

Universidade do Minho

Escola de Engenharia

Departamento de Produção e Sistemas



## **GESTÃO DE PROJECTOS**

# **Alocação adaptativa de recursos em redes de actividades multimodais**

Anabela Pereira Tereso

Tese de Doutoramento

2002



**Escola de Engenharia**



**Dep. Produção e Sistemas**

## **GESTÃO DE PROJECTOS**

# **Alocação adaptativa de recursos em redes de actividades multimodais**

Tese submetida à Universidade do Minho para obtenção do grau de  
Doutor em Produção e Sistemas, elaborada sob a orientação da  
Professora Doutora Maria Madalena Teixeira Araújo

Anabela Pereira Tereso



Junho, 2002

# Resumo

Nas últimas décadas a *gestão de projectos* tem-se revelado um instrumento poderoso que permite às organizações enfrentar novos desafios, como a competitividade dos mercados globais e a conseqüente necessidade da diminuição das margens de lucro. Com o aumento do tamanho e da complexidade dos projectos, o seu planeamento e controlo tornou-se uma das mais importantes tarefas de gestão. No processo de planeamento do projecto, os gestores têm de lidar com a *incerteza* e necessitam manipular a *alocação* dos seus recursos *adaptativamente*, por forma a alcançar os seus objectivos. Contudo, tratamentos do bem conhecido problema de alocação de recursos a projectos têm sido determinísticos e estáticos, e tratam na sua maioria actividades unimodais. Nesta tese é apresentada uma abordagem à *alocação de recursos* sob *condições estocásticas*, para *redes de actividades multimodais*. A optimização é feita utilizando *programação dinâmica*, que demonstrou ser, como se esperava, exigente em termos computacionais. Neste sentido foram analisados esquemas de aproximação que não se afastam significativamente do óptimo, mas são mais modestos em termos computacionais. É também estudada a aplicação de uma técnica de optimização global baseada numa modificação do *algoritmo “Electromagnético”* de Birbil e Fang [BirbilFang00].



# Abstract

In the last decades, *project management* has revealed it self a powerful tool, which allows organizations to face new challenges, like the competitiveness of global markets and the consequent necessity of decreasing profit margins. With the increase of the project size and complexity, planning and control has become one of the most important management tasks. In practice, project managers must cope with *uncertainty*, and must manipulate the *allocation* of their resources *adaptively* in order to achieve their ultimate objectives. Yet, treatments of the well-known ‘resource constrained project scheduling problem’ have been deterministic and static, and have addressed mostly unimodal activities. We present an approach to *resource allocation* under *stochastic conditions* for *multimodal activity networks*. Optimization is via *dynamic programming*, which proves to be demanding computationally, as expected. We investigate approximation schemes that do not detract significantly from optimality, but are modest in their computational requirements. We also investigate the application of a global optimization technique based on a modification of the “*Electromagnetic*” *algorithm* of Birbil and Fang [BirbilFang00].



# Dedicatória

Dedico esta tese,  
ao João, à Joana e à Sofia.





# Agradecimentos

Esta dissertação, apesar de resultar de um trabalho individual, só foi possível graças ao empenho e à colaboração de várias pessoas e instituições. Quero expressar aqui o meu reconhecimento e a minha gratidão, a algumas pessoas em especial.

À Dr<sup>a</sup> Madalena Araújo, minha orientadora, pelo precioso apoio científico no delinear dos objectivos e no suporte à execução de todo o trabalho, e pelo seu constante incentivo.

Ao Prof. Elmaghraby, reconhecido especialista na área, pela sua colaboração na definição e apoio à resolução do problema específico, e pelo seu carinho, especialmente durante a minha estadia nos EUA, em Março de 2001, onde tive oportunidade de trabalhar com ele diariamente, o que permitiu um avanço significativo na minha investigação. Agradeço também a sua esposa, Amina, pela forma calorosa com que me recebeu.

A Ilker Birbil agradeço a disponibilização do seu algoritmo, que foi aplicado ao problema estudado.

Quero também agradecer ao Dr. José Carlos Costa, do Departamento de Matemática, pela sua ajuda na resolução de alguns problemas de cálculo mais complexos.

Agradeço aos meus colegas do Departamento de Produção e Sistemas, em especial à Anabela Alves, pela sua leitura atenta e sugestões, ao Lino Costa, à Ana Maria Rocha e ao Ismael Vaz, pelo seu apoio na utilização de software específico.

Ao programa PRODEP e à Universidade do Minho agradeço pelo apoio

financeiro proporcionado e pela dispensa do serviço docente.

À Fundação Calouste Gulbenkian e à Fundação Luso-Americana para o Desenvolvimento, agradeço as bolsas concedidas, que me permitiram a deslocação aos EUA.

A todos os meus amigos e à minha família, agradeço pelo apoio permanente e incondicional.

# Abreviaturas

<i>A(PS)</i>	Available jobs of Partial Schedule
<i>ACTIM</i>	Activity Control TIME
<i>AF</i>	Activities to Fix
<i>ALP</i>	Activities on the Longest Path
<i>AoA</i>	Activity-on-Arc
<i>AoN</i>	Activity-on-Node
<i>CB</i>	Capacity Bound
<i>CI</i>	Criticality Index
<i>CII</i>	Cutset Intersection Index
<i>CPB</i>	Critical Path Bound
<i>CPM</i>	Critical Path Method
<i>CSB</i>	Critical Sequence Bound
<i>CT</i>	Completion Time
<i>CTE</i>	Custo Total Esperado
<i>DFS</i>	Depth-First Search strategy
<i>EF</i>	Earliest Finish
<i>EFT</i>	Earliest Finishing Time
<i>EM</i>	Electro Magnético
<i>ES</i>	Earliest Start

<i>EST</i>	Earliest Starting Time
<i>ET</i>	Early event Time
<i>GERT</i>	Graphical Evaluation and Review Technique
<i>GRD</i>	Greatest Resource Demand
<i>I&amp;D</i>	Investigação e Desenvolvimento
<i>IS</i>	Intersection Set
<i>LF</i>	Latest Finish
<i>LFT</i>	Latest Finishing Time
<i>LPT</i>	Latest Processing Time
<i>LS</i>	Latest Start
<i>LST</i>	Latest Starting Time
<i>LT</i>	Late event Time
<i>MBD</i>	Method of Bounding Distribution
<i>MES</i>	Minimum over certain ES strategies
<i>MLB</i>	Minimal Lower Bound search
<i>MSL</i>	Minimum SLack time
<i>NLP</i>	Nodes on the Longest Path
<i>PD</i>	Programação Dinâmica
<i>PERT</i>	Program Evaluation and Review Technique
<i>PLANET</i>	Project Length ANalysis and Evaluation Technique
<i>PLC</i>	Project Life Cycle
<i>PNL</i>	Programação Não Linear
<i>PS</i>	Partial Schedule
<i>PSS</i>	Parallel Scheduling Scheme
<i>RCPSP</i>	Resource Constraint Project Scheduling Problem

<i>SPP</i>	Set Partitioning Problem
<i>SPT</i>	Shortest Processing Time
<i>SS</i>	Scheduling Start
<i>SSS</i>	Serial Scheduling Scheme
<i>TCPSP</i>	Time-Constrained Project Scheduling Problem
<i>udc</i>	Uniformly Directed Cutset
<i>v.a.</i>	Variável Aleatória
<i>WBS</i>	Work Breakdown Structure
<i>WRUP</i>	Weighted Resource Utilization and Precedence
$\mathcal{E}$	Expected Value



# Conteúdo

<b>Resumo</b>	<b>i</b>
<b>Abstract</b>	<b>iii</b>
<b>Dedicatória</b>	<b>v</b>
<b>Agradecimentos</b>	<b>vii</b>
<b>Abreviaturas</b>	<b>ix</b>
<b>1 Introdução</b>	<b>1</b>
1.1 O que é um projecto . . . . .	1
1.2 Ciclo de vida de um projecto (PLC) . . . . .	4
1.3 Objectivos . . . . .	6
1.4 Organização da tese . . . . .	8
<b>2 Revisão bibliográfica</b>	<b>9</b>
2.1 Planeamento do projecto . . . . .	9
2.1.1 Estruturação . . . . .	10
2.1.2 Escalonamento (Schedule) . . . . .	11
2.1.3 Atribuição de recursos às actividades . . . . .	17
2.2 Alocação de recursos . . . . .	20
2.2.1 Detecção de conflitos na alocação de recursos . . . . .	20
2.2.2 Escalonamento com restrições de recursos . . . . .	22

2.3	RCPSP por optimização matemática . . . . .	23
2.4	RCPSP - Métodos de determinação de limites inferiores . . . . .	26
2.4.1	‘Critical Path Bound’ (CPB) . . . . .	26
2.4.2	‘Capacity Bound’ (CB) . . . . .	26
2.4.3	‘Critical Sequence Bound’ (CSB) . . . . .	26
2.4.4	Outros limites inferiores . . . . .	28
2.5	RCPSP - Procedimentos heurísticos . . . . .	29
2.5.1	Métodos construtivos . . . . .	29
2.5.2	Métodos de melhoramento . . . . .	34
2.6	RCPSP - Procedimentos exactos . . . . .	36
2.6.1	Procedimentos ‘branch and bound’ . . . . .	37
2.6.2	Procedimentos de programação inteira . . . . .	39
2.6.3	Procedimentos de programação dinâmica . . . . .	42
2.7	Redes estocásticas . . . . .	49
2.8	Apresentação do problema estudado . . . . .	57
<b>3</b>	<b>O modelo de programação dinâmica</b>	<b>61</b>
3.1	Apresentação do modelo . . . . .	61
3.2	UDC’s / CII . . . . .	65
3.2.1	Identificação dos udc’s . . . . .	65
3.2.2	Determinação do cii . . . . .	67
3.3	Projecto exemplo . . . . .	69
3.4	Análise de sensibilidade . . . . .	75
3.5	Ilustração de cenários . . . . .	79
<b>4</b>	<b>Implementação do modelo</b>	<b>87</b>
4.1	Versão 1 . . . . .	87
4.2	Versão 2 . . . . .	99
4.2.1	Exemplos . . . . .	105



<b>5</b>	<b>Aproximações ao modelo proposto</b>	<b>113</b>
5.1	Aproximação 1 . . . . .	113
5.1.1	Versão 1 . . . . .	113
5.1.2	Versão 2 . . . . .	115
5.2	Aproximação 2 . . . . .	116
5.2.1	Versão 1 . . . . .	116
5.2.2	Versão 2 . . . . .	117
5.3	Aproximação 3 . . . . .	118
5.3.1	Versão 1 . . . . .	118
5.3.2	Versão 2 . . . . .	120
5.4	Comparação dos resultados . . . . .	122
5.4.1	Versão 1 . . . . .	122
5.4.2	Versão 2 . . . . .	126
<b>6</b>	<b>Aplicação do algoritmo “Electromagnético”</b>	<b>131</b>
6.1	Algoritmo EM . . . . .	131
6.2	Aplicação do algoritmo EM ao RCPSP . . . . .	133
6.2.1	Versão 1 . . . . .	133
6.2.2	Versão 2 . . . . .	136
6.3	Aspectos da implementação . . . . .	138
<b>7</b>	<b>Conclusões e Trabalho Futuro</b>	<b>143</b>
7.1	Conclusões . . . . .	143
7.2	Trabalho Futuro . . . . .	153
<b>A</b>	<b>Versão 1</b>	<b>179</b>
A.1	mp.m . . . . .	179
A.2	mediumprojects1.m . . . . .	184
A.3	mediumprojects2.m . . . . .	185
A.4	mediumprojects3.m . . . . .	187

A.5	mediumprojects4.m . . . . .	189
A.6	generateW.m . . . . .	190
A.7	expectedValue.m . . . . .	190
A.8	calculateTvalues.m . . . . .	191
A.9	calculateMax.m . . . . .	191
A.10	calculatePmax.m . . . . .	192
A.11	cumProb.m . . . . .	192
A.12	cum.m . . . . .	192
A.13	reduce.m . . . . .	193
A.14	mass.m . . . . .	193
<b>B</b>	<b>Versão 2</b>	<b>195</b>
B.1	dp.m . . . . .	195
B.2	inputNetwork.m . . . . .	196
B.3	decisionVars.m . . . . .	197
B.4	sumNodesLen.m . . . . .	197
B.5	originNodes.m . . . . .	198
B.6	nodesLP.m . . . . .	198
B.7	activitiesLP.m . . . . .	199
B.8	generateWs.m . . . . .	199
B.9	generateW.m . . . . .	199
B.10	generateExpWs.m . . . . .	200
B.11	durations.m . . . . .	200
B.12	generateTlimits.m . . . . .	200
B.13	generateTvalues.m . . . . .	201
B.14	calculateTvalues.m . . . . .	201
B.15	activitiesToFix.m . . . . .	202
B.16	xFixedActivities.m . . . . .	202
B.17	calculateXs.m . . . . .	203

B.18	generateMainCode.m . . . . .	203
B.19	generateDps1Code.m . . . . .	207
B.20	generateDpsNCode.m . . . . .	210
B.21	reduce.m . . . . .	213
B.22	calculateResourceCost.m . . . . .	214
B.23	cum.m . . . . .	214
B.24	calculateMaxN.m . . . . .	214
B.25	calculateMax.m . . . . .	215
B.26	calculatePmax.m . . . . .	216
B.27	cumProb.m . . . . .	216
B.28	mass.m . . . . .	216
B.29	expectedValue.m . . . . .	217
<b>C</b>	<b>Código gerado pela versão 2</b>	<b>219</b>
C.1	main.m . . . . .	219
C.2	dps1.m . . . . .	221
C.3	dps2.m . . . . .	222
C.4	dps3.m . . . . .	223
C.5	dps4.m . . . . .	225
<b>D</b>	<b>Resultados do programa</b>	<b>227</b>
D.1	Procura do óptimo para as várias combinações de $\mathcal{F}$ . . . . .	227
D.2	Política óptima para o estágio 1 . . . . .	276
D.3	Política óptima para o estágio 2 . . . . .	280
D.4	Política óptima para o estágio 3 . . . . .	281
D.5	Política óptima para o estágio 4 . . . . .	281
<b>E</b>	<b>Aproximação 1 (Versão 1)</b>	<b>283</b>
E.1	mp.m . . . . .	283
E.2	mediumprojects1.m . . . . .	288

E.3	mediumprojects2.m . . . . .	289
E.4	mediumprojects3.m . . . . .	291
E.5	mediumprojects4.m . . . . .	292
<b>F</b>	<b>Aproximação 1 (Versão 2)</b>	<b>295</b>
F.1	dp.m . . . . .	295
F.2	generateDps1Code.m . . . . .	296
F.3	generateDpsNCode.m . . . . .	299
F.4	generateWs.m . . . . .	302
F.5	durations.m . . . . .	303
F.6	calculateMaxN.m . . . . .	303
<b>G</b>	<b>Aproximação 2 (Versão 1)</b>	<b>307</b>
G.1	mp.m . . . . .	307
G.2	mediumprojects1.m . . . . .	312
G.3	mediumprojects2.m . . . . .	313
G.4	mediumprojects3.m . . . . .	315
G.5	mediumprojects4.m . . . . .	316
<b>H</b>	<b>Aproximação 2 (Versão 2)</b>	<b>317</b>
H.1	dp.m . . . . .	317
H.2	generateDps1Code.m . . . . .	318
H.3	generateDpsNCode.m . . . . .	321
H.4	durations.m . . . . .	323
H.5	calculateMaxN.m . . . . .	324
<b>I</b>	<b>Algoritmo “Electromagnético”</b>	<b>325</b>
I.1	EM . . . . .	325
I.2	Initialize . . . . .	325
I.3	Local . . . . .	326
I.4	CalcF . . . . .	327

I.5	Move . . . . .	327
<b>J</b>	<b>Algoritmo EM - RCPSP</b>	<b>329</b>
J.1	GOfunc.m . . . . .	329
J.2	EMVIS.m . . . . .	330
J.3	SetStruct.m . . . . .	332
J.4	Initialize.m . . . . .	334
J.5	TotalCost.m . . . . .	335
J.6	Cpm.m . . . . .	335
J.7	OriginNodes.m . . . . .	336
J.8	CalcF.m . . . . .	336
J.9	Move.m . . . . .	337
J.10	Local.m . . . . .	338
J.11	MeanRes.m . . . . .	340



# Lista de Figuras

1.1	Ciclo de vida do projecto (PLC) [Klein00] . . . . .	4
1.2	O processo de gestão de projectos [Klein00] . . . . .	7
2.1	WBS do projecto de uma casa . . . . .	11
2.2	Representação AoN . . . . .	12
2.3	Representação AoA . . . . .	13
2.4	Forma de colocação dos tempos para o método CPM (AoA) . . . . .	13
2.5	Projecto exemplo (AoN) . . . . .	16
2.6	Projecto exemplo (AoA) . . . . .	16
2.7	Gráfico de Gantt para o projecto exemplo . . . . .	18
2.8	Projecto exemplo com informação sobre necessidade de recursos . . . . .	21
2.9	Perfil de utilização de recursos para o projecto exemplo . . . . .	22
2.10	Gráfico de Gantt orientado à capacidade para o projecto exemplo . . . . .	22
2.11	Escalonamento sequencial das actividades do caminho crítico para o projecto exemplo . . . . .	27
2.12	Solução obtida usando o Procedimento de Lang . . . . .	32
2.13	Solução obtida usando SSS com a regra SPT para o projecto exemplo . . . . .	33
2.14	Árvore exemplo . . . . .	39
2.15	Rede de actividades AOA . . . . .	47
3.1	Projecto exemplo com os seus udc's . . . . .	66
3.2	Projecto exemplo . . . . .	69
3.3	Caminho mais longo . . . . .	71

3.4	Estado do projecto no instante $t=0$ . . . . .	80
3.5	Estado do projecto no instante $t=10$ . . . . .	81
3.6	Estado do projecto no instante $t=18$ . . . . .	82
3.7	Estado do projecto no instante $t=30$ . . . . .	82
3.8	Estado do projecto no instante $t=34$ . . . . .	83
3.9	Estado do projecto no instante $t=46$ . . . . .	83
3.10	Estado do projecto no instante $t=54$ . . . . .	84
3.11	Estado do projecto no instante $t=58$ . . . . .	84
3.12	Estado do projecto no instante $t=66$ (Fim) . . . . .	85
4.1	$p(t) = 0.1 * e^{-0.1*t}$ . . . . .	88
4.2	Distribuição exponencial dividida em quatro áreas iguais . . . . .	89
4.3	Rede 1 . . . . .	105
4.4	Rede 2 . . . . .	106
4.5	Rede 3 . . . . .	107
4.6	Rede 4 . . . . .	109
4.7	Rede 5 . . . . .	110
5.1	Projecto Exemplo (Solver do Excel) . . . . .	121
5.2	Rede 1 (Solver do Excel) . . . . .	121
5.3	Rede 2 (Solver do Excel) . . . . .	121
5.4	Rede 3 (Solver do Excel) . . . . .	122
5.5	Rede 4 (Solver do Excel) . . . . .	122
5.6	Rede 5 (Solver do Excel) . . . . .	123
6.1	$F(t) = 1 - e^{-\lambda t}$ . . . . .	139
7.1	Rede original . . . . .	154
7.2	Rede série/paralelo . . . . .	154
7.3	Rede reduzida . . . . .	155



# Lista de Tabelas

2.1	Resultados da aplicação do método CPM ao projecto exemplo . .	17
2.2	Utilização do recurso 1 nas actividades do projecto exemplo . . .	21
2.3	Regras de prioridade . . . . .	30
2.4	Ordem de visita das estratégias DFS e MLB . . . . .	38
3.1	Geração dos udc's do projecto da fig. 2.1 . . . . .	66
3.2	Parâmetros da distribuição exponencial para todas as actividades do projecto . . . . .	70
3.3	Variação do CTE para diferentes valores das variáveis fixas . . . .	77
3.4	Variação do CTE para diferentes valores de $x_2$ . . . . .	78
3.5	Variação do CTE para diferentes valores de $x_6$ . . . . .	78
3.6	Variação do CTE para diferentes valores de $x_8$ . . . . .	78
3.7	Variação do CTE para diferentes valores de $x_{10}$ . . . . .	79
4.1	Valores de $W$ gerados pelo programa, para as várias actividades .	92
4.2	Tempos de realização dos nós da rede exemplo . . . . .	94
4.3	Máximo entre duas variáveis aleatórias . . . . .	96
4.4	Probabilidade do máximo entre duas variáveis aleatórias . . . . .	97
4.5	Probabilidade do máximo entre duas variáveis aleatórias (2) . . .	97
4.6	Net - Estrutura que armazena os dados do projecto . . . . .	100
4.7	Parâmetros da rede 1 . . . . .	106
4.8	Parâmetros da rede 2 . . . . .	107
4.9	Parâmetros da rede 3 . . . . .	108

4.10	Parâmetros da rede 4 . . . . .	109
4.11	Parâmetros da rede 5 . . . . .	111
5.1	Comparação de instruções para as aproximações 1 e 2 . . . . .	118
5.2	Comparação de resultados para as aproximações 1, 2 e 3 . . . . .	124
5.3	Comparação de resultados para a rede 1 . . . . .	126
5.4	Comparação de resultados para a rede 2 . . . . .	126
5.5	Comparação de resultados para a rede 3 . . . . .	127
5.6	Comparação de resultados para a rede 4 . . . . .	128
5.7	Comparação de resultados para a rede 5 . . . . .	129
7.1	Comparação dos resultados obtidos (projecto exemplo) . . . . .	147
7.2	Comparação dos resultados obtidos (rede 1) . . . . .	147
7.3	Comparação dos resultados obtidos (rede 2) . . . . .	148
7.4	Comparação dos resultados obtidos (rede 3) . . . . .	148
7.5	Comparação dos resultados obtidos (rede 4) . . . . .	149
7.6	Comparação dos resultados obtidos (rede 5) . . . . .	150
7.7	Quadro resumo dos principais resultados . . . . .	151

# Capítulo 1

## Introdução

A gestão de projectos tem vindo a consolidar-se como uma actividade fundamental em qualquer organização, quer os seus objectivos sejam económicos, financeiros, sociais ou políticos. As partes integrantes de qualquer projecto, nomeadamente os recursos humanos e/ou materiais, o custo, o tempo disponível e as actividades a desenvolver, necessitam de ser geridas globalmente da forma mais eficiente e eficaz possível. Muita investigação tem, portanto, sido dedicada a tão importante missão. Este trabalho é mais um contributo para o aprofundamento do conhecimento da gestão de projectos.

Vamos nesta introdução apresentar resumidamente várias definições do termo “projecto” e detalhar as suas fases. Procurar-se-á preparar a introdução dos objectivos da presente investigação. Por último apresenta-se a estrutura da tese.

### 1.1 O que é um projecto

Em termos gerais, um projecto pode ser definido como uma actividade que é realizada uma só vez, tendo objectivos bem definidos, com limitações nos recursos e no tempo disponível para a sua realização.

Consultando a literatura, podemos encontrar uma grande variedade de definições, caracterizações e condições, algumas das quais são apresentadas a seguir,

para permitir ter uma perspectiva mais alargada desta questão (confirmar refs. [MeredithMantel95], [PMI00], [Spinner97], [Shtub*et.al*94], [Turner92]).

- Um projecto é um empreendimento temporário realizado para criar um produto ou serviço únicos. Temporário significa que tem um início e um fim definidos, terminando quando os seus objectivos forem alcançados. Único significa que o produto ou serviço é diferente de alguma forma de todos os outros produtos ou serviços similares.
- Um projecto representa uma actividade que é realizada uma só vez, com um conjunto bem definido de objectivos. Para alcançar os objectivos têm que se levar a cabo um certo número de tarefas. A duração do projecto é restrita, i.e. o projecto termina quando se alcançam os objectivos definidos.
- Um projecto é um empreendimento no qual estão organizados recursos humanos, materiais e financeiros, para realizar um determinado trabalho, com uma determinada especificação, com restrições de custo e tempo, de forma a conseguir alterações benéficas, através da obtenção de objectivos quantitativos e qualitativos.
- Um projecto é único no sentido em que possui algumas características que o impedem de se tornar uma rotina. A complexidade de um projecto é tal que requiere uma cuidada coordenação para executar as sub-actividades.
- Os recursos disponíveis para a realização de um projecto são restritos. Estes recursos são essencialmente o capital, os recursos humanos e o equipamento.
- A execução de um projecto requiere a colaboração de vários departamentos funcionais de uma organização e mesmo de diferentes organizações. Um projecto pode interagir com outros projectos paralelos.
- A execução de um projecto pode envolver um grau considerável de incerteza. As principais fontes de incerteza incluem variação no desempenho

dos recursos, dados inadequados ou incorrectos, e a incapacidade de prever satisfatoriamente devido à falta de experiência anterior.

Estas definições permitem-nos concluir que a execução de um projecto é uma tarefa realmente complexa, sendo portanto necessária uma grande preocupação com uma correcta e eficiente gestão do projecto.

Para colmatar qualquer dúvida relacionada com o que é realmente um projecto, são apresentados, de seguida, alguns exemplos. Começando pelos tempos mais remotos, a construção das pirâmides do Egipto terá sido, segundo algumas opiniões, o primeiro grande projecto de que temos conhecimento (ano 2650 a.c.) [Lewis95]. Através da história pudemos testemunhar o desenvolvimento de novas ideias e produtos, que marcaram para sempre a nossa civilização, como por exemplo, o fogo, a roda, a luz eléctrica, o automóvel, o avião, a bomba atómica e outros. Nos dias de hoje, são inúmeros os projectos de investigação e desenvolvimento levados a cabo por cientistas de todo o mundo, em áreas como estudos médicos, ambientais, espaciais e militares, sendo esta última, sem dúvida, a grande impulsionadora do desenvolvimento das técnicas de gestão de projectos. Os projectos da área da construção civil usam também extensivamente estas técnicas. Assim, e resumindo, um projecto tanto pode ser desenvolver um novo produto ou serviço, como desenhar um novo veículo de transporte, ou construir um edifício ou uma fábrica, mas também efectuar uma alteração na estrutura de uma organização, desenvolver ou adquirir um novo sistema de informação, desenvolver uma campanha publicitária ou implementar um novo procedimento ou processo negocial.

A enumeração de exemplos poderia continuar indefinidamente, considerando todos os projectos já executados e os novos possíveis projectos, tendo como limite a imaginação!

## 1.2 Ciclo de vida de um projecto (PLC)

Uma vez que um projecto tem um início e um fim, podemos associar-lhe um ciclo de vida, normalmente designado por PLC - ‘Project Life Cycle’. Durante a vida de um projecto há várias fases de desenvolvimento, que são idênticas à maioria dos projectos. Perceber estas fases ajuda os gestores a melhor controlar os recursos, os tempos e as performances, para alcançar mais eficientemente os objectivos pretendidos. As fases de um projecto são tipicamente as que se podem ver na figura 1.1 ([AngusGunderson97], [Klein00], [Lewis97], [RuskinEstes95]).

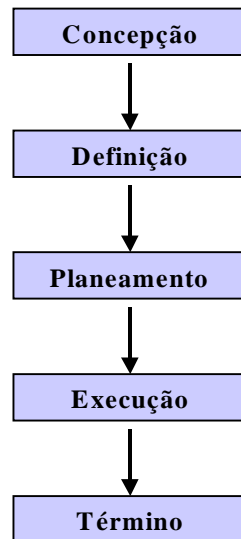


Figura 1.1: Ciclo de vida do projecto (PLC) [Klein00]

É na *fase de concepção*, que a ideia do projecto é delineada. Numa empresa a ideia pode surgir pela identificação de uma necessidade interna ou por um pedido de um cliente. Essa ideia é normalmente traduzida pela apresentação de um objectivo a ser atingido, por mais vago que ele seja. O objectivo pode ser posto da seguinte forma: “arranjar uma forma de transportar 50000 pessoas/dia” ou “pretende-se construir uma rede de metro com capacidade para 100000 viagens/dia”. O primeiro tipo de objectivo apesar de ser mais vago, deixa mais possibilidades de solução, o que pode ser vantajoso. Nesta fase não existe

portanto uma definição bem clara do que se pretende fazer, daí a necessidade de se proceder a uma análise de viabilidade do projecto, conjuntamente com uma análise económica e uma análise de risco. O resultado destas análises irá permitir decidir sobre a implementação do projecto.

É na *fase de definição*, que os objectivos devem ser mais detalhados, especificando claramente os requisitos, bem como a abordagem global que irá ser usada para alcançar os objectivos. É também nesta fase que se define a organização das pessoas envolvidas no projecto, com a necessária nomeação de um gestor do projecto.

Na *fase de planeamento* são desenvolvidos planos detalhados, com identificação das tarefas necessárias e respectivas durações. É também definido o custo e os recursos necessários para levar a cabo cada tarefa ou actividade. Para além das características de cada actividade, são também estabelecidas as relações de precedência entre as actividades. Com todas estas informações, é então possível construir um plano de execução do projecto (project schedule), o que nem sempre é tarefa fácil, devido às inúmeras restrições de precedências, recursos e custos, que por vezes existem. Nesta fase são também estabelecidas metas ou marcos, que irão servir para controlar a execução do projecto. É na fase de planeamento que se enquadra este trabalho de investigação, mais concretamente na temática da alocação de recursos escassos às actividades de um projecto, como veremos mais adiante.

A *fase de execução* consiste basicamente em fazer o trabalho e comunicar resultados. A execução do trabalho deve seguir o plano estabelecido, sendo necessário garantir isso mesmo, através de uma monitorização constante que esteja atenta às metas predefinidas, aos custos, bem como à qualidade do produto que vai sendo obtido. Caso haja um desvio do plano actual, a informação disponível irá ser usada para modificar o plano de forma a repor o projecto no seu curso normal. Se o desvio for muito grande, isto pode não ser possível. As consequências, neste caso, podem ser um atraso na conclusão do projecto, um aumento do

custo global e nos casos mais graves pode ser necessário abortar a execução do projecto, o que não é de todo desejável, pois já terão sido comprometidos muitos recursos humanos e materiais.

Uma vez alcançados os objectivos do projecto, entra-se na *fase de término*, que deve ser usada para avaliar o decurso do projecto. A informação mais importante a reter será a duração real do projecto, o custo das actividades, a utilização dos recursos e também o grau de satisfação do cliente perante o resultado final. Esta informação deve ser armazenada para servir de futura referência a outros projectos. Finalmente o projecto tem que ser desmantelado, repondo o equipamento e as instalações num estado apropriado à próxima utilização, e redistribuindo o pessoal conforme necessário.

Para concluir este ponto e ter uma visão geral das principais tarefas de gestão inerentes a cada fase, apresenta-se a figura 1.2, que permite estruturar mentalmente todo o processo.

### 1.3 Objectivos

A presente investigação tem por objectivo modelar o problema da alocação de recursos, sob condições estocásticas, para redes de actividades multimodais<sup>1</sup>. Ou seja, considerando que um projecto é representado por uma rede de actividades, na qual estão implícitas as restrições de precedências, pretende-se determinar qual a quantidade de recursos óptima, a atribuir a cada actividade, por forma a terminar o projecto com o mínimo custo possível. Esta alocação deverá ter em conta as restrições de recursos, o custo dos mesmos, e o custo de eventuais atrasos. Este problema é normalmente conhecido na literatura por RCPSP - ‘Resource Constraint Project Scheduling Problem’.

As técnicas que irão ser utilizadas vão centrar-se na programação

---

<sup>1</sup>Numa rede multimodal, a duração de uma actividade vai depender da quantidade de recursos a ela atribuída.



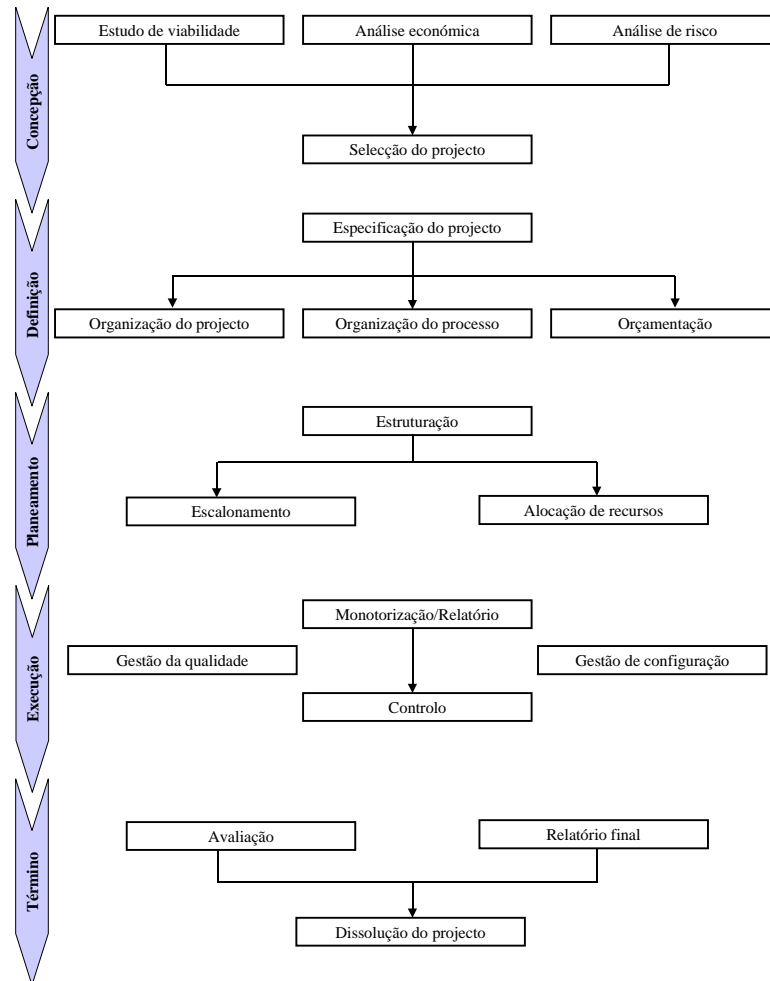


Figura 1.2: O processo de gestão de projectos [Klein00]

dinâmica (PD) aplicada a redes de actividades. Serão eventualmente desenvolvidas aproximações ao modelo de PD, se, como se espera, este modelo for muito exigente em termos computacionais.

Pensa-se vir a comparar os resultados obtidos, pela modelação desenvolvida, com outros derivados do recurso a técnicas como a Simulação de Monte Carlo, o cálculo CPM, e a programação não linear (PNL).

Será também estudada a possibilidade da aplicação de uma nova técnica de optimização global, ao problema em estudo, que tem por nome algoritmo “Electromagnético” [BirbilFang00].

## 1.4 Organização da tese

Neste capítulo faz-se uma apresentação dos aspectos mais importantes da gestão de projectos, começando por definir o que é um projecto e qual o seu ciclo de vida. É assim preparado o enquadramento para a apresentação concreta dos objectivos da investigação aqui relatada.

A restante tese vai estar organizada da seguinte forma. No capítulo 2 é feito o estudo do “estado da arte”. São destacados os aspectos mais relevantes do planeamento de projectos. É dada uma particular atenção à questão da alocação de recursos às actividades. Este capítulo pretende ser uma base teórica para o problema que vai ser tratado nesta tese. Na última secção é apresentado o problema detalhadamente. Serão feitas referências aos principais autores nesta área. No capítulo 3 será apresentado o modelo de PD proposto para resolver o problema, bem como uma ilustração do modelo para uma rede de actividades exemplo. Será também feita uma análise de sensibilidade aos resultados obtidos e uma ilustração de um possível cenário. No capítulo 4 serão discutidos aspectos da implementação do modelo, para duas versões desenvolvidas. Como se verificou que o esforço computacional do modelo de PD proposto é muito pesado, no capítulo 5 apresentam-se esquemas aproximativos, baseados na utilização de médias, ainda em PD e um modelo em PNL. Para testar outras abordagens que não a PD, apresenta-se no capítulo 6 a aplicação de uma técnica de optimização global, baseada no algoritmo “Electromagnético” de Birbil e Fang [BirbilFang00], em conjugação com Simulação de Monte Carlo e cálculo CPM. No capítulo 7 apresentam-se conclusões e são apontados caminhos para trabalho futuro.

No final é apresentada uma listagem detalhada de bibliografia relevante e um conjunto de apêndices, onde se apresenta o código dos vários programas desenvolvidos e os resultados obtidos.

# Capítulo 2

## Revisão bibliográfica

O tema desta tese está relacionado com a fase de planeamento. Vai-se então de seguida, fazer uma apresentação mais detalhada desta fase, remetendo o leitor interessado no conteúdo das outras fases para as referências [Klein00], [Lewis97] e [RuskinEstes95].

### 2.1 Planeamento do projecto

O planeamento é sem dúvida uma das tarefas mais importantes de qualquer gestor. Contudo, na vida real, encontramos muitos profissionais que se esquecem desta realidade. Teimam em levar avante os seus projectos sem um plano, porque pensam que fazê-lo seria uma perda de tempo. Com este comportamento, eles perdem contudo uma capacidade muito importante, a capacidade de controlar. Sem um plano é impossível haver controlo [Lewis95], uma vez que o controlo é a comparação do estado actual com o que foi planeado, tomando medidas correctivas, caso haja desvios do caminho pretendido. É precisamente quando existem mais restrições no tempo disponível para a execução do projecto, que o planeamento é mais importante.

Assim, é na fase de planeamento do projecto que se especifica em pormenor tudo o que deve ser feito, isto é, quais as actividades necessárias, e como deve

ser feito, ou seja, qual o ordenamento das actividades, que recursos necessitam, quanto tempo necessitam para ser executadas e quais os custos a elas associados.

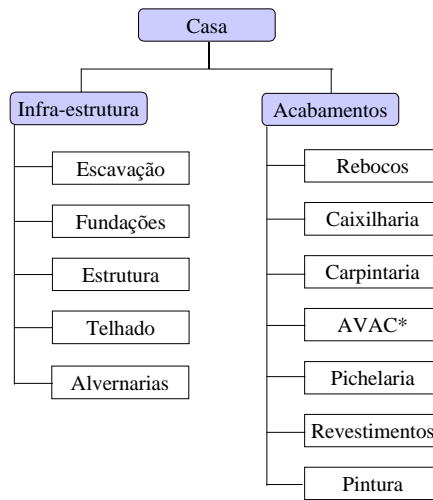
Dada a complexidade destas tarefas, elas são normalmente feitas por fases. Numa primeira fase, o projecto é estruturado, ou seja, dividido em actividades. De seguida, essas actividades são ordenadas, tendo em atenção problemas de precedências (escalonamento). No final são analisadas as questões de atribuição de recursos às actividades. Vamos, de seguida, ver estas fases em mais pormenor.

### 2.1.1 Estruturação

A primeira etapa do processo de planeamento é a divisão do projecto em partes lógicas, tentando associar a cada parte as respectivas actividades. Pode-se assim desenvolver um esquema da estrutura do projecto, designado na literatura anglo-saxónica por WBS - 'Work Breakdown Structure'. A WBS é normalmente representada como uma árvore, em que na raiz da árvore temos o projecto. No primeiro nível são identificadas as actividades principais. Nos níveis seguintes, cada actividade vai sendo dividida em sub-actividades [Burke92], [Kerzner98]. Este processo termina quando não houver interesse em subdividir mais. Na figura 2.1 podemos ver um exemplo simplificado duma WBS do projecto de uma casa.

A principal ideia de fazer uma WBS é identificar todas as tarefas envolvidas no projecto, sem necessariamente pensar na sua sequenciação. Esta forma de estruturar o projecto facilita a distribuição dos recursos humanos e materiais, bem como a estimativa da duração e do custo, associada a cada actividade. Permite portanto ter uma visão global de todo o projecto [Lewis97].

Normalmente, a alocação das pessoas às tarefas é feita com o auxílio de uma tabela de responsabilidades. Esta tabela terá nas linhas uma lista de todas as pessoas envolvidas no projecto e nas colunas uma lista das tarefas identificadas. As entradas na tabela poderão ser de dois tipos: 1-responsável pela tarefa, ou 2-colaborador. Sendo assim, cada tarefa terá no máximo uma pessoa responsável



\* Aquecimento, Ventilação e Ar Condicionado

Figura 2.1: WBS do projecto de uma casa

e pode ter um ou mais colaboradores. Por outro lado, cada pessoa pode ser responsável e/ou colaborador em mais do que uma tarefa [Burke92], [Lewis97], [MeredithMantel95], [Shtub*et. al*94].

Quanto às estimativas de tempos e custos elas podem ser feitas com base no bom senso, mas será preferível fundamentá-las em dados históricos, quando disponíveis.

### 2.1.2 Escalonamento (Schedule)

Depois de estar definida a estrutura do projecto, é necessário proceder ao seu escalonamento. O escalonamento de um projecto permite essencialmente encaixar as actividades no tempo, conseguindo desta forma prever datas de início e fim de cada actividade e consequentemente de todo o projecto. Para iniciar este processo, para além da estruturação do projecto em actividades (WBS) e das estimativas já referidas na secção anterior, é necessário ter em atenção as

precedências entre as actividades. Estas precedências podem ter origem em questões tecnológicas ou processuais.

Normalmente, o encadeamento temporal das actividades é representado por um diagrama em rede, com nós ligados através de arcos. Existem duas técnicas de representar estas redes:

- AoN (Activity-on-Node): segundo esta técnica, as actividades são representadas nos nós. Os arcos que unem os nós permitem representar as relações de precedência entre as actividades (ver figura 2.2). Esta técnica é mais simples, pois representa directamente as relações de precedência, sem necessidade de recurso a actividades fictícias.
- AoA (Activity-on-Arc): nesta técnica, as actividades são representadas nos arcos, representando os nós os acontecimentos de início e fim de actividade (ver figura 2.3). Esta técnica tem a vantagem de tornar mais óbvia a representação do progresso das actividades, ao longo da execução do projecto.

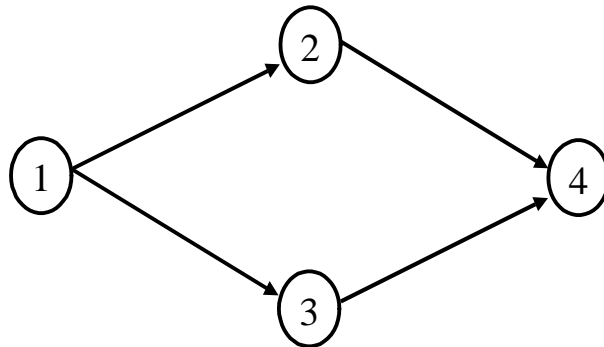


Figura 2.2: Representação AoN

Depois de construída a rede que representa o projecto, e tendo em conta a duração de cada actividade ( $d_j$ ), é usado um método, conhecido historicamente por CPM ('Critical Path Method'). Este método permite, através de um 'forward pass' e um 'backward pass', determinar os tempos de início e fim de acontecimento

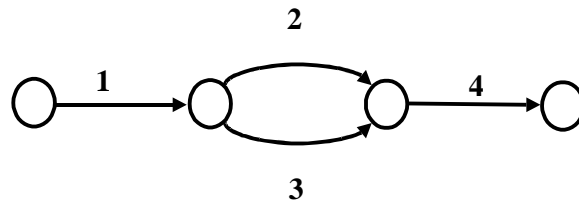


Figura 2.3: Representação AoA

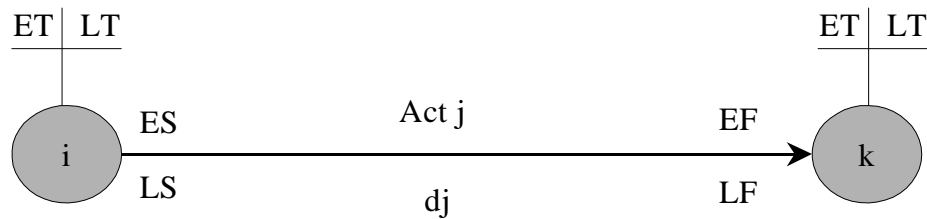


Figura 2.4: Forma de colocação dos tempos para o método CPM (AoA)

(ET - 'Early event Time', LT - 'Late event Time') e de início e fim de actividade (ES - 'Earliest Start', EF - 'Earliest Finish', LS - 'Latest Start', LF - 'Latest Finish'). Na figura 2.4 está representada a forma de colocação dos tempos, calculados por este método, no caso de uma rede AoA. No 'forward pass' completam-se os 'Early Times'. Começa-se pelo primeiro nó da rede, onde  $ET = 0$ . De seguida calcula-se  $ES = ET$  e  $EF = ES + dj$ . Nos restantes nós,  $ET = \max(EF \text{ dos nós precedentes})$ . No 'backward pass' completam-se os 'Late Times'. Começa-se pelo último nó da rede, onde  $LT = ET$ . Depois  $LF = LT$  e  $LS = LF - dj$ . Nos restantes nós, o  $LT = \min(LS \text{ dos nós sucessores})$ . No final deste processo, é possível obter as folgas existentes bem como a duração total do projecto (CT - 'Completion Time'). A folga representa a quantidade de tempo que a actividade se pode atrasar sem comprometer o fim do projecto. Existe normalmente um caminho crítico, como o nome do método indica, que é o caminho em que todas as actividades têm folga zero. Isto significa que um atraso numa destas actividades iria comprometer a data de conclusão do projecto, tendo o gestor que lhes dar

uma particular atenção [Klein00].

