáveis e os objectivos consistentes.

Como é natural, ao longo dos anos surgiram numerosos artigos com aplicações práticas de planeamento hierárquico. São exemplo disso os artigos [Gelders and Van Steeland, 1980] e [Gelders and Van Wassenhove, 1982], onde é apresentado o problema de planeamento da produção numa laminação e numa empresa de produtos químicos. Em [Axsäter and Jönsson, 1984], é tratado teoricamente o problema da agregação e desagregação no caso do planeamento hierárquico da produção numa empresa que produz máquinas para furar pedra. Em [Van Wassenhove and Vanderhenst, 1983] é apresentado o caso de uma outra empresa de produtos químicos, onde é feito o

planeamento hierárquico de um departamento “bottleneck”.

Em [Günther, 1986] o caso de estudo baseia-se numa empresa química de produção de detergentes. O problema de decisão global é dividido em 4 níveis que correspondem a um planeamento agregado, a um escalonamento e sequenciamento detalhados, à determinação das ordens de fabrico e à distribuição e despacho, e são apresentados procedimentos heurísticos para obter soluções para cada um dos níveis.

Lin e Moodie, apresentam no seu artigo [Lin and Moodie, 1989], o caso de uma empresa produtora de aço, cujo planeamento global foi dividido em três níveis. Um nível que corresponde a um escalonamento de alto nível, ao qual os autores chamaram MPS (“Master Production Scheduling”), um outro que corresponde a um planeamento das necessidades da produção e um terceiro nível, que corresponde a um escalonamento e sequenciamento a baixo nível, PPS (“Production Sequencing and Scheduling”). O objectivo é a satisfação das encomendas dos clientes com o menor custo possível de energia.

Há relativamente pouco tempo, alguns autores escreviam que não existia um quadro teórico geral para o planeamento hierárquico. Mais recentemente, trabalhos de Schneeweiß apresentam um quadro teórico geral para o planeamento hierárquico, onde são tidas em conta as interdependências entre os vários níveis de planeamento, [Schneeweiß, 1994] [Schneeweiß, 1995]. Citando [Schneeweiß, 1994], “o planeamento hierárquico é a geração de hierarquias de planeamento e a concepção de sequências de planeamento dentro dessas hierarquias”. No entanto, e continuando a citar [Schneeweiß, 1994], “ainda não foi abordada a questão complexa de como gerar a hierarquização óptima de um problema, nem de qual o tipo de planeamento a realizar em cada um dos níveis”.

Refere-se ainda o artigo [Meal et al., 1987], onde é descrito o acoplamento de um sistema MRP com um sistema hierárquico de planeamento, no caso de uma empresa produtora de periféricos para computador, realçando que a abordagem hierárquica a um problema de planeamento da produção não é antagónica, mas sim complementar, a uma abordagem do tipo MRP. De facto, o planeamento hierárquico pode abranger todos os níveis de planeamento, desde o nível estratégico até ao nível operacional, enquanto que o

MRP pode ser encarado como um componente do planeamento hierárquico, num nível intermédio, que permite realizar cálculos para determinação das necessidades de materiais e das cargas das máquinas.

Por último, recomenda-se a consulta de [Bukh, 1994], onde é feita uma descrição extensa da bibliografia sobre Planeamento Hierárquico da Produção.

1.4.2 Carácter multicritério do problema

Os problemas de decisão que surgem na prática, e em particular os que ocorrem no âmbito do planeamento da produção, têm normalmente associados objectivos múltiplos que, na maioria dos casos, são conflituosos. Em qualquer um dos níveis hierárquicos associados a um problema de planeamento da produção existem vários objectivos, alguns só dificilmente quantificáveis e outros mesmo difíceis de explicitar, e que o Agente de Decisão (AD) pretende ver satisfeitos. No entanto normalmente o AD não sabe definir à priori precedências ou relações de troca entre esses objectivos.

Cabe no entanto ao analista não se deixar tentar por abordagens que passem por “pesar” os vários objectivos, que pressupõem, por um lado, que o AD é capaz de fornecer esses “pesos” e, por outro lado, que os “pesos” reflectem fielmente as suas preferências. Esse tipo de métodos torna-se também pouco transparente, dado que as verdadeiras decisões são tomadas previamente, e a “solução” aparece automaticamente e de forma indiscutível, como resultado de um conjunto de cálculos mais ou menos obscuros para o AD.

Por outro lado, nas abordagens em que todos os objectivos excepto um são transformados em restrições, e em que é feita uma optimização relativamente ao objectivo “livre”, é discutível qual o objectivo a deixar “livre” e é também discutível qual o nível de satisfação a fixar para os outros objectivos. Nesta situação, a “solução” surge também como resultado de cálculos mais ou menos obscuros, e finalmente ao AD pode restar apenas a hipótese de a rejeitar.

As reflexões apresentadas anteriormente ainda se tornam mais pertinentes se estivermos perante mais do que um AD que, mesmo que tenham interesses globais comuns, têm certamente entendimentos diferentes do que são esses interesses comuns.

Formalizando um pouco a linguagem utilizada nesta dissertação no âmbito dos Problemas de Decisão Multicritério, a denominação genérica de Multicritério abrange dois tipos de problemas, os problemas Multiobjectivo e os problemas Multiatributo. Um problema de decisão com objectivos múltiplos, ou Multiobjectivo, caracteriza-se pela existência de um conjunto de soluções admissíveis definidas através de um conjunto de restrições e onde os objectivos são explicitados através de funções objectivo. Em oposição, designam-se por problemas Multiatributo aqueles em que o número de alternativas é limitado e os atributos são conhecidos explicitamente.

Aos problemas de planeamento da produção tratados ao longo deste trabalho, estão associados problemas de decisão Multiobjectivo. Cabe ao analista, com o apoio do AD, construir um sistema que apoie realmente o AD, fornecendo-lhe, não uma única solução, seja ela “óptima” ou apenas “boa”, mas um conjunto de soluções não dominadas ou eficientes (ver definição na secção 4.2), das quais ele possa escolher a que melhor satisfaz os objectivos, tal como ele os “percebe” no momento (parte-se do princípio que não é aceitável, em nenhuma circunstância, que o AD prefira uma solução dominada).

As metodologias desenvolvidas, para obtenção do conjunto de soluções não dominadas ou eficientes em problemas Multiobjectivo, designam-se por métodos de geração. Estes métodos não utilizam nenhuma informação quanto às preferências do AD, limitando-se a obter o conjunto de soluções eficientes relativamente aos objectivos previamente definidos pelo AD. Incluem-se nesses métodos de geração o método paramétrico e o método das restrições, descritos na secção 4.4 (recomenda-se a consulta de [Matos, 1988], de [Chankong and Haimes, 1983] e de [Ulungu and Teghem, 1994], para uma descrição detalhada desses métodos).

Dado que tanto o método paramétrico como o método das restrições se baseiam na repetição da resolução analítica de cada instância dos problemas, e dado que os problemas de planeamento da produção abordados nesta dissertação, em instâncias reais (de dimensão elevada), não podem ser resolvidos analiticamente em tempo útil, procura-se neste trabalho obter apenas um subconjunto do conjunto das soluções, recorrendo a uma heurística multiobjectivo.

Após a obtenção do conjunto de soluções eficientes, é necessário apoiar o AD no processo de avaliação de soluções, dado que teoricamente o número de soluções eficientes para um problema de decisão Multiobjectivo é infinito (e, em termos práticos, será sempre elevado). Além disso, é reconhecido que há um limite para o número de alternativas diferentes que o AD pode avaliar em simultâneo (um número inferior a 10, segundo alguns autores). Assim, é importante por um lado manter uma visão global das alternativas para que a percepção do problema pelo AD aumente, mas, por outro lado, é necessário fornecer ao AD elementos que lhe permitam reunir soluções em subconjuntos com características semelhantes, de modo a reduzir o número de alternativas a analisar, chegando no fim desse processo à solução considerada no momento como a melhor.

Recomenda-se a leitura de [Matos, 1988, pp 6-62], para uma “síntese do estado do conhecimento” nesta matéria. Nessa mesma referência é também possível encontrar uma extensa bibliografia sobre o assunto.

1.4.3 Sistema de Apoio à Decisão (SAD)

Em todas as organizações, e em particular nas empresas industriais, as decisões são tomadas em ambientes cada vez mais complexos e dinâmicos. A avalanche de informação a que se tem acesso hoje em dia, associada a uma enorme quantidade de tecnologias emergentes, aumentou, quase sem limites, as linhas de acção alternativas. Simultaneamente, o aumento da concorrência conduz a uma maior incerteza quanto ao futuro e faz com que erros na tomada de decisões possam ter consequências muito graves para as empresas. Os Agentes de Decisão devem então ter acesso a ferramentas que lhes possibilitem a análise de diferentes cenários, a partir dos quais poderão reconhecer oportunidades e ameaças para as suas empresas. A informação de que necessitam para traçar esses cenários, pode em parte ser disponibilizada através de modelos mais ou menos sofisticados da realidade. Os sistemas de informação tradicionais, tais como sistemas de gestão de bases de dados e sistemas convencionais de gestão de informação, não apoiam os AD na sua necessidade de informação e normalmente não possuem as capacidades de modelização necessárias para lidar com realidades complexas e dinâmicas.

Essas necessidades podem explicar a crescente utilização de Sistemas de Apoio à Decisão (SAD). Os SAD fornecem capacidades de modelização de situações reais, de tal modo que os problemas possam ser abordados pela análise de cenários alternativos e que possam ser determinadas acções com um impacto positivo no desempenho das organizações.

Segundo a definição de Keen e Scott Morton [Keen and Morton, 1978], considerados por muitos como os pais do movimento SAD: “um SAD é um sistema coerente, com tecnologia baseada em sistemas informáticos, utilizado pelos gestores para a tomada de decisões em problemas pouco ou nada estruturados, com o objectivo de apoiar e não de substituir as decisões do gestor, aumentando mais a eficácia do que a eficiência da tomada de decisões”.

Os SAD são constituídos basicamente por:

- um gestor de dados;
- um gestor de modelos;
- um gestor de diálogos com o AD.

Esta estrutura genérica está representada na Figura 1.1, onde se evidenciam também as ligações a bases de dados e sistemas informáticos exteriores ao próprio SAD (recomenda-se a leitura de [Turban, 1988], para obtenção de informações gerais sobre SAD e para ter acesso a uma extensa bibliografia sobre o assunto).

A discussão sobre as vantagens e os inconvenientes dos SAD, que esteve focada na integração de todos os seus componentes, passou actualmente a centrar-se no facto de ser permitido e aconselhável o seu uso ao longo das várias fases do processo de decisão. As fases do processo de decisão não podem de modo algum ser tratadas independentemente, e não se deve esquecer que a sua envolvente muda ao longo do tempo, o que implica uma alteração do próprio problema. Um sistema que apoie a decisão deve-se adaptar a essas circunstâncias e também à evolução do conhecimento por parte do AD. O desenvolvimento de SAD com características modulares, permite a integração de novas Bases de Dados, novos Interfaces e novos Modelos e Algoritmos, facilitando uma evolução incremental e permitindo

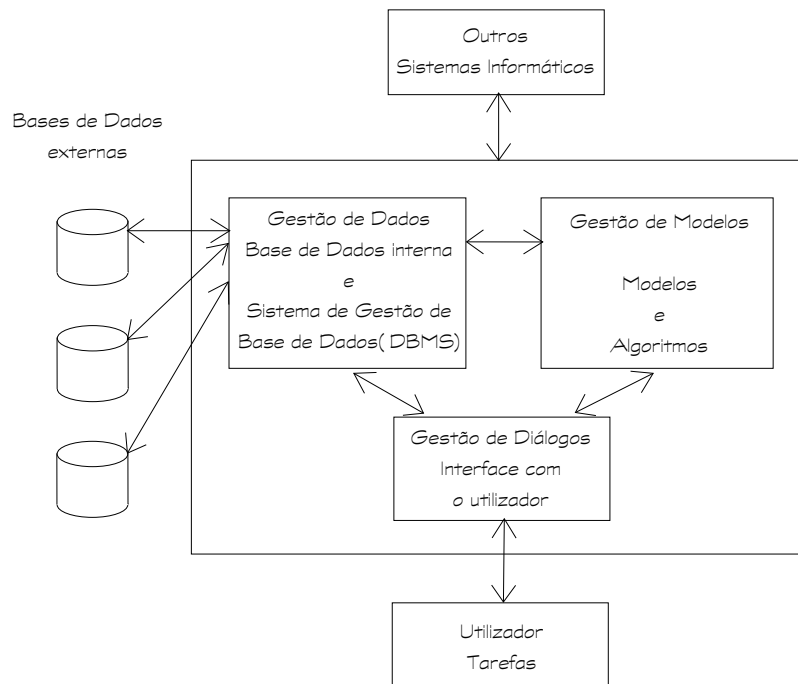


Figura 1.1: Modelo conceptual de um SAD

assim acompanhar o conhecimento “progressivo” dos problemas de decisão (pouco estruturados).

Ao longo dos anos foram surgindo aplicações de SAD às mais diversas situações práticas, cujas concretizações se encontram descritas em numerosos artigos e livros. Em [Eom and Lee, 1990], pode-se encontrar bibliografia muito completa sobre aplicações de SAD entre 1971 e 1988. Ainda relativamente a aplicações de SAD, cita-se especialmente [Kroon, 1990], onde é feita uma descrição de um SAD criado para apoiar o escalonamento de tarefas e planeamento de capacidades no departamento de manutenção de aviões da KLM, no aeroporto de Schiphol em Amsterdão.

O Caso de Estudo, tratado nesta dissertação, com a necessidade de associar os diversos níveis do planeamento hierárquico, e de tratar explicitamente os objectivos de cada um dos níveis, apontava claramente para o recurso a um SAD, que foi projectado especialmente para esta situação e que será descrito em pormenor na secção 2.6.

1.4.4 Modelos de Optimização Combinatória (OC)

Os modelos de Optimização Combinatória (OC) ligados ao Planeamento da Produção têm merecido sempre uma atenção especial por parte da comunidade científica, devido por um lado ao grande número de aplicações práticas existentes e, por outro lado, às dificuldades associadas à sua resolução.

De todos os modelos de OC ligados ao Planeamento da Produção, focar-se-à aqui apenas os modelos de “lot-sizing” e de “scheduling”, que correspondem respectivamente à determinação das quantidades (lotes) de todos os produtos a serem produzidos em cada período de um determinado horizonte temporal (“lot-sizing”), e ao escalonamento das operações num conjunto de recursos da planta fabril (“scheduling”).

Determinação de lotes de produção (“lot-sizing”)

Nos últimos anos foi apresentada na literatura uma grande variedade de modelos de OC para determinação de lotes de produção. Esses modelos têm tipicamente como objectivos a minimização dos custos de manutenção de inventário (“stocks”), a minimização dos tempos de “setup” ou a minimização do número de horas extraordinárias necessário para cumprir o plano de produção. O ênfase destes modelos está em geral no compromisso entre custos de “setup” e custos de manutenção de inventário, pois quando se determina quanto e quando se deve produzir, tem que se ter em conta, em simultâneo, a perda de produtividade devida a “setups” frequentes, se os lotes forem pequenos, e os elevados custos de inventário, se os lotes forem grandes.

Os modelos e algoritmos para determinação de lotes económicos têm em geral sido desenvolvidos ao longo de duas linhas. Uma linha corresponde a considerar capacidade limitada, em que os produtos competem por esse recurso limitado e os custos de “setup” se tornam um factor importante nos custos totais. Esses modelos também tratam o problema do planeamento da produção de vários produtos, com variação da procura ao longo dos períodos do horizonte temporal. Uma outra linha corresponde a considerar capacidade infinita. Esses modelos utilizam a fórmula do lote económico de produção com interacção entre “setups” de produtos individuais e procura

constante durante o horizonte de planeamento. Se houver custos de “setup”, os modelos passam a ser não lineares, podendo ser formulados com variáveis inteiras (binárias, em geral). Uma vantagem destes modelos está na incorporação, neste nível, de questões relacionadas com o escalonamento e com a indivisibilidade dos lotes. No entanto essa característica implica a necessidade de mais informação e mais detalhada, ao longo do horizonte de planeamento. Recomenda-se a leitura de [Kuik et al., 1995], onde é feita uma descrição e um levantamento de referências bibliográficas sobre este tema.

Escalonamento (“scheduling”)

Os problemas de escalonamento atraem há décadas a atenção dos investigadores e são exemplo da extrema dificuldade dos problemas de OC.

Uma forma possível de divisão dos problemas de escalonamento é em problemas de uma máquina ou de mais do que uma máquina. Nos problemas de uma única máquina, pretende-se conhecer a ordem pela qual as tarefas são processadas nessa máquina. Este problema é aplicável apenas a um pequeno número de situações reais (por exemplo no caso em que há uma máquina que é responsável pelo estrangulamento da produção); no entanto pode ser muito difícil de resolver, e serve como base para desenvolvimentos em problemas com mais do que uma máquina.

Uma forma possível de classificação de problemas de mais do que uma máquina é em problemas de máquinas paralelas e em problemas de operações múltiplas (como os problemas de “flow shop” e os problemas de “job shop”), por ordem crescente de complexidade. Os problemas de máquinas paralelas surgem quando existem várias tarefas com apenas uma operação que pode ser realizada em qualquer uma das máquinas. Os problemas de “flow shop” baseiam-se na situação em que existe um conjunto de máquinas, dispostas em linha, e em que todas as tarefas têm que passar por todas as máquinas pela ordem em que estas estão dispostas. Os problemas de “job shop” são os mais complexos, e correspondem à situação em que cada tarefa tem um conjunto de operações a realizar nas máquinas existentes, percorrendo as máquinas por uma ordem diferente. No caso de a ordem das operações das tarefas ser irrelevante, o problema designa-se por “open shop”.

Para cada um dos problemas descritos, o facto de se considerar ou não datas mínimas para entrada em produção ou então o facto de se considerar que as tarefas podem ser interrompidas (“preemption”) ou não, faz com que o grau de dificuldade na resolução dos problemas varie muito. Podem ainda considerar-se inúmeras “complicações” adicionais que “multiplicam” indefinidamente as possibilidades e a complexidade dos problemas: precedências, tempos de “setup”, etc.

Os objectivos que se pretende minimizar podem ser também muito variados, tais como tempo total de processamento de todas as tarefas (“makespan”), tempo de processamento da tarefa mais demorada, número de tarefas em atraso, soma pesada dos avanços relativamente à data de entrega, etc.

Recomenda-se a consulta de [Baker, 1974], [Rinnooy Kan, 1976] e de [Lawler, 1982], para um levantamento exaustivo dos princípios fundamentais e do “estado da arte” neste tipo de problemas.

Interligação entre “lot-sizing” e “scheduling”

Uma importante linha de investigação, que recebe actualmente bastante atenção por parte da comunidade científica, relaciona-se com a interligação dos dois tipo de problemas descritos atrás, os problemas de determinação de lotes económicos e os problemas de escalonamento (para informações sobre este tópico, recomenda-se a leitura de [Dauzère-Pérès and Lasserre, 1994]). Um plano de produção detalhado não é admissível se não prever a produção de tudo o que se pretende ou, por outro lado, se as cargas excederem as capacidades. Assim, quando se escolhe um plano agregado, tem que se considerar o impacto dos níveis de inventário e as necessidades de capacidade.

Discretização temporal

Habitualmente, tanto os modelos para determinação de lotes económicos como os modelos de escalonamento, baseiam-se numa abordagem que se pode considerar natural e que consiste em discretizar o horizonte temporal (dividindo-o em períodos de duração igual), o que permite associar ao modelo variáveis que indicam que parte de uma determinada entidade (encomenda, ordem de fabrico, . . .), se vai produzir num determinado período.

A discretização temporal permite também considerar, de uma maneira

relativamente simples, um grande número de restrições, que incluem datas de início e de fim de produção e relações de precedência, e também funções objectivo relacionadas com os períodos em que as tarefas são realizadas (o que proporciona uma grande generalidade na sua definição).

Técnicas de resolução

Neste quadro, as formulações dos problemas associados ao planeamento da produção utilizam variáveis inteiras ou binárias (para determinar por exemplo se uma encomenda é ou não produzida num determinado período), o que dá origem a problemas de carácter combinatório de difícil resolução.

Entre as abordagens com melhor desempenho na resolução destes problemas, incluem-se as baseadas em Programação Inteira (com base nos recentes resultados da Teoria Polédrica e na obtenção de desigualdades válidas “fortes”) e as que se baseiam na utilização de meta-heurísticas.

Do primeiro tipo, são exemplo os algoritmos de planos de corte (ver Figura 6.3), nos quais se começa por obter a solução da relaxação linear do modelo de Programação Inteira. Se a solução obtida for inteira, então é solução óptima para o problema original. Se a solução obtida for fraccionária, utiliza-se uma heurística de separação para gerar desigualdades válidas (que, como referido, deverão ser fortes). As desigualdades válidas são a seguir incluídas no modelo, e resolve-se de novo a relaxação linear. Se não se conseguir gerar nenhuma desigualdade válida, poder-se-á obter a solução óptima, utilizando um procedimento do tipo “branch-and-bound” (que, em princípio, deverá ser muito mais eficiente, após a adição das desigualdades válidas).

O procedimento descrito será tanto mais eficiente quanto mais próximo o valor da relaxação linear estiver da solução óptima inteira. Essa característica está intimamente ligada à qualidade da formulação inicial (recomenda-se a leitura de [Schrijver, 1986] e de [Nemhauser and Wolsey, 1988], considerados como livros de texto fundamentais nesta área).

As meta-heurísticas, por outro lado, baseiam-se na aplicação iterada de processos de pesquisa local com algum grau de aleatorização, para obtenção de soluções “satisfatórias” e têm sido utilizadas com sucesso em muitos problemas de Optimização Combinatória. Entre estes procedimentos contam-

se por exemplo a Pesquisa Tabu, o “Simulated Annealing”, os Algoritmos Genéticos e as Redes Neurais (para uma descrição sucinta destas meta-heurísticas, recomenda-se a leitura de [Pirlot, 1994]).

Estes procedimentos constituem em geral processos de optimização “inspirados” na natureza e podem ser considerados genericamente como métodos de Pesquisa Local, em que se pretende minimizar uma função $f(x)$ sobre um conjunto finito e discreto de pontos $\mathcal{X}$, passando iterativamente de uma solução x_n para uma solução x_{n+1} , “vizinha” de x_n , segundo uma determinada estratégia de pesquisa. Dado que, em cada passo, estes procedimentos utilizam uma heurística mais fundamental para passar de uma solução para a seguinte, “limitando-se” a guiar a pesquisa por uma trajectória dentro do conjunto $\mathcal{X}$, designam-se genericamente por meta-heurísticas. A eficiência destas meta-heurísticas depende não só do problema a optimizar, mas também, fortemente, da estrutura de vizinhança definida para este e dos procedimentos ou regras heurísticas utilizadas na geração das soluções. Foi sobre estes aspectos que se concentrou uma parte considerável da investigação realizada no âmbito desta dissertação.

Capítulo 2

Caso de Estudo

2.1 Introdução

Nas indústrias de produção por encomenda (designadas em inglês por MTO de “Make-To-Order”), a negociação das datas de entrega com os clientes reveste-se de elevada importância, pois do seu estabelecimento pode depender a adjudicação das encomendas. O não cumprimento posterior dessas datas pode ter consequências graves, sendo corrente a existência de cláusulas contratuais que conferem ao cliente o direito de não aceitar a encomenda se ultrapassado o prazo de entrega acordado, ou pelo menos de ser devidamente indemnizado pelos atrasos verificados.

Tal como é afirmado por [Hendry and Kingsman, 1989], com as pressões actuais do mercado, os prazos de entrega devem ser o mais curtos possível, por forma a manter-se a competitividade, mas simultaneamente realistas, por forma a poderem ser cumpridos com as capacidades de produção disponíveis. Citando novamente [Hendry and Kingsman, 1989], nas indústrias MTO, fazer previsões correctas de procura e agrupar produtos para efeitos de cargas é muito difícil, pois frequentemente desconhece-se à priori o processo tecnológico que será utilizado na sua produção. Assim, a avaliação das capacidades deverá ser feita apenas no momento da recepção das encomendas. Por outro lado, os elevados custos de armazenamento levam a que se tente produzir o mais perto possível das datas de entrega, para manter os níveis de inventário (de produtos acabados ou intermédios) baixos.

A redução dos prazos de entrega, associada à produção o mais perto possível da data de entrega e à redução dos níveis de inventário intermédios, correspondem a uma vertente do JIT (“just-in-time”). Para além dessa vertente, a aplicação da “filosofia” JIT a uma empresa, tem implicações nas implantações fabris, no “design” do produto, no controlo da qualidade e no sequenciamento da produção e, levada às últimas consequências, implica alterações profundas no funcionamento das empresas. O JIT foi a chave do sucesso industrial dos japoneses, no entanto a sua aplicação na globalidade só é possível em indústrias nas quais a produção se mantenha estável por muito tempo, o que não acontece nas indústrias ligadas à moda.

O trabalho descrito nesta dissertação tem por base o caso de uma empresa da indústria do calçado (ver secção 1.1). Neste capítulo, começa-se por descrever o processo de fabrico de calçado, abordando-se em seguida os processos de decisão associados à produção e as razões que conduziram à opção por um Sistema de Apoio à Decisão (SAD) para o planeamento. Seguidamente, será feita uma “interpretação hierárquica” do SAD em estudo, com base num esquema hierárquico proposto na literatura. Será também feita uma descrição sucinta dos dois níveis de planeamento considerados e, por último, serão descritos os diversos componentes do SAD.

2.2 Descrição do processo de fabrico

O processo produtivo de calçado pode ser dividido em 4 sectores (ou fases) principais: Corte, Costura, Montagem e Acabamentos/Embalagem (ver Figura 2.1).

2.2.1 Corte

No sector do Corte é iniciada a produção de calçado, sendo cortadas as peles e os forros. Os operários que pertencem ao sector do Corte são periodicamente divididos em grupos e cada grupo fica encarregue de cortar um determinado modelo de sapatos. Cada um dos grupos é constituído por operários mais experientes, que cortam as peles, por operários menos experientes, que cortam os produtos sintéticos e ainda por operários que têm como tarefa “arrumar” as peças cortadas, formando conjuntos de 10 pares

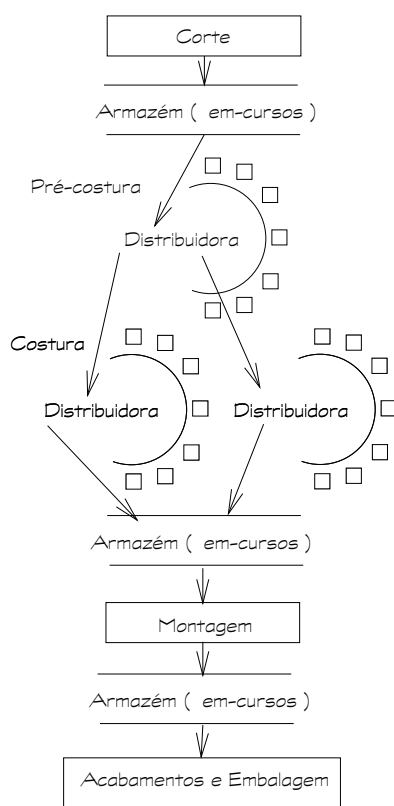


Figura 2.1: Diagrama do processo produtivo

(um lote). A sequência de operações no sector do Corte está representada esquematicamente na Figura 2.2.

As ferramentas utilizadas neste sector são os cortantes, que cortam a pele ao serem pressionados por meio de balancés (prensas). Sempre que se pretende produzir um novo modelo, são encomendados novos cortantes. Os cortantes, são fabricados em fita de aço com aproximadamente 2mm de espessura. A fita de aço é afiada de um dos lados e em seguida é-lhe dado o formato da peça a cortar. É necessário produzir um conjunto de cortantes para cada tamanho que se pensa vir a produzir.

A cada um dos cortadores mais experientes é dado um conjunto de cortantes (correspondentes a uma ou mais peças constituintes do modelo a cortar, para vários tamanhos). O cortador estende a pele e, por inspecção visual, procura defeitos ou irregularidades. Seguidamente, coloca o cor-

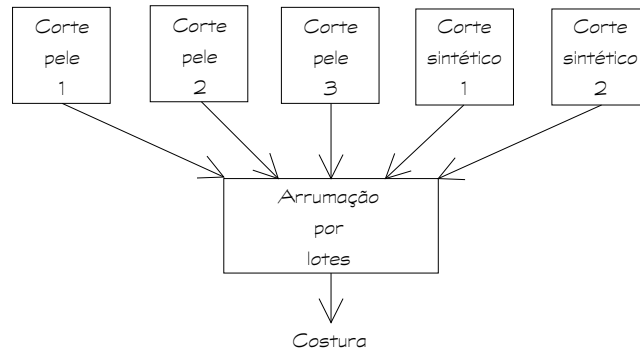


Figura 2.2: Sequência de operações no sector do Corte

tante sobre uma zona da pele sem defeitos e faz actuar um balancé, que faz pressão sobre o cortante, que por sua vez corta a pele. Estes cortadores têm de cortar um número de peças “equilibrado” de cada tipo, minimizando simultaneamente os desperdícios. Aproveitar as peles é crucial dado que estas, além de constituírem a matéria prima mais importante (em termos de custos de incorporação), têm um tempo de reaprovisionamento muito longo, porque provêm de regiões distantes (como por exemplo da América do Sul e do Paquistão). Os cortadores menos experientes (que cortam os produtos sintéticos), colocam a matéria prima em várias camadas sobrepostas, pousam o cortante em cima e fazem actuar um balancé, cortando várias peças iguais em simultâneo (dado que se trata de produtos sintéticos, não há neste caso o mesmo tipo de preocupação relativamente aos defeitos). À medida que vão sendo cortadas, as peças são reunidas num posto de trabalho onde se juntam por forma a constituir lotes (de 10 pares). Cada lote é então colocado em cestos que serão posteriormente utilizados no sector de Costura.

A quantidade de mão-de-obra necessária na zona de Corte é relativamente baixa, com uma produtividade, em geral, mais que suficiente para suprir largamente as necessidades dos sectores seguintes.

2.2.2 Costura

O sector da Costura encontra-se, em geral, subdividido em Pré-costura, onde são cosidos os forros, e Costura propriamente dita, onde é cosido o exterior do calçado, são feitos bordados e acrescentados alguns pequenos componentes, tais como ilhós. Existe normalmente apenas uma linha de pré-costura que alimenta várias linhas de costura em paralelo. O produto que sai das linhas de costura é a gáspea.

A Costura é o sector de uma fábrica de calçado onde é utilizada mais mão-de-obra (habitualmente feminina), e onde há uma grande quantidade de postos de trabalho predominantemente manuais. Para resolver os problemas de absentismo (que, por vezes, são significativos) e para melhor “compatibilizar” as competências e “preferências” das operárias, a tarefa de afectação dos operários aos postos das linhas tem que ser realizada no início de cada dia pelas encarregadas de linha.

A constituição das várias linhas de costura (em termos de postos de trabalho) pode variar bastante ao longo do tempo, adaptando-se às necessidades de produção, dado que as máquinas utilizadas são facilmente transportáveis de uma linha para outra. A sua estrutura básica é, no entanto, sempre a mesma. Todas têm um transportador central, sistema mecânico controlado por uma operária experiente, a “distribuidora”, que, através do transportador, recebe e distribui os lotes de calçado de e para os diferentes postos de trabalho. Estas “linhas de costura” não funcionam realmente como linhas de produção tradicionais, dado que a transferência entre postos de trabalho é sempre feita através da distribuidora (os lotes não passam directamente de um posto de trabalho para o seguinte). No fundo, cada “linha” da Costura está mais próxima de um “job-shop”, com um posto intermédio entre cada duas operações. Nesse posto intermédio (que corresponde à distribuidora), o produto pode “parar” durante pouco tempo, passando de imediato para o posto seguinte, ou pode ser armazenado durante períodos mais longos. A distribuidora é informada, através de um painel sinóptico, da carga de cada posto de trabalho, e deve conhecer em cada instante quais as operações já realizadas ou ainda por realizar em cada um dos lotes que se encontram na “linha” (incluindo os que estão temporariamente armazenados junto de si).

Esta tarefa exige um bom conhecimento do processo e uma grande percepção do “estado” da produção. Uma “boa distribuidora” é, por isso, realmente uma peça chave para a produtividade de uma linha.

Refira-se ainda que, em fábricas que produzam grandes quantidades de um pequeno número de artigos e simultaneamente pequenas quantidades de muitos produtos diferentes, as linhas de costura são geralmente separadas em linhas “dedicadas” e em linha de “diversos”. Nas primeiras são produzidos um ou dois modelos (de maior procura) de cada vez, com um “layout” optimizado, de modo a atingir-se a máxima produtividade. Nas linhas de “diversos”, são normalmente produzidos vários modelos em simultâneo. Dada a tendência crescente neste tipo de indústria para encomendas de pequenas quantidades e grande variedade de modelos, as linhas “dedicadas” tendem a desaparecer.

2.2.3 Montagem

Existem actualmente dois tipos básicos de organização para a Montagem, em “anel” ou em “célula”. Numa Montagem em “anel”, a gáspea (produto final da Costura) é transportada automaticamente de um posto de trabalho para o seguinte, constituindo-se, assim, uma linha de produção “de facto”. No início do “anel”, as gáspeas são colocadas em formas, nelas circulando até ao fim da linha. Numa Montagem em “célula”, os postos de trabalho estão fisicamente muito mais próximos e as gáspeas (previamente colocadas em formas) são transportadas pelos operários de um posto de trabalho para o seguinte.

Tal como já foi referido, o produto que entra na Montagem é a gáspea, que corresponde à parte superior do calçado. Na Montagem, a gáspea é enformada e é colocada a sola. As formas, de material plástico, são produzidas especificamente para cada modelo por empresas externas especializadas e só muito raramente um jogo de formas é comum a mais do que um modelo. Normalmente, no fim de uma época, as formas são inutilizadas. A gestão das formas é um factor muito importante no planeamento. Não sendo directamente tido em consideração neste trabalho, constitui frequentemente um problema, dado ser limitado o número de formas disponíveis.

A Montagem é o sector que necessita de equipamento mais sofisticado e especializado e de um maior controlo da qualidade. Por essas razões as operações desta fase não podem ser, normalmente, subcontractadas. Por isso, a capacidade da Montagem determina, na prática, a capacidade de produção e, consequentemente, a capacidade de aceitação de encomendas.

2.2.4 Acabamentos/Embalagem

Numa última fase, os sapatos são limpos, são metidos os cordões e é feito um controlo de qualidade final (por inspecção visual individual). Finalmente, os sapatos são embalados em caixas individuais, e estas agrupadas em caixas de cartão que são colocadas no armazém de produtos acabados.

2.3 Processos de tomada de decisões

O grupo empresarial para o qual foi feito o estudo que serviu de base a esta dissertação, é constituído por várias unidades fabris (ver secção 1.1), nas quais o processo produtivo é semelhante ao descrito na secção 2.2. Para desenvolver um sistema informático que pudesse apoiar os Agentes de Decisão (AD) no planeamento da produção das unidades fabris, foi feito o levantamento dos processos de tomada de decisões, tendo sido possível estruturá-los em quatro níveis distintos.

Nível 1

Num primeiro nível de planeamento, de carácter mais estratégico ou tático, são decididos, face a previsões da procura, grandes tipos de produtos a produzir em cada estação criando-se um sortido de artigos a apresentar aos clientes (também denominado “catálogo” ou “coleção”). Este nível de decisão envolve todas as unidades da empresa, o que contribui para dificultar o processo de decisão.

Cada uma das unidades fabris da empresa está à partida (e historicamente) mais vocacionada para um determinado tipo de produtos, e as decisões a este nível, a cargo do Director Geral, assessorado pelo Director Comercial e pelo Director de Produção, envolvem a afectação de produções genéricas a cada unidade fabril. Essa afectação tem eventualmente influência

na aquisição de máquinas e na contratação de pessoal, para cada uma das unidades fabris. A nível do grupo, as decisões tomadas neste nível implicam ainda a colocação de encomendas aos fornecedores de peles e de outros componentes específicos com prazos de entrega dilatados.

Nível 2

Num segundo nível, são recebidas encomendas dos clientes, negociadas datas de entrega, e efectuada a distribuição dessas encomendas pelas fábricas. Neste nível, o AD é actualmente um elemento do sector Comercial, com o apoio da Produção. Antes da utilização do SAD, esta tarefa era realizada pelo Director de Produção que, na ausência de um sistema informático que lhe fornecesse informações sobre a capacidade disponível, estimava essa capacidade frequentemente de forma irrealista, o que o levava a aceitar datas de entrega que depois não podia cumprir sem recorrer a horas extraordinárias ou a subcontratações de última hora.

Ainda no segundo nível, as encomendas afectadas a cada fábrica são associadas, formando conjuntos que correspondem ao mesmo artigo, à mesma cor, e a datas de entrega “próximas” (normalmente esse grau de “proximidade” é parametrizável). Um conjunto desses, com cerca de 400 pares é denominado Ordem de Fabrico (OF).

Nível 3

No terceiro nível, cabe ao Departamento de Produção determinar os períodos de entrada em Corte/Costura e em Montagem para cada OF, sendo geradas as necessidades de materiais por um sistema tipo MRP. É também determinado o número de máquinas de cada tipo a afectar, em cada período, a cada uma das linhas de costura.

Nível 4

No quarto nível é determinada a ordem de entrada diária de cada OF em Corte/Costura e é decidida a afectação das pessoas aos postos de trabalho em cada linha. Neste nível, os AD são elementos do Departamento da Produção, e também os encarregados da fábrica.

Note-se que a definição precisa dos níveis de planeamento, do seu inter-

relacionamento, dos horizontes e períodos de planeamento a considerar em cada nível, bem como da informação a utilizar em cada fase, constituem aspectos cruciais do processo de modelização. É, além disso, difícil obter indicações dos diversos AD acerca de todos os objectivos que pretendem ver satisfeitos em cada um dos níveis, bem como da sua quantificação precisa. Por estas razões, e dadas as reconhecidas dificuldades de comunicação entre analistas e Agentes de Decisão, devidas essencialmente às diferentes linguagens utilizadas, esses dados só podem ser obtidos através da realização de visitas regulares às empresas. Ao longo dessas visitas, o analista deve tentar “adquirir” a linguagem dos AD e deve-se aperceber no local dos processos de decisão que decorrem no dia-a-dia, tentando adquirir sensibilidade para a existência de objectivos e restrições muitas vezes difíceis de verbalizar.

2.4 Dos processos de tomada de decisões ao SAD

O Sistema de Apoio à Decisão (SAD) desenvolvido no âmbito deste trabalho, trata apenas do segundo e do terceiro níveis do processo de tomada de decisões descrito na secção 2.3. O nível superior foi excluído por se considerar estar ainda pouco estruturado e ser muito dependente de decisões de carácter mais estratégico, como as que dizem respeito à política de vendas. Relativamente ao nível inferior (quarto), que controlaria directamente a planta fabril, evitou-se introduzi-lo no SAD por se ter constatado na prática que, tal como refere [Tobin et al., 1988, pp. 47], “...a tentativa de substituição dos controladores humanos na planta fabril, por algoritmos computacionais, tem dado maus resultados, sendo difícil igualar e muito menos superar o desempenho de um ser humano com experiência, na confusão do dia-a-dia da planta fabril ...”.

Tradicionalmente, o planeamento da produção é realizado de forma hierárquica, e neste trabalho, foi seguida essa abordagem genérica, tendo contudo em atenção as características especiais das empresas com produção por encomenda. As ferramentas de planeamento desenvolvidas neste projecto, foram implementadas num SAD, sendo o seu objectivo genérico auxiliar o AD, em cada nível de planeamento, na definição de linhas de acção robustas e na sua avaliação para vários cenários alternativos.

Como se pode ver na Figura 2.3, o SAD funciona como “parasita” do Sistema Informático de Base (SIB) da empresa. Este tipo de arquitectura permite por um lado que o SIB (onde está armazenada toda a informação necessária ao bom funcionamento da empresa), disponibilize ao SAD todos os dados necessários para o planeamento, e receba do SAD todos os resultados relevantes do planeamento. Por outro lado, é uma arquitectura modular, que permite que o SAD seja implementado numa máquina diferente daquela onde está o SIB, com capacidade de cálculo e capacidade gráfica muito superiores. A capacidade de cálculo permite desenvolver algoritmos mais sofisticados para resolução dos problemas, e a capacidade gráfica permite desenvolver melhores “interfaces”, que melhoram a interacção entre o SAD e os diversos AD.

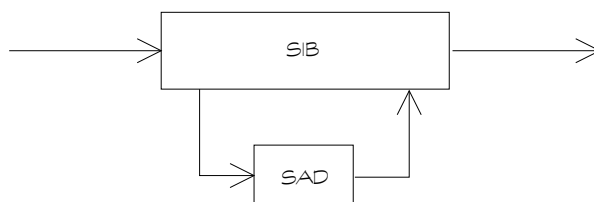


Figura 2.3: O SAD como parasita do Sistema Informático de Base (SIB)

No primeiro nível de planeamento, aqui designado por Plano Director de Produção (PDP), pretende-se apoiar o AD na indicação de datas de entrega a clientes, com base nas capacidades da Montagem, mantendo-o informado sobre as capacidades ainda disponíveis. A necessidade de efectuar uma gestão da carteira de encomendas com uma grande antecedência (no início de cada estação começa-se a planear a estação seguinte), levou a considerar, para este nível, um horizonte de planeamento de cerca de um ano. Os resultados do primeiro nível são introduzidos no SIB, onde as encomendas são reunidas em Ordens de Fabrico (OF).

No segundo nível de planeamento, aqui designado por Programação (PROG), são planeadas as OF, numa primeira fase na Montagem e numa segunda fase na Costura. Esses planos são utilizados pelo SIB para determinação das necessidades de materiais (segundo uma lógica MRP). No segundo nível, é ainda produzida informação sobre cargas (e capacidades

disponíveis) nas diversas linhas, e relativamente às linhas de costura, é fornecida uma relação das máquinas de cada tipo que são necessárias em cada linha e em cada período. Sempre que o número de máquinas necessárias exceder o número total de máquinas existentes de um determinado tipo, ou sempre que o número de postos de trabalho de uma linha exceder o número máximo permitido, o AD é avisado.

2.5 Planeamento Hierárquico

... a razão humana está dependente não de um único centro cerebral mas de vários sistemas cerebrais que funcionam de forma concertada ao longo de muitos níveis de organização neuronal. Tanto as regiões cerebrais de “alto nível” como as de “baixo nível”, ... cooperam umas com as outras na feitura da razão.

Os níveis mais baixos do edifício neurológico da razão são os mesmos que regulam o processamento das emoções e dos sentimentos e ainda as funções do corpo necessárias para a sobrevivência do organismo. Por sua vez, estes níveis mais baixos mantêm relações directas e mútuas com praticamente todos os órgãos do corpo, colocando assim o corpo directamente na cadeia de operações que dá origem aos desempenhos de mais alto nível da razão, da tomada de decisão e, por extensão, do comportamento social e da capacidade criadora. Todos estes aspectos, emoção, sentimento e regulação biológica, desempenham um papel na razão humana. As ordens de nível inferior do nosso organismo fazem parte do mesmo circuito que assegura o nível superior da razão. ...

O Erro de Descartes - emoção, razão e cérebro humano
António R. Damásio

2.5.1 Enquadramento hierárquico

O processo de planeamento descrito, com os seus diferentes níveis hierárquicos, e o SAD que lhe está associado, podem ser analisados de um ponto de vista mais geral tal como o que é proposto por [Schneeweiß, 1995]. A Figura 2.4 representa o esquema proposto em [Schneeweiß, 1994], para representação de um sistema com dois níveis hierárquicos e sem processos de negociação entre os níveis.

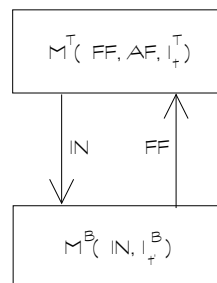


Figura 2.4: Planeamento Hierárquico

Neste esquema, a notação utilizada é a seguinte:

- M^T modelo do nível superior (T , de “top”);
- M^B modelo do nível inferior (B , de “bottom”);
- FF informação “feedforward” do nível inferior para o nível superior;
- AF função de antecipação - descrição de reacções hipotéticas do nível inferior a estímulos hipotéticos do nível superior;
- I_t^T informação disponível no nível superior no instante t ;
- $I_{t'}^B$ informação disponível no nível inferior no instante t'
(sendo $t < t'$, por se considerar que a decisão do nível superior é tomada antes da decisão do nível inferior);
- IN “instruções” do nível superior para o nível inferior (as “instruções” incluem objectivos e restrições impostas pelo nível superior ao nível inferior) ;
- C^T objectivo a ser optimizado no nível superior;
- C^B objectivo a ser optimizado no nível inferior.

2.5.2 Interpretação “hierárquica” do SAD em estudo

Na Figura 2.5 está representado o sistema implementado no SAD do Caso de Estudo, encarado do ponto de vista do esquema hierárquico apresentado por Schneeweiß.

Nível superior (PDP)

- O Sistema Informático de Base (SIB), fornece os seguintes dados ao algoritmo PDP (do SAD):
 - calendário de produção;
 - encomendas (tipo de produto encomendado, respectivas quantidades, datas possíveis de entrada em produção e datas propostas para entrega);

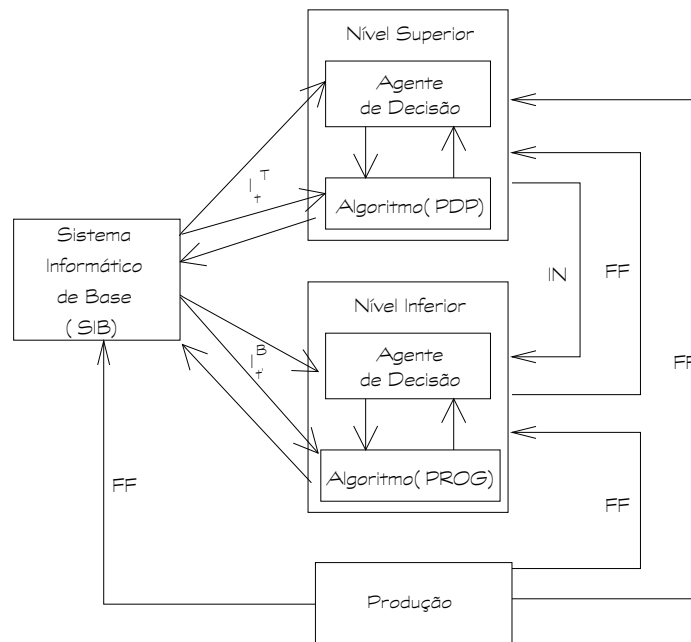


Figura 2.5: Ligações entre níveis de planeamento e entre estes e o SIB

- linhas de montagem de cada uma das fábricas.
- O SIB fornece os seguintes dados ao Agente de Decisão (AD):
 - gamas operatórias.
- O AD obtém a seguinte informação directamente do processo:
 - quantidades produzidas de cada tipo de produto em cada linha;
 - índices de preferência das linhas para produção de determinados tipos de produtos.
- O AD fornece ao algoritmo PDP informação sobre:
 - grandes grupos de produtos com respectivas gamas operatórias;
 - índices de preferência e capacidades de produção de cada linha para cada tipo de produto.
- O algoritmo PDP fornece os seguintes resultados ao SIB:

- períodos e linhas de montagem para produção de cada encomenda;
 - carga utilizada e carga disponível em cada linha.
- Os objectivos do PDP são:
 - produzir o mais perto possível da data de entrega proposta, e se possível antes;
 - produzir os produtos nas linhas mais preferidas.
- O AD pode comparar várias soluções, através de índices de desempenho, fazendo variar as capacidades das linhas e os índices de preferência. Esse processo de teste de soluções pode ser considerado como um processo de negociação entre o AD e o algoritmo.

Actuação do SIB entre o PDP e o PROG

O SIB recebe a informação sobre os períodos e linhas de montagem onde deverão ser colocadas as encomendas. A tarefa do SIB é gerar Ordens de Fabrico (OF) a partir dessas encomendas e associar-lhes também datas de entrega (ou de fim de produção).

As ordens de fabrico são constituídas por aproximadamente 400 pares, e há duas situações que podem ocorrer. Se a encomenda é maior que os 400 pares, terá que ser dividida em OF de aproximadamente 400 pares. Se a encomenda é menor que os 400 pares, procura-se juntar mais do que uma encomenda numa OF. Para se juntarem encomendas em OF, é necessário que as encomendas sejam todas do mesmo produto, com a mesma cor, que estejam planeadas para a mesma fábrica e que tenham datas de entrega próximas (normalmente, esse grau de “proximidade” é parametrizável).

Nível inferior (PROG)

- O SIB dá informações ao algoritmo PROG (do SAD) sobre:
 - calendário de produção;
 - número de máquinas existentes de cada tipo;

- máximo número de máquinas em cada linha;
 - “lead-times” do Corte, Costura e Montagem;
 - tipos de produtos;
 - gamas operatórias para cada tipo de produto;
 - OF (tipo de produto, data possível de início de produção e data de entrega).
- O SIB fornece os seguintes dados ao AD:
 - gamas operatórias;
 - OF a produzir.
- O Agente de Decisão fornece ao algoritmo PROG informação sobre:
 - índices de preferência e capacidades de produção de cada linha para cada tipo de produto.
- Os objectivos do PROG são:
 - produzir o mais perto possível da data de entrega;
 - produzir em cada linha os tipos de produtos de maior preferência;
 - minimizar as alterações de “layout”.
- O PROG dá as seguintes instruções ao SIB:
 - “layout” de cada linha em cada período;
 - para cada período, quantidade de cada OF a produzir em cada linha.
- O AD pode comparar várias soluções, através de índices de desempenho, fazendo variar as capacidades das linhas e os índices de preferência. Esse processo de teste de soluções pode ser considerado como um processo de negociação entre o AD e o algoritmo.

2.5.3 Plano Director da Produção (PDP)

Com o Plano Director da Produção (PDP), pretende-se apoiar a negociação das datas de entrega com os clientes e fazer uma primeira afectação das encomendas a linhas das diferentes unidades fabris. Essa afectação é feita tendo em conta as capacidades das linhas de montagem. Considera-se que cada linha tem capacidades e índices de preferência diferenciados para cada tipo de produto, e que estão disponíveis (para planeamento) encomendas dos diversos tipos de produtos, para as quais se conhecem datas possíveis para entrada em produção, datas de entrega propostas e quantidades encomendadas. É também possível suspender temporariamente o planeamento de uma encomenda bem como “reservar” capacidade produtiva, correspondente a previsões mais ou menos seguras da procura.

Pretende-se, com o PDP, afectar as encomendas às linhas de cada unidade fabril, planeando a sua produção para o mais perto possível da data de entrega proposta, nas linhas de maior preferência (para os produtos em causa), numa única fábrica, no menor número possível de períodos e sem exceder um limite, imposto por razões operacionais, para a produção de cada tipo de produto em cada período.

No Capítulo 3, faz-se uma descrição pormenorizada do PDP.

2.5.4 Programação da Produção (PROG)

Existe um módulo de Programação independente para cada unidade fabril. Com esse módulo é feita a afectação das Ordens de Fabrico (OF) previamente alocadas à fábrica, às linhas de montagem e de corte/costura.

Uma OF deve ser produzida numa única linha, tanto de montagem como de costura. Considera-se que cada linha de produção tem capacidades e índices de preferência diferenciados para cada tipo de produto. Considera-se ainda que a Costura tem uma linha suplementar com capacidade infinita para qualquer tipo de produto, e preferência baixa. Essa linha corresponde à subcontratação de costura (que só deverá ser utilizada após a ocupação dos recursos internos). A cada OF está associada uma data mínima para entrada em produção, uma data de entrega, um tipo de produto e uma quantidade. Pretende-se, neste contexto, determinar as datas de entrada

em Corte/Costura e em Montagem. Por razões operacionais, é desejável que cada OF seja alocada no máximo a dois períodos consecutivos. Após a determinação das datas de entrada em Corte/Costura e em Montagem, calcula-se o número de máquinas de cada tipo que são necessárias em cada linha e em cada período do horizonte temporal, por forma a conseguir produzir todas as OF e, simultaneamente, de modo a não exceder o número de máquinas existentes e o número máximo de postos permitidos em cada linha.

No capítulo 5, é feita uma descrição pormenorizada do PROG.

2.6 Sistema de Apoio à Decisão (SAD)

Nesta secção será feita uma descrição breve da Base de Dados, da Base de Modelos e do “Interface” com o utilizador do Sistema de Apoio à Decisão (SAD) desenvolvido para o Caso de Estudo. Para uma descrição mais detalhada deste sistema informático, consultar [Gonçalves, 1994].

Com o SAD, procura-se apoiar a tomada de decisões nas duas fases de planeamento, PDP e PROG, descritas sucintamente nas secções anteriores. Note-se que o PDP e o PROG foram especificados por forma a satisfazer as necessidades de planeamento da produção, a médio e curto prazo, numa situação que se pode considerar representativa do sector industrial em causa. Para esse efeito é utilizada informação existente no Sistema Informático de Base (SIB) da empresa. A ligação entre o SAD e o SIB é bastante estreita, por forma a assegurar a utilização eficiente da informação disponível, garantindo em simultâneo que não existem redundâncias de informação, devendo o SAD ser encarado como um “parasita” do SIB, dado que vai a esse sistema buscar a maioria dos dados necessários para o processo de planeamento.

A implementação do SAD foi feita numa plataforma de hardware tipo “workstation” com sistema operativo UNIX e um sistema de gestão de bases de dados INFORMIX-ONLINE. As interfaces gráficas foram desenvolvidas em X11/MOTIF e SAGA (Sistema Automático de Geração de Aplicações) e os algoritmos foram desenvolvidos em C e ESQL/C. A Figura 2.6 esquematiza a forma como estão ligadas as várias ferramentas usadas no SAD [Gonçalves, 1994].

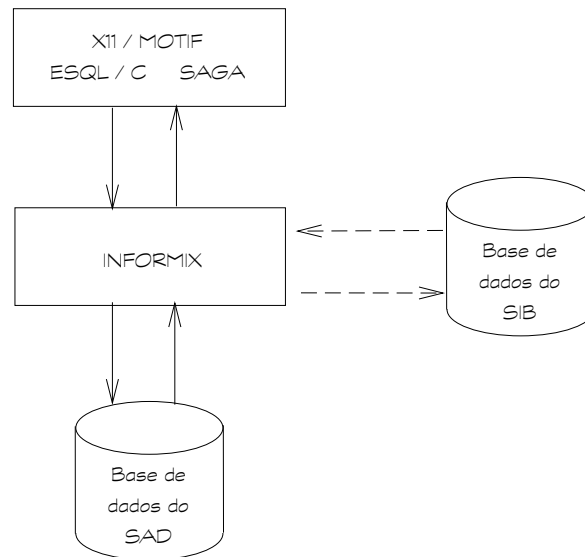


Figura 2.6: Plataforma do SAD

O SAD permite a realização de operações que alteram um cenário de planeamento e a realização de operações de consulta da informação associada a cada cenário. De uma forma geral, um cenário é criado incrementalmente (a partir de cópia e modificação da informação de um cenário previamente criado). No entanto, é também possível criar um cenário novo, de raiz.