Quando não existem dados históricos, sendo difícil estimar com precisão, a única possibilidade é tentar estimar a duração das actividades, com base na pouca experiência existente, sendo portanto maior o risco envolvido. Os projectos de I&D são um exemplo em que pode surgir este problema. Foi precisamente para tentar reduzir o risco da estimativa que foi desenvolvido o método PERT - ‘Program Evaluation and Review Technique’, por volta de 1958, por uma firma de consultadoria, Allen & Hamilton, para ser aplicado no projecto do submarino Polaris. Segundo o método PERT, a duração das actividades é representada por uma distribuição beta. Baseia-se na definição de três valores para a duração:

$a$  - o valor optimista,

$m$  - o valor mais provável,

$b$  - o valor pessimista.

Utiliza-se a fórmula (2.1), para calcular a duração média esperada, para cada actividade  $i$  e a fórmula (2.2) para calcular o respectivo desvio padrão.

$$\mu_i = \frac{a + 4m + b}{6}, \quad (2.1)$$

$$\sigma = \frac{b - a}{6}. \quad (2.2)$$

Os valores das médias assim calculados, são usados para os cálculos PERT, que são idênticos aos do CPM.

De seguida é aplicado o teorema do limite central<sup>1</sup> para obter a distribuição de probabilidades para a duração do projecto. A duração do projecto é dada pelo somatório da duração das actividades do caminho crítico. No caso do caminho crítico ser constituído por três actividades, a média e o desvio padrão da duração do projecto podem ser calculados por,

---

<sup>1</sup>A soma de um grupo de variáveis independentes com média  $\mu_i$  e variância  $\sigma_i^2$  terá uma distribuição que se aproxima da normal, com média  $\mu = \sum \mu_i$ 's e variância  $\sigma^2 = \sum \sigma_i^2$ 's.



$$\mu_{CP} = \mu_1 + \mu_2 + \mu_3, \quad (2.3)$$

$$\sigma_{cp} = \sqrt{\sigma_1^2 + \sigma_2^2 + \sigma_3^2}. \quad (2.4)$$

Podemos então concluir que há uma probabilidade de 68% de completar o projecto em mais ou menos um desvio padrão da média, 95% dentro de dois desvios padrão e 99,74% dentro de três desvios padrão.

O facto de PERT necessitar de três estimativas de tempos para cada actividade do projecto, e serem necessários cálculos para chegar à média e ao desvio padrão, representa trabalho adicional, comparado com o método CPM, pelo que muitos gestores de projectos consideram o PERT como não merecedor do esforço adicional requerido, preferindo o CPM. Além disso, como os três valores são estimativas, pode colocar-se a questão se chegaremos a uma melhor estimativa global, conjugando três estimativas ou partindo só de uma. Para um tratamento mais profundo desta técnica, ver [BedworthBailey87], [ClelandKing83] e [Moder*et.al*83].

*Exemplo.* Para exemplificar alguns dos conceitos apresentados neste capítulo, vai ser introduzida uma rede exemplo (ver figura 2.5 e 2.6), onde  $j$  representa uma actividade e  $d_j$  a sua duração.

As actividades com duração 0 são actividades fictícias (dummy). No caso da rede AoN, são utilizadas só para representar os acontecimentos de início e fim do projecto, enquanto no caso da rede AoA são necessárias para representar relações de precedência (como na figura 2.6, a tracejado). A aplicação do método CPM permite-nos chegar aos resultados representados na figura 2.6 e na tabela 2.1. O caminho crítico é o caminho com folga 0, ou seja o caminho constituído pelas actividades 3 e 5.

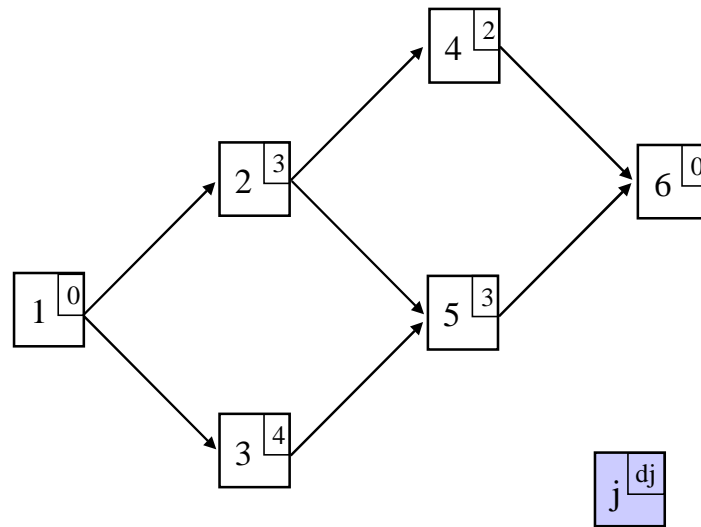


Figura 2.5: Projecto exemplo (AoN)

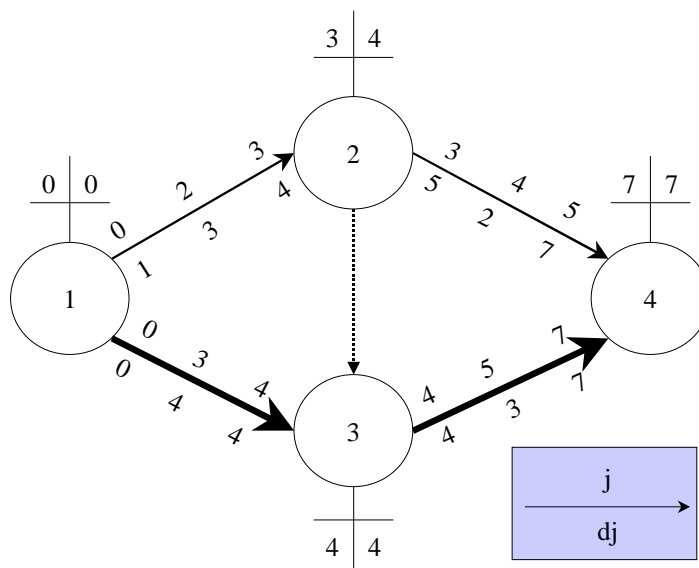


Figura 2.6: Projecto exemplo (AoA)

Depois de construída esta rede, que tem a vantagem de facilitar a análise do relacionamento entre as actividades do projecto, ela deve ser transformada num gráfico de Gantt, que irá permitir às pessoas que vão executar o projecto uma

Tabela 2.1: Resultados da aplicação do método CPM ao projecto exemplo

<b>Actividade</b>	<b>2</b>	<b>3</b>	<b>4</b>	<b>5</b>	
<b>ES</b>	0	0	3	4	
<b>EF</b>	3	4	5	7	$CT = 7$
<b>LS</b>	1	0	5	4	
<b>LF</b>	4	4	7	7	
<b>Folga Total</b>	1	0	2	0	

leitura mais fácil. O gráfico de Gantt é um tipo especial de gráfico de barras. No eixo vertical são enumeradas todas as actividades a serem executadas, e no eixo horizontal estão representados os tempos. À frente de cada actividade é representada uma barra com uma localização temporal no eixo horizontal, o que permite ver claramente quando deve iniciar, quando deve terminar e conseqüentemente a sua duração. A abordagem mais utilizada é construir o escalonamento seguindo os valores mais cedo de início e fim das actividades, o que deixa mais margem a possíveis atrasos que venham a surgir. Na figura 2.7 está representado o gráfico de Gantt, para o projecto exemplo, segundo esta abordagem. A preto estão representadas as actividades críticas. Há extensões a esta representação em que é possível incluir, no final de cada barra, a folga total disponível, além de outras informações como relações de precedência, marcos, etc. [Lee-KwangFavrel88]. Contudo, um excesso de informação pode prejudicar a sua leitura.

### 2.1.3 Atribuição de recursos às actividades

O método CPM assume que as únicas restrições existentes são as precedências entre as actividades, o que é falso na maior parte dos casos. Tipicamente, a disponibilidade de recursos é uma questão essencial a ter em conta no planeamento. De nada interessa ter um escalonamento com duas actividades que deveriam ser executadas em paralelo, se elas necessitam de um mesmo recurso não compartilhá-

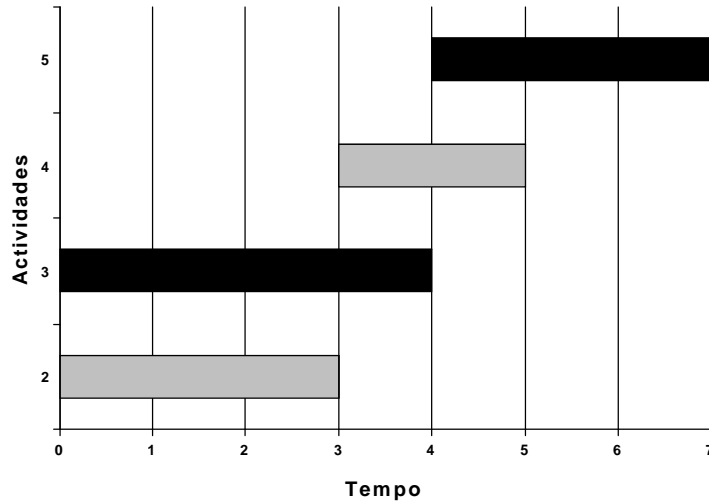


Figura 2.7: Gráfico de Gantt para o projecto exemplo

vel. O problema da disponibilidade limitada de recursos no escalonamento pode ser visto, p.e. em [JustMurphy94], [Lewis95], [LiWillis91], [MacleodPeterson96], [Willis85], [Woodworth93], e é preocupação também deste trabalho.

Quando se analisam estes problemas, os recursos são normalmente classificados em três categorias. *Recursos renováveis* são recursos que estão disponíveis em quantidades limitadas, durante todo o projecto, p.e. mão de obra e equipamento. *Recursos não renováveis* são aqueles que existem numa quantidade fixa, no início do projecto, e vão sendo gastos, à medida que o projecto avança, sem a possibilidade de serem repostos, p.e. materiais escassos. Por último, existem os *recursos mistos*, que são limitados na sua disponibilidade por período, bem como na disponibilidade total. Um exemplo deste tipo de recursos é o capital disponível para a execução do projecto.

Qualquer que seja o tipo, se a quantidade de recursos necessária excede o limite disponível, ocorre um conflito e o escalonamento previsto não pode ser executado. Existe um conjunto de técnicas básicas que podem ser aplicadas neste caso, por forma a tentar obter um escalonamento viável [Klein00]. Para uma consulta de técnicas mais elaboradas ver secções seguintes.

- Em primeiro lugar, devemos analisar a possibilidade de atrasar o início de actividades com folga, caso isso resolva o conflito de recursos. Esta é uma solução desejável, pois não vai comprometer o término do projecto, nem aumentar os custos.
- Na mesma linha da anterior, podemos também verificar se é possível reduzir a quantidade de recursos aplicada às actividades com folga, por forma a resolver o conflito, tendo em atenção que o conseqüente aumento da duração dessas actividades não ultrapasse a folga.
- Uma outra possibilidade seria tentar aplicar outro tipo de recursos, com uma menor utilização, a algumas das actividades em conflito.
- É ainda possível por vezes dividir algumas actividades em sub-actividades que não necessitem da totalidade dos recursos, e resolver assim os conflitos. Neste caso é necessário ter atenção às relações de precedência e a possíveis tempos acrescidos de preparação das sub-actividades.
- Uma última medida, mas que já implica um aumento dos custos, será aumentar a quantidade de recursos nos períodos críticos, utilizando p.e. mão de obra em horas extra ou alugando equipamentos adicionais. Este aumento deve ser tal que permita resolver o conflito, com o mínimo custo possível. Devemos ter atenção contudo em não ultrapassar o orçamento disponível.

Depois de determinar um escalonamento viável, pode-se proceder a um nivelamento dos recursos, re-arranjando as actividades dentro das suas folgas, por forma a minimizar as flutuações nas necessidades de recursos. Esta medida justifica-se pois uma utilização dos recursos mais regular pode permitir reduzir custos. Por exemplo, no caso de recursos humanos, se houver necessidade repetida de despedir, contratar e formar pessoas, os custos irão aumentar. Para os recursos materiais, uma grande flutuação irá aumentar os custos de armazenamento. Este aspecto é analisado em [AshokRajendran89], [Easa89] e [SeibertEvans91].

## 2.2 Alocação de recursos

Como já foi referido anteriormente, as restrições de precedência entre as actividades não são as únicas existentes na implementação de um projecto. Existem também as restrições orçamentais e as restrições de recursos, às quais iremos dedicar mais atenção, devido à sua importância como base teórica do trabalho desenvolvido.

O problema da alocação de recursos a projectos é classificado cientificamente como ‘Resource Constraint Project Scheduling Problem’, e tem sido amplamente estudado como iremos tentar mostrar de seguida. Contudo, não foram encontradas referências que indiquem soluções para redes multimodais estocásticas, objectivo deste trabalho.

### 2.2.1 Detecção de conflitos na alocação de recursos

Depois de feito o escalonamento do projecto, tendo em conta as restrições de precedência entre as actividades, como descrito na secção 2.1.2, deverão ser analisados os efeitos das restrições de recursos nesse escalonamento. Para tal, e para cada período do horizonte de planeamento, devemos determinar a necessidade dos vários recursos. Se para um destes períodos, a quantidade de recursos necessária exceder a disponível, ocorre um conflito de recursos que tem que ser resolvido. Normalmente o tipo de recursos mais estudado neste contexto são os recursos renováveis.

Seja então:

$R = \{1, \dots, m\}$  - o conjunto dos recursos renováveis envolvidos no projecto;

$a_r$  - a quantidade de recurso  $r$  disponível em cada período do horizonte de planeamento;

$u_{jr}$  - as unidades de recurso  $r$  necessárias para executar a actividade  $j$ ;

e dado um escalonamento, podemos calcular, para cada actividade, a quantidade de recursos necessária (na tabela 2.2 está representada a quantidade necessária de

um recurso para o projecto exemplo; por uma questão de visibilidade, repete-se a rede do projecto exemplo, acrescida da informação adicional - figura 2.8).

Tabela 2.2: Utilização do recurso 1 nas actividades do projecto exemplo

Actividade	1	2	3	4	5	6
$u_{j1}$	0	3	1	2	2	0

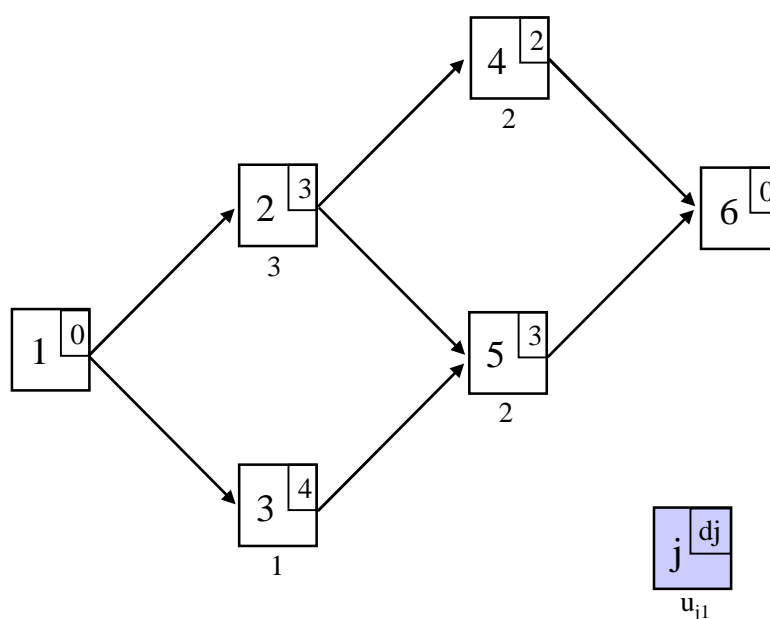


Figura 2.8: Projecto exemplo com informação sobre necessidade de recursos

De seguida, para cada período do horizonte de planeamento, são identificadas as actividades em progresso. Para estas actividades, calcula-se a procura total, obtendo-se para cada recurso um gráfico com o seu perfil de utilização (ver figura 2.9)<sup>2</sup>. Este tipo de gráfico permite visualizar imediatamente em que períodos ocorrem conflitos de recursos, tendo contudo a desvantagem de não permitir identificar quais as actividades que causaram esses conflitos. Alternativamente é possível construir um gráfico de Gantt orientado à capacidade, que já

<sup>2</sup>No exemplo considera-se  $a_1 = 3$ .

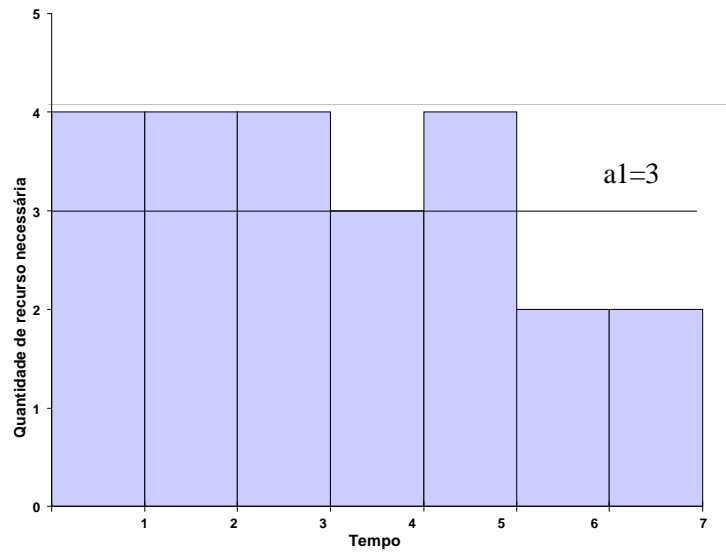


Figura 2.9: Perfil de utilização de recursos para o projecto exemplo

nos permite visualizar essa informação (ver figura 2.10).

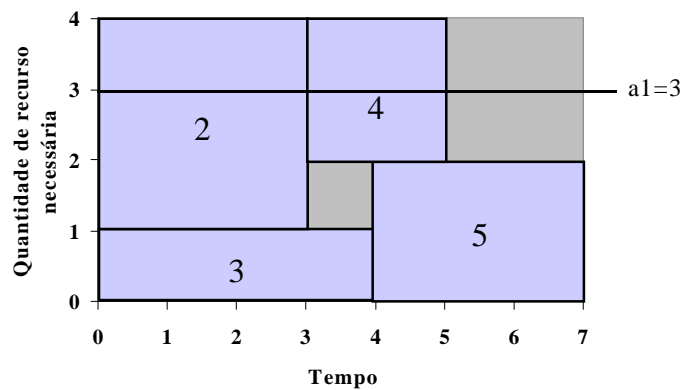


Figura 2.10: Gráfico de Gantt orientado à capacidade para o projecto exemplo

## 2.2.2 Escalonamento com restrições de recursos

A determinação do perfil de utilização de recursos permite-nos detectar eventuais conflitos de recursos, que têm que ser resolvidos para que o escalonamento seja viável. A forma mais simples de resolver esses conflitos seria alocar recur-



tos adicionais nos períodos críticos, o que nem sempre é possível por restrições orçamentais e outras. Perante esta situação, o objectivo será desenvolver um escalonamento que respeite as restrições de precedências e de recursos, de tal forma que o projecto termine o mais cedo possível. Este problema, conhecido na literatura por ‘Resource Constraint Project Scheduling Problem’, como já foi anteriormente referido, é de difícil resolução, especialmente no caso de serem considerados vários tipos de recursos, pelo que tem representado um desafio para muitos investigadores. Na literatura podemos encontrar inúmeros procedimentos para resolução deste problema, seguindo abordagens distintas. Vamos tentar apresentar alguns deles nas secções seguintes.

## 2.3 RCPSP por optimização matemática

Uma das primeiras formulações matemáticas desenvolvidas explicitamente para resolver o RCPSP, foi apresentada por Pritsker, Watters e Wolfe ([Pritsker *et al.* 69]). Neste modelo as soluções são representadas por variáveis binárias ( $x_{jt}$ ) definidas da seguinte forma:

$$x_{jt} = \begin{cases} 1, \text{ se a actividade } j \text{ termina no fim do período } t \\ 0, \text{ caso contrário} \end{cases} \quad \left. \begin{array}{l} j \in J \\ t \in [EF_j, LF_j], \end{array} \right\} \quad (2.5)$$

onde  $J$  representa o conjunto das actividades (‘Jobs’).

Para um dado vector solução, o período  $SF_j$  - ‘Scheduled to Finish’, no qual a actividade é escalonada para terminar, calcula-se da seguinte forma:

$$SF_j = \sum_{t=EF_j}^{LF_j} t \cdot x_{jt}. \quad (2.6)$$

Considerando as variáveis acima introduzidas, o modelo pode ser representado da seguinte forma (função objectivo + restrições):

$$\text{Minimizar } CT(x) = \sum_{t=EF_n}^{LF_n} t.x_{nt} \quad (2.7)$$

$$\text{Sujeito a } \sum_{t=EF_j}^{LF_j} x_{jt} = 1 \text{ para } j \in J \quad (2.8)$$

$$\sum_{t=EF_j}^{LF_j} (t - d_j).x_{jt} - \sum_{t=EF_i}^{LF_i} t.x_{it} \geq 0 \text{ para } j \in J \text{ e } i \in P_j \quad (2.9)$$

$$\sum_{j \in \bar{E}(t)} u_{jr} \cdot \sum_{q=\max\{t, EF_j\}}^{\min\{t+d_j-1, LF_j\}} x_{jq} \leq a_r \text{ para } r \in R \text{ e } t \in [1, \dots, \bar{T}] \quad (2.10)$$

$$x_{jt} \in \{0, 1\} \text{ para } j \in J \text{ e } t \in [EF_j, \dots, LF_j], \quad (2.11)$$

$P_j$  representa os precedentes imediatos de  $j$ ,  $R$  o conjunto dos recursos renováveis,  $\bar{T}$  o fim do horizonte de planeamento e  $E(t)$  as actividades que podem ser processadas no período  $t$  (ver fórmula (2.12)).  $\bar{T}$  deve representar um limite máximo para a duração do projecto e pode ser calculado somando as durações de todas as actividades envolvidas, como se fossem executadas sequencialmente.

$$E(t) = \{j \mid j \in J \text{ e } ES_j + 1 \leq t \leq LF_j\}. \quad (2.12)$$

O modelo pode ser interpretado da seguinte forma: a função objectivo (2.7) minimiza o tempo de conclusão da última actividade. As restrições (2.8) e (2.11) garantem que só um final é atribuído a cada actividade. As restrições de precedências são representadas por (2.9), que garante que nenhuma actividade possa começar antes dos seus precedentes terem terminado. Finalmente (2.10) representa as restrições de recursos, onde se determina a necessidade total de recursos em cada período  $t$  e se compara com a disponibilidade total.

Existe um grande conjunto deste tipo de formulações, que se baseiam no mesmo princípio de minimizar a data de conclusão do projecto, tendo como restrições as precedências e os recursos, também baseadas em variáveis binárias com significados idênticos ([AlvarezTamarit93], [McKnew*et.al*91], [Mingozi*et.al*98], [NeumannMorlock93], [Scholl99]). Algumas não foram desenvolvidas para resolver directamente o RCPSP, mas podem ser adaptadas para tal [Klein00].

Existem também extensões destes modelos que permitem, por exemplo, interrupção na execução das actividades<sup>3</sup> - ‘Preemption’. Neste caso o que é feito é dividir cada actividade com duração  $d_j$  em  $d_j$  sub-actividades, com duração unitária, e introduzir relações de precedência entre as sub-actividades consecutivas. Este problema foi tratado por [Bianco*et.al*99], [DemeulemeesterHerroelen96] e [Kaplam88].

Uma outra extensão é o RCPSP multimodal. Este problema consiste na possibilidade de variar a duração das actividades, pela aplicação de diferentes níveis de recursos. Assim é introduzida mais uma variável: o modo de execução de cada actividade. Este problema foi tratado por [HartmannSprecher96], [Patterson*et.al*89], [Pesch99], [SperanzaVercellis93], [Sprecher94], [SprecherDrex198], [Sprecher*et.al*97] e [Talbot82] e é também focado neste trabalho.

Um problema relacionado com o RCPSP é o TCPSP - ‘Time Constrained Project Scheduling Problem’. Este problema consiste na tentativa de minimizar o custo dos recursos, por forma a completar o projecto numa data especificada, utilizando se necessário recursos adicionais. Soluções para este problema foram apresentadas por [DeckroHebert89], [Kolisch95], [NeumannZimmermann97], [NeumannZimmermann98] e [NeumannZimmermann99].

---

<sup>3</sup>Normalmente considera-se que depois de iniciar uma actividade ela tem que ser executada até ao fim sem interrupção.

## 2.4 RCPSP - Métodos de determinação de limites inferiores

A determinação de limites inferiores para a conclusão do projecto tem sido estudada por vários autores, pois permite comparar soluções heurísticas ou mesmo integrar procedimentos exactos, para reduzir o esforço de cálculo necessário. Nesta secção vamos tentar descrever alguns destes métodos e deixar também referências para consultas mais aprofundadas.

### 2.4.1 ‘Critical Path Bound’ (CPB)

Este limite inferior é o mais simples e o mais utilizado, e é calculado omitindo as restrições de recursos. A duração mínima do projecto neste problema relaxado é obtida calculando o comprimento do caminho crítico. No caso do projecto exemplo (ver figura 2.8),  $CPB = 7$ .

### 2.4.2 ‘Capacity Bound’ (CB)

No cálculo deste limite inferior é considerado apenas um único recurso e ignorados os restantes.

$$CB = \max \left\{ \left\lceil \frac{(\sum_{j=1}^n u_{jr} \cdot d_j)}{a_r} \right\rceil \mid r = 1, \dots, m \right\}. \quad (2.13)$$

CB é calculado somando a necessidade total por recurso e dividindo esse valor pela sua disponibilidade por período. O valor obtido arredonda-se para cima ao inteiro mais próximo. Para o projecto exemplo,  $CB = \lceil 23/3 \rceil = 8$ .

### 2.4.3 ‘Critical Sequence Bound’ (CSB)

Este limite inferior pode ser considerado uma junção dos dois anteriores [Stinson *et. al*78]. Em primeiro lugar colocam-se as actividades do caminho crítico (CP - ‘Critical Path’) sequencialmente (ver figura 2.11).

## 2.4. RCPSP - MÉTODOS DE DETERMINAÇÃO DE LIMITES INFERIORES 27

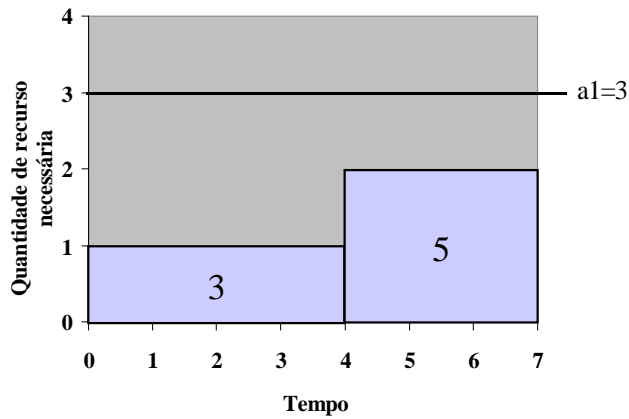


Figura 2.11: Escalonamento sequencial das actividades do caminho crítico para o projecto exemplo

O processamento das outras actividades  $j \in J - CP$  em paralelo com as actividades do caminho crítico requiere que na janela temporal  $[ES_j, LF_j]$  exista um intervalo com tamanho mínimo  $d_j$  durante o qual a capacidade residual não seja inferior a  $u_{jr}$ , para cada tipo de recurso. Seja  $e_j^{\max}$  o tamanho máximo desse intervalo. Se  $e_j^{\max} < d_j$ , a actividade  $j$  não pode ser processada completamente, e o projecto não pode terminar em menos de  $CPB + d_j - e_j^{\max}$  períodos. Assim  $CSB$  é definido como se segue:

$$CSB = CPB + \max \{0, \max \{d_j - e_j^{\max} \mid j \in J - CP\}\}. \quad (2.14)$$

Para o nosso exemplo:

para  $j = 2$ ;  $d_j = 3$ ;  $[ES_j, LF_j] = [0, 4] \rightarrow e_j^{\max} = 0$ ;

para  $j = 4$ ;  $d_j = 2$ ;  $[ES_j, LF_j] = [3, 7] \rightarrow e_j^{\max} = 1$ ;

então  $CSB = 7 + \max \{0, \max \{3 - 0, 2 - 1\}\} = 10$ .

Este limite inferior considera as actividades do caminho crítico e mais uma (a que necessita de mais tempo extra para ser executada), ignorando as restantes.

Existem depois extensões a estes limites inferiores que consideram várias actividades simultaneamente, para além das actividades do caminho crítico ([Brucker *et al*96], [Demeulemeester92]).

#### 2.4.4 Outros limites inferiores

Outro tipo de limites inferiores são os ‘Bin Packing Bounds’, que são calculados ignorando todas as restrições de precedências e todas as restrições de recursos excepto uma ([Berger *et al*92], [Labbé *et al*91], [MartelloToth90], [Scholl *et al*97]). O seu funcionamento baseia-se na ordenação das actividades por necessidade decrescente de recursos. De seguida, são escalonadas as actividades com maiores necessidades sequencialmente, e as últimas da lista são escalonadas paralelamente com as anteriores. A duração total do projecto assim obtido dará o valor do limite.

Os ‘Node Packing Bounds’ baseiam-se na procura de um sub-conjunto de actividades que sejam incompatíveis entre si, ou seja, que não possam ser processadas em paralelo, por restrições de precedências ou recursos. De seguida, estas actividades são escalonadas sequencialmente, definindo um limite inferior igual à soma das suas durações [De\_ReyckHerroelen98], [DemeulemeesterHerroelen97], [Mingozi *et al*98].

Ainda um outro tipo de limites inferiores são os ‘Precedence Bounds’, que consistem em identificar pares de actividades incompatíveis. De seguida, são introduzidos arcos temporários entre essas actividades, sendo depois analisadas as redes assim obtidas, calculando-se a duração deste projecto teste. O limite inferior é obtido calculando, em primeiro lugar, o mínimo para cada par e depois o máximo destes valores [Klein00].

Existem ainda mais algumas abordagens para calcular limites inferiores, que não irão ser descritas devido à sua complexidade, mas que vamos referir. ‘LP-Relaxation with Cutting Planes’ apresentada por Christofides, Alvarez-Valdés e Tamarit ([Christofides *et al*87]), ‘Lagrangean Relaxation’ por Fisher e

Geoffrion ([Fisher73], [Fisher81], [Geoffrion74]) e ainda ‘Set Covering Based Approach’ descrita por Mingozzi, Maniezzo, Ricciardelle e Bianco ([Mingozzi*et.al*98]). Para terminar, Klein e School ([KleinScholl99]) introduziram uma meta-estratégia para o cálculo destes limites, conhecida por ‘Destructive Improvement’.

## 2.5 RCPSP - Procedimentos heurísticos

Dada a complexidade do RCPSP, a utilização de optimização matemática para obter soluções óptimas, é por vezes pouco eficiente em termos de tempo de computação. Na prática, a utilização de heurísticas pode resolver esta situação, permitindo obter escalonamentos realizáveis num curto espaço de tempo.

Existem duas classes de heurísticas que podem ser aplicadas ao RCPSP - os métodos construtivos e os métodos de melhoramento. Os métodos construtivos, partem de um estado inicial, e através de um conjunto de operações, encontram um escalonamento viável. Como o seu funcionamento se baseia na ordenação das actividades segundo uma determinada regra, também são conhecidas como heurísticas baseadas em regras de prioridade. Os métodos de melhoramento (‘Improvement Methods’), são utilizados para aperfeiçoar uma solução inicial, que tenha sido obtida por um método construtivo. Usam para tal um conjunto de operações, que vão transformando as soluções intermédias, até se alcançar a solução pretendida.

### 2.5.1 Métodos construtivos

Os métodos construtivos têm dois componentes principais, o esquema de escalonamento e as regras de prioridade. O esquema de escalonamento descreve como construir um escalonamento viável, associando sucessivamente tempos de início às actividades. Enquanto isto é feito, as regras de prioridade são aplicadas para seleccionar qual a actividade a ser escalonada a seguir.

Existem dezenas de regras de prioridade descritas na literatura, das quais destacamos algumas, na tabela 2.3.

Tabela 2.3: Regras de prioridade

<b>Regra</b>	<b>Significado</b>
SPT	‘Shortest Processing Time’
EST	‘Earliest Starting Time’
EFT	‘Earliest Finishing Time’
LST	‘Latest Starting Time’
LFT	‘Latest Finishing Time’
MSL	‘Minimum SLack time’
GRD	‘Greatest Resource Demand’
WRUP	‘Weighted Resource Utilization and Precedence’
ACTIM	‘Activity Control TIME’

A regra SPT, equivale a escolher a actividade com menor duração. Um outro conjunto de regras (EST, EFT, LST, LFT) utilizam os tempos mais cedo/mais tarde de início/fim das actividades. Estas regras são baseadas em informações da rede e do CPM. Existem também regras centradas nos recursos, como por exemplo, seleccionar a actividade com maior necessidade de recursos (GRD), e regras que fazem uma conjugação da informação sobre os recursos e sobre as precedências (WRUP). Finalmente, ACTIM determina o caminho mais longo de uma actividade  $j$  até ao nó final. Para uma consulta mais aprofundada deste e de outros tipos de regras, ver [Bedworth73], [Brand*et.al*64] e [UlusoyÖzdamar89].

De seguida apresentam-se alguns procedimentos que traduzem esquemas de escalonamento para o RCPSP.



### Procedimento de Lang

Este procedimento [Lang77] utiliza valores retirados da aplicação do método CPM. Assim, o escalonamento das actividades é feito da seguinte forma:

1. Ordenar as actividades pelo seu LST.
2. Quando o LST for o mesmo para mais do que uma actividade, seleccionar a actividade com:
  - a. menor folga,
  - b. maior duração,
  - c. maior necessidade de recursos,

tendo em atenção a não violação das restrições de recursos.

*Exemplo.* No caso do projecto exemplo, é suficiente a aplicação da primeira regra de prioridade, pois não há empates. Assim, a regra LST dá-nos a seguinte ordenação:

Actividade	LST
3	0
2	1
5	4
4	5

O escalonamento resultante está representado na figura 2.12. Este procedimento permite obter um escalonamento viável, isto é, que respeita as restrições de precedência e de recursos, com um tempo de conclusão do projecto igual a 9 períodos.

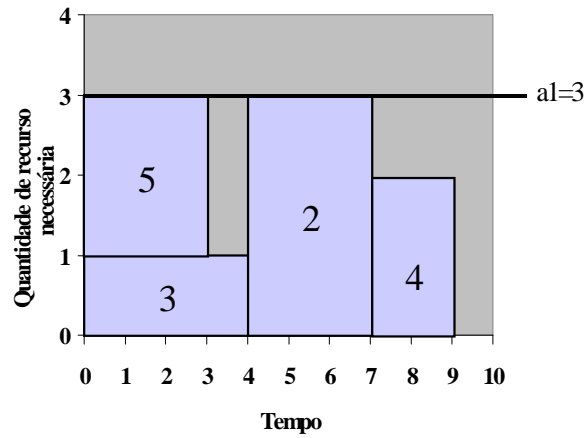


Figura 2.12: Solução obtida usando o Procedimento de Lang

### ‘Serial Scheduling Scheme’ - SSS

Este esquema de escalonamento foi proposto por Kelley ([Kelley63]), e constrói um escalonamento viável em exactamente  $n$  iterações, sendo  $n$  o número de actividades. Em cada iteração, adiciona-se uma actividade ao escalonamento actual parcial (PS- ‘Partial Schedule’), da seguinte forma: depois de determinar o conjunto das actividades disponíveis ( $A(PS)$  - ‘Available jobs of Partial Schedule’), ou seja, as actividades cujos precedentes fazem parte do PS, selecciona-se uma dessas actividades por uma regra de prioridade. Esta actividade deve começar no instante mais cedo  $t \geq ES_j(PS)$ , de tal forma que não viole as restrições de recursos durante a sua duração. Para o escalonamento parcial obtido ( $PS'$ ), calculam-se as disponibilidades de recurso residuais, bem como os novos  $ES_j(PS')$  e  $EF_j(PS')$  de todas as actividades por escalonar. Este procedimento continua até todas as actividades estarem escalonadas.

*Exemplo.* A figura 2.13 mostra a solução obtida para o projecto exemplo usando SSS, com a utilização da regra SPT, que dentro das actividades disponíveis, selecciona aquela com menor duração. Depois de escalonar a actividade fictícia 1, as actividades disponíveis são a 2 e a 3 ( $A(PS) = \{2, 3\}$ ). Utilizando a regra

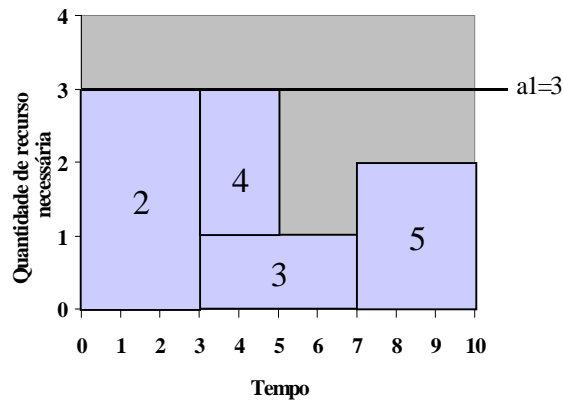


Figura 2.13: Solução obtida usando SSS com a regra SPT para o projecto exemplo

SPT, é seleccionada a actividade 2 e  $A(PS) = \{3, 4\}$ . De seguida, é seleccionada a actividade 4, ficando  $A(PS) = \{3\}$ .  $ES_3 = 0$ , mas como a actividade 2 está a consumir todos os recursos disponíveis, 3 só pode começar em  $t = 3$  ( $SS_3 = 3$ ). Finalmente  $A(PS) = \{5\}$  só pode ser escalonada para iniciar em  $t = 7$ , pois tem que esperar pela conclusão da 2 e da 3. Assim obtem-se um escalonamento viável, com um tempo de conclusão do projecto igual a 10 períodos.

### ‘Parallel Scheduling Scheme’ - PSS

O PSS gera um escalonamento viável em  $n$  iterações no máximo [BrooksWhite65]. Em cada iteração, actualiza-se o ‘Partial Schedule’ e o ponto de decisão  $t$ . O ponto de decisão escolhe-se de tal forma que todas as actividades já escalonadas comecem antes de  $t$  ( $SS_j(PS) < t$ ), onde SS significa ‘Scheduling Start’. No início de cada iteração, determina-se o conjunto das actividades elegíveis  $E(PS, t)$ . De seguida, e seguindo uma determinada regra de prioridade, selecciona-se e remove-se de  $E(PS, t)$  uma actividade  $j$ . Se for possível começar  $j$  no instante  $t$ , sem violar restrições de recursos nos instantes seguintes,  $SS_j = t$ . Se não, ignora-se  $j$ . No final de cada iteração, calcula-se o próximo ponto de decisão  $t'$ , como sendo o mínimo tempo de conclusão das actividades do PS.

Uma vez que o nosso exemplo é muito simples, a aplicação do PSS resulta na

mesma solução obtida usando SSS.

Existem outros esquemas de escalonamento que se baseiam na utilização de diferentes direcções de planeamento e que segundo [Klein98] podem aumentar a eficiência destas heurísticas. São exemplos o ‘Backward Planning’ proposto por Li e Willis ([LiWillis92]) e ‘Bidirectional Planning’ aplicado por Dell’Amico e Trubian ([Dell’AmicoTrubian93]).

### 2.5.2 Métodos de melhoramento

Estes métodos assentam na sucessiva transformação de uma solução noutra, na tentativa de obter melhores resultados. Se o problema for de minimização, que é o caso do RCPSP, existem procedimentos que só aceitam novas soluções se melhorarem a anterior (‘Steepest Descent Procedures’). Esta restrição leva a que muito frequentemente se determinem óptimos locais, sendo impossível sair deles. Para ultrapassar este problema, deve ser possível aceitar soluções que piorem o valor da função objectivo, correndo ainda assim o risco de visitar soluções óptimas locais. Estes problemas levaram ao desenvolvimento de métodos heurísticos modernos como o ‘Simulated Annealing’ e o ‘Tabu Search’, que são normalmente designados por meta-heurísticas. Existe ainda outro tipo de meta-heurísticas, os algoritmos evolutivos, que diferem dos métodos de melhoramento tradicionais, pois em vez de considerar só uma solução de cada vez, combinam um conjunto de soluções para obter novas soluções. É o caso dos algoritmos genéticos e das redes neuronais. Vamos, de seguida, descrever sucintamente alguns deles.

#### ‘Tabu Search’

Esta meta-heurística foi inicialmente proposta por Glover ([Glover89], [Glover90]). Consiste basicamente em transformar sucessivamente a solução actual  $x$  numa outra  $x'$ , até obter um valor para a função objectivo próximo do óptimo. No

caso do RCPSP, as soluções podem ser representadas por escalonamentos viáveis. Existe o conceito de vizinhança de  $x$ , que é o conjunto de todas as soluções viáveis que se podem obter com uma única movimentação ('move'). A abordagem normalmente seguida pelo 'Tabu Search' consiste em utilizar o procedimento 'Steepest Descent', mas com movimentos de deterioração intercalares, a que se dá o nome de 'Steepest Descent/Mildest Ascent Procedure'. Uma outra característica do 'Tabu Search' é proibir certos movimentos que levem a ciclos, classificando essas soluções como tabu. Sempre que é encontrada uma solução melhor, ela é armazenada. O processo termina quando se verificar um determinado critério de paragem.

### 'Simulated Annealing'

Esta meta-heurística simula um processo físico que consiste em arrefecer um sólido derretido até atingir um estado de baixa energia [AartsKorst89], [Dowsland93], [Kirkpatrick*et.al*83]. Em cada iteração selecciona-se aleatoriamente um movimento possível. Se esse movimento resultar numa solução com melhor valor para a função objectivo, ele é executado imediatamente. Caso contrário, é executado com uma certa probabilidade, que depende da deterioração  $\delta$  do valor da função objectivo bem como do parâmetro  $\Theta$ , a temperatura. Esta probabilidade é calculada usando uma função do tipo  $e^{-\delta/\Theta}$ . À medida que o processo de procura avança, a temperatura vai sendo reduzida, de tal forma que a probabilidade de executar um movimento de deterioração vá diminuindo. A procura continua até o critério de paragem ser verificado, como por exemplo a temperatura estar abaixo de um determinado nível.

Esta técnica foi aplicada ao caso do RCPSP por [Boctor96], [BouleimenLecocq98], [ChoKim97] e [LeeKim96].

### ‘Genetic Algorithms’

Esta meta-heurística foi inspirada no processo de evolução biológico. Ela manipula um conjunto (população) de soluções (indivíduos) em cada iteração [Dowsland96], [Goldberg89], [Holland75], [KolenPesch94], [Michalewicz94], [Reeves97]. Antes de se iniciar o algoritmo genético, têm que se determinar um conjunto de soluções viáveis, para formar a população inicial, o que pode ser feito usando métodos construtivos. O funcionamento do algoritmo consiste em gerar uma nova população em cada iteração, com o mesmo tamanho da anterior, usando as seguintes operações: ‘crossover’ que, através de duas soluções aleatoriamente escolhidas gera descendentes por “acasalamento” e ‘mutation’ que modifica aleatoriamente indivíduos para obter mutações. De seguida, é analisado o valor da função objectivo para seleccionar os elementos que vão formar a próxima população. Este processo continua até que se verifique um determinado critério de paragem.

Os algoritmos genéticos foram utilizados para RCPSP por [Hartmann98], [LeeKim96] e [LeonBalakrishnan95].

## 2.6 RCPSP - Procedimentos exactos

Estes procedimentos permitem determinar soluções óptimas. Contudo, e porque são mais exigentes computacionalmente, podem não ser aplicáveis a problemas de grande dimensão, ou a problemas com necessidade de soluções em tempo real. Desempenham contudo um papel fundamental na avaliação do desempenho de procedimentos heurísticos, permitindo comparar as suas soluções com o valor óptimo. Podem eles próprios funcionar também como procedimentos heurísticos, se a pesquisa do óptimo for suspensa depois de decorrido um certo intervalo de tempo.

Existem três tipos de procedimentos exactos, aplicados a este caso: ‘branch and bound’, programação inteira e programação dinâmica.

### 2.6.1 Procedimentos ‘branch and bound’

Estes procedimentos começam por dividir o problema inicial em subproblemas, continuando este processo até obter instâncias de fácil resolução. A representação desta divisão é feita através de uma árvore, em que os subproblemas obtidos são representados pelos nós.

Existem alguns princípios básicos que impedem a análise dos subproblemas que não levem a soluções óptimas, por exemplo, podemos ignorar os subproblemas cujos limites inferiores são maiores do que a melhor solução conhecida até ao momento. Podemos utilizar regras de redução para tentar modificar os dados e identificar possíveis contradições em determinados nós [Dorndorf*et.al*99] e finalmente, aplicando regras de dominância, podemos detectar subproblemas que não consigam fornecer uma melhor solução do que a melhor existente até ao momento [Ibaraki77].

#### Esquemas de ‘branching’

O esquema de ‘branching’ traduz a forma como são divididos os problemas, divisão que resulta numa árvore, como já foi referido. A raiz da árvore representa o problema inicial. Todos os outros nós são descendentes. O processo de ‘branching’ pára quando para cada problema do último nível (folhas) se conhece a solução óptima, ou se pode calcular facilmente. O caminho entre a raiz e qualquer outro nó é chamado ramo.

Os subproblemas são normalmente construídos fixando uma ou mais variáveis do problema inicial, ou restringindo o intervalo dos seus valores. O esquema de ‘branching’ tem que controlar quais as variáveis a serem seleccionadas, e que valores lhes são atribuídos, influenciando desta forma o tamanho e a estrutura da árvore. Esta selecção é normalmente feita heurísticamente, por exemplo, com regras de prioridade.

### Estratégias de pesquisa

A estratégia de pesquisa define a sequência pela qual os subproblemas da árvore são considerados. Essa sequência baseia-se normalmente nos limites inferiores do valor da função objectivo, para os vários subproblemas.

Existem duas estratégias de pesquisa básicas, que são usadas na maior parte dos procedimentos de ‘branch-and-bound’ [Johnson88].

- DFS - ‘Depth-First Search strategy’: é feita uma procura por profundidade, ou seja, é desenvolvido um único ramo da árvore até chegar à folha. De seguida, continua-se pela primeira alternativa, subindo na direcção do nó raiz, ou seja, todos os descendentes de um nó são construídos antes da procura regressar ao nó anterior.
- MLB - ‘Minimal Lower Bound search’: funciona com base numa lista global de candidatos, que começa por conter o nó raiz. Os nós devem ser armazenados nessa lista por ordem crescente do limite inferior. Constrói-se a árvore retirando da lista o nó com o mínimo valor, e colocando na lista todos os seus descendentes. Desta forma, ao construir a árvore, são sempre explorados todos os descendentes de cada nó.

Considere a árvore de figura 2.14. O número próximo dos nós representa o valor que o limite inferior local tem a mais relativamente ao limite inferior global, do nó raiz. Podemos ver na tabela 2.4 os efeitos da escolha de cada uma das estratégias de pesquisa apresentadas, ou seja, a ordem de visita dos vários nós, para cada estratégia.

Tabela 2.4: Ordem de visita das estratégias DFS e MLB

Estratégia	Ordem de visita
DFS	0, 1, 4, 11, 12, 5, 13, 2, 6, 14, 7, 15, 16, 3, 8, 17, 9, 18, 19, 10, 20
MLB	0, 2, 3, 1, 7, 9, 10, 5, 6, 13, 16, 18, 4, 11, 15, 19, 20, 8, 12, 14, 17



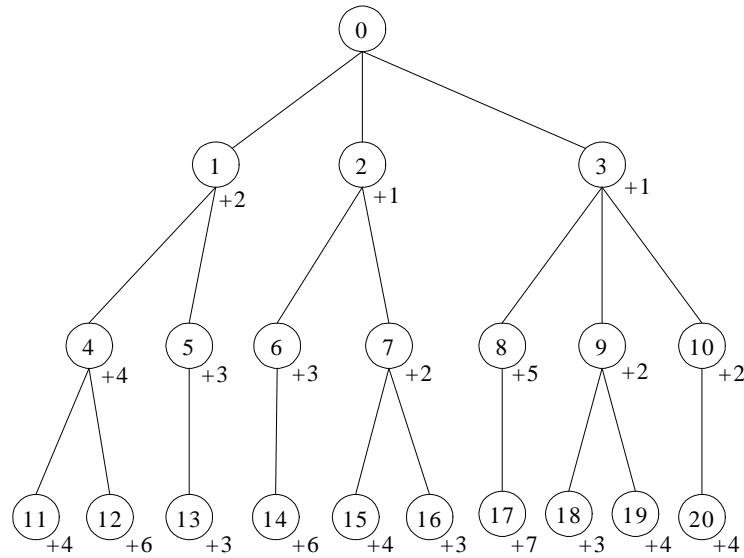


Figura 2.14: Árvore exemplo

Para uma consulta mais aprofundada dos procedimentos ‘branch-and-bound’, ver [Geoffrion74], [HillierLieberman95], [Johnson88] e [Roucairol96]. A aplicação desta técnica ao RCPSP é apresentada em [KleinScholl98].

### 2.6.2 Procedimentos de programação inteira

As primeiras tentativas de utilizar programação inteira para resolver o RCPSP falharam [Brand*et.al*64], [Davis73]. Foram contudo desenvolvidas abordagens especiais, normalmente usando enumeração, para resolver este problema, de uma forma óptima [Balas70], [DavisHeidorn71], [Gorenstein72], [Schrage70], [Talbot78]. Vamos, de seguida, ver duas delas.

#### Procedimento de enumeração limitada de Davis

Este procedimento de Davis e Heidorn [DavisHeidorn71] começa por dividir cada uma das actividades do projecto numa série de tarefas de duração unitária, ou seja, cada actividade é dividida num número de tarefas unitárias igual à sua duração. Depois de feita esta divisão, é criada uma ‘A-network’, em que cada nó

representa uma sequência que respeita as precedências, para um subconjunto de tarefas unitárias. Constrói-se assim uma árvore de soluções parciais, sem considerar as restrições de recursos. São depois adicionados os arcos, tendo já em conta as restrições de recursos. O número de estados (níveis) nos quais são adicionados nós, é igual ao tamanho do caminho crítico da rede original, correspondendo cada nível a um período diferente na construção da rede. O número de nós (sequências viáveis em termos de precedências) em cada estado, determina-se usando um procedimento de Held e Karp [HeldKarp62]. É usada programação dinâmica para determinar o caminho mais curto na rede de soluções parciais gerada, tendo como resultado a determinação do escalonamento mais curto para o problema em causa.

Uma vez que o número de sequências viáveis em cada estado pode ser muito grande, são usados critérios de eliminação de subconjuntos baseados tanto em restrições de precedências, como de recursos. Por exemplo, se tivermos para cada actividade o valor do seu LPT - 'Latest Processing Time', podemos ignorar os escalonamentos que coloquem as actividades a executar para além desse tempo. Podemos também ignorar os escalonamentos que não tenham um número suficiente de recursos para a execução das suas actividades. Existem também métodos que permitem reduzir o número de arcos gerados, como por exemplo, para existir um arco de  $i$  para  $j$ ,  $i$  tem que ser menor do que  $j$ . Estas medidas reduzem o esforço computacional necessário.

As vantagens deste procedimento têm a ver com a possibilidade de considerar as actividades de uma forma não contínua, ou seja, a sua execução pode iniciar e ser interrompida ('preemption'), podendo o nível de recursos variar durante a execução da actividade.

### **Procedimento de enumeração implícita de Talbot**

Esta formulação de Talbot [Talbot78], apesar de ser equivalente em termos de implementação a várias formulações por programação binária, utiliza variáveis

inteiras, conseguindo assim maior poupança em termos de memória. O procedimento baseia-se na avaliação sistemática de todos os possíveis tempos de término das actividades do projecto. São contudo utilizadas técnicas de sondagem que permitem excluir de consideração escalonamentos não interessantes, na fase inicial de enumeração do algoritmo, o que contribui também para a poupança de memória acima referida.

O procedimento começa por determinar o menor tempo de término possível da actividade 1. Continua depois atribuindo consecutivamente às várias actividades do projecto o seu menor tempo de término possível. O processo continua até chegarmos à última actividade. Nesse momento, obtém-se uma solução melhorada para o problema.

Se neste processo não for possível atribuir um recurso a uma actividade, ou não for possível melhorar o seu tempo de término, o procedimento recua para a actividade anterior. Se para esta actividade se conseguir atribuir um tempo de término que não prolongue a duração do escalonamento existente, faz-se nova tentativa para determinar o tempo de término para a actividade que originou o problema. Se isto não for possível, recua-se novamente. Se este processo chegar à actividade 1, e tentar recuar, temos a certeza que conseguimos um óptimo.

A formulação conceptual deste algoritmo é a que se segue:

$$\text{Minimizar } f_N \quad (2.15)$$

Sujeito a

$$\max_{n \in P_j} \{f_n\} + d_j \leq f_j \quad j = 1, \dots, N \quad (2.16)$$

$$\sum_{j \in S_t} r_{jk} \leq R_{kt}^* \quad k = 1, \dots, K \text{ e } t = 1, \dots, \bar{T}, \quad (2.17)$$

onde  $f_j$  representa o tempo real de término da actividade,  $P_j$  o conjunto dos precedentes imediatos de  $j$ ,  $S_t$  o conjunto de todas as actividades em curso no

período  $t$ ,  $R_{kt}^*$  a quantidade total de recurso  $k$  disponível no período  $t$ , e  $r_{jk}$  a quantidade de recurso  $k$  necessária para a actividade  $j$  em cada período de tempo em que  $j$  está activa.

O objectivo (2.15) é minimizar o tempo de conclusão da última actividade e logo a duração do projecto. A restrição (2.16) traduz as relações de precedência e (2.17) as restrições de recursos.  $\bar{T}$  pode ser calculado como o somatório das durações de todas as actividades do projecto.

### 2.6.3 Procedimentos de programação dinâmica

A programação dinâmica é um método desenvolvido para ser aplicado a processos de decisão multi-estágio [Bellman57]. Em 1955, Bellman e Dreyfus levaram a cabo um estudo sistemático da aplicabilidade da PD a um grande conjunto de problemas de optimização, oriundos de diferentes áreas, visando essencialmente a utilização do computador. Depois deste estudo concluíram que a PD podia realmente ser usada para resolver um grande número de problemas de optimização. Dentro destes problemas, alguns podiam ser resolvidos por outras técnicas com dificuldade, alguns só por tentativa e erro, e outros não conseguiam ser resolvidos por nenhum outro método conhecido [BellmanDreyfus62].

Um exemplo que é apresentado pelos autores, enquadra-se precisamente neste trabalho. Trata-se de um processo de alocação de recursos, em que o objectivo é maximizar o lucro ou retorno, através de uma correcta distribuição dos recursos disponíveis pelas actividades. Assim, o processo pode ser descrito da seguinte forma: determinar o máximo da seguinte função de  $N$  variáveis,

$$R(x_1, x_2, \dots, x_N) = g_1(x_1) + g_2(x_2) + \dots + g_N(x_N) \quad (2.18)$$

sujeito a

$$\begin{aligned} x_1 + x_2 + \dots + x_N &= x \\ \text{e } x_i &\geq 0 \end{aligned} \tag{2.19}$$

em que  $g_i(x_i)$  é o retorno obtido pela aplicação da quantidade de recurso  $x_i$  à actividade  $i$ .

A restrição (2.19) prende-se com a existência de uma quantidade total limitada de recursos  $x$ . Os autores tentaram usar cálculo matemático para resolver este problema, através da utilização de multiplicadores de Lagrange. Depararam contudo com algumas dificuldades, nomeadamente no que se refere à necessidade de garantir máximos absolutos, entre outras (ver [BellmanDreyfus62]). Apontam assim a PD como a solução para esses problemas.

A aplicação da PD requiere, em primeiro lugar, que o problema que foi formulado como um processo estático, seja convertido num processo dinâmico. Ou seja, em vez das alocações serem feitas todas simultaneamente, irão ser feitas uma de cada vez. Uma vez que o máximo de  $R(x_1, x_2, \dots, x_N)$  depende de  $x$  e de  $N$ , torna-se essa dependência específica, pela introdução da sequência de funções  $f_N(x)$ ,

$$\begin{aligned} f_N(x) &= \max_{\{x_i\}} R(x_1, x_2, \dots, x_N) \\ \text{com } N &= 1, 2, \dots; \quad x_i \geq 0 \quad \text{e} \quad \sum_{i=1}^N x_i = x. \end{aligned} \tag{2.20}$$

$f_N(x)$  representa então o retorno máximo da alocação da quantidade de recursos  $x$  a  $N$  actividades.

Há dois casos particulares, em que  $f_N(x)$  é representada por expressões mais simples,

$$\begin{aligned} f_N(0) &= 0 \quad N = 1, 2, \dots \\ \text{e } f_1(x) &= g_1(x) \quad x \geq 0. \end{aligned} \tag{2.21}$$

O próximo passo será obter a relação de recorrência, típica de PD, que liga  $f_N(x)$  a  $f_{N-1}(x)$ . Seja  $0 \leq x_N \leq x$  a alocação feita à actividade  $N$ . Então, independentemente do valor de  $x_N$ , sabemos que a restante quantidade de recurso  $x - x_N$  será usada para obter um retorno máximo das restantes  $N - 1$  actividades. Uma vez que o retorno óptimo para as  $N - 1$  actividades, começando com a quantidade  $x - x_N$  é, por definição  $f_{N-1}(x - x_N)$ , vemos que a alocação inicial de  $x_N$  à actividade  $N$  resulta num retorno total de,

$$g_N(x_N) + f_{N-1}(x - x_N). \tag{2.22}$$

A escolha óptima para  $x_N$  é obviamente a que optimiza esta função. Obtém-se assim a equação básica fundamental, típica da PD,

$$\begin{aligned} f_N(x) &= \max_{0 \leq x_N \leq x} [g_N(x_N) + f_{N-1}(x - x_N)] \\ \text{para } N &= 2, 3, \dots \quad x \geq 0 \quad \text{com } f_1(x) = g_1(x). \end{aligned} \tag{2.23}$$

A PD baseia-se então na utilização de equações funcionais e no *princípio da optimalidade*, que diz que *uma política óptima tem a propriedade de que quaisquer que sejam o estado inicial e a decisão inicial, as restantes decisões devem constituir uma política óptima, em relação ao estado que resulta da primeira decisão*. Uma demonstração deste princípio pode ser encontrada em [BellmanDreyfus62].

Como vimos pelo exemplo apresentado, a PD tem a vantagem de transformar um problema com  $N$  dimensões, em  $N$  problemas com uma dimensão.

A aplicação informática desta técnica, para o caso de variáveis contínuas, requiere uma discretização dos valores, como será aliás feito na resolução do problema em estudo neste trabalho, como veremos no capítulo seguinte.

Depois desta introdução, vamos apresentar dois procedimentos que ilustram a aplicação da PD ao RCPSP, que se distinguem da aplicação feita neste trabalho, por não contemplarem aleatoriedade.

### Procedimento de Lofts

Neste procedimento [Lofts74] existe o conceito de ‘pool’ de recursos. Os recursos são libertados quando são requeridos por actividades a iniciar e são repostos quando as actividades terminam. A quantidade de recursos é limitada, sendo necessário verificar se existe quantidade suficiente para fornecer a actividade, antes da sua libertação. É suposto que uma vez iniciada, uma actividade tenha que ser executada até ao fim, sem interrupção.

Existe o conceito de estado do sistema, que define o estado das várias actividades, da seguinte forma: para uma actividade  $i$ ,  $X(i) = -1$ , se a actividade está à espera de ser processada;  $X(i) = 0$ , se a actividade já terminou e  $X(i) = t$ , se a actividade está a ser processada e necessita ainda de  $t$  unidades de tempo para terminar. O estado do sistema é um vector com o estado das várias actividades do projecto. Assim o estado inicial será  $(-1 -1 -1 \dots -1)$ . Durante o processamento podemos ter, por exemplo  $(2 0 1 \dots -1)$ , e o estado final será  $(0 0 0 \dots 0)$ . Dá-se uma mudança de estado quando pelo menos uma actividade é totalmente processada.

São depois atribuídos custos de penalização a cada tarefa ainda não terminada, que em conjugação com os tempos de processamento, permitem criar uma função que descreve o sistema durante a procura do óptimo. O óptimo será um escalonamento que induza um menor custo total. A função é uma relação

recorrente típica de programação dinâmica, com a seguinte forma:

$$v(n + \tau, X(n + \tau)) = \min_k (r(X, k) + v(n, X(n))) \quad (2.24)$$

sujeito às restrições

do sistema

em que,

$n$  - estágio,

$X$  - estado,

$\tau$  - tempo mínimo que falta para completar uma das tarefas de  $X$ ,

$k$  - acção que permite mudar de estado (processamento de uma ou mais actividades),

$r(X, k)$  - penalização (custo) por actividades incompletas,

$v(n, X)$  - custo de alcançar o estado  $(n, X)$  de uma forma óptima.

Ou seja, o custo óptimo de um estágio é o custo óptimo do estágio anterior mais o custo de executar as actividades em curso.

### Procedimento de Elmaghraby

Esta abordagem [Elmaghraby93], que tem como objectivo minimizar o tempo de conclusão do projecto, sujeita às restrições dos recursos, considera uma rede (representação AoA) como um conjunto de caminhos com actividades em série e em paralelo. A duração de uma actividade é uma função dos recursos a ela alocados  $g_a(x_a)$ . A rede subdivide-se assim em zonas em série e zonas em paralelo. Para o conjunto das actividade em série, aplica-se a seguinte formulação:

$$f_m(r) = \min_{x_m} [g_m(x_m)], \quad 0 \leq x_m \leq r, \quad 0 \leq r \leq R \quad (2.25)$$



$$f_i(r) = \min_{x_i} [g_i(x_i) + f_{i+1}(r - x_i)], \quad 0 \leq x_i \leq r, \quad 0 \leq r \leq R, \quad i = m - 1, \dots, 1 \quad (2.26)$$

sendo  $m$  o número de arcos, ou seja, actividades;  $r$  a quantidade máxima de recursos a atribuir a cada actividade e  $R$  a quantidade total de recursos disponível.

A formulação apresentada traduz o facto de que no caso de actividades em série, a duração do projecto é a soma das durações individuais das suas actividades.

No caso de actividades em paralelo, elas podem ser executadas simultaneamente, tendo em atenção as restrições de recursos. Neste caso, a duração do projecto é igual à duração máxima das várias actividades. Assim o óptimo é assegurado da seguinte forma:

$$f_m(r) = \min_{x_m} [g_m(x_m)], \quad 0 \leq x_m \leq r, \quad 0 \leq r \leq R \quad (2.27)$$

$$f_i(r) = \min_{x_i} [\max\{g_i(x_i), f_{i+1}(r - x_i)\}], \quad 0 \leq x_i \leq r, \quad 0 \leq r \leq R, \quad i = m - 1, \dots, 1. \quad (2.28)$$

O óptimo obtém-se quando se determina  $f_1(R)$ , para ambos os casos.

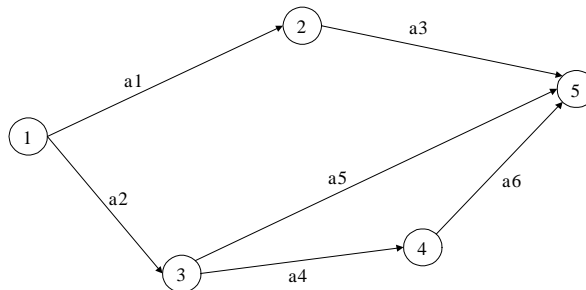


Figura 2.15: Rede de actividades AOA

Para o caso da figura 2.15 seriam geradas as seguintes equações recursivas de programação dinâmica:

$$f_6(r) = \min_{x_6} [g_6(x_6)] \quad (2.29)$$

$$f_5(r) = \min_{x_4} [g_4(x_4) + f_6(r - x_4)] \quad (2.30)$$

$$f_4(r) = \min_{x_5} [\max\{g_5(x_5), f_5(r - x_5)\}] \quad (2.31)$$

$$f_3(r) = \min_{x_2} [g_2(x_2) + f_4(r - x_2)] \quad (2.32)$$

$$f_2(r) = \min_{x_1} [g_1(x_1) + g_3(r - x_1)] \quad (2.33)$$

$$f_1(r) = \min_r [\max\{f_2(r), f_3(R - r)\}]. \quad (2.34)$$

Existem contudo redes que não são série/paralelo, como seria a do exemplo, no caso de existir uma actividade entre os nós 2 e 4.

Neste caso teria que se seguir uma das seguintes estratégias:

1. fixar a alocação de um subconjunto de actividades, resolver e no final percorrer os possíveis valores para remover a condição;
2. fazer optimização com mais de uma variável simultaneamente.

A estratégia seguida por Elmaghraby foi a primeira, remetendo-se a apresentação de mais detalhes para o referido artigo.

## 2.7 Redes estocásticas

Os métodos apresentados, com excepção do PERT, introduzido na secção 2.1.2, tratam de problemas determinísticos, ou seja, problemas onde a duração das actividades tem um valor único. Nesta secção vamos analisar alguns dos problemas que surgem quando a duração das actividades é aleatória, e que métodos existem para os resolver.

O primeiro grande problema que surge quando estamos perante uma rede estocástica é a determinação do caminho crítico. Podemos dizer mesmo que isso não é possível, pois todos os caminhos têm uma determinada probabilidade de serem críticos, dependendo da duração das várias actividades.

A abordagem utilizada pelo PERT é calcular o caminho crítico baseado nas médias das actividades, e assumir que as actividades nesse caminho devem ser as mais vigiadas. Mas pode acontecer que actividades com uma grande probabilidade de estarem no caminho crítico, não façam parte do caminho definido pelas médias.

Normalmente, no caso de redes estocásticas, não podemos falar do caminho crítico, mas sim no índice de criticalidade (CI-‘Criticality Index’). O CI é um índice associado a todos os caminhos da rede, e indica a probabilidade desse caminho ser crítico. Todos os caminhos terão uma maior ou menor probabilidade de se tornarem críticos.

Este conceito também pode ser aplicado às actividades. Assim o CI associado a uma actividade, indica qual a probabilidade dessa actividade pertencer a um caminho crítico.

A definição destes índices pode ser feita da seguinte forma [AdlakaKulkarni89]: seja  $G = (V, A)$  uma rede representativa de um projecto, onde  $A$  é o conjunto dos arcos que representam as actividades e  $V$  o conjunto dos nós que representam os eventos. Seja  $n = |A|$ ,  $m = |V|$  e  $Y_a$  a duração da actividade  $a \in A$ . Sejam  $P_1, P_2, \dots, P_J$  todos os caminhos que vão do nó inicial até ao final. O comprimento de um caminho é definido por:

$$L_i = \sum_{a \in P_i} Y_a. \quad (2.35)$$

Define-se então o CI, para um caminho, da seguinte forma:

$$\alpha(P_i) = P\{L_i \geq L_j \quad \forall j = 1, 2, \dots, J\}. \quad (2.36)$$

Para uma actividade  $a$ , define-se o CI, como se segue:

$$\alpha(a) = \sum_{i: a \in P_i} \alpha(P_i). \quad (2.37)$$

Este cálculo pode não ser fácil, pois o número de caminhos possíveis pode ser muito grande (cresce exponencialmente com o número de actividades). Esta dificuldade leva a que muitas vezes se substitua a determinação do CI, para todos os caminhos, pela determinação dos caminhos com maior CI. Isto permitirá dar uma maior atenção a estes caminhos, que são críticos.

Uma outra questão extremamente importante, quando se fala em redes estocásticas, é a determinação da duração do projecto, que está intimamente ligada à

determinação do comprimento dos caminhos, da seguinte forma [AdlakaKulkarni89]:

$$T = \max_{1 \leq i \leq J} \{L_i\}. \quad (2.38)$$

Contudo, uma vez que  $L_i$  é uma variável aleatória (*v.a.*), isso também vai acontecer a  $T$ , a duração do projecto. Assim teremos:

$$\begin{aligned} P\{T \leq t\} &= P\{\max_{1 \leq i \leq J} \{L_i\} \leq t\} \\ &= P\{L_i \leq t\} \text{ para } i = 1, 2, \dots, J. \end{aligned} \quad (2.39)$$

Quando se fala em redes estocásticas, as referências encontradas focam essencialmente a função de distribuição da conclusão do projecto ( $f(T)$ ), a média ( $\mathcal{E}(T)$ ), o desvio padrão ( $\sigma(T)$ ) e os índices de criticalidade dos caminhos e das actividades ( $\alpha(P_i)$  e  $\alpha(a)$ ). Vamos de seguida ver algumas.

Uma referência fundamental, para redes de actividades em geral, e redes estocásticas em particular, é o livro Elmaghraby ([Elmaghraby77]).

Martin ([Martin65]) introduziu uma forma exacta de calcular a distribuição do tempo de conclusão do projecto, no caso de redes série/paralelo, isto é, redes que podem ser reduzidas a dois nós e um arco, através de uma série de operações de conjugação de pares de actividades em série ou paralelo. Para este tipo de redes, há várias soluções propostas, que permitem determinar a função de distribuição da data de conclusão do projecto ([HartleyWortham66], [Ringer69], [Ringer71], [SahnerTrivedi87]).

Quando as redes deixam de ser série/paralelo, os cálculos tornam-se mais difíceis. Para uma rede genérica, os cálculos aumentam exponencialmente, com o número de actividades. Martin trata estas redes, com arcos independentes, convertendo-as em redes série/paralelo com arcos dependentes.

Fisher, Goldstein e Saisi ([FisherGoldstein83], [Fisher*et.al*85]) seguem uma abordagem condicional, onde criam o conjunto de todas as sequências de actividades possível ( $S$ ), sendo então:

$$F(t) = P(T \leq t) = \sum_{s \in S} P(T \leq t \mid s)P(s) \quad (2.40)$$

e

$$\begin{aligned} \alpha(P_i) &= P\{P_i \text{ estar no caminho mais longo}\} \\ &= \sum_{s \in S} P\{P_i \text{ ser o caminho mais longo} \mid s\}P(s) \end{aligned} \quad (2.41)$$

onde  $P(s)$  é a probabilidade das actividades serem executadas de acordo com a sequência  $s$ . A maior dificuldade é que o tamanho de  $S$  é enorme, mesmo para pequenas redes.

Para evitar estes cálculos complexos, para redes genéricas, há abordagens que determinam aproximações ou limites para estas quantidades. Por exemplo, Kleindorfer [Kleindorfer71] define recursivamente o limite inferior e superior para a distribuição de  $T$ , da seguinte forma:

$$F_1^L(t) = F_1^U(t) = 1 \quad \text{para todos os } t \geq 0 \quad (2.42)$$

$$F_j^L(t) = \prod_{i \in B_j} F_i^L \times F_{ij}(t) \quad j = 2, \dots, n \quad (2.43)$$

$$F_j^U(t) = \min_{i \in B_j} \{F_i^L \times F_{ij}(t)\} \quad j = 2, \dots, n \quad (2.44)$$

então

$$F_n^L(t) \leq F(t) \leq F_n^U(t). \quad (2.45)$$

onde  $B_j$  representa os precedentes de  $j$ .

Podemos ver outras abordagens equivalentes em [Dodin85], [Shogan77] e [Spelde77].

Este tipo de abordagens permite também determinar limites inferiores para a média de  $T$  ( $\mathcal{E}(T)$ ). Assim, em [Elmaghraby77], podemos ver que o somatório da média da duração das actividades ( $X_{ij}$ ), calculado usando CPM, é um limite inferior para a média de  $T$ , designado por  $g_n$ , e definido da seguinte forma:

$$\begin{aligned} g_1 &= 0 \\ g_j &= \max_{i \in B_j} (g_i + \mathcal{E}(X_{ij})) \quad j = 2, \dots, n \end{aligned} \quad (2.46)$$

então

$$g_n \leq \mathcal{E}(T) \quad (2.47)$$

Outros limites inferiores para  $\mathcal{E}(T)$  foram introduzidos por Fulkerson, Lindsey, Robillard e Trahan ([Fulkerson62], [Lindsey72] e [RobillardTrahan76]).

Existem também limites superiores para a média e para a variância de  $T$ . Devroye ([Devroye79]) propôs o cálculo destes limites, da seguinte forma. Seja

$$\begin{aligned} m_{ij} &= \mathcal{E}(X_{ij}) \\ \sigma_{ij}^2 &= \text{Var}(X_{ij}) \end{aligned} \quad (2.48)$$

então, recursivamente calcula-se:

$$\begin{aligned} v_1 &= 0, \quad m_1 = 0 \\ v_j &= \sum_{i \in B_j} (v_i + \sigma_{ij}^2) \\ m_j &= \max_{i \in B_j} (m_i + m_{ij}) + \{|B_j| \max_{i \in B_j} (v_i + \sigma_{ij}^2)\}^{\frac{1}{2}} \quad j = 2, \dots, n \end{aligned} \quad (2.49)$$

e mostra que,

$$\mathcal{E}(T) \leq m_n \text{ e } Var(T) \leq v_n. \quad (2.50)$$

Sculli, assumindo que as durações das actividades seguem a distribuição normal, deriva valores aproximados para a média e a variância de  $T$  [Sculli83].

Devido à dificuldade de calcular estes valores analiticamente, surgiram contribuições na área da simulação, nomeadamente [CookJennings79], [Kleindorfer71], [Kliengel66] e [Van\_Slyke63]. É neste contexto que Prisker, Happ e Whitehorse ([PritskerHapp66], [PritskerWhitehorse66]) desenvolvem o modelo GERT - ‘Graphical Evaluation and Review Technique’, onde a duração das actividades pode ser representada por qualquer distribuição. A distribuição de  $F(t)$  podia então ser calculada usando simulação de Monte Carlo. Contudo, GERT não permite calcular os CI’s. Esta limitação foi superada por PLANET - ‘Project Length Analysis and Evaluation Technique’, desenvolvida por Kennedy e Thrall ([KennedyThrall76]), que através da simulação, permitia o cálculo da probabilidade de uma actividade ser crítica, para além da distribuição de  $T$ , e da distribuição das folgas associadas às actividades.

Kleindorfer ([Kleindorfer71]) desenvolveu o MBD - ‘Method of Bounding Distribution’, que permitia estimar  $F(t)$  e  $\mathcal{E}(T)$ . Este método é aplicado a distribuições discretas. Contudo ele pode ser também aplicado a distribuições contínuas, desde que se divida o intervalo, procedendo a uma discretização fina. Isto irá contudo aumentar o tempo de computação.

Existe também outro tipo de contribuições, como por exemplo a de Elmaghraby ([Elmaghraby*et.al*98]), onde é estudado o impacto da alteração da duração média de uma actividade, na variabilidade da duração do projecto.

Foldes e Soumis abordam o problema da minimização do custo total do projecto, onde o custo de uma actividade é função da sua duração [FoldesSoumis93]. Apresentam um modelo de programação matemática básico, (2.51), que depois



reformulam, para ser mais eficiente, através da utilização de multiplicadores Lagrangianos. O modelo de programação matemática consiste no seguinte: o par  $ij$  representa uma actividade e  $t_{ij}$  a sua duração;  $t_i$  representa o instante de tempo em que começam as actividades do nó  $i$ ;  $s$  é o nó inicial e  $t$  o final; o custo de uma actividade é uma função convexa da sua duração  $c_{ij}(t_{ij})$ . Dadas estas definições, podemos calcular o custo mínimo  $c(\lambda)$ , para o qual o projecto fica concluído no instante  $\lambda$ :

$$\begin{aligned} c(\lambda) &= \min \sum_{ij \in A} c_{ij}(t_{ij}) \\ t_{ij} + t_i - t_j &\leq 0, \quad ij \in A \\ t_t - t_s &\leq \lambda \\ a_{ij} &\leq t_{ij} \leq b_{ij}, \quad ij \in A. \end{aligned} \tag{2.51}$$

Tradicionalmente a função  $c(\lambda)$  é designada por curva de custo do projecto.

Bowman ([Bowman94]) aborda um problema similar: a minimização da data de conclusão esperada do projecto, sujeita a um orçamento  $B$  ('Budget'), da seguinte forma: seja

$$\begin{aligned} X_i &= \text{comprimento do arco } i \text{ (v.a.)} \\ \theta_i &= \text{média de } X_i \\ f_i(x_i) &= \text{função densidade de probabilidade de } X_i \\ T &= \text{comprimento do caminho mais longo na rede} \\ \mathcal{E}(T) &= \text{valor esperado de } T \end{aligned} \tag{2.52}$$

então

$$T = \max_j \left( \sum_{i: \text{arco } i \text{ está no caminho } j} X_i \right) \tag{2.53}$$

e

$$\mathcal{E}(T) = \int_0^\infty \dots \int_0^\infty \max_j \left( \sum_{i: \text{arco } i \text{ está no caminho } j} X_i \right) \prod_i f_i(x_i) dx_i. \quad (2.54)$$

Assume-se que se podem gastar recursos, para reduzir  $\theta_i$ , de  $u_i$  para  $l_i$ . Seja  $Y_i$  a quantidade de que  $\theta_i$  foi reduzido, sendo  $a_i$  o custo por unidade reduzida de  $\theta_i$ .

O problema de minimização pode então ser colocado, da seguinte forma:

$$\min \mathcal{E}(T) \quad (2.55)$$

sujeito a

$$\sum_i a_i Y_i \leq B \quad (2.56)$$

$$Y_i \leq u_i - l_i.$$

Quando se fala em RCPSP estocástico, ou seja, problemas de optimização do custo do projecto, sujeito a restrições de recursos, onde as actividades são variáveis aleatórias, o número de contribuições diminui muito. Este é o tipo de problema mais próximo do que vai ser abordado nesta tese, como iremos ver nos próximos capítulos.

Uma das primeiras estratégias desenvolvidas para resolver este problema foi a estratégia ES ('Earliest Start') [Radermacher81], que se baseia na definição de escalonamentos viáveis, que respeitem as restrições de precedências e recursos, na mesma linha dos procedimentos heurísticos, introduzidos na secção 2.5, mas adaptada ao caso estocástico, e consequentemente muito mais complexa. Esta estratégia é também analisada em [IgelmundRadermacher83], que apresenta aliás mais algumas estratégias, para resolver este problema. É o caso da estratégia MES - 'Minimum over certain Earliest Start strategies' e a estratégia pré-selectiva.

Golenko-Ginz e Gonik desenvolveram um procedimento heurístico para o RCPSP com tempos de actividades estocásticos, com o objectivo de minimizar a duração esperada do projecto [Golenko-GinzGonik97]. O procedimento funciona

por estágios, onde são tomadas decisões para escalonar a próxima actividade, baseadas em restrições de precedências e de recursos. Quando existem conflitos de recursos, é utilizada uma função que permite seleccionar as próximas actividades a escalonar. Esta função representa a contribuição das actividades seleccionadas, na duração esperada do projecto. Esta contribuição é calculada como o produto da média da duração da actividade pela probabilidade dessa actividade pertencer ao caminho crítico. Os valores das probabilidades são aproximados usando valores de frequências, obtidos por simulação. O objectivo deste procedimento é dar prioridade às actividades críticas, devido ao seu grande impacto na duração esperada do projecto.

## 2.8 Apresentação do problema estudado

Depois de feita uma introdução aos aspectos fundamentais da gestão de projectos, podemos situar o problema estudado na fase do planeamento do projecto, mais concretamente na questão da atribuição de recursos às actividades. Assim, é preocupação deste trabalho a alocação óptima dos recursos às actividades de um projecto, por forma a minimizar o custo total, em presença de incerteza. Assume-se que a estrutura do projecto é conhecida, ou seja, estão já definidas as actividades e as suas relações de precedência. Desta forma é possível representar o projecto como uma rede de actividades AoA<sup>4</sup>. Considera-se que existe um único recurso, em quantidade suficiente, por uma questão de simplicidade.

Um dos aspectos que distingue este trabalho de contributos anteriores para a teoria das redes de actividades é a nossa perspectiva sobre a incerteza. A perspectiva normalmente utilizada é a de associar a incerteza de um projecto à duração das actividades (PERT). Nós consideramos que a incerteza não está na duração das actividades, mas sim no seu conteúdo de trabalho. Quando se diz que uma actividade demora 4 dias a ser executada, já se está a ter em conta

---

<sup>4</sup>Para detalhes da representação AoA, ver secção 2.1.2

outros compromissos que possam eventualmente existir, de tal forma que o esforço dedicado a essa actividade é, por exemplo, 2 horas por dia. Assim, a chave para a decisão sobre a duração da actividade é a estimativa do conteúdo de trabalho a ela associado, neste caso 8 homens-hora. Se existir a necessidade de terminar esta actividade em menos de 4 dias, a reacção imediata será deixar outras tarefas pendentes, para dedicar mais tempo à actividade em causa, ou seja, modificar a alocação dos recursos, para dar resposta a este requisito. Um pedido para terminar a actividade em 2 dias resultaria imediatamente numa duplicação do esforço original. É segundo esta perspectiva que um gestor de projectos manipula os seus recursos, por forma a alcançar os objectivos especificados. Conhecendo o conteúdo total de trabalho das actividades, os recursos são atribuídos por forma a obter as durações pretendidas. A incerteza surge precisamente da dificuldade de definir o conteúdo de trabalho de uma actividade. Então podemos dizer que são necessários entre 8 a 10 homens-hora, para executar uma actividade, por exemplo.

Podemos distinguir dois tipos de actividades: as actividades unimodais, em que a necessidade de recursos e a duração são fixos à priori (p.e. são necessários 2 homens a trabalhar 6 dias), e as actividades multimodais, que podem ser executadas com diferentes alocações de recursos, sendo a duração uma função dessa alocação (p.e. o conteúdo de trabalho total de 12 homens-dia, pode ser executado por 1 homem em 12 dias, 2 homens em 6 dias ou mesmo por 1.5 homens em 8 dias). O modelo que vai ser proposto considera actividades multimodais.

Mais especificamente, o problema pode ser definido da seguinte forma: dada uma rede de actividades AoA, existe associado a cada actividade  $a$  um conteúdo de trabalho ('Work content') designado por  $W_a$ . Assume-se que esse conteúdo de trabalho é aleatório e segue a distribuição exponencial,

$$W_a \sim \exp(\lambda_a), \quad \forall a \in A, \quad (2.57)$$

em que  $A$  é o conjunto das actividades. Esta escolha foi feita por uma questão de simplicidade no cálculo do valor esperado ( $1/\lambda_a$ ). Considera-se só um recurso, em que  $x_a$  representa a quantidade de recurso alocado à actividade  $a$ ,

$$0 \leq l_a \leq x_a \leq u_a < \infty, \forall a \in A, \quad (2.58)$$

onde  $l_a$  é um limite inferior da alocação de recursos e  $u_a$  um limite superior.

Quando se atribui uma quantidade de recurso  $x_a$  à actividade  $a$ , incorre-se num custo de recurso igual a

$$C_a = x_a W_a, \quad (2.59)$$

o que resulta na duração

$$Y_a = \frac{W_a}{x_a}. \quad (2.60)$$

Como  $W_a$  é uma variável aleatória, o mesmo acontece com  $C_a$  e  $Y_a$ .

Assume-se também que o projecto tem uma data de conclusão especificada  $T$ , e uma penalização por atrasos, que é função da data real de conclusão do projecto, designada por  $c(\Upsilon_n - T)$ , onde  $\Upsilon_n$  é a data de realização do nó  $n^5$ , uma *v.a.*. Por uma questão de simplicidade, considera-se que a função de penalização é proporcional ao atraso, com uma constante de proporcionalidade  $c_L$  (custo do atraso por unidade de tempo)

$$c(\Upsilon_n - T) = c_L \times \max\{0, \Upsilon_n - T\}. \quad (2.61)$$

---

<sup>5</sup> Assume-se que os nós da rede estão ordenados topologicamente, sendo o nó 1 o nó de início e o nó  $n$  o último nó.

O objectivo da análise é determinar o vector de alocação de recursos  $X$ , para todas as actividades do projecto, de tal forma que o custo total esperado, de executar as actividades e ser penalizado pelo atraso, seja minimizado,

$$\min_{\bar{X}} \mathcal{E} \{C(X) + c_L \times \max\{0, \Upsilon_n - T\}\}, \quad (2.62)$$

onde  $C(X)$  é uma *v.a.* dada por

$$C(X) = \sum_{a \in \mathcal{A}} x_a W_a, \quad (2.63)$$

e cada  $x_a$  obedece à restrição (2.58).

# Capítulo 3

## O modelo de programação dinâmica

### 3.1 Apresentação do modelo

A aplicação do formalismo de programação dinâmica requiere a representação da função a otimizar (2.62), como uma função recorrente, ou por estágios. Como já vimos na secção 2.8, o objectivo deste trabalho é desenvolver um modelo que permita minimizar o custo total associado a um projecto, dando indicação de qual o valor óptimo da alocação dos recursos às actividades, para que o objectivo seja cumprido.

Seja  $A$  o conjunto de todas as actividades. Para conseguir obter um modelo viável, foi necessário impor a existência de uma única variável a otimizar, em cada estágio. Para tal foi necessário fixar temporariamente o valor das restantes variáveis.

Vamos supor que somos capazes de identificar um subconjunto de actividades a ser “condicionado”, no sentido em que a sua alocação de recursos é fixada. Esse subconjunto pode então ser removido do conjunto das variáveis de decisão. Mas este não é um subconjunto qualquer; ele deve permitir que cada

$udc^1$  - ‘uniformly directed cutset’ da rede contenha exactamente uma variável de decisão. A este subconjunto de variáveis fixas vamos chamar  $\mathcal{F}$ . Naturalmente, o condicionamento terá que ser removido no final, enumerando todas as possíveis alocações de  $\mathcal{F}$  e seleccionando a melhor.

O procedimento utilizado para identificar os  $udc$ 's de uma rede de actividades e a sua intersecção, que é equivalente ao conjunto  $\mathcal{F}$ , é apresentado na secção 3.2.

Seja  $\mathcal{D} \equiv \overline{\mathcal{F}}$  o conjunto das variáveis de decisão. Definimos o *estágio* do formalismo de PD como o momento da decisão sobre o valor de  $x_a$ , para cada actividade  $a \in \mathcal{D}$ . Em cada estágio da PD, é feita a optimização de uma única variável de decisão. Desta forma haverá tantos estágios quantas as variáveis do conjunto  $\mathcal{D}$ . Vamos supor que há  $K$  estágios, logo  $K = |\mathcal{D}| = |A| - |\mathcal{F}|$ .