2.6.1 Gestão de dados

A gestão de dados do SAD é feita através de um conjunto de funcionalidades associadas ao manuseamento genérico da informação das Bases de Dados, tais como inserções, selecções e extracções. Todas as funcionalidades foram implementadas de modo a garantir a integridade da informação. A base de dados do SAD contém simultaneamente informação existente na base de dados do SIB, informação introduzida/alterada pelo utilizador e informação gerada pelo próprio SAD.

- Informação obtida por consulta à base de dados do SIB:
 - calendário de produção;
 - encomendas, com indicação de:

- * tipo de produto;
 - * quantidade;
 - * data possível para entrada em produção;
 - * data de entrega proposta;
 - OF, com indicação de:
 - * tipo de produto;
 - * quantidade;
 - * data possível para entrada em produção em cada fase;
 - * data de entrega;
 - linhas de produção, com indicação de:
 - * número máximo de postos;
 - * capacidades por tipo de produto;
 - * índices de preferência por tipo de produto;
 - número de máquinas disponíveis de cada tipo;
 - catálogo de artigos;
 - gamas operatórias de fabrico.
- Informação introduzida/alterada pelo utilizador
 - agrupamento de tipos de produtos;
 - agrupamento de encomendas;
 - datas de entrega propostas para as encomendas;
 - índices de preferência das linhas por tipo de produto.
 - Informação gerada pelo SAD
 - cenários alternativos com:
 - * planos de produção das encomendas;
 - * planos de produção das OF;
 - * necessidades (quantidade) de máquinas de cada tipo em cada linha de costura.

2.6.2 Modelos e Algoritmos

Para cada um dos níveis de planeamento incluídos no SAD (PDP e PROG), começaram por ser desenvolvidos modelos de Programação Matemática, que permitiram, numa primeira fase, estruturar ideias e mesmo compreender melhor os problemas a tratar.

Dado que as variáveis dos modelos deveriam indicar que parte da encomenda (ou da ordem de fabrico) se deveria produzir num determinado período, teve que se considerar um horizonte temporal discretizado em ambos os modelos. A discretização temporal permitiu também considerar um grande número de restrições, que incluem datas de início e de fim de produção e a produção num número limitado de períodos e ainda funções objectivo relacionadas com os períodos em que as tarefas são realizadas. Por outro lado, a discretização temporal implica a utilização de variáveis inteiras ou binárias, o que dá origem a problemas de Programação Mista de difícil resolução. Esse facto, associado à existência de objectivos múltiplos a considerar em simultâneo, forçou o desenvolvimento de heurísticas específicas para cada um dos níveis.

As heurísticas desenvolvidas para cada um dos níveis (que têm em conta de uma forma implícita os vários objectivos a considerar em cada nível) foram inspiradas nos procedimentos manuais de planeamento e ajustadas progressivamente, por forma a que os resultados obtidos tivessem as características exigidas pelos Agentes de Decisão. Manteve-se no entanto um grande número de valores parametrizáveis, que permitem a obtenção de diversos cenários que o Agente de Decisão poderá comparar, escolhendo o que melhor satisfaz os objectivos, tal como ele os “percebe” no momento. As heurísticas referidas encontram-se descritas em pormenor nas secções 3.3 e 5.4.

2.6.3 “Interface”

Procurou-se que o diálogo entre o utilizador e o SAD fosse tão amigável quanto possível, para que este se pudesse concentrar nas questões relacionadas com os problemas de planeamento e não no modo de utilização da ferramenta informática. Por outro lado, entendeu-se que, dado o carácter

pouco estruturado dos problemas em causa, era de grande importância “explorar” a experiência e os conhecimentos do utilizador. Tal só pode ser conseguido através de uma interacção fácil do utilizador com o SAD, permitindo no limite que o utilizador “construa” uma solução por “fixação” de períodos e linhas/fábricas, para produção de uma determinada ordem de fabrico/encomenda. O utilizador pode interagir com o SAD por meio de menus, de comandos, de teclas de função e por meio do rato, sendo os resultados normalmente apresentados pelo SAD de forma gráfica e a cores. Sempre que necessário, a informação fornecida pelos gráficos pode ser complementada por tabelas. É também possível manter várias janelas, com gráficos e tabelas, abertas em simultâneo, o que facilita muito a comparação de resultados e a opção por uma ou outra solução por parte do utilizador.

Na Figura 2.7, apresenta-se um exemplo de um “ecran” do SAD, que poderia surgir ao longo do processo de planeamento. Na Figura 2.8, está representado um gráfico de “Gantt” que permite ao utilizador comparar dois cenários de planeamento obtidos com valores diferentes dos parâmetros de entrada do algoritmo. No gráfico estão representados os períodos a que foram afectadas as encomendas nos dois cenários. Num cenário, as encomendas estão representadas a escuro. No outro cenário estão representadas a claro. A encomenda nº 51591, que no cenário representado a escuro tinha sido afectada a um período, foi afectada a dois períodos no cenário representado a claro.

2.6.4 Exemplo de utilização do SAD

Nesta secção descreve-se uma sessão de trabalho típica, sem se pretender ser exaustivo, mas apenas com o objectivo de ilustrar os princípios gerais de “funcionamento” do SAD. A sessão de trabalho corresponde ao primeiro nível de planeamento (PDP).

No início da sessão de trabalho, o Agente de Decisão (AD) pode alterar:

- o agrupamento por tipos de produtos;
- as capacidades das linhas para cada tipo de produto;
- os índices de preferência das linhas para cada tipo de produto;

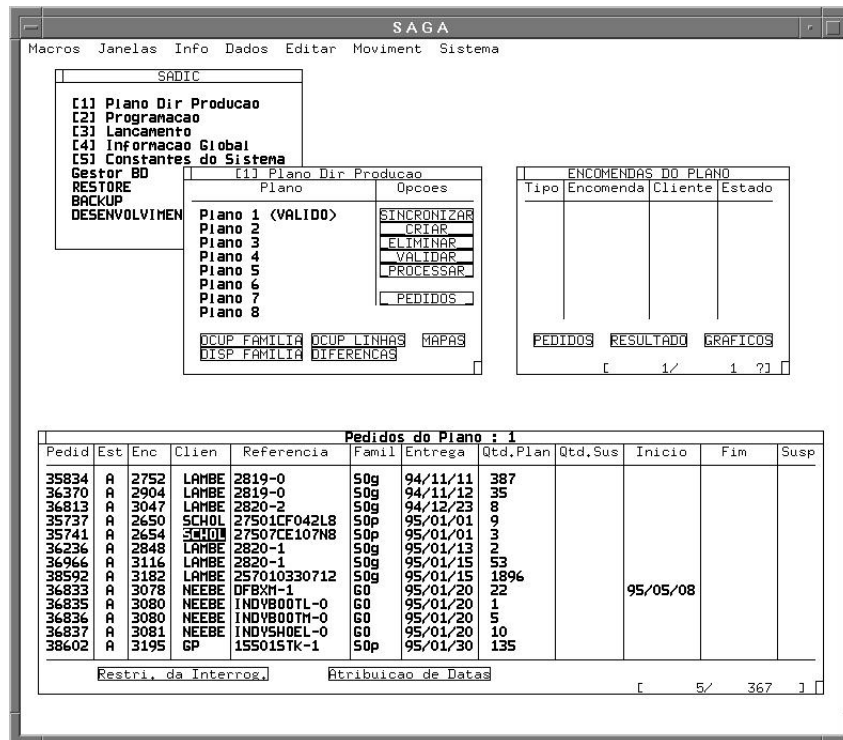


Figura 2.7: Exemplo de um ecran do SAD

devendo-se certificar de que todos os tipos de produtos têm associada uma capacidade e um índice de preferência para cada linha. Seguidamente, deve ser feita a “sincronização” da Base de Dados do SAD com a Base de Dados do SIB, o que corresponde a actualizar na Base de Dados do SAD a informação relevante para o planeamento. Após a sincronização, o AD poderá fazer uma consulta à Base de Dados do SAD, para saber que encomendas ainda não foram planeadas. Na Figura 2.7, pode-se ver o resultado de uma consulta desse tipo.

Suponha-se que o AD está a negociar com o cliente a data de entrega de uma futura encomenda, que ainda está registada como encomenda de reserva e cujo número é -39 (correspondendo ao pedido número 53157). Para tal é necessário criar um novo plano (o que corresponde à criação de um conjunto de informação para a simulação de um cenário de planeamento) e seguidamente processar os dados “correndo” o algoritmo PDP.

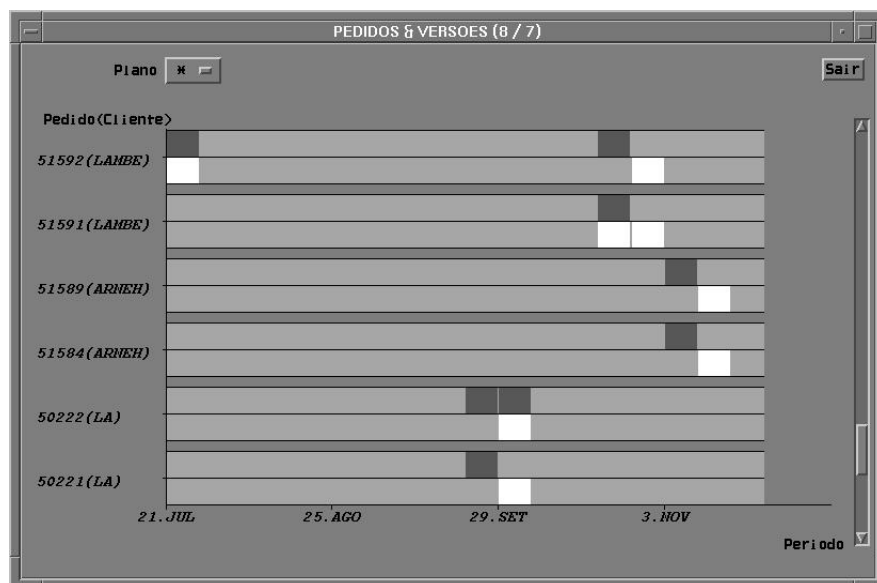


Figura 2.8: Exemplo de um gráfico obtido como resultado no SAD

Depois de “correr” o algoritmo PDP, o AD pode analisar os resultados e verificar qual a data de entrega que o SAD propõe para a encomenda e em que linhas a encomenda foi colocada. Na Figura 2.9 pode-se verificar que a encomenda número -39 (pedido número 53157) foi planeada nas linhas 2.MTG-I (1795 pares) e 3.MTG-I (705 pares), na semana de 10 de Novembro. Se o cliente pretender uma data mais próxima, por exemplo 95/11/03, o AD pode introduzir esse dado na base de dados (ver Figura 2.10) e processar novamente, obtendo o resultado também apresentado na Figura 2.10, onde se mostra que a encomenda foi planeada para a semana de 3 de Novembro, na linha 4.MTG-I.

Por fim, é possível comparar os dois planos obtidos (antes e depois da introdução da data de fim), tal como se encontra representado na Figura 2.11, e onde se pode ver que a encomenda número -39 (pedido número 53157) foi antecipada (da semana de 10 de Novembro para a semana de 3 de Novembro) e que, como contrapartida, os pedidos números 51589 e 51584 foram atrasados, passando da semana de 10 de Novembro para a semana de 17 de Novembro. Se houver acordo entre o AD e o cliente sobre esta nova data

SAGA

Macros Janelas Info Dados Editar Moviment Sistema

SADIC

[1] Plano Dir Producao

[2] Programacao

Pedid

Est

Enc

Clien

Referencia

Famil

Entrega

Qtd.Plan

Qtd.Sus

Inicio

Fim

Susp

47360

A

3456

PAVER

233024RF8552

50p

95/09/10

60

95/07/17

47361

A

3456

PAVER

233024RG8551

50p

95/09/10

60

95/07/17

47362

A

3457

PAVER

265015M55201

CS

95/09/10

61

95/07/17

47363

A

3457

PAVER

265015M55260

CS

95/09/10

60

95/07/17

47364

A

3458

PAVER

265045M75201

SC

95/09/10

60

95/07/17

47365

A

3459

PAVER

233214RF8201

50p

95/09/10

80

95/07/17

47366

A

3459

PAVER

233214PH8487

50p

95/09/10

60

95/07/17

49904

A

3458

PAVER

265195M78089

CS

95/09/10

57

95/06/19

49905

A

3458

PAVER

194178008583

BA

95/09/10

60

95/07/17

53157

N

-39

LAMBE

257010330712

50g

95/11/01

2500

48607

A

3545

CLARK

TERRI TORV-2

G0

95/08/31

600

95/08/31

95/09/01

48606

A

3544

CLARK

TERRI TORV-2

G0

95/09/01

156

95/09/04

95/09/01

49034

A

3571

CLARK

WANDERER-0

G0

95/09/01

504

Restri. da Interrog.

Atribuicao de Datas

[48/ 51]

MAPA PLANEAMENTO POP

Plano 2

Semana

Encom

Cliente

Pedido

Familia

Referencia

Entrega

Qtd

Linha

10.NOV

39

LAMBE

53157

50g

257010330712

95/11/01

1795

2.MTG-I

10.NOV

-39

LAMBE

53157

50g

257010330712

95/11/01

705

3.MTG-I

Total:

2500

Imprimir (^Xp)

[1/ 2]

Figura 2.9: Tabela de encomendas e resultados do planeamento para encomenda número -39 (pedido número 53157)

de entrega, e se o AD verificar que as alterações impostas aos outro pedidos são aceitáveis, o segundo plano pode ser validado, o que corresponde a introduzir os resultados obtidos pelo planeamento (plano 7) na base de dados do SIB.

The screenshot shows the SAGA software interface. The main window has a menu bar with 'Macros', 'Janelas', 'Info', 'Dados', 'Editar', 'Moviment', and 'Sistema'. Below the menu bar, there are two tabs: 'SADIC' and 'Plano Dir Producao'. The 'Plano Dir Producao' tab is active, showing a table of production orders. The table has columns: 'Pedid', 'Est', 'Enc', 'Clie', 'Referencia', 'Fam', 'Entrega', 'Qtd.Plan', 'Qtd.Sus', 'Inicio', 'Fim', and 'Susp'. The first row shows order 53157, status N, enclosure -39, client LAMBE, reference 257010330712, family 50g, delivery date 95/11/01, planned quantity 2500, and end date 95/11/03. Below this table, there is a 'MAPA PLANEAMENTO POP' section for 'Plano 7'. It shows a Gantt chart with columns for 'Semana', 'Encom', 'Cliente', 'Pedido', 'Familia', 'Referencia', 'Entrega', 'Qtd', and 'Linha'. The first row shows order 53157, status N, client LAMBE, reference 257010330712, family 50g, delivery date 95/11/01, planned quantity 2500, and line 4.HTG-I. At the bottom, there is a 'Total: 2500' and a button 'Imprimir (^Xp)'.

Pedid	Est	Enc	Clie	Referencia	Fam	Entrega	Qtd.Plan	Qtd.Sus	Inicio	Fim	Susp
53157	N	-39	LAMBE	257010330712	50g	95/11/01	2500			95/11/03	

Semana	Encom	Cliente	Pedido	Familia	Referencia	Entrega	Qtd	Linha
3.NOV	-39	LAMBE	53157	50g	257010330712	95/11/01	2500	4.HTG-I

Total: 2500

Imprimir (^Xp)

Figura 2.10: Introdução de data de fim para encomenda número -39 (pedido número 53157) e resultados do novo planeamento

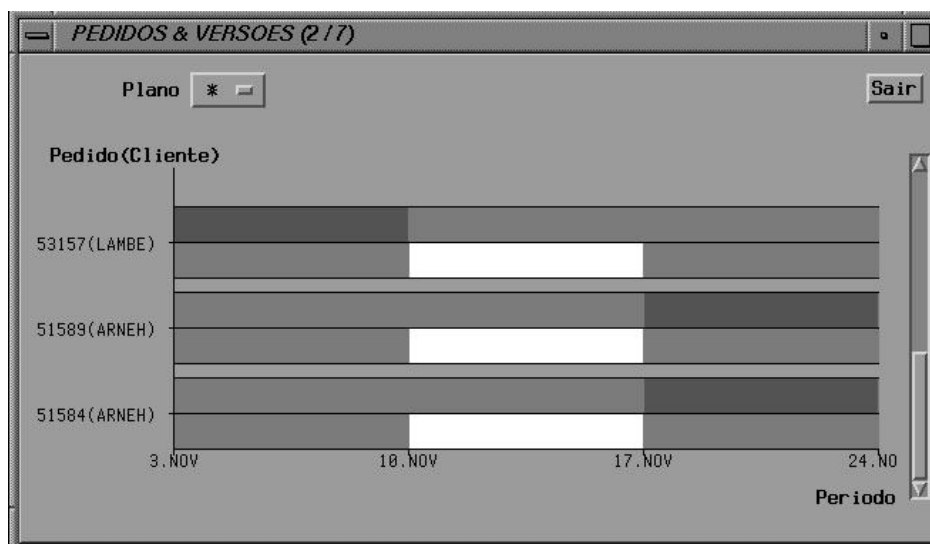


Figura 2.11: Alterações no plano com introdução de data de fim para encomenda número -39

Capítulo 3

Plano Director da Produção

3.1 Introdução

Com o processo de planeamento aqui designado por Plano Director da Produção (PDP), pretende-se apoiar a negociação de datas de entrega aos clientes e fazer uma primeira alocação das encomendas às linhas de produção das diferentes unidades fabris. Essa alocação é feita tendo em conta as capacidades da fase de montagem, considerada como o “bottleneck” do sistema produtivo.

Considera-se que cada linha de produção tem capacidades e índices de preferência diferenciados para cada tipo de produto, e que existem encomendas dos diversos tipos de produtos, para as quais se conhecem datas possíveis para entrada em produção, datas de entrega propostas e quantidades encomendadas.

Pretende-se com o PDP afectar as encomendas às linhas de cada unidade fabril, planeando a sua produção o mais perto possível da data de entrega proposta, nas linhas de maior preferência para o produto em causa, numa única fábrica, no menor número possível de períodos e sem exceder um limite, imposto por razões operacionais, para a produção de cada tipo de produto em cada período.

Neste capítulo começa-se por apresentar (secção 3.2) um modelo desenvolvido para este problema que, como já foi referido, envolve objectivos múltiplos e em larga medida conflituosos. Devido a esse facto e também à

dimensão do modelo do problema em situações reais, não é possível obter uma “solução óptima” do problema. Optou-se então por desenvolver uma heurística para a sua resolução, cuja descrição é feita na secção 3.3.

3.2 Modelo

Nesta secção, descreve-se um modelo em que são fundamentais as seguintes entidades:

x_{elt} número de unidades da encomenda e a produzir na linha l no período t ;

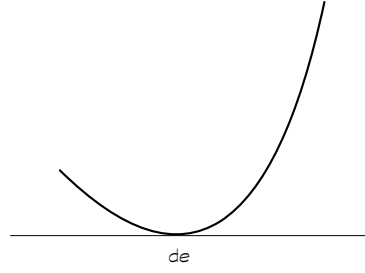
C_{jlt} capacidade de produção do produto tipo j na linha l no período t (em unidades).

No Apêndice A são descritas as restantes entidades, e é apresentada uma descrição completa da notação utilizada.

3.2.1 Funções objectivo

Da experiência prática acumulada, foi possível concluir que há três objectivos, normalmente considerados em simultâneo. Com a função objectivo “*JIT*”, representada na equação (3.1) e na Figura 3.1, procura-se que cada encomenda seja produzida o mais perto possível da data de entrega proposta e se possível antes dessa data. Assim são penalizadas na função objectivo, tanto a produção antes da data de entrega como a produção depois da data de entrega. Esta última é em geral muito menos desejável, sendo o custo associado controlado pelo parâmetro a . Nos exemplos práticos tem sido usado o valor $a=3$, com resultados considerados satisfatórios. Com o coeficiente 6 na segunda parcela da expressão ($6x_{elt}$), pretende-se impor que o custo de produzir no período seguinte à data de entrega proposta ($d_e + 1$) seja superior ao custo de produzir no período anterior a essa mesma data.

Com a função objectivo “*PREF*”, (equação 3.2) procura-se que as encomendas sejam produzidas nas linhas de maior “preferência”. A preferência é medida através de um índice. Um índice de preferência 0 implica sempre uma capacidade de produção nula. Os índices de preferência são valores inteiros não negativos e a preferência diminui com o valor do índice.

Figura 3.1: Função objectivo *JIT*

Com a função objectivo “*ESP*” (equação 3.3) procura-se minimizar o espalhamento da produção de cada encomenda, ou seja, pretende-se minimizar o número de períodos e de linhas em que é produzida cada encomenda (dado que $\gamma_{elt} = 1$ é equivalente a $x_{elt} \geq lb$ isto é x_{elt} positivo).

$$JIT \quad \min \quad \sum_{e,l,t \leq d_e} x_{elt}(d_e - t) + \sum_{e,l,t=d_e+1} 6x_{elt} + \sum_{e,l,t \geq d_e+2} x_{elt}(t - d_e)^a \quad (3.1)$$

$$PREF \quad \min \quad \sum_{e,l,t} x_{elt} * P_{jlt:\tau_e=j} \quad (3.2)$$

$$ESP \quad \min \quad \sum_{e,l,t} \gamma_{elt} \quad (3.3)$$

3.2.2 Restrições

$$\forall_e \quad \sum_{l,t} x_{elt} = q_e \quad (3.4)$$

$$\forall_{j,t} \quad \sum_{e:\tau_e=j} \sum_l x_{elt} \leq U_{jt} \quad (3.5)$$

$$\forall_{l,t} \quad \sum_{e:\tau_e=j} \sum_{j:C_{jlt}>0} \frac{x_{elt}}{C_{jlt}} \leq 1 \quad (3.6)$$

$$\forall_e \quad \sum_f \delta_{ef} \leq 1 \quad (3.7)$$

$$\forall_{e,f} \quad \sum_t \sum_{l:l \in f} x_{elt} - q_e \delta_{ef} \leq 0 \quad (3.8)$$

$$\forall_{e,l,t} \quad x_{elt} - lb\gamma_{elt} \geq 0 \quad (3.9)$$

$$\forall_{e,l,t} \quad x_{elt} - q_e\gamma_{elt} \leq 0 \quad (3.10)$$

$$\forall_{e,f} \quad \delta_{ef} \in \{0, 1\} \quad (3.11)$$

$$\forall_{e,l,t} \quad \gamma_{elt} \in \{0, 1\} \quad (3.12)$$

$$\forall_{e,l,t \geq r_e} \quad \text{se } P_{jlt:\tau_e=j} > 0 \text{ e } C_{jlt:\tau_e=j} > 0 \Rightarrow x_{elt} \in \{0, 1\} \quad (3.13)$$

As restrições (3.4) garantem que para cada encomenda e , a quantidade produzida é igual à quantidade encomendada.

As restrições (3.5) garantem que em cada período t , a produção do produto tipo j não excede um limite superior previamente definido pela administração da empresa.

As restrições (3.6) garantem que a capacidade produtiva de cada linha não é excedida (para uma interpretação mais detalhada desta equação, recomenda-se a leitura da explicação associada à Tabela 3.1).

As restrições (3.7) e (3.8) garantem, através das variáveis binárias δ_{ef} , que cada encomenda e é produzida apenas numa fábrica f .

As restrições (3.9) e (3.10) definem as variáveis γ_{elt} (necessárias sempre que se considera a função objectivo (3.3)).

Adicionalmente, impõem-se as condições (3.13) por forma a garantir que um produto tipo j nunca será produzido numa linha para a qual a capacidade ou preferência para esse tipo de produto sejam nulas, e ainda que uma encomenda nunca será produzida antes da data de início possível de produção (em termos computacionais, as variáveis não são definidas nesse caso).

3.2.3 Dimensão do modelo para problemas reais

Para se fazer uma estimativa do tamanho deste modelo nos problemas reais, considerem-se os seguintes valores “médios”: número de encomendas $E = 1000$, número de tipos de produtos $J = 10$, número de linhas $L = 10$, número de fábricas $F = 3$ e número de períodos $T = 25$.

Ignorando a função objectivo (3.3), o número de variáveis contínuas será $ELT = 250000$, o número de variáveis binárias será $EF = 3000$ e o número de restrições será igual a $2E + JT + LT + EF = 5500$. No entanto, se se considerar explicitamente a função objectivo (3.3), o número de variáveis binárias passa para $EF + ELT = 253000$ e o número de restrições aumenta para $2E + JT + LT + EF + 2ELT = 505500$.

Mesmo quando se ignora a função objectivo (3.3), as 3000 variáveis binárias a considerar numa instância real “normal”, tornariam a resolução do problema por programação matemática impossível em tempo útil. Para além desse facto, estamos perante um problema multiobjectivo, para o qual não é possível encontrar uma “solução óptima” (ver Capítulo 4).

Por último, a estrutura das soluções “óptimas” obtidas por resolução do problema por programação matemática podem mudar radicalmente, mesmo com uma pequena alteração dos dados. Essa situação pode levar o Agente de Decisão a não aderir ao método de resolução, por não o poder “seguir”.

Esses factos, levaram ao desenvolvimento de uma heurística orientada para o Caso de Estudo, que se descreve na secção 3.3.

3.3 Heurística para o Caso de Estudo (*H-PDP*)

Nesta secção descreve-se a heurística desenvolvida para o Plano Director de Produção que se passará a designar por *H-PDP*.

Trata-se, em linha gerais, de uma heurística construtiva em que o planeamento é realizado encomenda a encomenda e cujo método de planeamento se pretendeu aproximar o mais possível do método de planeamento do Caso de Estudo, de modo que as soluções obtidas pela heurística, tivessem as mesmas características que as soluções obtidas anteriormente de forma manual.

3.3.1 Critérios para avaliação de soluções

Os critérios considerados para avaliação de soluções e que se pretendem minimizar são o *JIT*, a *PREF* e o *ESP*, que correspondem às funções objectivo apresentadas no modelo (ver secção 3.2.1) e que se descrevem a seguir. Podem-se apresentar ao Agente de Decisão outros critérios para avaliação de soluções, tais como o atraso da encomenda mais atrasada, ou ainda o número de encomendas atrasadas.

- *JIT*, com o qual se pretende “medir” a “distância” entre a data de produção e a data de entrega proposta, para todas as encomendas. A produção antes da data de entrega proposta é menos penalizada que a produção depois dessa data.

$$JIT = \sum_{e,l,t \leq d_e} x_{elt}(d_e - t) + \sum_{e,l,t=d_e+1} 6x_{elt} + \sum_{e,l,t \geq d_e+2} x_{elt}(t - d_e)^3$$

- *PREF*, com o qual se pretende quantificar a afectação das encomendas às linhas de maior “preferência”. A preferência é medida através de um índice introduzido pelo utilizador. Um índice de preferência 0 implica sempre uma capacidade de produção nula. Os índices de preferência são valores inteiros não negativos, e a preferência diminui com o valor do índice.

$$PREF = \sum_{e,l,t} x_{elt} * P_{jlt_{comj}:\tau_e=j}$$

- *ESP*, com o qual se pretende minimizar o número de períodos e de linhas em que é produzida cada encomenda.

$$ESP = \sum_{e,l,t:x_{elt}>0} 1$$

3.3.2 Heurística *H-PDP*: descrição geral

Para resolver o problema do PDP, foi desenvolvida uma heurística inspirada nos procedimentos manuais, e que tem em conta de uma forma implícita os critérios de avaliação de soluções apresentados, valorizando especialmente o “JIT”, dado que, de acordo com o levantamento do problema do Caso de Estudo, é esse o critério mais valorizado pelo Agente de Decisão.

Esta heurística é constituída pelas seguintes fases:

Fase 1 Para cada encomenda, calcular o número máximo de linhas de produção a utilizar, o número máximo de períodos nos quais pode ser produzida e o máximo número de períodos entre o início da produção e a data de entrega. Esses valores dependem das constantes A_1 , A_2 e A_3 , introduzidas pelo utilizador e descritas a seguir, e da quantidade encomendada. O objectivo é evitar que uma encomenda de uma pequena quantidade seja dividida por um número grande de linhas de produção ou por um número grande de períodos (para mais pormenores, ver descrição da função **DeterminaLimites(e)** no Apêndice B).

A_1 constante introduzida pelo utilizador - quanto maior o valor de A_1 , menor o número de linhas pelas quais se podem dividir as encomendas;

A_2 constante introduzida pelo utilizador - quanto maior o valor de A_2 , menor o número de períodos pelos quais se podem dividir as encomendas;

A_3 constante introduzida pelo utilizador - quanto maior o valor de A_3 , maior o número de períodos permitido entre o início da produção de uma encomenda e d_e .

Fase 2 Tentar colocar a encomenda na linha (ou conjunto de linhas) de maior preferência, e o mais perto possível da data de entrega, satisfazendo as datas de entrega e tendo em conta as capacidades ainda disponíveis para o tipo de produto ao qual pertence a encomenda.

Fase 3 No caso de não ser possível colocar a encomenda na fase 2, “ir relaxando” progressivamente os limites calculados na fase 1 e, se necessário, tentar planear para depois da data de entrega proposta ou tentar mesmo planear uma menor quantidade.

3.3.3 Dados de entrada

Os dados necessários para o algoritmo descrito, são:

1. Encomendas:

- data de entrega;
- data de início possível de produção, “release date”;
- tipo de produto;
- quantidade.

2. Linhas de Produção:

- número de identificação;
- localização física (fábrica na qual está a linha).

3. Tipos de produtos.

4. Matriz das capacidades, com os valores de C_{jlt} (capacidade de produção do produto tipo j na linha l no período t (em unidades)) e de U_{jt} (máxima produção possível do produto tipo j em todas as linhas no período t (em unidades)) (ver Tabela 3.1).

No exemplo da Tabela 3.1, pode-se por exemplo ler que se só for produzido o produto tipo A na linha 1, então poderão ser produzidas 1000 unidades durante o período t . No caso de se pretender produzir os produtos tipo A e B na linha 1, então pode-se produzir p_1 unidades do tipo A e p_2 unidades do tipo B, de tal modo que $\frac{p_1}{1000} + \frac{p_2}{500} \leq 1$

	C_{jlt}				U_{jt}
	Linha 1	Linha 2	Linha 3	Linha 4	
$j = A$	1000	1500	1200	1300	1000
$j = B$	500	300	700	800	1000
$j = C$	250	2000	500	200	1500

Tabela 3.1: Matriz com capacidades de produção por tipo de produto j , para um período t

5. Matriz dos índices de preferência de cada linha de produção l , para cada tipo de produto j , para um dado período t (valores de P_{jlt}) (ver Tabela 3.2).

Um índice de preferência 0 implica sempre uma capacidade de produção nula. Os índices de preferência são valores inteiros positivos e a preferência diminui com o valor do índice.

	P_{jlt}			
	Linha 1	Linha 2	Linha 3	Linha 4
$j = A$	0	1	1	3
$j = B$	0	1	3	9
$j = C$	2	2	2	1

Tabela 3.2: Matriz de índices de preferência de cada tipo de produto j para cada linha l , para um dado período t

Se se pretender por exemplo gerar uma lista de índices de preferência de tipos de produtos para a linha 3, esta será (por ordem decrescente de preferência) tipo A, tipo C e tipo B.

Se se pretender gerar uma lista de índices de preferência de linhas para

o tipo de produto B, esta será (por ordem decrescente de preferência) linha 2, e a seguir linha 3 e linha 4 (o tipo de produto B não pode ser produzido na linha 1).

6. Calendário de Produção (valores de b_t), ver exemplo na Tabela 3.3. Por exemplo, no período 1 há 2880 minutos de produção. Estes valores são usados para calcular os valores de C_{jlt} e de U_{jt} .

t	1	2	3	4	...	n
b_t	2880	3010	2900	2950	...	3000

Tabela 3.3: Calendário de Produção

3.3.4 Resultados

1. Para cada encomenda:
 - períodos em que vai ter lugar a produção;
 - quantidades a produzir em cada período;
 - linhas em que vai ser produzida.
2. Para cada linha:
 - cargas utilizadas (em unidades);
 - cargas disponíveis (em unidades).

3.3.5 Descrição genérica do algoritmo

No Algoritmo 3.1, está uma descrição genérica da heurística *H-PDP*. Segue-se uma descrição sucinta das funções mencionadas no algoritmo. Para uma descrição completa das funções utilizadas, ver Apêndice B.

Inicializa() inicializa variáveis globais.

DeterminaLimites(e) permite que o utilizador controle o tipo de solução que irá obter, pela introdução de três constantes, limitando o “espalhamento” das encomendas pelas linhas e períodos e o número de períodos entre o início da produção e a data de entrega d_e .

```

Inicializa()
para ( $e = 1; e \leq E; e = e + 1$ )
  DeterminaLimites( $e$ )
  se (ColocaMaisTarde( $e$ ) = 1)
    próxima  $e$ 
  se não
    se (RelaxaLimites( $e$ ) = 1)
      próxima  $e$ 

```

Algoritmo 3.1: Descrição genérica de *H-PDP*

ColocaMaisTarde(e) coloca a encomenda e o mais tarde possível, no menor número de períodos possível e no menor número de linhas possível (tendo em conta os índices de preferência, as capacidades disponíveis e os limites determinados em **DeterminaLimites**(e)).

RelaxaLimites(e) encontra um conjunto de períodos e um conjunto de linhas para produzir e , mesmo que para isso devam ser relaxados alguns dos limites calculados inicialmente (relaxando mesmo, se necessário, a data de entrega, e em último caso, tentando produzir uma quantidade menor que a quantidade encomendada).

3.3.6 Exemplo ilustrativo

No Apêndice C descreve-se pormenorizadamente a resolução de uma instância do problema PDP com a heurística *H-PDP*. Apresentam-se a seguir os dados da instância e ainda os resultados, sob a forma de gráfico de Gantt (ver Figura 3.2).

(Para uma descrição completa da notação utilizada, recomenda-se a consulta do Apêndice A)

Dados de entrada

$T = 3$, $A_1 = 2000$, $A_2 = 4000$ e $A_3 = 1$.

As linhas 1 e 2 pertencem à fábrica 1 e as linhas 3 e 4 pertencem à fábrica 2. Os restantes dados necessários encontram-se nas Tabelas 3.4 e 3.5.

	Linha 1	Linha 2	Linha 3	Linha 4	
					P_{jlt}
$j = 1$	0	1	1	4	
$j = 2$	1	0	1	1	
					C_{jlt}
$j = 1$	0	5000	2500	5000	U_{jt}
$j = 2$	5000	0	5000	2500	2500

Tabela 3.4: Índices de preferência, P_{jlt} e capacidades de produção, C_{jlt} , das linhas l por tipo de produto j , para todos os períodos t e máxima produção possível do produto tipo j em todas as linhas no período t , U_{jt}

e	τ_e	r_e	d_e	q_e
1	1	1	1	5000
2	1	1	2	10000
3	2	1	2	3000
4	2	2	3	3000

Tabela 3.5: Lista de encomendas: tipo de produto (τ_e), “release date” (r_e), data de entrega (d_e) e quantidade (q_e)

Resultados

Na Figura 3.2 apresentam-se os resultados da aplicação da heurística *H-PDP* ao exemplo apresentado. Os resultados são representados por meio de um diagrama de Gantt em simultâneo com um diagrama de carga, indicando para que períodos e linhas foram planeadas as encomendas e qual a carga de cada linha em cada período.

3.3.7 Aplicação da heurística *H-PDP* ao Caso de Estudo

Dimensão dos problemas

No Caso de Estudo, os problemas reais tratados com esta heurística têm tipicamente as seguintes dimensões:

- 25 períodos de planeamento;
- 10 linhas;
- 3 fábricas;
- 10 tipos de produtos;

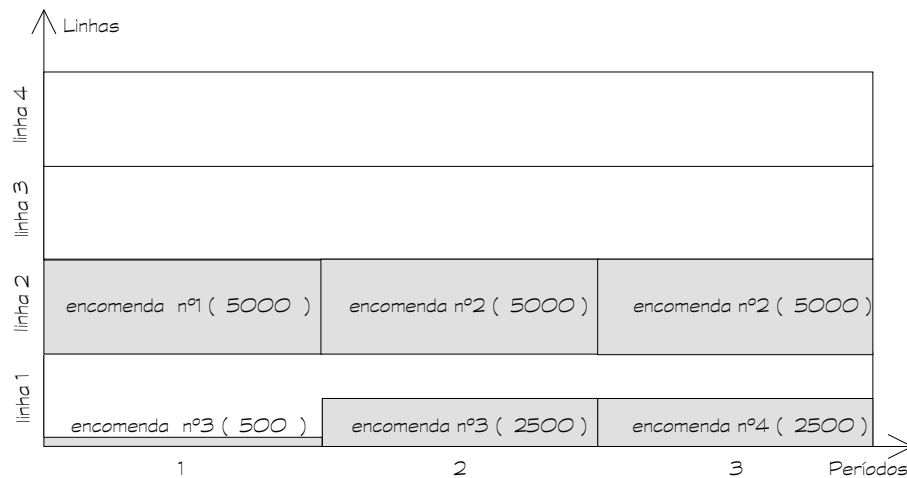


Figura 3.2: Resultados do exemplo PDP

- 1000 encomendas.

A solução para uma instância típica do Caso de Estudo, com as dimensões apresentadas, é obtida em aproximadamente 20 segundos.

Comentários à utilização da heurística e suas limitações

Ao longo do processo de implementação do SAD, foram sendo recolhidas, junto dos Agentes de Decisão (AD) apreciações sobre as soluções obtidas. Os resultados computacionais, obtidos pela aplicação da heurística *H-PDP* aos problemas do Caso de Estudo, mereceram globalmente a aprovação dos planeadores dado que, por um lado, as soluções são estruturalmente semelhantes às obtidas pelos métodos utilizados no planeamento anterior na empresa e, por outro lado, são obtidas em apenas alguns segundos, o que permite fornecer indicações aos clientes sobre datas de entrega, durante uma curta chamada telefónica.

No entanto, houve funcionalidades que não foram implementadas na heurística e para as quais os AD nos foram alertando ao longo do processo de instalação. Sempre que possível, tentou-se incorporar essas novas funcionalidades na heurística, o que, nalguns casos criou dificuldades que foram sendo, estão a ser e provavelmente ainda virão a ser resolvidas, e que se apresentam nos parágrafos seguintes.

Inicialmente, a heurística *H-PDP* foi desenvolvida considerando que as encomendas seriam planeadas por data de entrada no sistema informático. Essa ordem de planeamento foi alterada a pedido do AD, e actualmente as encomendas são planeadas por ordem de data de entrega. Esse facto deu origem a que, sempre que é introduzida uma nova encomenda (e essencialmente se essa nova encomenda tiver uma data de entrega muito próxima da data actual), o planeamento total obtido é muito diferente do obtido antes da introdução da encomenda. Assim, as soluções obtidas são muito instáveis, e será necessário incorporar mecanismos para fixação parcial de soluções, quer automática, quer manualmente pelo AD. Para além da situação descrita, pode mesmo interessar ao AD fixar o planeamento de uma encomenda ou de um conjunto de encomendas numa linha ou num período, antes de executar o algoritmo. Também os parâmetros A_i , que podem ser fixados pelo AD, deveriam existir em maior número, permitindo controlar independentemente diversos tipos de produtos.

Apresentam-se a seguir dois exemplos, para os quais a heurística não dá uma resposta satisfatória, e que terão que ser resolvidos num futuro muito próximo:

Quando se planeia uma encomenda que não pode ser totalmente produzida antes da data de entrega proposta, tenta-se planear com atraso, alargando o número de períodos em que esta pode ser produzida. Esse alargamento do número de períodos pode dar origem a que uma encomenda possa ser planeada em vários períodos espalhados no tempo. Assim sendo, uma encomenda pode ser planeada em parte para a semana seguinte e em parte para daí a 6 meses!

Dado que as encomendas são planeadas por uma ordem pré-determinada, pode ocorrer a seguinte situação. Considerem-se duas encomendas do mesmo tipo de produto, e que esse produto só pode ser planeado para uma linha. As encomendas têm datas de início de produção respectivamente r_1 e r_2 (com $r_1 < r_2$) e data de entrega pretendida $d_1 = d_2 = d$. A encomenda 1 será planeada em primeiro lugar para o mais tarde possível e considere-se que ocupa toda a capacidade disponível entre r_2 e d . Nesse caso a encomenda 2 terá que ser planeada para depois da data de entrega pretendida, apesar de haver capacidade disponível entre r_1 e r_2 .

Capítulo 4

PDP - uma Abordagem Multicritério

4.1 Introdução

O Planeamento Agregado da Produção em empresas de produção por encomenda implica a optimização simultânea de objectivos múltiplos e frequentemente conflituosos (ver [Carravilla and Sousa, 1995]). Para além disso, o carácter combinatório destes problemas torna a sua resolução difícil e, mesmo quando se considera apenas um objectivo, encontrar uma solução admissível aceitável pode, só por si, constituir uma tarefa extremamente complicada. Se se pretender fornecer apenas uma solução ao Agente de Decisão (AD), é geralmente possível optimizar segundo um ou outro objectivo, utilizando um método analítico de resolução.

No caso de se usar um método heurístico, é sempre possível “moldar” a heurística de modo a privilegiar um dos objectivos. No entanto, a percepção do problema pelo AD aumenta, se este dispuser de uma visão holística sobre as alternativas em jogo, que lhe permita conhecer os limites entre os quais os diversos objectivos podem variar, e isso só é possível se lhe for disponibilizado um conjunto de soluções eficientes representativas da gama de variação de cada um dos objectivos.

Por outro lado, como é reconhecido, há limites para o número de alternativas que o AD pode avaliar em simultâneo, o que implica que o AD

deve ser apoiado na escolha da “melhor” solução. Esse apoio pode passar pelo fornecimento de elementos que lhe permitam reunir soluções em subconjuntos com características semelhantes, de modo a reduzir o número de alternativas que o AD tem que analisar em simultâneo.

Neste capítulo, estudar-se-á o problema PDP do Caso de Estudo, segundo este ponto de vista. Começa-se por apresentar duas metodologias (analíticas) multicritério para obtenção das soluções eficientes segundo os dois objectivos mais importantes do problema considerado. Apresentam-se, a seguir, alterações feitas à heurística *H-PDP* descrita na secção 3.3, de modo a ter em conta os dois objectivos mais importantes e a sua integração numa “heurística multiobjectivo”. Comparam-se em seguida as soluções obtidas pelos diferentes algoritmos, num problema gerado aleatoriamente para esse efeito. Por último, tecem-se algumas considerações acerca do processo de decisão do AD, face ao conjunto de soluções que lhe são apresentadas.

4.2 Conceitos fundamentais em problemas de decisão multiobjectivo

Tal como já foi referido no Capítulo 1, no âmbito dos Problemas de Decisão Multicritério, a denominação genérica de Multicritério abrange dois tipos de problemas, os problemas Multiobjectivo e os problemas Multiatributo. Um problema de decisão com objectivos múltiplos ou Multiobjectivo, caracteriza-se pela existência de um conjunto de soluções admissíveis definidas através de um conjunto de restrições e onde os objectivos são explicitados através de funções objectivo. Em oposição, designam-se por problemas Multiatributo aqueles em que o número de alternativas é limitado e os atributos são conhecidos explicitamente.

Nesta secção serão apresentados alguns conceitos fundamentais em problemas de decisão Multiobjectivo, com base em [Chankong and Haimes, 1983], [Matos, 1988] e [Ulungu and Teghem, 1994].

Um problema de decisão Multiobjectivo (*PMO*), pode ser genericamente descrito do seguinte modo:

$$\begin{aligned}
PMO \quad & \forall_{k=1,\dots,p} \quad \min_x f_k(x) \\
\text{sujeito a} \quad & x \in \mathcal{X} \\
& \mathcal{X} = \{x \mid \text{verifica restrições}\}
\end{aligned} \tag{4.1}$$

Trata-se de um problema com p funções objectivo f_k , em que as soluções admissíveis têm que verificar um conjunto de restrições (como, em geral, em qualquer problema de Programação Matemática) que podem incluir restrições de integralidade para parte das variáveis.

Na situação habitual, em que os objectivos são conflituosos, não existe nenhuma solução x que minimize simultaneamente todos os objectivos de (4.1). É, no entanto, possível definir o conjunto de soluções eficientes (ou não dominadas ou óptimas de Pareto) da seguinte forma:

Uma solução $x^* \in \mathcal{X}$ diz-se eficiente se não existir outra solução $x' \in \mathcal{X}$ tal que $\forall_{k=1,\dots,p} f_k(x') \leq f_k(x^*)$, com pelo menos uma desigualdade estrita. Nesse caso, o vector $f(x^*)$, dos valores $f_k(x^*) \forall_{k=1,\dots,p}$, diz-se não dominado no espaço das funções objectivo.

Os problemas de decisão Multiobjectivo são, em geral, abordados através de métodos que se baseiam na Função Utilidade do AD, de métodos interactivos ou de métodos de geração.

Métodos baseados na Função Utilidade do AD

Em todos os processos de decisão, está implícita a necessidade de construir, directa ou indirectamente, uma estrutura de preferências, para que seja possível ordenar alternativas e fazer a escolha final. Os métodos que se baseiam no conhecimento da Função Utilidade, partem da premissa de que a estrutura de preferências do AD pode ser representada por uma função a que se chama Função Utilidade. Conhecendo então a Função Utilidade, e na ausência de incerteza, uma alternativa com valores mais altos da Função Utilidade representaria a solução preferida. No caso de problemas de decisão Multiobjectivo, a Função Utilidade é uma função multivariável, difícil de construir, mesmo que se tente reduzir a dimensão do problema, agrupando vários atributos em grupos.

Outro método para simplificar o problema, baseia-se na construção de uma Função Utilidade para cada objectivo e adição posterior dos resultados, o que implica que se pode aproximar a Função Utilidade do AD por uma Função Utilidade aditiva. No entanto, mesmo assim, este método exige uma interacção inicial muito grande entre o analista (cuja função é ajudar a resolver o problema) e o AD, e baseia-se na premissa de que a estrutura de preferências do AD não se altera ao longo do processo de decisão.

Métodos interactivos

Os métodos interactivos pressupõem, em geral, uma participação activa do analista e baseiam-se na interactividade entre este e o AD, ao longo do processo de decisão. Começa-se por gerar uma solução admissível (e se possível eficiente) para o problema e obtêm-se as reacções do AD a essa solução. As reacções do AD são depois utilizadas como parâmetros ou novas restrições para o problema, que será resolvido de novo. Este processo é repetido até que o AD esteja “satisfeito” com a solução obtida ou até não ser possível gerar mais nenhuma solução. Este método, dada a grande interactividade exigida entre o analista e o AD, só pode ser utilizado para obter a solução de problemas de decisão mais estratégicos e que por isso surjam com uma periodicidade baixa.

Métodos de geração

Os métodos de geração, por seu lado, baseiam-se na geração do conjunto das soluções eficientes do problema (ou de um seu subconjunto), que pode ser obtido independentemente do AD, dado que não depende da sua estrutura de preferências. Depois de obtido este conjunto, aqui designado por $\mathcal{E}(PMO) \subseteq \mathcal{X}$, utiliza-se um procedimento adicional que apoiará o AD na escolha da “melhor” solução.

Dos três métodos descritos, e dadas as características do problema em estudo, optou-se então pelo método de geração. Põe-se portanto o “sub-problema” de gerar $\mathcal{E}(PMO)$.

Apresentam-se a seguir dois conceitos nos quais se baseiam dois métodos analíticos que permitem obter $\mathcal{E}(PMO)$.

O conceito de **equivalente escalar paramétrico**, decorre do **Teorema de Geoffrion**, e consiste no seguinte:

Se o conjunto das soluções admissíveis $\mathcal{X}$ for convexo e todas as funções objectivo f_k forem côncavas, então essa é uma condição necessária e suficiente para que, se x for solução óptima de:

$$\begin{aligned} \min_{x \in \mathcal{X}} \quad & \sum_{i=1}^p \lambda_i f_i(x) \\ \text{sujeito a} \quad & \sum_{i=1}^p \lambda_i = 1 \quad \lambda_i > 0 \\ & \mathcal{X} = \{x | \text{verifica restrições}\} \end{aligned} \quad (4.2)$$

seja solução eficiente de PMO .

Para cada $\bar{\lambda}$ ($\bar{\lambda} = [\lambda_1 \dots \lambda_p]$), há uma solução eficiente, e gerando todos os $\bar{\lambda}$ (se tal for possível) é obtido o conjunto completo das soluções eficientes de PMO , $\mathcal{E}(PMO)$ (conceito de **equivalente escalar paramétrico**).

Note-se que, se parte das variáveis x forem inteiras (ou, em particular, binárias) então o conjunto das soluções admissíveis $\mathcal{X}$ pode não ser convexo, e seguindo o processo descrito pode ser apenas obtido um subconjunto do conjunto das soluções eficientes $\mathcal{E}(PMO)$.

Nesse caso, para se obter o conjunto das soluções eficientes, é necessário utilizar o **equivalente escalar por restrições**:

$$\begin{aligned} \min_{x \in \mathcal{X}} \quad & f_i(x) \\ \text{sujeito a} \quad & \forall_{j=1, \dots, p \wedge j \neq i} f_j(x) \leq f_j^o \\ & \mathcal{X} = \{x | \text{verifica restrições}\} \end{aligned} \quad (4.3)$$

Demonstra-se que, fixados $f_j^o \forall_{j=1, \dots, p \wedge j \neq i}$, uma solução $x \in \mathcal{E}(PMO)$, se e só se for solução óptima de (4.3) (ver [Chankong and Haimes, 1983, Cap4]).

4.3 Objectivos do problema PDP

Em termos gerais, os objectivos que se podem considerar na prática para o problema PDP são os seguintes:

- produção o mais perto possível da data de entrega (*JIT*), que é modelado pela seguinte expressão (ver secção 3.2.1):

$$JIT \quad \min \quad \sum_{e,l,t \leq d_e} x_{elt}(d_e - t) + \sum_{e,l,t=d_e+1} 6x_{elt} + \sum_{e,l,t \geq d_e+2} x_{elt}(t - d_e)^a \quad (4.4)$$

- minimização do somatório dos índices de preferência:

$$PREF \quad \min \quad \sum_{e,l,t} x_{elt} * P_{jlt_{comj:\tau_e=j}} \quad (4.5)$$

- minimização do espalhamento (número de linhas e períodos diferentes em que é produzida cada encomenda):

$$ESP \quad \min \quad \sum_{e,l,t} \gamma_{elt} \quad (4.6)$$

- minimização do maior atraso:

$$AtrasoMax \quad \min \quad \max_{e,l,t} (t * \gamma_{elt} - d_e) \quad (4.7)$$

Dos objectivos apresentados, serão, no que se segue, apenas analisados os objectivos *JIT* e *PREF*, e em algumas situações serão fixados valores para o objectivo *ESP*.

4.4 Métodos analíticos

O método paramétrico e o método das restrições, descritos nesta secção, são dois métodos analíticos multiobjectivo, que se baseiam respectivamente nas noções de “equivalente escalar paramétrico” e de “equivalente escalar por restrições”, apresentadas na secção 4.2.

4.4.1 Método paramétrico

Na implementação do método paramétrico no caso do problema PDP, foi utilizado o modelo de Programação Matemática descrito na secção 3.2, com a função objectivo $\lambda_1 * JIT + \lambda_2 * PREF$.

Para se obter um conjunto de soluções eficientes, foi executado o Algoritmo 4.1, implementado em C, que utiliza a biblioteca de funções de programação matemática do CPLEX (para uma descrição pormenorizada da plataforma utilizada, ver o Apêndice E.1).

```

para ( $\lambda_1 = 0; \lambda_1 \leq 1; \lambda_1 = \lambda_1 + \delta$ )
     $\lambda_2 = 1 - \lambda_1$ 
    resolver Modelo PDP com função objectivo  $\lambda_1 * JIT + \lambda_2 * PREF$ 

```

Algoritmo 4.1: Método paramétrico

Dado que no modelo PDP há várias variáveis inteiras, o conjunto das soluções admissíveis poderá não ser convexo, o que implica que, com este método, poderá não se obter o conjunto completo das soluções eficientes, mesmo que se use um valor muito pequeno para δ .

4.4.2 Método das restrições

Na implementação do método das restrições, foi utilizado o modelo de Programação Matemática descrito na secção 3.2, ao qual foi acrescentada uma restrição que impõe valores limite para o objectivo $PREF$. Manteve-se JIT como função objectivo.

Para obter um conjunto de soluções eficientes, variaram-se os valores limite para $PREF$, de acordo com o Algoritmo 4.2, que foi implementado em C e utiliza a biblioteca de funções de programação matemática do CPLEX (para uma descrição pormenorizada da plataforma utilizada, ver o Apêndice E.1).

Os valores min e max entre os quais varia $limsup_{PREF}$, foram obtidos em testes realizados previamente, e que se passam a descrever: começa-se por obter o valor da função objectivo JIT , com $limsup_{PREF}$ muito grande, seguidamente reduz-se $limsup_{PREF}$. O valor max corresponde ao menor valor de $limsup_{PREF}$ para o qual JIT não se alterou. O valor min , por

outro lado, corresponde ao valor mínimo de $\limsup_{PREF}$, para o qual o problema tem solução admissível.

para ($\limsup_{PREF} = \min; \limsup_{PREF} \leq \max; \limsup_{PREF} = \limsup_{PREF} + \delta$)
 resolver Modelo PDP com função objectivo *JIT* e
 $PREF \leq \limsup_{PREF}$

Algoritmo 4.2: Método das restrições

Neste caso um valor adequado do passo δ , permitiria (teoricamente) obter todas as soluções eficientes.

4.5 Heurística multiobjectivo

Em ambos os métodos analíticos descritos na secção anterior, é necessário, em cada ciclo do algoritmo, resolver o problema PDP por programação matemática o que pode ser possível em instâncias de pequena dimensão, mas se torna inviável em instâncias reais (ver secção 3.2.3). Os métodos analíticos descritos tornam o problema PDP tratável do ponto de vista dos objectivos múltiplos, mas as dificuldades de resolução por programação matemática mantêm-se.