Neste modelo existe também o conceito de estado. Definimos o *estado* como o vector de tempos de realização de um conjunto de nós, que permitem tomar a decisão  $x_a$  e calcular o contributo do estágio, com  $a \in \mathcal{D}$ . O estado permite representar a evolução do projecto no tempo e as decisões tomadas irão reflectir esse estado. Por exemplo, se num determinado estágio do projecto, o estado for tal que possa provocar um atraso, a quantidade de recursos atribuída irá ser maior, para tentar repor o projecto no seu rumo.

O estado no estágio  $k$  será genericamente designado por  $s_k$ . Tipicamente,  $s_k = (t_{i_1}, \dots, t_{i_r})$  para um subconjunto de nós  $(i_1, \dots, i_r)$ .

Vamos numerar os estágios para trás, de tal forma que “estágio  $k$ ” signifique que faltam  $k$  estágios para completar o projecto, e vamos identificar a variável de decisão no estágio  $k$  por  $x_{[k]}$ . Assim, o “estágio 1” é o estágio que contém o nó terminal  $n$ , e a variável de decisão  $x_{[1]}$ , e o “estágio  $K$ ” é o estágio que contém o nó inicial 1 e a variável de decisão  $x_{[K]}$ .

Para todos os estágios excepto o primeiro (que contém o nó terminal  $n$ ) o *contributo do estágio* é simplesmente o custo do recurso, uma *v.a.* igual a  $x_{[k]}W_{[k]}$ .

---

<sup>1</sup>Um  $udc$  é um corte em que todos os arcos estão dirigidos de um conjunto de nós que contém o nó inicial (1) para um conjunto de nós que contém o nó terminal ( $n$ ). Representa as possíveis combinações de actividades activas ao longo da vida do projecto.



No estágio 1, o *contributo do estágio* é a soma do custo da utilização de recursos ( $= x_{[1]}W_{[1]}$ ) com o custo do atraso, se existir ( $= c_L \times \max\{0, \Upsilon_n - T\}$ ), sendo ambas *v.a.*'s. Seja  $f_k(s_k | \mathcal{F})$  o custo mínimo no estágio  $k$ , quando o estado é  $s_k$ , condicionado pelas alocações fixas de  $\mathcal{F}$ . Então, como acontece tipicamente em PD,

$$f_k(s_k | \mathcal{F}) = \min_{x_{[k]} \in D} \mathcal{E} \{C_{[k]}([x_{[k]}], s_k) + \mathcal{E} f_{k-1}(S_{k-1} | \mathcal{F})\}. \quad (3.1)$$

$S_{k-1}$  é uma *v.a.* que representa o tempo de realização do estado  $s_{k-1}$ , para  $k = 2, \dots, K$ ;  $x_{[k]}$  é a variável de decisão do estágio  $k$ , e  $f_1(s_1 | \mathcal{F})$  é determinado pela fórmula (3.4) abaixo. O óptimo é calculado removendo o condicionamento, e a solução é obtida por

$$f(s_K = 0) = \min_{\mathcal{F}} f_K(s_K | \mathcal{F}). \quad (3.2)$$

A solução via PD gera uma política que prescreve a alocação de recursos óptima, sob todos os possíveis estados do projecto, à medida que ele avança no tempo.

Um processo mais rápido (mas não necessariamente óptimo) para seleccionar as actividades a serem condicionadas (o conjunto  $\mathcal{F}$ ) é o seguinte<sup>2</sup>:

1. Determinar o caminho mais longo na rede (por *número de actividades*). As actividades ao longo desse caminho serão as variáveis de decisão (conjunto  $\mathcal{D}$ ).
2. As outras actividades,  $A - \mathcal{D}$ , serão as actividades a ser fixadas (conjunto  $\mathcal{F}$ ).

---

<sup>2</sup>O processo óptimo para seleccionar o conjunto  $\mathcal{F}$  é apresentado na secção 3.2.

A aplicação do modelo de PD faz-se da seguinte forma: é calculado o custo esperado do recurso para as variáveis fixas, designado por  $rcf$  ('Resource Cost of Fixed variables'),

$$rcf = \mathcal{E} \sum_{i \in \mathcal{F}} x_i W_i = \sum_{i \in \mathcal{F}} x_i \times \mathcal{E}(W_i). \quad (3.3)$$

O primeiro estágio começa no estado  $s_1$ , ponto de partida para o último nó da rede. Calcula-se

$$f_1(s_1 | \mathcal{F}) = rcf + \min_{x_{[1]} \in D} \mathcal{E} \{x_{[1]} W_{[1]} + c_L \times \mathcal{E}(U)\} \quad (3.4)$$

onde

$$U = \max\{0, \Upsilon_n - T\} \quad (3.5)$$

Depois do valor esperado do estágio 1 ser calculado, aplica-se a equação (3.1) até alcançar o último estágio (correspondente ao nó inicial da rede), onde o estado é  $s_K = t_1 = 0$ .

O resultado é a indicação da melhor alocação para a primeira actividade  $x_{[k]}$  que parte do nó 1. Este procedimento é depois repetido para todas as possíveis alocações das actividades no conjunto  $\mathcal{F}$ . Uma vez que  $x_a$  está definido no intervalo  $[l_a, u_a]$ , temos que recorrer a *discretização*. Seja  $m$  o número de diferentes valores de alocação de uma actividade do conjunto  $\mathcal{F}$ . Então existem  $m^{|\mathcal{F}|}$  possíveis alocações, de entre as quais é seleccionado o óptimo. Depois da obtenção do óptimo, ou seja, o mínimo custo esperado, fica disponível o valor óptimo da alocação de recursos a atribuir às actividades que partem do primeiro nó da rede. A política a seguir daí para a frente depende naturalmente do estado do processo quando for necessário tomar a decisão. Esta é a essência da natureza adaptativa da abordagem por PD: futuras alocações têm que aguardar a realização das actividades precedentes, à medida que o projecto avança no tempo. Não é difícil constatar o peso computacional deste modelo de PD.

## 3.2 UDC's / CII

### 3.2.1 Identificação dos *udc*'s

O procedimento para identificar os *udc*'s ([KulkarniAdlakha86], [Sigalet.al79], [Elmaghraby00]) numa rede de actividades AoA é o seguinte: começar pelo último nó  $n$ , e recuar interactivamente até chegar ao nó inicial 1. Para cada nó  $j$ ,  $j = n, \dots, 1$ :

1. Determinar uma sub-rede de actividades que preceda o nó  $j$  ( $X_j$ ); a rede residual será  $\overline{X}_j = N - X_j$ . O *udc* será o conjunto de arcos que une  $X_j$  a  $\overline{X}_j$ .  $X_j$  será uma sub-rede que contém o nó inicial e  $\overline{X}_j$  uma sub-rede que contém o nó final<sup>3</sup>.
2. Para cada nó  $i$  não relacionado com  $j$  (isto é, que não o precede nem o sucede), constituir uma sub-rede que preceda a união  $X_{\{i,j\}} = \{j\} \cup \{i\}$ . Definir um novo *udc* como no ponto (1).
3. Repetir (2) para cada conjunto de nós não relacionados com  $j$ , agrupando-os 2 a 2, 3 a 3, etc, até não haver mais possibilidades. Definir novos *udc*'s como em (1).
4. Parar quando se alcançar o nó 1.

Ordenar os *udc*'s por ordem crescente do número de actividades. O primeiro 'cutset' de uma actividade é o *udc* de menor índice que contém a actividade. Para aplicar este procedimento, vamos considerar o projecto que irá ser estudado no decorrer deste trabalho (figura 3.1).

Para este caso em particular, o procedimento que gera os *udc*'s está representado na tabela 3.1, e os *udc*'s gerados estão representados na figura 3.1.

---

<sup>3</sup> $X_j + \overline{X}_j = N$

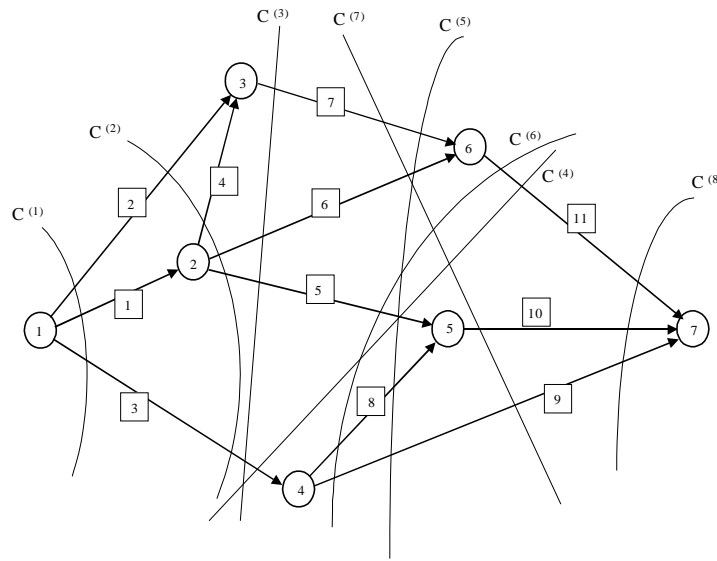


Figura 3.1: Projecto exemplo com os seus udc's

Tabela 3.1: Geração dos udc's do projecto da fig. 2.1

Node	Set $X^\ddagger$	Set $\bar{X}^\ddagger$	Cutset <sup>†</sup>	Identification
7	{1, 2, 3, 4, 5, 6}	{7}	{9, 10, 11}	$\mathcal{C}^{(8)}$
6	{1, 2, 3, 4, 5}	{6, 7}	{6, 7, 9, 10}	$\mathcal{C}^{(7)}$
5	{1, 2, 3, 4, 6}	{5, 7}	{5, 8, 9, 11}	$\mathcal{C}^{(6)}$
	{1, 2, 3, 4}	{5, 6, 7}	{5, 6, 7, 8, 9}	$\mathcal{C}^{(5)}$
4	{1, 2, 3, 6}	{4, 5, 7}	{3, 5, 11}	$\mathcal{C}^{(4)}$
	{1, 2, 3}	{4, 5, 6, 7}	{3, 5, 6, 7}	$\mathcal{C}^{(3)}$
3	{1, 2}	{3, 4, 5, 6, 7}	{2, 3, 4, 5, 6}	$\mathcal{C}^{(2)}$
2	{1}	{2, 3, 4, 5, 6, 7}	{1, 2, 3}	$\mathcal{C}^{(1)}$

<sup>‡</sup> Identificação pelos nós

<sup>†</sup> Identificação pelas actividades.

O primeiro 'cutset' de cada actividade é então:

<b>Actividade:</b>	1	2	3	4	5	6	7	8	9	10	11
<b>Primeiro <i>udc</i>:</b>	$\mathcal{C}^{(1)}$	$\mathcal{C}^{(1)}$	$\mathcal{C}^{(1)}$	$\mathcal{C}^{(2)}$	$\mathcal{C}^{(2)}$	$\mathcal{C}^{(2)}$	$\mathcal{C}^{(3)}$	$\mathcal{C}^{(5)}$	$\mathcal{C}^{(5)}$	$\mathcal{C}^{(7)}$	$\mathcal{C}^{(4)}$

### 3.2.2 Determinação do *cii*

Para determinar o *cii* - ‘*cutset intersection index*’, tem que se começar por gerar os *udc*'s da rede. Constrói-se depois a matriz de incidência actividade-*udc* ( $M$ ), em que  $m_{ij} = 1$  se a actividade  $i$  fica no *udc*  $j$ , e  $m_{ij} = 0$  caso contrário.

$$\begin{array}{c}
 \text{Cutset} \\
 \begin{array}{cccccccc}
 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8
 \end{array} \\
 \begin{array}{c}
 1 \\
 2 \\
 3 \\
 4 \\
 5 \\
 6 \\
 7 \\
 8 \\
 9 \\
 10 \\
 11
 \end{array}
 \begin{array}{c}
 \left[ \begin{array}{cccccccc}
 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\
 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 \\
 0 & 1 & 1 & 0 & 1 & 0 & 1 & 0 \\
 0 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\
 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\
 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\
 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\
 0 & 0 & 0 & 1 & 0 & 1 & 0 & 1
 \end{array} \right]
 \end{array}
 \begin{array}{c}
 \bigcirc \\
 \leftarrow \\
 \leftarrow \\
 \bigcirc \\
 \leftarrow \\
 \leftarrow \\
 \bigcirc \\
 \leftarrow \\
 \leftarrow \\
 \leftarrow \\
 \bigcirc
 \end{array}
 \end{array}
 \tag{3.6}$$

O objectivo é deixar por fixar o máximo número de actividades possível, o que se traduz em apagar o mínimo número de linhas na matriz  $M$ , de tal forma que a matriz residual contenha exactamente um 1 em cada coluna. Este problema é conhecido como o SPP - ‘Set Partitioning Problem’, e representa-se da seguinte forma:

$$\max \quad z = \sum_i x_i \quad (3.7)$$

de tal forma que,

$$\sum_i m_{ij} x_i = 1, \quad \forall j, \quad (3.8)$$

$$x_i \in \{0, 1\},$$

sendo  $x_i$  uma variável binária, que indica se a actividade  $i$  é ou não fixada:  $x_i = 0$  (actividade fixada),  $x_i = 1$  (actividade não fixada).

Na prática, isto pode ser feito da seguinte forma: para cada par de *udc*'s consecutivo, é fixado o subconjunto de actividades que se intersectam, isto é, as actividades que são comuns aos dois *udc*'s (indicado com uma seta ( $\leftarrow$ ) na matriz  $M$ ). Isto define o IS - 'Intersection Set' das actividades, que é o conjunto das actividades a ser fixado (representado no nosso formalismo de PD por  $\mathcal{F}$ ), e o tamanho deste conjunto define o *cii*. No caso do exemplo,

$$IS = \mathcal{F} = \{2, 3, 5, 6, 8, 9, 10\} \Rightarrow cii = 7. \quad (3.9)$$

Este procedimento faz com que em cada *udc* fique só uma actividade, como se pode ver a seguir:

$$M = \begin{array}{c} \text{Cutset} \\ \begin{array}{cccccccc} & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \\ \begin{array}{l} 1 \\ 4 \\ 7 \\ 11 \end{array} \text{Actividade} & \left[ \begin{array}{cccccccc} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 0 & 1 \end{array} \right] \end{array} \quad (3.10)$$

Afortunadamente, todo este procedimento pode ser substituído pela detecção do caminho mais longo na rede, por número de actividades, fixando as actividades que não façam parte desse caminho<sup>4</sup>. No caso do nosso exemplo, o caminho mais longo é 1-4-7-11. Então IS (ou  $\mathcal{F}$ ) será constituído pelas restantes actividades, como em (3.9).

### 3.3 Projecto exemplo

Considere a rede da figura 3.2.

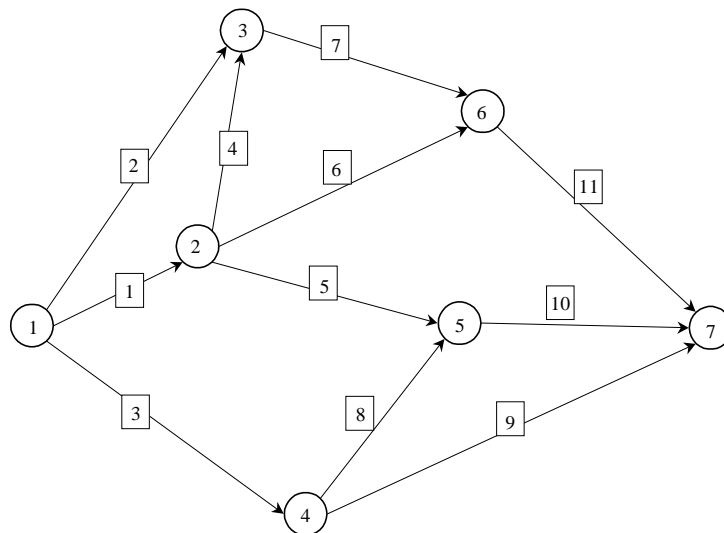


Figura 3.2: Projecto exemplo

Vamos admitir que o conteúdo de trabalho (‘Work Content’) de cada actividade segue a distribuição exponencial  $W_a \sim \exp(\lambda_a)$ , com os parâmetros apresentados na tabela 3.2.

<sup>4</sup>Agradecemos ao Professor Bert Deryck da London School of Business por sugerir este procedimento elementar para calcular as variáveis a fixar.

A data de conclusão do projecto é  $T = 65^5$  e o custo do atraso é  $c_L = 5$  por unidade de tempo (neste caso semanas) em atraso. Os valores da alocação de recursos  $x_a$  são limitados ao seguinte intervalo:  $0.5 \leq x_a \leq 1.5$ .

O custo da alocação de recursos para cada actividade será  $c(a) = x_a W_a$  e a sua duração  $Y_a = \frac{W_a}{x_a}$ .

Em primeiro lugar determina-se o caminho mais longo na rede, como se pode ver na figura 3.3, representado com linhas mais grossas.

Tabela 3.2: Parâmetros da distribuição exponencial para todas as actividades do projecto

Actividade $a$	Parâmetro $\lambda_a$	Qt. Trabalho Esperada
<b>1</b>	0.1	10.0
<b>2</b>	0.12	8.33
<b>3</b>	0.05	20.0
<b>4</b>	0.08	12.5
<b>5</b>	0.2	5.0
<b>6</b>	0.04	25.0
<b>7</b>	0.03	33.33
<b>8</b>	0.04	25.0
<b>9</b>	0.024	41.67
<b>10</b>	0.15	6.67
<b>11</b>	0.16	6.25

As actividades pertencentes ao caminho mais longo serão as variáveis de decisão:  $\mathcal{D} = \{x_1, x_4, x_7, x_{11}\}$ . O conjunto das actividades fixas é o conjunto

<sup>5</sup>A estimativa PERT do valor esperado para a duração do projecto, assumindo uma alocação de recursos nominal, para todas as actividades, é 63.33, que como sabemos subestima a verdadeira duração esperada (ver expressões (2.46) e (2.47)). Assim, a data de conclusão especificada é ligeiramente maior que a duração esperada do projecto.



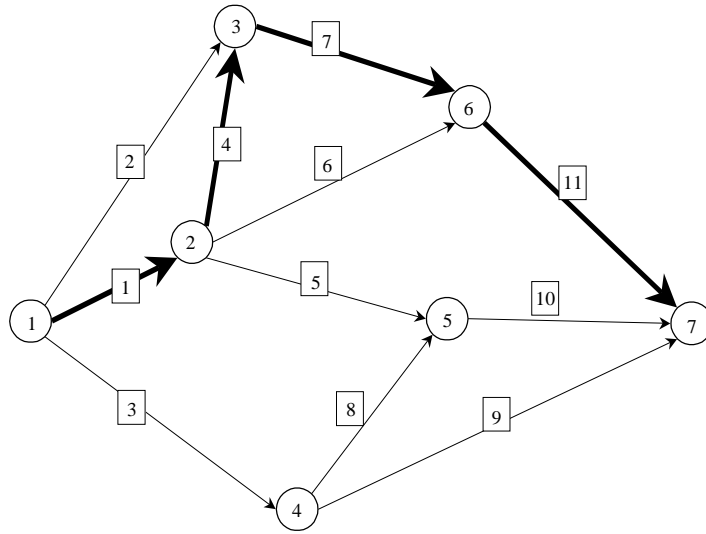


Figura 3.3: Caminho mais longo

$\mathcal{F} = \{\hat{x}_2, \hat{x}_3, \hat{x}_5, \hat{x}_6, \hat{x}_8, \hat{x}_9, \hat{x}_{10}\}$ <sup>6</sup>, nas quais se assinala a natureza fixa das suas alocações com um “^” sobre o  $x$ .

Para ilustrar a aplicação deste modelo, o intervalo da alocação de recursos  $[0.5, 1.5]$  foi discretizado em cinco valores: 0.5, 0.75, 1.0, 1.25, 1.5, sendo estes os valores possíveis para cada recurso. Guiados pelos valores esperados das actividades com a alocação “nominal” de recursos  $\hat{x}_a = 1.0$ , a alocação das actividades do conjunto  $\mathcal{F}$  foi fixada em

$$\{\hat{x}_2, \hat{x}_3, \hat{x}_5, \hat{x}_6, \hat{x}_8, \hat{x}_9, \hat{x}_{10}\} = \{1.0, 1.5, 0.5, 0.5, 1.0, 1.5, 1.0\}. \quad (3.11)$$

Significa que a alocação à actividade 2 foi fixada em  $\hat{x}_2 = 1.0$ , à actividade 3 em  $\hat{x}_3 = 1.5$ , *etc.*

<sup>6</sup>Reparar na igualdade entre este conjunto e o obtido utilizando o procedimento da secção 3.2.

O cálculo do *valor esperado do custo dos recursos* para as actividades fixas é facilmente calculado, sendo

$$rcf = \sum_{a \in \mathcal{F}} \hat{x}_a \times \mathcal{E}(W_a) = 147.5. \quad (3.12)$$

Mais uma vez, e por uma questão de ilustração, o conteúdo de trabalho das actividades  $W_a$  foi discretizado em quatro valores, cada um com probabilidade 0.25. Por exemplo, para  $W_1 \sim \exp(0.1)$  assumiu-se que só poderia ter quatro valores: 1.37, 4.77, 10.00, 23.86, todos com igual probabilidade<sup>7</sup>.

As iterações da PD iniciam no **estágio 1**, que é definido pela variável de decisão  $x_{[1]} = x_{11}$ , a alocação a atribuir à actividade 11. O estado pode ser definido pelo conjunto  $s_1 = (t_4, t_5, t_6)$ , os tempos de realização dos nós 4, 5 e 6. Então temos

$$f_1(t_4, t_5, t_6 | \mathcal{F}) = rcf + \min_{x_{11}} \mathcal{E} \{x_{11}W_{11} + c_L \mathcal{E}(U)\}, \quad (3.13)$$

onde

$$U = \max\{0, \Upsilon_7 - T\}, \quad (3.14)$$

e

$$\Upsilon_7 = \max\left\{t_6 + \frac{W_{11}}{x_{11}}, t_4 + \frac{W_9}{\hat{x}_9}, t_5 + \frac{W_{10}}{\hat{x}_{10}}\right\}. \quad (3.15)$$

O **estágio 2** é definido pela variável de decisão  $x_{[2]} = x_7$ . O seu estado é definido pelo conjunto  $s_2 = (t_2, t_3, t_4)$ , sendo,

$$f_2(t_2, t_3, t_4 | \mathcal{F}) = \min_{x_7} \mathcal{E} \{x_7 W_7 + \mathcal{E}[f_1(t_4, \Upsilon_5, \Upsilon_6)]\} \quad (3.16)$$

---

<sup>7</sup>Observe a igualdade entre a média da distribuição discreta  $(1.37 + 4.77 + 10.00 + 23.86)/4 = 10$  e a média teórica  $1/0.1 = 10$ .

onde

$$\Upsilon_5 = \max\left\{t_2 + \frac{W_5}{\hat{x}_5}, t_4 + \frac{W_8}{\hat{x}_8}\right\} \quad \text{e} \quad \Upsilon_6 = \max\left\{t_2 + \frac{W_8}{\hat{x}_8}, t_3 + \frac{W_7}{x_7}\right\}. \quad (3.17)$$

Para o **estágio 3**, definido pela variável de decisão  $x_{[3]} = x_4$ , o estado é definido por  $(t_2)$ , e

$$f_3(t_2 | \mathcal{F}) = \min_{x_4} \mathcal{E}\{x_4 W_4 + \mathcal{E}[f_2(t_2, \Upsilon_3, \Upsilon_4)]\} \quad (3.18)$$

com

$$\Upsilon_3 = \max\left\{t_2 + \frac{W_4}{x_4}, \frac{W_2}{\hat{x}_2}\right\} \quad \text{e} \quad \Upsilon_4 = \frac{W_3}{\hat{x}_3} \quad (3.19)$$

Finalmente, o **estágio 4**, que é definido pela variável  $x_{[4]} = x_1$ , tem o estado definido por  $t_1 = 0$ :

$$f_4(t_1 = 0 | \mathcal{F}) = \min_{x_1} \mathcal{E}\{x_1 W_1 + \mathcal{E}[f_3(\Upsilon_2)]\} \quad (3.20)$$

com

$$\Upsilon_2 = \frac{W_1}{x_1}. \quad (3.21)$$

A aplicação deste modelo permite obter os seguintes resultados:

$$x_1^* | \mathcal{F} = 1.25, \quad \text{com um custo total esperado de 348.28,}$$

para  $\mathcal{F} = \left\{ \begin{array}{l} \{\hat{x}_2, \hat{x}_3, \hat{x}_5, \hat{x}_6, \hat{x}_8, \hat{x}_9, \hat{x}_{10}\} = \\ \{1.0, 1.5, 0.5, 0.5, 1.0, 1.5, 1.0\} \end{array} \right\} \quad (3.22)$

Uma vez que cerca de 148 unidades deste custo se devem aos recursos alocados às actividades *fixas*, significa que para esta alocação do conjunto  $\mathcal{F}$ , o valor esperado da alocação de recursos para as quatro variáveis de decisão  $\{x_1, x_4, x_7, x_{11}\}$

mais o custo do atraso é cerca de 201. Uma vez que o custo esperado da alocação de recursos às variáveis de decisão varia entre  $(0.5 \times 62.083) \simeq 31$  e  $(1.5 \times 62.083) \simeq 93$ , deduz-se que o custo esperado do atraso esteja entre 108 e 170. Uma vez que o custo marginal do atraso é de 5, prevê-se que o projecto esteja atrasado no mínimo 21.6 semanas e no máximo 34 semanas.

Para remover o condicionamento de  $\mathcal{F}$ , teríamos que calcular o custo esperado e o valor óptimo das variáveis de decisão, para todas as possíveis combinações das variáveis fixas, existindo  $5^7 = 78125$  possibilidades. Como este cálculo se mostrou impraticável em termos de tempo de computação, optou-se por reduzir o número de valores possíveis, para as variáveis de  $\mathcal{F}$  a três  $\{0.5, 1.0, 1.5\}$ , originando assim  $3^7 = 2187$  enumerações, o que já foi razoável. O resultado desta procura (ver apêndice D), obtido ao fim de 141 minutos, foi o seguinte:

$$x_1^* | \mathcal{F}^* = 1.25, \quad \text{com um custo total esperado de } 280.85, \\ \text{com } \mathcal{F}^* = \left\{ \begin{array}{l} \{x_2^*, x_3^*, x_5^*, x_6^*, x_8^*, x_9^*, x_{10}^*\} = \\ \{0.5, 1.5, 0.5, 1.0, 1.0, 1.5, 1.0\} \end{array} \right\} \quad (3.23)$$

Assim, a alocação óptima para o *udc* inicial, composto pelas actividades 1, 2, 3, é dado por

$$\{x_1^*, x_2^*, x_3^*\} = \{1.25, 0.5, 1.5\}. \quad (3.24)$$

Está assim disponível o vector completo das alocações para as actividades fixas, que levaram ao óptimo (apresentado em (3.23)), bem como as políticas óptimas correspondentes (com esta alocação das variáveis fixas) nos restantes estágios do projecto. Estas políticas óptimas estão representadas no apêndice D, considerando já o resultado da análise de sensibilidade, feito na secção 3.4.

Os valores das alocações às outras variáveis de decisão (actividades 4, 7, e 11) dependem naturalmente do resultado das três actividades do primeiro *udc*. Assim,

- (i) Se a actividade 1 terminar primeiro, então será conhecido o tempo de realização do nó 2. Esta informação em conjugação com as alocações especificadas

para as variáveis fixas, iria permitir localizar a alocação óptima para  $x_4^*$ , na política óptima previamente desenvolvida para o estágio 3.

- (ii) Se a actividade 2 terminar primeiro, então deverá ficar “dormente” a aguardar a finalização da actividade 4, que por seu lado não pode iniciar sem que 1 tenha terminado.
- (iii) Se a actividade 3 terminar primeiro, então o nó 4 será realizado. Podemos iniciar as actividades 8 e 9 com as alocações especificadas no vector das alocações fixas  $\mathcal{F}$ .

A alocação óptima tem um custo de recursos esperado de 168.34<sup>8</sup>. Note que quando falamos do óptimo, não estamos a falar de um óptimo absoluto, uma vez que a procura pelo espaço de  $W$  e  $X$  foi limitado aos valores seleccionados para discretização. Para não cometer uma imprecisão, vamos passar a referir este óptimo entre aspas.

Um limite inferior para o custo dos recursos das restantes três variáveis de decisão é  $0.5 (\mathcal{E} [W_4] + \mathcal{E} [W_7] + \mathcal{E} [W_{11}]) = 26.04$ , sendo o limite superior de  $1.5 (\mathcal{E} [W_4] + \mathcal{E} [W_7] + \mathcal{E} [W_{11}]) = 78.12$ . Então, o limite inferior do custo dos recursos é 194.38 e o limite superior 246.46. Assim, o custo do atraso varia entre  $(280.85 - 246.46) = 34.39$  e  $(280.85 - 194.38) = 86.47$ . Uma vez que a penalização unitária por atrasos é de 5, isto implica que o atraso esperado sob a alocação óptima varia entre 6.88 semanas e 17.29 semanas. Podemos observar que a redução no custo se deve essencialmente à redução no valor esperado do atraso.

### 3.4 Análise de sensibilidade

Podemos ver uma experiência factorial completa (com  $5^7$  diferentes alocações para as actividades do conjunto  $\mathcal{F}$ ) como uma forma de análise de sensibilidade. Mas,

---

<sup>8</sup>  $x_1^* \cdot \mathcal{E} [W_1] + \sum_{k \in \mathcal{F}} \hat{x}_k \cdot \mathcal{E} [W_k]$ .

depois de conseguido o vector de alocações “óptimo”  $(x_1^*, x_2^*, x_3^*, x_5^*, x_6^*, x_8^*, x_9^*, x_{10}^*)$ , podemos ter interesse em testar a sensibilidade do custo total a pequenas alterações nas variáveis.

Uma forma simples de fazer análise de sensibilidade é fazer variações uma-a-uma, que pode ser descrita como se segue:

para cada  $x_k^* \in \mathcal{F}$ , avaliar a performance do custo do projecto, em cada um dos valores “vizinhos”.

Há quatro possíveis resultados para o custo. Pode ser:

1. monótono crescente: então diminuir  $x^*$  se possível<sup>9</sup>;
2. monótono decrescente: então aumentar  $x^*$  se possível<sup>10</sup>;
3. em forma de “ $\vee$ ”: então testar ambos os lados de  $x^*$ ;
4. em forma de “ $\wedge$ ”: então testar ambos os lados de  $x^*$ .

A análise de sensibilidade foi feita do seguinte modo: para cada  $x_a \in \mathcal{F}$ , fixou-se o valor 0.5, 1.0 ou 1.5, mantendo todos as outras alocações a variar. Isto resultou em três valores diferentes para o custo total esperado, como se pode ver na tabela 3.3. Por exemplo, para  $x_2 = 0.5$ , o custo total esperado (*CTE*) foi de 280.85. Já para  $x_2 = 1.0$ ,  $CTE = 285.01$ . Finalmente para  $x_2 = 1.5$ ,  $CTE = 283.80$ . O mesmo procedimento repetiu-se para as restantes variáveis fixas.

De seguida, passou-se a analisar o comportamento da curva do *CTE*, quando cada  $x_a \in \mathcal{F}$  varia. O valor mínimo do *CTE*, em cada coluna, está enfatizado, e sugere a melhor opção para o  $x_a$  correspondente.

Da análise do comportamento da curva, tiraram-se as seguintes conclusões.

---

<sup>9</sup>A impossibilidade de diminuir o valor ocorreria se  $x_a^* = l_a$ .

<sup>10</sup>A impossibilidade de aumentar o valor ocorreria se  $x_a^* = u_a$ .

Tabela 3.3: Variação do CTE para diferentes valores das variáveis fixas

CTE		$x_2^*$	$x_3^*$	$x_5^*$	$x_6^*$	$x_8^*$	$x_9^*$	$x_{10}^*$
x's	0.5	<b>280.85</b>	339.95	<b>280.85</b>	344.11	324.87	424.61	286.99
	1.0	285.01	281.11	283.16	<b>280.85</b>	<b>280.85</b>	296.52	<b>280.85</b>
	1.5	283.80	<b>280.85</b>	285.66	286.57	281.11	<b>280.85</b>	281.62
		Tab. 3.4	1.5	0.5	Tab. 3.5	Tab. 3.6	1.5	Tab. 3.7

Como para  $x_3$  e para  $x_9$ , o *CTE* tem um comportamento monótono decrescente, e não é possível aumentar mais  $x$ , conclui-se que 1.5 será o melhor valor para estas duas variáveis (como indicado na última linha da tabela 3.3).

Para o caso de  $x_5$ , *CTE* tem um comportamento monótono crescente, e como não é possível diminuir  $x_5$ , 0.5 é o melhor valor para esta variável.

Como  $x_6$ ,  $x_8$  e  $x_{10}$  induzem ao *CTE* uma forma em “ $\vee$ ”, então deve-se testar em ambos os lados do menor  $x$  (ver resultados nas tabelas indicadas).

Finalmente para  $x_2$ , em que o *CTE* tem uma forma de “ $\wedge$ ”, e não é possível testar ambos os lados de  $x_2 = 0.5$ , fez-se uma procura perto de 0.5, como se pode ver na tabela 3.4. Como *CTE* não melhorou, conclui-se que 0.5 é o melhor valor para  $x_2$ .

Na tabela 3.5 pode-se ver que para  $x_6 = 1.25$ , *CTE* melhorou ligeiramente.

Na tabela 3.6, fez-se uma procura com  $x_8$  perto de 1.00. Obteve-se um melhor valor para o *CTE* usando  $x_8 = 1.25$ . Mas como a curva não mostrou um comportamento monótono, para estes valores, testou-se à volta deste ponto, para ver se era possível melhorar. Mas o resultado manteve-se.

Para  $x_{10}$  (tabela 3.7), verificou-se que 1.00 seria, numa primeira pesquisa, o melhor valor. Mas como *CTE* não é monótono neste intervalo, testou-se à volta de 1.00 mais uma vez, com um intervalo mais apertado, concluindo-se que 1.125 daria melhores resultados.

Depois desta série de procuras, o melhor valor possível para *CTE* foi conseguido com a combinação de valores representada em (3.25), tendo sido conseguida

Tabela 3.4: Variação do CTE para diferentes valores de  $x_2$ 

CTE		$x_2^*$
$x_2$	0.5	<b>280.85</b>
	0.55	281.26
	0.65	282.1
		<b>0.5</b>

Tabela 3.5: Variação do CTE para diferentes valores de  $x_6$ 

CTE		$x_6^*$
$x_6$	0.75	294.45
	1.00	280.85
	1.25	<b>280.32</b>
		<b>1.25</b>

Tabela 3.6: Variação do CTE para diferentes valores de  $x_8$ 

CTE		$x_8^*(1^a)$	$x_8^*(2^a)$
$x_8$	0.75	275.69	
	1.00	280.85	
	1.125		283.97
	1.25	<b>274.86</b>	<b>274.86</b>
	1.375		277.98
		<b>2<sup>a</sup> col.</b>	<b>1.25</b>

uma diminuição do custo total esperado de 5.99.

$$\{x_1^*, x_2^*, x_3^*, x_5^*, x_6^*, x_8^*, x_9^*, x_{10}^*\} = \{1.25, 0.5, 1.0, 0.5, 1.0, 1.25, 1.5, 1.0\}$$

$$\text{com um custo esperado de } = 274.86. \quad (3.25)$$

A diminuição do custo total esperado pode parecer insignificante, mas a análise dos seus elementos constituintes pode ser esclarecedora.



Tabela 3.7: Variação do CTE para diferentes valores de  $x_{10}$ 

CTE		$x_{10}^*(1^a)$	$x_{10}^*(2^a)$
$x_{10}$	0.75	281.98	
	0.875		281.13
	1.00	<b>280.85</b>	280.85
	1.125		<b>280.81</b>
	1.25	280.95	
		<b>2<sup>a</sup> col.</b>	<b>1.125</b>

O custo dos recursos varia entre

$$x_1^* \times \mathcal{E}[W_1] + \sum_{a \in \mathcal{F}^*} x_k^* \times \mathcal{E}[W_k] + 0.5 (\mathcal{E}[W_4] + \mathcal{E}[W_7] + \mathcal{E}[W_{11}]) = 190.63 \quad (3.26)$$

e

$$\min \left\{ 274.86; x_1^* \times \mathcal{E}[W_1] + \sum_{a \in \mathcal{F}^*} x_k^* \times \mathcal{E}[W_k] + 1.5 (\mathcal{E}[W_4] + \mathcal{E}[W_7] + \mathcal{E}[W_{11}]) \right\} = 242.71. \quad (3.27)$$

Portanto, o custo do atraso varia entre 0 e 32.15, variando o atraso entre 0 e 6.43 semanas. Mais uma vez, a redução do custo total esperado explica-se essencialmente pela redução do valor esperado do atraso.

### 3.5 Ilustração de cenários

Esta secção tem por objectivo explicar todo o comportamento do modelo, para um possível cenário do projecto exemplo, introduzido na secção 3.3. Esta ilustração irá mostrar o comportamento do modelo, precisamente nos instantes de tempo em que é necessário tomar decisões, o que acontece no início do projecto

e quando terminam as actividades. Os conteúdos de trabalho das actividades vão ser aleatoriamente escolhidos. Os valores da alocação de recursos são, como iremos ver, indicados pelo modelo. Assim, o primeiro instante de tempo em que é necessário tomar decisões é em  $t = 0$ . Nesse momento, e como indicado na secção 3.4<sup>11</sup>, devemos iniciar o projecto com  $x_1 = 1.25$ ,  $x_2 = 0.5$  e  $x_3 = 1.0$ . Se  $w_1$  for igual a 12.5,  $w_2 = 9$  e  $w_3 = 18$ , por hipótese, iremos ter as seguintes durações para as primeiras actividades do projecto:  $y_1 = 10$ ,  $y_2 = 18$  e  $y_3 = 18$ . O estado do projecto nesta altura é ilustrado na figura 3.4, onde se representam as actividades activas numa tabela colocada do lado esquerdo da figura.

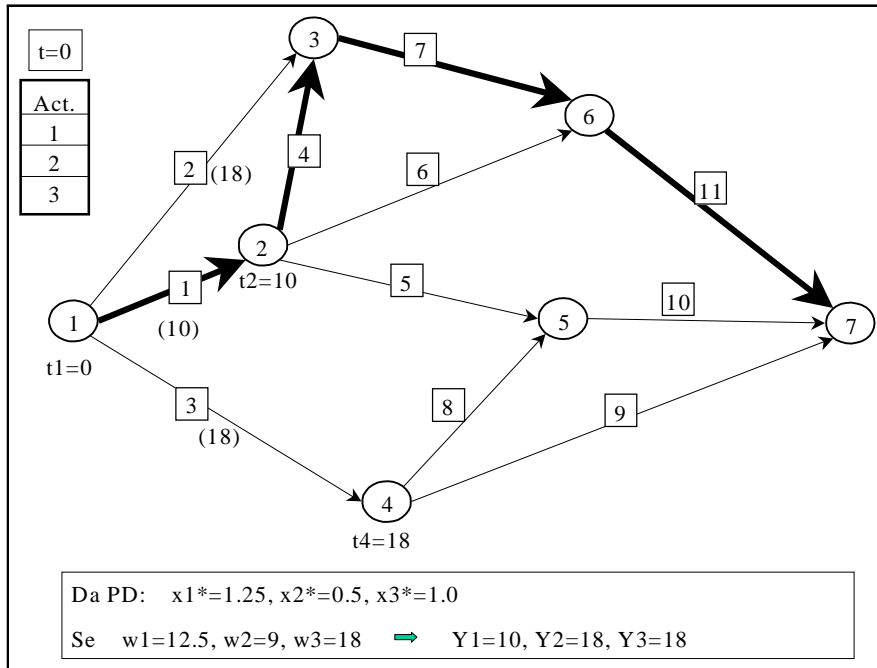
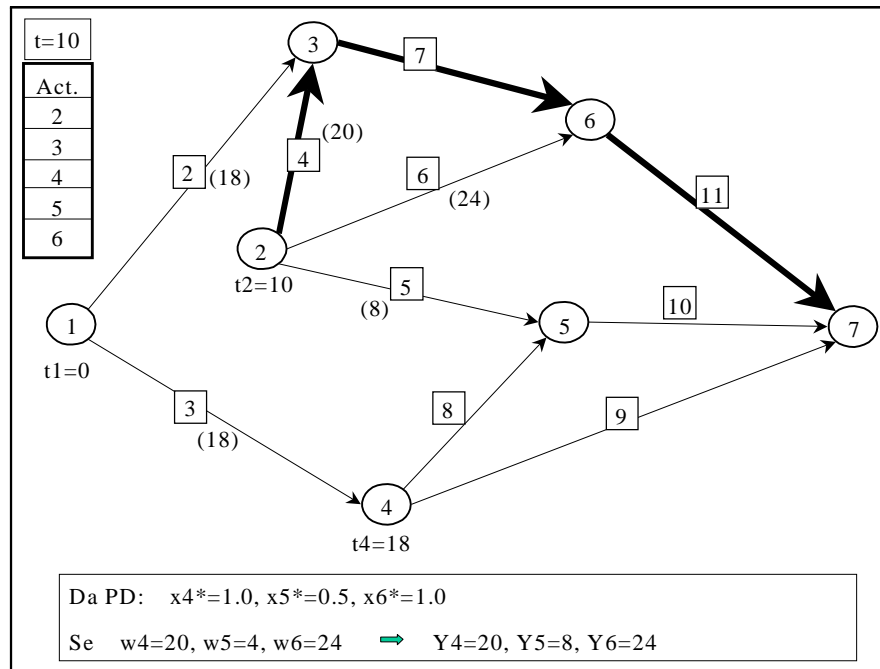


Figura 3.4: Estado do projecto no instante  $t=0$

O próximo momento em que é necessário tomar decisões é em  $t = 10$ , uma vez que termina a actividade 1. Nessa altura, as actividades 4, 5 e 6 podem iniciar. Como  $x_4$  era uma variável de decisão, o seu valor pode ser obtido pela política óptima desenvolvida para o estágio 3, utilizando a função (3.18), e para

<sup>11</sup>Optamos por utilizar os valores obtidos depois da análise de sensibilidade.

Figura 3.5: Estado do projecto no instante  $t=10$ 

$t_2 = 10$ , obteve-se  $x_4^* = 1.0$ . Este valor pode ser lido no apêndice D. Na tabela respectiva, o valor mais próximo de  $t_2$  é 16.5173. Para esse caso, o valor óptimo de  $x_4$  é 1. Para  $x_5$  e  $x_6$ , como fazem parte do conjunto das variáveis fixas, o seu valor foi determinado recorrendo à completa enumeração de  $\mathcal{F}$ , sendo o seu valor “óptimo” indicado no conjunto (3.25). De seguida, os conteúdos de trabalho das actividades 4, 5 e 6 ( $w_4, w_5$  e  $w_6$ ) irão tomar valores aleatórios, resultando em durações, como indicado na figura 3.5.

O próximo instante de tempo em que é necessário tomar decisões é  $t = 18$  (figura 3.6), onde  $x_8$  e  $x_9$  são variáveis fixas, sendo o seu valor indicado no conjunto (3.25). Continuando o processo (ver figuras 3.7 a 3.11), passamos ao instante 30, onde  $x_7^*$  pode ser obtido pela política óptima para o estágio 2 (apêndice D), onde  $\{t_2, t_3, t_4\} = \{10, 30, 18\}$ . Os valores que temos mais próximos são

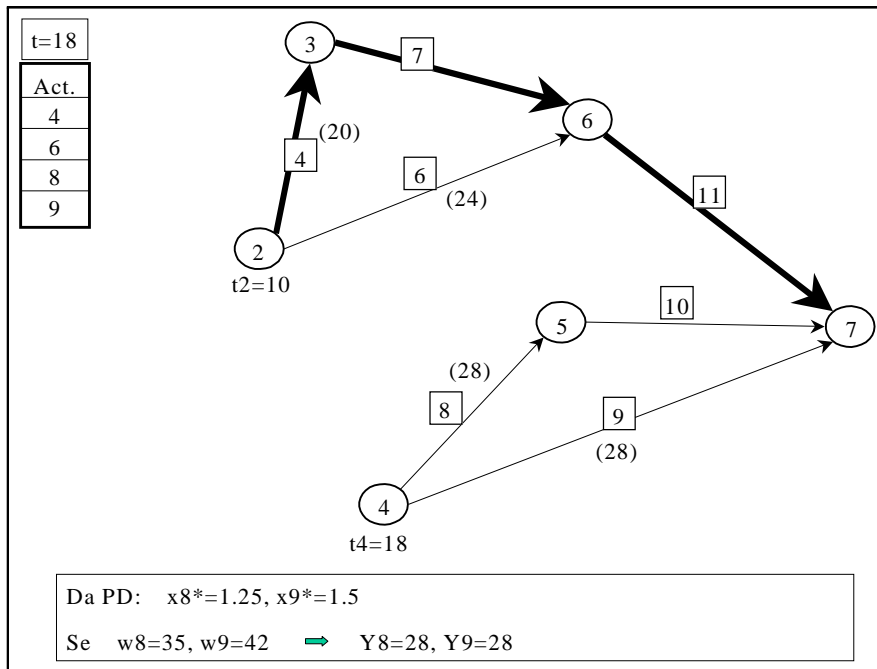


Figura 3.6: Estado do projecto no instante t=18

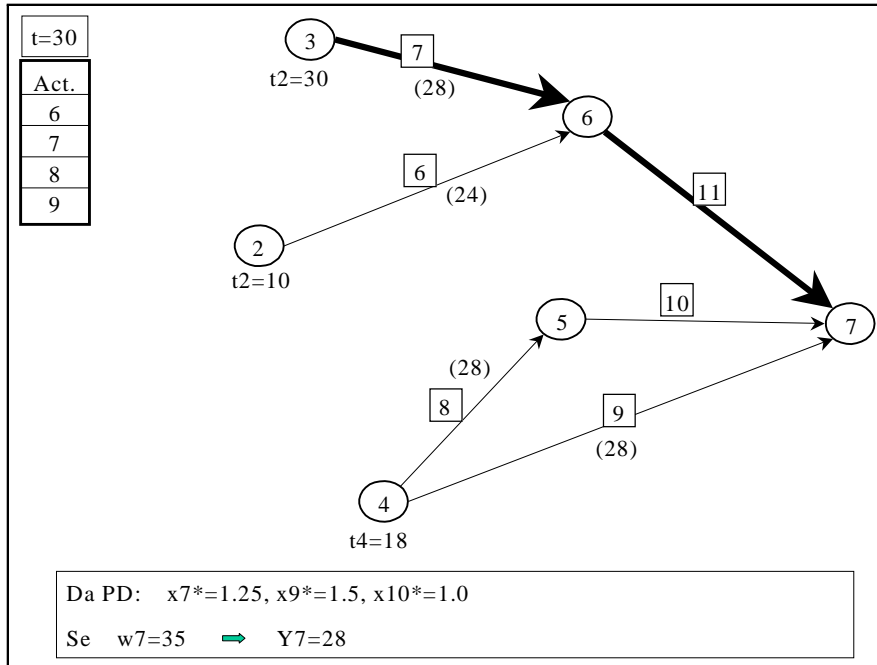


Figura 3.7: Estado do projecto no instante t=30

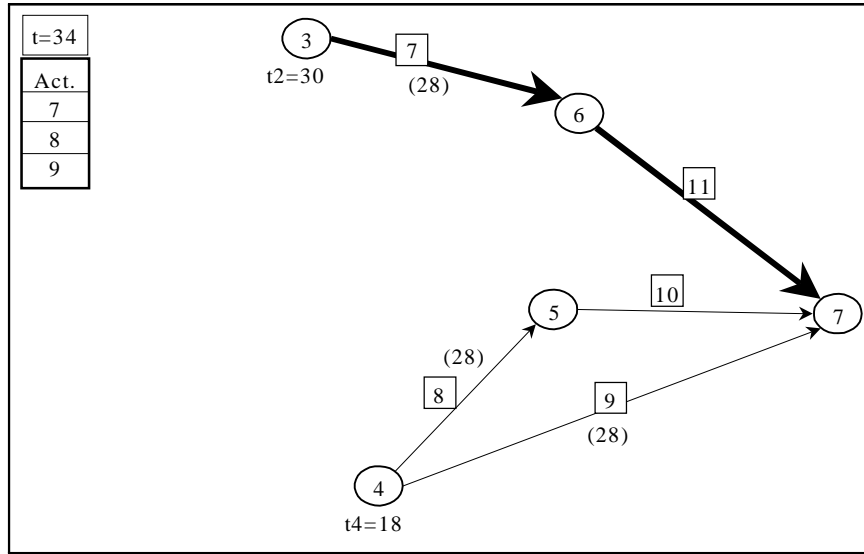


Figura 3.8: Estado do projecto no instante t=34

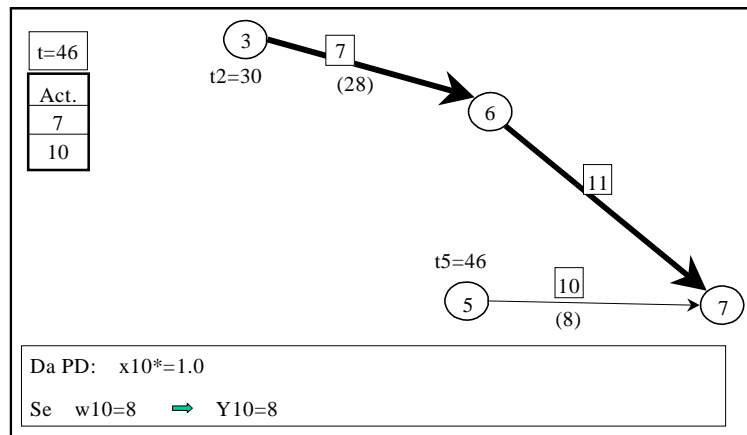
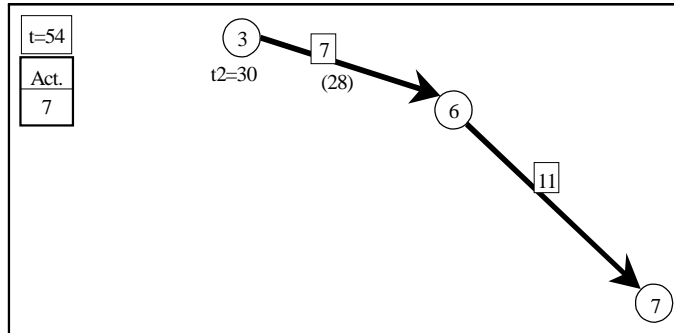
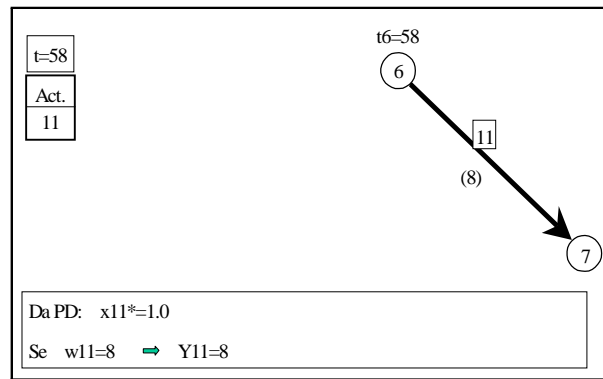


Figura 3.9: Estado do projecto no instante t=46

$\{16.52, 37.16, 33.03\}^{12}$ , sendo o valor óptimo correspondente igual a 1.25.  $x_9^*$  e  $x_{10}^*$  são indicados no conjunto (3.25). Em  $t = 34$  termina a actividade 6, mas não podemos iniciar a 11, pois a 7 ainda não terminou. Em  $t = 46$  temos que decidir o valor óptimo para  $x_{10}$ , que também é indicado no conjunto (3.25). Em

<sup>12</sup>Estas aproximações aos valores reais fazem-se, pois houve a necessidade de discretizar os vários tempos de realização do projecto, como irá ser oportunamente explicado.

Figura 3.10: Estado do projecto no instante  $t=54$ Figura 3.11: Estado do projecto no instante  $t=58$ 

$t = 54$  a actividade 10 termina, mas não é necessário tomar decisões. Por último, para  $t = 58$ , temos que decidir qual o valor óptimo para  $x_{11}$ . Consultando no apêndice D, a política óptima para o estágio 1, e procurando o valor mais próximo de  $\{t_4, t_5, t_6\} = \{18, 46, 58\}$ , obtem-se  $x_{11} = 1.0$ . Para este cenário, o projecto termina no instante  $t = 66$  (figura 3.12), ou seja, com um atraso de uma semana. Assim, o custo total do projecto (*Total Cost*) irá ser igual ao custo dos recursos ( $rc$ ) mais o custo do atraso ( $tc$ ).

$$rc = \sum_{a \in \mathcal{A}} x_a W_a \simeq 251 \quad (3.28)$$

$$tc = c_L \times \max\{0, \Upsilon_n - T\} = 5 \times \max\{0, 66 - 65\} = 5 \quad (3.29)$$

$$Total\ Cost = rc + tc \simeq 256 \quad (3.30)$$

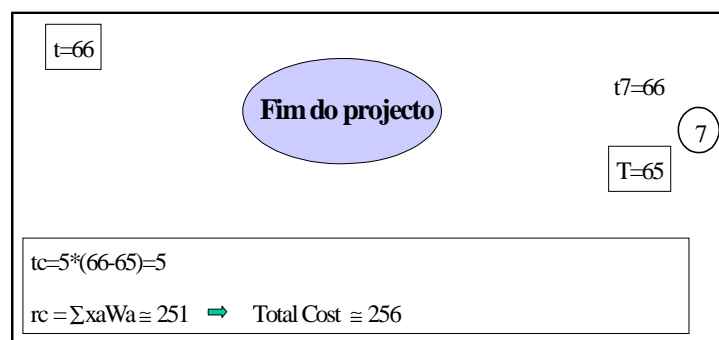


Figura 3.12: Estado do projecto no instante  $t=66$  (Fim)

Através desta ilustração é possível perceber a aplicação deste modelo de programação dinâmica, bem como constatar a sua natureza adaptativa, ao longo da vida do projecto. Uma solução estática poderia ser interessante em termos médios, mas não permitiria corrigir possíveis desvios nos tempos do projecto.





# Capítulo 4

## Implementação do modelo

O modelo de PD apresentado no capítulo 3 foi implementado em Matlab ([MathWorks96], [MathWorks97a], [MathWorks97b], [MathWorks98a], [MathWorks98b], [MathWorks98c], [MathWorks99a], [MathWorks99b]), tendo sido desenvolvidas duas versões. A primeira versão consiste na aplicação do modelo para a rede exemplo (ver figura 3.2), enquanto que a versão 2 permite a introdução de qualquer rede, sendo portanto mais genérica. Neste capítulo vão ser focados os aspectos mais importantes da implementação, para as duas versões, cujo código é apresentado nos apêndices A e B.

### 4.1 Versão 1

Esta versão (apêndice A) foi desenvolvida para aplicar o modelo de PD, apresentado no capítulo 3, ao projecto exemplo (figura 3.2).

A primeira questão que surgiu foi a identificação das variáveis necessárias para representar as actividades. Como vimos no capítulo 3, a cada actividade está associado um conteúdo de trabalho ( $W_a$ ), que é uma variável aleatória, e segue a distribuição exponencial. Assim, o parâmetro  $\lambda$  da distribuição exponencial (ver tabela 3.2) foi armazenado num ‘array’ (lambda), com tantas posições quantas as actividades do projecto (neste caso 11). De seguida, foi calculado o valor esperado

de cada  $W$ , através da fórmula (4.1).

$$\mathcal{E}(W) = \frac{1}{\lambda}. \quad (4.1)$$

Uma vez que o conteúdo de trabalho das actividades é uma variável aleatória contínua, é impossível representar informaticamente toda a sua extensão. Assim, optou-se por discretizar cada  $W$  em quatro valores, todos com igual probabilidade. A função que gera estes valores chama-se ‘*generateW.m*’. Ela recebe o parâmetro  $\lambda$  da distribuição exponencial, e retorna  $W_a$ , um ‘array’ com quatro posições. Vamos de seguida apresentar a formulação matemática que permitiu calcular estes quatro valores.

Vamos supor que  $W$  é distribuído exponencialmente, com  $\lambda = 0.1$ , então a função densidade de probabilidade é dada por (4.2) e pode ser visualizada na figura 4.1.

$$p(t) = \lambda e^{-\lambda t} \quad \text{com} \quad \lambda = 0.1 \quad (4.2)$$

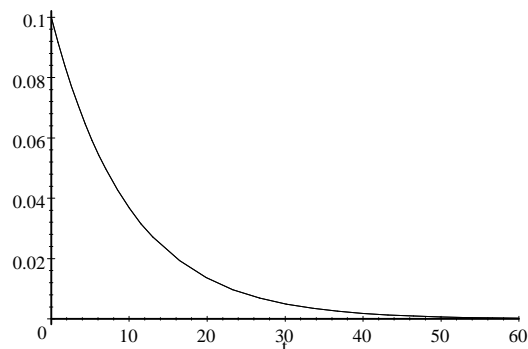


Figura 4.1:  $p(t) = 0.1 * e^{-0.1*t}$

Para  $0 \leq t_1 < t_2$ , se pretendermos achar a função distribuição de probabilidade entre  $t_1$  e  $t_2$ , teremos:

$$\int_{t_1}^{t_2} \lambda e^{-\lambda t} = [-e^{-\lambda t}]_{t_1}^{t_2} = e^{-\lambda t_1} - e^{-\lambda t_2}. \quad (4.3)$$

Como  $\lambda = 0.1$  fica,

$$P(t_1 \leq w \leq t_2) = e^{-0.1t_1} - e^{-0.1t_2}. \quad (4.4)$$

Uma vez que se pretende obter quatro valores para  $W$ , todos com igual probabilidade (0.25), optou-se por se dividir o eixo dos  $t$ 's em quatro zonas, todas com igual área (ver figura 4.2).

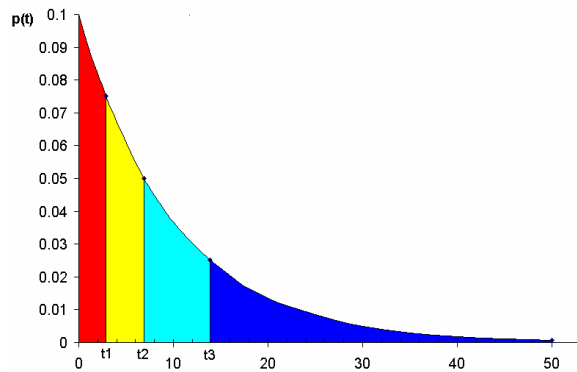


Figura 4.2: Distribuição exponencial dividida em quatro áreas iguais

Os limites dessas zonas serão  $0, t_1, t_2, t_3, +\infty$ . A determinação de  $t_1, t_2$  e  $t_3$  fez-se do seguinte modo:

$$\begin{aligned} P(0 \leq w \leq t_1) &= 0.25 \Leftrightarrow e^{-0.1 \times 0} - e^{-0.1 \times t_1} = 0.25 \Leftrightarrow \\ e^{-0.1 \times t_1} &= 0.75 \Leftrightarrow t_1 = \frac{\ln 0.75}{-0.1} \Leftrightarrow t_1 = 2.8768 \end{aligned} \quad (4.5)$$

$$P(t_1 \leq w \leq t_2) = 0.25 \Leftrightarrow t_2 = 6.9315 \quad (4.6)$$

$$P(t_2 \leq w \leq t_3) = 0.25 \Leftrightarrow t_3 = 13.8629. \quad (4.7)$$

Os valores de  $W$  pretendidos são as médias dos intervalos acima obtidos. A média global ( $\frac{1}{\lambda}$ ) pode ser calculada por:

$$\text{Média Global} = \frac{\int_0^{+\infty} tf(t)dt}{1} \quad (4.8)$$

e

$$\int_a^b tf(t)dt = \int_a^b t\lambda e^{-\lambda t} dt = \lambda \int_a^b te^{-\lambda t} dt. \quad (4.9)$$

Aplicando a regra do integral do produto de duas funções:

$$\int_a^b f'g dt = [fg]_a^b - \int_a^b fg' dt. \quad (4.10)$$

com  $f' = e^{-\lambda t} \Rightarrow f = -\frac{1}{\lambda}e^{-\lambda t}$  e  $g = t \Rightarrow g' = 1$ , temos:

$$\lambda \int_a^b te^{-\lambda t} dt = [-te^{-\lambda t}]_a^b - \lambda \int_a^b \left(-\frac{1}{\lambda}\right)e^{-\lambda t} dt = [-te^{-\lambda t}]_a^b - \frac{1}{\lambda}[e^{-\lambda t}]_a^b. \quad (4.11)$$

Assim,

$$\int_a^b tf(t)dt = [-te^{-\lambda t}]_a^b - \frac{1}{\lambda}[e^{-\lambda t}]_a^b. \quad (4.12)$$

Sendo a média global igual a:

$$\text{Média global} = [-te^{-\lambda t}]_0^{+\infty} - \frac{1}{\lambda}[e^{-\lambda t}]_0^{+\infty} = \frac{1}{\lambda}, \quad (4.13)$$

como sabido.

Da mesma forma, as médias parciais são calculadas usando a expressão (4.14), tendo em conta que a área engloba uma probabilidade de 0.25.

$$\text{Média parcial} = \frac{\int_{t_i}^{t_j} tf(t)dt}{0.25}. \quad (4.14)$$

Assim:

$$\begin{aligned} w1 &= \frac{\int_0^{t_1} tf(t)dt}{0.25} = \\ &= \frac{1}{0.25}([-te^{-\lambda t}]_0^{t_1} - \frac{1}{\lambda}[e^{-\lambda t}]_0^{t_1}) = \\ &= \frac{1}{0.25}(-t_1e^{-\lambda t_1} - \frac{1}{\lambda}(e^{-\lambda t_1} - e^0)) = \\ &= \frac{1}{0.25}(-2.8768e^{-0.1 \times 2.8768} - \frac{1}{0.1}(e^{-0.1 \times 2.8768} - 1)) = \\ &= 1.3695, \end{aligned} \quad (4.15)$$

$$w2 = \frac{\int_{t_1}^{t_2} tf(t)dt}{0.25} = 4.7675 \quad (4.16)$$

$$w3 = \frac{\int_{t_2}^{t_3} tf(t)dt}{0.25} = 10.0000 \quad (4.17)$$

$$w4 = \frac{\int_{t_3}^{+\infty} tf(t)dt}{0.25} = 23.8629. \quad (4.18)$$

Para confirmar, vemos que a média destes valores é igual à média global ( $\frac{w1+w2+w3+w4}{4} = 10$ ), tendo todos igual probabilidade e sendo representativos da distribuição exponencial. São estes valores que vamos usar para representar os conteúdos de trabalho das actividades ( $W_a$ ), que vão ser diferentes, dependendo do valor de  $\lambda$ .

Tabela 4.1: Valores de  $W$  gerados pelo programa, para as várias actividades

$a$	$\lambda_a$	$\mathcal{E}(W_a)$	$W_a(1)$	$W_a(2)$	$W_a(3)$	$W_a(4)$
<b>1</b>	0.10	10.00	1.3695	4.7675	10.0000	23.8629
<b>2</b>	0.12	8.33	1.1413	3.9729	8.3333	19.8858
<b>3</b>	0.05	20.00	2.7391	9.5350	20.0000	47.7259
<b>4</b>	0.08	12.50	1.7119	5.9594	12.5000	29.8287
<b>5</b>	0.20	5.00	0.6848	2.3838	5.0000	11.9315
<b>6</b>	0.04	25.00	3.4238	11.9188	25.0000	59.6574
<b>7</b>	0.03	33.33	4.5651	15.8917	33.3333	79.5431
<b>8</b>	0.04	25.00	3.4238	11.9188	25.0000	59.6574
<b>9</b>	0.024	41.67	5.7064	19.8647	41.6667	99.4289
<b>10</b>	0.15	6.67	0.9130	3.1783	6.6667	15.9086
<b>11</b>	0.16	6.25	0.8560	2.9797	6.2500	14.9143

No caso do projecto exemplo, os valores gerados para os  $W$ 's são os da tabela 4.1.

Um outro conjunto de variáveis fundamentais são as variáveis que representam a alocação de recursos. Assim, para cada actividade, existe um  $x_a$ , que representa a quantidade de recurso com que essa actividade deve ser executada. No nosso exemplo,

$$0.5 \leq x_a \leq 1.5. \quad (4.19)$$

Mais uma vez o intervalo dos  $x_a$ 's foi discretizado, tendo-se optado por

utilizar cinco valores para as variáveis de decisão (0.5, 0.75, 1.0, 1.25, 1.5) e três valores para as variáveis fixas (0.5, 1.0, 1.5). Esta discretização foi necessária para conseguir obter resultados em tempo útil.

Conhecidos os conteúdos de trabalho das actividades, e o recurso atribuído, é possível calcular a duração de cada actividade

$$Y_a = \frac{W_a}{x_a}. \quad (4.20)$$

Contudo esta duração vai tomar valores diferentes, dependendo do valor de  $x_a$  e do valor de  $W_a$  (cada actividade tem quatro valores possíveis para cada  $W_a$  e três ou cinco valores possíveis para  $x_a$ ). Assim, e para aplicar o modelo de PD, é necessário definir o estado, para os vários estágios. O estado não é mais do que o tempo de realização dos nós origem de cada estágio. O tempo de realização do nó  $i$  representa-se por  $t_i$ . Assim, para o primeiro nó temos  $t_1 = 0$ , uma vez que o projecto começa no instante 0. O tempo de realização do nó 2,  $t_2$ , vai depender da duração da actividade 1 (confirmar figura 3.2). Essa duração pode variar entre  $\frac{W_1(1)}{1.5}$  e  $\frac{W_1(4)}{0.5}$ . O seu valor mínimo ocorrerá quando o  $W_a$  for mínimo e o  $x_a$  for máximo. O seu valor máximo ocorrerá quando o  $W_a$  for máximo e o  $x_a$  for mínimo. Então  $0.9130 \leq t_2 \leq 47.7259$ .

Também vai ser necessário discretizar os tempos de realização dos nós. Criou-se então uma função (*calculateTvalues.m*) que, com o intervalo definido para uma variável, gera um conjunto de valores, e um ‘step’, que permitirá “saltar” de um valor para o seguinte. Por exemplo, para  $t_2$ , são gerados os seguintes valores:

$$T2 = [0.9130, 16.5173, 32.1216, 47.7259] \quad (4.21)$$

$$stT2 = 15.6043,$$

onde  $T2$  representa então os possíveis tempos de realização do nó 2, e  $stT2$  é o ‘step’ necessário para saltar entre esses valores.

Para  $t_3$  teremos que considerar o tempo de realização mínimo e máximo do nó 2 ( $t_2$ ) e somar-lhe a duração mínima e máxima da actividade 4 (confirmar figura 3.2). O restante procedimento é equivalente. Os valores gerados para os vários tempos de realização dos nós, são apresentados na tabela 4.2.

Tabela 4.2: Tempos de realização dos nós da rede exemplo

$i$	$\mathbf{T}_i$	$step_i$
<b>2</b>	[0.9130, 16.5173, 32.1216, 47.7259]	15.6043
<b>3</b>	[2.0543, 37.1640, 72.2736, 107.3832]	35.1096
<b>4</b>	[1.8261, 33.0346, 64.2432, 95.4518]	31.2086
<b>5</b>	[4.1086, 56.7731, 109.4376, 162.1020, 214.7665]	52.6645
<b>6</b>	[5.0977, 48.6597, 92.2217, 135.7836, 179.3456, 222.9076, 266.4695]	43.5620

Como se pode ver, o número de pontos considerados depende do tamanho do intervalo. Se utilizássemos sempre o mesmo número de pontos, os intervalos maiores ficariam com uma discretização muito aberta, o que implicaria a obtenção de resultados menos fiáveis.

Tendo estas variáveis definidas, podemos finalmente iniciar a aplicação do modelo de PD proposto. Já foi estabelecido que as variáveis fixas seriam  $x_2, x_3, x_5, x_6, x_8, x_9$  e  $x_{10}$  e as variáveis de decisão  $x_1, x_4, x_7$  e  $x_{11}$ . Também já está definido que existem tantos estágios quantas as variáveis de decisão (neste caso quatro), e quais as funções de contribuição de estágio (ver fórmulas (3.13), (3.16), (3.18) e (3.20)). Assim, a aplicação do modelo baseia-se no seguinte: é iniciado um ciclo, para cada variável fixa, com os valores 0.5, 1.0 e 1.5. Dentro deste conjunto de ciclos, são chamados os procedimentos para executar os quatro estágios. Para cada combinação de valores  $x_a \in \mathcal{F}$  ( $3^7 = 2187$  combinações) é armazenado o valor de todas as variáveis fixas, o valor “óptimo” do custo total esperado, para essa combinação, e o valor da primeira variável de decisão  $x_1$ . No final procura-se o menor valor esperado, sendo esse o resultado. É então apresentado o valor “óptimo” de  $x_1$ , dos vários  $x_a \in \mathcal{F}$  e do custo total esperado. O resultado



desta procura está representado no apêndice D.

Vamos então analisar com mais detalhe a aplicação das funções de contribuição de estágio. Para o estágio 1, por exemplo, a contribuição de estágio é calculada no procedimento ‘*mediumprojects1.m*’. Seguindo os passos descritos na secção 3.3, calcula-se, em primeiro lugar, o custo dos recursos das variáveis fixas, usando a fórmula (3.12). De seguida passa-se à aplicação da fórmula (3.13) que calcula o custo esperado do estágio 1 (recursos+atraso), onde é somado o custo dos recursos fixos de todo o projecto. É indiferente somar este valor neste estágio, ou noutro qualquer, uma vez que é uma constante.

Como os tempos de realização dos nós ( $t_4$ ,  $t_5$  e  $t_6$  para o estágio 1) podem tomar vários valores, é necessário prever as várias possibilidades, sendo utilizados novamente ciclos para tal.

Para calcular o custo do atraso é necessário, em primeiro lugar, calcular o tempo de realização do nó 7, o último nó da rede (ver fórmula (3.15)). Contudo, ao fazer este tipo de cálculos, é necessário ter em atenção que estamos a trabalhar com variáveis aleatórias. Assim, por exemplo,  $t_4 + \frac{W_9}{x_9}$  irá ter vários valores, com diferentes probabilidades.

Seja

$$X_1 = t_4 + \frac{W_9}{x_9} = [5.6303, 15.0692, 29.6038, 68.112] \quad (4.22)$$

com

$$pX_1 = [0.25, 0.25, 0.25, 0.25]. \quad (4.23)$$

Seja

$$X_2 = t_5 + \frac{W_{10}}{x_{10}} = [5.0216, 7.287, 10.7783, 20.0172] \quad (4.24)$$

com

$$pX_2 = [0.25, 0.25, 0.25, 0.25]. \quad (4.25)$$

Põe-se então o problema de calcular o máximo entre duas variáveis aleatórias. Para tal foi desenvolvida uma função ‘*calculateMax.m*’, que calcula o máximo entre duas variáveis aleatórias e ‘*calculatePmax.m*’ que calcula a probabilidade do máximo. Vamos ver como funcionam, num exemplo simples.

Seja

$$X_1 = [1, 3, 6] \quad \text{com} \quad pX_1 = [0.3, 0.3, 0.5] \quad (4.26)$$

e

$$X_2 = [2, 4, 5] \quad \text{com} \quad pX_2 = [0.1, 0.5, 0.4]. \quad (4.27)$$

O máximo será  $Y = [2, 3, 4, 5, 6]$  e não  $Y = [2, 4, 6]$ , como poderíamos erradamente pensar (ver tabelas 4.3 e 4.4).

Tabela 4.3: Máximo entre duas variáveis aleatórias

$X_1$	$X_2$	$Y = \max(X_1, X_2)$
1	2	2
1	4	4
1	5	5
3	2	3
3	4	4
3	5	5
6	2	6
6	4	6
6	5	6

A forma mais simples de implementar o cálculo da probabilidade do máximo é usar probabilidades cumulativas. Assim,

$$P(Y \leq y) = P(X_1 \leq y) \times P(X_2 \leq y). \quad (4.28)$$

Tabela 4.4: Probabilidade do máximo entre duas variáveis aleatórias

$Y$	$p$
2	$0.3 \times 0.1 = 0.03$
3	$0.2 \times 0.1 = 0.02$
4	$0.5 \times 0.5 + 0.2 \times 0.5 = 0.25$
5	$0.3 \times 0.4 + 0.2 \times 0.4 = 0.2$
6	$0.5 \times 0.1 + 0.5 \times 0.5 + 0.5 \times 0.4 = 0.5$

Tabela 4.5: Probabilidade do máximo entre duas variáveis aleatórias (2)

$Y$	$P(X_1 \leq y)$	$P(X_2 \leq y)$	$p(Y \leq y)$	$p(Y = y)$
2	0.3	0.1	0.03	0.03
3	0.5	0.1	0.05	0.02
4	0.5	0.6	0.3	0.25
5	0.5	1.0	0.5	0.2
6	1.0	1.0	1.0	0.5

Na tabela 4.5 pode-se confirmar que o cálculo assim efectuado resulta nos mesmos valores de  $p$  calculados na tabela 4.4. Foi este o procedimento utilizado no programa.

O resultado para

$$X_1 = [5.6303, 15.0692, 29.6038, 68.112], \quad PX_1 = [0.25, 0.5, 0.75, 1] \quad (4.29)$$

e

$$X_2 = [5.0216, 7.287, 10.7783, 20.0172], \quad PX_2 = [0.25, 0.5, 0.75, 1] \quad (4.30)$$

foi:

$$Y_7 = \max(X_1, X_2) = [5.6303, 7.287, 10.7753, 15.0692, 20.6038, 29.6038, 68.112] \quad (4.31)$$

$$PY_7 = Prob.\max(X_1, X_2) = [0.0625, 0.125, 0.1875, 0.375, 0.5, 0.75, 1].$$

Como  $x_{11}$  é a variável de decisão deste estágio,  $x_{11}$  irá tomar todos os valores possíveis, o mesmo acontecendo a  $W_{11}$ . Para cada um desses valores, calcula-se finalmente  $T_7 = \max(Y_7, t_6 + \frac{W_{11}}{x_{11}})$ . Neste caso o máximo é calculado com recurso a uma função interna do Matlab, uma vez que já não temos duas variáveis aleatórias, mas só uma. Ficaria  $T_7 = \max([5.6303, 7.287, 10.7753, 15.0692, 20.6038, 29.6038, 68.112], 6.8096) = [6.8096, 7.287, 10.7753, 15.0692, 20.6038, 29.6038, 68.112]$  com probabilidade igual à de  $Y_7$ . Depois de termos  $T_7$  (tempo de realização do último nó), podemos calcular o atraso  $U$  (ver fórmula (3.14)), que será uma variável aleatória, uma vez que  $T_7$  é aleatória. Calcula-se depois o valor esperado do atraso e multiplica-se por  $c_L$  (ver fórmula (3.13)). Isto é feito para os quatro valores de  $W_{11}$ . De seguida, para cada valor de  $x_{11}$  (existem cinco possíveis), é calculado o custo do recurso 11 ( $x_{11} \times W_{11}$ ) e o custo total, que é igual ao custo dos recursos fixos, mais o custo do atraso, mais o custo do recurso 11. Este custo total vai ser uma variável aleatória, com quatro valores possíveis (um valor por cada  $W$ ), sendo depois calculado o valor esperado do custo total. Esse valor é armazenado, para cada  $x_{11}$ .

Depois de  $x_{11}$  tomar os cinco valores possíveis, vamos ter um ‘array’ com os valores de custo total esperado deste estágio, para os vários valores de  $x_{11}$ . É depois calculado o mínimo valor do custo, sendo armazenado em conjunto com a variável de decisão  $x_{11}$ , que lhe deu origem. Todo este procedimento é repetido para as diferentes combinações dos tempos de realização  $t_4$ ,  $t_5$  e  $t_6$ . Obtém-se no final a política óptima a seguir, no estágio 1, para todas as combinações possíveis dos tempos de realização dos seus nós (ver apêndice D).

Terminado o estágio 1, passa-se para o 2. No estágio 2, a contribuição de estágio é calculada somando o custo do recurso associado à actividade 7, com o custo total esperado, calculado no estágio 1 (ver fórmula (3.16)). O estado é representado por  $(t_2, t_3, t_4)$ , mas para aceder ao custo esperado do estágio 1, necessitamos de ter o valor de  $\Upsilon_5$  e  $\Upsilon_6$  (ver fórmula (3.17)). Também aqui estão a ser utilizadas variáveis aleatórias, sendo o cálculo do máximo feito, como já explicado atrás. Depois de ter os valores de  $\Upsilon_5$  e  $\Upsilon_6$ , podemos aceder ao valor de  $f_1(t_4, \Upsilon_5, \Upsilon_6)$  na política óptima desenvolvida no estágio 1.

Os estágios 3 e 4 funcionam da mesma forma. No final do estágio 4 obtém-se o valor “óptimo” de  $x_1$  e o valor correspondente do custo total esperado “mínimo”.

Estes cálculos são repetidos para todas as combinações das variáveis fixas ( $3^7 = 2187$ ), resultando no final numa tabela com 2187 linhas, com as várias combinações das variáveis fixas, e para cada combinação, o valor “óptimo” de  $x_1$  e do custo total esperado. A optimização sobre as variáveis fixas faz-se neste momento. Para tal, procura-se o valor mínimo do custo, e a combinação dos  $x$ 's que lhe deu origem será a combinação “óptima” (ver apêndice D).

Todos os outros procedimentos desenvolvidos são auxiliares e de mais simples compreensão, pelo que não vão ser detalhados aqui. Foram contudo comentados (ver apêndice A), para facilitar a sua leitura.

## 4.2 Versão 2

Esta versão foi desenvolvida com o objectivo de tornar mais genérica a aplicação do modelo de programação dinâmica proposto. Assim, enquanto que a versão 1 visa essencialmente a aplicação do modelo ao projecto exemplo, a versão 2 generaliza essa aplicação a qualquer projecto. Nesta secção vão ser apresentados os aspectos mais importantes desta generalização, uma vez que a implementação da aplicação do modelo já foi explicada na secção anterior. O código desta versão encontra-se no apêndice B.

Na versão 1, são introduzidas directamente no programa, variáveis representativas do projecto exemplo, como os conteúdos de trabalho das actividades, os parâmetros da distribuição exponencial respectivos, a data de conclusão do projecto, o custo do atraso unitário, as quantidades de recursos, etc. Na versão 2, como se pretende possibilitar a introdução de qualquer projecto, é necessário permitir a introdução destes dados, e ter estruturas dinâmicas para os armazenar. Mas para além deste tipo de informações, é necessário também representar as relações de precedência entre as actividades do projecto. Para tal foi definida uma estrutura matricial, em que cada coluna representa uma actividade. Na primeira linha será colocado o nó origem dessa actividade, na segunda linha o nó destino, na terceira os parâmetros da distribuição exponencial e nas duas últimas linhas o nível mínimo e máximo de recursos.

Tabela 4.6: Net - Estrutura que armazena os dados do projecto

Actividade	1	2	3	4	5	6	7	8	9	10	11
<b>Origem</b>	1	1	1	2	2	2	3	4	4	5	6
<b>Destino</b>	2	3	4	3	5	6	6	5	7	7	7
$\lambda$	0.10	0.12	0.05	0.08	0.20	0.04	0.03	0.04	0.024	0.15	0.16
$\mathbf{x}_{\min}$	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5
$\mathbf{x}_{\max}$	1.5	1.5	1.5	1.5	1.5	1.5	1.5	1.5	1.5	1.5	1.5

Na tabela 4.6 podemos ver a representação do projecto exemplo (figura 3.2), na estrutura definida (*Net*).

O programa principal desta versão é o ‘*dp.m*’. Esse programa começa por chamar o procedimento ‘*inputNetwork.m*’, que contém as instruções que permitem ao utilizador introduzir os dados relativos a qualquer projecto. Assim, é pedido o número de actividades do projecto ( $n$ ) e para cada actividade são pedidos os dados constantes na tabela 4.6. É ainda pedido o valor de  $c_L$  e  $T$ . Por uma questão de comodidade, é possível gravar estes dados, para uma futura utilização.

O problema que surgiu de seguida foi como determinar o número de estágios do

projecto. Esta questão já tinha sido abordada no capítulo 3, e a estratégia seguida foi determinar o caminho mais longo na rede (por número de actividades). As actividades ao longo desse caminho seriam as variáveis de decisão e as restantes as variáveis fixas. O número de estágios iria ser igual ao número de variáveis de decisão. No desenvolvimento da versão 1 todos estes aspectos já estavam definidos. Isto é, ao escrever o programa já sabíamos que  $x_1$ ,  $x_4$ ,  $x_7$  e  $x_{11}$  eram as variáveis de decisão; que as restantes eram as variáveis fixas; que existiam quatro estágios e como eram definidos. Na versão 2 todo este trabalho mental tem que ser feito pelo programa, uma vez que não sabemos à partida que rede vamos ter. Para tal, foi desenvolvido um conjunto de procedimentos. O primeiro procedimento, ‘*decisionVars.m*’ permite determinar quais as variáveis de decisão. Para tal, chama a função ‘*sumNodesLen.m*’ que calcula o número máximo de actividades que é necessário percorrer, para alcançar um determinado nó. Para o projecto exemplo, teríamos,

$$NL = \begin{bmatrix} \mathbf{1} & \mathbf{2} & \mathbf{3} & \mathbf{4} & \mathbf{5} & \mathbf{6} & \mathbf{7} \\ 0 & 1 & 2 & 1 & 2 & 3 & 4 \end{bmatrix}, \quad (4.32)$$

onde os números de 1 a 7 representam os nós. O caminho mais longo vai ser o caminho que passa por um maior número de actividades. De seguida é chamada outra função, que determina quais os nós pertencentes ao caminho mais longo (‘*nodesLP.m*’). Esta função começa por colocar o último nó na estrutura *NLP*, que vai conter os nós do caminho mais longo, uma vez que este nó pertence sempre ao caminho mais longo. De seguida, e recuando até chegar ao nó inicial, vão sendo determinadas as origens do último nó colocado no *NLP*. Podem então ocorrer duas situações: o nó só tem uma origem, então essa origem também faz parte do *NLP*, sendo então colocado à cabeça, ou o nó tem mais do que uma origem. Neste caso, recorrendo à variável *NL*, pretende-se determinar o nó que tem um  $nl = nl(\text{nó em questão}) - 1$ , ou seja, o nó precedente que pertence ao caminho mais longo. Este processo resulta, para o projecto exemplo, no seguinte conjunto

de nós,  $NLP = [1, 2, 3, 6, 7]$ , como sendo o caminho mais longo. Por último, para saber quais as variáveis de decisão, temos que determinar as actividades no caminho mais longo (*'actividadesLP.m'*). Estas actividades vão ser as actividades que ligam os nós do caminho mais longo. Para tal, vai-se procurar na estrutura *Net*, qual a coluna que tem origem 1 e destino 2, por exemplo, sendo o resultado para este caso 1. O resultado final será  $ALP = [1, 4, 7, 11]$ . Estas serão as variáveis de decisão, e existirão quatro estágios, neste caso.

De seguida vão ser gerados os  $W$ 's (conteúdos de trabalho das actividades) a partir dos parâmetros introduzidos inicialmente. O processo de geração dos quatro valores de  $W$  é igual ao utilizado na versão 1 (ver secção 4.1), sendo estes valores armazenados num 'array' com quatro colunas e tantas linhas quantas as actividades do projecto, logo com dimensão  $n \times 4$ . Assim,  $W(2,4)$  representa o conteúdo de trabalho máximo da actividade 2, e  $W(5,1)$  representa o conteúdo de trabalho mínimo da actividade 5.

Segue-se o cálculo das durações mínima e máxima de cada actividade (*'durations.m'*), o que permite depois calcular os valores para os tempos de realização dos vários nós (*'generateTlimits.m'* e *'generateTvalues.m'*).  $Tlim$  irá conter os limites do intervalo dos tempos de realização dos nós, e a estrutura  $Tval$  irá conter, num campo, a lista discretizada de tempos de realização dos nós, e noutra o 'step' entre eles. Em  $Tlim(1)$  são armazenados os limites inferiores e em  $Tlim(2)$ , os superiores. Os limites para os  $T$ 's são gerados da seguinte forma: para o primeiro nó da rede,  $Tlim = 0$ ; para os restantes, determinam-se as suas origens, e aplica-se a seguinte fórmula:

$$Tlim_j(1) = \max_{i \rightarrow j} \{Tlim_i(1) + Y \min\}, \quad (4.33)$$

$$Tlim_j(2) = \max_{i \rightarrow j} \{Tlim_i(2) + Y \max\} \quad j = 2, \dots, m.$$

O procedimento que a partir dos valores limite, gera os vários  $T$ 's, é idêntico ao utilizado na versão 1.



Em  $AF$  são armazenadas as actividades a fixar, bem como os valores que elas devem tomar (na discretização). No caso do projecto exemplo, teríamos:

$$AF = \begin{bmatrix} \mathbf{2} & \mathbf{3} & \mathbf{5} & \mathbf{6} & \mathbf{8} & \mathbf{9} & \mathbf{10} \\ 0.5 & 0.5 & 0.5 & 0.5 & 0.5 & 0.5 & 0.5 \\ 1.0 & 1.0 & 1.0 & 1.0 & 1.0 & 1.0 & 1.0 \\ 1.5 & 1.5 & 1.5 & 1.5 & 1.5 & 1.5 & 1.5 \end{bmatrix}. \quad (4.34)$$

Terminados os preparativos, passou-se à aplicação do modelo de PD propriamente dito. Contudo, surgiu aqui uma importante questão: a escrita do código para executar o modelo, desconhecendo a rede que irá ser modelada!

Referindo a versão 1 (ver ‘*mp.m*’ no apêndice A), a aplicação do modelo inicia com um conjunto de ciclos ‘*for*’, onde são chamados os procedimentos que aplicam as funções de contribuição, para os vários estágios. Os procedimentos ‘*mediumprojects1.m*’ a ‘*mediumprojects4.m*’ foram feitos, sabendo à partida que estávamos a trabalhar com um projecto específico, onde existiam quatro estágios, com todas as variáveis bem definidas.

Na versão 2, isso não acontece. Só depois da introdução da rede pelo utilizador, poderemos determinar quais as variáveis de decisão e quais as fixas, e portanto quantos ciclos ‘*for*’ são necessários e quantos procedimentos de cálculo da contribuição de estágio tem que existir. O conteúdo desses procedimentos também varia, dependendo da rede em questão. Surgiu então uma ideia que permitiu resolver este problema: escrever o código em tempo real, ou seja, depois da introdução dos parâmetros do problema, são gerados os programas para o resolver. Esta foi uma tarefa complexa, pois foi necessário prever todas as situações possíveis.

Sucintamente, descreve-se a metodologia: depois de determinarmos as ALP (‘Activities on the Longest Path’) já sabemos quantos estágios vai ter o problema. Começa-se por gerar o programa principal, através do procedimento ‘*generateMainCode.m*’. Este procedimento abre um ficheiro ‘*main.m*’, que irá

conter as instruções do programa principal, que será depois chamado para otimizar a rede introduzida. Depois de aberto o ficheiro, vão sendo nele escritas as instruções, sob a forma de ‘strings’, equivalentes às instruções de ‘*mp.m*’, a partir dos ciclos ‘*for*’. O ficheiro ‘*main.m*’ gerado, pode ser visto no apêndice C, se o projecto introduzido for o projecto exemplo. É neste ficheiro que vão ser colocadas as instruções para chamar os procedimentos que irão calcular as contribuições de estágio, ou seja, ‘*dps1.m*’, ‘*dps2.m*’, ..., também em apêndice.