Este facto levou ao desenvolvimento de uma heurística multiobjectivo que consiste, essencialmente, num processo iterativo sobre duas heurísticas mais elementares. Uma das heurísticas (*H-JIT*) privilegia o objectivo (4.4), procurando colocar as encomendas o mais próximo possível das datas de entrega. A outra heurística (*H-PREF*) privilegia o objectivo (4.5), procurando que as encomendas sejam produzidas nas linhas com maior preferência. Trata-se de heurísticas construtivas, em que as encomendas são planeadas pela ordem em que são introduzidas no sistema. No entanto, para evitar que esta ordem seja um factor determinante na colocação das encomendas, dando origem apenas a um pequeno número de soluções diferentes, na heurística multiobjectivo (*H-MO*), começa-se, em cada iteração, por reordenar aleatoriamente as encomendas.

A heurística *H-MO* (Algoritmo 4.5) é um processo iterativo que executa de uma forma aleatória cada uma das sub-heurísticas referidas. A variação

do parâmetro p ao longo da execução do algoritmo (assumindo valores entre 0 e 1), leva a que sejam geradas soluções cobrindo toda a gama entre “puramente” *JIT* ($p = 1$) e “puramente” *PREF* ($p = 0$), esperando-se deste modo obter um conjunto significativo de soluções eficientes.

A heurística que privilegia o *JIT* (*H-JIT*) é a heurística base desenvolvida para o problema PDP (*H-PDP*) (ver secção 3.3), descrita de novo sucintamente no Algoritmo 4.3. Foi também desenvolvida uma heurística que privilegia o objectivo *PREF* (*H-PREF*), que é descrita no Algoritmo 4.4, e com a qual se pretende planear cada encomenda nas linhas preferidas, prescindindo de planear no menor número possível de períodos e, se necessário, prescindindo mesmo de planear antes da data de entrega.

```

DeterminaLimites(e)
  se (ColocaMaisTarde(e) = 1)
    próxima e
  se não
    se (RelaxaLimites(e) = 1)
      próxima e

```

Algoritmo 4.3: Heurística *H-JIT*

```

para ( $nlin = 1; nlin < nlinmax; nlin = nlin + 1$ )
  para ( $maxper = d_e; maxper \leq T; maxper = maxper + 1$ )
    para ( $nper = 1; nper \leq maxper - r_e + 1; nper = nper + 1$ )
      para ( $t_1 = maxper - nper + 1; t_1 \geq r_e; t_1 = t_1 - 1$ )
        se (CapProdTip( $\tau_e, nper, t_1$ )  $\geq q_e$ )
          GeraLista
            se (CapProdLin( $\tau_e, nlin, nper, t_1$ )  $\geq q_e$ )
              CarregaLinhas( $\tau_e, nlin, nper, t_1, 2$ )
              próxima e

```

Algoritmo 4.4: Heurística *H-PREF*

Finalmente, no Algoritmo 4.5 é descrita a heurística *H-MO*, construída de acordo com a ideia atrás apresentada.

Segue-se uma descrição sucinta das funções mencionadas nos Algoritmos *H-JIT* e *H-PREF* (para uma descrição completa destas funções, ver Apêndice B).

DeterminaLimites(e) permite que o utilizador controle o tipo de solução que irá obter, através de parâmetros que limitam o “espalhamento”

```

para ( $p = 0; p \leq 1; p = p + \delta$ )
  gerar sequência aleatória das encomendas
  para cada encomenda
    gerar um número aleatório Aleat ( $0 \leq Aleat \leq 1$ )
    se ( $Aleat \leq p$ )
      H-JIT
    se ( $Aleat > p$ )
      H-PREF

```

Algoritmo 4.5: Heurística multiobjetivo (*H-MO*)

das encomendas pelas linhas e períodos e o número de períodos entre o início da produção e a data de entrega d_e .

ColocaMaisTarde(e) coloca a encomenda e o mais tarde possível, no menor número de períodos possível e no menor número de linhas possível (tendo em conta os índices de preferência, as capacidades disponíveis e os limites determinados em **DeterminaLimites(e)**).

RelaxaLimites(e) identifica um conjunto de períodos e um conjunto de linhas para produzir e , mesmo que para isso devam ser relaxados alguns dos limites calculados inicialmente (relaxando mesmo, se necessário, a data de entrega, e em último caso, tentando produzir uma quantidade menor que a quantidade encomendada).

CapProdTip(j, nper, t₁) retorna a capacidade existente entre um período inicial t_1 e um período final $t_1 + nper - 1$, para produzir o tipo j sem ter em conta as linhas em que se pretende produzir.

CapProdLin(j, nlin, nper, t₁) retorna a capacidade existente nas linhas pertencentes a *Lista*, para produzir o tipo j , entre um período inicial t_1 e um período final $t_1 + nper - 1$.

CarregaLinhas(e, nlin, nper, t₁, caso) carrega as linhas incluídas em *Lista* com a encomenda e .

Se $caso = 1$, carrega as linhas de t_1 até $t_1 + nper - 1$, se $caso = 2$, carrega as linhas de $t_1 + nper - 1$ até t_1 .

4.6 Testes computacionais

4.6.1 Geração de um problema teste

Foi gerada aleatoriamente uma instância teste para o problema, tendo em conta que por um lado devia ser representativa dos problemas reais e que por outro lado devia ser suficientemente pequena para ser possível resolvê-la usando o CPLEX (ver o Apêndice E.1, para obter informações acerca deste software de optimização), uma vez que se pretendia comparar os resultados da heurística com os de um método analítico de referência. Além disso, nos dados de entrada, foi considerado um período adicional no horizonte temporal, com grande capacidade e com preferência diferente de zero para todos os tipos de produtos numa das linhas, de modo a que existisse sempre uma solução admissível para o problema.

A instância teste (PDP1) está descrita no Apêndice F.1.2 e as suas características mais importantes reproduzem-se na Tabela 4.1.

T - número de períodos	14
E - número de encomendas	14
L - número de linhas	4
F - número de fábricas	3
J - número de tipos de produtos	8

Tabela 4.1: Características da instância PDP1

4.6.2 Resultados obtidos pela heurística *H-PDP*

Aplicando a heurística *H-PDP* (descrita na secção 3.3) à instância PDP1, obteve-se a solução representada na Tabela 4.2.

Apresentando ao Agente de Decisão (AD) apenas uma solução, tal como a representada na Tabela 4.2, o problema de decisão do AD reduz-se à possibilidade de aceitar ou rejeitar a solução. Nesse caso, o AD não adquire nenhum conhecimento extra do problema que lhe permita ter noção de “valor de troca” (“trade-off”) entre objectivos, para que possa avaliar quanto tem de perder num dos objectivos para poder melhorar o outro. A solução apresentada na Tabela 4.2 não lhe permite saber se poderia melhorar *PREF* com uma pequena perda em *JIT*, e também até que valor pode melhorar

JIT, e quais as implicações dessa melhoria em *PREF*. Para que o AD adquira esse conhecimento é necessário, se possível, fornecer-lhe um conjunto de soluções que seja representativo do espaço das soluções.

e	τ_e	r_e	d_e	q_e	$x_{e,l,t}$		
1	6	1	14	3915	$x_{1,1,14}$	=	3915
2	1	11	12	3333	$x_{2,3,12}$	=	3333
3	4	6	10	4575	$x_{3,3,10}$	=	4575
4	7	3	14	1503	$x_{4,1,14}$	=	1503
5	3	1	12	906	$x_{5,3,12}$	=	906
6	3	5	14	3027	$x_{6,1,14}$	=	3027
7	6	5	10	1579	$x_{7,1,10}$	=	1579
8	4	3	13	4823	$x_{8,4,12}$	=	3241
					$x_{8,4,13}$	=	1582
9	8	10	14	4764	$x_{9,1,14}$	=	4764
10	1	4	9	2685	$x_{10,3,9}$	=	2685
11	7	2	13	152	$x_{11,3,13}$	=	152
12	5	7	9	2228	$x_{12,3,9}$	=	2228
13	7	8	14	4899	$x_{13,1,14}$	=	4899
14	6	3	5	3106	$x_{14,1,4}$	=	3106
<i>JIT</i>							6347
<i>PREF</i>							138507
<i>ESPALH</i>							15
<i>AtrasoMax</i>							0
<i>NumEncAtras</i>							0

Tabela 4.2: Solução de PDP1 usando a heurística *H-PDP*

4.6.3 Resultados obtidos pela heurística multiobjetivo

Aplicando a heurística *H-MO* (descrita no Algoritmo 4.5) com $\delta = 0.01$, foram obtidos os valores para *JIT* e *PREF* representados na Figura 4.1, onde o valor de *JIT* está representado no eixo horizontal e o valor de *PREF* no eixo vertical. Distinguiram-se os pontos correspondentes a soluções com *ESP* = 14 e 15, embora essa terceira dimensão não seja explicitamente considerada na análise.

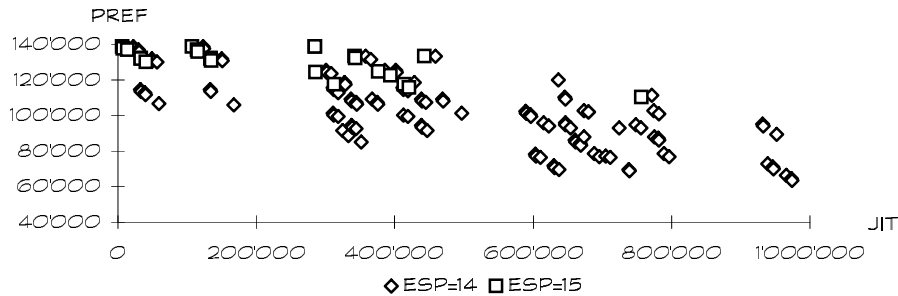


Figura 4.1: Valores de JIT e $PREF$, nas soluções da heurística multiobjetivo ($H-MO$)

Como se pode ver na Figura 4.1, a heurística gera um número importante de soluções, incluindo soluções eficientes e soluções dominadas. Na Figura 4.2, estão representados apenas os valores das funções objectivo para as soluções eficientes. Analisando os valores representados na Figura 4.2, pode-se verificar que ao transitar de uma solução eficiente para outra, se melhora um objectivo, piorando o outro. Note-se também que essa “troca” não é constante: na zona de menor JIT , $PREF$ varia mais rapidamente com alterações em JIT , passando-se o contrário na zona de maior JIT .

Na Tabela 4.3, estão representados os valores dos objectivos (incluindo alguns daqueles que não são explicitamente considerados nesta abordagem) para todas as soluções eficientes obtidas por $H-MO$.

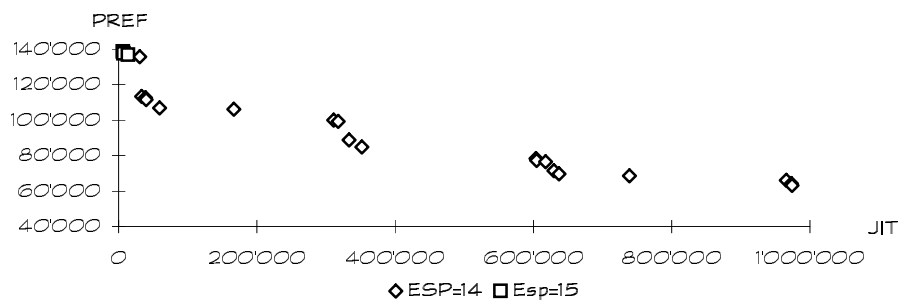


Figura 4.2: Valores de JIT e $PREF$, nas soluções eficientes obtidas por $H-MO$

<i>JIT</i>	<i>PREF</i>	<i>ESP</i>	<i>AtrasoMax</i>
6347	138507	15	0
7259	137443	15	1
13595	136695	15	2
30310	135631	14	2
32956	113328	14	1
39292	112580	14	2
40204	111516	14	2
59620	106662	14	2
167012	105914	14	4
311456	99960	14	5
317792	99212	14	5
333004	88641	14	4
351508	84851	14	4
603344	78149	14	5
604256	77085	14	5
610592	76336	14	5
630008	71483	14	5
637256	69671	14	5
739224	68607	14	5
965633	66113	14	5
972881	64301	14	5
973793	63237	14	5

Tabela 4.3: Valores dos objectivos nas soluções eficientes obtidas por *H-MO*

4.6.4 Resultados obtidos pelo método das restrições

Tal como foi referido antes, dado que algumas variáveis de decisão do modelo PDP são inteiras, o conjunto das soluções eficientes não é convexo e com o método paramétrico obter-se-ia apenas um número reduzido de soluções eficientes.

Para comparação de resultados, optou-se então por resolver a instância PDP1 usando o método das restrições descrito no Algoritmo 4.2, fazendo variar o limite superior para *PREF* ($limsup_{PREF}$) entre 138507 e 41495 (limites determinados em testes realizados previamente). Os resultados foram obtidos com $ESP = 14$, para se poder fazer uma comparação entre as soluções eficientes obtidas por este método e as soluções obtidas pela heurística multiobjectivo. As soluções eficientes obtidas estão representadas no espaço dos objectivos na Figura 4.3. Na Tabela 4.4, estão representados os valores dos objectivos (*JIT*, *PREF*, *ESP* e *AtrasoMax*) para as soluções eficientes.

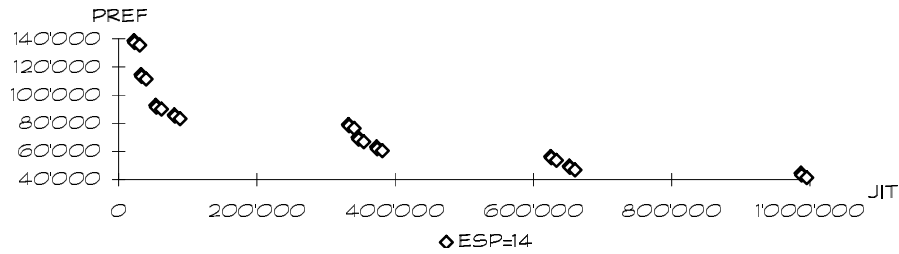


Figura 4.3: Valores de *JIT* e *PREF*, nas soluções eficientes obtidas pelo método das restrições

<i>JIT</i>	<i>PREF</i>	<i>ESP</i>	<i>AtrasoMax</i>
22150	138507	14	0
23062	137443	14	1
30310	135631	14	1
32044	114392	14	1
32956	113328	14	1
40204	111516	14	1
54698	91586	14	2
61946	89774	14	2
80450	85984	14	3
81326	84920	14	2
88610	83108	14	2
332286	79282	14	5
340446	76406	14	5
346586	69775	14	4
347498	68711	14	4
354746	66899	14	4
373250	63109	14	4
374162	62045	14	4
381410	60233	14	4
625086	56407	14	5
633246	53531	14	5
651750	49741	14	5
652662	48677	14	5
659910	46865	14	5
988287	43307	14	5
995535	41495	14	5

Tabela 4.4: Valores dos objectivos nas soluções eficientes obtidas pelo método das restrições

4.6.5 Comparação de resultados

Como pode ser verificado na Figura 4.4, as soluções obtidas para a instância PDP1 pela heurística multiobjectivo e pelo método das restrições agrupam-se em conjuntos que correspondem a valores próximos, em termos do objectivo *JIT*, e que devem corresponder a soluções do mesmo “tipo”. As soluções obtidas pela heurística multiobjectivo são no entanto sempre dominadas pelas soluções obtidas pelo método das restrições, dado que, para o mesmo valor de *JIT*, o valor de *PREF* é mais elevado. Esse facto indicia que ainda é necessário melhorar a heurística *H-PREF*. No entanto, e embora susceptíveis de melhoria, estes resultados mostram já que esta abordagem pode ser utilizada neste tipo de problemas. Convém relembrar que a dimensão da instância PDP1 é muito menor que a das instâncias reais, e as instâncias reais nunca poderiam ser resolvidas por métodos analíticos, dado o seu carácter combinatório.

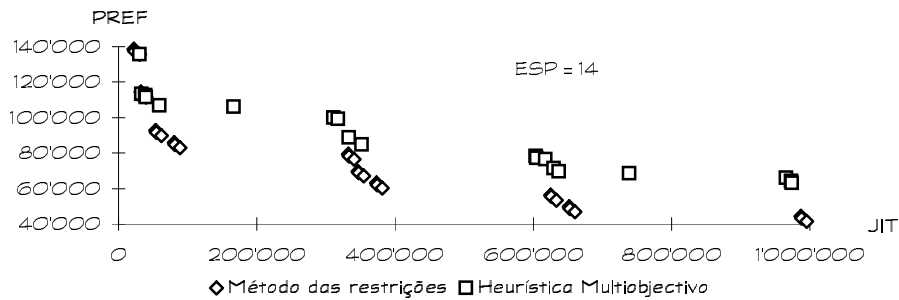


Figura 4.4: Comparação das soluções obtidas pelo método das restrições com as soluções obtidas por *H-MO* ($ESP = 14$)

4.7 Apoio na tomada de decisão

Após a obtenção de um conjunto de soluções, quer por métodos analíticos, quer recorrendo a uma heurística multiobjectivo, estas devem ser apresentadas ao Agente de Decisão (AD), que as avaliará e decidirá qual a “melhor” no momento. No entanto, o número de soluções a apresentar ao AD pode ser mesmo assim muito elevado e é reconhecido que há limites para o número de

alternativas diferentes que o AD pode avaliar em simultâneo. Uma primeira redução do conjunto das soluções é obtida retirando as soluções dominadas do conjunto, dado que não é aceitável, em nenhuma circunstância, que o AD prefira uma solução dominada.

Do ponto de vista do processo de decisão é importante, por um lado, manter uma visão global das alternativas para que a percepção do problema pelo AD aumente, mas, por outro lado, é necessário fornecer ao AD elementos que lhe permitam reunir soluções em subconjuntos com características semelhantes, de modo a reduzir o número de alternativas a analisar.

Relativamente à manutenção de uma visão global das alternativas, a apresentação ao AD de um conjunto de soluções “representativas” permite-lhe ter noção dos valores entre os quais os objectivos podem variar, permitindo-lhe também ganhar sensibilidade relativamente ao “valor de troca” entre objectivos. A partir da Figura 4.2, o AD pode, por exemplo, analisar de quanto tem que prescindir no objectivo *JIT*, para poder melhorar o objectivo *PREF* de $x\%$. Pode ainda, numa primeira fase, decidir qual a “zona de objectivos” mais interessante, quer porque não pretende exceder um determinado intervalo para o valor de um objectivo, quer porque conhece as características das soluções nessa zona. Por exemplo, limitando os valores de *JIT* a 400 000 na Figura 4.2, obtém-se a Figura 4.5.

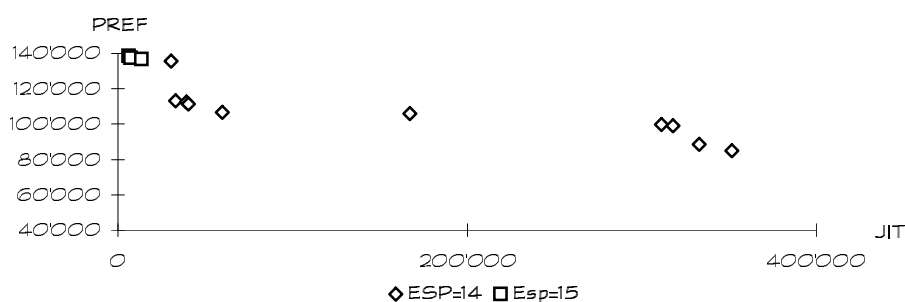


Figura 4.5: Valores dos objectivos *JIT* e *PREF*, nas soluções eficientes obtidas por *H-MO* (*JIT* limitado a 400 000)

Note-se ainda que, relativamente a cada um dos subconjuntos de soluções assim identificadas, deve-se evitar apresentar ao AD soluções com pequenas

diferenças em termos numéricos, mesmo que sejam todas não dominadas, dada a incapacidade do AD, na discriminação dessas soluções. Pode-se, assim, fazer a aglomeração (“clustering”) de soluções com valores semelhantes de *JIT* e de *PREF* (correspondendo por exemplo aos “grupos” visíveis nas Figuras 4.2 e 4.5).

Capítulo 5

Programação da Produção

5.1 Introdução

Uma organização hierárquica do processo de planeamento é tanto mais adequada, quanto mais possível for “separar” as suas diferentes fases (ou níveis). No contexto deste trabalho, procurou-se, nesse sentido, garantir um elevado grau de separabilidade entre os vários módulos, por forma a minimizar as dependências e a simplificar o processo de tomada de decisões. Assim, após a afectação das encomendas a cada fábrica pelo módulo PDP, estas são associadas pelo Sistema Informático de Base (SIB), formando conjuntos que correspondem ao mesmo artigo, à mesma cor e a datas de entrega “próximas”. Um conjunto desses, com cerca de 400 pares, constitui uma Ordem de Fabrico (OF).

Com o processo de planeamento aqui designado por Programação (PROG), é feita a alocação das OF previamente afectadas a uma dada fábrica, às linhas de montagem e de corte/costura. Por razões operacionais, uma OF deve ser produzida numa única linha, tanto de montagem como de costura. Considera-se que cada linha de produção tem capacidades e índices de “preferência” diferenciados para cada tipo de produto. Considera-se ainda que para a Costura, existe uma linha suplementar (“fictícia”) com capacidade infinita para qualquer tipo de produto, e preferência baixa (ou seja, menos desejável que as outras). Essa linha corresponde à subcontratação de costura. A cada OF está associada uma data mínima para entrada em

produção, uma data de entrega pretendida, um tipo de produto e uma quantidade. Pretende-se com o PROG determinar as datas de entrada em Corte/Costura e em Montagem de cada OF. Para facilitar o controlo de produção, pretende-se na prática que cada OF seja afectada, no máximo, a dois períodos consecutivos.

A afectação das OF às linhas de montagem é feita do mesmo modo que no módulo PDP (pela heurística *H-PDP*, ver secção 3.3), com a restrição adicional de que cada OF só pode ser produzida numa linha. Assim, neste capítulo, serão apenas apresentadas as questões associadas ao problema de afectação das OF às linhas de costura, dado que se pretendeu manter o conceito de linhas “dedicadas” e de linhas de “diversos” existente na empresa (ver secção 2.2).

Após a determinação das datas de entrada em Corte/Costura e em Montagem, é calculado o número de máquinas de cada tipo que são necessárias em cada linha e em cada período do horizonte temporal, por forma a conseguir produzir as OF e, simultaneamente, de modo a não exceder o número de máquinas existentes e o número máximo de postos permitidos em cada linha, por limitações físicas. Esta “componente” do problema será aqui designada por problema de “layout”.

Neste capítulo começa-se por apresentar um modelo de Programação Matemática desenvolvido para o problema PROG. A existência de várias funções objectivo associadas ao modelo e ainda o elevado número de variáveis inteiras a ele associadas, originou o desenvolvimento de uma abordagem que se baseia na decomposição do problema em duas partes e na relaxação linear de algumas das variáveis. Essa abordagem heurística, a que se deu o nome de *H-PROG1*, encontra-se descrita na secção 5.3. Dado que os modelos parciais são ainda de grande dimensão para problemas reais, foi necessário desenvolver, para o Caso de Estudo, uma heurística que não recorre explicitamente a modelos de Programação Matemática, e que é descrita na secção 5.4. A essa heurística deu-se o nome de *H-PROG*.

5.2 Modelo

Nesta secção, descreve-se um modelo de Programação Matemática envolvendo fundamentalmente as seguintes entidades:

x_{elt} quantidade da OF e a produzir na linha l no período t ;

k_{mlt} número de máquinas do tipo m a colocar na linha l no período t .

(Para uma descrição mais completa da notação utilizada, ver Apêndice A)

O modelo de base é muito semelhante ao desenvolvido para o problema PDP (ver secção 3.2), tendo sido acrescentadas restrições relativas às máquinas e às linhas, e restrições que impedem que uma OF seja colocada em mais do que dois períodos consecutivos.

5.2.1 Funções objectivo

Para este problema, podem-se considerar vários objectivos em simultâneo. Com a função objectivo apresentada na equação (5.1) pretende-se que cada OF seja produzida antes da data de entrega e o mais perto possível desta.

No Caso de Estudo, existe uma linha de pré-costura, por onde passam todos os produtos, e várias linhas de costura que funcionam em paralelo. A cada linha está associado um número de ordem, a linha 1 é a linha de pré-costura, e o número de ordem das restantes linhas cresce com a diversidade de produtos que se pretende que sejam produzidos nessa linha. Com a função objectivo (5.2) pretende-se que as OF sejam produzidas nas linhas de maior “preferência” e nas linhas com menor número de ordem. A preferência, que está associada à adequação “tecnológica” das linhas para produzir cada tipo de artigo, é medida através de um índice. Um índice de preferência 0 implica sempre uma capacidade de produção nula. Os índices de preferência são valores inteiros não negativos e a preferência diminui com o valor do índice.

Com a função objectivo (5.3) pretende-se minimizar o número de alterações ao “layout” (considera-se que retirar ou colocar uma máquina numa linha num dado período é uma alteração ao “layout”).

$$\min \sum_{e,l,t} x_{elt}(d_e - t) \quad (5.1)$$

$$\min \sum_{e,l,t} l * x_{elt} * P_{j:\tau_e=j}lt \quad (5.2)$$

$$\min \sum_{m,l,t} |k_{mlt} - k_{ml(t-1)}| \quad (5.3)$$

5.2.2 Restrições

$$\forall_e \quad \sum_t x_{elt} = q_e \quad (5.4)$$

$$\forall_e \quad \sum_{l>1} \sum_t x_{elt} = q_e \quad (5.5)$$

$$\forall_{j,t} \quad \sum_{e:\tau_e=j} \sum_l x_{elt} \leq U_{jt} \quad (5.6)$$

$$\forall_{l,t} \quad \sum_{e:\tau_e=j} \sum_{j:C_{jlt}>0} \frac{x_{elt}}{C_{jlt}} \leq 1 \quad (5.7)$$

$$\forall_e \quad \sum_l \delta_{el} \leq 1 \quad (5.8)$$

$$\forall_{e,l} \quad \sum_t x_{elt} - q_e \delta_{el} \leq 0 \quad (5.9)$$

$$\forall_{m,t} \quad \sum_{e:\tau_e=j} \sum_j p_{1jm} x_{elt} - k_{m1t} b_t \leq 0 \quad (5.10)$$

$$\forall_{m,l \neq 1,t} \quad \sum_{e:\tau_e=j} \sum_j p_{2jm} x_{elt} - k_{mlt} b_t \leq 0 \quad (5.11)$$

$$\forall_{m,t} \quad \sum_l k_{mlt} \leq K_m \quad (5.12)$$

$$\forall_{l,t} \quad \sum_m k_{mlt} \leq D_l \quad (5.13)$$

$$\forall_{e,t} \quad \sum_{t'=1}^t \sum_{l=2}^L x_{elt'} - \sum_{t'=1}^t x_{elt'} \leq 0 \quad (5.14)$$

$$\forall_{e,l,t} \quad x_{elt} - lb \gamma_{elt} \geq 0 \quad (5.15)$$

$$\forall_{e,l,t} \quad x_{elt} - q_e \gamma_{elt} \leq 0 \quad (5.16)$$

$$\forall_{e,l,t' \in \{0,1\}} \quad \sum_{n=0:t+2n \leq T}^N \gamma_{el(t+2n)} \leq 1 \quad (5.17)$$

$$\forall_{e,l,t' < T-2} \quad \sum_{n=0:t+2n-1 \geq 0}^N (\gamma_{el(t-2n)} + \gamma_{el(t-2n-1)}) \geq 0 \quad (5.18)$$

$$\forall_{e,l,t \notin I_e} \quad \text{se } P_{j:\tau_e=j}lt > 0 \text{ e } C_{j:\tau_e=j}lt > 0 \Rightarrow x_{elt} \in \{0,1\} \quad (5.19)$$

$$\forall_{e,l} \quad \delta_{el} \in \{0,1\} \quad (5.20)$$

$$\forall_{e,l,t} \quad \gamma_{elt} \in \{0,1\} \quad (5.21)$$

$$\forall_{m,l,t} \quad k_{mlt} \geq 0 \text{ inteiro} \quad (5.22)$$

As restrições (5.4) garantem que para cada OF e , a quantidade produzida na pré-costura é igual à quantidade encomendada.

As restrições (5.5) garantem que para cada OF e , a quantidade produzida na costura é igual à quantidade encomendada.

As restrições (5.6) garantem que em cada período t , a produção do produto tipo j não excede um limite superior previamente definido.

As restrições (5.7) garantem que a capacidade produtiva de cada linha não é excedida em nenhum período (para uma interpretação mais detalhada desta equação, recomenda-se a leitura da explicação associada à Tabela 5.1).

As restrições (5.8) e (5.9) garantem que cada OF e é produzida apenas numa linha l .

As restrições (5.10) e (5.11) garantem que em cada linha l e em cada período t , existe sempre um número suficiente de máquinas de cada tipo m .

As restrições (5.12) garantem que em cada período t , não são utilizadas mais máquinas tipo m do que as existentes.

As restrições (5.13) garantem que em cada período t , não são colocadas na linha l mais máquinas do que as permitidas.

As restrições (5.14) garantem que cada OF e , passa antes pela pré-costura e só depois pela costura.

As restrições (5.15) e (5.16) definem as variáveis γ_{elt} , necessárias para as equações (5.17) e (5.18) que garantem que cada OF e é produzida no máximo em dois períodos (ver [Sousa, 1989, págs. 50-55]).

Adicionalmente, impõem-se as condições (5.19) por forma a garantir que um produto tipo j nunca será produzido numa linha para a qual a capacidade ou o índice de preferência para esse tipo de produto sejam nulos. Essas condições garantem ainda que uma OF nunca será produzida fora do intervalo I_e (nem antes da data de início possível, nem depois da data de entrega). Em termos computacionais, as variáveis não são definidas nesse caso.

5.2.3 Dimensão do modelo para problemas reais

Para se fazer uma estimativa da dimensão deste modelo nos problemas reais, considerem-se, para os dados, os seguintes valores “médios”: número de OF $E = 200$, número de tipos de produtos $J = 20$, número de linhas $L = 10$, número de tipos de máquinas $M = 15$ e número de períodos $T = 25$.

Para esta instância “representativa” o número de variáveis contínuas será $ELT = 50000$, o número de variáveis binárias será $ELT + EL = 52000$, o número de variáveis inteiras será $MLT = 3750$ e o número de restrições será igual a $3E + 3ELT + 2LT + JT + EL + MT + ET + MLT = 162725 \approx 160000$.

Mesmo se não se considerasse a existência de vários objectivos em simultâneo, as 52000 variáveis binárias a considerar numa instância real “normal”, tornariam a resolução do problema por Programação Matemática impossível em tempo útil. Para além desse facto, estamos perante um problema multiobjectivo, para o qual não seria nunca possível encontrar uma “solução óptima”. Estes factos levaram ao desenvolvimento da heurística que se apresenta a seguir e que se baseia na divisão do problema nas suas duas componentes “naturais”.

5.3 Heurística baseada em PL (*H-PROG1*)

Dado o elevado número de variáveis e de restrições do modelo apresentado na secção anterior, e dado o seu carácter multiobjectivo, foi desenvolvida uma heurística com duas fases, e que se baseia na divisão do problema nas suas duas componentes principais. A essa heurística deu-se o nome de *H-PROG1*.

Na primeira fase de *H-PROG1*, são afectadas as OF a períodos e a linhas de produção, tentando-se produzir o mais perto possível da data de entrega. Na segunda parte, é utilizada a informação da primeira parte para calcular o número de máquinas necessárias em cada linha, tendo como objectivo a minimização das mudanças de “layout”. Pretende-se ainda, com uma realimentação da informação, construir sempre uma solução admissível para o problema globalmente considerado.

Nos modelos relacionados com esta heurística, são fundamentais as seguintes entidades:

x_{elt} quantidade da OF e a produzir na linha l no período t ;

k_{mlt} número de máquinas do tipo m a colocar na linha l no período t ;

c_{mlt} capacidade das máquinas do tipo m a colocar na linha l no período t .

(Para uma descrição mais completa da notação utilizada, ver Apêndice A)

5.3.1 *H-PROG1*, parte 1

Nesta primeira parte da heurística, o modelo utilizado é, em linhas gerais, o mesmo que foi apresentado na secção 5.2, passando as variáveis inteiras correspondentes ao número de máquinas (k_{mlt}), a serem consideradas como contínuas (c_{mlt}). O modelo apresentado pode ser resolvido por Programação Linear.

Função objectivo

Neste contexto, passou-se a considerar apenas a função objectivo (5.23) com a qual se pretende que cada OF seja produzida antes da data de entrega e o mais perto possível desta.

$$\min \sum_{e,l,t} x_{elt}(d_e - t) \quad (5.23)$$

Restrições

As restrições foram mantidas, apenas com as alterações, já referidas anteriormente, resultantes das variáveis referentes ao número de máquinas. Apresenta-se a seguir apenas as restrições que foram alteradas.

$$\forall_{m,t} \quad \sum_{e:\tau_e=j} \sum_j p_{1jm} x_{elt} - c_{m1t} b_t \leq 0 \quad (5.24)$$

$$\forall_{m,l \neq 1,t} \quad \sum_{e:\tau_e=j} \sum_j p_{2jm} x_{elt} - c_{mlt} b_t \leq 0 \quad (5.25)$$

$$\forall_{m,t} \quad \sum_l c_{mlt} \leq K_m b_t \quad (5.26)$$

$$\forall_{l,t} \quad \sum_m c_{mlt} \leq D_l b_t \quad (5.27)$$

$$\forall_{m,l,t} \quad c_{mlt} \geq 0 \quad (5.28)$$

As restrições (5.24) e (5.25) garantem que em cada linha l e em cada período t , não são ultrapassadas as capacidades das máquinas do tipo m .

As restrições (5.26) garantem que em cada período t , não são utilizadas mais máquinas do que as existentes (de cada tipo).

As restrições (5.27) garantem que em cada período t , não é colocada na linha l mais “capacidade” do que a linha permite (devido a restrições físicas).

As restrições (5.28) garantem que as variáveis c_{mlt} são positivas.

Dimensão do modelo para problemas reais

Para se fazer uma estimativa da dimensão deste modelo nos problemas reais, considerem-se novamente os seguintes valores “médios”: $E = 200$, $J = 20$, $L = 10$, $M = 15$ e $T = 25$.

Neste modelo, o número de variáveis contínuas será $ELT + MLT = 53750$, o número de variáveis binárias será $ELT + EL = 52000$ e o número de restrições será igual a $3E + 3ELT + 2LT + JT + EL + MT + ET + MLT = 162725 \approx 160000$.

5.3.2 Procedimento de arredondamento

Na primeira parte da heurística, são afectadas OF aos períodos e às linhas de produção, obtendo-se valores contínuos para as variáveis c_{mlt} . Nesta fase são transformados (arredondando para cima) os valores de c_{mlt} em k_{mlt} , $k_{mlt} = \lceil c_{mlt} \rceil$, definindo-se, desta forma, o número de máquinas a utilizar (de cada tipo, em cada linha e em cada período). No entanto, este procedimento pode criar soluções não admissíveis, dado que as seguintes equações podem não ser satisfeitas:

$$\forall_{m,t} \quad \sum_l k_{mlt} \leq K_m \quad (5.29)$$

$$\forall_{l,t} \quad \sum_m k_{mlt} \leq D_l \quad (5.30)$$

Em todas as equações que não sejam satisfeitas, é então reduzido de uma unidade o termo independente, resolvendo-se de novo o respectivo PL (primeira parte da heurística). Repete-se este procedimento até que, após o arredondamento, todas as equações sejam admissíveis, passando-se então à segunda parte da heurística (ver Figura 5.1).

5.3.3 *H-PROG1*, parte 2

O objectivo da segunda parte da heurística, é encontrar um “layout” para todas as linhas e para todos os períodos, que minimize as mudanças de máquinas entre períodos.

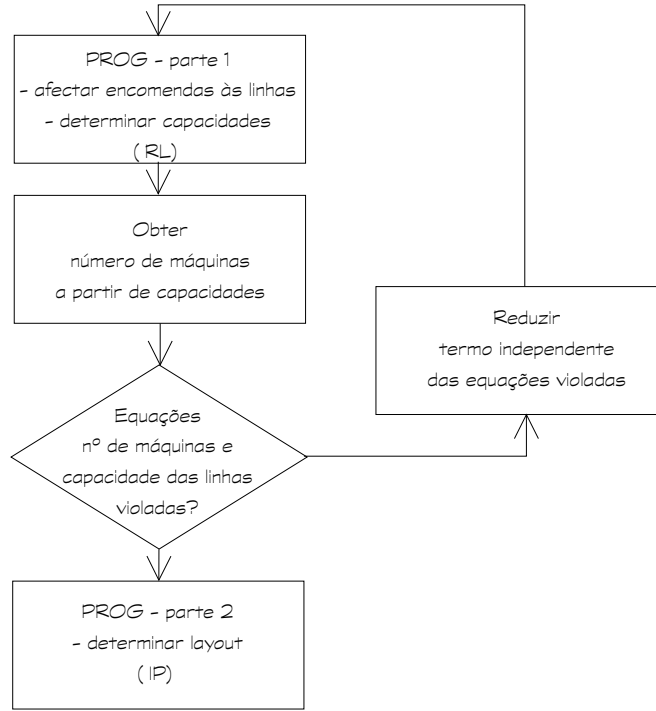


Figura 5.1: Procedimento de arredondamento

Parâmetros adicionais

c_{mlt}^* capacidade de máquinas tipo m , necessária na linha l durante o período t (correspondem ao valor de c_{mlt} resultante da parte 1 da heurística).

Variáveis contínuas adicionais

a_{mlt} variáveis auxiliares.

Função objectivo

A função objectivo a considerar neste caso é a que minimiza as alterações no número de máquinas, de um período para o seguinte (ver secção 5.2.1).

$$\min \sum_{m,l,t} |k_{mlt} - k_{ml(t-1)}| \quad (5.31)$$

Restrições

As restrições a considerar neste caso, são as referentes ao número de máquinas (ver secção 5.2.2).

$$\forall_{m,l,t} \quad k_{mlt} b_t \geq c_{mlt}^* \quad (5.32)$$

$$\forall_{m,t} \quad \sum_l k_{mlt} \leq K_m \quad (5.33)$$

$$\forall_{l,t} \quad \sum_m k_{mlt} \leq D_l \quad (5.34)$$

$$\forall_{m,l,t} \quad k_{mlt} \geq 0 \text{ inteiro} \quad (5.35)$$

Linearização da função objectivo

Para linearizar a função objectivo, é necessário substituí-la por:

$$\min \quad \sum_{m,l,t} a_{mlt} \quad (5.36)$$

e considerar dois novos grupos de restrições:

$$\forall_{m,l,t>2} \quad k_{mlt} - k_{ml(t-1)} - a_{mlt} \leq 0 \quad (5.37)$$

$$\forall_{m,l,t>2} \quad -k_{mlt} - k_{ml(t-1)} - a_{mlt} \leq 0 \quad (5.38)$$

Dimensão do modelo para problemas reais

Para se fazer uma estimativa da dimensão deste modelo nos problemas reais, considerem-se novamente os seguintes valores “médios”: $L = 10$, $M = 15$ e $T = 25$.

Neste modelo, o número de variáveis contínuas será $MLT = 3750$, o número de variáveis inteiras será $MLT = 3750$ e o número de restrições será igual a $MLT + MT + LT = 4375$.

5.3.4 Procedimento para reduzir as mudanças de “layout”

Foi desenvolvido um procedimento para resolver mais eficientemente a parte 2 da heurística *H-PROG1*. Como seria de esperar, esta heurística, dado o seu

carácter não optimizante, não obtém um número de mudanças de “layout” tão baixo como a resolução do modelo descrito anteriormente. No entanto, grande parte dos “buracos” dos “layouts” obtidos na parte 1, desaparecem (ver Figura 5.2).

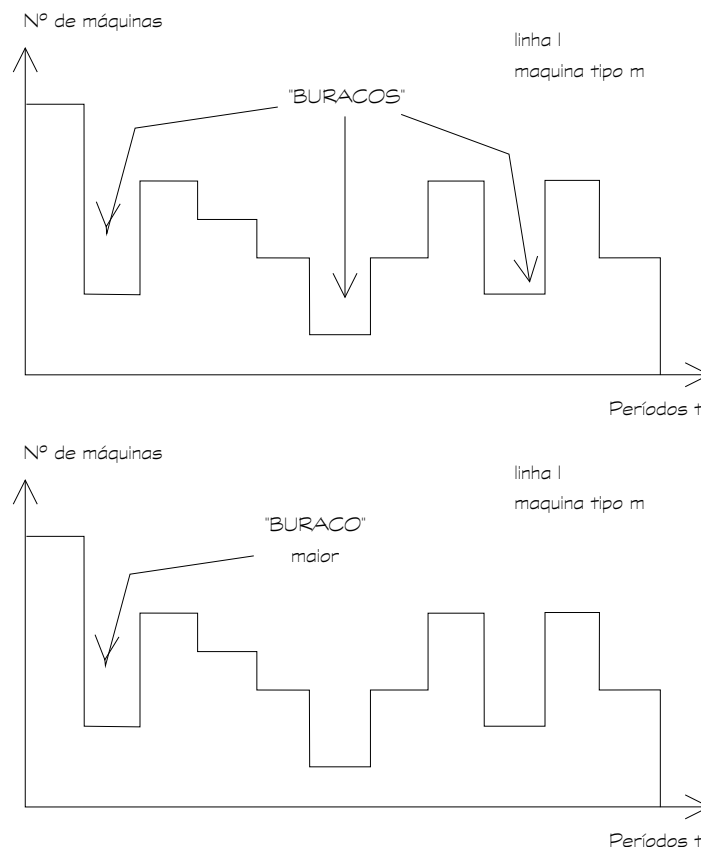


Figura 5.2: À procura do “buraco” mais fundo

Com o conceito de “buraco” pretende-se exprimir a existência de variações no “layout”. Assim para máquinas do tipo i , e numa dada linha l , considera-se que existe um “buraco” no período t , se são necessárias mais máquinas no período anterior e no período posterior. A esses “buracos” poderíamos chamar “buracos de primeira ordem”.

A heurística começa por detectar todos os “buracos” existentes, em todos os períodos, em todas as máquinas e todas as linhas e escolhe o mais “fundo”. A seguir tenta “encher” esse buraco, afectando a esse período e a essa linha

uma máquina adicional. Tal pode não ser possível, se não houver mais máquinas disponíveis nesse período, ou se o “layout” daí resultante exceder a capacidade da linha (D_j). Nesse caso “marca-se” esse “buraco” como não preenchível. A heurística termina quando já não houver mais buracos preenchíveis.

5.3.5 *H-PROG1* - conclusões

Utilizando a heurística *H-PROG1*, continua na parte 1, a ser necessário resolver problemas de PL de grande dimensão que tornam a resolução do problema impossível em tempo útil. Para além disso, a estrutura das soluções obtidas por resolução dos subproblema por programação matemática pode mudar radicalmente, mesmo com uma pequena alteração dos dados. Essa “instabilidade” das soluções pode levar o Agente de Decisão a não aderir ao método de resolução, por não o poder “seguir”. Estes factos levaram ao desenvolvimento de uma heurística orientada especificamente para o Caso de Estudo e que se descreve na secção seguinte.

5.4 Heurística para o Caso de Estudo (*H-PROG*)

O procedimento desenvolvido para resolver o problema da Programação no caso específico em estudo é uma heurística construtiva designada por *H-PROG*. Nesta heurística, o planeamento é realizado OF a OF, tendo-se procurado que o método de planeamento se aproximasse o mais possível dos procedimentos tradicionais e que as soluções obtidas pela heurística, tivessem as mesmas características que as soluções obtidas anteriormente de forma manual.

5.4.1 Critérios para avaliação das soluções

Para avaliação das soluções, foram considerados, de uma forma explícita ou implícita, os seguintes critérios:

- *JIT*, com o qual se pretende “medir” a “distância” entre a data de produção e a data de entrega proposta, para o conjunto das OF. A produção depois da data de entrega não é admissível neste caso.

$$JIT = \sum_{e,l,t} x_{elt}(d_e - t)$$

- *PREF*, com o qual se pretende quantificar a afectação das OF às linhas de maior “preferência”. A preferência é medida através de um índice introduzido pelo utilizador. Um índice de preferência 0 implica sempre uma capacidade de produção nula. Os índices de preferência são valores inteiros não negativos, e a preferência diminui com o valor do índice.
- **Utilização da capacidade das linhas**, com o qual se pretende quantificar a percentagem da capacidade disponível por período, nas linhas de produção da empresa (internas).
- **Índice de subcontratação**, com o qual se pretende quantificar as quantidades subcontratadas.
- **Número de máquinas disponíveis**, com o qual se pretende quantificar o número de máquinas disponíveis de cada tipo, por período.
- **Alterações de “layout”**, com o qual se pretende quantificar a quantidade de máquinas deslocadas entre os períodos de planeamento.

5.4.2 Heurística *H-PROG*: descrição geral

Para resolver o problema *PROG*, foi desenvolvida uma heurística inspirada nos procedimentos manuais, e que tem em conta de uma forma implícita os diferentes critérios de avaliação de soluções apresentados. São no entanto especialmente tidos em conta o cumprimento das datas de entrega e a necessidade de minimizar a subcontratação, dado que, de acordo com o levantamento do problema do Caso de Estudo, são esses os critérios mais valorizados pelo Agente de Decisão.

A afectação das OF às linhas de costura, é feita tendo em atenção as datas de entrada em montagem, o “lead-time” da costura e as capacidades das linhas de costura (em número de unidades por dia). Para distribuir as OF pelas linhas, tentou-se manter a filosofia actualmente existente na

fábrica, que consiste na existência, por um lado, de linhas ditas “dedicadas” nas quais são costurados 1 ou 2 tipos diferentes de produtos, e por outro lado, de linhas ditas de “diversos” nas quais é costurado simultaneamente um grande número de tipos de produtos (normalmente existem ainda linhas “intermédias”).

Assim, optou-se por fazer a distribuição das OF, do ponto de vista das linhas de produção. Começa-se pela linha dedicada, e tenta-se “encher” a produção, começando pelos tipos de produtos mais “preferidos” por essa linha. Quando essa linha estiver cheia, ou não houver mais tipos de produtos que possam lá ser produzidos, passa-se para a linha seguinte (menos “dedicada”). Toda a produção de costura que não possa ser produzida nas linhas internas, é afectada a uma linha que corresponde à produção exterior e que se considera ter capacidade infinita.

Depois de feita a afectação das OF às linhas e períodos, determina-se o “layout” das linhas de costura, em cada período.

5.4.3 Dados de entrada

Os dados necessários para a heurística *H-PROG*, são:

1. OF:

- data de entrada na montagem;
- data de início possível de produção (“release date”);
- tipo de produto;
- quantidade.

2. Linhas de costura:

- número de identificação;
- capacidade (em número de postos).

onde a *linha* 0 é a pré-costura, a *linha* 1 é a linha mais dedicada, ... e a *linha* $L - 1$ é a linha exterior.

3. “Lead time” da costura.

4. Tipos de produtos.
5. Gama operatória de cada tipo de produto.
6. Matriz das capacidades, com os valores de C_{jlt} (capacidade de produção do produto tipo j na linha l no período t (em unidades)) e de U_{jt} (máxima produção possível do produto tipo j em todas as linhas no período t (em unidades)).

No exemplo da Tabela 5.1, exprime-se por exemplo a restrição de que se só for produzido o produto tipo A na linha 1, então poderão ser produzidas 1000 unidades durante o período t . No caso de se pretender produzir os produtos tipo A e B na linha 1, então pode-se produzir p_1 unidades do tipo A e p_2 unidades do tipo B, de tal modo que $\frac{p_1}{1000} + \frac{p_2}{500} \leq 1$

	C_{jlt}				U_{jt}
	Linha 1	Linha 2	Linha 3	Linha 4	
$j = A$	1000	1500	1200	1300	1000
$j = B$	500	300	700	800	1000
$j = C$	250	2000	500	200	1500

Tabela 5.1: Matriz com capacidades de produção por tipo de produto j , para um período t

7. Matriz dos índices de preferência de cada linha de produção l , para cada tipo de produto j , para um dado período t (valores de P_{jlt}).

Um índice de preferência 0 implica sempre uma capacidade de produção nula. Os índices de preferência são valores inteiros positivos e a preferência diminui com o valor do índice. No exemplo da Tabela 5.2, se se pretender por exemplo gerar uma lista de índices de preferência de tipos de produtos para a linha 3, esta será (por ordem decrescente de preferência) tipo A, tipo C e tipo B. Se se pretender gerar uma lista de índices de preferência de linhas para o tipo de produto B, esta será (por ordem decrescente de preferência) linha 2, e a seguir linha 3 e linha 4 (o tipo de produto B não pode ser produzido na linha 1).

8. Tempo de funcionamento do sistema produtivo (em minutos por período de planeamento - b_t).

	P_{jlt}			
	Linha 1	Linha 2	Linha 3	Linha 4
$j = A$	0	1	1	3
$j = B$	0	1	3	9
$j = C$	2	2	2	1

Tabela 5.2: Matriz de índices de preferência de cada tipo de produto j para cada linha l , para um dado período t

No exemplo da Tabela 5.3, no período 1 há 2880 minutos de produção. Estes valores são usados para calcular os valores de C_{jlt} e de U_{jt} .

t	1	2	3	4	...	n
b_t	2880	3010	2900	2950	...	3000

Tabela 5.3: Calendário de Produção

5.4.4 Resultados

1. Para cada OF:

- períodos de produção;
- quantidades a produzir em cada período;
- linha em que vai ser produzida.

2. Para cada linha e em cada período:

- tipo e número de máquinas;
- cargas afectadas;
- cargas disponíveis.

5.4.5 Descrição genérica do algoritmo

O algoritmo desenvolvido para o PROG é constituído pelas seguintes funções, chamadas de uma forma sequencial (para uma descrição completa destas funções, ver Apêndice B):

Inicializa() inicializa as variáveis globais;

ColocaPROG(e) carrega as linhas de produção com as OF, começando pela linha mais dedicada e terminando na linha exterior, tendo em conta os índices de preferência e as capacidades disponíveis;

FazLayout() determina a carga de cada tipo de máquina (em minutos), para cada linha e em cada período;

RoundLayout() determina o número de máquinas de cada tipo necessário em cada linha e em cada período;

CalculaFolgas() determina o número de máquinas de cada tipo e o número de postos em cada linha que ficam disponíveis em cada período.

5.4.6 Exemplo ilustrativo

No Apêndice D descreve-se pormenorizadamente a resolução, com a heurística *H-PROG*, de uma instância do problema PROG. Aqui, apresentam-se apenas os dados dessa instância e ainda os resultados obtidos (para uma descrição completa da notação utilizada, recomenda-se a consulta do Apêndice A).

Dados de entrada

O número de minutos de produção disponíveis em cada período é constante e igual a 2000. Os restantes dados de entrada são apresentados nas Tabelas 5.4 a 5.8.

e	τ_e	r_e	d_e	q_e
1	1	1	3	4000
2	3	2	3	5000
3	2	2	3	3000
4	3	1	3	2500
5	2	1	1	5000

Tabela 5.4: Lista de OF: tipo de produto (τ_e), “release date” (r_e), data de entrega (d_e) e quantidade (q_e)

Resultados

Nas Figuras 5.3, 5.4 e 5.5 é apresentada a solução do exemplo, obtida pela heurística *H-PROG*. Na Figura 5.3 apresenta-se um diagrama, indicando

	Linha 1	Linha 2	Linha 3
	P_{jlt}		
$j = 1$	0	4	1
$j = 2$	1	0	3
$j = 3$	0	2	1
	C_{jlt}		U_{jt}
$j = 1$	0	5000	2500
$j = 2$	5000	0	2500
$j = 3$	0	2500	2500

Tabela 5.5: Índices de preferência, P_{jlt} e capacidades de produção, C_{jlt} , das linhas l por tipo de produto j , para todos os períodos t e máxima produção possível do produto tipo j em todas as linhas no período t , U_{jt}

PréCostura	$j = 1$	$j = 2$	$j = 3$
$m = 1$	2	3	1
$m = 2$	0	0	0
$m = 3$	10	2	1
Costura			
$m = 1$	3	4	1
$m = 2$	5	4	3
$m = 3$	0	0	0

Tabela 5.6: Gamas operatórias: Carga do tipo de produto j na máquina m (em minutos por unidade de produto)

para que períodos e linhas foram planeadas as OF e qual a carga de cada linha em cada período. Na Figura 5.4 é apresentado, para cada tipo de máquina, o número de máquinas existentes e o número de máquinas necessárias em cada período. O utilizador é alertado sempre que o número de máquinas necessárias num determinado período, exceder o número de máquinas existentes (ver seta na figura). Na Figura 5.5 é apresentado, para cada linha, o número máximo de postos permitido, e o número de postos

Tipo de máquina m	1	2	3
K_m	20	16	22

Tabela 5.7: Quantidade de máquinas existentes de cada tipo (K_m)

Linha	Tipo	Nº máximo de postos D_l
$l = 0$	Pré-costura	28
$l = 1$	Costura	15
$l = 2$	Costura	15
$l = 3$	Costura	15
$l = 4$	Exterior	∞

Tabela 5.8: Número máximo de postos em cada linha

necessários em cada período. Neste caso o utilizador também é alertado sempre que o número de postos necessários numa determinada linha e num determinado período, exceda o máximo número de postos admissível nessa linha (ver seta na figura).