O passo seguinte será gerar ‘*dps1.m*’, que é o procedimento que irá calcular a contribuição do estágio 1. Este procedimento é distinto dos seguintes (de ‘*dps2.m*’, ‘*dps3.m*’, ...) pois é no estágio 1 que é calculado o custo do atraso e o custo dos recursos fixos. O procedimento que gera ‘*dps1.m*’ é ‘*generateDps1Code.m*’.

Mais uma vez se colocava uma questão: como representar os estágios, ou seja, quais os nós que permitem definir os estágios? A fórmula utilizada para resolver este problema foi a seguinte: o estágio 1 é definido por todos os nós origem do último nó. Para o projecto exemplo,  $s_1 : t_4, t_5, t_6$ . Os estágios seguintes são definidos pelos nós origem dos nós do estágio precedente, eliminando valores repetidos. Assim,  $s_2 : t_2, t_3, t_4$ ,  $s_3 : t_2$  e  $s_4 : 0$ .

No estágio 1, a actividade a otimizar é a última actividade de ALP. A primeira instrução escrita é:  $Best_1 = [ ]$ , que inicializa o ‘array’ que irá conter a política óptima para o estágio 1. São depois escritas as restantes instruções que permitem calcular a contribuição do estágio 1, equivalentes às do programa ‘*mediumprojects1.m*’ da versão 1. O código gerado (‘*dps1.m*’) para o projecto exemplo, pode ser visto no apêndice C, e o programa gerador (‘*generateDps1Code.m*’), no apêndice B.

Para gerar os programas para os restantes estágios é sempre chamado o ‘*generateDpsNCode.m*’. Isto porque as instruções contidas nesses programas são suficientemente parecidas. É sempre calculado o custo do recurso do estágio actual, sendo esse valor adicionado ao custo total esperado resultante do estágio

anterior. Assim, e para o projecto exemplo, são gerados os ficheiros ‘*dps2.m*’, ‘*dps3.m*’ e ‘*dps4.m*’ (ver no apêndice C os programas gerados e no B o gerador).

Depois de gerado o código necessário à aplicação do modelo, é então utilizada a instrução ‘*rehash*’ do *Matlab*, que coloca em memória os novos programas, para poderem ser chamados e dar assim a solução para qualquer rede introduzida. Vamos ver de seguida alguns exemplos.

### 4.2.1 Exemplos

A versão 2 foi testada com uma série de exemplos, que vão ser aqui apresentados. Para cada exemplo, será definida a rede, bem como os restantes parâmetros representativos do problema. No final será apresentada a solução dada pelo programa. A solução indica os valores para as quantidades de recursos a atribuir às actividades que partem do primeiro nó, bem como para as restantes variáveis fixas. Os valores das variáveis de decisão seguintes dependem do estado do projecto, quando for necessário tomar a decisão.

O projecto que serviu de base à versão 1 foi também testado, tendo sido obtidos os mesmos resultados, como seria de esperar, mas com um acréscimo do tempo de computação de 141 minutos para 186 minutos, devido ao aumento da complexidade do código.

#### Rede 1

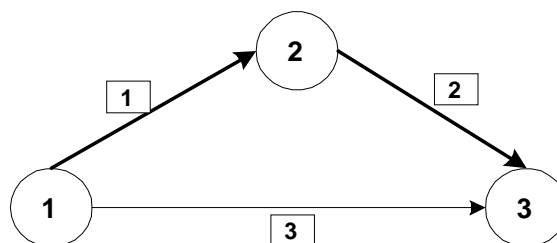


Figura 4.3: Rede 1

A primeira rede testada está representada na figura 4.3. É uma rede simples, só com três actividades.  $T = 14$  e  $c_L = 2$ . Os restantes parâmetros estão representados na tabela 4.7.

Tabela 4.7: Parâmetros da rede 1

Actividade	1	2	3
Origem	1	2	1
Destino	2	3	3
$\lambda$	0.2	0.1	0.07
$x_{\min}$	0.5	0.5	0.5
$x_{\max}$	1.5	1.5	1.5

A solução para esta rede foi encontrada em 4.8 segundos, e é apresentada em (4.35).

$$\{x_1^*, x_3^*\} = \{1.0, 1.0\} \quad (4.35)$$

com um custo esperado de  $= 45.53$ .

## Rede 2

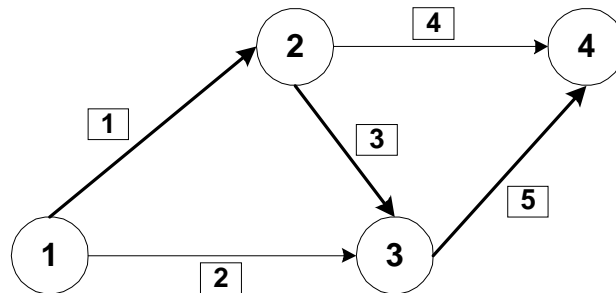


Figura 4.4: Rede 2

A segunda rede exemplo já é um pouco mais complexa, tendo 5 actividades

(ver figura 4.4).  $T = 120$  e  $c_L = 8$ . Na tabela 4.8 estão representados os restantes parâmetros.

Tabela 4.8: Parâmetros da rede 2

Actividade	1	2	3	4	5
Origem	1	1	2	2	3
Destino	2	3	3	4	4
$\lambda$	0.02	0.03	0.04	0.024	0.025
$x_{\min}$	0.5	0.5	0.5	0.5	0.5
$x_{\max}$	1.5	1.5	1.5	1.5	1.5

A solução para esta rede foi encontrada em 9.6 segundos, e pode ser vista em (4.36).

$$\{x_1^*, x_2^*, x_4^*\} = \{1.0, 1.0, 1.5\} \quad (4.36)$$

com um custo esperado de  $= 304.62$ .

### Rede 3

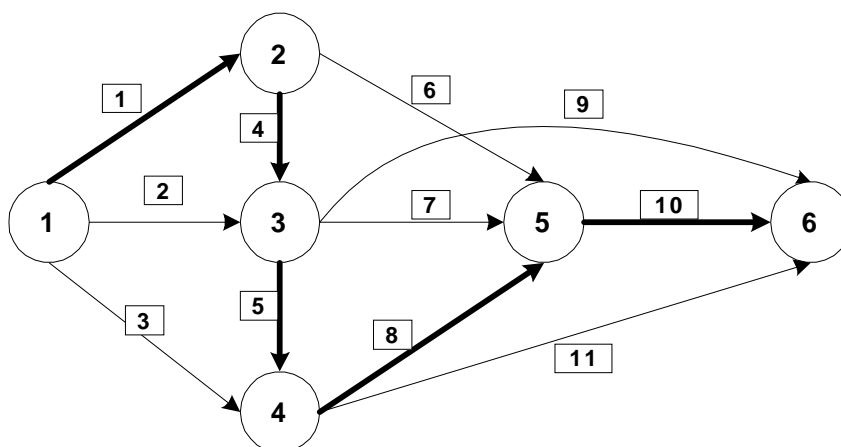


Figura 4.5: Rede 3

A rede 3, com uma dimensão maior (11 actividades), apresenta-se na figura 4.5. Para esta rede,  $T = 28$  e  $c_L = 8$ . Na tabela 4.9 estão representados os restantes parâmetros.

Tabela 4.9: Parâmetros da rede 3

Actividade	1	2	3	4	5	6	7	8	9	10	11
<b>Origem</b>	1	1	1	2	3	2	3	4	3	5	4
<b>Destino</b>	2	3	4	3	4	5	5	5	6	6	6
$\lambda$	0.1	0.09	0.4	0.2	0.3	0.08	0.4	0.2	0.1	0.3	0.3
$x_{\min}$	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5
$x_{\max}$	1.5	1.5	1.5	1.5	1.5	1.5	1.5	1.5	1.5	1.5	1.5

A solução para esta rede, devido à sua maior dimensão, demorou 30 minutos a ser encontrada. Apresenta-se em (4.37).

$$\{x_1^*, x_2^*, x_3^*, x_6^*, x_7^*, x_9^*, x_{11}^*\} = \{1.25, 1.0, 0.5, 1.0, 0.5, 1.0, 1.0\} \quad (4.37)$$

com um custo esperado de  $= 106.76$ .

#### Rede 4

A rede 4 tem uma dimensão ainda maior que a anterior, em número de actividades (ver figura 4.6). O valor de  $T$  é 47 e de  $c_L$  é 4. Os restantes parâmetros do problema podem ser vistos na tabela 4.10.

$$\{x_1^*, x_2^*, x_3^*, x_4^*, x_6^*, x_7^*, x_8^*, x_{10}^*, x_{12}^*\} = \{1.25, 1.0, 1.0, 0.5, 1.0, 1.0, 0.5, 1.0, 1.5\}$$

com um custo esperado de  $= 182.91$ . (4.38)

A solução para esta rede (4.38), apesar de ser apenas marginalmente maior que a anterior, demorou bastante mais tempo a ser encontrada (23 horas). Isto prende-se com o facto do número de variáveis de decisão ser menor (4 em vez de

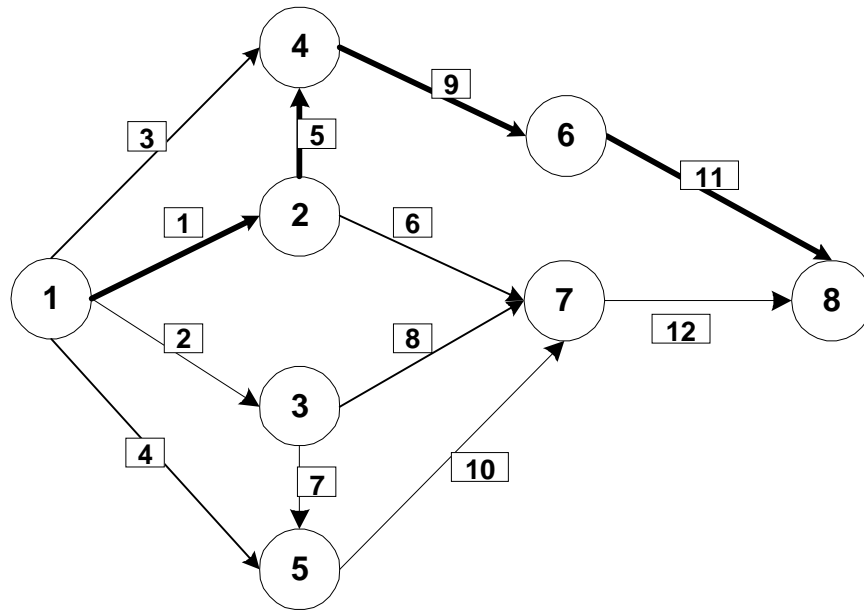


Figura 4.6: Rede 4

Tabela 4.10: Parâmetros da rede 4

Actividade	1	2	3	4	5	6	7	8	9	10	11	12
<b>Origem</b>	1	1	1	1	2	2	3	3	4	5	6	7
<b>Destino</b>	2	3	4	5	4	7	5	7	6	7	8	8
$\lambda$	0.1	0.09	0.08	0.1	0.09	0.08	0.1	0.09	0.08	0.1	0.09	0.1
$x_{\min}$	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5
$x_{\max}$	1.5	1.5	1.5	1.5	1.5	1.5	1.5	1.5	1.5	1.5	1.5	1.5

5 da rede 3), o que implica um maior número de variáveis fixas (8 em vez de 6 da rede 3). Assim, o número de combinações geradas para as variáveis fixas, para a rede 3 é de  $3^6 = 729$  e para a rede 4,  $3^8 = 6561$ , daí o aumento de tempo de cálculo verificado.

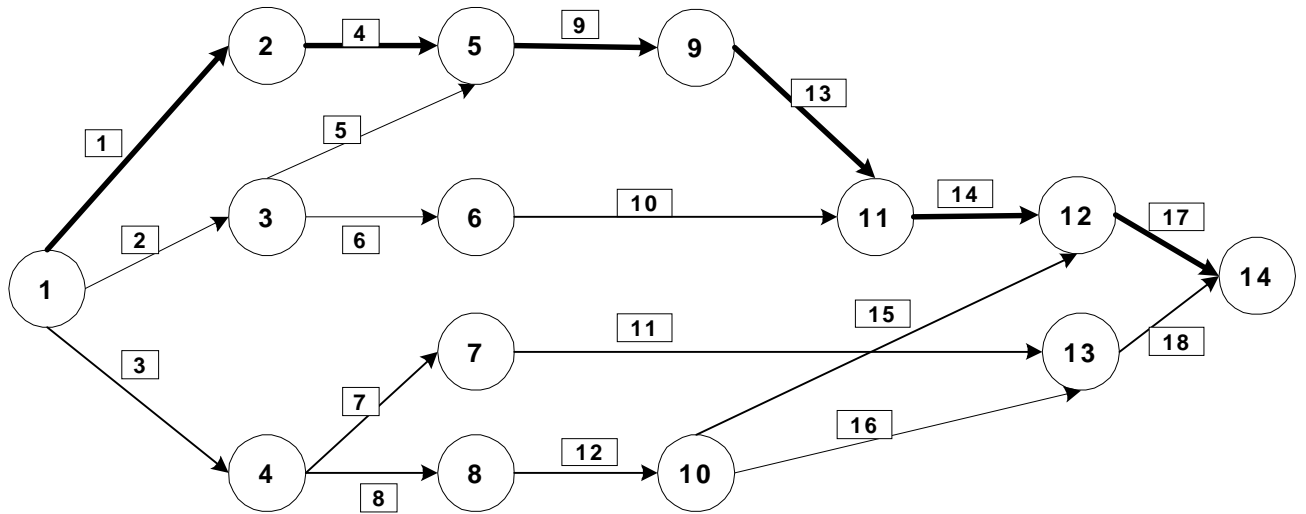


Figura 4.7: Rede 5

### Rede 5

Para terminar este conjunto de testes, introduziu-se uma rede (figura 4.7) significativamente maior que a anterior (18 actividades), com  $T = 110$  e  $c_L = 10$ . Os restantes parâmetros estão representados na tabela 4.11.

Para conseguir obter uma solução para esta rede, foi necessário reduzir a dois o número de pontos considerados para as variáveis fixas (0.75, 1.25). Mesmo depois desta redução, só conseguimos obter resultados ao fim de 5 dias. A solução obtida pode ver-se em (4.39).

$$\begin{aligned} & \{x_1^*, x_2^*, x_3^*, x_5^*, x_6^*, x_7^*, x_8^*, x_{10}^*, x_{11}^*, x_{12}^*, x_{15}^*, x_{16}^*, x_{18}^*\} \\ & = \{0.75, 1.25, 1.25, 0.75, 1.25, 0.75, 1.25, 1.25, 1.25, 0.75, 0.75, 1.25, 0.75\} \quad (4.39) \end{aligned}$$

com um custo esperado de  $= 339.07$ .

Este conjunto de testes pretende demonstrar que a aplicação do modelo de PD proposto pode ser feita a qualquer rede de actividades, utilizando a versão 2. Contudo, a aplicação de PD directa, para redes com mais de 10 actividades,



Tabela 4.11: Parâmetros da rede 5

Actividade	1	2	3	4	5	6	7	8	9
Origem	1	1	1	2	3	3	4	4	5
Destino	2	3	4	5	5	6	7	8	9
$\lambda$	0.06	0.04	0.1	0.07	0.08	0.04	0.08	0.2	0.07
$x_{\min}$	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5
$x_{\max}$	1.5	1.5	1.5	1.5	1.5	1.5	1.5	1.5	1.5
Actividade	10	11	12	13	14	15	16	17	18
Origem	6	7	8	9	11	10	10	12	13
Destino	11	13	10	11	12	12	13	14	14
$\lambda$	0.05	0.08	0.07	0.09	0.09	0.05	0.09	0.04	0.06
$x_{\min}$	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5
$x_{\max}$	1.5	1.5	1.5	1.5	1.5	1.5	1.5	1.5	1.5

torna-se proibitiva em termos de tempos de computação. No próximo capítulo vão ser apresentadas aproximações a este modelo, que permitem reduzir significativamente estes tempos.



# Capítulo 5

## Aproximações ao modelo proposto

Como vimos nos capítulos anteriores, o modelo de programação dinâmica proposto é muito oneroso em termos computacionais. Devido a este facto, foram desenvolvidas aproximações a este modelo, que permitem obter uma solução ainda razoável, com uma diminuição substancial do tempo de computação. Neste capítulo vão ser então analisadas duas aproximações baseadas no modelo de PD proposto e uma aproximação que recorre a programação não linear. A primeira versão destas aproximações permitiu testar, mais uma vez, a rede exemplo introduzida na secção 3.3. Foi também desenvolvida uma segunda versão, de cada aproximação, que possibilitou o teste das redes 1 a 5, introduzidas na secção 4.2.1.

### 5.1 Aproximação 1

#### 5.1.1 Versão 1

Esta aproximação (ver apêndice E) baseia-se no modelo de PD proposto, mas considera algumas variáveis aleatórias como sendo representadas pelos seus valores médios.

As variáveis escolhidas nesta aproximação foram os conteúdos de trabalho das actividades fixas, ou seja, para o projecto exemplo (figura 3.2), aproximou-se  $W_2$ ,  $W_3$ ,  $W_5$ ,  $W_6$ ,  $W_8$ ,  $W_9$  e  $W_{10}$  ao seu valor médio ( $\frac{1}{\lambda}$ ). Sabemos que a estimativa da data de conclusão do projecto, utilizando valores médios, subestima a data de conclusão real<sup>1</sup>. Daí que se esperem valores menores para o custo do projecto, como iremos ver pelos resultados, uma vez que o custo do atraso vai ser menor.

Com esta aproximação o programa foi substancialmente simplificado, pois os anteriores quatro valores para cada variável fixa, reduziram-se para um, o que permitiu simplificar uma série de instruções, resultando assim numa poupança de tempo substancial.

Os procedimentos que sofreram alterações em relação à versão 1, foram: ‘*mp.m*’, e ‘*mediumprojects1.m*’ até ‘*mediumprojects4.m*’, encontrando-se listados no apêndice E. No ficheiro ‘*mp.m*’ podemos ver que em lugar da instrução ‘ $Wi = generateW(lambda(i))$ ’, para as variáveis fixas, que iria gerar quatro valores, se tem ‘ $Wi = expW(i)$ ’, que gera só um valor. Como resultado desta simplificação podemos ver, por exemplo que, em ‘*mediumprojects1.m*’ as instruções:

$$\begin{aligned} X_1 &= t_4 + W_9/x_9; \\ X_2 &= t_5 + W_{10}/x_{10}; \\ Y_7 &= calculateMax(X_1, X_2); \\ PY_7 &= calculatePmax(Y_7, X_1, P, X_2, P); \end{aligned} \tag{5.1}$$

foram substituídas por:

---

<sup>1</sup>ver expressões (2.46) e (2.47).

$$\begin{aligned}
a_1 &= t_4 + W_9/x_9; \\
a_2 &= t_5 + W_{10}/x_{10}; \\
y_7 &= \max(a_1, a_2); .
\end{aligned}
\tag{5.2}$$

Este tipo de substituição representa uma poupança substancial de tempo de cálculo, uma vez que na primeira,  $X_1$  e  $X_2$  eram variáveis aleatórias, com quatro valores possíveis, originando assim um  $Y_7$  aleatório, com probabilidades associadas. Como depois da aproximação os  $W$ 's passaram a ter só um valor,  $a_1$  e  $a_2$  são números reais, sendo o cálculo do máximo muito simplificado, e feito sem necessidade de recorrer a uma função externa.

O resultado para a rede exemplo, obtido por esta aproximação, ao fim de 25 minutos, foi o seguinte:

$$\{x_1^*, x_2^*, x_3^*, x_5^*, x_6^*, x_8^*, x_9^*, x_{10}^*\} = \{1.25, 0.5, 1.0, 0.5, 0.5, 1.0, 1.0, 0.5\} \tag{5.3}$$

com um custo esperado de = 194.75.

Os resultados serão comentados na secção 5.4.1.

### 5.1.2 Versão 2

A versão 2 da aproximação 1 foi desenvolvida a partir da versão 2 original (modelo PD sem aproximações), tendo sido introduzidas simplificações no cálculo do conteúdo de trabalho das variáveis fixas. A estas variáveis foi atribuído o seu valor médio, como na secção anterior.

Esta versão como é genérica, permitiu testar as redes 1 a 5, introduzidas na secção 4.2.1. Os procedimentos que foram alterados em relação à versão 2 original, estão listados no apêndice F).

Vamos só realçar alguns dos aspectos mais importantes desta implementação.

Uma vez que o conteúdo de trabalho das actividades vai ter, no caso das variáveis fixas, só um valor, e no caso das variáveis de decisão quatro valores, utiliza-se uma estrutura (5.4), que permite a introdução de diferentes tipos de dados na mesma variável.

$$W = \text{struct}('w', \{\}) \quad (5.4)$$

Assim, por exemplo, a actividade 1 pode ter um conteúdo de trabalho médio, e a 2 um conteúdo de trabalho aleatório, como se pode ver em (5.5).

$$\begin{aligned} W(1).w &= 10, \\ W(2).w &= [2 \quad 7 \quad 11 \quad 16]. \end{aligned} \quad (5.5)$$

A função '*generateWs.m*' foi então alterada, testando se cada actividade pertence ao caminho mais longo (ALP). Se isso acontecer, ela é uma variável de decisão, sendo o seu conteúdo de trabalho gerado pela função '*generateW.m*', já introduzida no capítulo 4.

No caso da actividade não pertencer ao caminho mais longo, ela é uma variável fixa, e irá conter um único valor, representativo da média

Os resultados obtidos, para as redes 1 a 5, irão ser apresentados na secção 5.4.2, conjuntamente com os resultados das outras aproximações.

## 5.2 Aproximação 2

### 5.2.1 Versão 1

Na mesma linha da aproximação anterior, optou-se agora por reduzir todos os conteúdos de trabalho das actividades ao seu valor médio. Assim, também os conteúdos de trabalho das variáveis de decisão foram aproximados. Isto permitiu simplificar ainda mais o programa (ver no apêndice G os procedimentos alterados). Por exemplo, enquanto na aproximação 1 (versão 1), no primeiro estágio,

era necessário um ciclo ‘for’ para percorrer todos os valores de  $W_{11}$ , isso deixou de acontecer, permitindo simplificar ainda mais o código. Além disso, durante o cálculo, uma série de variáveis aleatórias foram reduzidas a valores únicos, como é o caso de  $TC$  - ‘*TardinessCost*’ (ver tabela 5.1 para comparação de instruções equivalentes nas duas aproximações).

Como resultado destas simplificações, o tempo de computação diminuiu consideravelmente. O valor do custo diminuiu também, como seria de esperar, devido à utilização de mais valores médios como substituição de variáveis aleatórias. Obteve-se então, para o projecto exemplo (figura 3.2), o resultado (5.6), que será depois comentado.

$$\{x_1^*, x_2^*, x_3^*, x_5^*, x_6^*, x_8^*, x_9^*, x_{10}^*\} = \{1.0, 0.5, 1.0, 0.5, 0.5, 1.0, 1.0, 0.5\} \quad (5.6)$$

com um custo esperado de  $= 168.94$ .

O tempo necessário para a obtenção deste resultado diminuiu de 25 para 4 minutos.

### 5.2.2 Versão 2

A versão 2 da aproximação 2 foi desenvolvida a partir da versão 2 original (modelo PD sem aproximações), tendo sido reduzido o conteúdo de todas as actividades ao seu valor médio.

Assim, em vez de ser chamada a função ‘*generateWs.m*’, para gerar os conteúdos de trabalho das actividades, eles passaram a ser calculados pela instrução

$$W = \exp W; \quad (5.7)$$

que atribui a cada  $W$  o valor médio da distribuição exponencial respectiva. Esta simplificação permitiu reduzir o programa, tendo sido eliminadas todas as instruções relacionadas com o cálculo de probabilidades.

Tabela 5.1: Comparação de instruções para as aproximações 1 e 2

Aproximação 1	Aproximação 2
for $x_{11} = 0.5 : 0.25 : 1.5$	for $x_{11} = 0.5 : 0.25 : 1.5$
$ev = 0;$	$ev = 0;$
for $i_{11} = 1 : 4$	
$w_{11} = W_{11}(i_{11});$	
$t_7 = \max(y_7, t_6 + w_{11}/x_{11});$	$t_7 = \max(y_7, t_6 + W_{11}/x_{11});$
$u = \max(0, t_7 - T);$	$u = \max(0, t_7 - T);$
$TC(i_{11}) = cl \times u;$	$tc = cl \times u;$
end % 4 values - Probability=P	
$rc_{11} = x_{11} \times \exp W(11);$	$rc_{11} = x_{11} \times W_{11};$
$F_1 = TC + rcf + rc_{11};$ % total cost	$f_1 = tc + rcf + rc_{11};$ % total cost
$ev = \text{expectedValue}(F_1, P);$	
$EV = [EV; x_{11}, ev];$	$F_1 = [F_1; x_{11}, f_1];$
end	end

Os procedimentos que sofreram alterações estão listados no apêndice H.

Os resultados obtidos para as redes 1 a 5 (introduzidas na secção 4.2.1) vão ser apresentados na secção 5.4.2.

## 5.3 Aproximação 3

### 5.3.1 Versão 1

Ao aproximarmos todos os conteúdos de trabalho das actividades pelo seu valor médio, o problema fica reduzido a um problema de programação não linear. Sendo assim, o modelo de PD proposto deixa de ser eficiente. Foi seguindo este princípio que foi desenvolvida a aproximação 3. O modelo de PD foi substituído por um modelo de PNL. Esse modelo foi desenvolvido utilizando o *Solver* do *Excel*, sendo



definido da seguinte forma: pretende-se minimizar o custo total do projecto, ou seja, o custo dos recursos, mais o custo do atraso. A função não linear que representa o custo total, para o projecto exemplo (figura 3.2), é a seguinte:

$$F(x) = x_1W_1 + x_2W_2 + \dots + x_{11}W_{11} + c_L \max(0, t_7 - T). \quad (5.8)$$

Os parâmetros introduzidos foram os valores de  $\lambda$  para as várias actividades, sendo assim possível calcular os conteúdos de trabalho

$$W = \frac{1}{\lambda}. \quad (5.9)$$

Foi também introduzido o valor de  $T = 65$  e  $c_L = 5$ . O objectivo é determinar os valores de  $x$ , que minimizem a função  $F(x)$ .

O problema que surge é como definir  $t_7$ , o tempo de realização do último nó da rede. Sabemos contudo que

$$t_7 = \max\left(t_4 + \frac{W_9}{x_9}, t_5 + \frac{W_{10}}{x_{10}}, t_6 + \frac{W_{11}}{x_{11}}\right). \quad (5.10)$$

Então precisamos de saber qual o valor de  $t_4$ ,  $t_5$  e  $t_6$ . Estes valores podem ser obtidos por:

$$\begin{aligned} t_4 &= t_1 + \frac{W_3}{x_3} \\ t_5 &= \max\left(t_2 + \frac{W_5}{x_5}, t_4 + \frac{W_8}{x_8}\right) \\ t_6 &= \max\left(t_2 + \frac{W_6}{x_6}, t_3 + \frac{W_7}{x_7}\right), \end{aligned} \quad (5.11)$$

sendo

$$t_1 = 0 \quad (5.12)$$

pois é o tempo de realização do primeiro nó da rede.  $t_2$  e  $t_3$  são obtidos do seguinte modo:

$$\begin{aligned} t_2 &= t_1 + \frac{W_1}{x_1} \\ t_3 &= \max\left(t_1 + \frac{W_2}{x_2}, t_2 + \frac{W_4}{x_4}\right). \end{aligned} \quad (5.13)$$

Temos então quase todas as restrições definidas. Só falta definir os valores limites dos  $x$ 's, o que foi feito da seguinte forma: para cada  $x$ , foi introduzida uma restrição do tipo

$$0.5 \leq x \leq 1.5. \quad (5.14)$$

Depois das restrições, e da função objectivo definidas, foi feita a minimização, tendo-se obtido o seguinte resultado:

$$\begin{aligned} \{x_1^*, x_2^*, x_3^*, x_4^*, x_5^*, x_6^*, x_7^*, x_8^*, x_9^*, x_{10}^*, x_{11}^*\} = \\ \{0.91, 0.5, 1.12, 0.9, 0.5, 0.55, 1.04, 0.71, 0.88, 0.55, 0.76\} \end{aligned} \quad (5.15)$$

com um custo esperado de  $= 161.51$ .

A folha de cálculo em Excel que permitiu resolver este problema pode ser vista na figura 5.1.

### 5.3.2 Versão 2

Também para esta aproximação, foram testadas as redes 1 a 5 (introduzidas na secção 4.2.1). As respectivas folhas de cálculo são apresentadas nas figuras 5.2 a 5.6. Os resultados serão comentados na secção 5.4.2.

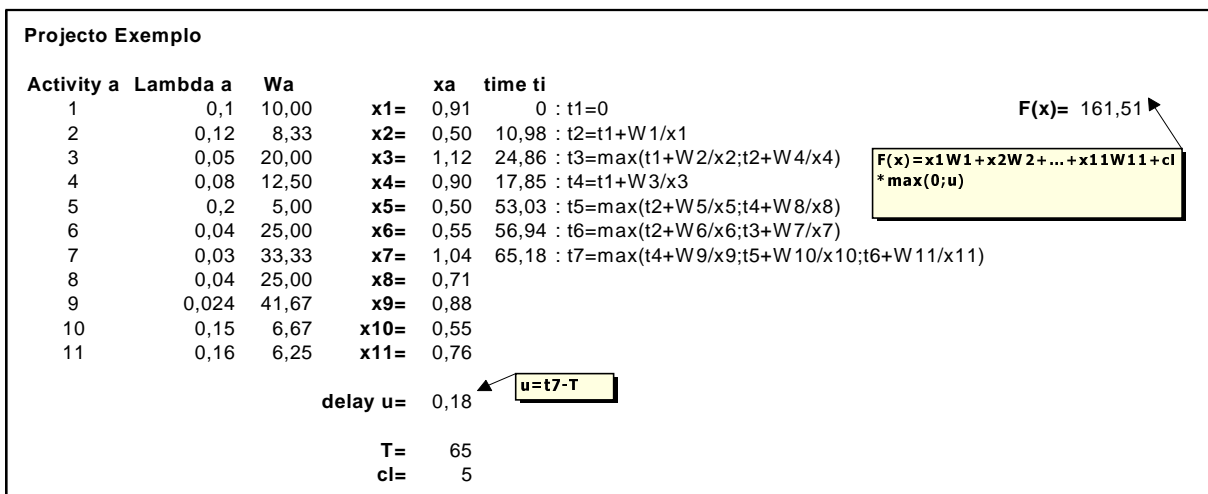


Figura 5.1: Projecto Exemplo (Solver do Excel)

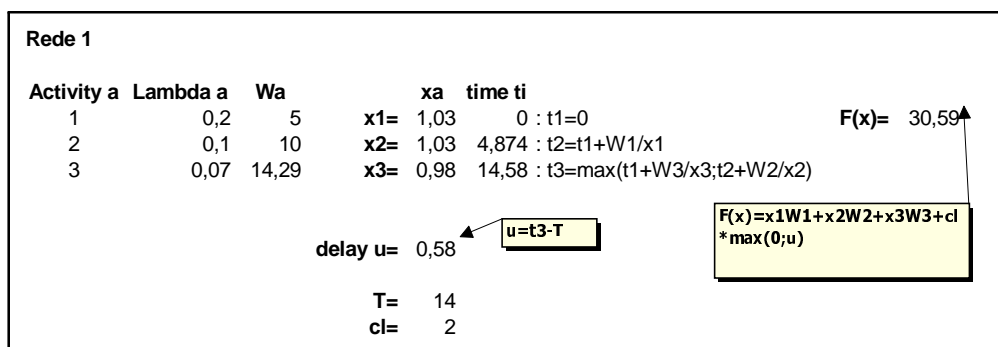


Figura 5.2: Rede 1 (Solver do Excel)

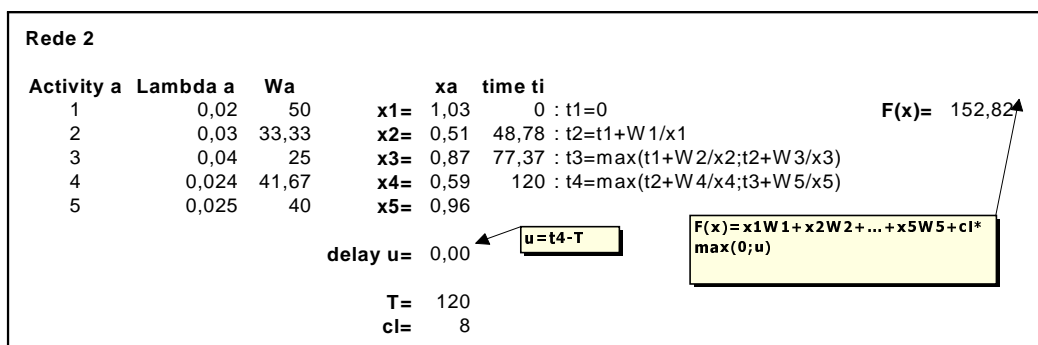


Figura 5.3: Rede 2 (Solver do Excel)

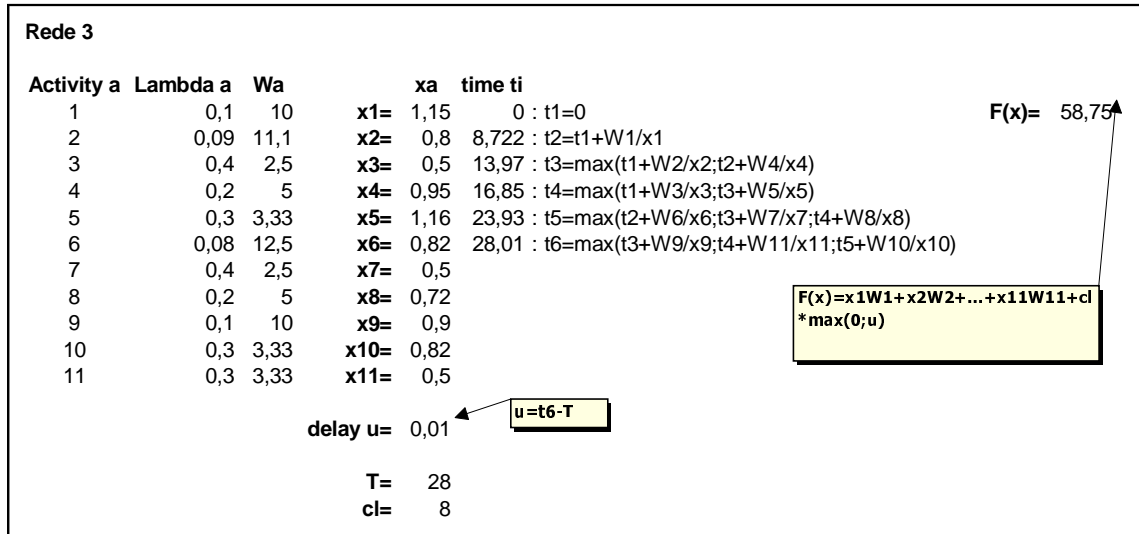


Figura 5.4: Rede 3 (Solver do Excel)

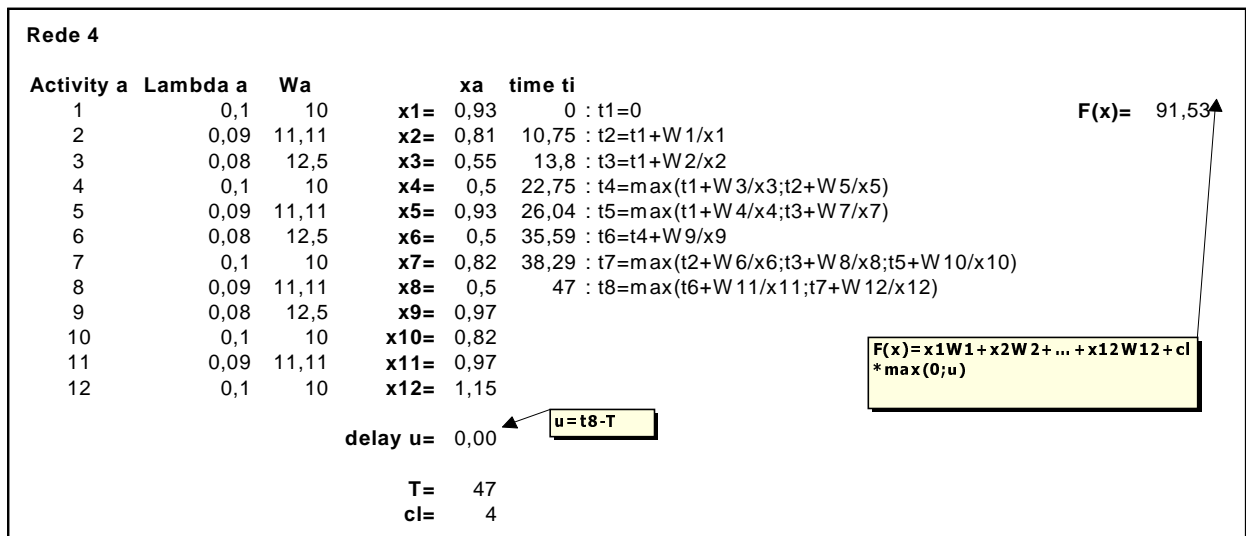


Figura 5.5: Rede 4 (Solver do Excel)

## 5.4 Comparação dos resultados

### 5.4.1 Versão 1

Nesta secção vão ser discutidos os resultados obtidos pela versão 1 dos vários programas desenvolvidos. Esta versão, como já foi referido, foi desenvolvida para

Rede 5				
Activity a	Lambda a	Wa	xa	time ti
1	0,06	16,67	<b>x1=</b> 0,689	0 : t1=0
2	0,04	25	<b>x2=</b> 1,295	24,2 : t2=t1+W1/x1
3	0,1	10	<b>x3=</b> 0,5	19,31 : t3=t1+W2/x2
4	0,07	14,29	<b>x4=</b> 0,667	20 : t4=t1+W3/x3
5	0,08	12,5	<b>x5=</b> 0,5	45,62 : t5=max(t2+W4/x4;t3+W5/x5)
6	0,04	25	<b>x6=</b> 1,286	38,74 : t6=t3+W6/x6
7	0,08	12,5	<b>x7=</b> 0,5	45 : t7=t4+W7/x7
8	0,2	5	<b>x8=</b> 0,5	30 : t8=t4+W8/x8
9	0,07	14,29	<b>x9=</b> 0,661	67,23 : t9=t5+W9/x9
10	0,05	20	<b>x10=</b> 1,142	54,18 : t10=t8+W12/x12
11	0,08	12,5	<b>x11=</b> 0,5	82,47 : t11=max(t6+W10/x10;t9+W13/x13)
12	0,07	14,29	<b>x12=</b> 0,591	93,33 : t12=max(t10+W15/x15;t11+W14/x14)
13	0,09	11,11	<b>x13=</b> 0,729	74,41 : t13=max(t7+W11/x11;t10+W16/x16)
14	0,09	11,11	<b>x14=</b> 1,023	110 : t14=max(t12+W17/x17;t13+W18/x18)
15	0,05	20	<b>x15=</b> 0,584	
16	0,09	11,11	<b>x16=</b> 0,549	
17	0,04	25	<b>x17=</b> 1,5	
18	0,06	16,67	<b>x18=</b> 0,557	

**F(x)=** 144,07▲

$$F(x) = x_1W_1 + x_2W_2 + \dots + x_{18}W_{18} + cl * \max(0; u)$$

**delay u=** 0,00 ▲ u=t14-T

**T=** 110  
**cl=** 10

Figura 5.6: Rede 5 (Solver do Excel)

ser aplicada especificamente a um único projecto exemplo (figura 3.2). Tem a vantagem de ser mais rápida a executar e mais simples de perceber, em termos de implementação.

Na tabela 5.2 encontram-se os resultados obtidos, para o projecto exemplo, das três aproximações introduzidas neste capítulo, conjuntamente com o resultado do modelo de PD puro.

O que realça imediatamente é a substancial redução do tempo de computação, à medida que se aproximam mais variáveis. Quando passamos da PD para a PNL, então a redução de tempo é enorme.

Poderíamos dizer que o valor de 280.85 é um valor óptimo, se não tivéssemos feito as discretizações dos  $W$ 's, dos  $x$ 's e dos  $t$ 's, necessárias para uma execução em tempo útil. Num ambiente de computação ideal, o modelo de PD proposto, permitiria chegar ao verdadeiro óptimo. Ele é teoricamente um modelo

Tabela 5.2: Comparação de resultados para as aproximações 1, 2 e 3

	Modelo PD	PD Aprox. 1	PD Aprox. 2	PNL
$x_1$	1.25	1.25	1.0	0.91
$x_2$	0.5	0.5	0.5	0.5
$x_3$	1.5	1.0	1.0	1.12
$x_4$				0.9
$x_5$	0.5	0.5	0.5	0.5
$x_6$	1.0	0.5	0.5	0.55
$x_7$				1.04
$x_8$	1.0	1.0	1.0	0.71
$x_9$	1.5	1.0	1.0	0.88
$x_{10}$	1.0	0.5	0.5	0.55
$x_{11}$				0.76
$EV(Cost)$	280.85	194.75	168.94	161.51
$Run Time$	141 min.	25 min.	4 min.	1 seg.

optimizante. Mas como esse ambiente ainda não existe, sabemos que o valor 280.85 é um limite superior para o custo total esperado, ou seja, o verdadeiro óptimo estará abaixo deste valor. Quando fizemos a análise de sensibilidade (secção 3.4) vimos que, com outra combinação de  $x$ 's, foi possível obter um valor inferior para o custo total esperado (274.86), o que reforça esta conclusão.

Para analisar os restantes resultados, temos que ter em conta que o CPM subestima a verdadeira data de conclusão do projecto, quando são utilizadas médias<sup>2</sup>. Daí que o valor do custo total esperado para o modelo de PD seja o maior, e depois vá diminuindo quantas mais aproximações à média utilizarmos. Assim, comparativamente com o modelo de PD original, a aproximação 1 deu um valor inferior, para o custo total esperado, uma vez que substituímos  $W$  pela sua média, para 7 actividades, e a aproximação 2 deu um valor ainda menor, uma

<sup>2</sup>Ver expressões (2.46) e (2.47).

vez que  $W$  foi substituído pela sua média em 11 actividades. Não nos podemos esquecer contudo, que estas duas aproximações também não são exactas, devido às discretizações necessárias para utilizar PD.

O erro cometido quando se fizeram as discretizações, pode ser medido, caso a caso, se compararmos os resultados das aproximações 2 e 3. Ambas as aproximações, substituem todos os conteúdos de trabalho das actividades, pelas suas médias. Portanto, deveriam dar o mesmo resultado. A diferença verificada é precisamente o erro da discretização, quando se usa PD, se considerarmos que o modelo de PNL encontrou um óptimo absoluto. A grandeza desse erro, para o projecto exemplo, é cerca de +4,6%, e pode dar uma ideia do erro associado aos restantes valores obtidos por PD. Podemos assim ter uma ideia da ordem de grandeza do verdadeiro óptimo:  $280.85 - 4.6\% \times 280.85 \approx 268$ .

A análise do valor esperado do custo é importante, mas será importante relembrar que o objectivo da análise era determinar que quantidade de recursos atribuir às actividades, para minimizar o custo total. Então devemos analisar também os valores de  $x$ . Em primeiro lugar, uma diferença realça nos resultados: a aproximação 3 apresenta valores para todos os  $x$ 's, enquanto as outras aproximações não têm valores para  $x_4$ ,  $x_7$  e  $x_{11}$ . Isto deve-se contudo à natureza adaptativa da PD. Os valores de  $x_4$ ,  $x_7$  e  $x_{11}$ , que foram variáveis de decisão, só podem ser fornecidos quando chegar o momento de decidir, pois dependem do estado do sistema nessa altura (ver estes valores no apêndice D).  $x_2$  e  $x_5$  dão precisamente o mesmo valor para todas as aproximações, daí podermos concluir que serão os melhores valores. Os outros  $x$ 's já variam um pouco mais, apresentando contudo algumas tendências que podem ser indicativas do sentido dos seus valores óptimos. Por exemplo,  $x_8$  deverá estar muito próximo de 1.0, e não deve ser menor que 0.7. Ao seguir essas tendências poderemos obter não o óptimo, mas valores próximos do óptimo para o custo total esperado.

### 5.4.2 Versão 2

A versão 2 é a mais genérica, tendo sido desenvolvida para permitir testar o modelo para qualquer rede de actividades. Foram usados cinco exemplos (da rede 1 à rede 5, introduzidas na secção 4.2.1), para os quais vamos apresentar os resultados obtidos (tabelas 5.3 a 5.7).

Tabela 5.3: Comparação de resultados para a rede 1

<b>Rede 1</b>	<b>Modelo PD</b>	<b>PD Aprox. 1</b>	<b>PD Aprox. 2</b>	<b>PNL</b>
$x_1$	1.0	1.25	1.25	1.03
$x_2$				1.03
$x_3$	1.0	1.0	1.0	0.98
$EV(Cost)$	45.53	37.37	31.11	30.59
$Run Time$	4.8 seg.	4.6 seg.	4.2 seg.	1 seg.

Tabela 5.4: Comparação de resultados para a rede 2

<b>Rede 2</b>	<b>Modelo PD</b>	<b>PD Aprox. 1</b>	<b>PD Aprox. 2</b>	<b>PNL</b>
$x_1$	1.0	1.0	1.25	1.03
$x_2$	1.0	1.0	0.5	0.51
$x_3$				0.87
$x_4$	1.5	1.0	0.5	0.59
$x_5$				0.96
$EV(Cost)$	304.62	279.14	148.79	152.82
$Run Time$	9.6 seg.	8.4 seg.	6 seg.	1 seg.



Tabela 5.5: Comparação de resultados para a rede 3

<b>Rede 3</b>	<b>Modelo PD</b>	<b>PD Aprox. 1</b>	<b>PD Aprox. 2</b>	<b>PNL</b>
$x_1$	1.25	1.25	0.75	1.15
$x_2$	1.0	1.0	1.0	0.8
$x_3$	0.5	0.5	0.5	0.5
$x_4$				0.95
$x_5$				1.16
$x_6$	1.0	0.5	1.0	0.82
$x_7$	0.5	0.5	0.5	0.5
$x_8$				0.72
$x_9$	1.0	0.5	1.0	0.9
$x_{10}$				0.82
$x_{11}$	1.0	0.5	0.5	0.5
$EV(Cost)$	106.76	75.04	58.94	58.75
$Run Time$	30 min.	13 min.	6 min.	1 seg.

Os resultados obtidos mostram, de uma forma geral, as mesmas tendências dos resultados obtidos para a rede exemplo, estudada na versão 1.

Assim, verifica-se uma redução progressiva no tempo de computação, quando passamos do modelo de PD puro, para as aproximações 1 e 2, e esse tempo é drasticamente reduzido quando usamos PNL.

Em relação ao valor do custo total esperado, verifica-se uma tendência geral para a sua diminuição, à medida que se aproximam mais conteúdos de trabalho das actividades, às suas médias, como acontece com o projecto exemplo, pelo motivo já exposto na secção anterior.

Em relação à PNL, a sua tendência geral é para apresentar um valor do custo total esperado inferior ao das restantes aproximações. Isto acontece, como já foi referido, por não ter associado o erro das discretizações do modelo de PD. É excepção a rede 2 que, na aproximação 2, conseguiu obter um valor inferior

Tabela 5.6: Comparação de resultados para a rede 4

Rede 4	Modelo PD	PD Aprox. 1	PD Aprox. 2	PNL
$x_1$	1.25	1.25	0.75	0.93
$x_2$	1.0	0.5	1.0	0.81
$x_3$	1.0	1.0	0.5	0.55
$x_4$	0.5	0.5	0.5	0.5
$x_5$				0.93
$x_6$	1.0	0.5	0.5	0.5
$x_7$	1.0	0.5	1.0	0.82
$x_8$	0.5	0.5	0.5	0.5
$x_9$				0.97
$x_{10}$	1.0	0.5	1.0	0.82
$x_{11}$				0.97
$x_{12}$	1.5	1.0	1.0	1.15
$EV(Cost)$	182.91	117.92	103.26	91.53
$Run Time$	23 h.	7 h.	3 h.	1 seg.

ao da PNL, apesar de próximo. Isto explica-se contudo pelo facto das técnicas disponíveis no Solver do Excel<sup>3</sup> não garantirem a obtenção de óptimos globais. O Solver garante sim que todas as restrições são satisfeitas e que o valor encontrado é um mínimo. Houve neste caso a “sorte” de uma das discretizações utilizadas na PD gerar um valor mínimo melhor do que o obtido pela PNL. Mas este será um caso excepcional.

Quanto ao valor dos conteúdos de trabalho das actividades, convém mais uma vez frisar que, quando usamos PD, são apresentados só alguns valores, o que não acontece quando usamos PNL. Isto deve-se à natureza adaptativa da PD, que irá ditando os restantes valores, à medida que o projecto avança no tempo.

<sup>3</sup>No Solver do Excel, a procura do óptimo pode ser feita utilizando o método de Quasi-Newton ou o método do Gradiente Conjugado.

Tabela 5.7: Comparação de resultados para a rede 5

Rede 5	Modelo PD	PD Aprox. 1	PD Aprox. 2	PNL
$x_1$	0.75	1.25	0.5	0.69
$x_2$	1.25	1.25	0.75	1.3
$x_3$	1.25	0.75	0.75	0.5
$x_4$				0.67
$x_5$	0.75	0.75	0.75	0.5
$x_6$	1.25	1.25	0.75	1.29
$x_7$	0.75	0.75	0.75	0.5
$x_8$	1.25	0.75	0.75	0.5
$x_9$				0.66
$x_{10}$	1.25	0.75	0.75	1.14
$x_{11}$	1.25	0.75	0.75	0.5
$x_{12}$	0.75	0.75	0.75	0.59
$x_{13}$				0.73
$x_{14}$				1.02
$x_{15}$	0.75	0.75	0.75	0.58
$x_{16}$	1.25	0.75	0.75	0.55
$x_{17}$				1.5
$x_{18}$	0.75	0.75	0.75	0.56
$EV(Cost)$	339.07	271.38	210.19	144.07
$Run Time$	5 dias	9 h.	2 h.	1 seg.

Se percorrermos as várias tabelas de resultados, nota-se que existem tendências claras, para alguns dos valores de  $x$ . Por exemplo, para a rede 3,  $x_3$  e  $x_7$  são sempre iguais a 0.5, o mesmo acontecendo com  $x_4$  e  $x_8$  da rede 4. Este facto dará uma indicação clara do valor a utilizar nestes casos.

Noutros casos, temos valores iguais quando utilizamos PD e ligeiramente diferentes quando utilizamos PNL (Rede 1 -  $x_3$ , Rede 3 -  $x_2$ , Rede 5 -  $x_5$ ,  $x_7$ ,  $x_{12}$ ,

$x_{15}, x_{18}$ ). Isto deve-se essencialmente à discretização necessária aos modelos de PD, e uma opção será utilizar um valor médio entre o valor indicado por PD e o valor indicado por PNL.

Existem outros casos em que os valores demonstram maiores variações, o que pode representar que o respectivo valor é pouco significativo para o resultado final.

Estes modelos podem assim ser utilizados conjuntamente, para auxiliar na alocação de recursos de um projecto real.

# Capítulo 6

## Aplicação do algoritmo

### “Electromagnético”

Uma vez que a PD exige um enorme esforço em termos computacionais, para problemas com muitas variáveis (‘curse of dimensionality’ [Bellman57]), e a PNL não consegue resolver o problema estocástico, procurou-se testar neste capítulo, a aplicação de uma nova técnica de optimização global, desenvolvida por Birbil e Fang ([BirbilFang00], [Birbil02], [BirbilFang02], [BirbilFangSheu02], [BirbilFang02b]), que tem por nome algoritmo “Electromagnético” (EM). Esta técnica foi desenvolvida para optimizar funções com uma ou mais variáveis, sendo necessário introduzir a fórmula que representa a função. Para ser aplicada ao problema em estudo, este algoritmo necessitou de várias adaptações.

#### 6.1 Algoritmo EM

O algoritmo EM, como o próprio nome indica, baseia-se nos princípios do electromagnetismo: dois corpos exercem forças de atracção ou repulsão entre si, conforme as cargas que possuem. O algoritmo EM inicia com a geração de pontos aleatórios. Podemos pensar em cada um desses pontos como uma partícula, com uma carga associada, que é libertada no espaço. A carga de cada ponto

está relacionada com o valor da função objectivo, que pretendemos otimizar. Essa carga vai determinar a magnitude de atracção ou repulsão do ponto sobre o resto da população - quanto melhor for o valor da função objectivo, maior será a magnitude de atracção.

O algoritmo EM baseia-se nesta analogia, para resolver problemas de optimização global. A ideia subjacente é que o grande poder de atracção dos mínimos locais e globais encoraja os pontos a convergir nessa direcção. Assim, as forças de atracção direccionam os pontos para melhores regiões, enquanto as forças de repulsão permitem às partículas explorar regiões não visitadas.

O princípio da sobreposição [Cowan68] também é aqui aplicado: a força electromagnética, numa partícula, causada por outras partículas, pode ser calculada adicionando vectorialmente as forças causadas por cada uma das outras partículas, separadamente. Assim, cada partícula vai-se movimentar na direcção estabelecida, resultante das forças exercidas, por todas as outras partículas do espaço.

O algoritmo EM consiste em quatro fases: a inicialização do algoritmo, o cálculo do vector da força total exercida em cada partícula, o movimento na direcção da força resultante e a aplicação de uma procura local. Este algoritmo foi concebido para resolver problemas do tipo (6.1), com  $S$  definido como em (6.2), sendo dados os seguintes parâmetros:

$n$ : dimensão do problema,

$l_k, u_k$ : limites inferior e superior na dimensão  $k$ ,

$m$ : número de pontos da amostra,

*MAXITER*: máximo número de iterações,

*LSITER*: máximo número de iterações para procura local,

$\delta$ : parâmetro da procura local,  $\delta \in [0, 1]$ .

$$\text{minimizar } f(x) \text{ sujeito a } x \in S \quad (6.1)$$

$$S := \{x_k \in \mathfrak{R} \mid l_k \leq x_k \leq u_k, l_k, u_k \in \mathfrak{R} \text{ para } k = 1, \dots, n\} \quad (6.2)$$

O algoritmo é apresentado no apêndice I, remetendo-se explicações mais pormenorizadas para as referências dos autores ([BirbilFang00], [Birbil02], [BirbilFang02], [BirbilFangSheu02], [BirbilFang02b]).

## 6.2 Aplicação do algoritmo EM ao RCPSP

### 6.2.1 Versão 1

A aplicação do algoritmo EM, tal como ele foi concebido, pressupõe a existência de uma função a otimizar do tipo (6.3). No caso do RCPSP, não temos propriamente uma função deste tipo. O que temos é um conjunto de actividades, com conteúdos de trabalho associados, que seguem uma dada distribuição de probabilidades; temos quantidades de recursos para atribuir às actividades, e pretende-se saber quais as quantidades óptimas, para minimizar o custo total do projecto.

Para o caso do projecto exemplo (ver figura 3.2), os valores de  $x$  serão as quantidades de recurso a atribuir às actividades  $(x_1, x_2, x_3, \dots, x_{11})$ . Assim, teremos um problema com 11 dimensões. A função  $f(x)$  será o custo total do projecto. Este custo engloba o custo dos recursos e o custo do atraso.

A aplicação do algoritmo EM faz-se do seguinte modo: em primeiro lugar é gerado aleatoriamente um conjunto de  $W$ 's (conteúdos de trabalho das actividades), com base nos parâmetros  $\lambda$  fornecidos para a distribuição exponencial. Para esse conjunto de  $W$ 's, é utilizado o algoritmo EM modificado, que irá gerar um conjunto de  $x$ 's óptimo, que minimiza o custo total  $f$ .

Este processo é repetido várias vezes, sendo sempre gerados novos  $W$ 's e obtidos os respectivos  $x$ 's e  $f$  óptimos.

A solução final será obtida calculando a média dos resultados encontrados nas diferentes experiências.

Depois desta explicação geral, vamos ver mais concretamente como foi utilizado o algoritmo EM para otimizar o custo total, para cada conjunto  $W$  gerado.

Em primeiro lugar são definidos os parâmetros principais do problema,

$$n = 11 \text{ (dimensão do problema)}$$

$$m = 15 \text{ (número de pontos da amostra)}$$

$$l_k = 0.5 \times 11 \text{ (limites inferiores)}$$

$$u_k = 1.5 \times 11 \text{ (limites superiores)}$$

$$MAXITER = 275$$

$$LSITER = 1$$

$$\delta = 0.05$$

Os valores de  $m$ ,  $MAXITER$ ,  $LSITER$  e  $\delta$  foram escolhidos por permitirem chegar a resultados satisfatórios num curto espaço de tempo.

O algoritmo EM começa então por gerar aleatoriamente  $m$  pontos, ou seja  $m$  conjuntos de  $x$  ( $x_1 \dots x_{11}$ ). Para cada um desses pontos, é calculado o valor da função objectivo.

Originalmente este algoritmo tinha uma função ‘*func.m*’, onde era determinado  $f(x)$ , para cada ponto  $x$ , do tipo:

$$\text{function val} = \text{func}(x) \tag{6.3}$$

$$\text{val} = (x(1))^3 - 3 \times x(1) \times x(2)^2 - 1.0)^2 + (3 \times x(1))^2 \times x(2) - x(2)^3)^2.$$

Para aplicar este algoritmo ao nosso caso, esta função foi adaptada para permitir calcular o custo total do projecto, para o conjunto  $W$  previamente gerado e para cada conjunto de  $x$ ’s, da seguinte forma: calcula-se, pela ordem apresentada,



1. O custo dos recursos,

$$rc = \sum_{i=1}^{11} x_i W_i. \quad (6.4)$$

2. A duração das actividades,

$$y = (y_1 \ y_2 \ \dots \ y_{11}) \quad \text{com} \quad y_i = \frac{W_i}{x_i} \quad (6.5)$$

3. O tempo de realização do último nó da rede ( $t_7$ ), usando CPM e a informação sobre as durações.

4. O custo do atraso,

$$tc = \max(0, t_7 - T). \quad (6.6)$$

5. O custo total,

$$f = rc + tc. \quad (6.7)$$

Depois de ter o valor da função objectivo, para cada conjunto de  $m$  pontos, calcula-se o valor mínimo, as cargas dos vários pontos, e as forças resultantes em cada ponto  $m$  gerado. Cada ponto será depois movido na direcção da força calculada, obtendo-se um novo conjunto de  $m$  pontos. O procedimento repete-se até atingirmos o número de iterações especificado. Nessa altura, e segundo os autores, o algoritmo deverá convergir para um valor óptimo ou quase óptimo. Para garantir esta convergência, o número de iterações máximo deve ser de  $25 \times n$  ( $MAXITER = 25 \times n$ ). No nosso caso, o número de iterações máximo será  $25 \times 11 = 275$ . Este valor será óptimo para o conjunto  $W$  gerado aleatoriamente, no início do processo.

Este processo foi repetido, no nosso caso, 100 vezes<sup>1</sup>, tendo-se obtido, ao fim de 45 minutos, os seguintes resultados:

---

<sup>1</sup>Quanto maior for o número de repetições, maior será a segurança relativamente ao resultado.

$$\{x_1^*, x_2^*, x_3^*, x_4^*, x_5^*, x_6^*, x_7^*, x_8^*, x_9^*, x_{10}^*, x_{11}^*\} = \{0.94, 0.6, 0.96, 0.83, 0.5, 0.64, 0.91, 0.74, 0.81, 0.74, 0.95\} \quad (6.8)$$

com um custo esperado de  $= 263.75$ .

Como podemos ver, os resultados são os esperados, ou seja, o valor do custo total esperado do projecto é inferior ao valor obtido por PD, e superior ao valor obtido por PNL.

O tempo de execução deste algoritmo é relativamente elevado, devido à necessidade da geração de um grande conjunto de pontos aleatórios, para representar eficientemente o conteúdo de trabalho real das actividades. A sua utilização para otimizar funções do tipo (6.3) é bastante mais eficiente, conseguindo-se resultados ao fim de alguns segundos.

### 6.2.2 Versão 2

Nesta secção, e para manter a mesma estrutura do capítulo anterior, vamos apresentar o resultado do algoritmo EM aplicado às redes 1 a 5 (introduzidas na secção 4.2.1), bem como o tempo necessário à sua obtenção.

#### Rede 1

$$\{x_1^*, x_2^*, x_3^*\} = \{0.8, 0.8, 0.83\} \quad (6.9)$$

com um custo esperado de  $= 44.99$

8 min.

**Rede2**

$$\{x_1^*, x_2^*, x_3^*, x_4^*, x_5^*\} = \{0.99, 0.59, 0.82, 0.69, 0.91\} \quad (6.10)$$

com um custo esperado de = 277.77

15 min.

**Rede 3**

$$\{x_1^*, x_2^*, x_3^*, x_4^*, x_5^*, x_6^*, x_7^*, x_8^*, x_9^*, x_{10}^*, x_{11}^*\} = \{1.11, 0.8, 0.5, 0.89, 0.87, 0.8, 0.55, 0.83, 0.77, 1.04, 0.6\} \quad (6.11)$$

com um custo esperado de = 94.69

44 min.

**Rede 4**

$$\{x_1^*, x_2^*, x_3^*, x_4^*, x_5^*, x_6^*, x_7^*, x_8^*, x_9^*, x_{10}^*, x_{11}^*, x_{12}^*\} = \{0.83, 0.95, 0.66, 0.57, 0.74, 0.61, 0.8, 0.65, 0.88, 0.87, 0.88, 1.05\} \quad (6.12)$$

com um custo esperado de = 146.29

52 min.

**Rede 5**

$$\{x_1^*, x_2^*, x_3^*, x_4^*, x_5^*, x_6^*, x_7^*, x_8^*, x_9^*, x_{10}^*, x_{11}^*, x_{12}^*, x_{13}^*, x_{14}^*, x_{15}^*, x_{16}^*, x_{17}^*, x_{18}^*\} = \{0.73, 0.98, 0.67, 0.71, 0.71, 0.83, 0.56, 0.65, 0.88, 0.84, 0.55, 0.68, 0.88, 1.1, 0.6, 0.55, 1.14, 0.61\}$$

com um custo esperado de = 306.61 (6.13)

1 h. 46 min.

### 6.3 Aspectos da implementação

No apêndice J encontra-se o código, desenvolvido em MatLab, para aplicar o algoritmo EM ao RCPSP.

Nesta secção vamos referir alguns aspectos mais relevantes da implementação desta aplicação.

Em primeiro lugar, no procedimento ‘*GOfunc.m*’, foram definidos os parâmetros do problema. Esses parâmetros incluem  $\lambda$ , o parâmetro da distribuição exponencial;  $T$ , a data de conclusão do projecto;  $cl$ , o custo unitário do atraso; para além dos parâmetros referidos na secção 6.2.1.

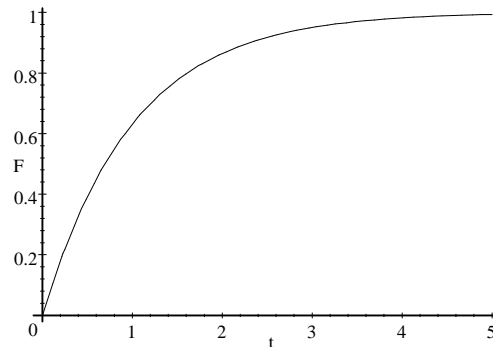
De seguida, foi necessário gerar aleatoriamente valores representativos dos conteúdos de trabalho das actividades. Por opção, e para manter equivalentes os vários modelos, esses valores seguem a distribuição exponencial. Esta geração de valores é contudo diferente da utilizada para o modelo de PD, onde foram gerados quatro valores discretos, representativos da distribuição. No caso deste modelo, pretendia-se gerar um valor aleatório, proveniente da distribuição exponencial. Mas no Matlab não existe uma função geradora de valores aleatórios desta distribuição. Existe sim a possibilidade de gerar valores de uma distribuição uniforme.

Assim, se partirmos da função exponencial cumulativa (ver figura 6.1), os valores desta função podem ser representados por uma distribuição uniforme entre 0 e 1.

Assim, se fizermos,

$$y = 1 - e^{-\lambda x} \tag{6.14}$$

e gerarmos  $y = \text{rand}(0, 1)$ , como uma variável aleatória, uniformemente distribuída entre 0 e 1, o que o Matlab permite, podemos obter  $x$ , da seguinte forma,

Figura 6.1:  $F(t) = 1 - e^{-\lambda t}$ 

$$\begin{aligned}
 y = 1 - e^{-\lambda x} &\Leftrightarrow e^{-\lambda x} = 1 - y \Leftrightarrow \ln(e^{-\lambda x}) = \ln(1 - y) & (6.15) \\
 &\Leftrightarrow -\lambda x = \ln(1 - y) \Leftrightarrow x = -\frac{\ln(1 - y)}{\lambda}
 \end{aligned}$$

sendo então  $x$  uma variável aleatória proveniente da distribuição exponencial. Este cálculo é feito no início da função *'Initialize.m'*.

Na função *'EMVIS.m'* foi acrescentada uma estrutura, *Results*, que irá armazenar os vários resultados obtidos, ou seja, os  $W$ 's (conteúdos de trabalho das actividades) gerados aleatoriamente, e os valores óptimos obtidos para os  $x$ 's (quantidade de recursos a atribuir) e o  $f$  (valor do custo total). Uma vez que este processo foi repetido 100 vezes, irão ser geradas 100 instâncias desta estrutura. O resultado final será obtido pela função *'MeanRes.m'*, que irá indicar os valores médios dos  $W$ 's, dos  $x$ 's e do  $f$ .

A função *'TotalCost.m'* foi criada de raiz, e é chamada pelo programa para calcular o valor da função objectivo, para os vários pontos gerados, o que irá permitir depois calcular as forças e continuar com o algoritmo. Vamos então ver em mais detalhe o seu conteúdo. A parte principal desta função é composta pelas seguintes instruções.

```

c = 0;
for i = 1 : n
    c = c + X(i) × W(i);
    Y(i) = W(i)/X(i);
end
tf = cpm(Y);
c = c + cl × max(0, tf - T);

```

Através de um ciclo *for*, calcula-se o custo dos recursos das várias actividades, bem como a sua duração. Com o vector que contém a duração das várias actividades, e utilizando CPM, calcula-se o tempo de realização do último nó da rede ( $t_f$ ). Por último, calcula-se o custo total, somando o custo dos recursos com o custo do atraso.

A função ‘*Cpm.m*’ baseia-se então no método CPM para calcular o tempo de realização do último nó da rede. Esta função recebe como parâmetro um vector com as durações das várias actividades ( $Y$ ), e com base nessas durações, e na informação sobre a estrutura da rede ( $Net$ ), calcula a data de conclusão do projecto. No caso da aplicação do algoritmo EM, só foi necessário considerar as duas primeiras linhas da estrutura  $Net$ , utilizada para a implementação da versão 2 de PD (ver tabela 4.6).

É depois definido o vector  $TNode$ , que irá conter o tempo de realização de todos os nós da rede, e irá ser construído recursivamente. Inicializa-se, em primeiro lugar,  $TNode(1)$  a 0, pois supostamente o projecto começa em  $t = 0$ . Para os restantes nós da rede, e sequencialmente, determinam-se todas as origens e respectivas actividades. Isto é feito com o procedimento ‘*OriginNodes.m*’. O tempo de realização de um nó será então o máximo valor obtido, somando o tempo de realização dos nós origem, com a duração das respectivas actividades,

$$TNode_j = \max_{i \rightarrow j} \{TNode_i + Y\}; \quad j = 2, \dots, n \quad (6.16)$$

O tempo de realização do último nó da rede é assim o valor da última posição do vector  $TNode$ , ou seja,  $TNode(Net(2, n))$ , onde  $Net(2, n)$  contém o número do último nó da rede.

As restantes funções do programa mantiveram basicamente as suas instruções originais.

O programa, tal como ele é apresentado no apêndice J, está apto a otimizar a rede exemplo (figura 3.2).

A adaptação deste programa à utilização para as outras redes é feita alterando alguns dos parâmetros representativos de cada problema em particular, no programa principal (*'GOfunc.m'*). Só para exemplificar, para a rede 1 (figura 5.2), o programa principal ficaria como se segue:

#### **Gofunc.m**

---

```

btime = clock; % to measure time
global lbda T cl W GLOBALCOUNT Net n
% Net - Each column is an activity
Net =[1 2 1;... % Origin node
      2 3 3]; % Target node
lbda=[0.2,0.1,0.07];
T=14;
cl=2;
n=3;
Params = struct('Optimum', 0.0, ...
               'FunctionName', strvcats('totalcost'), ...
               'OutputFileName', 'func.out', ...
               'UpperBounds', [1.5 1.5 1.5], ...
               'LowerBounds', [0.5 0.5 0.5], ...
               'OptimumCoords', [1 1 1], ...
               'NumOfIterations', 75, ...
               'NumOfRuns', 100, ...

```

```
'n', 3, ...  
'm', 15, ...  
'LocalIterations', 1, ...  
'delta', 0.05, ...  
'pertpar', 0.25, ...  
'graph', 0);  
[best,Results] = EMVIS(Params);  
save Workspace  
fprintf('\n\nElapsed time was %2.2f minutes\n',etime(clock,btime)/60);  
beep
```

---

Os restantes procedimentos mantêm-se inalterados.



# Capítulo 7

## Conclusões e Trabalho Futuro

### 7.1 Conclusões

A gestão de projectos é hoje, sem dúvida, um instrumento essencial que permite às organizações enfrentar melhor os desafios do mundo actual. Nesta tese foi abordada esta temática, lançando pistas para uma gestão mais eficiente, em particular no que se refere à questão da alocação de recursos escassos a projectos.

No primeiro capítulo é possível encontrar várias definições do termo projecto, bem como a sua organização por fases durante o seu ciclo de vida. Perceber estas fases pode ajudar o gestor a melhor organizar o seu trabalho.

Nesta tese é apresentado o objectivo de cada fase, dando-se especial atenção à fase de planeamento, na qual se enquadra o presente trabalho de investigação. A fase de planeamento engloba a estruturação do projecto, ou seja, o desenvolvimento da ‘Work Breakdown Structure’, onde são identificadas as actividades principais e as sub-actividades associadas. É depois feito o escalonamento do projecto, que consiste resumidamente em encadear as actividades no tempo, tendo em atenção as precedências existentes. Este encadeamento pode ser representado por uma rede de actividades. Depois de feita esta representação, pode ser utilizado o método CPM para calcular o tempo de conclusão do projecto, utilizando

informação sobre a duração das actividades e sobre as precedências.

Contudo, pode surgir aqui um problema: a existência de restrições de recursos. Este tipo de restrições pode inviabilizar a execução de um escalonamento definido pelo método CPM. Nesta tese são apontadas várias soluções para este problema. As mais simples, são do tipo, deslocar as actividades dentro das suas folgas ou aplicar mais recursos nos momentos críticos. Quando este tipo de soluções não é aplicável, então o problema passa a ser o de desenvolver um escalonamento que respeite as restrições de precedências e de recursos, de tal forma que o projecto termine o mais cedo possível. Este problema, conhecido na literatura por ‘Resource Constrained Project Scheduling Problem’, foi amplamente estudado durante esta investigação.

Existem muitas abordagens desenvolvidas para tratar este problema, principalmente para o caso determinístico. Algumas dessas abordagens foram apresentadas, nomeadamente abordagens que utilizam optimização matemática, métodos de determinação de limites inferiores para a data de conclusão do projecto, procedimentos heurísticos e procedimentos exactos, onde estão incluídos os procedimentos de programação dinâmica, estudados neste trabalho.

Quando passamos ao problema estocástico, o número de contribuições na literatura reduz-se drasticamente. Este tipo de problemas surge quando a duração das actividades é representada por uma variável aleatória.

A maioria das referências encontradas visa essencialmente o cálculo do índice de criticalidade das actividades e dos caminhos de uma rede estocástica, a determinação da função de distribuição da duração do projecto, bem como a média e o desvio padrão.

No caso do RCPSP estocástico, foram encontradas apenas três referências ([IgelmundRadermacher83], [Golenko-GinzGonik97] e [Radermacher81]), mas nenhuma trata o caso multimodal. É precisamente neste caso que se insere a investigação desenvolvida.

Sucintamente, o problema estudado pode ser definido do seguinte modo: dada

uma rede de actividades representativa de um projecto, pretende-se determinar a alocação óptima de recursos, para minimizar o custo total, em presença de incerteza. A forma como esta incerteza é definida distingue este trabalho de contribuições anteriores para a teoria das redes de actividades. A incerteza em vez de estar associada à duração das actividades, está associada ao seu conteúdo de trabalho, o que faz muito mais sentido, na nossa opinião. Enquanto a duração de uma actividade é uma variável dependente da quantidade de recursos a ela atribuída, o conteúdo de trabalho de uma actividade é independente dos recursos utilizados. O que se pretende é precisamente decidir que quantidade de recursos utilizar, tornando a duração maior ou menor.

O problema estudado associa então a incerteza ao conteúdo de trabalho das actividades ( $W_a$ ), que é representado por uma variável aleatória com uma determinada distribuição de probabilidades. Considera-se que existe uma quantidade de recurso limitada que se pode atribuir a cada actividade ( $x_a$ ). Quanto maior for a quantidade de recurso atribuída, menor será a duração da actividade, e maior será o custo do recurso. Existe uma data de conclusão especificada para o projecto ( $T$ ). Assim, se o projecto se atrasar, haverá um custo associado a esse atraso. O objectivo é então determinar a alocação óptima, que minimize o custo total do projecto, ou seja, o custo de aplicar os recursos seleccionados e o custo do atraso que daí resulta.

Enquanto que por um lado, a utilização de maior quantidade de recursos diminui a duração das actividades, a duração do projecto e consequentemente o custo por atrasos, por outro lado aumenta o custo dos recursos. Haverá um ponto de equilíbrio, que resultará num custo total menor. Esse é o ponto que tentamos encontrar.

O método utilizado para resolver este problema foi a programação dinâmica, que nos permitiu desenvolver um modelo simples e intuitivo. Este método foi concebido para ser aplicado a processos de decisão multi-estágio [Bellman57]. O nosso problema adaptou-se facilmente a uma estrutura deste tipo, tendo sido

apontadas formas para definir os estágios de um projecto, bem como as funções de contribuição de estágio resultantes, nomeadamente através da apresentação de um exemplo.

A solução que se obtém do modelo é uma política que indica a alocação de recursos “óptima”<sup>1</sup>, para todos os possíveis estados do projecto, à medida que ele avança no tempo. Este tipo de solução tem a vantagem de ser por natureza adaptativa, ou seja, se as condições se alterarem, as soluções parciais vão ser diferentes. O grande problema associado à PD é o seu peso computacional, para este tipo de problemas, o que se pode confirmar pelos tempos necessários à obtenção de soluções, registados durante os testes do modelo.

Para superar estes problemas, foram desenvolvidas duas aproximações a este modelo, que utilizam ainda PD, e uma aproximação que utiliza programação não linear. Estas aproximações baseiam-se na utilização de valores médios para representar algumas variáveis aleatórias. São realmente melhorados os tempos, como iremos ver de seguida.

Por último, foi testada a aplicação de uma nova técnica de optimização global, o algoritmo “Electromagnético” [BirbilFang00]. Este algoritmo foi desenvolvido para otimizar funções com uma ou mais variáveis, e foi adaptado por nós para ser aplicado ao problema em análise. Os resultados obtidos foram bastante satisfatórios, tendo sido possível reduzir, para as redes maiores, o tempo necessário à obtenção da solução, por comparação com o modelo de PD inicial.

Nas tabelas 7.1 a 7.6 podem ver-se os vários resultados obtidos, nas diferentes experiências realizadas, para o projecto exemplo, introduzido na secção 3.3 e para as redes 1 a 5, introduzidas na secção 4.2.1. É também apresentado um quadro resumo (ver tabela 7.7), que permite conjugar os principais resultados obtidos, para todas as redes testadas.

Destes resultados podemos retirar algumas conclusões finais. Em termos de

---

<sup>1</sup>A palavra “óptima” está entre aspas, pois a aplicação deste modelo requiere a discretização de algumas variáveis, pelo que não temos toda a gama de valores coberta.

Tabela 7.1: Comparação dos resultados obtidos (projecto exemplo)

Proj. Exemplo	Modelo PD	PD Aprox. 1	PD Aprox.2	PNL	EM
$x_1$	1.25	1.25	1.0	0.911	0.94
$x_2$	0.5	0.5	0.5	0.5	0.6
$x_3$	1.5	1.0	1.0	1.1202	0.96
$x_4$				0.9007	0.83
$x_5$	0.5	0.5	0.5	0.5	0.5
$x_6$	1.0	0.5	0.5	0.548	0.64
$x_7$				1.0389	0.91
$x_8$	1.0	1.0	1.0	0.7107	0.74
$x_9$	1.5	1.0	1.0	0.8805	0.81
$x_{10}$	1.0	0.5	0.5	0.5493	0.74
$x_{11}$				0.759	0.95
<i>EV(Cost)</i>	280.85	194.75	168.94	161.509	263.75
<i>Run Time</i>	141 min.	25 min.	4 min.	1 seg.	45 min.

Tabela 7.2: Comparação dos resultados obtidos (rede 1)

Rede 1	Modelo PD	PD Aprox. 1	PD Aprox. 2	PNL	EM
$x_1$	1.0	1.25	1.25	1.03	0.8
$x_2$				1.03	0.8
$x_3$	1.0	1.0	1.0	0.98	0.83
<i>EV(Cost)</i>	45.53	37.37	31.11	30.59	44.99
<i>Run Time</i>	4.8 seg.	4.6 seg.	4.2 seg.	1 seg.	8 min.

tempo de computação, o modelo de PD inicial é sem dúvida o mais oneroso, quando aplicado às redes de maior dimensão (projecto exemplo, rede 4 e rede 5). O número de combinações que é necessário gerar é muito grande. Este é contudo um modelo optimizante, que num ambiente de computação ideal, poderia chegar ao verdadeiro óptimo.

Tabela 7.3: Comparação dos resultados obtidos (rede 2)

<b>Rede 2</b>	<b>Modelo PD</b>	<b>PD Aprox. 1</b>	<b>PD Aprox. 2</b>	<b>PNL</b>	<b>EM</b>
$x_1$	1.0	1.0	1.25	1.03	0.99
$x_2$	1.0	1.0	0.5	0.51	0.59
$x_3$				0.87	0.82
$x_4$	1.5	1.0	0.5	0.59	0.69
$x_5$				0.96	0.91
$EV(Cost)$	304.62	279.14	148.79	152.82	277.77
$Run Time$	9.6 seg.	8.4 seg.	6 seg.	1 seg.	15 min.

Tabela 7.4: Comparação dos resultados obtidos (rede 3)

<b>Rede 3</b>	<b>Modelo PD</b>	<b>PD Aprox. 1</b>	<b>PD Aprox. 2</b>	<b>PNL</b>	<b>EM</b>
$x_1$	1.25	1.25	0.75	1.15	1.11
$x_2$	1.0	1.0	1.0	0.8	0.8
$x_3$	0.5	0.5	0.5	0.5	0.5
$x_4$				0.95	0.89
$x_5$				1.16	0.87
$x_6$	1.0	0.5	1.0	0.82	0.8
$x_7$	0.5	0.5	0.5	0.5	0.55
$x_8$				0.72	0.83
$x_9$	1.0	0.5	1.0	0.9	0.77
$x_{10}$				0.82	1.04
$x_{11}$	1.0	0.5	0.5	0.5	0.6
$EV(Cost)$	106.76	75.04	58.94	58.75	94.69
$Run Time$	30 min.	13 min.	6 min.	1 seg.	45 min.

Como a gama de valores utilizados para as diferentes variáveis é limitada, não podemos garantir que o valor do custo total esperado fornecido pelo modelo seja um óptimo, mas é sem dúvida um limite superior para o óptimo.

Tabela 7.5: Comparação dos resultados obtidos (rede 4)

Rede 4	Modelo PD	PD Aprox. 1	PD Aprox. 2	PNL	EM
$x_1$	1.25	1.25	0.75	0.93	0.83
$x_2$	1.0	0.5	1.0	0.81	0.95
$x_3$	1.0	1.0	0.5	0.55	0.66
$x_4$	0.5	0.5	0.5	0.5	0.57
$x_5$				0.93	0.74
$x_6$	1.0	0.5	0.5	0.5	0.61
$x_7$	1.0	0.5	1.0	0.82	0.8
$x_8$	0.5	0.5	0.5	0.5	0.65
$x_9$				0.97	0.88
$x_{10}$	1.0	0.5	1.0	0.82	0.87
$x_{11}$				0.97	0.88
$x_{12}$	1.5	1.0	1.0	1.15	1.05
$EV(Cost)$	182.91	117.92	103.26	91.53	146.29
$Run Time$	23 h.	7 h.	3 h.	1 seg.	52 min.

A aplicação do algoritmo EM, permitindo por um lado reduzir o tempo de computação, para as redes maiores, parece-nos também gerar um valor esperado do custo mais próximo do verdadeiro óptimo. Isto porque a aplicação do algoritmo não se baseia na discretização de variáveis, mas sim na optimização de um conjunto aleatório de pontos, que seguem a distribuição especificada. O erro aqui verificado, tem a ver com a impossibilidade de gerar todas as situações possíveis, e poderia ser diminuído no caso de gerarmos um maior número de valores. Este também é um modelo teoricamente optimizante.

Continuando a analisar o algoritmo EM, verificamos que o tempo de computação é maior que o do modelo de PD básico, para as redes mais pequenas. Isto deve-se ao facto de mantermos igual, para todas as redes, o número de amostras

Tabela 7.6: Comparação dos resultados obtidos (rede 5)

Rede 5	Modelo PD	PD Aprox. 1	PD Aprox. 2	PNL	EM
$x_1$	0.75	1.25	0.5	0.69	0.73
$x_2$	1.25	1.25	0.75	1.3	0.98
$x_3$	1.25	0.75	0.75	0.5	0.67
$x_4$				0.67	0.71
$x_5$	0.75	0.75	0.75	0.5	0.71
$x_6$	1.25	1.25	0.75	1.29	0.83
$x_7$	0.75	0.75	0.75	0.5	0.56
$x_8$	1.25	0.75	0.75	0.5	0.65
$x_9$				0.66	0.88
$x_{10}$	1.25	0.75	0.75	1.14	0.84
$x_{11}$	1.25	0.75	0.75	0.5	0.55
$x_{12}$	0.75	0.75	0.75	0.59	0.68
$x_{13}$				0.73	0.88
$x_{14}$				1.02	1.1
$x_{15}$	0.75	0.75	0.75	0.58	0.6
$x_{16}$	1.25	0.75	0.75	0.55	0.55
$x_{17}$				1.5	1.14
$x_{18}$	0.75	0.75	0.75	0.56	0.61
$EV(Cost)$	339.07	271.38	210.19	144.07	306.61
$Run Time$	5 dias	9 h.	2 h.	1 seg.	1h.46min.

geradas (100), e o número de pontos da amostra (16). A dimensão do problema é que vai ditar, neste caso, o tempo de computação<sup>2</sup>.

No caso das aproximações 1 e 2, que usam PD, vemos que os tempos de computação são bastante menores do que o tempo do modelo de PD sem

<sup>2</sup>Note que o tempo de computação do projecto exemplo e da rede 3 são iguais, pois tem igual dimensão (11 variáveis).



Tabela 7.7: Quadro resumo dos principais resultados

	Modelo PD	PD Ap. 1	PD Ap. 2	PNL	EM
	( EV Run Time)	( EV Run Time)	( EV Run Time)	( EV Run Time)	( EV Run Time)
( Rede Exemplo ) ( 11 actividades )	( 280.85 ) ( 141 min. )	( 194.75 ) ( 25 min. )	( 168.94 ) ( 4 min. )	( 161.509 ) ( 1 seg. )	( 263.75 ) ( 45 min. )
( Rede 1 ) ( 3 actividades )	( 45.53 ) ( 4.8 seg. )	( 37.37 ) ( 4.6 seg. )	( 31.11 ) ( 4.2 seg. )	( 30.59 ) ( 1 seg. )	( 44.99 ) ( 8 min. )
( Rede 2 ) ( 5 actividades )	( 304.62 ) ( 9.6 seg. )	( 279.14 ) ( 8.4 seg. )	( 148.79 ) ( 6 seg. )	( 152.82 ) ( 6 seg. )	( 277.77 ) ( 15 min. )
( Rede 3 ) ( 11 actividades )	( 106.76 ) ( 30 min. )	( 75.04 ) ( 13 min. )	( 58.94 ) ( 6 min. )	( 58.75 ) ( 1 seg. )	( 94.69 ) ( 45 min. )
( Rede 4 ) ( 12 actividades )	( 182.91 ) ( 23 h. )	( 117.92 ) ( 7 h. )	( 103.26 ) ( 3 h. )	( 91.53 ) ( 1 seg. )	( 146.29 ) ( 52 min. )
( Rede 5 ) ( 18 actividades )	( 339.07 ) ( 5 dias )	( 271.38 ) ( 9 h. )	( 210.19 ) ( 2 h. )	( 144.07 ) ( 1 seg. )	( 306.61 ) ( 106 min. )

aproximações, mas existe deterioração no resultado obtido. Isto deve-se ao facto de estarmos a substituir o valor do conteúdo de trabalho das actividades, pelas suas médias, o que irá gerar uma duração do projecto subestimada, e logo custos menores. Mas como por outro lado, e devido às discretizações efectuadas nos modelos de PD, há uma tendência para obter valores superiores ao óptimo, podemos considerar que estes efeitos se compensam e que estas aproximações dão resultados razoáveis.

Ao usar PNL o efeito da duração subestimada mantém-se, não existindo contudo aqui o erro das discretizações necessárias na PD. Assim, podemos dizer que o valor obtido por PNL acarreta em si, só um tipo de erro<sup>3</sup>: a subestimação da duração do projecto. Será então um limite inferior para o óptimo.

Como estamos a tratar de um problema estocástico, também não esperaríamos obter um resultado óptimo, perfeito para qualquer situação. O facto dos conteúdos de trabalho das actividades serem aleatórios, significa que existe uma tendência para determinados valores, mas qualquer combinação é possível. Sendo assim, o valor óptimo seria o valor que em média iríamos obter, para o custo total, mas não um valor certo. Além disso, o que na prática interessa é o valor

<sup>3</sup>Isto se considerarmos que se conseguiu obter um mínimo absoluto, o que não acontece por exemplo com a rede 2.

da alocação dos recursos, sugerida pelo modelo. Sendo assim, os modelos que usam PD são sem dúvida os mais adequados, pois são adaptativos, indicando os melhores valores para a alocação dos recursos, dependendo das situações que na prática forem surgindo, no caso de um projecto real.

Pensamos que este trabalho constitui um importante contributo para o desenvolvimento do conhecimento nesta área.