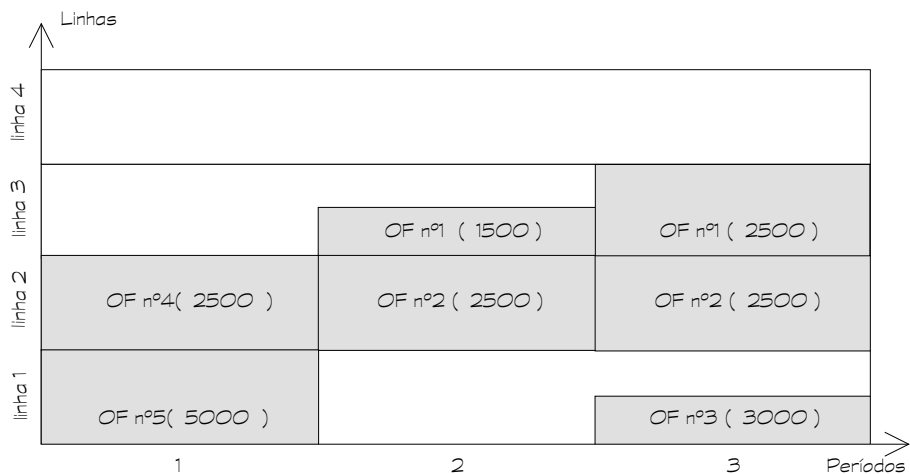


Figura 5.3: Solução obtida: Períodos e linhas em que foram colocadas as OF e ocupação das linhas

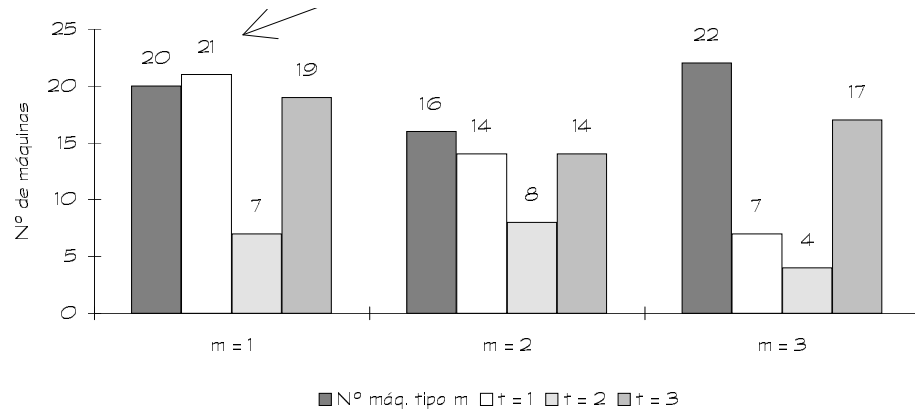


Figura 5.4: Solução obtida: Número de máquinas existentes de cada tipo (K_m) e número de máquinas de cada tipo necessárias em cada período

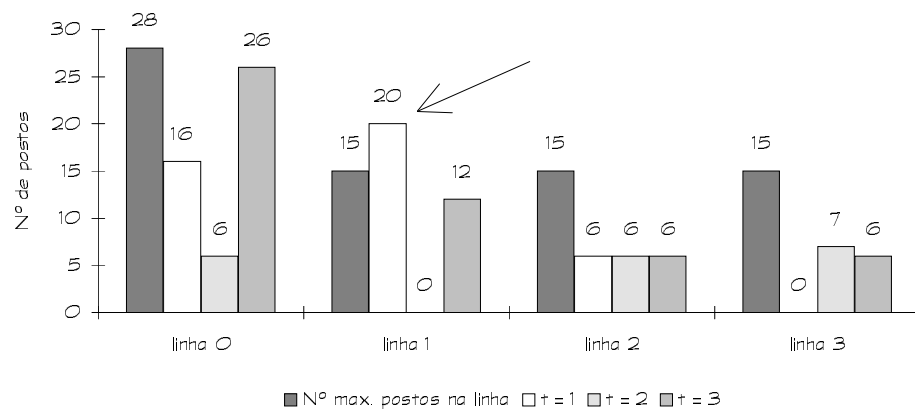


Figura 5.5: Solução obtida: Número máximo de postos permitido em cada linha (D_l) e número de postos necessários em cada período e em cada linha

5.4.7 Aplicação da heurística *H-PROG* ao Caso de Estudo

Dimensão dos problemas

No Caso de Estudo, os problemas reais tratados com esta heurística têm tipicamente as seguintes dimensões:

- 8 períodos de planeamento;
- 5 linhas;
- 3 fábricas;
- 14 tipos de produtos;
- 500 OF.

Comentários à utilização da heurística e suas limitações

Os resultados computacionais obtidos pela aplicação da heurística *H-PROG* aos problemas do Caso de Estudo, embora de carácter preliminar, mereceram globalmente (tal como os resultados da heurística *H-PDP*) a aprovação dos planeadores, dado que as soluções são obtidas em poucos segundos e são estruturalmente semelhantes às obtidas pelos métodos tradicionais.

Na heurística desenvolvida para a Programação (*H-PROG*), tal como na *H-PDP* (ver secção 3.3.7), é no entanto necessário introduzir mecanismos que permitam ao Agente de Decisão controlar a solução, permitindo-lhe, no limite, construir uma solução de raiz ou alterar parcialmente uma solução produzida automaticamente pela heurística.

A opção tomada, de se começar a planear pela linha “dedicada”, pode ser posta em causa pela introdução de “células” de fabrico. Essa alteração na organização da produção pode mesmo pôr em causa outras opções tomadas na concepção da heurística, como por exemplo o cálculo de “layouts”.

Relativamente aos “layouts”, foi reconhecido posteriormente que seria útil proceder ao seu cálculo diariamente, de acordo com a produção de cada dia, e não semanalmente (período considerado no *PROG*). Por isso seria talvez interessante incluir esse cálculo num nível de planeamento mais baixo, com períodos mais curtos (dia).

Deveria ser também possível fazer uma “realimentação” de informação, que permitisse resolver questões relacionadas com a seguinte situação: No *PDP* é gerado um plano para as encomendas, que indica a sua data de entrada em montagem. No *PROG*, as OF são planeadas na montagem e na costura de acordo com a data fornecida pelo *PDP*. Se não houver capacidade de costura, as OF que não puderem ser planeadas na costura interna, podem

sempre ser planeadas na costura externa. No entanto, se a data de entrada em produção dessas OF sofrer um atraso, devido por exemplo a falta de materiais, então o plano do PDP deixa de ser válido. No entanto, na versão actual do sistema, não é possível passar essa informação automaticamente para o PDP, passando a haver uma inconsistência entre os vários níveis de planeamento.

Note-se finalmente que, relativamente ao cálculo de “layouts”, o Agente de Decisão (AD) é alertado para excessos de postos em determinadas linhas ou para necessidades de máquinas de um determinado tipo, que excedem o número de máquinas existentes. Face a essas informações, o AD pode sempre alterar alguns parâmetros e executar novamente o algoritmo, até obter uma solução admissível. No entanto, deveria ser possível fazer uma realimentação interna, que permitisse alterar automaticamente dados de planeamento, tais como capacidades das linhas para cada tipo de produto, de modo a obter automaticamente uma solução que satisfizesse as restrições de número de postos e de número de máquinas de cada tipo.

Capítulo 6

Um Modelo de Programação Inteira e Desigualdades Válidas

6.1 Motivação

No Capítulo 3, foi apresentado um problema de planeamento agregado relativamente geral, designado por Plano Director de Produção (PDP), bem como um modelo de Programação Matemática e uma heurística para a sua resolução. Dada a complexidade do problema PDP original, foi considerado interessante, após uma análise detalhada do modelo desenvolvido, explorar simplificações desse modelo, por forma a desenvolver métodos de resolução eficientes.

Começa-se assim neste capítulo por descrever vários problemas que se podem obter por simplificação do problema PDP. Desses problemas, foi escolhido um para estudo por se considerar relativamente simples e ainda suficientemente geral, mantendo no entanto as características básicas de um problema real de planeamento. Esse problema é aqui designado por PAP (de Planeamento Agregado da Produção).

O objectivo deste capítulo é apresentar uma abordagem considerada promissora para este tipo de problemas, que se baseia na exploração de propriedades estruturais do problema, com vista à obtenção de desigualdades

válidas para o modelo proposto, que permitam calcular minorantes (“lower bounds”) sobre o valor da solução óptima. Estes minorantes, integrados em métodos de resolução do tipo “branch-and-bound”, permitem a resolução eficiente do problema de optimização.

6.2 Identificação do problema

6.2.1 Modelo 0 (PDP)

Como referido, o ponto de partida deste trabalho é o problema PDP do Caso de Estudo (ver Capítulo 3).

Considere-se que existe um conjunto de encomendas e , cada uma associada a um tipo de produto $\tau_e = j$, com uma quantidade q_e , com uma data proposta para entrega e uma data de início de produção, os período d_e e r_e de um horizonte temporal discretizado. Para produzir essas encomendas, existem linhas de produção l em diversas fábricas f . Cada linha de produção tem uma capacidade e um índice de preferência por tipo de produto e por período, C_{jlt} (capacidade de produção do produto tipo j na linha l no período t (em unidades)) e P_{jlt} (índices de preferência de cada linha de produção l , para cada tipo de produto j , para um dado período t) respectivamente. As encomendas têm que ser totalmente produzidas, e cada encomenda só pode ser produzida numa fábrica. Em cada período, não se pode ultrapassar um dado limite máximo de produção por tipo de produto, U_{jt} .

Pretende-se conhecer a quantidade (em unidades) de cada encomenda e a produzir na linha l e período t , x_{elt} , e os objectivos são produzir o mais perto possível da data de entrega proposta (de preferência antes, isto é, penalizando mais os atrasos), produzir nas linhas preferidas e produzir cada encomenda no menor número possível de linhas e períodos. Assim, são penalizadas na função objectivo (6.1), tanto a produção antes da data de entrega como a produção depois da data de entrega. Esta última é em geral muito menos desejável, sendo o custo associado controlado pelo parâmetro a . Neste contexto foi usado o valor $a = 2$. Por outro lado, com o coeficiente 2 na segunda parcela da soma ($2x_{elt}$), pretende-se que o custo de produzir

no período seguinte à data de entrega proposta seja superior ao custo de produzir no período anterior a essa mesma data.

(Para uma descrição completa da notação, ver o Apêndice A)

As equações (6.1), (6.2) e (6.3), representam as três funções objectivo.

$$JIT \quad \min \sum_{e,l,t \leq d_e} x_{elt}(d_e - t) + \sum_{e,l,t=d_e+1} 2x_{elt} + \sum_{e,l,t \geq d_e+2} x_{elt}(t - d_e)^a \quad (6.1)$$

$$PREF \quad \min \sum_{e,l,t} x_{elt} * P_{jlt:\tau_e=j} \quad (6.2)$$

$$ESP \quad \min \sum_{e,l,t} \gamma_{elt} \quad (6.3)$$

O modelo completo, bem como a justificação para as opções realizadas, foram apresentados na secção 3.2 deste trabalho.

6.2.2 Modelo 1 (encomendas unitárias e $r_e = 0$)

Nesta primeira simplificação, introduziram-se as seguintes alterações relativamente ao modelo 0 (PDP):

- todas as encomendas “gastam” a mesma capacidade (isto é, são do mesmo tamanho) e têm que ser totalmente produzidas numa só linha/período;
- a menor data possível para início de produção da encomenda e (“release date”) é igual a zero;
- todas as linhas pertencem à mesma fábrica (só há uma fábrica).

Estas alterações implicaram a substituição das variáveis de decisão anteriores pelas seguintes variáveis binárias:

$$x_{elt} = \begin{cases} 1 & \text{se a encomenda } e \text{ é produzida na linha } l \text{ no período } t \\ 0 & \text{se não} \end{cases}$$

Continua-se a considerar neste modelo as funções objectivo (6.1) e (6.2).

Dado que se pretende produzir cada encomenda num só período e numa só linha, a função objectivo (6.3) deixou de ter razão de ser, dado que o seu valor seria sempre igual ao número de encomendas.

Restrições

$$\forall_e \quad \sum_{l,t} x_{elt} = 1 \quad (6.4)$$

$$\forall_{j,t} \quad \sum_{e:\tau_e=j} \sum_l x_{elt} \leq U_{jt} \quad (6.5)$$

$$\forall_{j,t} \quad \sum_{e:\tau_e=j} \sum_{j:C_{jlt}>0} \frac{x_{elt}}{C_{jlt}} \leq 1 \quad (6.6)$$

$$\forall_{e,l,t} \quad \text{se } P_{jlt:\tau_e=j} > 0 \text{ e } C_{jlt:\tau_e=j} > 0 \quad x_{elt} \in \{0, 1\} \quad (6.7)$$

As restrições (6.4) garantem que cada encomenda, é produzida.

As restrições (6.5) garantem que, em cada período t , a produção do produto tipo j não excede um limite superior previamente definido.

As restrições (6.6) garantem que a capacidade produtiva de cada linha não é excedida.

Adicionalmente, impõem-se as condições (6.7), por forma a garantir que um produto tipo j só será produzido numa linha para a qual a capacidade e índice de preferência para esse tipo de produto sejam > 0 e ainda que as variáveis x_{elt} são binárias.

6.2.3 Modelo 2 (encomendas do mesmo tipo de produto)

Considera-se aqui uma especialização do modelo 1, em que se introduzem as alterações seguintes:

- todas as encomendas são do mesmo tamanho e do mesmo tipo;
- não há índices de preferência P_{jlt} ;
- não há limites máximos para produção do produto tipo j no período t , U_{jt} ;
- C_{lt} , capacidade de produção na linha l no período t (em unidades), substitui C_{jlt} , dado que só há um tipo de produto.

Continua-se a considerar neste modelo a função objectivo (6.1), que obriga a que cada encomenda seja produzida o mais perto possível do período de entrega proposto, d_e , e se possível antes. Dado que todas as encomendas são do mesmo tipo, não há índices de preferência e deixa de se considerar a função objectivo (6.2).

Restrições

$$\forall_e \quad \sum_{l,t} x_{elt} = 1 \quad (6.8)$$

$$\forall_{l,t} \quad \sum_e x_{elt} \leq C_{lt} \quad (6.9)$$

$$\forall_{e,l,t:C_{lt}>0} \quad x_{elt} \in \{0, 1\} \quad (6.10)$$

As restrições (6.8) garantem que cada encomenda, é produzida.

As restrições (6.9) garantem que em qualquer período, a capacidade produtiva de cada linha não é excedida.

Adicionalmente, impõem-se as condições (6.10), por forma a garantir que uma encomenda e só será produzida numa linha e período tais que $C_{lt} > 0$ e ainda que as variáveis x_{elt} são binárias.

Proposição

O modelo 2 é um problema de transportes, em que as encomendas e estão associadas a origens e os pares lt a destinos. Cada encomenda só pode ser colocada num destino lt (isto é, a “origem” tem de “disponibilidade” 1 unidade), mas cada destino pode ter até C_{lt} encomendas (a “procura” associada a cada “destino” é de C_{lt} unidades).

Observação

Nas restrições 6.8 e 6.9 do modelo 2, pode-se observar que as linhas l e os períodos t estão desacoplados. Pode-se então substituir o par lt por um único índice s , sem necessitar de fazer outras alterações no modelo, estando associado a cada valor do índice s uma determinada capacidade C_s . Esta observação levou-nos a considerar a situação, não limitativa, de uma só linha no modelo 3.

6.2.4 Modelo 3 (encomendas de 1, 2 e 3 unidades e uma só linha)

Relativamente ao modelo 2, considere-se agora que as encomendas se podem dividir em pequenas, médias e grandes. Considere-se ainda q_e como sendo

a “carga” associada a e , que pode tomar os valores 1, 2 ou 3, o que corresponde a dividir as encomendas nesses três grandes grupos. Esta opção é justificada na prática, dado que frequentemente os dados disponíveis para o planeamento são grosseiros e é muito difícil ser mais específico quanto às quantidades a produzir. As encomendas pequenas corresponderão a $q_e = 1$, as encomendas médias corresponderão a $q_e = 2$ e as encomendas grandes corresponderão a $q_e = 3$.

Dado que no modelo 2 as linhas e os períodos estão desacoplados, considera-se no modelo 3 uma só linha de produção, o que implica a criação das seguintes variáveis:

$$x_{et} = \begin{cases} 1 & \text{se a encomenda } e \text{ é produzida no período } t \\ 0 & \text{se não} \end{cases}$$

e do parâmetro:

C_t capacidade de produção no período t (em unidades).

A função objectivo é igual à do modelo 0, tendo em conta que o custo de produzir antes ou depois de d_e , deve ser proporcional ao tamanho da encomenda (q_e):

$$JIT \quad \min \quad \sum_{e,t \leq d_e} q_e * x_{et}(d_e - t) + \sum_{e,t = d_e + 1} 2 * q_e * x_{et} + \sum_{e,l,t \geq d_e + 2} q_e x_{et}(t - d_e)^2 \quad (6.11)$$

Restrições

$$\forall_e \quad \sum_t x_{et} = 1 \quad (6.12)$$

$$\forall_t \quad \sum_e q_e * x_{et} \leq C_t \quad (6.13)$$

$$\forall_{e,t: C_t > 0} \quad x_{et} \in \{0, 1\} \quad (6.14)$$

As restrições (6.12) garantem que as encomendas são todas produzidas.

As restrições (6.13) garantem que em qualquer período t , a capacidade produtiva não é excedida.

Adicionalmente, impõem-se as condições (6.14), por forma a garantir que uma encomenda e só será produzida num período com capacidade $C_t > 0$ e ainda que as variáveis x_{et} são binárias.

6.3 Modelo PAP (modelo 3)

De todos os modelos obtidos por simplificação do modelo do PDP, foi escolhido o modelo 3 por se considerar relativamente simples e geral, mantendo no entanto as características básicas de um problema de planeamento agregado da produção (inspirado no Caso de Estudo).

Este problema aqui designado por “Planeamento Agregado da Produção” (PAP) corresponde à seguinte situação:

Dado um horizonte de planeamento discretizado $[t=1, \dots, T]$, é conhecida a capacidade disponível de produção para o período t , C_t , que não pode ser ultrapassada. É também conhecido o conjunto de encomendas a produzir, classificadas como encomendas pequenas (que correspondem a uma unidade de carga), médias (que correspondem a duas unidades de carga) e grandes (que correspondem a três unidades de carga). Cada encomenda tem uma data de entrega pretendida. O objectivo é produzir cada encomenda o mais perto possível dessa data de entrega e se possível antes. Por razões operacionais, cada encomenda tem que ser produzida num único período.

6.3.1 Propriedade 1 - Ordenação das encomendas do mesmo tamanho

Se o problema PAP tiver soluções admissíveis para uma determinada instância, então existe uma solução óptima na qual as encomendas de cada tamanho (1, 2 ou 3) estão colocadas por ordem de data de entrega (EDD - “earliest due date”). A demonstração desta propriedade pode encontrar-se no Apêndice G.

Exemplo

Considere-se um pequeno exemplo ilustrativo, com 6 períodos e 8 encomendas. Na Tabela 6.1, está representada a capacidade em cada período e na

Tabela 6.2, indica-se o tamanho q_e e a data de entrega d_e de cada encomenda.

t	1	2	3	4	5	6
C_t	4	5	2	5	2	1

Tabela 6.1: Capacidade disponível, C_t em cada período t

e	1	2	3	4	5	6	7	8
q_e	1	1	2	2	2	3	3	3
d_e	2	1	3	1	2	5	1	3

Tabela 6.2: Encomendas e , com quantidades q_e e datas de entrega d_e

Considerando a função objectivo representada em (6.11), a solução óptima para este problema está representada na Figura 6.1, onde se pode verificar que as encomendas de cada tamanho estão colocadas por ordem de data de entrega (2/1; 4/5/3; 7/8/6).

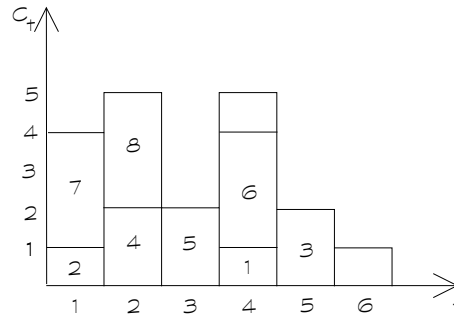


Figura 6.1: Solução óptima do exemplo

6.3.2 Propriedade 2 - Valor das variáveis na relaxação linear

A observação repetida dos resultados computacionais obtidos com o modelo apresentado, conduziu à formulação da seguinte hipótese.

Na solução da relaxação linear, as variáveis correspondentes a encomendas de 1 unidade, tomam sempre os valores 0 ou 1, as variáveis correspondentes a encomendas de 2 unidades, tomam sempre os valores 0, $\frac{1}{2}$ ou 1, e as variáveis correspondentes a encomendas de 3 unidades, tomam sempre os valores 0, $\frac{1}{3}$, $\frac{2}{3}$ ou 1.

Embora ainda não tenha sido possível provar formalmente esta hipótese, na tentativa de justificar a sua consistência foi escrito um outro modelo, para um problema que corresponde a uma relaxação do problema original e que sugere a forma como essa prova poderá ser feita.

Problema auxiliar

As variáveis de decisão x_{et} (binárias), passam a y_{et} (reais), onde y_{et} é a quantidade da encomenda e a produzir no período t (em unidades).

Para este problema, as restrições são:

$$\forall_e \quad \sum_t y_{et} = q_e \quad (6.15)$$

$$\forall_t \quad \sum_e y_{et} \leq C_t \quad (6.16)$$

$$\forall_{et} \quad y_{et} \geq 0 \quad (6.17)$$

O modelo apresentado é um problema de transportes, o que implica que, sendo $q_e \in \{1, 2, 3\}$, na solução, as variáveis só tomam valores inteiros (0 e 1 para as encomendas de 1; 0, 1 ou 2 para as encomendas de 2; e 0, 1, 2 ou 3 para as encomendas de 3).

Considerando novamente a relaxação linear do modelo PAP, e conseguindo provar formalmente que os dois modelos são “equivalentes” (no sentido de representarem o mesmo problema), esses valores correspondem a valores de x_{et} de 0 ou 1 para as encomendas de 1, de 0, $\frac{1}{2}$ ou 1 para as encomendas de 2 e de 0, $\frac{1}{3}$, $\frac{2}{3}$ ou 1 para as encomendas de 3, o que demonstraria a hipótese apresentada.

6.4 Pré-processamento

Pode-se realizar um pré-processamento do modelo, fixando variáveis a 0, a partir da consideração de algumas observações simples.

- Não se pode colocar uma encomenda num período que não tenha capacidade para produzir a totalidade dessa encomenda:

$$\forall_{e,t} \quad C_t < q_e \Rightarrow x_{et} = 0 \quad (6.18)$$

- Se as encomendas do mesmo tamanho devem ser colocadas por ordem de data de entrega numa solução óptima (ver secção 6.3.1), então uma encomenda com data de entrega d_e só pode ser colocada num dado período, desde que todas as encomendas do mesmo tamanho com datas de entrega anteriores possam ser colocadas antes. Considere-se que:

$\mathcal{I}_n$ é o conjunto de todas as encomendas e tais que $q_e = n$;

$|\mathcal{I}_n^{d_e}|$ é o número de encomendas $\in \mathcal{I}_n$, com data de entrega inferior a d_e .

Então:

$$\forall_t \forall_{e \in \mathcal{I}_n} \text{ se } \sum_{t'=1}^t C_{t'} < n * (|\mathcal{I}_n^{d_e}| + 1) \Rightarrow x_{et} = 0 \quad (6.19)$$

Voltando ao exemplo anterior pode-se afirmar a partir de (6.18), que:

$$x_{63} = x_{73} = x_{83} = 0$$

ou que:

$$x_{36} = x_{46} = x_{56} = 0$$

A partir de (6.19), podemos afirmar por exemplo sobre x_{6t} (encomenda 6), que há duas encomendas do mesmo tamanho com data de entrega inferior (a encomenda 7 e a encomenda 8), o que implica que $|\mathcal{I}_3^{d_6}| = 2$. Assim, $x_{61} = 0$ dado que para $t = 1$:

$$C_1 = 4 < 3 * (2 + 1) = 9$$

Não se pode afirmar nada relativamente a x_{6t} com $t > 1$, dado que:

$$C_1 + C_2 = 9$$

6.5 Desigualdades válidas

Com vista a construir procedimentos eficientes para resolução do problema em estudo, procurou-se identificar famílias de desigualdades válidas para a formulação apresentada, que permitissem calcular minorantes melhores do que os obtidos pela relaxação linear. A motivação desta tentativa tem por base os resultados gerais da Teoria Poliédrica, que constitui actualmente uma das bases dos desenvolvimentos mais promissores para a resolução de problemas de Optimização Combinatória. Recomenda-se a leitura de [Nemhauser and Wolsey, 1988] para uma abordagem aprofundada destas técnicas. Em [Sousa and Wolsey, 1992] por exemplo, podem-se encontrar desenvolvimentos dessas técnicas e a sua aplicação a problemas de sequenciamento de tarefas, nalguns aspectos semelhantes ao problema aqui tratado.

A análise detalhada da estrutura das soluções obtidas resolvendo a relaxação linear de várias instâncias teste, motivou o estabelecimento das desigualdades válidas (6.20) a (6.25) que a seguir se apresentam. Essa motivação assenta na observação de que, para cada período e para cada valor da capacidade, existe um número máximo de encomendas de cada tamanho que podem ser colocadas nesse período. Suponha-se, por exemplo, que se pretende colocar encomendas de 3 unidades num período com capacidade 7. Como é evidente, não se pode colocar mais do que duas encomendas de 3 unidades nesse período. Esta observação é um caso particular da desigualdade (6.20). Refira-se que estas desigualdades são, no fundo, casos particulares de “cover inequalities” para os problemas de “knapsack” definidos pelas restrições de capacidade (ver [Nemhauser and Wolsey, 1988]).

Considere-se que $\mathcal{I}_n$ é o conjunto de todas as encomendas e tais que $q_e = n$ e que $\lfloor b \rfloor$ corresponde ao maior inteiro contido em b . Então, estendendo o raciocínio atrás expresso, é fácil derivar as seguintes desigualdades, trivialmente válidas para o problema em estudo:

$$\forall_t \quad \sum_{e \in \mathcal{I}_3} x_{et} \leq \left\lfloor \frac{C_t}{3} \right\rfloor \quad (6.20)$$

$$\forall_t \forall_{w \in \mathcal{I}_2} \quad x_{wt} + \sum_{e \in \mathcal{I}_3} x_{et} \leq \left\lfloor \frac{C_t - 2}{3} \right\rfloor + 1 \quad (6.21)$$

$$\forall_t \quad \sum_{e \in \mathcal{I}_2} x_{et} + \sum_{e \in \mathcal{I}_3} x_{et} \leq \left\lfloor \frac{C_t}{2} \right\rfloor \quad (6.22)$$

$$\forall_t \forall_{z \in \mathcal{I}_1} \quad x_{zt} + \sum_{e \in \mathcal{I}_3} x_{et} \leq \left\lfloor \frac{C_t - 1}{3} \right\rfloor + 1 \quad (6.23)$$

$$\forall_t \forall_{z \in \mathcal{I}_1} \forall_{w \in \mathcal{I}_2} \quad x_{zt} + x_{wt} + \sum_{e \in \mathcal{I}_3} x_{et} \leq \left\lfloor \frac{C_t - 3}{3} \right\rfloor + 2 \quad (6.24)$$

$$\forall_t \forall_{z \in \mathcal{I}_1} \quad x_{zt} + \sum_{e \in \mathcal{I}_2} x_{et} + \sum_{e \in \mathcal{I}_3} x_{et} \leq \left\lfloor \frac{C_t - 1}{2} \right\rfloor + 1 \quad (6.25)$$

Exemplo (cont.)

Considere-se novamente o exemplo apresentado anteriormente (secção 6.3.1). A solução da relaxação linear deste problema encontra-se na Tabela 6.3 e pode ser representada graficamente com o diagrama da Figura 6.2.

e	2	4	7	1	5	7	8	3	5	6	8	3
t	1	1	1	2	2	2	2	3	3	4	4	5
x_{et}	1	1	$\frac{1}{3}$	1	$\frac{1}{2}$	$\frac{2}{3}$	$\frac{1}{3}$	$\frac{1}{2}$	$\frac{1}{2}$	1	$\frac{2}{3}$	$\frac{1}{2}$

Tabela 6.3: Resultado da relaxação linear

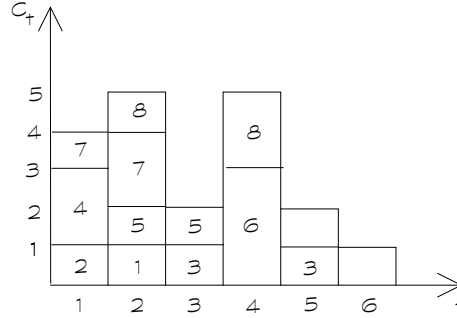


Figura 6.2: Solução da relaxação linear do exemplo

No período 1 é violada uma desigualdade do tipo (6.21):

$$\begin{aligned} x_{41} + x_{71} &\leq \left\lfloor \frac{4-2}{3} \right\rfloor + 1 = 1 \\ 1 + \frac{1}{3} &= \frac{4}{3} \end{aligned}$$

No período 4, é violada uma desigualdade do tipo (6.20):

$$\begin{array}{rclcl} x_{64} & + & x_{84} & \leq & \left\lfloor \frac{5}{3} \right\rfloor = 1 \\ 1 & + & \frac{2}{3} & = & \frac{5}{3} \end{array}$$

6.6 Algoritmo de planos de corte

Na Figura 6.3, apresenta-se o fluxograma de um algoritmo genérico de planos de corte (“cutting planes”), que se passa a descrever de uma forma sucinta. Refira-se que a qualidade das desigualdades válidas desenvolvidas foi avaliada com um algoritmo deste tipo.

Para cada instância de um problema de Programação Inteira, começa-se por obter a solução da relaxação linear do modelo. Se a solução obtida for inteira, então é solução óptima para o problema original. Se a solução obtida for fraccionária, utiliza-se uma “heurística de separação” para gerar desigualdades válidas. Se, com esta heurística, não se conseguir gerar nenhuma desigualdade válida, poder-se-á prosseguir, utilizando um procedimento do tipo “branch-and-bound” que, em geral, será mais eficiente após a adição das desigualdades. Se a heurística de separação gerar desigualdades válidas, essas desigualdades são adicionadas ao modelo e resolve-se novamente a relaxação linear (por um processo de re-optimização geralmente muito eficiente). O processo prossegue de uma forma iterativa.

6.6.1 Heurística de separação

A heurística de separação utilizada, aproveita as características observadas nas soluções da relaxação linear, e consiste essencialmente no seguinte:

Dada uma solução fraccionária, para cada um dos tipos de desigualdades válidas (6.20) a (6.25), são percorridos os períodos completamente “carregados” na solução e aos quais esteja associada pelo menos uma variável com valor fraccionário. É então construída uma desigualdade, verificando-se se é ou não violada pela solução fraccionária em análise. A heurística termina quando se tiver encontrado um número máximo pré-determinado de desigualdades válidas, ou quando não for possível gerar mais desigualdades válidas violadas.

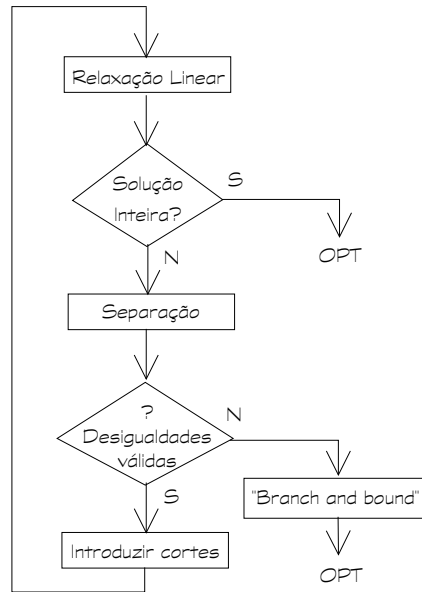


Figura 6.3: Algoritmo de planos de corte

6.7 Resultados computacionais

Para fazer uma primeira avaliação destas desigualdades e do seu impacto na resolução do problema, bem como da qualidade da heurística de separação, foram geradas aleatoriamente 5 instâncias do problema, designadas de PAP1 a PAP5 (ver Tabela 6.4). O algoritmo de planos de corte anteriormente descrito foi aplicado a cada uma destas instâncias, sendo introduzida apenas uma desigualdade válida por ciclo, já que neste contexto se pretende apenas analisar o impacto geral da sua utilização (numa implementação definitiva, seria útil adicionar, em cada ciclo, conjuntos de desigualdades em vez de uma só). Na Figura 6.4, pode-se ver a evolução do valor da função objetivo da relaxação linear para a instância PAP1, à medida que foram sendo introduzidas as desigualdades válidas.

	PAP1	PAP2	PAP3	PAP4	PAP5
T - número de períodos	31	20	34	31	38
E - número de encomendas	36	42	58	63	46
P - número de encomendas pequenas	16	19	26	28	21
M - número de encomendas médias	8	9	13	14	10
G - número de encomendas grandes	12	14	19	21	15
Dados do modelo (com $x_{et} = 0$ se $C_t < q_e$)					
nº de variáveis binárias x_{et}	764	766	1464	1568	1194
nº de restrições	60	62	85	91	75

Tabela 6.4: Características das instâncias PAP

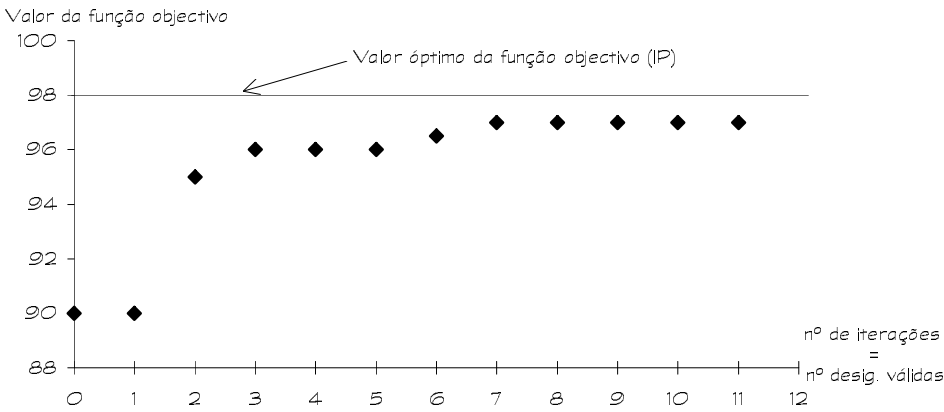


Figura 6.4: Instância PAP1 - evolução do valor da função objectivo com introdução de desigualdades válidas

Na Tabela 6.5, apresentam-se resumidamente os resultados obtidos antes e depois da aplicação do algoritmo de planos de corte às instâncias PAP1 a PAP5. Nesta tabela, os valores apresentados têm a seguinte interpretação:

valor RL	valor da função objectivo da relaxação linear;
valor IP	valor da função objectivo da solução óptima (inteira);
gap (%)	$= \frac{ValorIP - ValorRL}{ValorRL} 100;$

	PAP1	PAP2	PAP3	PAP4	PAP5
Antes da introdução das desigualdades válidas:					
valor RL	90	75	289	178	90
valor IP	98	75	319	205	118
gap (%)	8.89	0	10.38	15.17	31.11
nº de nós	1253	65	*	**	*
tempo (seg.)	10.73	0.77			
Depois da introdução das desigualdades válidas:					
nº de desigualdades válidas	11	4	12	10	12
valor RL	97	75	309	194	98
gap (%)	1.03	0	3.24	5.67	20.41
nº de nós	3	31	16	*	1198
tempo (seg.)	0.33	0.59	0.99		23.64

Tabela 6.5: Instâncias PAP, soluções antes e depois da introdução das desigualdades válidas

nº de nós	número de nós do “branch-and-bound”, necessários para obter a solução IP;
tempo	tempo necessário para obter a solução IP (em segundos);
nº de desig. válidas	número total de desigualdades válidas introduzidas no modelo.

Ainda relativamente à Tabela 6.5, * indica que não foi possível obter o óptimo com o esforço computacional disponível, e ** indica que só foi possível obter o óptimo após introdução das desigualdades válidas.

6.7.1 Conclusões

Nesta secção tentou-se obter limites inferiores para o valor da função objectivo do problema PAP, através da relaxação linear de um modelo de Programação Inteira com a adição de desigualdades válidas que tiram partido da estrutura particular do problema. Essas desigualdades são utilizadas

num algoritmo de planos de corte construído em torno de uma heurística de separação relativamente simples, que aproveita algumas das características observadas nas soluções da relaxação linear, gerando desigualdade violadas nessas soluções. Com este procedimento, conseguiu-se reduzir consideravelmente o “gap” entre a solução óptima do problema e a solução da relaxação linear (ver Tabela 6.5). Estes resultados computacionais, embora preliminares, indiciam que este tipo de abordagem poderá ser bastante útil na obtenção de limites inferiores em problemas de dimensão real, e por consequência, na sua resolução eficiente.

Poderá ainda ser interessante gerar desigualdades válidas de outros tipos, obtidas por exemplo a partir das propriedades do modelo PAP enunciadas na secção 6.3.1.

Capítulo 7

Meta-heurísticas

Nos últimos anos têm sido usados com sucesso em muitos problemas de Optimização Combinatória, procedimentos gerais de carácter heurístico, que se baseiam na aplicação iterada de processos de pesquisa local com algum grau de aleatorização. Entre estes procedimentos, contam-se por exemplo a Pesquisa Tabu, o “Simulated Annealing”, os Algoritmos Genéticos e as Redes Neurais.

Estes procedimentos constituem, em geral, processos de optimização “inspirados” na natureza e podem ser considerados métodos de Pesquisa Local, em que se pretende genericamente minimizar uma função de custo $f(x)$ sobre um conjunto finito e discreto de pontos $\mathcal{X}$, passando iterativamente de uma solução x_n para uma solução x_{n+1} , vizinha de x_n , seguindo uma determinada estratégia de pesquisa.

Dado que, em cada passo, estes procedimentos utilizam uma heurística mais fundamental para passar de uma solução para a seguinte, “guiando” a pesquisa por uma trajectória dentro do conjunto $\mathcal{X}$, convencionou-se chamá-lhes meta-heurísticas.

A eficiência e a eficácia destas meta-heurísticas depende de inúmeros factores, entre os quais se contam as características específicas do problema a optimizar, e também a estrutura de vizinhança definida e os procedimentos ou regras heurísticas utilizadas na geração das soluções.

Para uma descrição pormenorizada destas meta-heurísticas, recomenda-se a leitura de [Pirlot, 1994].

Neste capítulo, descrevem-se os testes realizados com o problema PAP (ver secção 6.3), utilizando a Pesquisa Tabu e o “Simulated Annealing” (no Apêndice F, faz-se uma descrição pormenorizada de cada uma das instâncias geradas aleatoriamente para este problema).

Começa-se por descrever o algoritmo básico de Pesquisa Local, que servirá de suporte à descrição posterior dos algoritmos desenvolvidos especificamente para o problema PAP. Descreve-se também um procedimento que permite melhorar qualquer solução obtida e que se baseia na propriedade 1 do problema PAP (apresentada na secção 6.3.1 e demonstrada no Apêndice G).

São ainda descritas as duas estratégias de pesquisa utilizadas (a Pesquisa Tabu e o “Simulated Annealing”) e, por último, é apresentada uma comparação do desempenho respectivo.

7.1 Notação

$\mathcal{X}$	conjunto de todas as soluções admissíveis x de uma determinada instância de um problema;
x_0	solução inicial;
$\bar{x}$	solução actual;
x_n	solução obtida no passo n ;
x^*	melhor solução obtida até ao momento;
$f(x_n)$	valor da função objectivo para a solução x_n ;
$\mathcal{N}(x_n)$	vizinhança da solução x_n (isto é, o conjunto das soluções que se podem atingir a partir de x_n , realizando movimentos elementares específicos, cuja definição caracteriza a vizinhança em causa). Por convenção supõe-se que nenhuma solução é vizinha de si mesma:

$$\mathcal{N}_i(x_n) \subset \mathcal{X}$$

$$\forall_{x_n \in \mathcal{X}} \quad x_n \notin \mathcal{N}_i(x_n);$$

$\mathcal{SN}_i(x_n)$ subvizinhança da solução x_n , de dimensão i (em geral este subconjunto da vizinhança é gerado aleatoriamente):

$$\mathcal{SN}_i(x_n) \subseteq \mathcal{N}(x_n) \wedge |\mathcal{SN}_i(x_n)| = i$$

7.2 Pesquisa Local (LS - “Local Search”)

O algoritmo de Pesquisa Local parte de uma solução inicial arbitrária x_0 e em cada iteração é escolhida a melhor solução $x_{n+1} \in \mathcal{N}(x_n)$ (a solução com menor valor para a função objectivo, neste caso em que o problema tratado é um problema de minimização). Só se passa de uma solução x_n para uma solução x_{n+1} , se $f(x_{n+1}) \leq f(x_n)$. Trata-se de uma estratégia “gulosa” que leva geralmente a que o processo “caia” muito rapidamente num mínimo local.

No Algoritmo 7.1 está representada uma “restrição” do procedimento que se acabou de descrever, onde em cada iteração se geram apenas i elementos da vizinhança $\mathcal{N}(x_n)$ ($\mathcal{SN}_i(x_n)$).

```

gerar  $x_0 \in \mathcal{X}$ 
 $x^* = x_0$ 
 $n = 0$ 
 $stop = falso$ 
repetir
    gerar aleatoriamente  $\mathcal{SN}_i(x_n)$ 
    encontrar o melhor  $\bar{x} \in \mathcal{SN}_i(x_n)$ 
        (tal que  $f(\bar{x}) \leq f(\bar{x}_i) \forall \bar{x}_i \in \mathcal{SN}_i(x_n)$ )
    se  $f(\bar{x}) \geq f(x_n)$ 
         $stop = verdadeiro$ 
    se  $f(\bar{x}) < f(x^*)$ 
         $x^* = \bar{x}$ 
     $n = n + 1$ 
     $x_n = \bar{x}$ 
até  $stop$ 

```

Algoritmo 7.1: Algoritmo de Pesquisa Local

O Algoritmo 7.1 é independente do tipo de problema, reflectindo-se a especificidade de cada problema na geração de soluções iniciais e na estrutura de vizinhança. São esses dois tópicos que se abordarão de seguida.

7.3 Geração de soluções iniciais

Foram desenvolvidas duas heurísticas construtivas básicas, para geração de soluções iniciais x_0 do problema PAP e que se passam a descrever.

7.3.1 Heurística Básica para geração de soluções iniciais

Cada encomenda é colocada num período o mais próximo possível de d_e , de modo a minimizar o valor da função objectivo (ver Algoritmo 7.2). Começa-se por colocar as encomendas maiores, reduzindo assim a possibilidade de se obter uma solução não admissível.

```

ordenar encomendas por ordem decrescente de tamanho
para cada encomenda
    encontrar período com capacidade disponível e menor custo
    colocar encomenda nesse período
  
```

Algoritmo 7.2: Heurística Básica para geração de uma solução inicial

7.3.2 Heurística Aleatória para geração de soluções iniciais

Para cada encomenda ainda não colocada (começando por colocar as encomendas maiores e reduzindo assim a possibilidade de se obter uma solução não admissível), são gerados aleatoriamente os períodos de colocação (números de 1 a T), até que seja gerado um período com capacidade disponível para a colocação da encomenda (ver Algoritmo 7.3).

```

ordenar encomendas por ordem decrescente de tamanho
para cada encomenda
    colocou = falso
    enquanto (colocou = falso)
        gerar número aleatório  $t_e$ 
        se ( $CapDisp_{t_e} \geq q_e$ )
            colocar  $e$  em  $t_e$ 
            colocou = verdadeiro
  
```

Algoritmo 7.3: Heurística Aleatória para geração de uma solução inicial

7.4 Melhoramento de soluções

Se uma instância do problema PAP tiver soluções admissíveis, então existe uma solução óptima na qual as encomendas de cada tamanho estão colocadas por ordem de data de entrega (EDD) (ver secção 6.3.1). Assim, qualquer que seja a solução de que se parte, se cada conjunto de encomendas (de 1, 2 ou 3 unidades) for recolocado por ordem de datas de entrega, obtém-se uma solução que, no caso mais desfavorável, é pelo menos tão boa como a solução original.

Considere-se que:

$\mathcal{I}_n$ é o conjunto de todas as encomendas com $q_e = n$;

$\mathcal{P}_n$ é o conjunto de todos os períodos onde se encontram colocadas as encomendas do conjunto $\mathcal{I}_n$ na solução que se pretende melhorar (se estiverem i encomendas colocadas num determinado período, esse período estará i vezes representado em $\mathcal{P}_n$).

O procedimento de melhoria de soluções baseia-se na propriedade referida e encontra-se representado no Algoritmo 7.4.

```

para( $n = 1; n \leq 3; n = n + 1$ )
    ordenar as encomendas de  $\mathcal{I}_n$  por ordem crescente de data de entrega
    ordenar os períodos em  $\mathcal{P}_n$  por ordem crescente
    afectar ordenadamente as encomendas de  $\mathcal{I}_n$  aos períodos de  $\mathcal{P}_n$ 

```

Algoritmo 7.4: Heurística MelhoraSolução

7.5 Estruturas de vizinhança e subvizinhança

Para cada solução x_n , uma vizinhança de x_n ($\mathcal{N}(x_n)$), é o conjunto das soluções que se podem atingir a partir de x_n , realizando movimentos elementares específicos, cuja definição caracteriza a vizinhança em causa.

Para este problema, foram considerados dois tipos de estruturas de vizinhança, que se poderiam designar por “naturais”: uma vizinhança a que se chamou $\mathcal{N}_{Desloca(dist)}(x_n)$, em que um vizinho se obtém por deslocamento de uma encomenda e do período t_e em que estava colocada, para um período

t ; e uma vizinhança $Troca(dist)$, em que um vizinho se obtém trocando a posição de duas encomendas.

Ambas as vizinhanças dependem de um parâmetro $dist$, que corresponde, no caso da vizinhança $Desloca(dist)$, à maior deslocação possível para uma encomenda (“distância” relativamente à posição inicial). No caso da vizinhança $Troca(dist)$, o parâmetro $dist$ corresponde ao maior afastamento possível entre as duas encomendas a trocar.

Dada uma solução x_n , seja t_e o período em que está colocada a encomenda e , então o valor da função objectivo (com a qual se pretende penalizar tanto a produção antes da data de entrega como a produção depois da data de entrega, numa perspectiva “just-in-time” (JIT)) para uma solução x_n é dado por:

$$f(x_n) = \sum_e JIT(e, t_e) \quad (7.1)$$

onde:

$$JIT(e, t_e) = \begin{cases} (d_e - t)q_e & \text{se } t_e \leq d_e \\ 2q_e & \text{se } t_e = d_e + 1 \\ (d_e - t)^2 q_e & \text{se } t_e > d_e + 1 \end{cases} \quad (7.2)$$

Se $\mathcal{N}(x_n)$ for demasiado grande para ser totalmente pesquisada, pode-se gerar aleatoriamente apenas uma subvizinhança de $\mathcal{N}(x_n)$, de dimensão i , $\mathcal{SN}_i(x_n)$:

$$\mathcal{SN}_i(x_n) \subseteq \mathcal{N}(x_n) \wedge |\mathcal{SN}_i(x_n)| = i$$

7.5.1 Vizinhança $\mathcal{N}_{Desloca(dist)}(x_n)$

Considerando que a solução actual é x_n , a vizinhança $\mathcal{N}_{Desloca(dist)}(x_n)$ é constituída por todas as soluções admissíveis que se atingem a partir de x_n por deslocamento de cada encomenda de, no máximo, $dist$ períodos a partir do período inicial.

A geração aleatória de um vizinho $\bar{x}$ de uma solução x_n , processa-se do seguinte modo (ver Algoritmo 7.5):

É gerado aleatoriamente um par (e, t) , constituído por uma encomenda e e um novo período para colocação da encomenda, $t \neq t_e$. A encomenda e , é colocada no período t se a capacidade disponível nesse período for igual ou superior a q_e .

Como a encomenda estava inicialmente colocada no período t_e , o novo valor da função objectivo $f(\bar{x})$ é calculado do seguinte modo:

$$f(\bar{x}) = f(x_n) - JIT(e, t_e) + JIT(e, t) \quad (7.3)$$

Considerando que o número total de encomendas é E , um limite superior (nalguns casos relativamente grosseiro) para o tamanho da vizinhança $Desloca(dist)$ seria dado por:

$$|\mathcal{N}_{Desloca(dist)}(x_n)| = E * (2 * dist - 1) \quad (7.4)$$

Este limite superior é obtido, partindo do princípio que é possível fazer todas as “deslocações” das encomendas, isto é, que há uma grande folga de capacidade. À medida que a folga de capacidade diminui, também o limite superior diminui, sendo nulo para uma folga nula, (isto é, se a folga de capacidade for nula, então o tamanho da vizinhança $Desloca(dist)$ é zero).

```

gerar aleatoriamente número de encomenda  $e$ 
gerar aleatoriamente período  $t$  ( $t_e - dist \leq t \leq t_e + dist$ )
se ( $CapDisp_t \geq q_e$ )
     $t_e = t$ 
     $f(\bar{x}) = f(x_n) - JIT(e, t_e) + JIT(e, t)$ 

```

Algoritmo 7.5: Obtenção de $\bar{x} \in \mathcal{N}_{Desloca(dist)}(x_n)$

7.5.2 Vizinhança $\mathcal{N}_{Troca(dist)}(x_n)$

Considerando que a solução actual é x_n , a vizinhança $\mathcal{N}_{Troca(dist)}(x_n)$ é constituída por todas as soluções admissíveis que se atingem a partir de x_n por troca de cada par de encomendas que se encontrem afastadas de menos do que $dist$ períodos.

A geração aleatória de um vizinho $\bar{x}$ de uma solução x_n , processa-se do seguinte modo (ver Algoritmo 7.6):

É gerado aleatoriamente um par de encomendas (e_1, e_2) , com a encomenda e_1 colocada no período t_1 e a encomenda e_2 colocada no período t_2 (estas encomendas estão afastadas de menos do que $dist$ ($|t_1 - t_2| \leq dist$)). As encomendas são trocadas se houver capacidade disponível.

No caso de as encomendas serem trocadas, a variação na função objectivo é a seguinte:

$$f(\bar{x}) = f(x_n) - JIT(e_1, t_1) - JIT(e_2, t_2) + JIT(e_1, t_2) + JIT(e_2, t_1) \quad (7.5)$$

Considerando que o número total de encomendas é E , um limite superior (grosseiro) para o tamanho da vizinhança $Troca(dist)$ seria dado por:

$$|\mathcal{N}_{Troca(dist)}(x_n)| = \binom{E}{2} = E * (E - 1) \quad (7.6)$$

Este limite superior é obtido, partindo do princípio de que todas as encomendas estão afastadas de menos do que $dist$ e que é possível fazer todas as “trocas” de encomendas.

No caso de encomendas do mesmo tamanho, essa troca é sempre possível. No caso de as encomendas serem de tamanhos diferentes, só será sempre possível proceder à troca, se houver folga de capacidade. À medida que a folga de capacidade diminui, também o limite superior referido diminui. Com folga nula, só é possível trocar encomendas do mesmo tamanho.

```

gerar aleatoriamente no encomenda  $e_1$ 
gerar aleatoriamente no encomenda  $e_2$  ( $t_1 - dist \leq t_2 \leq t_1 + dist$ )
se houver capacidade
  trocar
   $f(\bar{x}) = f(x_n) - JIT(e_1, t_1) - JIT(e_2, t_2) + JIT(e_1, t_2) + JIT(e_2, t_1)$ 

```

Algoritmo 7.6: Obtenção de $\bar{x} \in \mathcal{N}_{Troca(dist)}(x_n)$

7.5.3 Vizinhança $\mathcal{N}_{Mista(dist,p)}(x_n)$

Na sequência de testes realizados com as vizinhanças $\mathcal{N}_{Desloca(dist)}(x_n)$ e $\mathcal{N}_{Troca(dist)}(x_n)$, optou-se por gerar uma outra vizinhança (de carácter aleatório) que correspondesse a uma “mistura” das duas vizinhanças anteriormente descritas, no sentido de aproveitar as vantagens de uma e de outra.

Para definir a vizinhança $Mista(dist, p)$, é necessário o parâmetro $dist$, que corresponde ao parâmetro $dist$ definido para as vizinhanças $\mathcal{N}_{Desloca(dist)}(x_n)$ e $\mathcal{N}_{Troca(dist)}(x_n)$ e ainda um parâmetro p , que corresponde à probabilidade de se usar a vizinhança $\mathcal{N}_{Desloca(dist)}(x_n)$.

O procedimento seguido é apresentado no Algoritmo 7.7.

```

para cada solução  $x_n$ 
  gerar aleatoriamente  $p_1$  ( $0 \leq p_1 \leq 1$ )
  se  $p_1 \leq p$ 
     $\mathcal{N}_{Desloca(dist)}(x_n)$ 
  se  $p_1 > p$ 
     $\mathcal{N}_{Troca(dist)}(x_n)$ 

```

Algoritmo 7.7: Procedimento seguido na vizinhança $\mathcal{N}_{Mista(dist,p)}(x_n)$

7.6 Pesquisa Tabu (TS - “Taboo Search”)

... Dado que somos seres com memória e usamos uma grande variedade de funções de memória que nos ajudam a construir um caminho ao longo de um labirinto de considerações associadas à resolução de problemas, é razoável que tentemos que os nossos métodos de resolução tenham essas mesmas capacidades. ...

[Hammer, 1993]

A Pesquisa Tabu (tal como o “Simulated Annealing”), baseia-se na hipótese de que é possível evitar que uma solução “caia irremediavelmente” num mínimo local, tal como pode acontecer no caso do algoritmo básico de Pesquisa Local (ver secção 7.2), admitindo passar para soluções piores, sempre que não haja soluções melhores que a actual, na vizinhança (ou sub-vizinhança) da solução actual. Essa possibilidade que mantém ou melhora a eficácia do procedimento, tem como contrapartida óbvia uma diminuição da sua eficiência computacional, já que exige uma mais vasta exploração do espaço das soluções.

Em termos gerais, poder-se-ia dizer que o que distingue a Pesquisa Tabu de uma Pesquisa Local tradicional é o conceito de memória, orientando a trajectória das soluções em função das soluções anteriores.

A Pesquisa Tabu “foge” de um mínimo local onde tenha caído, escolhendo para solução x_{n+1} , a melhor solução pertencente a $\mathcal{N}(x_n)$ (ou a $\mathcal{SN}_i(x_n)$ se $\mathcal{N}(x_n)$ for demasiado grande para ser totalmente explorada), mesmo que $f(x_{n+1}) > f(x_n)$.

Se a estrutura de vizinhança for simétrica, isto é, se

$$x_{n+1} \in \mathcal{N}(x_n) \Leftrightarrow x_n \in \mathcal{N}(x_{n+1})$$

há sempre a possibilidade de o processo entrar em ciclo.

A Pesquisa Tabu introduz uma estrutura de memória que proíbe ou penaliza certos movimentos, impedindo que o processo retorne a uma solução visitada recentemente. Essa estrutura de memória baseia-se na construção de uma lista contendo informação associada aos últimos movimentos realizados, a que se chama *lista tabu*.

A *lista tabu* assume neste método uma particular importância, e a ideia é guardar na *lista tabu* os últimos pares (x_n, x_{n+1}) de soluções visitadas em sequência. Se o par (x_n, x_{n+1}) estiver na *lista tabu*, então o movimento da solução x_n para a solução x_{n+1} é tabu (isto é, não pode ser realizado). Se o tamanho da *lista tabu* for k , podem ser armazenados k movimentos e conseguem-se evitar ciclos de tamanho inferior a k . Quando a *lista tabu* está completa, sempre que entra um novo par de soluções, sai o par mais antigo (lista “FIFO - first in, first out”).

Por razões de limite de memória e de eficiência computacional, o tamanho da *lista tabu* é finito, sendo teoricamente sempre possível que o processo entre em ciclo. Para o evitar, é necessário definir um critério de paragem para o algoritmo.

Em resumo, pode-se pois afirmar que na Pesquisa Tabu, o critério utilizado para passar de uma solução x_n , para uma solução $x_{n+1} \in \mathcal{N}(x_n)$, é que x_{n+1} seja a melhor solução da subvizinhança, e que o “movimento” que é realizado não pertença à *lista tabu*.

O que se pretende evitar com a *lista tabu* é, como foi referido, a repetição do “movimento” de passagem de uma solução x_n para uma solução x_{n+1} . No entanto, para realizar esse objectivo de uma forma absoluta, seria necessário armazenar completamente as duas soluções, e seria além disso necessário comparar uma grande quantidade de dados, para verificar se um determinado movimento estava ou não na *lista tabu*. Assim, opta-se normalmente por guardar na *lista tabu* apenas algumas características dos movimentos realizados. Essa “simplificação” da *lista tabu* pode ter a desvantagem de restringir demasiado os movimentos, impedindo o algoritmo de atingir uma

solução boa. Para que tal não aconteça, existe o conceito de “nível de aspiração” que consiste em se aceitar excepcionalmente um movimento tabu, desde que a nova solução tenha determinadas características desejáveis, tais como por exemplo um valor da função objectivo melhor do que o melhor obtido até ao momento.

Como a Pesquisa Tabu é uma técnica relativamente difundida, optou-se por não fazer aqui a sua descrição detalhada, recomendando-se a leitura de [Hammer, 1993] ou [Pirlot, 1994], para uma descrição mais pormenorizada.

Para o problema em estudo neste capítulo, consideraram-se, em alternativa, os quatro tipos de listas que a seguir se descrevem.

Lista tabu 1 (encomendas): são colocadas na lista tabu as encomendas cuja posição foi alterada no último movimento;

Lista tabu 2 (períodos): são colocados na lista tabu os períodos nos quais foi colocada/retirada uma encomenda no último movimento;

Lista tabu 3 (encomenda/período): são colocados na lista tabu pares (encomenda, período), que correspondem a cada encomenda cuja posição foi alterada, e ao seu novo período de colocação;

Lista tabu 4 (encomenda/sentido oposto ao do movimento): se o período de colocação de uma encomenda foi alterado de t_1 para t_2 , com $t_2 > t_1$, então o movimento foi realizado no sentido positivo, e é armazenado na *lista tabu* o par (encomenda/sentido negativo), para evitar um movimento de t_2 para t_3 (com $t_3 < t_2$).

O procedimento de Pesquisa Tabu está representado no Algoritmo 7.8. O algoritmo pára (condição de paragem), quando não é encontrado nenhum vizinho admissível depois de um número limite de tentativas, ou quando o número de iterações excede um determinado limite.

```

gerar  $x_0 \in \mathcal{X}$ 
 $x^* = x_0$ 
 $n = 0$ 
 $stop = falso$ 
repetir
    gerar aleatoriamente  $\mathcal{SN}_i(x_n)$ 
    encontrar o melhor  $\bar{x} \in \mathcal{SN}_i(x_n)$ 
        (tal que  $\bar{x}$  corresponde a  $\min_i f(\bar{x}_i)$ 
        e movimento não pertence à lista tabu,
        ou movimento pertence à lista tabu
        e nova solução ultrapassa nível de aspiração)
    se condição de paragem
         $stop = verdadeiro$ 
    se  $f(\bar{x}) < f(x^*)$ 
         $x^* = \bar{x}$ 
     $n = n + 1$ 
     $x_n = \bar{x}$ 
até stop

```

Algoritmo 7.8: Algoritmo de Pesquisa Tabu

7.7 “Simulated Annealing” (SA)

... na sua forma original o algoritmo de “Simulated Annealing” baseia-se na analogia entre a simulação da cristalização de sólidos e o problema relacionado com a resolução de problemas de Optimização Combinatória de grande dimensão. É por essa razão que o algoritmo se chama “Simulated Annealing”. Em física, o termo “annealing” refere-se ao processo físico segundo o qual um sólido mergulhado num banho quente é aquecido, subindo a temperatura do banho até um valor máximo, para o qual todas as partículas do sólido se posicionam aleatoriamente, seguido de um arrefecimento gradual e lento da temperatura do banho. Deste modo, todas as partículas se posicionam no estado de menor energia da estrutura cristalina, desde que a temperatura inicial seja suficientemente elevada e desde que o arrefecimento seja realizado muito lentamente.

[Van Laarhoven and Aarts, 1987, pp. 7]

Em termos gerais, poder-se-ia dizer que o que distingue o “Simulated Annealing” de uma Pesquisa Local tradicional é aceitar soluções piores com uma determinada probabilidade que vai diminuindo ao longo do tempo, “arrefecendo” a pesquisa. A fase de arrefecimento do processo de “annealing”, pode ser descrita do seguinte modo: Se a diferença de nível de energia entre o estado actual e o estado com uma pequena perturbação ΔE é negativa, então a perturbação corresponde a um estado com menor energia e o processo prossegue com o novo estado. Se ΔE é positiva, então a probabilidade de aceitação do estado perturbado é dada por $e^{\frac{-\Delta E}{k_b T}}$. Nesse caso podemos fazer a seguinte analogia entre os parâmetros do processo físico e os parâmetros do “Simulated Annealing” (heurística para resolução de problemas de Optimização Combinatória):

- uma solução no problema de Optimização Combinatória, corresponde a um estado do sólido;
- a função objectivo $f(x)$ corresponde à energia E do estado do sólido;
- o parâmetro de controlo T , a que chamaremos temperatura, corresponde à temperatura no processo físico.

Em [Van Laarhoven and Aarts, 1987] e em [Otten and van Ginneken, 1989], podem ser encontradas descrições pormenorizadas, incluindo a justificação teórica, deste método, e em [Vidal, 1993], alguns exemplos práticos da sua aplicação. Para uma descrição pormenorizada do método em língua portuguesa, com a descrição de uma aplicação prática, recomenda-se a leitura de [Oliveira, 1995].

O Algoritmo 7.9 representa um procedimento genérico de “Simulated Annealing”, ligeiramente diferente da versão tradicional. A diferença, com que se procura flexibilizar o método base, está na possibilidade de se gerar aleatoriamente uma sub-vizinhança de i elementos de x_n ($\mathcal{SN}_i(x_n)$) e escolher o melhor, em vez de se gerar apenas um elemento da vizinhança de x_n .

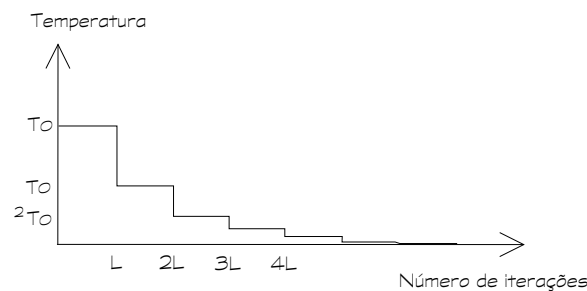


Figura 7.1: Esquema de arrefecimento

7.7.1 Parâmetros

Para implementar o “Simulated Annealing”, é necessário determinar previamente o valor de um conjunto de parâmetros, relacionados com o “arrefecimento” do algoritmo.

```

gerar  $x_0 \in \mathcal{X}$ 
 $T_0$  - temperatura inicial
 $x^* = x_0$ 
 $n = 0$ 
 $stop = falso$ 
repetir
    gerar aleatoriamente  $\mathcal{SN}_i(x_n)$ 
    encontrar o melhor  $\bar{x} \in \mathcal{SN}_i(x_n)$ 
        (tal que  $f(\bar{x}) \leq f(\bar{x}_i) \forall \bar{x}_i \in \mathcal{SN}_i(x_n)$ )
    se  $f(\bar{x}) \leq f(x_n)$ 
         $n = n + 1$ 
         $x_n = \bar{x}$ 
    se  $f(\bar{x}) > f(x_n)$ 
        se  $p \in [0, 1[ \leq e^{-\frac{f(x_n) - f(\bar{x})}{T_n}}$ 
             $n = n + 1$ 
             $x_n = \bar{x}$ 
        se  $p \in [0, 1[ > e^{-\frac{f(x_n) - f(\bar{x})}{T_n}}$ 
             $x_{n+1} = x_n$ 
             $n = n + 1$ 
    se condição de paragem
         $stop = verdadeiro$ 
    se  $f(\bar{x}) < f(x^*)$ 
         $x^* = \bar{x}$ 
    actualiza  $T_n$ 
até stop

```

Algoritmo 7.9: Algoritmo de “Simulated Annealing”

Relativamente à forma da curva de arrefecimento, optou-se por uma forma em degrau (esquema de arrefecimento geométrico), sendo a temperatura alterada por multiplicação por um factor α entre 0 e 1 ($T_j = \alpha^j T_0$), e sendo realizado um número fixo de iterações (L) para cada temperatura. Na Figura 7.1 apresenta-se esquematicamente a evolução da temperatura T , com o número de iterações.

Escolhido o esquema de arrefecimento, foram afinados experimentalmente os valores dos seguintes parâmetros (ver secção seguinte):

T_0 temperatura inicial;

α factor de arrefecimento;

L número de iterações à mesma temperatura.

Finalmente, relativamente à condição de paragem (ver Algoritmo 7.9), optou-se por terminar o algoritmo quando não houver alteração no valor da função objectivo após um número pré-determinado de iterações (múltiplo do número de iterações à mesma temperatura), ou quando não se encontrar nenhum vizinho após um número pré-determinado de tentativas:

$$MaxTentativa = int(30000/TamSubViz)$$

ou se o número de iterações exceder o limite de $40000/TamSubViz$. Nos testes computacionais realizados, a condição de paragem que esteve activa foi sempre a primeira das referidas.

7.8 Determinação de parâmetros

No contexto deste trabalho, o principal objectivo da realização de testes com as meta-heurísticas descritas, foi avaliar, em termos genéricos, das suas potencialidades, e ainda comparar o seu desempenho na obtenção de soluções para o problema PAP.

No entanto, antes de se poder fazer essa avaliação, foi necessário realizar um conjunto de testes que permitisse optar entre as diversas possibilidades referidas na secção anterior, bem como definir os valores dos parâmetros a utilizar. Para poder realizar esses testes de uma forma sistemática, foi de-

envolvido um programa que se pretendeu que fosse o mais modular possível. No Algoritmo 7.10, está representada a estrutura desse programa.

```

ler
  Semente
  NumSolIni
  TipSolIni
  NumIter
  TipViz, dist, p
  TamSubViz
  TipPesquisa
Usar Semente, para gerar vector semente(j) com
número de elementos igual a NumSolIni
para ( $j = 1; j \leq \text{NumSolIni}; j = j + 1$ )
  semente(j)
   $x_0$  do tipo TipSolIni
  repetir
    iterações de TipPesquisa com:
      TipViz, dist, p
      TamSubViz
  até verificar condição de paragem
ou até NumIter
melhora solução

```

Algoritmo 7.10: Estrutura geral do programa para fazer testes com as meta-heurísticas

Em todos os testes foi usada $Semente = 3$ (ponto de partida para os processos de geração aleatória de dados) e $NumSolIni = 40$ (40 soluções iniciais). As soluções iniciais consideradas foram (ver secção 7.3):

$TipSolIni = 1$ heurística aleatória;

$TipSolIni = 2$ heurística básica;

$TipSolIni = 3$ heurística aleatória, com melhoramento de solução;

$TipSolIni = 4$ heurística básica, com melhoramento de solução.

Relativamente às vizinhanças e subvizinhanças, procurou-se determinar o tipo de vizinhança e os respectivos parâmetros, e também definir a dimensão da subvizinhança a utilizar.

$TipViz = 1$ $\mathcal{N}_{Desloca(dist)}$ (determinar $dist$);

$TipViz = 2$ $\mathcal{N}_{Troca(dist)}$ (determinar $dist$);

TipViz = 3 $\mathcal{N}_{Mista(dist,p)}$ (determinar $dist$ e p);

TamSubViz dimensão i da subvizinhança $\mathcal{SN}_i()$.

Para cada um dos tipos de pesquisa considerados, foi ainda necessário avaliar o impacto das escolhas seguintes:

Pesquisa Tabu (TS)

Lista tabu 1 encomendas;

Lista tabu 2 períodos;

Lista tabu 3 (encomenda/período);

Lista tabu 4 (encomenda/sentido oposto ao do movimento);

TamLista tamanho da *lista tabu*.

“Simulated Annealing” (SA)

condição de paragem;

T_0 temperatura inicial;

α parâmetro de arrefecimento;

L número de iterações sem alteração de temperatura.

Apresentam-se a seguir os resultados obtidos nos testes realizados com o problema PAP, para determinação de parâmetros das meta-heurísticas. Os testes são apresentados em dois conjuntos distintos. Num primeiro conjunto de testes, foram determinados alguns parâmetros e feitas algumas escolhas gerais. No segundo conjunto de testes, foram afinados os parâmetros para o “Simulated Annealing”.

Todos os testes foram realizados com as instâncias PAP1 a PAP5, seguindo o algoritmo descrito na secção 7.10, com $Semente = 3$ e 40 soluções iniciais (40 cadeias aleatórias independentes). Os resultados são apresentados através de gráficos construídos da seguinte forma:

Os valores obtidos para a função objectivo em cada uma das 40 soluções iniciais, foram agregados de acordo com os intervalos indicados na legenda de cada figura, calculando-se, para cada intervalo, a percentagem de vezes que o valor da função objectivo estava dentro do intervalo. Essas percentagens foram colocadas na mesma coluna, perfazendo (obviamente) 100%. Os intervalos correspondentes a valores melhores (mais baixos) da função objectivo foram representados por uma cor mais clara (e mais próximos do eixo horizontal) e, à medida que os intervalos correspondem a valores piores da função objectivo, a cor escurece. Deste modo, obtiveram-se gráficos com os quais é simples comparar desempenhos, comparando alturas das manchas da mesma cor. Também se representou nos gráficos o tempo que demorou a obter as 40 soluções. Os tempos estão representados por uma área a escuro, e o eixo das ordenadas correspondente está representado à direita.

7.8.1 Primeiro conjunto de testes

No primeiro conjunto de testes, foi utilizado um procedimento do tipo Pesquisa Tabu mas sem lista, dado que este tipo de pesquisa não depende da afinação específica de parâmetros (não há lista tabu, por isso não é necessário escolher os atributos dos movimentos a colocar na lista tabu, nem o tamanho desta). Os testes realizados tinham como objectivo determinar o tipo de solução inicial a utilizar na Pesquisa Tabu, e ainda o tipo de vizinhança e tamanho de subvizinhança a utilizar nos restantes testes.

Os tipos de soluções iniciais consideradas foram,

$TipSolIni = 1$ heurística aleatória;

$TipSolIni = 2$ heurística básica;

$TipSolIni = 3$ heurística aleatória com melhoramento de solução;

$TipSolIni = 4$ heurística básica com melhoramento de solução.

As vizinhanças consideradas foram:

$TipViz = 1$ $\mathcal{N}_{Desloca(dist)}$ (determinar $dist$);

$TipViz = 2$ $\mathcal{N}_{Troca(dist)}$ (determinar $dist$);

cada uma delas com $dist$: 4, 8 e 12. Consideraram-se ainda os seguintes valores para a dimensão da subvizinhança ($TamSubViz$): 1, 10, 50 e 90.

Os resultados dos testes realizados são apresentados nas Figuras 7.2 a 7.6, onde se representou em abcissas $TipSolIni_TipViz_dist.TamSubViz$. Analisando essas figuras, podem-se tirar as conclusões que a seguir se apresentam.

Tipo de solução inicial ($TipSolIni$)

Os resultados são sempre melhores para $TipSolIni = 3$ e $TipSolIni = 4$ (isto é, quando sobre as heurísticas básicas é efectuado um melhoramento). Comparando os resultados obtidos com $TipSolIni = 3$ e com $TipSolIni = 4$, parece à primeira vista que os resultados são melhores no primeiro caso, no entanto deve-se notar que na maioria dos casos, as soluções obtidas, para cada uma das 40 soluções iniciais, foram quase sempre iguais. Esse facto indicia que a solução inicial obtida pela heurística básica com melhoria de solução ($TipSolIni = 4$), corresponde a um mínimo local profundo, do qual é difícil sair. Por outro lado, a solução inicial aleatória com melhoria de solução ($TipSolIni = 3$), dá origem a soluções finais melhores e piores. Se se considerar que se escolhe apenas a melhor solução obtida nas 40 corridas do algoritmo, então deve-se escolher $TipSolIni = 3$ para solução inicial.

Tipo de vizinhança ($TipViz$)

Os resultados obtidos com a vizinhança $\mathcal{N}_{Desloca(dist)}$ ($TipViz = 1$) são francamente melhores em todas as instâncias excepto na instância PAP4. Explica-se essa situação relativamente à instância PAP4, devido ao facto de a “folga” (relação entre a quantidade total encomendada e a capacidade total), ser muito pequena ($p = 1.02$ na Tabela F.9), o que implica que não haja capacidade suficiente para “deslocar” encomendas.

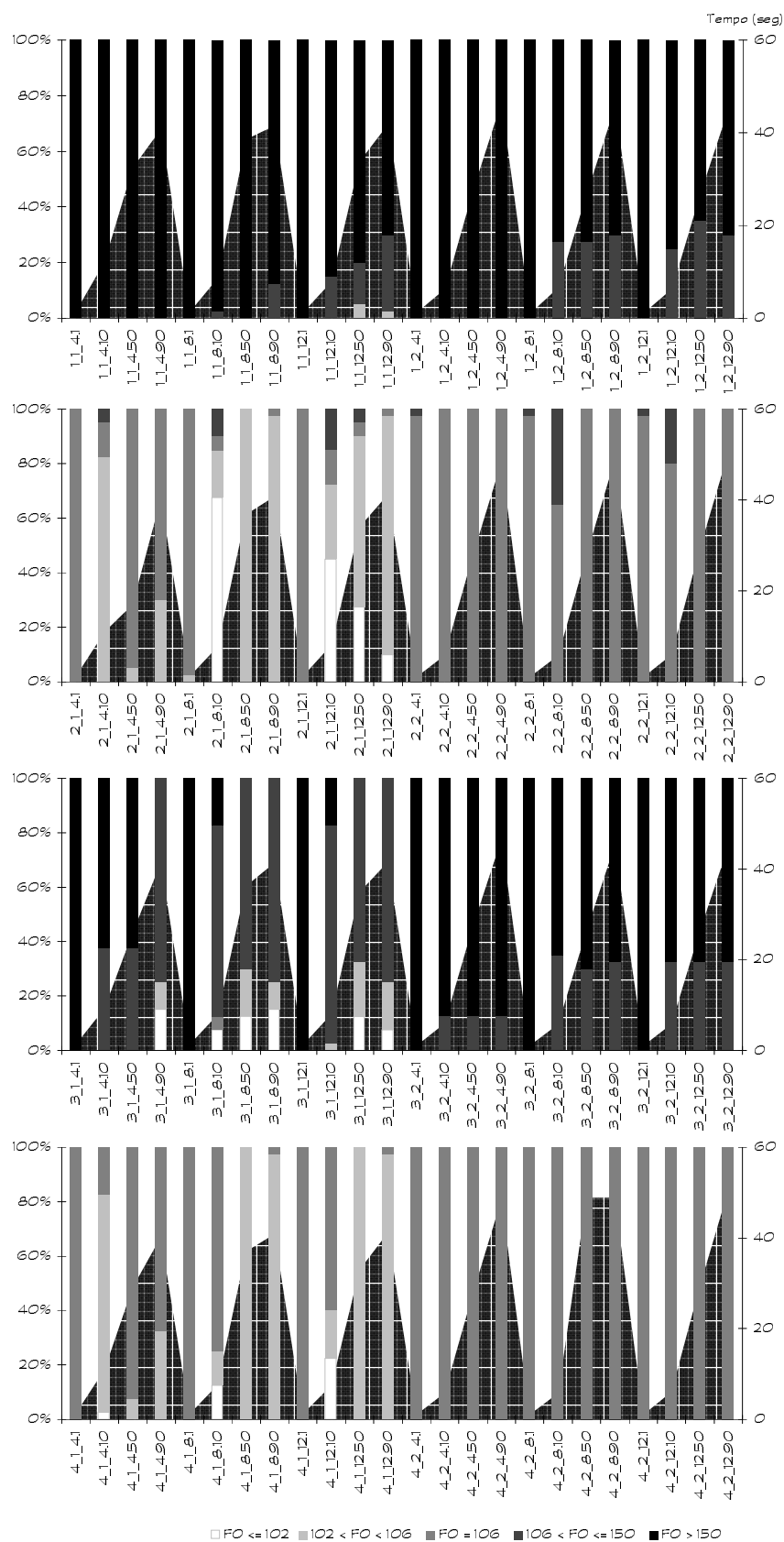
Parâmetro $dist$

Os resultados não foram conclusivos, dado que variações em $dist$ não implicaram variações significativas no valor da função objectivo (sendo portanto

os resultados robustos relativamente a esse parâmetro, para a gama de valores testada). Optou-se então, de um modo arbitrário, por $dist = 8$.

Parâmetro $TamSubViz$

Neste caso, os resultados também não foram conclusivos, dado que variações em $TamSubViz$ não implicaram variações significativas no valor da função objectivo (sendo portanto os resultados robustos relativamente a esse parâmetro, para a gama de valores analisada). Optou-se, então, também arbitrariamente, por usar $TamSubViz = 50$ esse valor de $TamSubViz$ será usado apenas nos testes com Pesquisa Tabu. Nos testes com “Simulated Annealing”, esse parâmetro será deixado em aberto até aos testes comparativos finais entre pesquisas.

Figura 7.2: PAP1 - Determinação de *TipSolIni*, *TipViz*, *dist* e *TamSubViz*

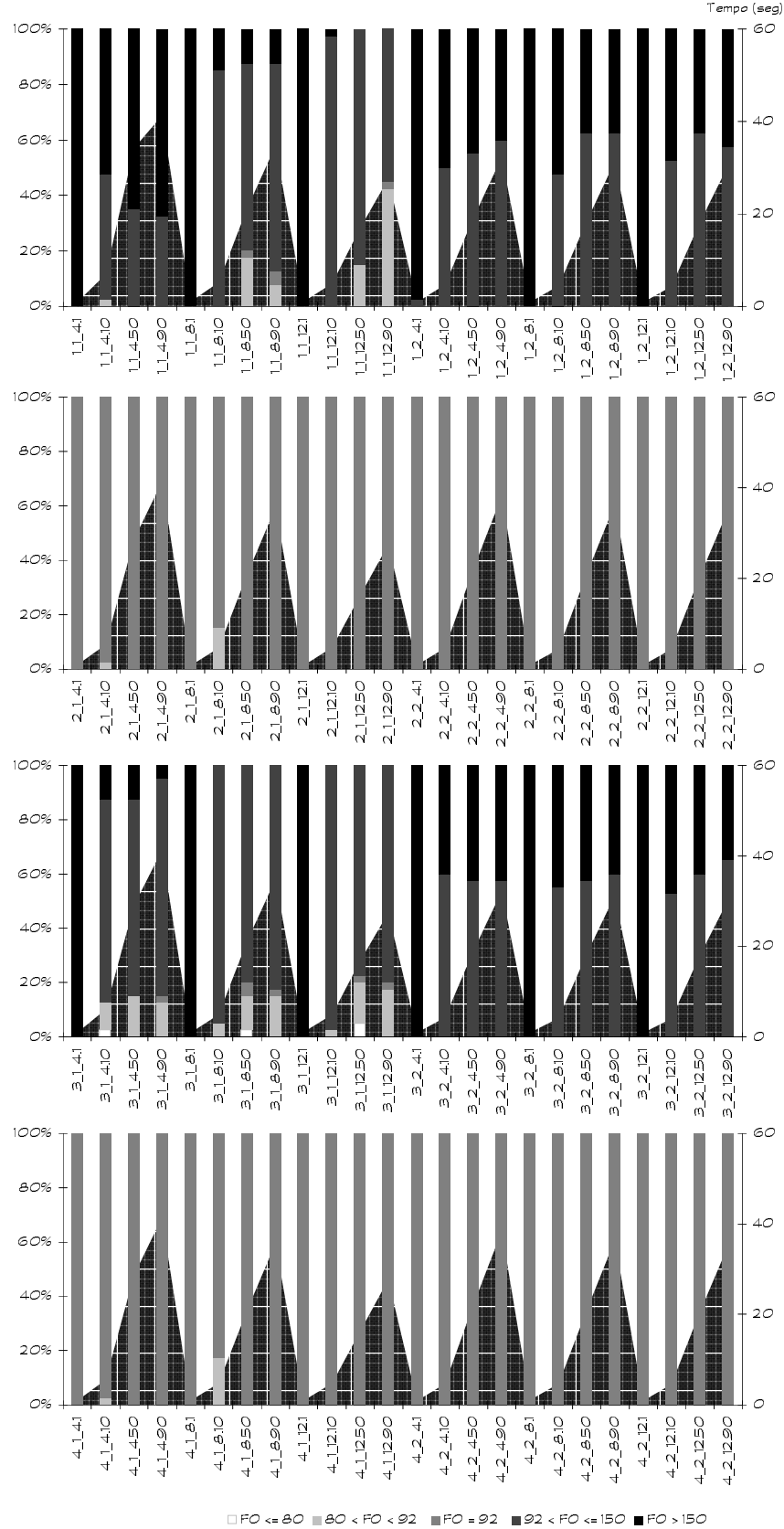
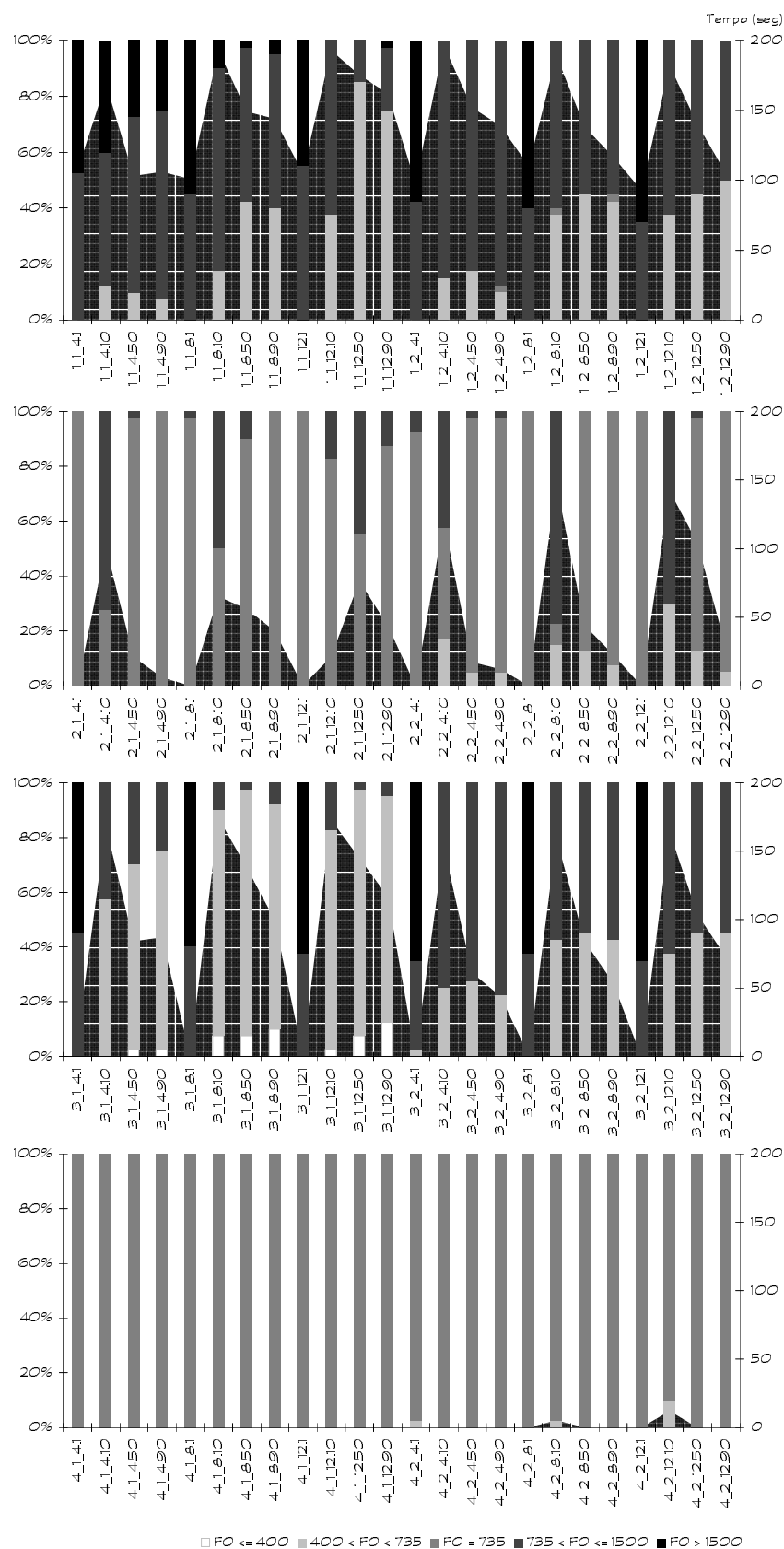


Figura 7.3: PAP2 - Determinação de *TipSolIni*, *TipViz*, *dist* e *TamSubViz*

Figura 7.4: PAP3 - Determinação de *TipSolIni*, *TipViz*, *dist* e *TamSubViz*

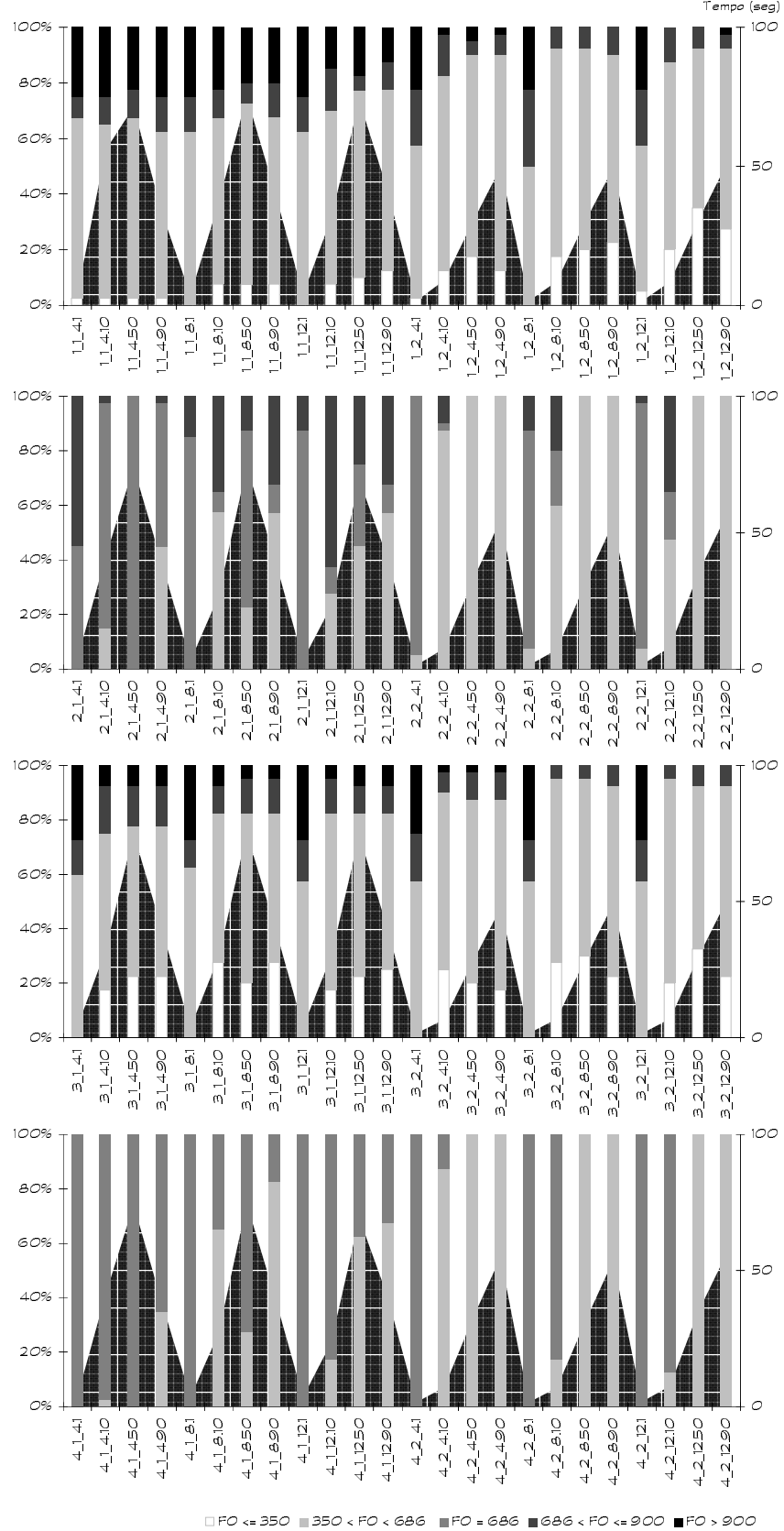
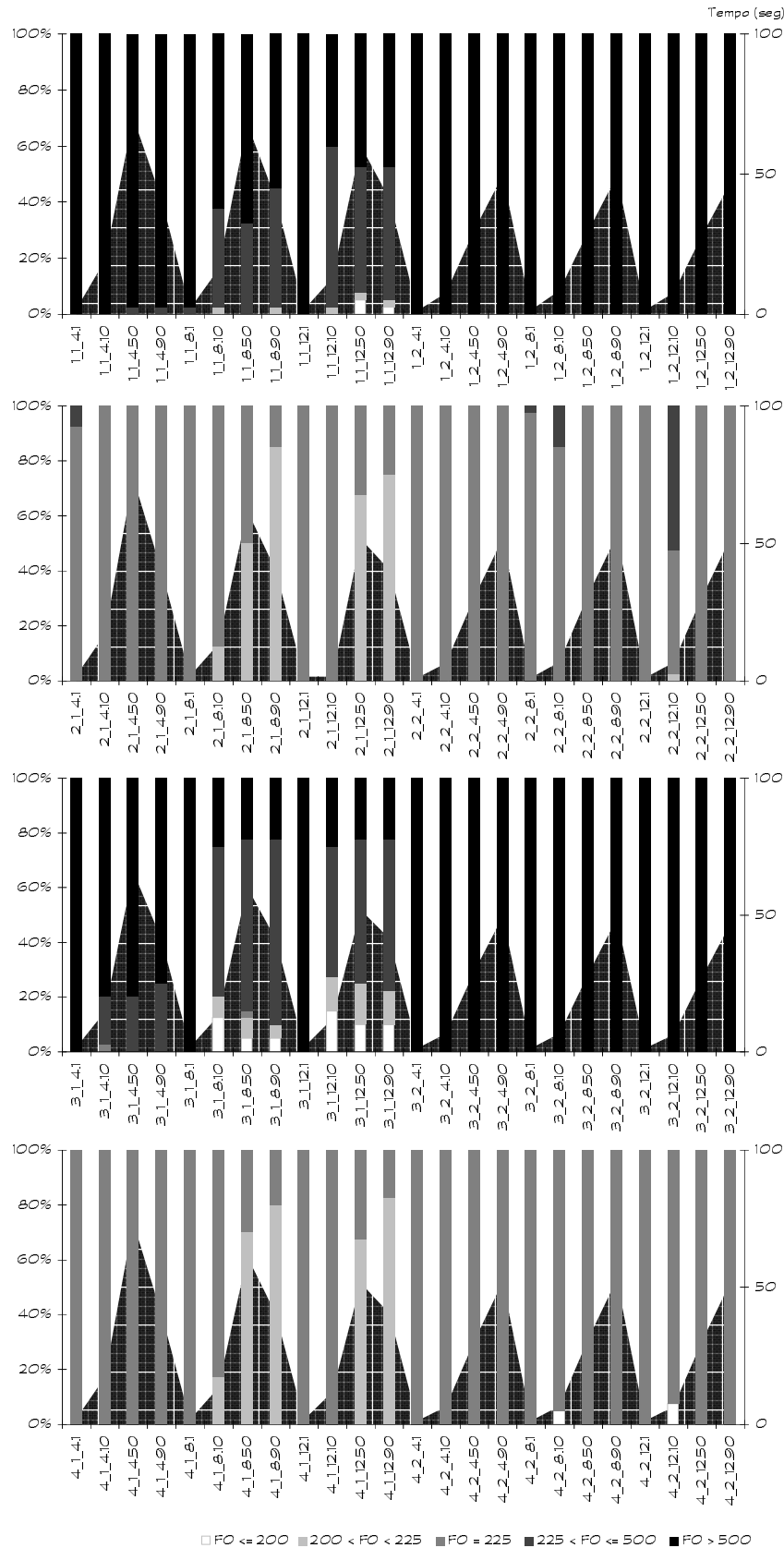


Figura 7.5: PAP4 - Determinação de *TipSolIni*, *TipViz*, *dist* e *TamSubViz*

Figura 7.6: PAP5 - Determinação de *TipSolIni*, *TipViz*, *dist* e *TamSubViz*

Depois de extraídas as conclusões dos primeiros testes, realizaram-se novos testes, com todas as instâncias (PAP1 a PAP5), nos quais se partiu de uma solução inicial obtida pela heurística aleatória seguida de melhoramento de solução ($TipSolIni = 3$). Os testes foram realizados relativamente à vizinhança $\mathcal{N}_{Mista(dist,p)}$ com $dist = 8$, $TamSubViz = 50$

Fez-se variar p entre 0 e 1 por forma a determinar qual a melhor “composição das duas heurísticas básicas a adoptar nesta heurística mista ($p = 0$ corresponde a uma estratégia pura de trocas e $p = 1$ corresponde a uma estratégia pura de deslocamento) (ver Algoritmo 7.7). Os resultados são apresentados nas Figuras 7.7 a 7.11 e os valores em abcissas correspondem neste caso ao valor de p .

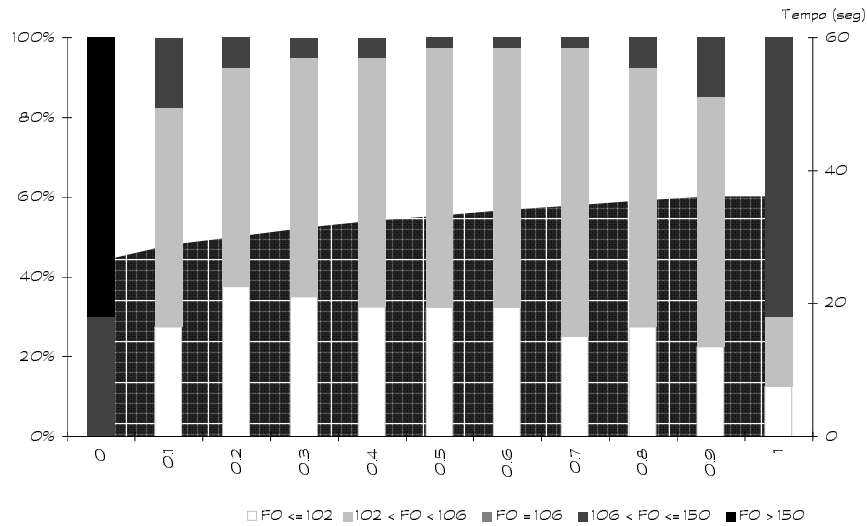


Figura 7.7: PAP1 - Determinação de p em $\mathcal{N}_{Mista(dist,p)}$

Analisando as Figuras 7.7 a 7.11, verifica-se que os resultados obtidos são consideravelmente melhores para valores de p baixos, mas diferentes de zero, tendo-se optado por usar o valor $p = 0.2$. É interessante notar que embora a vizinhança $\mathcal{N}_{Desloca(dist)}$, por si só, fornecesse melhores resultados que a vizinhança $\mathcal{N}_{Troca(dist)}$, considerando a vizinhança $\mathcal{N}_{Mista(dist,p)}$, obtêm-se os melhores resultados para valores baixos de p (isto é, com uma pequena percentagem de utilização da vizinhança $\mathcal{N}_{Desloca(dist)}$).

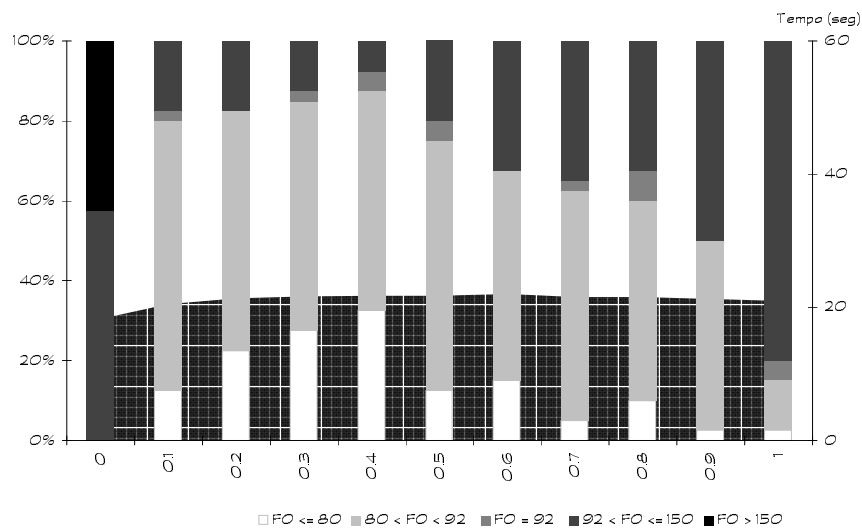


Figura 7.8: PAP2 - Determinação de p em $\mathcal{N}_{Mista(dist,p)}$

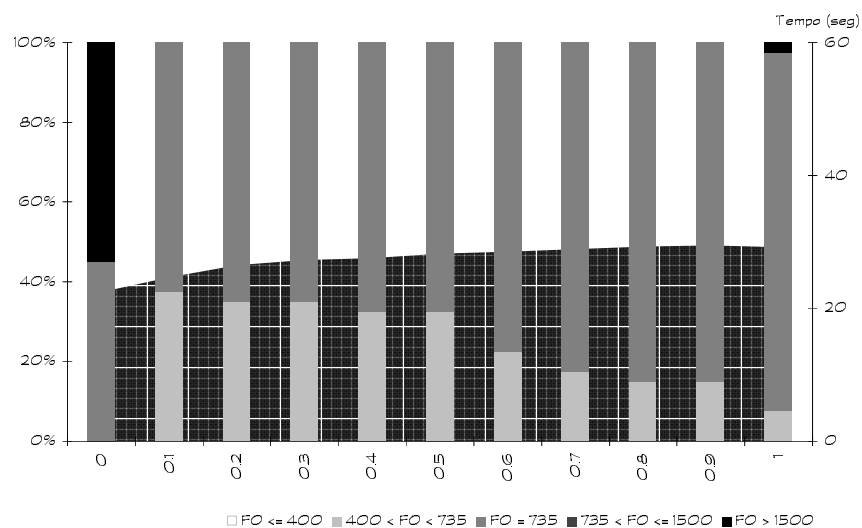


Figura 7.9: PAP3 - Determinação de p em $\mathcal{N}_{Mista(dist,p)}$

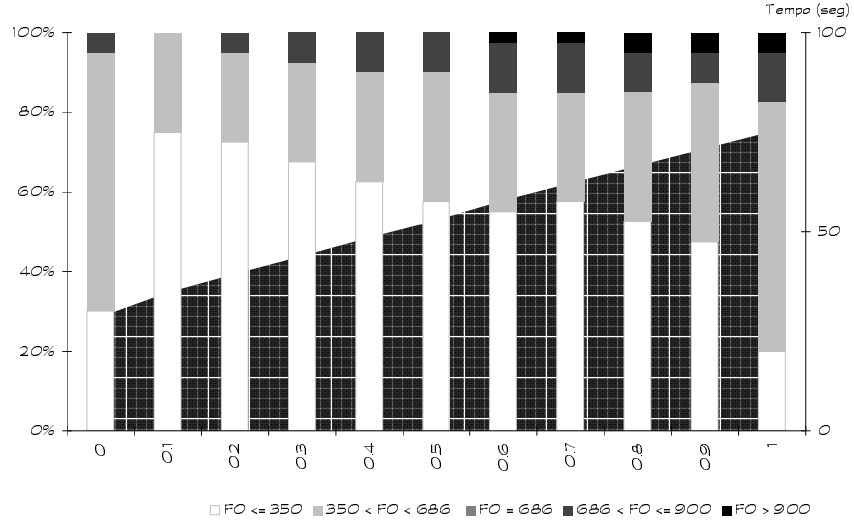


Figura 7.10: PAP4 - Determinação de p em $\mathcal{N}_{Mista(dist,p)}$

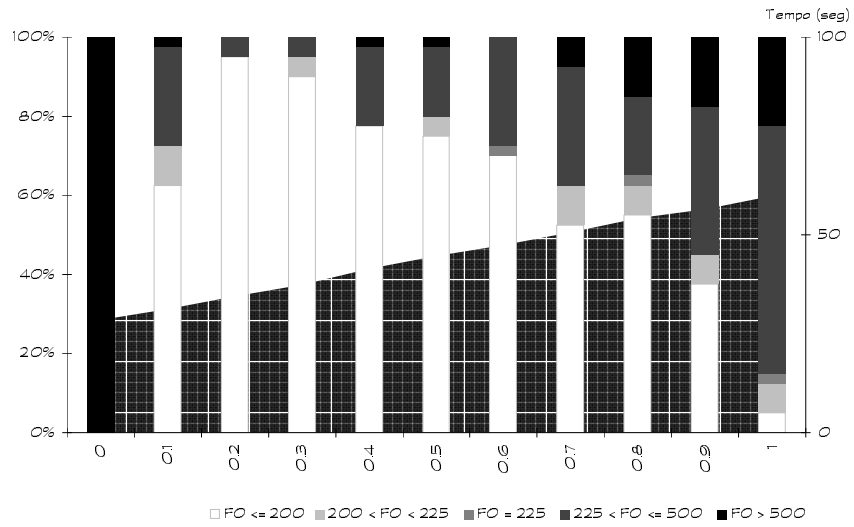


Figura 7.11: PAP5 - Determinação de p em $\mathcal{N}_{Mista(dist,p)}$

7.8.2 Determinação dos valores dos parâmetros para o “Simulated Annealing”

Partindo dos resultados dos testes apresentados anteriormente, realizou-se um novo conjunto de testes para afinação dos parâmetros do “Simulated Annealing”, utilizando novamente todas as instâncias (PAP1 a PAP5), com 40 soluções iniciais e com a vizinhança $\mathcal{N}_{Mista(8,0.2)}$. Nestes testes, utilizou-se no entanto a solução inicial aleatória sem melhoria de solução ($TipSolIni = 1$), por se considerar que a solução inicial para o “Simulated Annealing” deveria ser o mais aleatória possível, dado que inicialmente, com uma temperatura alta, as soluções são fortemente “destruídas”. Além disso, utilizou-se $TamSubViz = 1$, dado que no algoritmo de “Simulated Annealing” “tradicional”, em cada iteração, só se avalia um vizinho da solução actual.

Testes para determinação de T_0

O primeiro conjunto de testes foi realizado para determinar a temperatura inicial T_0 . Para tal, escolheram-se valores “máximos” para os restantes parâmetros: $\alpha = 0.99$; $L = 500$; condição de paragem = $10 * L = 5000$ soluções sem alteração da função objectivo e fizeram-se testes com os seguintes valores de T_0 : 50, 100, 150, 200, 250 e 300. Os resultados dos testes realizados são apresentados nas Figuras 7.12 a 7.16, onde em abcissas estão representados os valores de T_0 .

Analisando as Figuras 7.12 a 7.16, não se notam variações significativas nos resultados obtidos com a variação de T_0 (os resultados são robustos relativamente a esse parâmetro, para a gama de valores analisada). Sendo assim, optou-se por $T_0 = 50$, dado que essa escolha corresponde eventualmente a menores tempos de execução.

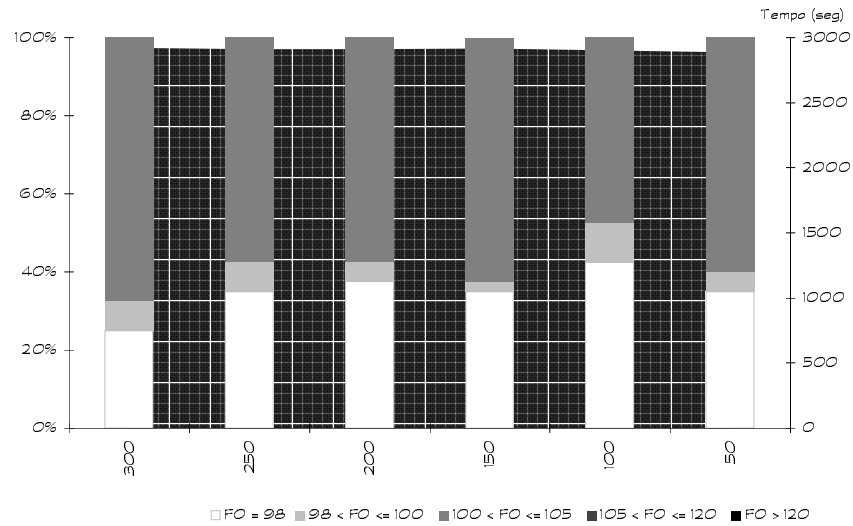


Figura 7.12: PAP1 - SA, determinação de T_0

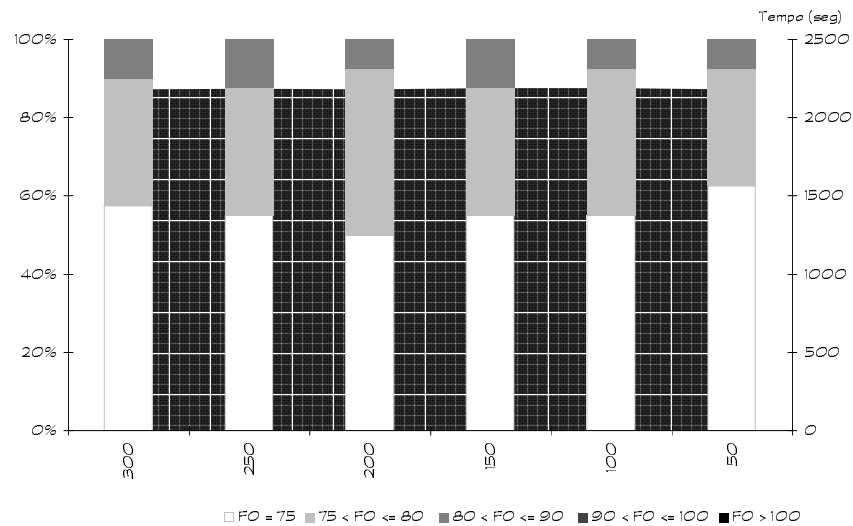
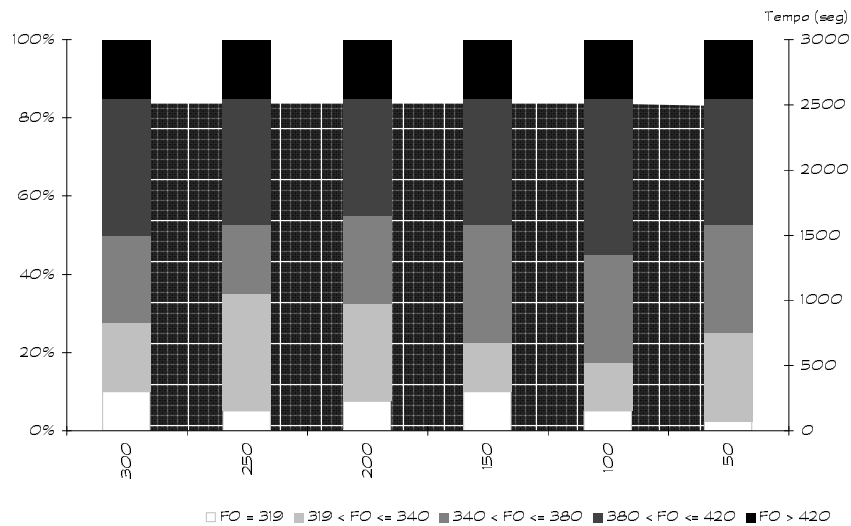
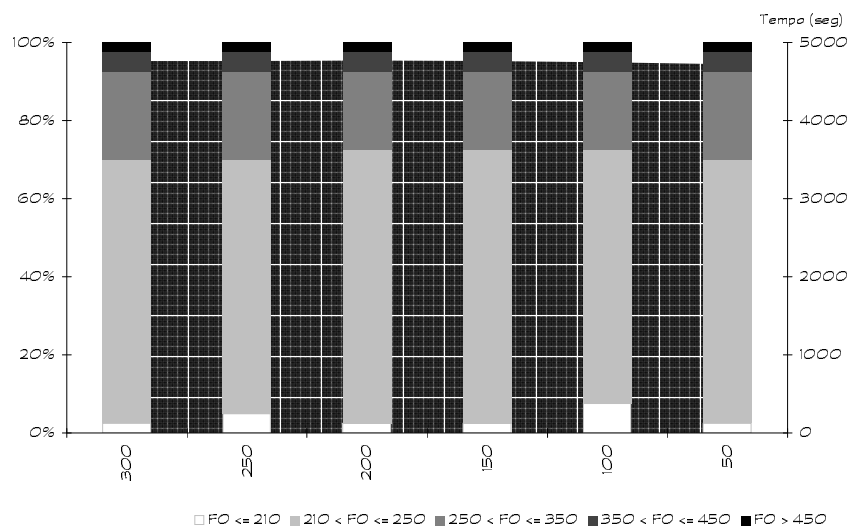


Figura 7.13: PAP2 - SA, determinação de T_0

Figura 7.14: PAP3 - SA, determinação de T_0 Figura 7.15: PAP4 - SA, determinação de T_0

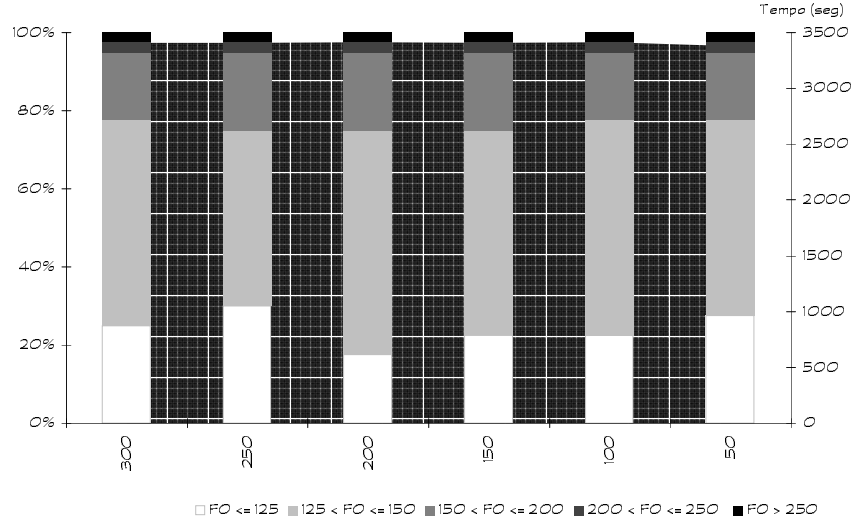
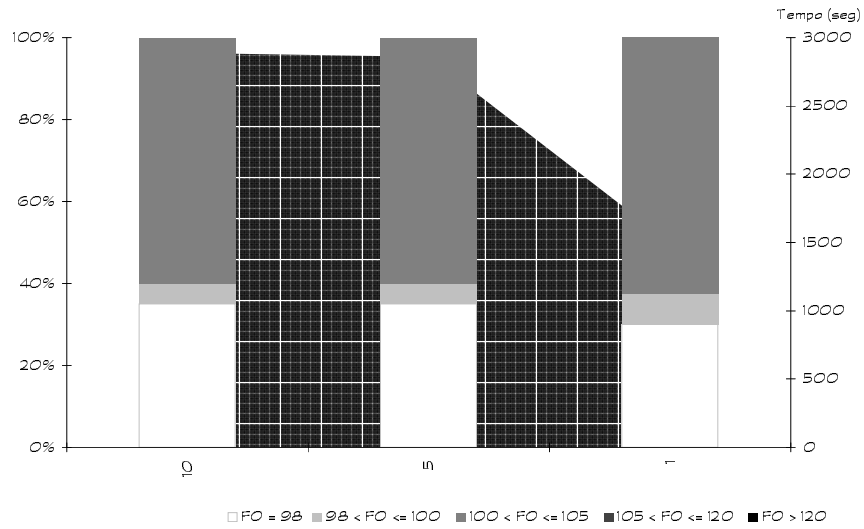
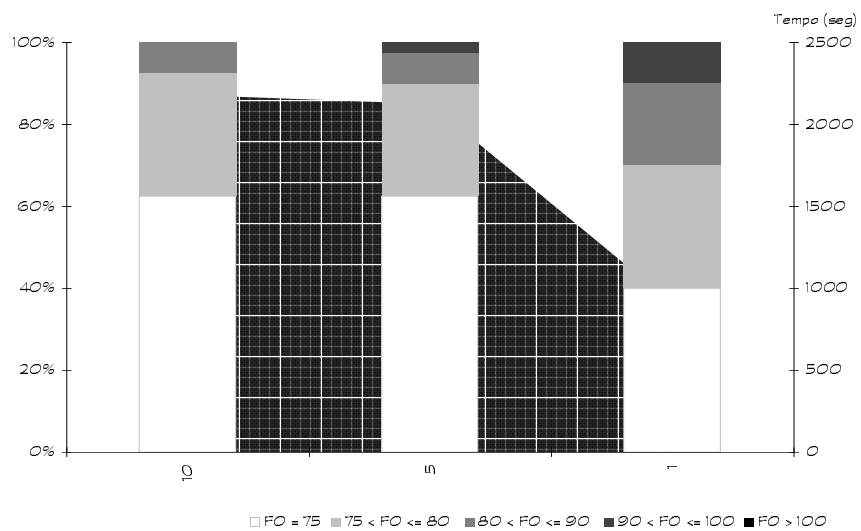


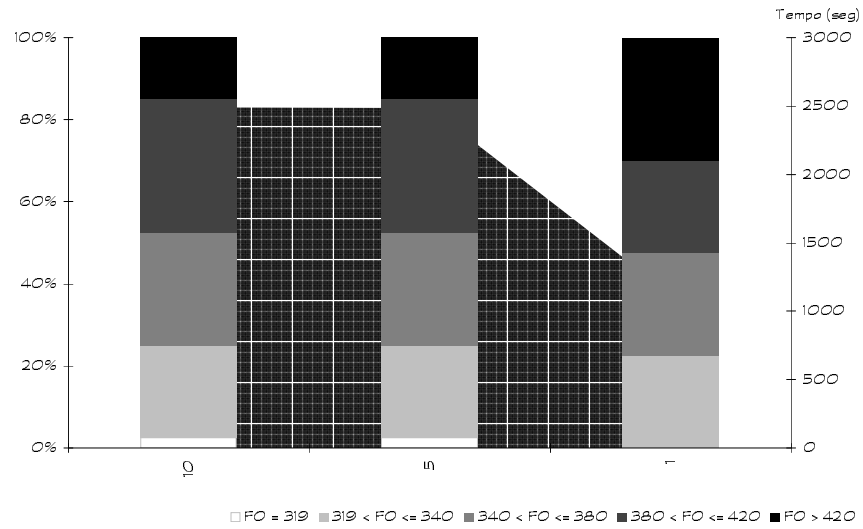
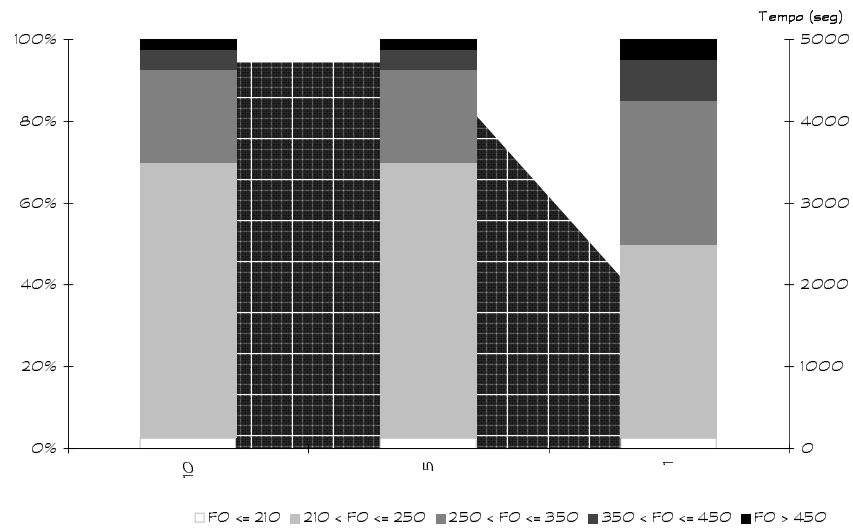
Figura 7.16: PAP5 - SA, determinação de T_0

Determinação da condição de paragem

Depois de determinado T_0 , passou-se a determinar a condição de paragem, mantendo $\alpha = 0.99$, $L = 500$ e agora $T_0 = 50$. Fizeram-se testes com os seguintes valores para a condição de paragem J : 1, 5 e 10 ($J * L$ iterações sem alteração da função objectivo). Os resultados dos testes realizados são apresentados nas Figuras 7.17 a 7.21 e os valores apresentados em abcissas correspondem aos valores de J .

Analisando os resultados apresentados nas Figuras 7.17 a 7.21, verifica-se que as soluções obtidas melhoram um pouco com o aumento de J . No entanto, essa melhoria é acompanhada por um grande aumento nos tempos para obtenção da solução e optou-se por usar $J = 1$ (500 soluções sem alteração do valor da função objectivo).

Figura 7.17: PAP1 - SA, determinação de J Figura 7.18: PAP2 - SA, determinação de J

Figura 7.19: PAP3 - SA, determinação de J Figura 7.20: PAP4 - SA, determinação de J

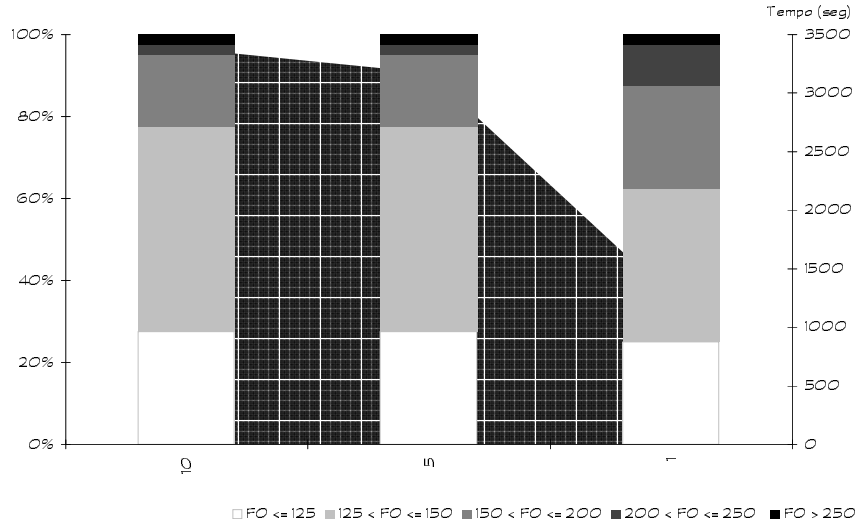
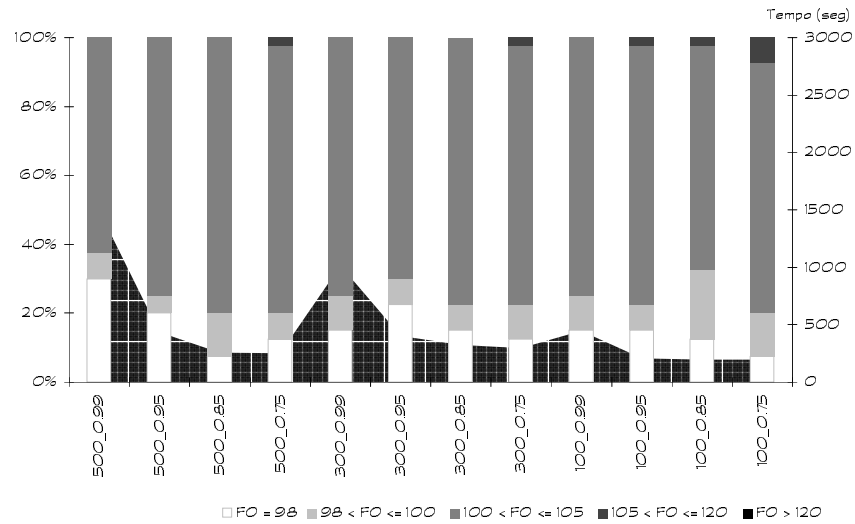
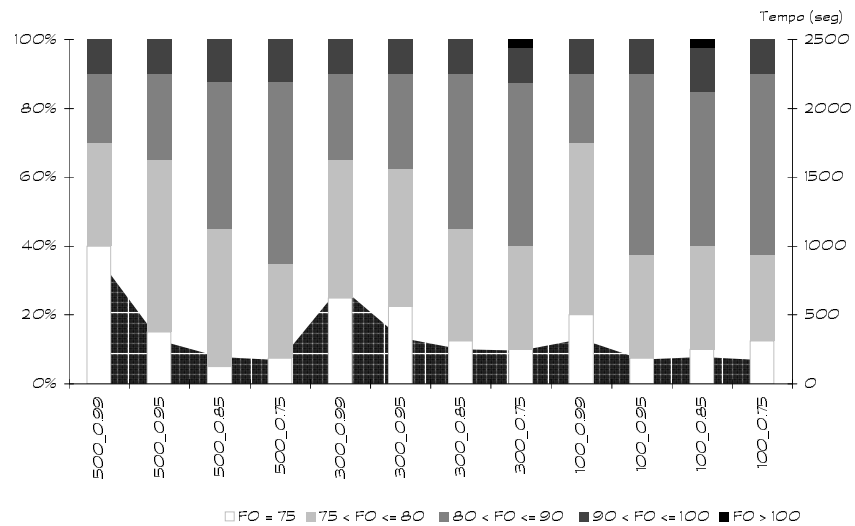


Figura 7.21: PAP5 - SA, determinação de J

Determinação de α e de L

Depois de determinada a condição de paragem, passou-se a determinar o par α, L . Para tal fizeram-se testes com $L = 500, 300$ e 100 e com $\alpha = 0.75, 0.85, 0.95$ e 0.99 . Os resultados dos testes realizados são apresentados nas Figuras 7.22 a 7.26, onde os valores apresentados em abcissas correspondem aos valores de L_{α} .

Como se pode ver nas Figuras 7.22 a 7.26, para cada valor de L , uma redução de α implica simultaneamente uma redução na qualidade das soluções obtidas e nos tempos de processamento. Do mesmo modo, para cada valor de α , uma redução de L implica simultaneamente uma redução na qualidade das soluções obtidas e nos tempos de processamento. Esse “padrão” repete-se para todas as instâncias. Face aos resultados obtidos e para obter resultados com tempos de processamento da mesma ordem de grandeza dos obtidos com a Pesquisa Tabu, optou-se por usar $L = 100$ e $\alpha = 0.99$.

Figura 7.22: PAP1 - SA, determinação de L e de α Figura 7.23: PAP2 - SA, determinação de L e de α

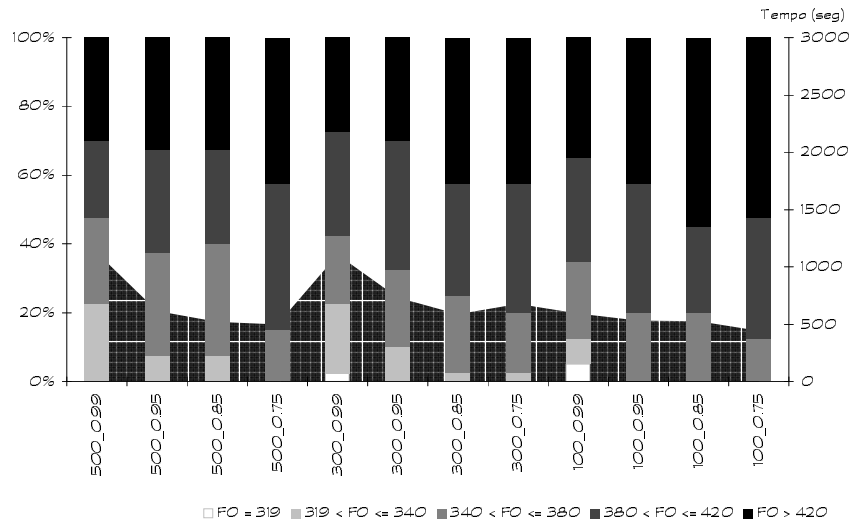


Figura 7.24: PAP3 - SA, determinação de L e de α

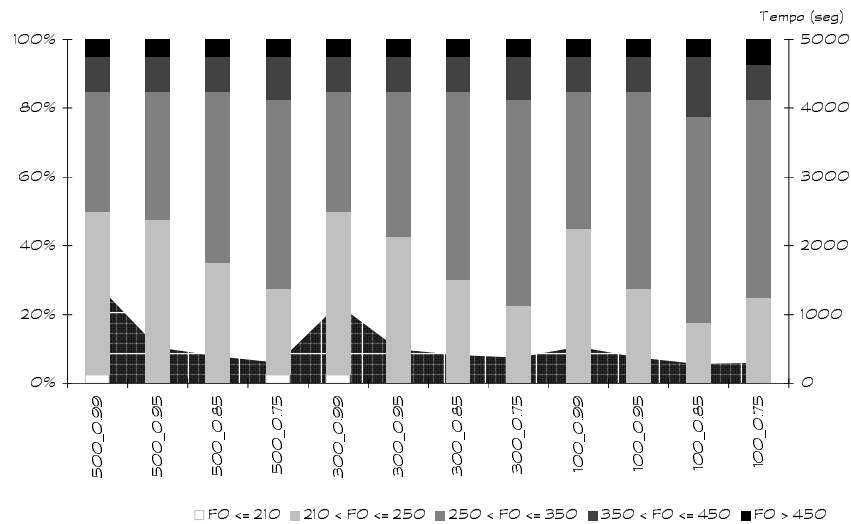


Figura 7.25: PAP4 - SA, determinação de L e de α

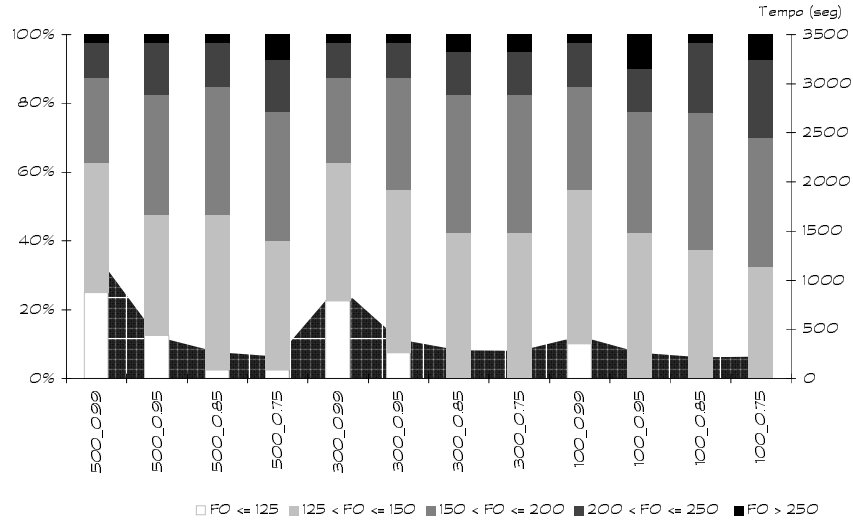


Figura 7.26: PAP5 - SA, determinação de L e de α

7.8.3 Resumo dos resultados obtidos

Na Tabela 7.1, encontram-se as opções para cada um dos parâmetros, resultantes dos testes apresentados. Nessa tabela, um ‘?’ significa que esse parâmetro ainda foi deixado em aberto, um ‘-’ significa que esse parâmetro não é aplicável à pesquisa em causa. A condição de paragem de 500 para “Simulated Annealing” significa que o algoritmo pára depois de 500 iterações sem alteração do valor da função objectivo.

	Pesquisa Tabu	“Simulated Annealing”
<i>TipSolIni</i>	3	?
<i>TipViz</i>	$\mathcal{N}_{Mista(8,0.2)}$	$\mathcal{N}_{Mista(8,0.2)}$
<i>TamSubViz</i>	50	?
<i>TipListaTabu</i>	?	-
<i>TamLista</i>	?	-
T_0	-	50
Condição de paragem	-	500
L	-	100
α	-	0.99

Tabela 7.1: Resumo dos resultados obtidos para parâmetros meta-heurísticas

7.9 Desempenho relativo das meta-heurísticas

Após a determinação dos valores adequados para alguns dos parâmetros das meta-heurísticas, mantiveram-se em aberto outros parâmetros, assinalados com ‘?’ na Tabela 7.1, que serão usados nos testes para comparação do desempenho relativo das meta-heurísticas. Os testes foram realizados com todas as instâncias (PAP1 a PAP5), com 40 soluções iniciais aleatórias e com $TipViz = \mathcal{N}_{Mista(8,0.2)}$. Nas Tabelas 7.2 e 7.3, apresentam-se os valores dos parâmetros utilizados nos testes. Nas Figuras 7.27 a 7.31, estão representados os resultados obtidos para as várias instâncias teste (PAP1 a PAP5).

	TipListaTabu	TamListaTabu	NumIter	TipSolIni	TamSubViz
TS-0	-	0	400	3	50
TS11	1	1	400	3	50
TS13	1	3	400	3	50
TS15	1	5	400	3	50
TS17	1	7	400	3	50
TS21	2	1	400	3	50
TS23	2	3	400	3	50
TS25	2	5	400	3	50
TS27	2	7	400	3	50
TS31	3	1	400	3	50
TS33	3	3	400	3	50
TS35	3	5	400	3	50
TS37	3	7	400	3	50
TS41	4	1	400	3	50
TS43	4	3	400	3	50
TS45	4	5	400	3	50
TS47	4	7	400	3	50

Tabela 7.2: Testes realizados com Pesquisa Tabu

	T_0	α	L	TipSolIni	TamSubViz
SA11_100	50	0.99	100	1	1
SA13_100	50	0.99	100	1	3
SA15_100	50	0.99	100	1	5
SA31_100	50	0.99	100	3	1
SA33_100	50	0.99	100	3	3
SA35_100	50	0.99	100	3	5

Tabela 7.3: Testes realizados com “Simulated Annealing”

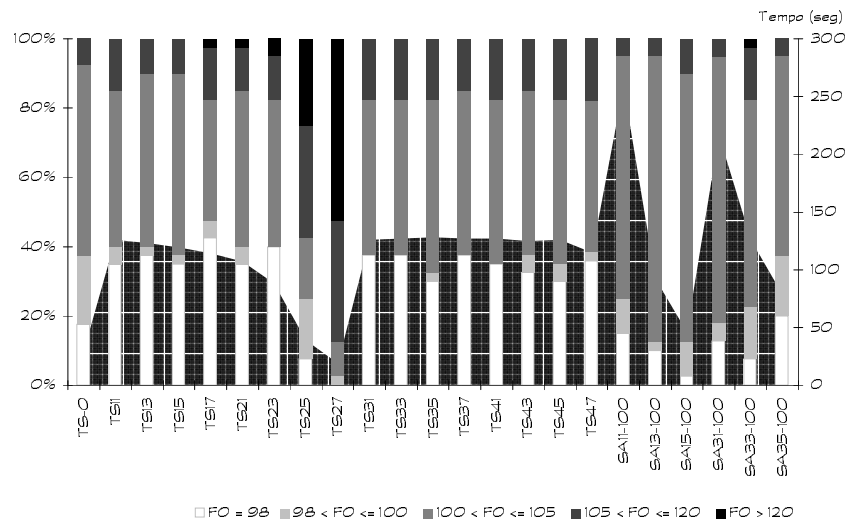


Figura 7.27: PAP1: Comparação de desempenho entre algoritmos

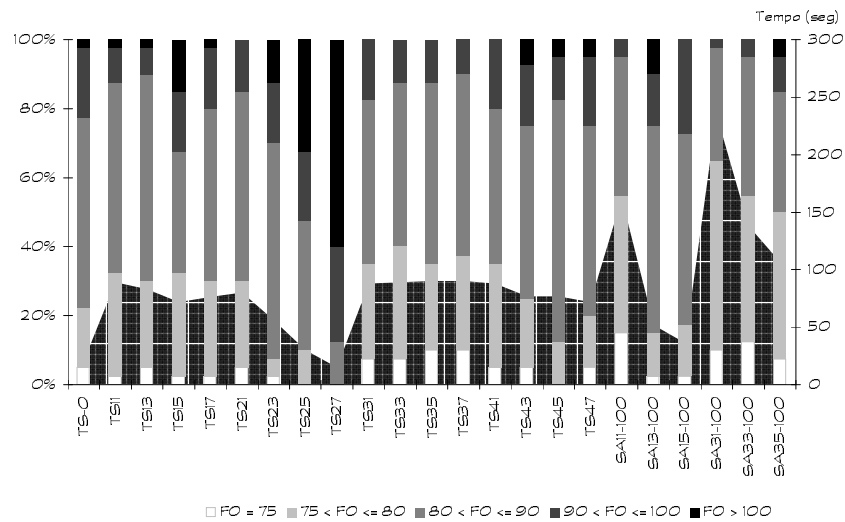


Figura 7.28: PAP2: Comparação de desempenho entre algoritmos

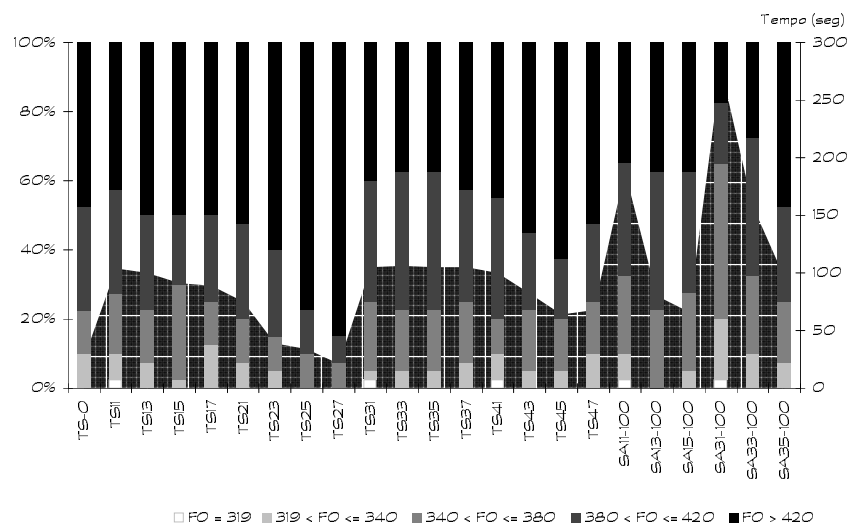


Figura 7.29: PAP3: Comparação de desempenho entre algoritmos

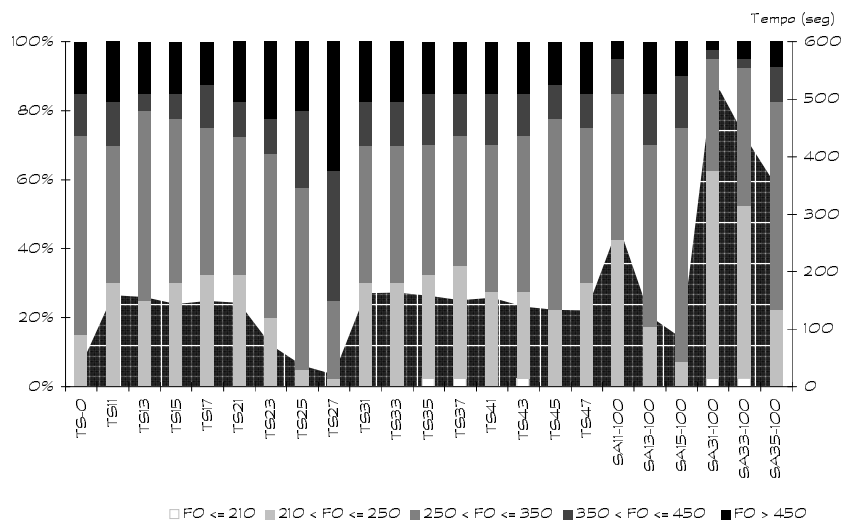


Figura 7.30: PAP4: Comparação de desempenho entre algoritmos

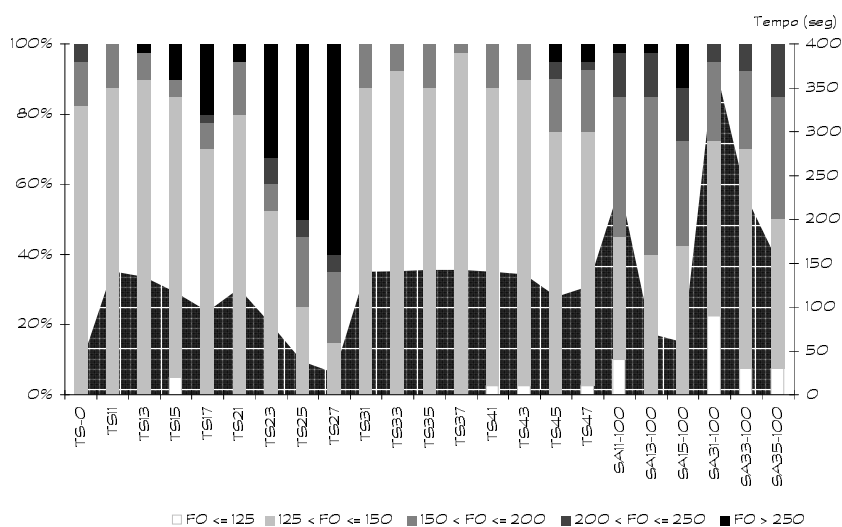


Figura 7.31: PAP5: Comparação de desempenho entre algoritmos

7.9.1 Conclusões

As conclusões que se podem extrair destes resultados são, relativamente à Pesquisa Tabu, que as melhores soluções são obtidas com a *lista tabu* 3 (lista com pares encomenda/período), independentemente do tamanho da lista tabu. No entanto também se pode notar que a existência de uma lista tabu tem, por si só, um efeito positivo. É de salientar que apenas na *lista tabu* 2 (lista com períodos) é que se nota uma variação dos resultados com o tamanho da lista (piores resultados com lista maior).

Relativamente ao “Simulated Annealing”, obtêm-se melhores resultados com $TipSolIni = 3$ (heurística aleatória, com melhoria de solução) e para $TamSubViz = 1$.

Comparando os resultados da Pesquisa Tabu com os resultados do “Simulated Annealing”, os melhores resultados obtidos com ambas as meta-heurísticas, têm uma qualidade semelhante; no entanto os tempos necessários para obter as soluções são maiores no “Simulated Annealing”.

7.9.2 Testes adicionais

Por último, e com o objectivo de explorar, no seu limite, as possibilidades do “Simulated Annealing”, optou-se por fazer um novo conjunto de testes, desta vez com $L = 500$ (número de iterações à mesma temperatura. Os resultados são apresentados novamente em simultâneo com os resultados já apresentados para a Pesquisa Tabu, nas Figuras 7.32 a 7.36, para que se possa avaliar esses resultados, em termos das componentes qualidade das soluções/tempos de resolução. As soluções obtidas pelo “Simulated Annealing” com $L = 500$, são melhores do que as obtidas com $L = 100$. No entanto, como seria de esperar, os tempos para obtenção das soluções são muito mais elevados.

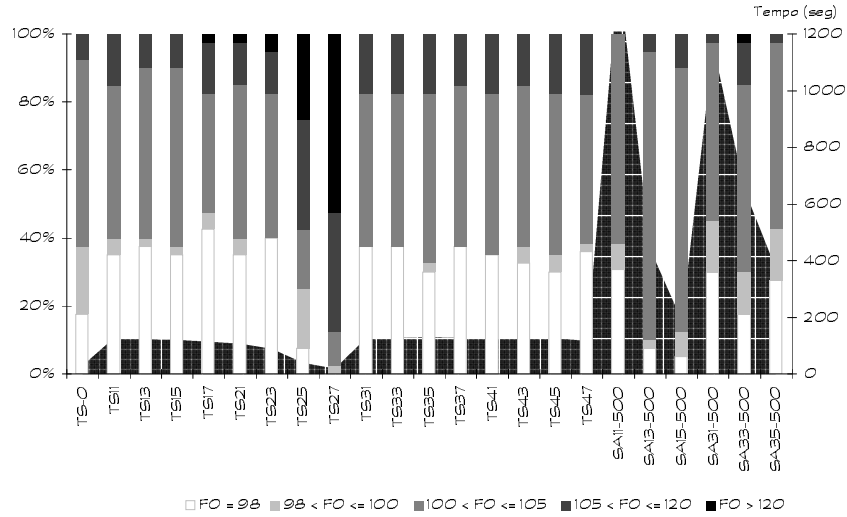


Figura 7.32: PAP1: Comparação de desempenho entre algoritmos

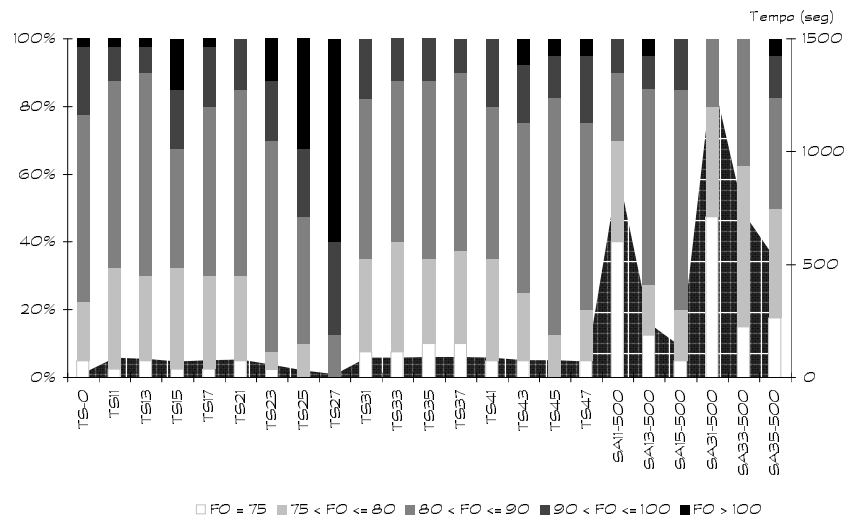


Figura 7.33: PAP2: Comparação de desempenho entre algoritmos

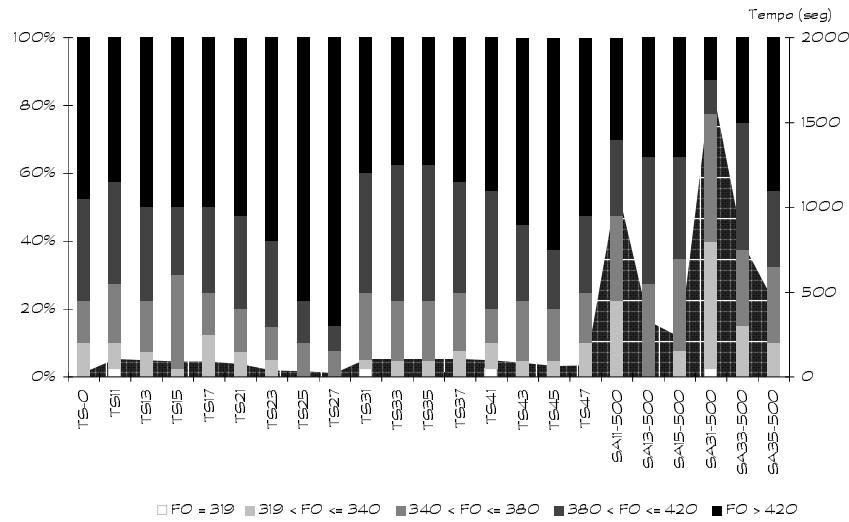


Figura 7.34: PAP3: Comparação de desempenho entre algoritmos

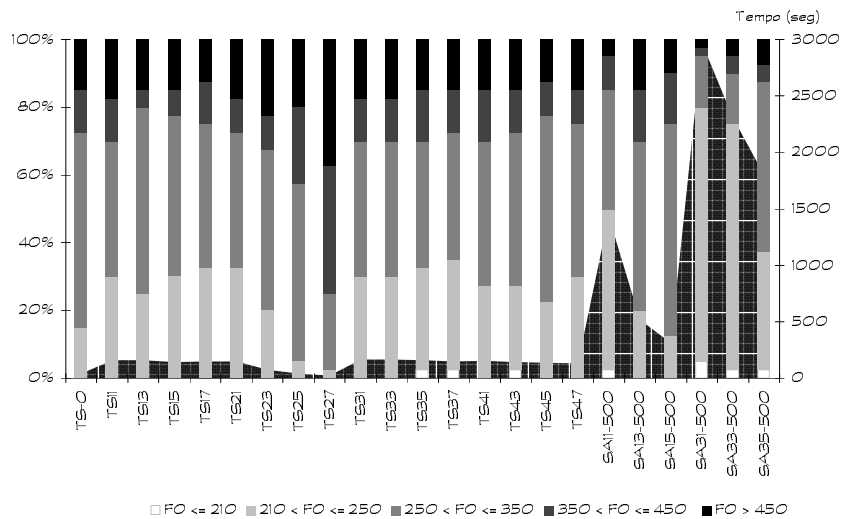


Figura 7.35: PAP4: Comparação de desempenho entre algoritmos

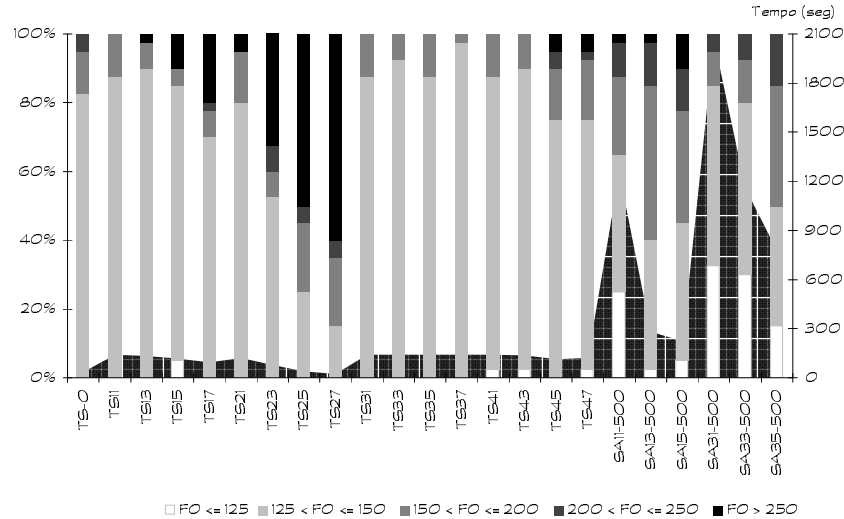


Figura 7.36: PAP5: Comparação de desempenho entre algoritmos

7.10 Conclusões

Os resultados obtidos com o recurso a meta-heurísticas, podem ser considerados promissores, dado que se obtêm soluções admissíveis e em geral bastante boas (ver Tabela 7.4), sem se recorrer a nenhum software específico de optimização. Esses resultados são úteis por si só, enquanto representam soluções de boa qualidade para o problema em estudo, mas também pelo facto de poderem ser usados como limites superiores (“upper bounds”) em algoritmos de optimização, com vista a torná-los mais eficientes.

	PAP1	PAP2	PAP3	PAP4	PAP5
valor IP	98	75	319	205	118
mínimo obtido com MH	98	75	319	207	118

Tabela 7.4: Comparação entre as melhores soluções obtidas com meta-heurísticas e as soluções óptimas

Julga-se, finalmente, que o conhecimento adquirido com este problema simplificado poderá permitir desenvolver heurísticas melhores para proble-

mas mais complicados e mesmo para o problema que constituiu a motivação para este estudo (PDP).

Capítulo 8

Conclusões e Desenvolvimentos Futuros

Nesta dissertação foi descrito um trabalho de investigação que assentou em duas linhas fundamentais. Uma que consistiu no projecto e implementação de um Sistema de Apoio à Decisão (SAD) para uma empresa da indústria do calçado, e uma outra que consistiu no estudo e avaliação das potencialidades de algumas das técnicas mais promissoras da Investigação Operacional actual, tais como as abordagens de Programação Inteira e Desigualdades Válidas por um lado, e as Meta-heurísticas por outro.

Caso de Estudo

Esta dissertação teve origem na implementação de um SAD a um Caso de Estudo, do qual se descreveu o processo de fabrico e o processo de tomada de decisões. Foi feito um enquadramento hierárquico do processo de tomada de decisões e uma interpretação hierárquica do SAD, que constituíram contribuições originais da autora para este trabalho. Esta perspectiva foi tida em conta durante o projecto e implementação do SAD.

Níveis de planeamento

O SAD inclui dois níveis de planeamento, um nível mais elevado a que se chamou PDP (Plano Director de Produção), com o qual se pretende apoiar a negociação com os clientes de datas de entrega de encomendas, e fazer

uma primeira alocação das encomendas às linhas de produção, e um nível mais baixo, a que se chamou PROG (Programação), com o qual se pretende fazer a alocação das ordens de fabrico às linhas de produção. Para cada um dos níveis foi desenvolvido um modelo de Programação Matemática e construídas heurísticas, posteriormente incorporadas no SAD (*H-PDP* e *H-PROG*).

Os resultados computacionais obtidos pela aplicação destas heurísticas aos problemas do Caso de Estudo, mereceram globalmente a aprovação dos utilizadores, dado que por um lado as soluções são estruturalmente semelhantes às obtidas pelos métodos utilizados no planeamento tradicional e por outro lado são obtidas em apenas alguns segundos. No entanto, ainda é necessário acrescentar às duas heurísticas funcionalidades que permitam uma mais fácil intervenção dos utilizadores no planeamento.

Ao longo do processo de implementação do SAD, revelou-se interessante dar aos utilizadores maiores capacidades de controlo das soluções, quer por fixação de períodos e linhas para produção de uma determinada encomenda/ordem de fabrico, quer acrescentando alguns parâmetros com os quais os utilizadores poderiam “moldar” a solução obtida. A introdução dessas funcionalidades passa, por um lado, por alterações nas heurísticas mas, por outro lado, passará também por alterações no próprio sistema informático. É óbvio que, para prosseguir este trabalho, será indispensável uma colaboração estreita entre os utilizadores e a equipa informática responsável pelo SAD.

Quanto ao PROG, tornou-se evidente, ao longo do projecto, a necessidade de existir uma realimentação de informação, que permitisse ao planeamento realizado no PDP (em encomendas) tomar em conta restrições impostas ao nível do PROG no planeamento das ordens de fabrico. Pode ainda ser referida a necessidade de introduzir uma realimentação interna ao PROG, que permita alterar automaticamente dados de planeamento, tais como capacidades das linhas para cada tipo de produto, de modo a obter automaticamente uma solução que satisfaça as restrições de número de postos e de número de máquinas de cada tipo.

Por último, a médio prazo, seria interessante desenvolver uma nova versão do SAD, flexibilizando-o ao ponto de possibilitar a realização de

planeamentos, mesmo no caso de se darem alterações profundas na organização da produção, tais como passar da produção em linha para produção em célula de fabrico.

Multicritério

A análise do problema PDP do ponto de vista multicritério, levou ao desenvolvimento de uma heurística multiobjectivo original, que consiste essencialmente num processo iterativo sobre duas heurísticas mais elementares (onde cada uma planeia apenas de acordo com um objectivo), cujo desempenho foi comparado com o de métodos analíticos de geração, para uma pequena instância de teste.

Os resultados obtidos indiciam que a integração de uma heurística multiobjectivo num sistema de apoio à decisão pode ser de grande utilidade. No entanto, seria importante construir heurísticas elementares que permitissem melhorar mais objectivos separadamente. Uma abordagem com meta-heurísticas poderia levar à obtenção de resultados interessantes, dado que se poderia usar a mesma estrutura de vizinhança em todos os casos e alterar apenas a função de avaliação, adaptando-a a cada um dos objectivos.

Planeamento Agregado da Produção

Dada a complexidade do problema PDP original, foi considerado interessante, após uma análise detalhada do modelo desenvolvido, explorar simplificações desse modelo, por forma a desenvolver métodos de resolução eficientes. Desses problemas simplificados, foi escolhido um para estudo (PAP), por se considerar relativamente simples e geral, mantendo no entanto as características básicas de um problema real de planeamento.

Modelo de Programação Inteira e Desigualdades Válidas

Foi seguidamente tentada uma primeira abordagem à resolução do problema PAP, que se baseia na tentativa de obtenção de limites inferiores para o valor da função objectivo, através da relaxação linear de um modelo de Programação Inteira com a adição de desigualdades válidas que tiram partido da estrutura particular do problema. Essas desigualdades foram utilizadas num algoritmo de planos de corte construído em torno de uma heurística de separação relativamente simples, que aproveita algumas das características

observadas nas soluções da relaxação linear, gerando desigualdades violadas nessas soluções. Com este procedimento, conseguiu-se reduzir consideravelmente o “gap” entre a solução óptima do problema e a solução da relaxação linear. Os resultados computacionais, embora preliminares, indiciam que este tipo de abordagem poderá ser bastante útil na obtenção de limites inferiores em problemas de dimensão real, e por consequência, na sua resolução eficiente.

O trabalho de investigação pode ser prosseguido na interpretação das desigualdades obtidas, enquanto casos particulares de outras mais gerais (associadas aos problemas de “knapsack” ou extensões), bem como no estudo das suas propriedades poliédricas, já que os resultados obtidos sugerem tratar-se de desigualdades “fortes”. Poderá ainda ser interessante gerar desigualdades válidas de outros tipos, obtidas por exemplo a partir de propriedades do modelo PAP identificadas neste trabalho.

Meta-heurísticas

No sentido de avaliar das possibilidades das meta-heurísticas na resolução deste tipo de problema, foram desenvolvidos e testados no problema PAP, algoritmos baseados na aleatorização de procedimentos de Pesquisa Local (Pesquisa Local Aleatorizada, Pesquisa Tabu, “Simulated Annealing”) que permitiram a obtenção eficiente de soluções satisfatórias, ou seja de majorantes (“upper bounds”) sobre o valor da solução óptima. A comparação das diferentes abordagens e dos algoritmos resultantes, foi feita com instâncias do problema geradas aleatoriamente. Relativamente à obtenção de limites superiores para o valor da função objectivo, através do recurso a meta-heurísticas, os resultados podem ser considerados promissores, dado que se obtêm soluções admissíveis e, em geral, de qualidade sem recorrer a técnicas pesadas de Programação Matemática. Além disso, os resultados obtidos podem também ser usados como limites superiores (“upper bounds”) em métodos de optimização, com vista a torná-los mais eficientes.

Note-se finalmente que o conhecimento adquirido com as diversas abordagens para resolução deste problema simplificado deverá permitir o desenvolvimento de heurísticas eficientes para classes mais gerais de problemas, ou para outros de acrescida complexidade mas de maior interesse prático.

Comentários finais

Refira-se finalmente que a organização deste trabalho de investigação assentou em duas linhas fundamentais. Por um lado, houve a preocupação de estudar um problema prático complicado, estruturando o processo de decisão numa perspectiva hierárquica e de influenciar o projecto de um SAD. Essa vertente do trabalho teve resultados francamente positivos, dada a grande receptividade ao produto final (o SAD) demonstrada pelos utilizadores.

Por outro lado, procurou-se, em torno de problemas simplificados “inspirados” nessa situação, aprofundar o conhecimento e testar novas potencialidades em técnicas representativas das orientações actuais da Investigação Operacional, tais como a Tomada de Decisões Multicritério, os modelos de Programação Inteira e Desigualdades Válidas, e as Meta-heurísticas. Em todas estas abordagens, tentou-se, com sucesso, chegar a ideias e princípios de abordagem dos problemas de aplicação bastante mais geral.

Referências

- [Ari and Axsäter, 1988] Ari, E. A. and Axsäter, S. (1988). Disaggregation under uncertainty in hierarchical production planning. *European Journal of Operational Research*, 35:182–186.
- [Axsäter, 1981] Axsäter, S. (1981). Aggregation of product data for hierarchical production planning. *Operations Research*, 29(4):744–756.
- [Axsäter and Jönsson, 1984] Axsäter, S. and Jönsson, H. (1984). Aggregation and disaggregation in hierarchical production planning. *European Journal of Operational Research*, 17:338–350.
- [Baker, 1974] Baker, K. R. (1974). *Introduction to Sequencing and Scheduling*. Wiley, New York.
- [Bitran et al., 1981] Bitran, G. R., Haas, E. A., and Hax, A. C. (1981). Hierarchical production planning: A single stage system. *Operations Research*, 29(4):717–743.
- [Bitran et al., 1982] Bitran, G. R., Haas, E. A., and Hax, A. C. (1982). Hierarchical production planning: A two-stage system. *Operations Research*, 30(2):233–251.
- [Bitran and Hax, 1977] Bitran, G. R. and Hax, A. C. (1977). On the design of hierarchical production planning systems. *Decision Sciences*, 8:28–55.
- [Boskma, 1982] Boskma, K. (1982). Aggregation and the design of models for medium-term planning of production. *European Journal of Operational Research*, 10:244–249.
- [Bukh, 1994] Bukh, N. D. (1994). A bibliography of hierarchical production

- planning. Technical report, Institute of Management - University of Aarhus.
- [Carravilla and Sousa, 1995] Carravilla, M. A. and Sousa, J. P. d. (1995). Hierarchical production planning in a make-to-order company - a case study. *European Journal of Operational Research*, 86(1):43–56.
- [Chankong and Haimes, 1983] Chankong, V. and Haimes, Y. (1983). *Multi-objective Decision Making. Theory and Methodology*. North Holland, New York.
- [Dauzère-Pérès and Lasserre, 1994] Dauzère-Pérès, S. and Lasserre, J. B. (1994). *An Integrated Approach in Production Planning and Scheduling*. Springer-Verlag, Heidelberg.
- [Eom and Lee, 1990] Eom, H. B. and Lee, S. M. (1990). Decision support systems applications research: A bibliography (1971-1988). *European Journal of Operational Research*, 46:333–342.
- [Erschler et al., 1986] Erschler, J., Fontan, G., and Merce, C. (1986). Consistency of the disaggregation process in hierarchical planning. *Operations Research*, 34(3):464–468.
- [Gelders and Van Steeland, 1980] Gelders, L. F. and Van Steeland, F. V. (1980). Design and implementation of a production planning system in a rolling mill: A case study. *AIIE Transactions*, 12(1):54–58.
- [Gelders and Van Wassenhove, 1982] Gelders, L. F. and Van Wassenhove, L. N. (1982). Hierarchical integration in production planning: Theory and practice. *Journal of Operations Management*, 3(1):27–35.
- [Gonçalves, 1994] Gonçalves, A. M. d. C. (1994). Um sistema de apoio à decisão para o planeamento da produção na indústria do calçado. Master's thesis, Faculdade de Engenharia da Universidade do Porto.
- [Graves, 1982] Graves, S. C. (1982). Using lagrangean techniques to solve hierarchical production planning problems. *Management Science*, 28(3):260–275.
- [Günther, 1986] Günther, H. O. (1986). The design of an hierarchical model for production planning and scheduling. In *Lecture Notes in Economics and Mathematical Systems - MultiStage Production Planning*

- and Inventory Control*, volume 266, pages 227–260. Springer Verlag, Berlin, Heidelberg, New York.
- [Hammer, 1993] Hammer, P. L., editor (1993). *Annals of Operations Research - Tabu Search*. J. C. Baltzer AG science publishers Basel - Switzerland.
- [Hax and Meal, 1975] Hax, A. C. and Meal, H. C. (1975). *Hierarchical integration of production planning and scheduling*, volume 1, Logistics, pages 53–69. Elsevier, Amsterdam.
- [Hendry and Kingsman, 1989] Hendry, L. C. and Kingsman, B. G. (1989). Production planning systems and their applicability to make-to-order companies. *European Journal of Operational Research*, 40:1–15.
- [Keen and Morton, 1978] Keen, P. G. W. and Morton, M. S. S. (1978). *Decision Support Systems - an organizational perspective*. Addison-Wesley series on decision support. Addison-Wesley Publishing Company.
- [Kroon, 1990] Kroon, L. G. (1990). *Job Scheduling and Capacity Planning in Aircraft Maintenance*. PhD thesis, Business School of the Erasmus University in Rotterdam.
- [Kuik et al., 1995] Kuik, R., Salomon, M., and Van Wassenhove, L. N. (1995). Batching decisions: structure and models. *European Journal of Operational Research*, 75:243–263.
- [Lawler, 1982] Lawler, E. L. (1982). *Recent Results in the Theory of Machine Scheduling*, chapter II. Springer Verlag, Berlin, Bonn.
- [Lin and Moodie, 1989] Lin, C.-W. and Moodie, C. L. (1989). Hierarchical production planning for a modern steel manufacturing system. *International Journal of Production Research*, 27(4):613–628.
- [Matos, 1988] Matos, M. A. C. d. C. (1988). *Ajuda á Decisão Multicritério - novas contribuições*. PhD thesis, Faculdade de Engenharia da Universidade do Porto.
- [Meal et al., 1987] Meal, H. C., Wachter, M. H., and Whybark, D. C. (1987). Material requirements planning in hierarchical production planning

- systems. *Internacional Journal of Production Research*, 25(7):947–956.
- [Nemhauser and Wolsey, 1988] Nemhauser, G. L. and Wolsey, L. A. (1988). *Integer and Combinatorial Optimization*. John Wiley & Sons, New York.
- [Oliveira, 1995] Oliveira, J. F. (1995). *Problemas de Posicionamento de Figuras Irregulares - Uma perspectiva de optimização*. PhD thesis, Faculdade de Engenharia da Universidade do Porto.
- [Otten and van Ginneken, 1989] Otten, R. H. J. M. and van Ginneken, L. P. P. P. (1989). *The annealing algorithm*. SECS 72. Kluwer Academic Publishers.
- [Pirlot, 1994] Pirlot, M. (1994). General local search heuristics in combinatorial optimization. *Belgian Journal of Operations Research, Statistics and Computer Science*, 32(1-2):7–67.
- [Rinnooy Kan, 1976] Rinnooy Kan, A. H. G. (1976). *Machine Scheduling Problems : Classification, Complexity and Computations*. Nijhoff, The Hague.
- [Schneeweiß, 1994] Schneeweiß, C. A. (1994). Elemente einer Theorie hierarchischer Planung. *Operations Research Spektrum*, 16:161–168.
- [Schneeweiß, 1995] Schneeweiß, C. A. (1995). Hierarchical structures in organisations - a conceptual framework. *European Journal of Operational Research*, 86:5–31.
- [Schrijver, 1986] Schrijver, A. (1986). *Theory of Linear and Integer Programming*. Wiley, New York.
- [Sousa, 1989] Sousa, J. P. d. (1989). *Time Indexed Formulations Of Non-Preemptive Single-Machine Scheduling Problems*. PhD thesis, Université Catholique du Louvain.
- [Sousa and Wolsey, 1992] Sousa, J. P. d. and Wolsey, L. A. (1992). A time indexed formulation of non-preemptive single machine scheduling problems. *Mathematical Programming*, 54:353–367.
- [Tobin et al., 1988] Tobin, N. R., Mercer, A., and Kingsman, B. G. (1988).

- A study of small subcontract and make-to-order firms in relation to quotation of orders. *International Journal of Operations and Production Management*, 8(6):46–59.
- [Turban, 1988] Turban, E. (1988). *Decision Support and Expert Systems - managerial perspectives*. Macmillan Publishing Company.
- [Ulungu and Teghem, 1994] Ulungu, E. L. and Teghem, J. (1994). Multi-objective combinatorial optimization problems: A survey. *Journal of Multi-Criteria Decision Analysis*, 3.
- [Van Laarhoven and Aarts, 1987] Van Laarhoven, P. J. M. and Aarts, E. H. L. (1987). *Simulated Annealing: Theory and Applications*. Kluwer Academic Publishers.
- [Van Wassenhove and Vanderhenst, 1983] Van Wassenhove, L. N. and Vanderhenst, P. (1983). Planning production in a bottleneck department. *European Journal of Operational Research*, 12:127–137.
- [Vidal, 1993] Vidal, R. V. V., editor (1993). *Applied Simulated Annealing*. Number 396 in Lecture Notes in Economics and Mathematical Systems. Springer-Verlag.

Apêndice A

Notação

A.1 Índices

e	encomenda (Ordem de Fabrico (OF) na Programação) ($e = 1, \dots, E$);
j	tipo de produto ($j = 1, \dots, J$);
t	período ($t = 1, \dots, T$);
l	linha de produção ($l = 1, \dots, L$);
f	fábrica ($f = 1, \dots, F$);
m	máquina ($m = 1, \dots, M$).

A.2 Parâmetros

q_e	quantidade encomendada na encomenda (OF) e (em unidades);
τ_e	tipo de produto da encomenda (OF) e ;
r_e	menor data possível para início de produção da encomenda (OF) e (“release date”);
d_e	data de entrega proposta para a encomenda (OF) e (“due date”);
I_e	$\{r_e, r_e+1, \dots, d_e\}$ janela temporal para produção da encomenda (OF) e ;

b_t	número de minutos disponíveis no período t (capacidade genérica do período);
C_{jlt}	capacidade de produção do produto tipo j na linha l no período t (em unidades);
mmc_{lt}	menor múltiplo comum de C_{jlt} para todos os tipos de produtos (parâmetros introduzidos por razões numéricas, ver Apêndice E);
P_{jlt}	índice de preferência para produção do produto tipo j na linha l no período t (este índice diminui com a preferência e um índice nulo indica impossibilidade);
U_{jt}	máxima produção possível do produto tipo j em todas as linhas no período t (em unidades);
A_1	constante introduzida pelo utilizador - quanto maior o valor de A_1 , menor o número de linhas pelas quais se podem dividir as encomendas;
A_2	constante introduzida pelo utilizador - quanto maior o valor de A_2 , menor o número de períodos pelos quais se podem dividir as encomendas;
A_3	constante introduzida pelo utilizador - quanto maior o valor de A_3 , maior o número de períodos permitido entre o início da produção de uma encomenda e d_e ;
K_m	número de máquinas disponíveis do tipo m ;
D_l	número máximo de máquinas que podem ser colocadas na linha l em cada período (número máximo de postos de trabalho da linha l em cada período);
p_{jm}	tempo de processamento do produto tipo j nas máquinas tipo m (em minutos);
p_{1jm}	tempo de processamento na pré-costura, do produto tipo j nas máquinas tipo m (em minutos);

p_{2jm} tempo de processamento na costura, do produto tipo j nas máquinas tipo m (em minutos).

A.3 Variáveis de decisão

x_{elt} número de unidades da encomenda (OF) e a produzir na linha l no período t ;

$$\delta_{ef} = \begin{cases} 1 & \text{se encomenda } e \text{ é produzida na fábrica } f \\ 0 & \text{se não} \end{cases}$$

$$\gamma_{elt} = \begin{cases} 1 & \text{se } x_{elt} \geq lb \text{ (onde } lb \text{ é um limite inferior para o tamanho dos} \\ & \text{lotos, imposto por razões operacionais)} \\ 0 & \text{se } x_{elt} < lb \end{cases}$$

k_{mlt} número de máquinas do tipo m a colocar na linha l no período t ;

c_{mlt} capacidade a instalar das máquinas do tipo m a na linha l no período t (variável de decisão contínua).

A.4 Grandezas auxiliares

a_1 número máximo de linhas utilizadas na produção de e :

$$a_1 = \min \left\{ \text{int} \left\{ \frac{q_e}{A_1} \right\} + 1, 3 \right\}$$

a_2 número máximo de períodos entre o início e o fim da produção de e :

$$a_2 = \min \left\{ \text{int} \left\{ \frac{q_e}{A_2} \right\} + 1, d_e - r_e + 1 \right\}$$

a_3 número máximo de períodos entre o início da produção de e e a sua data de entrega d_e :

$$a_3 = \min \{a_2 + A_3, d_e - r_e + 1\}$$

$CargTot_{lt}$ carga total na linha l e período t (em percentagem):

$$CargTot_{lt} = \sum_{e:\tau_e=j} \sum_{l:C_{jlt}>0} \frac{x_{elt}}{C_{jlt}}$$

$CapDispLin_{jlt}$ capacidade disponível para cada tipo de produto j , em cada linha l e para cada período t (em unidades):

$$CapDispLin_{jlt} = \{1 - CargTot_{lt}\} * C_{jlt}$$

$CargTip_{jt}$ carga induzida pelos produtos do tipo j no período t (em unidades):

$$CargTip_{jt} = \sum_{e:\tau_e=j} \sum_l x_{elt}$$

$CapDisp_{jt}$ capacidade disponível para os produtos do tipo j no período t (em unidades):

$$CapDisp_{jt} = U_{jt} - CargTip_{jt}$$

$PrimPerDispJL_{jl}$ primeiro período disponível para o tipo j , na linha l (-1 se não há nenhum período disponível):

$$PrimPerDispJL_{jl} = \max \left\{ -1, \min_t : \min \{ CapDispLin_{jlt}, CapDisp_{jt} \} > 0 \right\}$$

$PrimPerDispJ_j$ primeiro período disponível para o tipo j em qualquer linha (-1 se não há nenhum período disponível):

$$PrimPerDispJ_j = \max \left\{ PrimPerDispJ_j, \min_l PrimPerDispJL_{jl} \right\}$$

$UltimPerDispJL_{jl}$ último período disponível na linha l para o tipo j (-1 se não há nenhum período disponível):

$$UltimPerDispJL_{jl} = \max \left\{ -1, \max_t : \min \{ CapDispLin_{jlt}, CapDisp_{jt} \} > 0 \right\}$$

$UltimPerDispJ_j$ último período disponível para o tipo j em qualquer linha (-1 se não há nenhum período disponível):

$$UltimPerDispJ_j = \min \left\{ UltimPerDispJ_j, \max_l UltimPerDispJL_{jl} \right\}$$

$CapMaq_{mlt}$ número de minutos de máquinas do tipo m necessários para produzir os produtos planeadas para a linha l no período t :

$$CapMaq_{mlt} = \sum_e \sum_{l=2}^L x_{elt} * p_{1\tau_e m}$$

$$CapMaq_{mlt} = \sum_e x_{elt} * p_{2\tau_e m}$$

$NMaq_{mlt}$ número de máquinas m necessárias para produzir os produtos planeadas para a linha l no período t :
 ($\lceil x \rceil$ representa o menor inteiro $\geq x$)

$$NMaq_{mlt} = \left\lceil \frac{CapMaq_{mlt}}{b_t} \right\rceil$$

$FolgMaq_{mt}$ número de máquinas do tipo m que ficarão livres no período t :

$$FolgMaq_{mt} = K_m - \sum_l NMaq_{mlt}$$

$FolgLin_{lt}$ número de postos de trabalho da linha l que ficarão livres no período t :

$$FolgLin_{lt} = D_l - \sum_m NMaq_{mlt}$$

Apêndice B

Funções usadas nas heurísticas PDP e PROG

Neste apêndice, é feita uma descrição de cada função utilizada nas heurísticas implementadas no PDP e no PROG (ver secções 3.3, 4 e 5.4). As funções são apresentadas por ordem alfabética e da descrição individual de cada função consta, sempre que necessário:

- designação da função;
- descrição sucinta;
- dados de entrada da função;
- resultados da função;
- algoritmo implementado;
- funções chamadas;
- observações.

No Apêndice A faz-se uma descrição pormenorizada de todas as variáveis mencionadas neste apêndice.

B.1 ActualizaCapDisp()

Actualiza o valor de $CapDisp_{jt}$ (capacidade disponível para o produto tipo j no período t), calculando, para cada período t e cada tipo de produto j , o mínimo entre a capacidade disponível em todas as linhas e a capacidade disponível no período para esse tipo de produto.

Dados

$CapDispLin_{jlt}$ capacidade disponível para cada tipo de produto j , em cada linha l e para cada período t (em unidades);

$CapDisp_{jt}$ capacidade disponível para o produto tipo j no período t (em unidades).

Resultados

$$\forall_{jt} \quad CapDisp_{jt} = \min \{CapDisp_{jt}, \sum_l CapDispLin_{jlt}\}$$

Observações

A actualização periódica de $CapDisp_{jt}$, permite que **CapProdTip(j, nper, t₁)**, retorne um valor mais real, reduzindo o número de vezes que é necessário pesquisar as linhas.

B.2 ActualizaCargas(q, j, l, t)

Actualiza os valores da capacidade total disponível para o tipo j , no período t ($CapDisp_{jt}$) e os valores da capacidade disponível para o tipo j na linha l e período t ($CapDispLin_{jlt}$), dada uma quantidade q do produto tipo j que foi alocada à linha l no período t .

Dados

C_{jlt} capacidade de produção do produto tipo j na linha l no período t (em unidades);

$CargTip_{jt}$ carga do tipo j no período t (em unidades);

$CargTot_{lt}$ carga total na linha l período t (em percentagem);

U_{jt} máxima produção possível do produto tipo j em todas as linhas no período t (em unidades);

$CapDisp_{jt}$ capacidade disponível para o produto tipo j no período t (em unidades).

Resultados

$$\begin{aligned}
 CargTot_{lt} &= CargTot_{lt} + \frac{quant}{C_{jlt}} \\
 \forall_j \quad CapDispLin_{jlt} &= \{1 - CargTot_{lt}\} * C_{jlt} \\
 CargTip_{jt} &= CargTip_{jt} + quant \\
 CapDisp_{jt} &= U_{jt} - CargTip_{jt}
 \end{aligned}$$

Funções chamadas

ActualizaCapDisp()
 ActualizaPrimPerDisp()
 ActualizaUltimPerDisp()

B.3 ActualizaPrimPerDisp()

Calcula o primeiro período em que há capacidade disponível para produzir um produto tipo j numa linha l , e o primeiro período com capacidade disponível para produzir o produto tipo j .

Dados

$CapDispLin_{jlt}$ capacidade disponível para cada tipo de produto j , em cada linha l e para cada período t (em unidades);

$CapDisp_{jt}$ capacidade disponível para o produto tipo j no período t (em unidades).

Resultados

$PrimPerDispJL_{jl}$ primeiro período disponível para o tipo j , na linha l (-1 se não há nenhum período disponível):

$$\forall_{jl} \quad PrimPerDispJL_{jl} = \max \{-1, \min_t : \min \{CapDispLin_{jlt}, CapDisp_{jt}\} > 0\}$$

$PrimPerDispJ_j$ primeiro período disponível para o tipo j em qualquer linha (-1 se não há nenhum período disponível);

$$\forall_j \quad PrimPerDispJ_j = \max \{PrimPerDispJ_j, \min_l PrimPerDispJL_{jl}\}$$

Observações

Mantendo $PrimPerDispJL_{jl}$ e $PrimPerDispJ_j$ actualizados, reduz-se consideravelmente o número de períodos a percorrer na pesquisa de capacidade disponível.

B.4 ActualizaUltimPerDisp()

Calcula o último período em que há capacidade disponível para produzir um produto tipo j numa linha l , e o primeiro período com capacidade disponível para produzir o produto tipo j .

Dados

$CapDispLin_{jlt}$ capacidade disponível para cada tipo de produto j , em cada linha l e para cada período t (em unidades);

$CapDisp_{jt}$ capacidade disponível para o produto tipo j no período t (em unidades).

Resultados

$UltimPerDispJL_{jl}$ último período disponível na linha l para o tipo j (-1 se não há nenhum período disponível):

$$\forall_{jl} \quad UltimPerDispJL_{jl} = \max \{-1, \max_t : \min \{CapDispLin_{jlt}, CapDisp_{jt}\} > 0\}$$

$UltimPerDispJ_j$ último período disponível para o tipo j em qualquer linha (-1 se não há nenhum período disponível):

$$\forall_j \quad UltimPerDispJ_j = \min \{UltimPerDispJ_j, \max_l UltimPerDispJL_{jl}\}$$

Observações

Mantendo $UltimPerDispJL_{jl}$ e de $UltimPerDispJ_j$ actualizados, reduz-se consideravelmente o número de períodos a percorrer na pesquisa de capacidade disponível.

B.5 CalculaFolgas()

Determina o número de máquinas de cada tipo e o número de postos em cada linha que ficam disponíveis em cada período.

Dados

$NMaq_{mlt}$ número de máquinas m necessárias na linha l no período t ;

D_l número máximo de máquinas que podem ser colocadas na linha l em cada período;

K_m número de máquinas disponíveis do tipo m .

Resultados

$FolgMaq_{mt}$ Número de máquinas do tipo m que ficarão livres no período t ;

$FolgLin_{lt}$ Número de postos da linha l que ficarão livres no período t ;

$$FolgMaq_{mt} = K_m - \sum_l NMaq_{mlt}$$

$$FolgLin_{lt} = D_l - \sum_m NMaq_{mlt}$$

B.6 CapProdLin(**j**, **nlin**, **nper**, **t₁**)

Retorna a capacidade existente nas linhas pertencentes a *Lista*, para produzir o tipo *j*, entre um período inicial t_1 e um período final $t_1 + nper - 1$.

Dados

<i>Lista</i>	vector com linhas a testar;
<i>CapDisp_{jt}</i>	capacidade disponível para o produto tipo <i>j</i> no período <i>t</i> (em unidades);
<i>CapDispLin_{jlt}</i>	capacidade disponível para cada tipo de produto <i>j</i> , em cada linha <i>l</i> e para cada período <i>t</i> (em unidades).

Resultados

$$\sum_{t=t_1}^{t_1+nper-1} \min(\sum_{l \in Lista} CapDispLin_{jlt}, CapDisp_{jt})$$

Observações

Se se pretender produzir *e* apenas em períodos consecutivos, pode-se fazer com que esta função retorne um valor nulo, sempre que a capacidade disponível seja 0 num período entre t_1 e $t_1 + nper - 1$.

B.7 CapProdTip($j, nper, t_1$)

Retorna a capacidade existente entre um período inicial t_1 e um período final $t_1 + nper - 1$, para produzir o tipo j sem ter em conta as linhas em que se pretende produzir.

Dados

$CapDisp_{jt}$ capacidade disponível para o produto tipo j no período t (em unidades).

Resultados

$$\sum_{t=t_1}^{t_1+nper-1} CapDisp_{jt}$$

Observações

Se se pretender produzir e apenas em períodos consecutivos, pode-se fazer com que esta função retorne um valor nulo, sempre que a capacidade disponível seja 0 num período entre t_1 e $t_1 + nper - 1$.

B.8 CarregaLinhas($e, nlin, nper, t_1, caso$)

Carrega as linhas incluídas em *Lista* com a encomenda e . Se $caso = 1$, carrega as linhas de t_1 até $t_1 + nper - 1$, se $caso = 2$, carrega as linhas de $t_1 + nper - 1$ até t_1 .

Dados

<i>Lista</i>	vector com linhas a testar;
$CapDisp_{jt}$	capacidade disponível para o produto tipo j no período t (em unidades);
$CapDispLin_{jlt}$	capacidade disponível para cada tipo de produto j , em cada linha l e para cada período t (em unidades).

Resultados

x_{elt}	quantidade da encomenda e a produzir na linha l no período t .
-----------	--

Funções chamadas

$$\forall_{x_{elt} \neq 0} \quad \text{ActualizaCargas}(\mathbf{x}_{elt}, \tau_e, \mathbf{l}, \mathbf{t})$$

B.9 ColocaMaisCedo(e)

Coloca e o mais cedo possível, no menor número de períodos possível e no menor número de linhas possível (tendo em conta os índices de preferência, as capacidades disponíveis e os limites determinados em **DeterminaLimites()**).

Retorna 1 se consegue colocar a encomenda dentro dos limites impostos, e retorna 0 se não consegue.

Dados

- a_1 número máximo de linhas utilizadas na produção de e ;
- a_2 número máximo de períodos entre o início e o fim da produção de e ;
- a_3 número máximo de períodos entre o início da produção de e e a sua data de entrega d_e ;
- $LinOrd_{jtn}$ linhas por ordem de preferência, para cada par tipo de produto/período.

Algoritmo

```

para ( $nlin = 1; nlin \leq a_1; nlin = nlin + 1$ )
  para ( $nper = 1; nper \leq a_2; nper = nper + 1$ )
    para ( $t_1 = d_e - nper + 1; t_1 \geq d_e - a_3 + 1; t_1 = t_1 - 1$ )
      se (CapProdTip( $\tau_e, nper, t_1$ )  $\geq q_e$ )
        GeraLista
        se (CapProdLin( $\tau_e, nlin, nper, t_1$ )  $\geq q_e$ )
          CarregaLinhas( $\tau_e, nlin, nper, t_1, 2$ )
          retorna 1
retorna 0

```

Algoritmo B.1: Algoritmo ColocaMaisCedo

Funções chamadas

CapProdTip($\tau_e, nper, t_1$)
CapProdLin($\tau_e, nlin, nper, t_1$)
CarregaLinhas($\tau_e, nlin, nper, t_1, 1$)

B.10 ColocaMaisTarde(e)

Coloca e o mais tarde possível, no menor número de períodos possível e no menor número de linhas possível (tendo em conta os índices de preferência, as capacidades disponíveis e os limites determinados em **DeterminaLimites()**).

Retorna 1 se consegue colocar a encomenda dentro dos limites impostos, e retorna 0 se não consegue.

Dados

- a_1 número máximo de linhas utilizadas na produção de e ;
- a_2 número máximo de períodos entre o início e o fim da produção de e ;
- a_3 número máximo de períodos entre o início da produção de e e a sua data de entrega d_e ;
- $LinOrd_{jtn}$ linhas por ordem de preferência, para cada par tipo de produto/período.

Algoritmo

```

para ( $nlin = 1; nlin \leq a_1; nlin = nlin + 1$ )
  para ( $nper = 1; nper \leq a_2; nper = nper + 1$ )
    para ( $t_1 = d_e - nper + 1; t_1 \geq d_e - a_3 + 1; t_1 = t_1 - 1$ )
      se (CapProdTip( $\tau_e, nper, t_1$ )  $\geq q_e$ )
        GeraLista
          se (CapProdLin( $\tau_e, nlin, nper, t_1$ )  $\geq q_e$ )
            CarregaLinhas( $\tau_e, nlin, nper, t_1, 2$ )
            retorna 1
retorna 0

```

Algoritmo B.2: Algoritmo ColocaMaisTarde

Funções chamadas

CapProdTip($\tau_e, nper, t_1$)
CapProdLin($\tau_e, nlin, nper, t_1$)
CarregaLinhas($\tau_e, nlin, nper, t_1, 2$)

B.11 ColocaPROG()

Carrega as linhas de produção com as ordens de fabrico, começando pela linha mais dedicada e acabando na linha exterior, tendo em conta os índices de preferência e as capacidades disponíveis.

Algoritmo

```

para ( $l = 1; l \leq L - 1; l = l + 1$ )
  GeraLista
  para ( $t = 1; t \leq T; t = t + 1$ )
    para ( $n = 1; n \leq J; n = n + 1$ )
      se ( $TipProdOrd_{ltn} = -1$ ) pára
      para ( $e = 1; e \leq E; e = e + 1$ )
        se ( $colocou_e = 0 \wedge d_e \leq t \wedge \tau_e = TipProdOrd_{ltn}$ )
          para ( $nper = 1; nper \leq \min(2, d_e - r_e + 1); nper = nper + 1$ )
            para ( $t_1 = d_e - nper + 1; t_1 \geq r_e; t_1 = t_1 - 1$ )
              se ( $CapProdLin(\tau_e, 1, nper, t_1) \geq q_e$ )
                CarregaLinhas( $\tau_e, 1, nper, t_1, 1$ )
                 $colocou_e = 1$ 
    para ( $e = 1; e \leq E; e = e + 1$ )
      se ( $colocou_e = 0$ )
        colocar  $e$  em  $l = L$  (linha exterior)

```

Algoritmo B.3: Algoritmo ColocaPROG

Funções chamadas

CapProdLin($\tau_e, 1, nper, t_1$)
CarregaLinhas($\tau_e, 1, nper, t_1, 2$)

B.12 DeterminaLimites(e)

A manipulação dos parâmetros A_1 , A_2 e A_3 , permite que o utilizador controle o tipo de solução que irá obter, limitando o “espalhamento” das encomendas pelas linhas e períodos e o número de períodos entre o início da produção e a data de entrega d_e . Esses limites são calculados nesta função.

Dados

- A_1 constante introduzida pelo utilizador - quanto maior o valor de A_1 , menor o número de linhas pelas quais se podem dividir as encomendas;
- A_2 constante introduzida pelo utilizador - quanto maior o valor de A_2 , menor o número de períodos pelos quais se podem dividir as encomendas;
- A_3 constante introduzida pelo utilizador - quanto maior o valor de A_3 , maior o número de períodos permitido entre o início da produção de uma encomenda e d_e .

Resultados

- a_1 número máximo de linhas utilizadas na produção de e ;
- a_2 número máximo de períodos entre o início e o fim da produção de e ;
- a_3 número máximo de períodos entre o início da produção de e e a sua data de entrega d_e .

$$a_1 = \min \left\{ \text{int} \left\{ \frac{q_e}{A_1} \right\} + 1, 3 \right\}$$

$$a_2 = \min \left\{ \text{int} \left\{ \frac{q_e}{A_2} \right\} + 1, d_e - r_e + 1 \right\}$$

$$a_3 = \min \{a_2 + A_3, d_e - r_e + 1\}$$

B.13 FazLayout()

Determina a carga de cada tipo de máquina (em minutos), para cada linha e em cada período.

Dados

x_{elt}	quantidade da encomenda e a produzir na linha l no período t ;
p_{1jm}	tempo de processamento na pré-costura, do produto tipo j nas máquinas tipo m (em minutos);
p_{2jm}	tempo de processamento na costura, do produto tipo j nas máquinas tipo m (em minutos).

Resultados

$CapMaq_{mlt}$	número de minutos da máquina m necessários na linha l no período t .
----------------	--

$$CapMaq_{m0t} = \sum_e \sum_l x_{elt} * p_{1\tau_e m}$$

$$CapMaq_{mlt} = \sum_e x_{elt} * p_{2\tau_e m}$$

B.14 Inicializa()

Inicializa as variáveis globais.

Dados

C_{jlt} capacidade de produção do produto tipo j na linha l no período t (em unidades).

Resultados

$$\begin{aligned}\forall_{elt} \quad x_{elt} &= 0 \\ \forall_{lt} \quad CargTot_{lt} &= 0 \\ \forall_{jt} \quad CargTip_{jt} &= 0 \\ \forall_{lt} \quad mmc_{lt} &= mmc_j[C_{jlt}]\end{aligned}$$

Funções chamadas

$$\begin{aligned}\forall_{jlt} \quad &\text{ActualizaCargas}(0, j, l, t) \\ &\text{OrdPrefJL}()\end{aligned}$$

B.15 OrdPrefJL()

Gera um vector com as linhas por ordem de preferência, para cada par tipo de produto/período.

Dados

P_{jlt} índice de preferência para produção do produto tipo j na linha l no período t .

Resultados

$LinOrd_{jtn}$ linhas por ordem de preferência, para cada par tipo de produto/período;

Partindo da matriz de índices de preferência representada na Tabela 3.2 obter-se-iam os valores apresentados na Tabela B.1, para $LinOrd_{jtn}$

$LinOrd_{jtn}$	$TipoA$	$TipoB$	$TipoC$
$n = 0$	$l = 2$	$l = 2$	$l = 4$
$n = 1$	$l = 3$	$l = 3$	$l = 1$
$n = 2$	$l = 4$	$l = 4$	$l = 2$
$n = 3$	-1	-1	$l = 3$

Tabela B.1: Matriz $LinOrd_{jtn}$

B.16 OrdPrefLJ()

Gera um vector com os tipos de produtos por ordem de preferência, para cada par linha/período.

Dados

P_{jlt} índice de preferência para produção do produto tipo j na linha l no período t .

Resultados

$TipProdOrd_{ltn}$ Tipos de produtos por ordem de preferência, para cada par linha/período.

Partindo da matriz de índices de preferência representada na Tabela 3.2 obter-se-iam os valores apresentados na Tabela B.2, para $TipProdOrd_{ltn}$

$TipProdOrd_{ltn}$	$Linha1$	$Linha2$	$Linha3$	$Linha4$
$n = 0$	$\tau = C$	$\tau = A$	$\tau = A$	$\tau = C$
$n = 1$	$\tau = -1$	$\tau = B$	$\tau = C$	$\tau = A$
$n = 2$	$\tau = -1$	$\tau = C$	$\tau = B$	$\tau = B$

Tabela B.2: Matriz $TipProdOrd_{ltn}$

B.17 RelaxaLimites(e)

Encontra um conjunto de períodos e um conjunto de linhas para produzir e , mesmo que para isso devam ser relaxados alguns dos limites calculados inicialmente (relaxando mesmo, se necessário, a data de entrega, e em último caso, tentando produzir uma quantidade menor que a quantidade encomendada).

Retorna 1 se consegue colocar a encomenda, e retorna 0 se não consegue.

Algoritmo

```

para ( $q_e = q_{eini}$ ;  $q_e \geq 0$ ;  $q_e = 0.9 * q_e$ ) (relaxar quantidade)
  para ( $d_e = d_{eini}$ ;  $d_e \leq T$ ;  $d_e = d_e + 1$ ) (relaxar data de entrega)
    para ( $a_2 = a_2$ ;  $a_2 \leq d_e - r_e + 1$ ;  $a_2 = a_2 + 1$ )
      (relaxar o número de períodos de produção)
        para ( $a_3 = a_3$ ;  $a_3 \leq d_e - r_e$ ;  $a_3 = a_3 + 1$ )
          para ( $a_1 = a_1$ ;  $a_1 \leq 3$ ;  $a_1 = a_1 + 1$ )
            (relaxar o número de linhas de produção)
              se ( $d_e = d_{eini}$ )
                se ColocaMaisTarde(e)
                  retorna 1
              se não
                se ColocaMaisCedo(e)
                  retorna 1
  retorna 0

```

Algoritmo B.4: Algoritmo RelaxaLimites

Funções chamadas

ColocaMaisTarde(e)

ColocaMaisCedo(e)

B.18 RoundLayout()

Determina o número de máquinas de cada tipo necessário em cada linha e em cada período.

Dados

$CapMa_{mlt}$ número de minutos da máquina m necessários na linha l no período t ;

Resultados

NMa_{mlt} número de máquinas m necessárias na linha l no período t ;

$$NMa_{mlt} = \lceil CapMa_{mlt} \rceil$$

Apêndice C

Exemplo PDP

Apresenta-se a seguir um exemplo de resolução de uma instância do problema PDP com a heurística *H-PDP*. Para uma descrição completa da notação utilizada neste apêndice, recomenda-se a consulta do Apêndice A. Para uma descrição completa das funções utilizadas neste apêndice, recomenda-se a consulta do Apêndice B.

C.1 Dados de entrada

T - número de períodos	3
E - número de encomendas	4
L - número de linhas	4
F - número de fábricas	2
J - número de tipos de produtos	2

Tabela C.1: Características gerais da instância

Linha	1	2	3	4
Fábrica	1	1	2	2

Tabela C.2: Fábrica a que pertence cada uma das linhas l

Considere-se ainda que $A_1 = 2000$, $A_2 = 4000$ e $A_3 = 1$.

Os restantes dados encontram-se nas Tabelas C.3 e C.4.

	Linha 1	Linha 2	Linha 3	Linha 4	
	P_{jlt}				
$j = 1$	0	1	1	4	
$j = 2$	1	0	1	1	
	C_{jlt}				U_{jt}
$j = 1$	0	5000	2500	5000	5000
$j = 2$	5000	0	5000	2500	2500

Tabela C.3: Índices de preferência, P_{jlt} e capacidades de produção, C_{jlt} , das linhas l por tipo de produto j , para todos os períodos t e máxima produção possível do produto tipo j em todas as linhas no período t , U_{jt}

e	τ_e	r_e	d_e	q_e
1	1	1	1	5000
2	1	1	2	10000
3	2	1	2	3000
4	2	2	3	3000

Tabela C.4: Lista de encomendas: tipo de produto (τ_e), “release date” (r_e), data de entrega (d_e) e quantidade (q_e)

C.2 Resolução

A função **Inicializa()** chama **ActualizaCargas(0,j,l,t)** e gera os seguintes valores:

$$\begin{aligned}
 \forall_{elt} \quad x_{elt} &= 0 \\
 \forall_{lt} \quad CargTot_{lt} &= 0 \\
 \forall_{jt} \quad CargTip_{jt} &= 0 \\
 \forall_{lt} \quad mmc_{lt} &= 5000
 \end{aligned}$$

e ainda os valores apresentados nas Tabelas C.5 e C.6.

Na Tabela C.5, $CapDispLin_{jlt}$ representa o capacidade disponível para cada tipo de produto j , em cada linha l e para cada período t (em unidades), e $CapDisp_{jt}$ representa a capacidade disponível para os produtos do tipo j no período t (em unidades)

Na Tabela C.6, $PrimPerDispJL_{jl}$ representa o primeiro período disponível para o tipo j , na linha l , $PrimPerDispJ_j$ representa o primeiro período disponível para o tipo j em qualquer linha, $UltimPerDispJL_{jl}$ re-

		<i>CapDispLin_{jlt}</i>				<i>CapDisp_{jt}</i>
	t	Linha 1	Linha 2	Linha 3	Linha 4	
$j = 1$	1	0	5000	2500	5000	5000
$j = 1$	2	0	5000	2500	5000	5000
$j = 1$	3	0	5000	2500	5000	5000
$j = 2$	1	5000	0	5000	2500	2500
$j = 2$	2	5000	0	5000	2500	2500
$j = 2$	3	5000	0	5000	2500	2500

Tabela C.5: Valores iniciais de *CapDispLin_{jlt}* e *CapDisp_{jt}*

presenta o último período disponível na linha l para o tipo j e *UltimPerDispJ_j* representa o último período disponível para o tipo j em qualquer linha (tomam o valor -1 se não há nenhum período disponível).

	Linha 1	Linha 2	Linha 3	Linha 4	
		<i>PrimPerDispJL_{jl}</i>			<i>PrimPerDispJ_j</i>
$j = 1$	-1	1	1	1	1
$j = 2$	1	-1	1	1	1
		<i>UltimPerDispJL_{jl}</i>			<i>UltimPerDispJ_j</i>
$j = 1$	-1	3	3	3	3
$j = 2$	3	-1	3	3	3

Tabela C.6: Primeiros e últimos períodos onde existe capacidade disponível

Seguidamente, a função **OrdPrefJL()**, utiliza os dados da Tabela C.3, para gerar a matriz que se encontra representada na Tabela C.7, que contém, para cada tipo de produto, as linhas de produção por ordem de preferência.

<i>LinOrd_{jtn}</i>	$j = 1$	$j = 2$
$n = 1$	$l = 2$	$l = 1$
$n = 2$	$l = 3$	$l = 3$
$n = 3$	$l = 4$	$l = 4$
$n = 4$	-1	-1

Tabela C.7: Matriz *LinOrd_{jtn}*

Encomenda 1 ($\tau_1 = 1, r_1 = 1, d_1 = 1, q_1 = 5000$)

- **DeterminaLimites(e = 1)**

$$a_1 = 3, a_2 = 1 \text{ e } a_3 = 1$$

DeterminaLimites(e), calcula um valor máximo para o número de linhas nas quais se pode planejar a encomenda (a_1), um valor máximo para o número de períodos nos quais se pode planejar a encomenda (a_2) e um valor máximo para o número de períodos entre o início de produção e d_e (a_3).

- **ColocaMaisTarde(e = 1)**

ColocaMaisTarde(e) coloca a encomenda o mais tarde possível, no menor número de períodos possível e no menor número de linhas possível (tendo em conta os índices de preferência, as capacidades disponíveis e os limites determinados em **DeterminaLimites(e)**).

$nper = 1, t_1 = 1$

– **CapProdTip(j = 1, nper = 1, t₁ = 1)** = 5000 $\geq q_1$

CapProdTip(j, nper, t₁) retorna a capacidade existente entre um período inicial t_1 e um período final $t_1 + nper - 1$, para produzir o tipo j sem ter em conta as linhas em que se pretende produzir.

$nlin = 1, Lista[1] = 2, Lista[2] = -1$

– **CapProdLin(j = 1, nlin = 1, nper = 1, t₁ = 1)** = 5000 $\geq q_1$

CapProdLin(j, nlin, nper, t₁) retorna a capacidade existente nas linhas pertencentes a *Lista*, para produzir o tipo j , entre um período inicial t_1 e um período final $t_1 + nper - 1$.

– **CarregaLinhas(e = 1, nlin = 1, nper = 1, t₁ = 1, caso = 1)**

CarregaLinhas(e, nlin, nper, t₁, caso), carrega as linhas incluídas em *Lista* com a encomenda e .

Se $caso = 1$, carrega as linhas de t_1 até $t_1 + nper - 1$,

se $caso = 2$, carrega as linhas de $t_1 + nper - 1$ até t_1 .

$x_{1,2,1} = 5000$

AtualizaCargas(quant = 5000, j = 1, l = 2, t = 1), gera os valores apresentados na Tabela C.8

ActualizaCargas(q, j, l, t) actualiza os valores da capacidade total disponível para o tipo j , no período t ($CapDisp_{jt}$) e os valores da capacidade disponível para o tipo j na linha l e período t ($CapDispLin_{jlt}$), dada uma quantidade q do produto tipo j que foi alocada à linha l no período t .

	t	Linha 1	Linha 2	Linha 3	Linha 4	
		<i>CapDispLin_{jlt}</i>				<i>CapDisp_{jt}</i>
$j = 1$	1	0	0	2500	5000	0
$j = 1$	2	0	5000	2500	5000	5000
$j = 1$	3	0	5000	2500	5000	5000
$j = 2$	1	5000	0	5000	2500	2500
$j = 2$	2	5000	0	5000	2500	2500
$j = 2$	3	5000	0	5000	2500	2500
		<i>PrimPerDispJL_{jl}</i>				<i>PrimPerDispJ_j</i>
$j = 1$		-1	2	2	2	2
$j = 2$		1	-1	1	1	1
		<i>UltimPerDispJL_{jl}</i>				<i>UltimPerDispJ_j</i>
$j = 1$		-1	3	3	3	3
$j = 2$		3	-1	3	3	3

Tabela C.8: Valores da capacidade disponível, primeiro período disponível e último período disponível.

Encomenda 2 ($\tau_2 = 1, r_2 = 1, d_2 = 2, q_2 = 10000$)

- **DeterminaLimites($e = 2$)**

$$a_1 = 3, a_2 = 2 \text{ e } a_3 = 2$$

- **ColocaMaisTarde($e = 2$)**

$$nper = 1, t_1 = 2$$

$$- \text{CapProdTip}(j = 1, nper = 1, t_1 = 2) = 5000 \leq q_2$$

$$nper = 2, t_1 = 1$$

$$- \text{CapProdTip}(j = 1, nper = 2, t_1 = 3) = 5000 \leq q_2$$

- **RelaxaLimites**(**e** = 2)

$$d_2 = 3$$

RelaxaLimites(**e**) encontra um conjunto de períodos e um conjunto de linhas para produzir e , mesmo que para isso devam ser relaxados alguns dos limites calculados inicialmente (relaxando mesmo, se necessário, a data de entrega, e em último caso, tentando produzir uma quantidade menor que a quantidade encomendada).

- **ColocaMaisCedo**(**e** = 2)

$$nper = 2, t_1 = 2$$

ColocaMaisCedo(**e**) coloca e o mais cedo possível, no menor número de períodos possível e no menor número de linhas possível (tendo em conta os índices de preferência, as capacidades disponíveis e os limites determinados em **DeterminaLimites**()).

$$* \text{CapProdTip}(\mathbf{j} = 1, \mathbf{nper} = 2, \mathbf{t_1} = 2) = 10000 \geq q_2$$

$$nlin = 1, Lista[1] = 2, Lista[2] = -1$$

$$* \text{CapProdLin}(\mathbf{j} = 1, \mathbf{nlin} = 1, \mathbf{nper} = 2, \mathbf{t_1} = 2) = 10000 \geq q_2$$

$$* \text{CarregaLinhas}(\mathbf{e} = 2, \mathbf{nlin} = 1, \mathbf{nper} = 2, \mathbf{t_1} = 1, \mathbf{caso} = 2)$$

$$x_{2,2,2} = 5000$$

$$x_{2,2,3} = 5000$$

$$\text{AtualizaCargas}(\mathbf{quant} = 5000, \mathbf{j} = 1, \mathbf{l} = 2, \mathbf{t} = 2)$$

AtualizaCargas(**quant** = 5000, **j** = 1, **l** = 2, **t** = 3), geram os valores apresentados na Tabela C.9

Encomenda 3 ($\tau_3 = 2, r_3 = 1, d_3 = 2, q_3 = 3000$)

- **DeterminaLimites**(**e** = 3)

$$a_1 = 2, a_2 = 1 \text{ e } a_3 = 2$$

- **ColocaMaisTarde**(**e** = 3)

$$nper = 1, t_1 = 2$$

	t	Linha 1	Linha 2	Linha 3	Linha 4	
		<i>CapDispLin_{jlt}</i>				<i>CapDisp_{jt}</i>
$j = 1$	1	0	0	2500	5000	0
$j = 1$	2	0	0	2500	5000	0
$j = 1$	3	0	0	2500	5000	0
$j = 2$	1	5000	0	5000	2500	2500
$j = 2$	2	5000	0	5000	2500	2500
$j = 2$	3	5000	0	5000	2500	2500
		<i>PrimPerDispJL_{jl}</i>				<i>PrimPerDispJ_j</i>
$j = 1$		-1	-1	-1	-1	-1
$j = 2$		1	-1	1	1	1
		<i>UltimPerDispJL_{jl}</i>				<i>UltimPerDispJ_j</i>
$j = 1$		-1	-1	-1	-1	-1
$j = 2$		3	-1	3	3	3

Tabela C.9: Valores da capacidade disponível, primeiro período disponível e último período disponível.

$$- \text{CapProdTip}(j = 2, nper = 1, t_1 = 2) = 2500 \leq q_3$$

$$t_1 = 1$$

$$- \text{CapProdTip}(j = 2, nper = 1, t_1 = 1) = 2500 \leq q_3$$

$$nper = 2, t_1 = 1$$

$$- \text{CapProdTip}(j = 2, nper = 2, t_1 = 1) = 5000 \geq q_3$$

$$nlin = 1, Lista[1] = 1, Lista[2] = -1$$

$$- \text{CapProdLin}(j = 2, nlin = 1, nper = 2, t_1 = 1) = 5000 \geq q_3$$

$$- \text{CarregaLinhas}(e = 2, nlin = 1, nper = 2, t_1 = 1, caso = 1)$$

$$x_{3,1,2} = 2500$$

$$x_{3,1,1} = 500$$

$$\text{AtualizaCargas}(\text{quant} = 500, j = 2, l = 1, t = 1)$$

$\text{AtualizaCargas}(\text{quant} = 2500, j = 2, l = 1, t = 2)$, geram os valores apresentados na Tabela C.10

	t	Linha 1	Linha 2	Linha 3	Linha 4	
		<i>CapDispLin_{jlt}</i>				<i>CapDisp_{jt}</i>
$j = 1$	1	0	0	2500	5000	0
$j = 1$	2	0	0	2500	5000	0
$j = 1$	3	0	0	2500	5000	0
$j = 2$	1	4500	0	5000	2500	2000
$j = 2$	2	2500	0	5000	2500	0
$j = 2$	3	5000	0	5000	2500	2500
		<i>PrimPerDispJL_{jl}</i>				<i>PrimPerDisp_{Jj}</i>
$j = 1$		-1	-1	-1	-1	-1
$j = 2$		1	-1	1	1	1
		<i>UltimPerDispJL_{jl}</i>				<i>UltimPerDisp_{Jj}</i>
$j = 1$		-1	-1	-1	-1	-1
$j = 2$		3	-1	3	3	3

Tabela C.10: Valores da capacidade disponível, primeiro período disponível e último período disponível.

Encomenda 4 ($\tau_4 = 2, r_4 = 2, d_4 = 3, q_4 = 3000$)

- **DeterminaLimites**($e = 4$)

$$a_1 = 2, a_2 = 1 \text{ e } a_3 = 2$$

- **ColocaMaisTarde**($e = 4$)

$$nper = 1, t_1 = 3$$

$$- \text{CapProdTip}(j = 2, nper = 1, t_1 = 3) = 2500 \leq q_4$$

$$t_1 = 2$$

$$- \text{CapProdTip}(j = 2, nper = 1, t_1 = 2) = 0 \leq q_4$$

$$nper = 2, t_1 = 2$$

$$- \text{CapProdTip}(j = 2, nper = 2, t_1 = 2) = 2500 \leq q_4$$

- **RelaxaLimites**(4)

Como o horizonte de planeamento termina no período 3, não é possível relaxar a data de entrega, e é necessário produzir menor quantidade ($q_4 = 2500$).

– **ColocaMaisTarde**($e = 4$)

$$nper = 1, t_1 = 3$$

$$* \text{CapProdTip}(j = 2, nper = 1, t_1 = 3) = 2500 \geq q_4$$

$$nlin = 1, Lista[1] = 1, Lista[2] = -1$$

$$* \text{CapProdLin}(j = 2, nlin = 1, nper = 1, t_1 = 3) = 2500 \geq q_4$$

$$* \text{CarregaLinhas}(e = 4, nlin = 1, nper = 2, t_1 = 3, caso = 1)$$

$$x_{4,1,3} = 2500$$

AtualizaCargas(quant = 2500, $j = 2$, $l = 1$, $t = 3$), gera os valores apresentados na Tabela C.11

Na Figura C.1 apresenta-se um gráfico onde se encontram representadas as linhas e períodos onde foram colocadas as encomendas e simultaneamente as cargas de cada uma das linhas.

	t	Linha 1	Linha 2	Linha 3	Linha 4	
			<i>CapDispLin_{jlt}</i>			<i>CapDisp_{jt}</i>
$j = 1$	1	0	0	2500	5000	0
$j = 1$	2	0	0	2500	5000	0
$j = 1$	3	0	0	2500	5000	0
$j = 2$	1	4500	0	5000	2500	2000
$j = 2$	2	2500	0	5000	2500	0
$j = 2$	3	2500	0	5000	2500	0
			<i>PrimPerDispJL_{jl}</i>			<i>PrimPerDispJ_j</i>
$j = 1$		-1	-1	-1	-1	-1
$j = 2$		1	-1	1	1	1
			<i>UltimPerDispJL_{jl}</i>			<i>UltimPerDispJ_j</i>
$j = 1$		-1	-1	-1	-1	-1
$j = 2$		1	-1	1	1	1

Tabela C.11: Valores da capacidade disponível, primeiro período disponível e último período disponível.

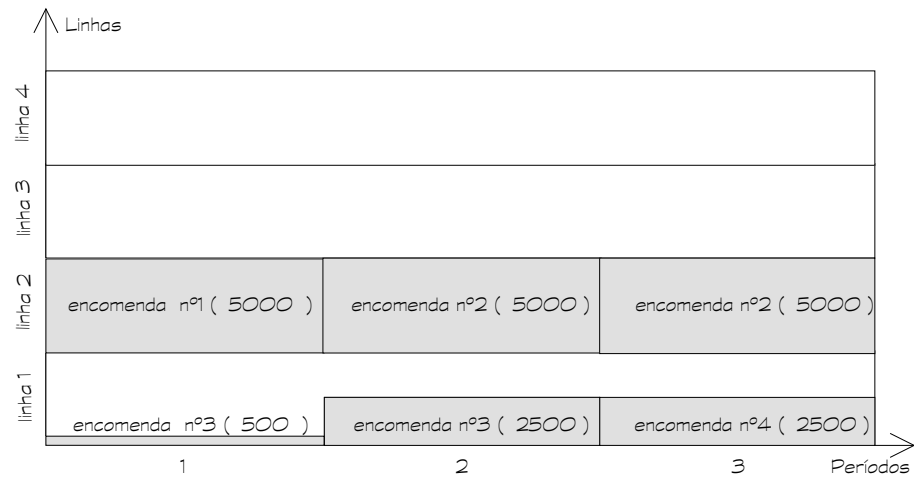


Figura C.1: Períodos e linhas em que foram colocadas as encomendas e ocupação das linhas

Apêndice D

Exemplo PROG

Apresenta-se a seguir um exemplo de resolução de uma instância do problema PROG com a heurística *H-PROG*. Para uma descrição completa da notação utilizada neste apêndice, recomenda-se a consulta do Apêndice A, e para uma descrição completa das funções utilizadas neste apêndice, recomenda-se a consulta do Apêndice B.

D.1 Dados de entrada

Os dados de entrada do PROG, são apresentados nas Tabelas D.1 a D.6. Considere-se que o número de minutos de produção de cada período é 2000.

T - número de períodos	3
E - número de Ordens de Fabrico (OF)	5
L - número de linhas	5
J - número de tipos de produtos	3
M - número de tipos de máquinas	3

Tabela D.1: Características gerais da instância

D.2 Resolução

A função **Inicializa()** chama **ActualizaCargas(0, j, l, t)** e gera os seguintes valores:

	Linha 1	Linha 2	Linha 3
	P_{jlt}		
$j = 1$	0	4	1
$j = 2$	1	0	3
$j = 3$	0	2	1
	C_{jlt}		U_{jt}
$j = 1$	0	5000	2500
$j = 2$	5000	0	2500
$j = 3$	0	2500	2500

Tabela D.2: Índices de preferência, P_{jlt} e capacidades de produção, C_{jlt} , das linhas l por tipo de produto j , para todos os períodos t e máxima produção possível do produto tipo j em todas as linhas no período t , U_{jt}

PréCostura	$j = 1$	$j = 2$	$j = 3$
$m = 1$	2	3	1
$m = 2$	0	0	0
$m = 3$	10	2	1
Costura			
$m = 1$	3	4	1
$m = 2$	5	4	3
$m = 3$	0	0	0

Tabela D.3: Gamas operatórias para cada tipo de produto: Carga do tipo de produto j na máquina m (em minutos por unidade)

$$\begin{aligned} \forall_{elt} \quad x_{elt} &= 0 \\ \forall_{lt} \quad CargTot_{lt} &= 0 \\ \forall_{jt} \quad CargTip_{jt} &= 0 \\ \forall_{lt} \quad mmc_{lt} &= 5000 \end{aligned}$$

e ainda os valores apresentados nas Tabelas D.7 e D.8

Tipo de máquina m	1	2	3
K_m	20	16	22

Tabela D.4: Quantidade de máquinas existentes de cada tipo (K_m)

Linha l	Tipo	Nº máximo de postos D_l
$l = 0$	Pré-costura	28
$l = 1$	Costura	15
$l = 2$	Costura	15
$l = 3$	Costura	15
$l = 4$	Exterior	∞

Tabela D.5: Número máximo de postos em cada linha

e	τ_e	r_e	d_e	q_e
1	1	1	3	4000
2	3	2	3	5000
3	2	2	3	3000
4	3	1	3	2500
5	2	1	1	5000

Tabela D.6: Lista de OF: tipo de produto (τ_e), “release date” (r_e), data de entrega (d_e) e quantidade (q_e)

Na Tabela D.7, $CapDispLin_{jlt}$ representa a capacidade disponível para cada tipo de produto j , em cada linha l e para cada período t (em unidades), e $CapDisp_{jt}$ representa a capacidade disponível para os produtos do tipo j no período t (em unidades)

	t	Linha 1	Linha 2	Linha 3	
		$CapDispLin_{jlt}$			$CapDisp_{jt}$
$j = 1$	1	0	5000	2500	2500
$j = 1$	2	0	5000	2500	2500
$j = 1$	3	0	5000	2500	2500
$j = 2$	1	5000	0	2500	5000
$j = 2$	2	5000	0	2500	5000
$j = 2$	3	5000	0	2500	5000
$j = 3$	1	0	2500	2500	5000
$j = 3$	2	0	2500	2500	5000
$j = 3$	3	0	2500	2500	5000

Tabela D.7: Valores iniciais de $CapDispLin_{jlt}$ e $CapDisp_{jt}$

Na Tabela D.8, $PrimPerDispJL_{jl}$ representa o primeiro período disponível para o tipo j , na linha l , $PrimPerDispJ_j$ representa o primeiro período disponível para o tipo j em qualquer linha, $UltimPerDispJL_{jl}$ re-

presenta o último período disponível na linha l para o tipo j e $UltimPerDispJ_j$ representa o último período disponível para o tipo j em qualquer linha. (Tomam o valor -1 se não há nenhum período disponível.)

	Linha 1	Linha 2	Linha 3	
	<i>PrimPerDispJL_{jl}</i>			<i>PrimPerDispJ_j</i>
$j = 1$	-1	1	1	1
$j = 2$	1	-1	1	1
$j = 3$	-1	1	1	1
	<i>UltimPerDispJL_{jl}</i>			<i>UltimPerDispJ_j</i>
$j = 1$	-1	3	3	3
$j = 2$	3	-1	3	3
$j = 3$	-1	3	3	3

Tabela D.8: Primeiros e últimos períodos onde existe capacidade disponível

Seguidamente, a função **OrdPrefLJ()**, utiliza os dados da Tabela D.2, para gerar a matriz que se encontra representada na Tabela D.9, que contém, para cada linha de produção, os tipos de produtos por ordem de preferência.

	<i>Linha1</i>	<i>Linha2</i>	<i>Linha3</i>
$n = 1$	$j = 2$	$j = 3$	$j = 1$
$n = 2$	-1	$j = 1$	$j = 3$
$n = 3$	-1	-1	$j = 2$

Tabela D.9: Matriz $TipProdOrd_{ltn}$, igual para todos os períodos t

Linha 1

$$Lista[1] = 1, Lista[2] = -1$$

$$t = 1 \quad n = 1 \quad TipProdOrd_{111} \rightarrow j = 2$$

Da lista de OF D.6, retirou-se a OF 5 que é a primeira OF do produto tipo 2 ($j = 2$), com $d_e \leq 1$.

- **CapProdLin($j = 2, nlin = 1, nper = 1, t_1 = 1$) = 5000 $\geq$ $q_5 = 5000$**

CapProdLin($j, nlin, nper, t_1$) retorna a capacidade existente nas linhas pertencentes a *Lista*, para produzir o tipo j , entre um período inicial t_1 e um período final $t_1 + nper - 1$.

- **CarregaLinhas**($e = 5, nlin = 1, nper = 1, t_1 = 1, caso = 1$)

CarregaLinhas($e, nlin, nper, t_1, caso$), carrega as linhas incluídas em *Lista* com a OF e .

Se $caso = 1$, carrega as linhas de t_1 até $t_1 + nper - 1$,

se $caso = 2$, carrega as linhas de $t_1 + nper - 1$ até t_1 .

$$x_{5,1,1} = 5000$$

$$colocou_5 = 1$$

AtualizaCargas($5000, j = 1, l = 1, t = 0$), gera os valores apresentados na Tabela D.10.

AtualizaCargas(q, j, l, t) actualiza os valores da capacidade total disponível para o tipo j , no período t ($CapDisp_{jt}$) e os valores da capacidade disponível para o tipo j na linha l e período t ($CapDispLin_{jlt}$), dada uma quantidade q do produto tipo j que foi alocada à linha l no período t .

Dado que só se pode produzir produtos tipo 2 na linha 1, só resta colocar a OF 3, com:

$$t = 3 \quad n = 1 \quad TipProdOrd_{131} \rightarrow j = 2$$

- **CapProdLin**($j = 2, nlin = 1, nper = 1, t_1 = 3$) = $5000 \geq q_3 = 3000$

- **CarregaLinhas**($e = 3, nlin = 1, nper = 1, t_1 = 3, caso = 1$)

$$x_{3,1,3} = 3000$$

$$colocou_3 = 1$$

AtualizaCargas($3000, j = 2, l = 1, t = 3$) gera os valores apresentados na Tabela D.11

Dado que não é possível colocar mais nenhuma OF na linha 1, passa-se para a linha 2.

Linha 2

$$Lista[1] = 2, Lista[2] = -1$$

t	Linha 1	Linha 2	Linha 3	
		$CapDispLin_{jlt}$		$CapDisp_{jt}$
$j = 1$	1	0	5000	2500
$j = 1$	2	0	5000	2500
$j = 1$	3	0	5000	2500
$j = 2$	1	0	0	2500
$j = 2$	2	5000	0	2500
$j = 2$	3	5000	0	2500
$j = 3$	1	0	2500	2500
$j = 3$	2	0	2500	2500
$j = 3$	3	0	2500	2500
		$PrimPerDispJL_{jl}$		$PrimPerDispJ_j$
$j = 1$		-1	1	1
$j = 2$		2	-1	1
$j = 3$		-1	1	1
		$UltimPerDispJL_{jl}$		$UltimPerDispJ_j$
$j = 1$		-1	3	3
$j = 2$		3	-1	3
$j = 3$		-1	3	3

Tabela D.10: $CapDisp$, $PrimPerDisp$, $UltimPerDisp$

$\mathbf{t} = \mathbf{1} \quad n = 1 \quad TipProdOrd_{211} \rightarrow j = 3$

Na lista de OF D.6, não há nenhuma OF do produto tipo 3 ($j = 3$), com $d_e \leq 1$.

$\mathbf{t} = \mathbf{1} \quad n = 2 \quad TipProdOrd_{212} \rightarrow j = 1$

Na lista de OF D.6, não há nenhuma OF do produto tipo 1 ($j = 1$), com $d_e \leq 1$.

$\mathbf{t} = \mathbf{1} \quad n = 3 \quad TipProdOrd_{213} \rightarrow -1$

$\mathbf{t} = \mathbf{2} \quad n = 1 \quad TipProdOrd_{221} \rightarrow j = 3$

Da lista de OF D.6, retirou-se a OF 2 que é a primeira OF do produto tipo 3 ($j = 3$) ainda não colocada, com $d_e \leq 2$ e $r_e \leq 2$. No entanto não é possível produzir essa OF dado que $CapDispLin_{322} = 2500 \leq q_2 = 5000$ e a data de início de produção de $e = 2$ é $r_2 = 2$.

$\mathbf{t} = \mathbf{2} \quad n = 2 \quad TipProdOrd_{222} \rightarrow j = 1$

	t	Linha 1	Linha 2	Linha 3	
		<i>CapDispLin_{jlt}</i>			<i>CapDisp_{jt}</i>
$j = 1$	1	0	5000	2500	2500
$j = 1$	2	0	5000	2500	2500
$j = 1$	3	0	5000	2500	2500
$j = 2$	1	0	0	2500	0
$j = 2$	2	5000	0	2500	5000
$j = 2$	3	2000	0	2500	2000
$j = 3$	1	0	2500	2500	5000
$j = 3$	2	0	2500	2500	5000
$j = 3$	3	0	2500	2500	5000
		<i>PrimPerDispJL_{jl}</i>			<i>PrimPerDispJ_j</i>
$j = 1$		-1	1	1	1
$j = 2$		2	-1	1	2
$j = 3$		-1	1	1	1
		<i>UltimPerDispJL_{jl}</i>			<i>UltimPerDispJ_j</i>
$j = 1$		-1	3	3	3
$j = 2$		3	-1	3	3
$j = 2$		-1	3	3	3

Tabela D.11: *CapDisp*, *PrimPerDisp*, *UltimPerDisp*

Na lista de OF D.6, não há nenhuma OF do produto tipo 1 ($j = 1$), com $d_e \leq 2$.

$\mathbf{t} = \mathbf{2} \quad n = 3 \quad \text{TipProdOrd}_{223} \rightarrow -1$

$\mathbf{t} = \mathbf{3} \quad n = 1 \quad \text{TipProdOrd}_{231} \rightarrow j = 3$

Da lista de OF D.6, retirou-se a OF 2 que é a primeira OF do produto tipo 3 ($j = 3$) ainda não colocada, com $d_e \leq 3$. É necessário produzir a OF em dois períodos $t = 3$ e $t = 2$.

- **CapProdLin**($\mathbf{j} = \mathbf{3}, \mathbf{nlin} = \mathbf{1}, \mathbf{nper} = \mathbf{2}, \mathbf{t_1} = \mathbf{2}$) = **5000** $\geq$ **q₂ = 5000**
- **CarregaLinhas**($\mathbf{e} = \mathbf{2}, \mathbf{nlin} = \mathbf{1}, \mathbf{nper} = \mathbf{2}, \mathbf{t_1} = \mathbf{2}, \mathbf{caso} = \mathbf{1}$)

$$x_{2,2,2} = 2500$$

$$x_{2,2,3} = 2500$$

$$colocou_2 = 1$$

AtualizaCargas(2500, j = 3, l = 2, t = 2) e
AtualizaCargas(2500, j = 3, l = 2, t = 3) geram os valores apresentados
na Tabela D.12

	t	Linha 1	Linha 2	Linha 3	
		<i>CapDispLin_{jlt}</i>			<i>CapDisp_{jt}</i>
$j = 1$	1	0	5000	2500	2500
$j = 1$	2	0	0	2500	2500
$j = 1$	3	0	0	2500	2500
$j = 2$	1	0	0	2500	0
$j = 2$	2	5000	0	2500	5000
$j = 2$	3	2000	0	2500	2000
$j = 3$	1	0	2500	2500	5000
$j = 3$	2	0	0	2500	2500
$j = 3$	3	0	0	2500	2500
		<i>PrimPerDispJL_{jl}</i>			<i>PrimPerDispJ_j</i>
$j = 1$		-1	1	1	1
$j = 2$		2	-1	1	2
$j = 3$		-1	1	1	1
		<i>UltimPerDispJL_{jl}</i>			<i>UltimPerDispJ_j</i>
$j = 1$		-1	1	3	3
$j = 2$		2	-1	3	3
$j = 3$		-1	1	3	3

Tabela D.12: *CapDisp*, *PrimPerDisp*, *UltimPerDisp*

Da lista de OF D.6, retirou-se a OF 4 que é a última OF do produto tipo 3 ($j = 3$) ainda não colocada, com $d_e \leq 3$.

- **CapProdLin(j = 3, nlin = 1, nper = 1, t₁ = 1) = 5000 ≥ q₄ = 2500**
- **CarregaLinhas(e = 4, nlin = 1, nper = 2, t₁ = 2, caso = 1)**

$$x_{4,2,1} = 2500$$

$$colocou_4 = 1$$

AtualizaCargas(2500, j = 3, l = 2, t = 1) gera os valores apresentados na Tabela D.13.

A linha 2 está completamente carregada.

Linha 3

$$Lista[1] = 3, Lista[2] = -1$$

A única OF que falta colocar é a $e = 1$, com

	t	Linha 1	Linha 2	Linha 3	
		<i>CapDispLin_{jt}</i>			<i>CapDisp_{jt}</i>
$j = 1$	1	0	0	2500	2500
$j = 1$	2	0	0	2500	2500
$j = 1$	3	0	0	2500	2500
$j = 2$	1	0	0	2500	0
$j = 2$	2	5000	0	2500	5000
$j = 2$	3	2000	0	2500	2000
$j = 3$	1	0	0	2500	2500
$j = 3$	2	0	0	2500	2500
$j = 3$	3	0	0	2500	2500
		<i>PrimPerDispJL_{jl}</i>			<i>PrimPerDispJ_j</i>
$j = 1$		-1	-1	1	1
$j = 2$		2	-1	1	2
$j = 3$		-1	-1	1	1
		<i>UltimPerDispJL_{jl}</i>			<i>UltimPerDispJ_j</i>
$j = 1$		-1	-1	3	3
$j = 2$		3	-1	3	3
$j = 3$		-1	-1	3	3

Tabela D.13: *CapDisp*, *PrimPerDisp*, *UltimPerDisp*

$t = 3$ $n = 1$ $TipProdOrd_{331} \rightarrow j = 1$

É necessário produzir a OF em dois períodos.

- **CapProdLin**($j = 1, nlin = 1, nper = 2, t_1 = 2$) = **5000** $\geq q_1 = 4000$
- **CarregaLinhas**($e = 1, nlin = 1, nper = 2, t_1 = 2, caso = 1$)

$$x_{1,3,2} = 1500$$

$$x_{1,3,3} = 2500$$

$$colocou_1 = 1$$

AtualizaCargas(1500, $j = 1, l = 3, t = 2$) e

AtualizaCargas(2500, $j = 1, l = 3, t = 3$), geram os valores apresentados na Tabela D.14

Cálculo de Layouts

Na Tabela D.15 apresentam-se resumidamente os resultados obtidos pela primeira parte da heurística, a carga para cada linha l , de cada tipo de produto j e em cada período t .

	t	Linha 1	Linha 2	Linha 3	
		$CapDispLin_{jlt}$			$CapDisp_{jt}$
$j = 1$	1	0	0	2500	2500
$j = 1$	2	0	0	1000	2000
$j = 1$	3	0	0	0	0
$j = 2$	1	0	0	2500	0
$j = 2$	2	5000	0	1500	5000
$j = 2$	3	2000	0	0	2000
$j = 3$	1	0	0	2500	2500
$j = 3$	2	0	0	1000	2000
$j = 3$	3	0	0	0	0
		$PrimPerDispJL_{jl}$			$PrimPerDispJ_j$
$j = 1$		-1	-1	1	1
$j = 2$		2	-1	1	2
$j = 3$		-1	-1	1	1
		$UltimPerDispJL_{jl}$			$UltimPerDispJ_j$
$j = 1$		-1	-1	2	2
$j = 2$		3	-1	2	3
$j = 3$		-1	-1	2	2

Tabela D.14: $CapDisp$, $PrimPerDisp$, $UltimPerDisp$

Os resultados obtido por **FazLayout()** e por **RoundLayout()** estão representados nas Figuras D.16 a D.18 e ainda nas Figuras D.19 e D.20.

A função **FazLayout()**, determina o número de minutos de cada tipo de máquina, que é necessário em cada linha e em cada período.

A função **RoundLayout()**, determina o número de máquinas de cada tipo necessário em cada linha e em cada período.

	t	Linha0	Linha1	Linha2	Linha3
$j = 1$	1	0	0	0	0
$j = 1$	2	500	0	0	1500
$j = 1$	3	2500	0	0	2500
$j = 2$	1	5000	5000	0	0
$j = 2$	2	0	0	0	0
$j = 2$	3	3000	3000	0	0
$j = 3$	1	2500	0	2500	0
$j = 3$	2	2500	0	2500	0
$j = 3$	3	2500	0	2500	0

Tabela D.15: Carga para cada linha l , tipo de produto j e período t

	$j = 1$		$j = 2$		$j = 3$		Total		
<i>Linha0</i>	Quant	p_{1jm}	Quant	p_{1jm}	Quant	p_{1jm}	$CapMaq_{mlt}$	$\frac{CapMaq_{mlt}}{b_t}$	$NMaq_{mlt}$
$m = 1$	0	2	5000	3	2500	1	17500	8.75	9
$m = 2$	0	0	5000	0	2500	0	0	0	0
$m = 3$	0	10	5000	2	2500	1	12500	6.25	7
<i>Linha1</i>		p_{2jm}		p_{2jm}		p_{2jm}			
$m = 1$	0	3	5000	4	0	1	20000	10	10
$m = 2$	0	5	5000	4	0	3	20000	10	10
$m = 3$	0	0	5000	0	0	0	0	0	0
<i>Linha2</i>									
$m = 1$	0	3	0	4	2500	1	2500	1.25	2
$m = 2$	0	5	0	4	2500	3	7500	3.75	4
$m = 3$	0	0	0	0	2500	0	0	0	0
<i>Linha3</i>									
$m = 1$	0	3	0	4	0	1	0	0	0
$m = 2$	0	5	0	4	0	3	0	0	0
$m = 3$	0	0	0	0	0	0	0	0	0

Tabela D.16: Número de máquinas necessárias em $t = 1$

	$j = 1$		$j = 2$		$j = 3$		Total		
<i>Linha0</i>	Quant	p_{1jm}	Quant	p_{1jm}	Quant	p_{1jm}	$CapMaq_{mlt}$	$\frac{CapMaq_{mlt}}{b_t}$	$NMaq_{mlt}$
$m = 1$	500	2	0	3	2500	1	3500	1.75	2
$m = 2$	500	0	0	0	2500	0	0	0	0
$m = 3$	500	10	0	2	2500	1	7500	3.75	4
<i>Linha1</i>		p_{2jm}		p_{2jm}		p_{2jm}			
$m = 1$	0	3	0	4	0	1	0	0	0
$m = 2$	0	5	0	4	0	3	0	0	0
$m = 3$	0	0	0	0	0	0	0	0	0
<i>Linha2</i>									
$m = 1$	0	3	0	4	2500	1	2500	1.25	2
$m = 2$	0	5	0	4	2500	3	7500	3.75	4
$m = 3$	0	0	0	0	2500	0	0	0	0
<i>Linha3</i>									
$m = 1$	500	3	0	4	0	1	4500	2.25	3
$m = 2$	500	5	0	4	0	3	7500	3.75	4
$m = 3$	500	0	0	0	0	0	0	0	0

Tabela D.17: Número de máquinas necessárias em $t = 2$

	$j = 1$		$j = 2$		$j = 3$		Total		
<i>Linha0</i>	Quant	p_{1jm}	Quant	p_{1jm}	Quant	p_{1jm}	$CapMaq_{mlt}$	$\frac{CapMaq_{mlt}}{b_t}$	$NMaq_{mlt}$
$m = 1$	2500	2	3000	3	2500	1	16500	8.25	9
$m = 2$	2500	0	3000	0	2500	0	0	0	0
$m = 3$	2500	10	3000	2	2500	1	33500	16.75	17
<i>Linha1</i>		p_{2jm}		p_{2jm}		p_{2jm}			
$m = 1$	0	3	3000	4	0	1	12000	6	6
$m = 2$	0	5	3000	4	0	3	12000	6	6
$m = 3$	0	0	3000	0	0	0	0	0	0
<i>Linha2</i>									
$m = 1$	0	3	0	4	2500	1	2500	1.25	2
$m = 2$	0	5	0	4	2500	3	7500	3.75	4
$m = 3$	0	0	0	0	2500	0	0	0	0
<i>Linha3</i>									
$m = 1$	0	3	0	4	2500	1	2500	1.25	2
$m = 2$	0	5	0	4	2500	3	7500	3.75	4
$m = 3$	0	0	0	0	2500	0	0	0	0

Tabela D.18: Número de máquinas necessárias em $t = 3$

$$K_m \quad \sum_l NMaq_{mlt} \quad t=1 \quad t=2 \quad t=3$$

$m = 1$	20	21	7	19
$m = 2$	16	14	8	14
$m = 3$	22	7	4	17

Tabela D.19: Número de máquinas existentes de cada tipo (K_m) e número de máquinas de cada tipo, necessárias em cada período

$$D_l \quad \sum_m NMaq_{mlt} \quad t=1 \quad t=2 \quad t=3$$

<i>Linha0</i>	28	16	6	26
<i>Linha1</i>	15	20	0	12
<i>Linha2</i>	15	6	6	6
<i>Linha3</i>	15	0	7	6

Tabela D.20: Número máximo de postos permitido em cada linha (D_l) e número de postos necessários em cada período e em cada linha

Nas Figuras D.1, D.2 e D.3, pode-se ver um resumo dos resultados obtidos. Na Figura D.1 apresenta-se um diagrama de Gantt em simultâneo com um diagrama de carga, indicando para que períodos e linhas foram planeadas as OF e qual a carga de cada linha em cada período. Na Figura D.2 é apresentado, para cada tipo de máquina, o número de máquinas existentes e o número de máquinas necessárias em cada período. O utilizador é alertado sempre que o número de máquinas necessárias num determinado período, exceder o número de máquinas existentes (ver seta na figura). Na Figura D.3 é apresentado, para cada linha, o número máximo de postos permitido, e o número de postos necessários em cada período. Neste caso o utilizador também é alertado sempre que o número de postos necessários numa determinada linha e num determinado período, exceda o máximo número de postos admissível nessa linha (ver seta na figura).

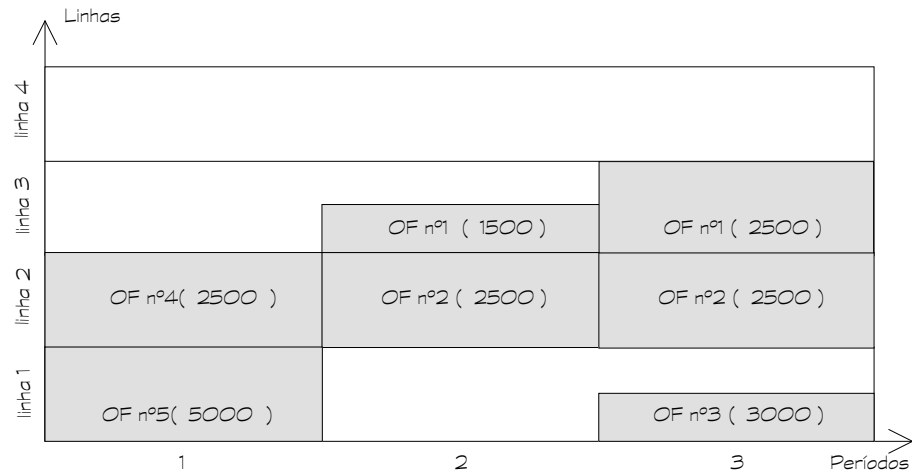


Figura D.1: Solução obtida: Períodos e linhas em que foram colocadas as OF e ocupação das linhas

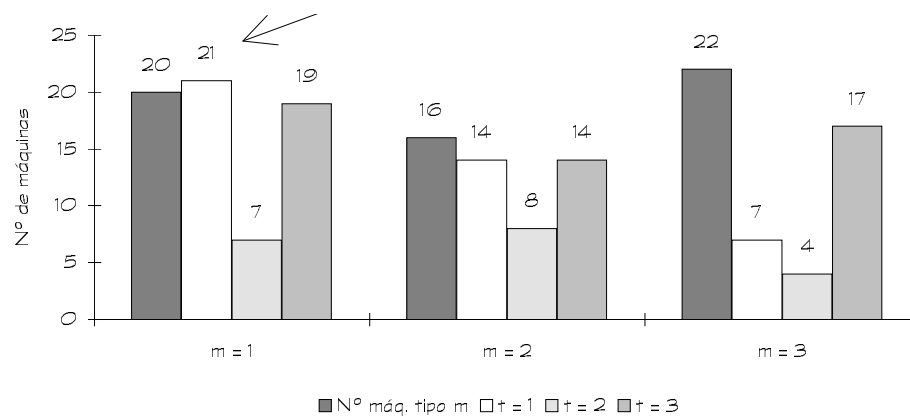


Figura D.2: Solução obtida: Número de máquinas existentes de cada tipo (K_m) e número de máquinas de cada tipo necessárias em cada período

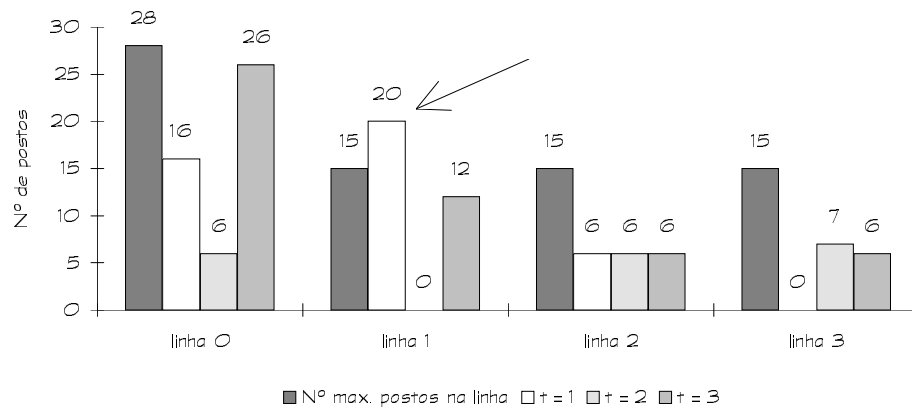


Figura D.3: Solução obtida: Número máximo de postos permitido em cada linha (D_t) e número de postos necessários em cada período e em cada linha

Apêndice E

Algumas Questões de Implementação

Para uma descrição completa da notação utilizada e das funções que se mencionam neste apêndice, recomenda-se a consulta, respectivamente, do Apêndice A e do Apêndice B.

E.1 Plataformas utilizadas nos testes

Estação de trabalho

A estação de trabalho que foi utilizada nesta dissertação, foi uma Hewlett Packard HP9000/720, com 32Mb de RAM. Os programas foram escritos em C e foram compilados pelo GNUCC (compilador de C++ da GNU).

CPLEX

O CPLEX é uma ferramenta informática para resolução de problemas lineares de otimização. O “CPLEX Linear Optimizer” resolve problemas com variáveis contínuas, enquanto que o “CPLEX Mixed Integer Solver”, permite resolver problemas com variáveis inteiras e contínuas. O CPLEX compreende também uma biblioteca de funções “CPLEX Callable Library” que cada utilizador pode usar em programas próprios. A versão do CPLEX que foi utilizada neste trabalho foi a 3.0, instalada na estação de trabalho da HP referida.

Computador pessoal

Para fazer testes para esta dissertação, foi também utilizado um computador pessoal, da Hewlett Packard, um HP Vectra VL, com o processador Pentium da Intel, a 75Mhz e com 8Mb de RAM. Os programas que correram neste computador, foram escritos em C e compilados usando o Borland C 3.1.

E.2 Utilização de mmc_{lt}

O cálculo de $CargTot_{lt}$, seguido do cálculo de $CapDispLin_{jlt}$ em **AtualizaCargas()**, de acordo com (ver Apêndice B):

$$CargTot_{lt} = CargTot_{lt} + \frac{quant}{C_{jlt}}$$

$$CapDispLin_{jlt} = \{1 - CargTot_{lt}\} * C_{jlt}$$

introduz muitos erros numéricos, dado que em alguns casos se somam números muito pequenos, correspondentes a $\frac{quant}{C_{jlt}}$. Para reduzir a possibilidade de surgirem esses erros numéricos, passou a utilizar-se mmc_{lt} que corresponde ao menor múltiplo comum dos valores de C_{jlt} . As equações para $CargTot_{lt}$ e $CapDispLin_{jlt}$ em **AtualizaCargas()**, passam a ser as seguintes:

$$CargTot_{lt} = CargTot_{lt} + \frac{quant}{C_{jlt}} * mmc_{lt}$$

$$CapDispLin_{jlt} = \{mmc_{lt} - CargTot_{lt}\} * \frac{C_{jlt}}{mmc_{lt}}$$

E.3 Alteração dos valores de C_{jlt} , após a leitura de dados

Os valores de C_{jlt} são utilizados no cálculo de mmc_{lt} :

$$\forall_{lt} \quad mmc_{lt} = mmc_j[C_{jlt}]$$

e é necessário assegurar que o valor obtido para mmc_{lt} não excede o número de dígitos de precisão de um “double”.

C_{jlt} , corresponde a uma previsão da capacidade de produção diária do tipo j na linha l num período t , se se produzir apenas o tipo j . Dado que pequenas alterações nessa previsão não dão origem a alterações significativas no planeamento da produção, optou-se por alterar ligeiramente os valores de C_{jlt} , de tal modo que ficasse assegurado que mmc_{lt} não excedia o número de dígitos de precisão de um “double”.

Sabe-se, por um lado, que os valores de C_{jlt} não excedem 10000, e por outro lado, que o número de dígitos de precisão de um “double” é 16. É então necessário assegurar que o maior número que pode ser gerado pelo algoritmo, mmc_{lt} , não excede 10^{16} .

O método utilizado para fazer a substituição de C_{jlt} foi o seguinte: Valores de C_{jlt} entre 0 e 199 são substituídos pelo valor mais próximo com maior factor primo ≤ 17 . Os valores de C_{jlt} entre 200 e 10000 são substituídos pelo valor mais próximo do maior inteiro contido em $\frac{C_{jlt}}{50}$, com maior factor primo ≤ 17 , multiplicado por 50. Deste modo, os novos valores para C_{jlt} têm que pertencer a um dos dois intervalos seguintes: $0 \leq C_{jlt} \leq 199$ ou $\frac{200}{50}50 \leq C_{jlt} \leq \frac{10000}{50}50$. Nesse caso, pode-se verificar que mmc_{lt} nunca excede $17^1 * 13^2 * 11^2 * 7^2 * 5^3 * 3^4 * 2^7 * 50$, porque $17^2 = 289 > 200$, $13^3 = 2197 > 200$, $11^3 = 1331 > 200$, $7^3 = 343 > 200$, $5^4 = 625 > 200$, $3^5 = 243 > 200$ e $2^8 = 256 > 200$. Por outro lado, $17^1 * 13^2 * 11^2 * 7^2 * 5^3 * 3^4 * 2^7 * 50 = 17^1 * 13^2 * 11^2 * 7^2 * 3^4 * 2^3 * 10^5 = 1103804301600000 < 10^{16}$.

E.4 Transição entre orçamentos

Descreve-se nesta secção o procedimento utilizado para ultrapassar problemas que surgem na transição entre orçamentos (chama-se orçamento a um grupo de períodos com capacidades de produção e índices de preferência iguais para todas as linhas e tipos de produtos). Na Tabela E.1, apresenta-se um exemplo em que há dois orçamentos, um tipo de produto e duas linhas de produção.

Considere-se que se pretende planear uma encomenda do tipo 1, com

Orçamento 1 ($t_{ini} = 1$ $t_{fim} = 3$)	Linha 1	Linha 2
Tipo 1	0	1
Orçamento 2 ($t_{ini} = 4$ $t_{fim} = 5$)	Linha 1	Linha 2
Tipo 1	1	0

Tabela E.1: Índices de preferência

data de entrega proposta de 3, e que se tentou sem sucesso produzir a encomenda dentro da data de entrega. A seguir, atrasa-se a data de entrega e tenta-se produzir o mais cedo possível. Do período 1 ao período 3, a encomenda pode ser produzida na linha 2 e a partir do período 4, pode ser produzida na linha 1. Dado que o algoritmo só verifica os índices de preferência no primeiro período de produção (o período 1), vai escolher apenas a linha 2 (e não as linhas 1 e 2) e não consegue planejar a encomenda. Para ultrapassar este problema, optou-se por alterar os dados de entrada do seguinte modo:

Desde que haja um período t_1 para o qual $P_{jlt} \neq 0$, todos os períodos t para os quais $P_{jlt} = 0$ passam a ter um índice de preferência P_{jlt} superior a todos os outros (por exemplo 9).

Introduzindo as alterações indicadas na Tabela E.1, obtém-se a Tabela E.2, e se se considerar a mesma encomenda, neste caso já seria possível planejar a encomenda para as duas linhas (1 e 2), sendo a encomenda planeada para a linha 2 no orçamento 1 e na linha 1 no orçamento 2.

Orçamento 1 ($t_{ini} = 1$ $t_{fim} = 3$)	Linha 1	Linha 2
Tipo 1	9	1
Orçamento 2 ($t_{ini} = 4$ $t_{fim} = 5$)	Linha 1	Linha 2
Tipo 1	1	9

Tabela E.2: Novos índices de preferência

Apêndice F

Problemas Teste

Neste apêndice apresentam-se as instâncias que foram utilizadas nos testes comparativos apresentados nesta dissertação. Para cada tipo de problema em questão, apresenta-se em primeiro lugar o gerador de instâncias utilizado e a seguir as várias instâncias de teste. Nos geradores de instâncias de teste. Para uma descrição completa da notação utilizada neste apêndice, recomenda-se a consulta do Apêndice A.

F.1 Problemas teste para o PDP

F.1.1 Processo de geração

O gerador de instâncias teste para o modelo PDP baseia-se na distribuição Γ , cuja função densidade de probabilidade se encontra representada na Figura F.1.

Os parâmetros que podem ser controlados para se obterem problemas com características pré-definidas são:

T , com o qual se obtém o número de períodos de planeamento T

$$T = \mathcal{T} * \Gamma(3, 2); \quad (\text{F.1})$$

$NumDiasPer$, com o qual se obtém o número de dias de cada período T

$$NumDias_t = NumDiasPer * \Gamma(3, 2); \quad (\text{F.2})$$

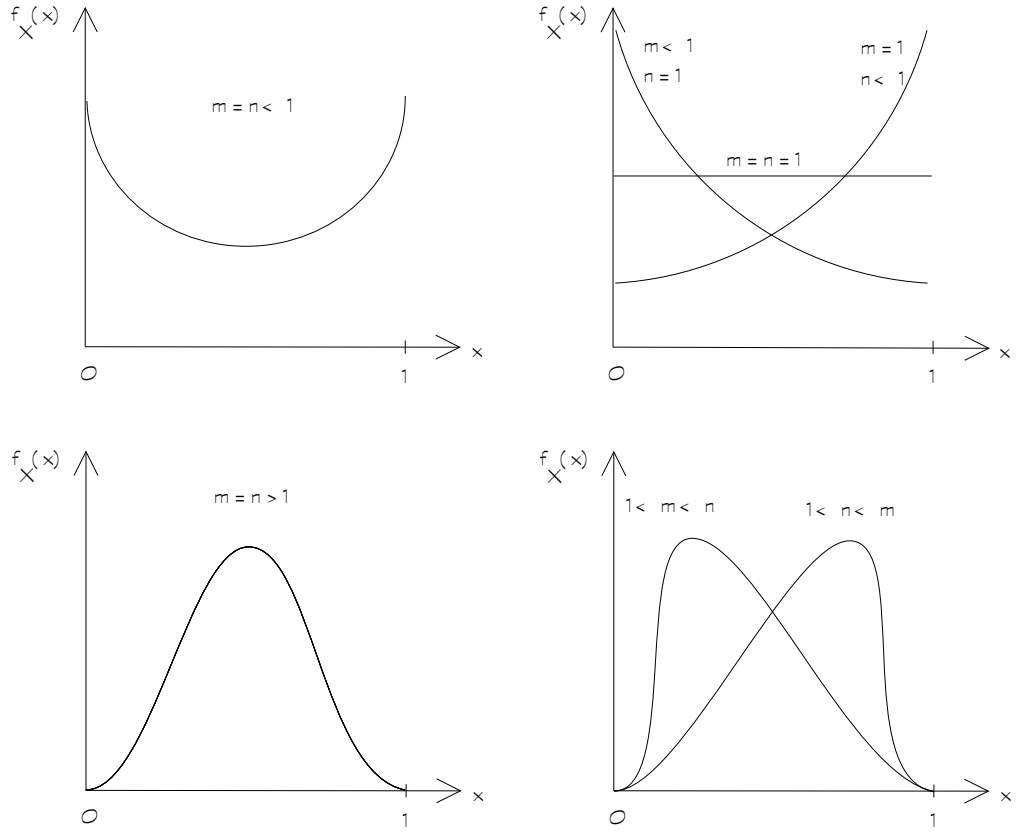


Figura F.1: Função densidade de probabilidade da distribuição $\Gamma(m, n)$, para diversos valores dos parâmetros m e n

$\mathcal{L}$, com o qual se obtém o número de linhas L

$$L = \mathcal{L} * \Gamma(3, 2); \quad (\text{F.3})$$

$\mathcal{F}$, com o qual se obtém o número de fábricas F

$$F = \mathcal{F} * \Gamma(3, 2); \quad (\text{F.4})$$

e seguidamente a distribuição das linhas pelas fábricas

$$F_l = F * \Gamma(1, 1); \quad (\text{F.5})$$

$\mathcal{J}$, com o qual se obtém o número de tipos de produtos J

$$J = \mathcal{J} * \Gamma(3, 2); \quad (\text{F.6})$$

$\mathcal{U}$, com o qual se obtém a quantidade máxima de produção do tipo j no período t , U_{jt}

$$U_{jt} = \mathcal{U} * \Gamma(3, 2); \quad (\text{F.7})$$

$\mathcal{C}$, com o qual se obtém a quantidade máxima de produção do tipo j , na linha l e período t , C_{jlt}

$$C_{jlt} = \mathcal{C} * \Gamma(3, 2); \quad (\text{F.8})$$

$\mathcal{P}$, com o qual se obtém o índice de preferência para produção do tipo j , na linha l e período t , P_{jlt}

$$P_{jlt} = \mathcal{P} * \Gamma\left(\frac{1}{2}, \frac{1}{2}\right); \quad (\text{F.9})$$

$\mathcal{E}$, com o qual se obtém o número de encomendas, E

$$E = \mathcal{E} * \Gamma(3, 2). \quad (\text{F.10})$$

Com o valor de T obtido anteriormente, obtém-se a data de entrega de cada encomenda, d_e

$$d_e = T * \Gamma(2, 3); \quad (\text{F.11})$$

e a data menor para início de produção de cada encomenda, r_e

$$r_e = T * \Gamma(3, 2). \quad (\text{F.12})$$

Com o valor de J obtido anteriormente, obtém-se o tipo de produto de cada encomenda, τ_e

$$\tau_e = J * \Gamma(1, 1). \quad (\text{F.13})$$

Por fim, com $\mathcal{Q}$, obtém-se a quantidade de cada encomenda, q_e

$$q_e = \mathcal{Q} * \Gamma(1, 1). \quad (\text{F.14})$$

F.1.2 Instância PDP1

T	20
$NumDiasPer$	5
$\mathcal{L}$	5
$\mathcal{J}$	10
$\mathcal{U}$	2000
$\mathcal{C}$	2000
$\mathcal{P}$	10
$\mathcal{E}$	15
$\mathcal{Q}$	5000

Tabela F.1: Parâmetros introduzidos no gerador para gerar PDP1

T - número de períodos	14
E - número de encomendas	14
L - número de linhas	4
F - número de fábricas	3
J - número de tipos de produtos	8

Tabela F.2: Características da instância PDP1

Linha	1	2	3	4
Fábrica	1	2	3	1

Tabela F.3: Fábrica a que pertence cada uma das linhas l

Período t	1	2	3	4	5	6	7	8	9	10	11	12	13	14
Número de dias	1	4	4	4	1	2	4	1	3	4	2	3	1	932

Tabela F.4: Número de dias de cada período t

$t = 1$	Linha 1	Linha 2	Linha 3	Linha 4	
P_{jlt}					
$j = 1$	0	8	9	0	
$j = 3$	0	9	8	9	
$j = 3$	9	0	9	0	
$j = 4$	0	3	0	0	
$j = 5$	0	9	1	1	
$j = 6$	8	0	5	3	
$j = 7$	0	4	4	0	
$j = 7$	3	6	2	0	
C_{jlt}					
U_{jt}					
$j = 1$	0	1260	1250	0	1560
$j = 2$	0	1360	1980	1890	1890
$j = 3$	1950	0	1350	0	1530
$j = 4$	0	1440	0	0	1800
$j = 5$	0	1700	1330	1800	1650
$j = 6$	1870	0	1820	1750	810
$j = 7$	0	810	1620	0	1980
$j = 8$	1170	1330	1650	0	500
$t = 2$	Linha 1	Linha 2	Linha 3	Linha 4	
P_{jlt}					
$j = 1$	3	9	9	9	
$j = 2$	0	0	2	7	
$j = 3$	9	5	2	0	
$j = 4$	6	0	4	9	
$j = 5$	9	0	8	8	
$j = 6$	0	9	4	2	
$j = 7$	0	9	0	1	
$j = 8$	3	6	0	6	
C_{jlt}					
U_{jt}					
$j = 1$	1360	900	1560	480	1950
$j = 2$	0	0	1540	1140	1250
$j = 3$	1820	1900	840	0	1120
$j = 4$	880	0	880	1760	1800
$j = 5$	1400	0	1800	1560	1870
$j = 6$	0	1050	1650	1400	1600
$j = 7$	0	1960	0	1250	1710
$j = 8$	1300	1700	0	1210	1430

Tabela F.5: Índices de preferência e capacidades de produção diárias por tipo de produto j , no período 1 e no período 2

$3 \leq t \leq 4$	Linha 1	Linha 2	Linha 3	Linha 4	
P_{jlt}					
$j = 1$	8	7	0	0	
$j = 2$	8	9	9	9	
$j = 3$	7	1	0	8	
$j = 4$	9	0	9	8	
$j = 5$	5	6	1	7	
$j = 6$	8	0	9	8	
$j = 7$	9	0	0	2	
$j = 8$	4	6	8	7	
C_{jlt}					
U_{jt}					
$j = 1$	1400	1800	0	0	1470
$j = 2$	1560	1680	1870	680	910
$j = 3$	1330	1710	0	1520	1540
$j = 4$	1470	0	1400	1650	1050
$j = 5$	1020	1360	1210	1890	1980
$j = 6$	1960	0	900	1900	1470
$j = 7$	1430	0	0	990	1440
$j = 8$	810	1430	1360	1470	520
$5 \leq t \leq 13$	Linha 1	Linha 2	Linha 3	Linha 4	
P_{jlt}					
$j = 1$	0	9	3	7	
$j = 2$	8	3	9	8	
$j = 3$	9	9	3	9	
$j = 4$	8	9	6	6	
$j = 5$	9	0	7	0	
$j = 6$	1	9	9	8	
$j = 7$	9	9	8	0	
$j = 8$	9	9	9	0	
C_{jlt}					
U_{jt}					
$j = 1$	0	1560	1760	1960	1870
$j = 2$	1600	1500	1560	1020	1280
$j = 3$	910	1600	1960	1710	960
$j = 4$	1600	1760	1440	1960	1560
$j = 5$	840	0	1760	1900	1250
$j = 6$	1540	1020	1920	1540	1700
$j = 7$	880	1330	1530	0	1430
$j = 8$	1870	640	1800	0	1600

Tabela F.6: Índices de preferência e capacidades de produção diárias por tipo de produto j , nos períodos 3 e 4 e do período 5 ao período 13

$t = 14$	Linha 1	Linha 2	Linha 3	Linha 4
P_{jlt}				
$j = 1$	1	0	0	0
$j = 2$	1	0	0	0
$j = 3$	1	0	0	0
$j = 4$	1	0	0	0
$j = 5$	1	0	0	0
$j = 6$	1	0	0	0
$j = 7$	1	0	0	0
$j = 8$	1	0	0	0
C_{jlt}				
$j = 1$	2000	0	0	0
$j = 2$	2000	0	0	0
$j = 3$	2000	0	0	0
$j = 4$	2000	0	0	0
$j = 5$	2000	0	0	0
$j = 6$	2000	0	0	0
$j = 7$	2000	0	0	0
$j = 8$	2000	0	0	0
U_{jt}				
$j = 1$	2000	0	0	2000
$j = 2$	2000	0	0	2000
$j = 3$	2000	0	0	2000
$j = 4$	2000	0	0	2000
$j = 5$	2000	0	0	2000
$j = 6$	2000	0	0	2000
$j = 7$	2000	0	0	2000
$j = 8$	2000	0	0	2000

Tabela F.7: Índices de preferência e capacidades de produção diárias por tipo de produto j , no período 14

e	τ_e	r_e	d_e	q_e
1	6	1	14	3915
2	1	11	12	3333
3	4	6	10	4575
4	7	3	14	1503
5	3	1	12	906
6	3	5	14	3027
7	6	5	10	1579
8	4	3	13	4823
9	8	10	14	4764
10	1	4	9	2685
11	7	2	13	152
12	5	7	9	2228
13	7	8	14	4899
14	6	3	5	3106

Tabela F.8: Lista de encomendas

F.2 Problemas teste para o modelo PAP

F.2.1 Processo de geração

O gerador de instâncias teste para o modelo PAP, baseia-se na distribuição Γ , cuja função densidade de probabilidade se encontra representada na Figura F.1.

Os parâmetros que podem ser controlados para se obterem problemas com características pré-definidas são, $\mathcal{T}$ e $\mathcal{C}$, e com os quais se obtém o número de períodos de planeamento, T

$$T = \mathcal{T} * \Gamma(3, 1); \quad (\text{F.15})$$

a capacidade disponível em cada período, C_t

$$C_t = \mathcal{C} * \Gamma\left(\frac{1}{2}, \frac{1}{2}\right); \quad (\text{F.16})$$

e a data de entrega de cada encomenda, d_e

$$d_e = T * \Gamma(1, 1). \quad (\text{F.17})$$

Também se tem que definir a relação entre a quantidade total encomendada e capacidade total (folga), p

$$\sum_t C_t = \frac{p}{100} * \sum_e d_e; \quad (\text{F.18})$$

a relação entre o número de encomendas médias mais pequenas ($M + P$) e o número de encomendas grandes (G), p_1

$$G = p_1 * (M + P); \quad (\text{F.19})$$

e a relação entre o número de encomendas pequenas (P) e o número de encomendas médias (M), p_2

$$M = p_2 * P. \quad (\text{F.20})$$

F.2.2 Instâncias geradas

Apresentam-se nesta secção todas as instâncias geradas para o problema PAP.

	PAP1	PAP2	PAP3	PAP4	PAP5
$\mathcal{T}$	40	40	60	60	60
$\mathcal{C}$	5	10	8	8	6
p	1.3	1.4	1.2	1.02	1.1
p_1	0.5	0.5	0.5	0.5	0.5
p_2	0.5	0.5	0.5	0.5	0.5

Tabela F.9: Parâmetros introduzidos no gerador para gerar instâncias PAP

	PAP1	PAP2	PAP3	PAP4	PAP5
T - número de períodos	31	20	34	31	38
E - número de encomendas	36	42	58	63	46
P - número de encomendas pequenas	16	19	26	28	21
M - número de encomendas médias	8	9	13	14	10
G - número de encomendas grandes	12	14	19	21	15

Tabela F.10: Características das instâncias PAP

Instância PAP1

t	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
C_t	5	3	2	4	4	4	0	5	5	3	2	1	0	1	4	2
t	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	
C_t	0	4	3	2	0	5	5	5	0	5	0	5	0	2	5	

Tabela F.11: PAP1 - Capacidade disponível, C_t em cada período t

16 encomendas pequenas ($q_e = 1$)																
e	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
d_e	29	21	1	20	23	16	15	13	11	3	19	26	28	5	28	3
8 encomendas médias ($q_e = 2$)																
e	17	18	19	20	21	22	23	24								
d_e	0	16	20	25	16	12	0	24								
12 encomendas grandes ($q_e = 3$)																
e	25	26	27	28	29	30	31	32	33	34	35	36				
d_e	27	19	4	2	27	28	1	27	11	7	19	1				

Tabela F.12: PAP1 - Encomendas e , com respectivos períodos de entrega propostos d_e

Instância PAP2

t	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
C_t	7	7	1	9	9	2	2	6	6	8	3	4	5	3	5	6

t	17	18	19	20
C_t	8	9	1	5

Tabela F.13: PAP2 - Capacidade disponível, C_t em cada período t 19 encomendas pequenas ($q_e = 1$)

e	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
d_e	12	9	14	0	13	2	14	14	10	9	17	10	13	16	4	5

e	17	18	19
d_e	10	12	2

9 encomendas médias ($q_e = 2$)

e	20	21	22	23	24	25	26	27	28
d_e	7	13	2	13	18	12	17	14	11

14 encomendas grandes ($q_e = 3$)

e	29	30	31	32	33	34	35	36	37	38	39	40	41	42
d_e	7	11	14	12	6	7	15	12	18	1	8	15	7	9

Tabela F.14: PAP2 - Encomendas e , com respectivos períodos de entrega propostos d_e

Instância PAP3

t	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
C_t	0	2	0	7	4	1	0	4	0	5	6	0	7	4	0	7	2

t	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34
C_t	5	7	3	7	7	6	1	7	0	4	3	6	7	5	6	6	6

Tabela F.15: PAP3 - Capacidade disponível, C_t em cada período t 26 encomendas pequenas ($q_e = 1$)

e	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
d_e	27	8	26	21	29	15	32	20	11	29	19	7	15	2	8	28

e	17	18	19	20	21	22	23	24	25	26
d_e	9	13	7	19	28	7	4	30	18	26

13 encomendas médias ($q_e = 2$)

e	27	28	29	30	31	32	33	34	35	36	37	38	39
d_e	28	28	6	2	29	27	22	24	4	0	21	21	21

19 encomendas grandes ($q_e = 3$)

e	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55
d_e	26	25	25	31	9	2	22	11	3	30	32	4	12	23	13	4

e	56	57	58
d_e	31	12	10

Tabela F.16: PAP3 - Encomendas e , com respectivos períodos de entrega propostos d_e

Instância PAP4

t	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
C_t	0	7	0	5	2	5	2	0	7	3	2	7	3	7	7	1

t	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
C_t	3	7	7	3	7	6	3	7	7	2	2	1	2	5	3

Tabela F.17: PAP4 - Capacidade disponível, C_t em cada período t 28 encomendas pequenas ($q_e = 1$)

e	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
d_e	0	9	6	18	16	12	29	18	3	1	26	15	23	23	4	14

e	17	18	19	20	21	22	23	24	25	26	27	28
d_e	21	14	18	11	2	25	24	29	19	16	17	12

14 encomendas médias ($q_e = 2$)

e	29	30	31	32	33	34	35	36	37	38	39	40	41	42
d_e	24	0	8	10	4	25	19	24	27	5	17	15	7	1

21 encomendas grandes ($q_e = 3$)

e	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58
d_e	18	29	17	17	10	24	4	11	24	5	12	25	12	15	4	4

e	59	60	61	62	63
d_e	23	27	17	12	19

Tabela F.18: PAP4 - Encomendas e , com respectivos períodos de entrega propostos d_e

Instância PAP5

t	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
C_t	5	4	4	1	0	3	5	5	0	0	5	0	4	0	3	4
t	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32
C_t	3	3	4	5	0	1	1	0	4	5	1	5	4	0	3	3
t	33	34	35	36	37	38										
C_t	3	0	1	2	5	5										

Tabela F.19: PAP5 - Capacidade disponível, C_t em cada período t

21 encomendas pequenas ($q_e = 1$)																
e	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
d_e	0	3	23	33	36	0	19	5	14	2	22	30	21	32	14	28
e	17	18	19	20	21											
d_e	2	2	3	4	2											
10 encomendas médias ($q_e = 2$)																
e	22	23	24	25	26	27	28	29	30	31						
d_e	8	36	21	8	20	15	1	27	33	7						
15 encomendas grandes ($q_e = 3$)																
e	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	
d_e	6	33	23	14	26	0	28	21	14	7	2	16	31	16	11	

Tabela F.20: PAP5 - Encomendas e , com respectivos períodos de entrega propostos d_e

Apêndice G

Demonstração da Propriedade 1 (secção 6.3.1)

Proposição

Se o modelo PAP apresentado na secção 6.2.4 tiver soluções admissíveis para uma determinada instância, então existe uma solução óptima na qual as encomendas de cada tamanho estão colocadas por ordem de data de entrega (EDD).

Na demonstração apresentada a seguir, generalizou-se a função objectivo do modelo PAP, considerando uma função objectivo tal como a representada na Figura G.1, em que f_1 e f_2 verifiquem as equações G.1, G.2 e G.3 e se possam desenvolver em série de Taylor com termos positivos.

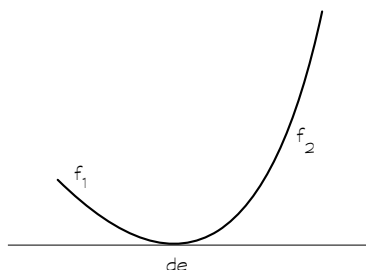


Figura G.1: Função objectivo genérica

$$\forall_{d_e} \quad f_1(d_e - x - y) \geq f_1(d_e - x) + f_1(d_e - y) \quad (\text{G.1})$$

$$\forall_{d_e} \quad f_2(d_e + x + y) \geq f_2(d_e + x) + f_2(d_e + y) \quad (\text{G.2})$$

$$\forall_{x>0} \quad 0 < f_1(d_e - x) < f_2(d_e + x) \quad (\text{G.3})$$

$$f_1(d_e) = f_2(d_e) = 0 \quad (\text{G.4})$$

Com a inequação G.1, pretende-se indicar que a função f_1 é decrescente para valores inferiores a d_e . Com a inequação G.2, pretende-se indicar que a função f_2 é crescente para valores superiores a d_e . Com a inequação G.3, pretende-se indicar que, à mesma “distância” de d_e , o valor de f_2 é sempre superior ao valor de f_1 .

Note-se que, se $d_e = 0$, as equações G.1, G.2, G.3 e G.4 passam respectivamente a:

$$f_1(-x - y) \geq f_1(-x) + f_1(-y) \quad (\text{G.5})$$

$$f_2(x + y) \geq f_2(x) + f_2(y) \quad (\text{G.6})$$

$$\forall_{x>0} \quad 0 \leq f_1(-x) \leq f_2(x) \quad (\text{G.7})$$

$$f_1(0) = f_2(0) = 0 \quad (\text{G.8})$$

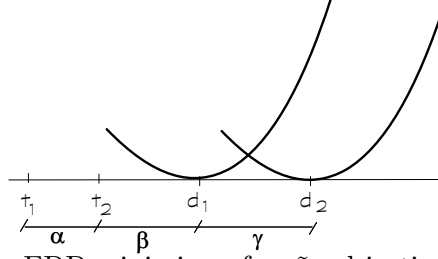
Demonstração

Um conjunto de encomendas está afecto a um conjunto de períodos por EDD, se e só se qualquer par de encomendas estiver colocado segundo EDD.

Vai-se a seguir demonstrar que dadas quaisquer duas encomendas do mesmo tamanho com datas de entrega d_1 e d_2 com $d_1 < d_2$ e quaisquer dois períodos de colocação, t_1 e t_2 , com $t_1 < t_2$, a colocação da encomenda com data de entrega d_1 no período t_1 e da encomenda com data de entrega d_2 no período t_2 , minimiza a função objectivo. Teremos nesse caso que analisar as seguintes 5 posições relativas de d_1 , d_2 , t_1 e t_2 :

Caso 1

$$t_1 < t_2 \leq d_1 < d_2$$



Para que a colocação por EDD minimize a função objectivo, é necessário que:

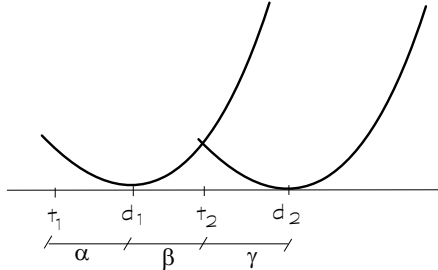
$$f_1(d_1 - t_1) + f_1(d_2 - t_2) \leq f_1(d_1 - t_2) + f_1(d_2 - t_1) \quad (G.9)$$

O que é equivalente a:

$$f_1(\alpha + \beta) + f_1(\beta + \gamma) \leq f_1(\beta) + f_1(\alpha + \beta + \gamma) \quad (G.10)$$

Caso 2

$$t_1 \leq d_1 \leq t_2 \leq d_2$$



Para que a colocação por EDD minimize a função objectivo, é necessário que:

$$f_1(d_1 - t_1) + f_1(d_2 - t_2) \leq f_1(d_2 - t_1) + f_2(t_2 - d_1) \quad (G.11)$$

O que é equivalente a:

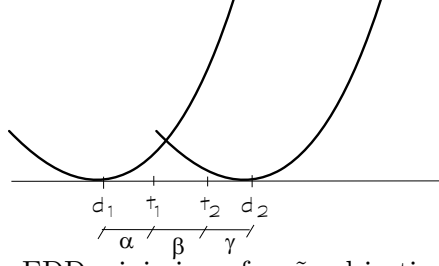
$$f_1(\alpha) + f_1(\gamma) \leq f_1(\alpha + \beta + \gamma) + f_2(\beta) \quad (G.12)$$

E se f_1 e f_2 verificarem as equações G.5, G.6 e G.7, a desigualdade é estrita (o que significa que neste caso a função objectivo melhora realmente se as encomendas forem colocadas por EDD).

$$f_1(\alpha) + f_1(\gamma) < f_1(\alpha + \beta + \gamma) + f_2(\beta) \quad (\text{G.13})$$

Caso 3

$$d_1 \leq t_1 < t_2 \leq d_2$$



Para que a colocação por EDD minimize a função objectivo, é necessário que:

$$f_2(t_1 - d_1) + f_1(d_2 - t_2) \leq f_2(t_2 - d_1) + f_1(d_2 - t_1) \quad (\text{G.14})$$

O que é equivalente a:

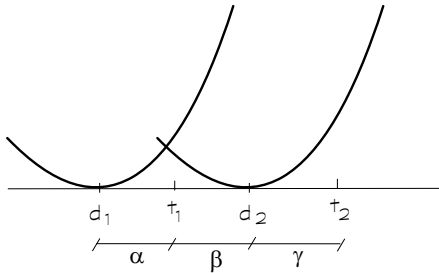
$$f_2(\alpha) + f_1(\gamma) \leq f_2(\alpha + \beta) + f_1(\beta + \gamma) \quad (\text{G.15})$$

E se f_1 e f_2 verificarem as equações G.5, G.6 e G.7, a desigualdade é estrita (o que significa que neste caso a função objectivo melhora realmente se as encomendas forem colocadas por EDD).

$$f_2(\alpha) + f_1(\gamma) < f_2(\alpha + \beta) + f_1(\beta + \gamma) \quad (\text{G.16})$$

Caso 4

$$d_1 \leq t_1 \leq d_2 \leq t_2$$



Para que a colocação por EDD minimize a função objectivo, é necessário que:

$$f_2(t_1 - d_1) + f_2(t_2 - d_2) \leq f_2(t_2 - d_1) + f_1(d_2 - t_1) \quad (\text{G.17})$$

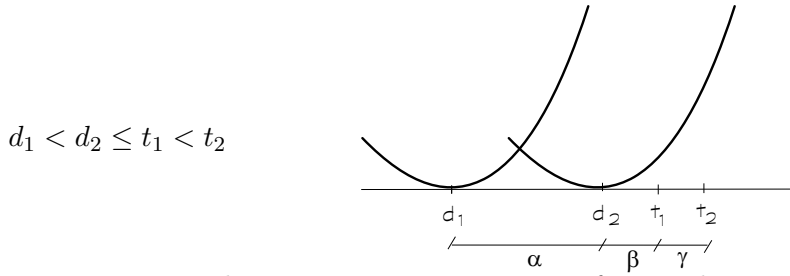
O que é equivalente a:

$$f_2(\alpha) + f_1(\gamma) \leq f_2(\alpha + \beta + \gamma) + f_1(\beta) \quad (\text{G.18})$$

E se f_1 e f_2 verificarem as equações G.5, G.6 e G.7, a desigualdade é estrita (o que significa que neste caso a função objectivo melhora realmente se as encomendas forem colocadas por EDD).

$$f_2(\alpha) + f_1(\gamma) < f_2(\alpha + \beta + \gamma) + f_1(\beta) \quad (\text{G.19})$$

Caso 5



Para que a colocação por EDD minimize a função objectivo, é necessário que:

$$f_2(t_1 - d_1) + f_2(t_2 - d_2) \leq f_2(t_2 - d_1) + f_2(t_1 - d_2) \quad (\text{G.20})$$

O que é equivalente a:

$$f_2(\alpha + \beta) + f_2(\beta + \gamma) \leq f_2(\alpha + \beta + \gamma) + f_2(\beta) \quad (\text{G.21})$$

Conclusão

Dado que consideramos que é possível desenvolver f_1 e f_2 em série de Taylor, com coeficientes positivos, para demonstrar que as desigualdades expressas

pelas equações G.10 e G.21 se verificam, comecemos por verificar se o termo de ordem n de uma função polinomial, verifica essas desigualdades:

$$\begin{aligned} \beta^n + (\alpha + \beta + \gamma)^n &= \beta^n + (\alpha + \beta)^n + \gamma^n \\ &\quad + \sum_{k=1}^{n-1} \binom{k}{n} (\alpha + \beta)^k \gamma^{n-k} \end{aligned} \quad (\text{G.22})$$

$$\begin{aligned} \beta^n + (\alpha + \beta + \gamma)^n &= (\alpha + \beta)^n + (\gamma + \beta)^n \\ &\quad + \sum_{k=1}^{n-1} \binom{k}{n} (\alpha + \beta)^k \gamma^{n-k} \\ &\quad - \sum_{k=1}^{n-1} \binom{k}{n} \beta^k \gamma^{n-k} \end{aligned} \quad (\text{G.23})$$

$$\begin{aligned} \beta^n + (\alpha + \beta + \gamma)^n &= (\alpha + \beta)^n + (\gamma + \beta)^n \\ &\quad + \sum_{k=1}^{n-1} \binom{k}{n} \gamma^{n-k} ((\alpha + \beta)^k - \beta^k) \end{aligned} \quad (\text{G.24})$$

com $\alpha, \beta, \gamma \geq 0$

$$\sum_{k=1}^{n-1} \binom{k}{n} \gamma^{n-k} ((\alpha + \beta)^k - \beta^k) \geq 0 \quad (\text{G.25})$$

O que implica que:

$$\beta^n + (\alpha + \beta + \gamma)^n \geq (\alpha + \beta)^n + (\gamma + \beta)^n \quad (\text{G.26})$$

Se o termo de ordem n verifica as desigualdades, então todos os termos de uma função polinomial verificam essa desigualdade, e também verifica a desigualdade qualquer combinação linear, desde que tenha coeficientes positivos. Podemos então generalizar, dizendo que as desigualdades são válidas para qualquer função que se possa desenvolver em série de Taylor com termos positivos.

Assim, concluindo, para quaisquer duas encomendas com datas de entrega d_1 e d_2 , e $d_1 < d_2$ e quaisquer dois períodos t_1 e t_2 , com $t_1 < t_2$ e considerando ainda que a função objectivo em causa tem as características enunciadas inicialmente, se:

$$t_1 \leq d_1 \leq t_2 \leq d_2 \text{ ou}$$

$$d_1 \leq t_1 \leq t_2 \leq d_2 \text{ ou}$$

$$d_1 \leq t_1 \leq d_2 \leq t_2,$$

então a colocação da encomenda 1 no período t_1 e a colocação da encomenda 2 no período t_2 , minimiza a função objectivo. Se por outro lado:

$$t_1 < t_2 \leq d_1 < d_2 \text{ ou}$$

$$d_1 < d_2 \leq t_1 < t_2,$$

então a colocação das encomendas por EDD nunca piora o valor da função objectivo, minimizando o seu valor na maioria dos casos. $\diamond$