- Pela primeira vez, foi abordado o problema da alocação de recursos a redes de actividades multimodais, sob condições estocásticas, tendo sido desenvolvido um modelo teórico, utilizando PD, para resolver este problema.
- Foi feita a implementação informática do modelo de PD para a aplicação a uma rede exemplo (versão 1), tendo sido obtidos e analisados os respectivos resultados.
- Foi feita uma análise de sensibilidade dos resultados obtidos, tendo sido possível melhorá-los ligeiramente.
- Para permitir o leitor perceber o funcionamento do modelo, foi feita uma ilustração de um possível cenário real.
- Como se pretendia possibilitar a utilização do modelo para qualquer rede de actividades, foi desenvolvida e implementada informaticamente uma versão genérica desta aplicação (versão 2), tendo sido testadas mais cinco redes de actividades, que permitiram comparar resultados e retirar conclusões.
- Como se constatou que o modelo de PD era muito exigente em termos computacionais, foram desenvolvidas aproximações ao modelo original, nomeadamente:

- Aproximação 1 O conteúdo de trabalho das variáveis fixas foi aproximado à média, tendo sido utilizada PD.
  - Aproximação 2 Todos os conteúdos de trabalho foram aproximados à média, tendo sido utilizada PD.
  - Aproximação 3 Todos os conteúdos de trabalho foram aproximados à média, tendo sido utilizada PNL.
- Estas aproximações foram programadas e testadas com as versões desenvolvidas, para a rede exemplo inicial e para as cinco redes suplementares. Daí foi possível retirar mais conclusões, relativamente à qualidade relativa dos resultados, a à diminuição de tempos de computação conseguida.
  - Por último foi analisada a possibilidade da aplicação de uma nova técnica de optimização global a este problema, o algoritmo EM, tendo-se conseguido com sucesso adaptar o algoritmo a redes de actividades genéricas. Os resultados foram depois comparados com os resultados obtidos pelas outras técnicas. Para esta aplicação, foi necessário recorrer a simulação de Monte Carlo e ao cálculo CPM.

## **7.2 Trabalho Futuro**

Um problema que acompanhou este trabalho foi a impossibilidade de obter um valor óptimo que servisse de comparação aos valores obtidos nas várias aproximações desenvolvidas. Esse problema só poderia ser ultrapassado se fosse possível utilizar PD com distribuições contínuas. Mas como isso ainda não é possível informaticamente, as ideias para trabalho futuro prendem-se essencialmente com a tentativa de obter melhores aproximações ao modelo de PD apresentado.

### **Aproximação futura 1**

Uma possibilidade em aberto é seguir uma abordagem completamente diferente e tentar agregar actividades em série e em paralelo, reduzindo assim o número

de variáveis de decisão e logo os tempos de computação. Para tal, será necessário, em primeiro lugar, transformar a rede em análise numa rede série/paralelo. Vejamos como ficaria depois dessa transformação o projecto exemplo introduzido no capítulo 3 (ver figuras 7.1 e 7.2).

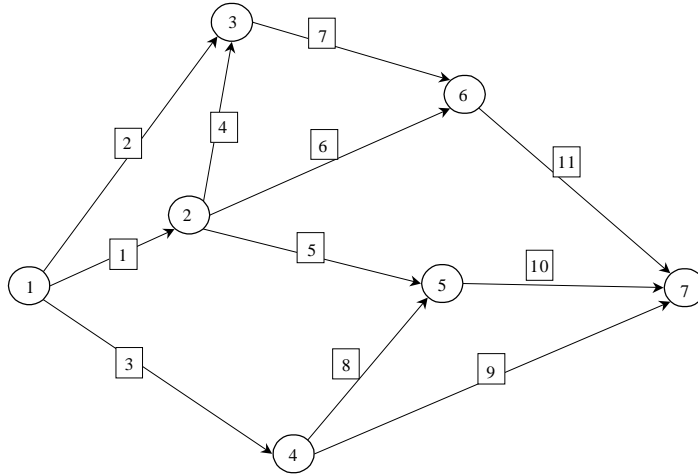


Figura 7.1: Rede original

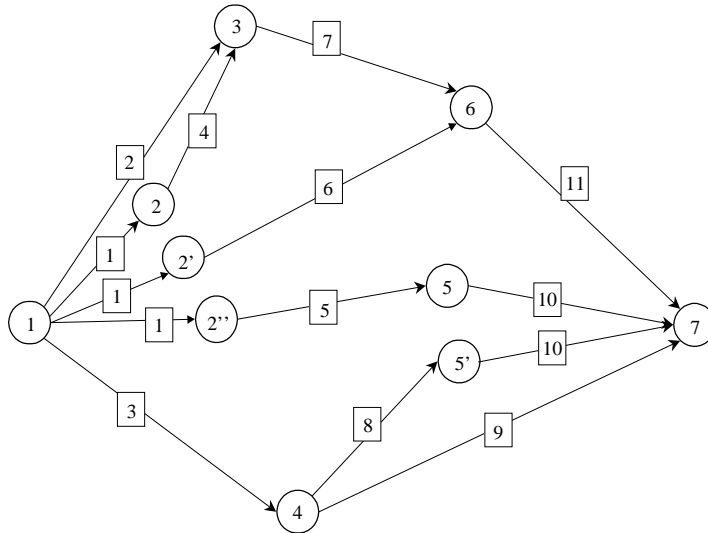


Figura 7.2: Rede série/paralelo

Depois de termos uma rede série/paralelo, ou seja, uma rede em que todas

as actividades estão em série ou em paralelo entre si, podemos começar a construir blocos de actividades. Assim, por exemplo, as actividades 1, 2 e 4 podem ser agrupadas num bloco que terá um conteúdo de trabalho igual à soma dos conteúdos de trabalho das actividades constituintes<sup>4</sup>,

$$W(1, 2, 4) = W_1 + W_2 + W_4. \quad (7.1)$$

As actividades 5 e 10 também podem ser agrupadas num bloco,

$$W(5, 10) = W_5 + W_{10}, \quad (7.2)$$

e as actividades 8, 9 e 10 noutro bloco,

$$W(8, 9, 10) = W_8 + W_9 + W_{10}. \quad (7.3)$$

A rede reduzida ficaria então, como se pode ver na figura 7.3.

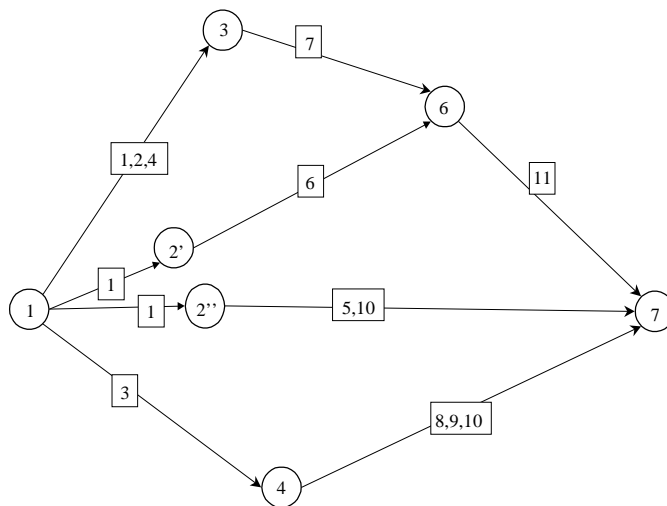


Figura 7.3: Rede reduzida

<sup>4</sup>Atenção: esta é uma soma entre variáveis aleatórias.

De notar que as actividades 1 e 10 seriam comuns a mais do que um caminho, depois desta simplificação. Poder-se-ia então optar por fixar a alocação destas variáveis em  $\hat{x}_1$  e  $\hat{x}_{10}$ . Aplicaríamos assim PD para resolver este problema simplificado. O resultado iria ser obtido mais rapidamente, pois o número de actividades foi reduzido, mas o preço por essa redução seria uma perda de flexibilidade na alocação de recursos às actividades individuais dos blocos.

### Aproximação futura 2

Uma outra ideia de aproximação seria gerar aleatoriamente o conteúdo de trabalho das várias actividades, obtendo-se o vector  $\{w_1, \dots, w_A\}$ , sendo  $A$  o número de actividades. Para o vector gerado usar PNL para obter o vector de alocações  $\{x_1, \dots, x_A\}$ . Repetir a amostra  $K$  vezes (um valor inicial para  $K$  poderia ser 50), e armazenar os vectores de alocação  $\{x_1^k, \dots, x_A^k\}$ , com  $k = 1, \dots, K$ .

Depois desta série de amostragens, poder-se-ia analisar a variância dos vários componentes destes vectores. Vamos supor que para um subconjunto de actividades ( $A1$ ), a variância existente, para os valores de  $x$ , é pequena. Então podemos fixar a alocação dessas actividades, pertencentes ao subconjunto  $A1$ , no seu valor médio. Ficaríamos assim com um número menor de actividades ( $A - A1$ ), cuja alocação não teria sido determinada. O passo seguinte seria repetir novamente a amostragem do conteúdo de trabalho  $K$  vezes, e usar PNL para obter os vectores de alocação para as actividades  $A - A1$ . Se para estas actividades se verificar novamente a existência de uma pequena variância nos valores de  $x$ , fixar então as respectivas alocações nos seus valores médios e repetir a experiência.

Este ciclo de experiências pode parar a qualquer momento em que já hajam poucas variáveis a fixar, optando-se nesse caso, por atribuir também a estas últimas o seu valor médio.

Depois destas experiências efectuadas, o seu resultado deveria ser analisado do ponto de vista de tempo de computação e da qualidade dos valores obtidos,

por comparação com as experiências já desenvolvidas durante a investigação. Daí seria possível retirar mais conclusões, certamente úteis para a resolução de problemas deste tipo.

Um outro tipo de investigação, com interesse nesta área, seria modificar ligeiramente o problema, aumentando o seu grau de dificuldade, como por exemplo, considerando:

- uma disponibilidade total de recursos limitada,
- uma disponibilidade total de recursos limitada e aleatória,
- a existência de mais do que um recurso, associado a cada actividade.

Estas alterações iriam tornar o problema bastante mais complexo, o que iria tornar a investigação mais aliciante. Mas não é este o verdadeiro espírito da investigação?





# Bibliografía

- [AartsKorst89] Aarts, E. and Korst, J. (1989), “Simulated Annealing and Boltzmann Machines”, Wiley, Chichester.
- [AdlakaKulkarni89] Adlaka, V.G. and Kulkarni, V.G. (1989), “A Classified Bibliography of Research on Stochastic PERT Networks: 1966-1987”, *INFOR* vol 27, n<sup>o</sup>3.
- [AlvarezTamarit89] Alvarez-Valdés, R. and Tamarit, J.M. (1989), “Algoritmos Heurísticos Deterministas y Aleatorios en Secuenciación de Proyectos con Recursos limitados”, *Questiío* 13, 173-191.
- [AlvarezTamarit93] Alvarez-Valdés, R. and Tamarit, J.M. (1993), “The Project Scheduling Polyhedron: dimension, facets and lifting theorems”, *European Journal of Operational Research* 67, 204-220.
- [AngusGunderson97] Angus, R.B. and Gunderson, N.A. (1997), “Planning, Performing, and Controlling Projects: principles and applications”, Prentice Hall, London.
- [AshokRajendran89] Ashok Kumar, V.K. and Rajendran, C. (1989), “Manpower Resource Leveling in the Maintenance of a Drillship: a computer simulation approach”, *International Journal of Modelling & Simulation* 13, 152-155.

- [Balas70] Balas, E. (1970), "Project Scheduling with Resource Constraints", Applications of Mathematical Programming Techniques, Carnegie-Mellon University, Pittsburgh.
- [Bedworth73] Bedworth, D.D. (1973), "Industrial Systems: planning, analysis, control", Ronald Press, New York.
- [BedworthBailey87] Bedworth, D.D. and Bailey, J.E. (1987), "Integrated Production Control Systems - management, analysis, design 2/E", John Wiley & Sons, Inc., New York.
- [Bellman57] Bellman, R.E. (1957), "Dynamic Programming", Princeton University Press, New Jersey.
- [BellmanDreyfus62] Bellman, R.E. and Dreyfus, S.E. (1962), "Applied Dynamic Programming", Princeton University Press, New Jersey.
- [Bergeret.al92] Berger, I.; Bourjolly, J.-M.; Laporte, G. (1992), "Branch-and-Bound Algorithms for the Multi-Product Assembly Line Balancing Problem", European Journal of Operational Research 58, 215-222.
- [Biancoet.al99] Bianco, L.; Caramia, M.; Dell'Olmo, P. (1999), "Solving a Preemptive Scheduling Problem with Coloring Techniques" in Weglarz, J. (Ed.), "Handbook on Recent Advances in Project Scheduling", Kluwer, Dordrecht, 1-27.
- [BirbilFang00] Birbil, S.I. and Fang, S-C. (2000), "A New Heuristic for Global Optimization", Proc. of International Work-Shop on Intelligent Systems Resolutions, 8th Bellman Continuum, pp. 352-357, Hsinchu, Taiwan, December 11-12.

- [Birbil02] Birbil, S.I. (2002), “Stochastic Global Optimization Techniques”, Ph.D. Thesis, North Carolina State University - USA.
- [BirbilFang02] Birbil, S.I. and Fang, S-C. (2002), “An Electromagnetism-like Mechanism for Global Optimization”, accepted for publication in Journal of Global Optimization.
- [BirbilFangSheu02] Birbil, S.I.; Fang, S-C.; Sheu R.L.(2002), “On the Convergence of a Population Based Global Optimization Algorithm”, accepted for publication in Journal of Global Optimization.
- [BirbilFang02b] Birbil, S.I. and Fang, S-C. (2002), “A Multi-point Stochastic Search Method for Global Optimization”, accepted for publication in Proc. of the 4th International Symposium on Operations Research and Its Applications, Yichang-Chongqing, China, June 1-4.
- [Boctor96] Boctor, F.F. (1996), “Resource-Constrained Project scheduling by Simulated Annealing”, International Journal of Production Research 34, 2335-2351.
- [Boehm81] Boehm, B.W. (1981), “Software Engineering Economics”, Prentice-Hall, Englewood Cliffs.
- [BouleimenLecocq98] Bouleimen, K. and Lecocq, H. (1998), “A New Efficient Simulated Annealing Algorithm for the Resource-Constrained Project Scheduling Problem”, Working Paper, University of Liège.

- [Bowman94] Bowman, R.A. (1994). "Stochastic Gradient-Based Time-Cost Tradeoffs in PERT Networks Using Simulation", *Annals of Operations Research* 53, 533-551.
- [Burke92] Burke, R. (1992), "Project Management: planning and control", 2nd ed., Wiley, Chichester.
- [Brand*et.al*64] Brand, J.D.; Meyer, W.L.; Shaffer,, L.R. (1964), "The Resource Scheduling Method for Construction", *Civil engineering Studies Report N°5*, University of Illinois.
- [BrooksWhite65] Brooks, G.H. and White, C.R. (1965), "An Algorithm for Finding Optimal or Near Optimal Solutions to the Production Scheduling Problem", *Journal of Industrial Engineering*, January-February Issue, 34-40.
- [Brucker*et.al*96] Brucker, P.; Schoo, A.; Thiele O. (1996), "A Branch & Bound Algorithm for the Resource-Constrained Project Scheduling Problem", Working Paper, Department of Mathematics, University of Osnabrück.
- [CookJennings79] Cook, T.M. and Jennings, R.H. (1979), "Estimating a Project's Completion Time Distribution Using Intelligente Simulation Methods", *Journal of the Operational Research Society* 30, 1103-1108.
- [ChoKim97] Cho, J.-H. and Kim, Y.-D. (1997), "A Simulated Annealing Algorithm for Resource Constrained Project Scheduling Problem", *Journal of the Operational Research Society* 48, 736-744.
- [Cowan68] Cowan, E.W. (1968), "Basic Electromagnetism", Academic Press, New York.

- [Cleland99] Cleland, D.I. (1999), "Project Management: strategic design and implementation". McGraw-Hill. New York.
- [ClelandKing83] Cleland, D.I. and King, W.R. (1983), "Project Management Handbook", Van Nostrand, New York.
- [Christofides *et al*87] Christofides, N.; Alvarez-Valdés, R.; Tamarit, J.M. (1987), "Project Scheduling with Resource-Constraints: a branch-and-bound approach", *European Journal of Operational Research* 29, 262-273.
- [DavisHeidorn71] Davis, E.W. and Heidorn, G.E. (1971), "An Algorithm for Optimal Project Scheduling under Multiple Resource Constraints", *Management Science*, vol. 17, n°12, B803-B816.
- [Davis73] Davis. E.W.(1973), "Project Scheduling under Resource Constraints: historical review and categorization of procedures", *AIIE Transactions*, vol. 5, n°4, 297-313.
- [DeckroHebert89] Deckro, R.F. and Hebert, J.E. (1989), "Resource Constrained Project Crashing", *Omega* 17, 69-79.
- [Dell'AmicoTrubian93] Dell'Amico, M. and Trubian, M. (1993), "Applying Tabu-Search to the Job-Shop Scheduling problem", *Annals of Operations Research* 41, 231-252.
- [Demeulemeester92] Demeulemeester, E.L. (1992), "Optimal Algorithms for Various Classes of Multiple Resource-Constrained Project Scheduling Problems", Unpublished Ph.D Thesis, Department of Applied Economics, Katholieke Universiteit Leuven.
- [DemeulemeesterHerroelen96] Demeulemeester, E.L. and Herroelen, W.S. (1996), "An Efficient Optimal Solution Procedure for the Preemp-

- tive Resource-Constrained Project Scheduling Problem”, European Journal of Operational Research 90, 334-248
- [DemeulemeesterHerroelen97] Demeulemeester, E.L. and Herroelen, W.S. (1997), “New Benchmark Results for the Resource-Constrained Project Scheduling Problem”, Management science 43, 1485-1492.
- [Devroye79] Devroye, L.P. (1979), “Inequalities for the Completion Times of Stochastic PERT Networks”, Math of Operations Research 4, 441-447.
- [De\_ReyckHerroelen98] De Reyck, B. and Herroelen, W.S. (1998), “A Branch-and-Bound Procedure for the Resource-Constrained Project Scheduling Problem with Generalized Precedence Relations”, European Journal of Operational Research 111, 12-174.
- [Dodin85] Dodin, B.M. (1985), “Bounding the Project Completion Time Distribution in PERT Networks”, Operations Research 33, 862-881.
- [Dorndorf*et.al*99] Dorndorf, U.; Phan Huy, T. and Pesch, E. (1999), “A Survey of Interval Capacity Consistency Tests for Time- and Resource-Constrained Scheduling”, in: Weglarz, J. (Ed.), “Handbook on Recent Advances in Project Scheduling”, Kluwer, Dordrecht, 213-238.
- [Dowland93] Dowland, K. (1993), “Simulated Annealing”, in: Reeves, C. (ED.), “Modern Heuristic Techniques for Combinatorial Problems”, Blakwell, Oxford, 20-69.

- [Dowland96] Dowland, K. (1996), "Genetic Algorithms - a tool for OR", *Journal of the Operational Research Society* 47, 550-561.
- [DrexlKimms98] Drexl, A. and Kimms, A. (1998), "Optimization Guided Lower and Upper Bounds for the Resource Investment Problem", *Manuskripte aus den Instituten für Betriebswirtschaftslehre* 481, University of Kiel.
- [Easa89] Easa, S.M. (1989), "Resource Leveling in Construction by Optimization", *Journal of Construction Engineering and Management* 115, 302-316.
- [Elmaghraby93] Elmaghraby, S.E. (1993), "Resource Allocation via Dynamic Programming in Activity Networks", *European Journal of Operational Research* 64, 199-215.
- [Elmaghraby *et. al*98] Elmaghraby, S.E.; Fathi, Y.; Taner, M.R. (1998), "On The Sensitivity of Project Variability to Activity Mean Duration", *International Journal of Production Economics* 62, 219-232.
- [Elmaghraby77] Elmaghraby, S.E. (1977), "Activity Networks - project planning and control by network models", John Wiley and Sons, Inc., New York.
- [Elmaghraby00] Elmaghraby, S.E. (2000), "Optimal Resource Allocation and Budget Estimation in Multimodal Activity Networks", Private Communication, North Carolina State University, Reileigh - North Carolina -USA.
- [Fisher73] Fisher, M.L. (1973), "Optimal Solution of Scheduling Problem Using Lagrange Multipliers: Part I", *Operations Research* 21, 1114-1127.

- [Fisher81] Fisher, M.L. (1981), "The Lagrangean Relaxation Method for Solving Integer Programming Problems", *Management Science* 27, 1-18.
- [FisherGoldstein83] Fisher, D.L. and Goldstein, W.M. (1983), "Stochastic PERT Networks as Models of Cognition: derivation of the mean, variance and distribution of the reaction time using order-of-processing (OP) diagrams", *Journal of Mathematical Psychology* 27, 121-151.
- [Fisher*et.al*85] Fisher, D.L.; Saisi, D.; Goldstein, W.M. (1985), "Stochastic PERT Networks: OP diagrams, critical paths and the project completion time", *Computers and Operations Research* 12, 471-482.
- [FoldesSoumis93] Foldes, S. and Soumis, F. (1993), "PERT and Crashing Revisited: mathematical generalizations", *European Journal of Operational Research* 64, 286-294.
- [Fulkerson62] Fulkerson, D.R. (1962), "Expected Critical Path Lengths in PERT Networks", *Operations Research*, 10, 808-817.
- [GeoffrionMersten72] Geoffrion, A.M. and Mersten, R.E. (1972), "Integer Programming Algorithms: a framework and state-of-the-art survey", *Management Science* 18, 465-491.
- [Geoffrion74] Geoffrion, A.M. (1974), "Lagrangean Relaxation for Integer Programming", *Mathematical Programming Study* 2, 82-114.
- [Glover89] Glover, F. (1989), "Tabu Search - Part I", *ORSA Journal on Computing* 1, 190-206.



- [Glover90] Glover, F. (1990), "Tabu Search - Part II", *ORSA Journal on Computing* 2, 4-32.
- [Goldberg89] Goldberg, D.E. (1989), "Genetic Algorithms in Search, Optimization, and Machine Learning", Addison Wesley, Reading.
- [Golenko-GinzGonik97] Golenko-Ginz, B. and Gonik, A. (1997), "Stochastic Network Project Scheduling with Non-Consumable Limited Resources", *International Journal of Production Economics* 48, 29-37.
- [Gorenstein72] Gorenstein, A.L. (1972), "An Algorithm for Project Sequencing with Resource Constraints", *Operations Research*, vol. 20, n°4, 835-850.
- [Harhalakis89] Harhalakis, G. (1989), "Evaluation of Resource Allocations in Project Scheduling", in: Slowinski, R. and Weglarz, J. (Eds), "Advances in Project Scheduling", Elsevier, Amsterdam, 67-86.
- [HartleyWortham66] Hartley, H.O. and Wortham, A.W. (1966), "A statistical theory for PERT critical path analysis", *Management Science* 12, 469-481.
- [Hartmann98] Hartmann, S. (1998), "A Competitive Genetic Algorithm for Resource-Constrained Project Scheduling", *Naval Research Logistics* 45, 733-750.
- [HartmannSprecher96] Hartmann, S. and Sprecher, A. (1996), "A Note on Hierarchical Models for Multi-Project Planning and scheduling", *European Journal of Operational Research* 94, 377-383.

- [HeldKarp62] Held, M. and Karp, R.M. (1962), "A Dynamic Programming Approach to Sequencing Problems", Journal of the Society for Industrial and Applied Mathematics, March.
- [HillierLieberman95] Hillier, F.S. and Lieberman, G.J. (1995), "Introduction to Operations Research", 6th Ed, McGraw-Hill, New York.
- [Holland75] Holland, J.H. (1975), "Adaptation in Natural and Artificial Systems", University of Michigan Press, Ann Arbor.
- [Ibaraki77] Ibaraki, T. (1977), "The Power of Dominance Relations in Branch-and-Bound Algorithms", Journal of the Association for Computing Machinery 24, 264-279.
- [IgelmundRadermacher83] Igelmund, G. and Radermacher, F.J. (1983), "Preselective Strategies for the Optimization of Stochastic Project Networks Under Resource Constraints", Networks 13, 1-28.
- [Johnson88] Johnson, R.V. (1988), "Efficient Modular Implementation of Branch-and-Bound Algorithms", Decision Sciences 19, 17-38.
- [JustMurphy94] Just, M.R. and Murphy J.P. (1994), "The Effect of Resource Constraints on Project Schedules", AACE Transactions, DCL.2.1-DCL.2.6.
- [Kaplam88] Kaplam, L.A. (1988), "Resource-Constrained Project Scheduling with Preemption of Jobs", Unpublished Ph.D. Thesis, University of Michigan.
- [Kelley63] Kelley, J.E. (1963), "The Critical-Path Method: resources planning and scheduling" in Muth, J.F. and Tompson, G.L. (Eds.), "Industrial Scheduling", Prentice Hall, Englewood Cliffs, 347-365.

- [KennedyThrall76] Kennedy, K.W. and Thrall, R.M. (1976), "Planet: a simulation approach to PERT", *Computers and Operations Research* 3, 313-325.
- [Kerzner98] Kerzner, H. (1998), "Project Management: a systems approach to planning, scheduling and controlling", 6th ed, John Wiley & Sons, New York.
- [Kirkpatrick*et.al*83] Kirkpatrick, S.; Gellat, C.D. and Vecchi, M.P. (1983), "Optimization by Simulating Annealing", *Science* 220, 671-680.
- [Klein98] Klein, R. (1998), "Bidirectional Planning - improving priority rule based heuristics for scheduling resource-constrained projects", to appear in: *European Journal of Operational Research*.
- [KleinScholl98] Klein, R. and Scholl, A. (1998), "PROGRESS: Optimally Solving the Generalized Resource-Constrained Project Scheduling Problem", *Schriften zur Quantitativen Betriebswirtschaftslehre* 5/98, Darmstadt University of Technology.
- [KleinScholl99] Klein, R. and Scholl, A. (1999), "Computing Lower Bounds by Destructive Improvement: an application to resource-constrained project scheduling", *European journal of Operational Research* 112, 322-346.
- [Klein00] Klein, R. (2000), "Scheduling of Resource-Constrained Projects", Kluwer Academic Publishers, Boston.
- [Kleindorfer71] Kleindorfer, G.B. (1971), "Bounding Distributions for a Stochastic Acyclic Network", *Operations Research* 19, 1586-1601.

- [Kliengel66] Kliengel, A.R. (1966), "Bias in PERT Completion Times Calculations for a Real Network", *Management Science* 13, 476-489.
- [KolenPesch94] Kolen, A. and Pesch, E. (1994), "Genetic Local Search in Combinatorial Optimization", *Discrete Applied Mathematics* 48, 273-284.
- [Kolisch95] Kolisch, R. (1995), "Project scheduling under Resource Constraints - efficient heuristics for several problem classes", *Physica, Heidelberg*.
- [KulkarniAdlakha86] Kulkarni, V.G. and Adlakha, V.G. (1986), "Markov and Markov-regenerative PERT networks", *Operations Research* 34, 769-781.
- [Labbé*et.al*91] Labbé, M.; Laport, G.; Mercure, H. (1991), "Capacitated Vehicle Routing on Trees", *Operations Research* 39, 616-622.
- [Lang77] Lang, D.W. (1977), "Critical Path Analysis", David McKay, New York.
- [LeeKim96] Lee, J.-K. and Kim, Y.-D. (1996), "Search Heuristics for Resource Constrained Project Scheduling", *Journal of the Operational Research society* 47, 678-689.
- [Lee-KwangFavrel88] Lee-Kwang, H. and Favrel, J. (1988), "The SSD Graph: a tool for project scheduling and visualization", *IEEE Transactions on Engineering Management* 35, 25-30.
- [LeonBalakrishnan95] Leon, V.J. and Balakrishnan, R. (1995), "Strength and Adaptability of Problem-Space Based Neighborhoods for

- Resource-Constrained Scheduling”, *OR Spektrum* 17, 173-182.
- [Lewis95] Lewis, J.P. (1995), “Project Planning, Scheduling & Control: a hands-on guide to bringing projects in on time and budget”. New York: McGraw-Hill.
- [Lewis97] Lewis, J.P. (1997), “Fundamentals of Project Management”, Amacom, New York.
- [LiWillis91] Li, R.K.-Y. and Willis, R.J. (1991), “Alternate Resources in Project Scheduling”, *Computers and Operations Research* 8, 663-668.
- [LiWillis92] Li, R.K.-Y. and Willis, R.J. (1992), “An Interactive Scheduling Technique for Resource-Constrained Project Scheduling”, *European Journal of Operational Research* 56, 370-379..
- [Lindsey72] Lindsey, J.H. (1972), “An Estimate of Expected Critical-Path Length in PERT Networks”, *Operations Research* 20, 800-812.
- [Lofts74] Lofts, N.R. (1974), “Multiple Allocation of Resources in a Network - an optimal scheduling algorithm”, *INFOR*, vol.12, n°1, 25-38.
- [MacleodPeterson96] Macleod, K.R. and Peterson, P.F. (1996), “Estimating the Tradeoff Between Resource Allocation and Probability of On-time Completion in Project Management”, *Project Management Journal*, Issue 3, 26-33.

- [MartelloToth90] Martello, S. and Toth, P. (1990), "Lower Bounds and Reduction Procedures for the Bin Packing Problem", *Discrete Applied Mathematics* 28, 59-70.
- [Martin65] Martin, J.J. (1965), "Distribution of the Time Through a Directed Acyclic Network", *Operations Research* 13, 46-66.
- [MathWorks96] The MathWorks, Inc (1996), "Using Matlab Graphics".
- [MathWorks97a] The MathWorks, Inc (1997), "Using Matlab".
- [MathWorks97b] The MathWorks, Inc (1997), "Building GUIs with Matlab".
- [MathWorks98a] The MathWorks, Inc (1998), "Getting Started with Matlab".
- [MathWorks98b] The MathWorks, Inc (1998), "Matlab Notebook User's Guide".
- [MathWorks98c] The MathWorks, Inc (1998), "Application Program Interface Guide".
- [MathWorks99a] The MathWorks, Inc (1999), "Release 11 - new features".
- [MathWorks99b] The MathWorks, Inc (1999), "Optimization Toolbox - user's guide".
- [MeredithMantel95] Meredith, J.R. and Mantel, S.J. (1995), "Project Management: a managerial approach", 3rd ed., Wiley, Chichester.
- [McKnew*et.al*91] McKnew, M.A.; Saydam, C.; Coleman, B.J. (1991), "An Efficient Zero-One Formulation of the Multi-Level Lotsizing Problem", *Decision Sciences* 22, 280-295.
- [Michalewicz94] Michalewicz, Z. (1994), "Genetic Algorithms + Data Structures = Evolution Programs", 2nd Ed., Springer, Berlin.

- [Mingozi*et.al*98] Mingozi, A.; Maniezzo, V.; Ricciardelli, S.; Bianco, L. (1998), "An Exact Algorithm for the Resource-Constrained Project Scheduling Problem Based on a New Mathematical Formulation", *Management Science* 44, 715-729.
- [Moder*et.al*83] Moder, J.J.; Phillips, C.R.; Davis, E.W. (1983), "Project Management with CPM, PERT, and Precedence Diagramming", 3d ed., Van Nostrand, New York.
- [NeumannMorlock93] Neumann, K. and Morlock, M. (1993), "Operations Research", Carl Hanser, München.
- [NeumannZimmermann97] Neumann, K. and Zimmermann, J. (1997), "Resource Levelling for Projects with Schedule-Dependent Time Windows", Technical Report WIOR 508, University of Karlsruhe.
- [NeumannZimmermann98] Neumann, K. and Zimmermann, J. (1998), "Exact and Heuristic Procedures for Net Present Value and Resource Levelling in Project Scheduling", Technical Report WIOR 538, University of Karlsruhe.
- [NeumannZimmermann99] Neumann, K. and Zimmermann, J. (1999), "Methods for Resource-Constrained Project Scheduling with Regular and Non-Regular Objective Functions and Schedule-Dependent Time Windows", in Weglarz, J. (Ed.), "Handbook on Recent Advances in Project Scheduling", Kluwer, Dordrecht, 262-288.
- [Patterson*et.al*89] Patterson, J.H.; Slowinski, F.B.; Talbot, F.B.; Weglarz, J. (1989), "An Algorithm for a General Class of Precedence and Resource Constrained Project Scheduling Problems" in

- Slowinski, R. and Weglarz, J. (Eds), "Advances in Project Scheduling", Elsevier, Amsterdam, 3-29.
- [Pesch99] Pesch, E. (1999), "Lower Bounds in Different Problem Classes of Project Schedules with Resource Constraints" in Weglarz, J. (Ed.), "Handbook on Recent Advances in Project Scheduling", Kluwer, Dordrecht, 53-75.
- [PMI00] PMI (2000). "A guide to the Project Management Body of Knowledge", Project Management Institute Publishing Division, North Carolina-USA.
- [PritskerHapp66] Pritsker, A.A.B and Happ W.W (1966), "GERT: Graphical Evaluation and Review Technique - Part I: fundamentals", *Journal of Industrial Engineering* 17, 267-274.
- [PritskerWhitehorse66] Pritsker, A.A.B and Whitehorse G.E (1966), "GERT: Graphical Evaluation and Review Technique - Part II: probabilistic and industrial engineering applications", *Journal of Industrial Engineering* 17, 293-301.
- [Pritsker *et.al*69] Pritsker, A.A.B.; Watters, L.J.; Wolfe, Ph.M. (1969), "Multiproject Scheduling with Limited Resources: a zero-one programming approach", *Management Science* 16, 93-108.
- [Radermacher81] Radermacher, F.J. (1981), "Cost-Dependent Essential Systems of ES Strategies for Stochastic Scheduling-Problems", *Methods of Operations Research* 42, 17-31.
- [Reeves97] Reeves, C.R. (1997), "Genetic Algorithms for the Operations Researcher", *INFORMS Journal of Computing* 9, 231-250.



- [Ringer69] Ringer, L.J. (1969), "Numerical Operators for Statistical PERT Critical Path Analysis", *Management Science* 16, 136-143.
- [Ringer71] Ringer, L.J. (1971), "A Statistical Theory for PERT in which Completion Times of Activities are Inter-dependent", *Management Science* 17, 717-723.
- [RobillardTrahan76] Robillard, P and Trahan, M. (1976), "Expected Completion Time in PERT Networks", *Operations Research* 24, 177-182.
- [Roucairol96] Roucairol, C. (1996), "Parallel Processing for Difficult Combinatorial Optimization Problems", *European Journal of Operational Research* 92, 573-590.
- [RuskinEstes95] Ruskin, A.M. and Estes, W.E. (1995), "What Every Engineer Should Know About Project Management", Marcel Dekker, Inc, USA.
- [SahnerTrivedi87] Sahner, R.A. and Trivedi, K.S. (1987), "Performance and Reliability Analysis Using Directed Acyclic Graphs", *IEEE Transactions on Software Engineering* 13, 1105-1114.
- [Schollet.al97] Scholl, A.; Klein, R.; Jürgens, Ch. (1997), "Bison: a fast hybrid procedure for solving the one-dimensional bin packing problem", *Computers & Operations Research* 24, 627-645.
- [Scholl99] Scholl, A. (1999), "Balancing and Sequencing of Assembly Lines", 2nd Ed., Physica, Heidelberg.
- [Schrage70] Schrage, L. (1970), "Solving Resource-Constrained Network Problems by Implicit Enumeration: non-

- preemptive case”, *Operations Research*, vol.18, n°2, 225-235.
- [Sculli83] Sculli, D. (1983), “The Completion Time of PERT Networks”, *Journal of the Operational Research Society* 34, 155-158.
- [SeibertEvans91] Seibert, J.E. and Evans, G.W. (1991), “Time-Constrained Resource-Leveling”, *Journal of Construction Engineering* 117, 503-520.
- [Shogan77] Shogan, A.W. (1977), “Bounding Distributions for Stochastic PERT Networks”, *Networks* 7, 359-381.
- [Shtubet.al94] Shtub, A.; Bard, J.F.; Globerson, S. (1994), “Project Management: engineering, technology, and implementation”, Prentice Hall, Englewood Cliffs.
- [Sigal<sup>et.al</sup>79] Sigal, C.E., Pritsker, A.B. and Solberg, J.J. (1979), “The Use of Cutsets in Monte Carlo Analysis of Stochastic Networks”, *Math. Comput. Simulation* 21, 376-384.
- [Spelde77] Spelde, H.G. (1977), “Bounds for the Distribution Function of Network Variables”, *First Symposium of Operations Research (University Heidelberg)* 3, 112-123.
- [SperanzaVercellis93] Speranza, G.M. and Vercellis, C. (1993), “Hierarchical Models for Multi-Project Planning and Scheduling”, *European Journal of Operational Research* 64, 312-325.
- [Spinner97] Spinner, M.P. (1997), “Project Management: principles and practices”. Prentice-Hall, Englewood Cliffs.

- [Sprecher94] Sprecher, A. (1994), "Resource-Constrained Project Scheduling - Exact Methods for the Multi-Mode Case", Springer, Berlin.
- [Sprecher*et.al*97] Sprecher, A.; Hartmann, S.; Drexl, A. (1997), "An Exact Algorithm for Project Scheduling with Multiple Modes", OR Spektrum 19, 195-203.
- [SprecherDrexl98] Sprecher, A.; Drexl, A. (1998), "Multi-Mode Resource-Constrained Project Scheduling by a Simple, General and Powerful Sequencing Algorithm", European Journal of Operational Research 107, 431-450.
- [Stinson*et.al*78] Stinson, J.P.; Davis, E.W.; Khumawala, B.M (1978), "Multiple Resource-Constrained Scheduling Using Branch and Bound", AIIE Transactions 10, 252-259.
- [Talbot78] Talbot, F.B. (1978), "An Efficient Integer Programming Algorithm with Network Cuts for Solving Resource-Constrained Scheduling Problems", Management Science 24, n°11, 1163-1174.
- [Talbot82] Talbot, F.B. (1982), "Resource Constrained Project Scheduling with Time-Resource Tradeoffs: the nonpreemptive case", Management Science 28, 1197-1210.
- [Turner92] Turner, J. R. (1992), "The Handbook of Project Based Management: improving processes for achieving your strategic objectives", McGraw-Hill, New York.
- [UlusoyÖzdamar89] Ulusoy, G. and Özdamar, L. (1989) "Heuristic Performance and Network/Resource Characteristics in Resource-

- Constrained Project Scheduling”, *Journal of the Operational Research Society* 40, 1145-1152.
- [Van\_Slyke63] Van Slyke, R.M. (1963), “Monte Carlo Methods and the PERT Problem”, *Operations Research* 11, 839-860.
- [Willis85] Willis, R.J. (1985), “Critical Path Analysis and Resource-Constrained Project Scheduling: theory and practice”, *European Journal of Operational Research* 23, 149-155.
- [Woodworth93] Woodworth, B.M.(1993), “A Statistical Evaluation of the Impact of Limited Resources on Project Scheduling”, *Cost Engineering* 35, Issue 2, 25-32.

# Apêndice A

## Versão 1

### A.1 mp.m

```
% mp.m (performs a 3 factorial experiment
% for every fixed variable xi (0.5(Lo),1.0(Medium),
% 1.5(Hi))calculating x1star and the expected value of
% cost for the project with 4 states (ev4))
btime = clock; % to measure time
% Global variables: lambda, expW, p, P, cl, T, Wi
% Lambda is the parameter of W
lambda=[0.1,0.12,0.05,0.08,0.2,0.04,0.03,0.04,0.024,0.15,0.16];
expW=1./lambda;
p=[0.25 0.25 0.25 0.25]; % probability of Wi
P=[0.25 0.5 0.75 1]; % cumulative probability of Wi
cl=5; % delay cost per unit time
T=65; % due date
% to enumerate over the possible values of Wi
W1=generateW(lambda(1));
W2=generateW(lambda(2));
```

```

W3=generateW(lambda(3));
W4=generateW(lambda(4));
W5=generateW(lambda(5));
W6=generateW(lambda(6));
W7=generateW(lambda(7));
W8=generateW(lambda(8));
W9=generateW(lambda(9));
W10=generateW(lambda(10));
W11=generateW(lambda(11));
It2min=W1(1)/1.5;
It2max=W1(4)/0.5;
It2=[It2min,It2max];
[stT2,T2]=calculateTvalues(It2);
It3min=It2min+W4(1)/1.5;
It3max=It2max+W4(4)/0.5;
It3=[It3min,It3max];
[stT3,T3]=calculateTvalues(It3);
It4min=W3(1)/1.5;
It4max=W3(4)/0.5;
It4=[It4min,It4max];
[stT4,T4]=calculateTvalues(It4);
It5min=It4min+W8(1)/1.5;
It5max=It4max+W8(4)/0.5;
It5=[It5min,It5max];
[stT5,T5]=calculateTvalues(It5);
It6min=It3min+W7(1)/1.5;
It6max=It3max+W7(4)/0.5;
It6=[It6min,It6max];
[stT6,T6]=calculateTvalues(It6);

```



```

    fprintf('%6.2f %6.2f %6.2f %6.2f %6.2f %6.2f %6.2f %6.2f %8.2f\n',Res(i,:));
end
disp('Best value was:');
disp(' x2 x3 x5 x6 x8 x9 x10 x1star ev4');
fprintf('%6.2f %6.2f %6.2f %6.2f %6.2f %6.2f %6.2f %6.2f %8.2f\n',Res(k,:));
% -----
% Write result to a file (to export to excel)
fid = fopen('resBestAll.txt','w');
fprintf(fid,'\nCurrent parameters');
fprintf(fid,'\nDue date (T) = %3d',T);
fprintf(fid,'\nCost of lateness (cl) = %2d\n\n',cl);
fprintf(fid,'x2,x3,x5,x6,x8,x9,x10,x1star,ev4\n');
for i=1:size(Res,1)
    fprintf(fid,'%6.2f,%6.2f,%6.2f,%6.2f,%6.2f,%6.2f,%6.2f,%6.2f,%8.2f\n',Res(i,:));
end
fprintf(fid,'\nBest value was:\n');
fprintf(fid,'x2,x3,x5,x6,x8,x9,x10,x1star,ev4\n');
fprintf(fid,'%6.2f,%6.2f,%6.2f,%6.2f,%6.2f,%6.2f,%6.2f,%6.2f,%8.2f',Res(k,:));
fprintf(fid,'\n\nElapsed time was %2.2f minutes\n',etime(clock,btime)/60);
fclose(fid);
% -----
% Write Best1 to a file (to export to excel)
fid = fopen('resBest1.txt','w');
fprintf(fid,'t2,t4,t6,x11star,ev1\n');
for i=1:size(Best1,1)
    fprintf(fid,'%6.4f,%6.4f,%6.4f,%6.4f,%8.4f\n',Best1(i,:));
end
fclose(fid);
% -----

```



```

% Write Best2 to a file (to export to excel)
fid = fopen('resBest2.txt','w');
fprintf(fid,'t2,t3,t4,x7star,ev2\n');
for i=1:size(Best2,1)
    fprintf(fid,'%6.4f,%6.4f,%6.4f,%6.4f,%8.4f\n',Best2(i,:));
end
fclose(fid);
% -----
% Write Best3 to a file (to export to excel)
fid = fopen('resBest3.txt','w');
fprintf(fid,'t2,t4,x4star,ev3\n');
for i=1:size(Best3,1)
    fprintf(fid,'%6.4f,%6.4f,%6.4f,%8.4f\n',Best3(i,:));
end
fclose(fid);
% -----
% Write Best4 to a file (to export to excel)
fid = fopen('resBest4.txt','w');
fprintf(fid,'x1star,ev4\n');
for i=1:size(Best4,1)
    fprintf(fid,'%6.4f,%8.4f\n',Best4(i,:));
end
fclose(fid);
% -----
% Write result to a file (to read later into Matlab Workspace)
save resultBestAll T cl Best1 Best2 Best3 Best4 Res
fprintf('\n\nElapsed time was %2.2f minutes\n',etime(clock,btime)/60);
clear W* I* T* P x* st* p i cl m k btime lambda
beep

```

## A.2 mediumprojects1.m

```

% State is s1:(t4,t5,t6)
% Global variables: Best1
% Decision variable is x11
Best1=[ ];
% resource cost for activities whose resource level is fixed
rcf = x2*expW(2) + x3*expW(3) + x5*expW(5) + x6*expW(6) + x8*expW(8)
      + x9*expW(9) + x10*expW(10);
for t4=T4
  for t5=T5
    % Calculation of Y7: Y7=max(t4+W9/x9,t5+W10/x10)
    X1=t4+W9/x9;
    X2=t5+W10/x10;
    Y7=calculateMax(X1,X2);
    PY7=calculatePmax(Y7,X1,P,X2,P);
    % Calculation of T7, U , ...
    for t6=T6
      EV=[ ];
      for x11=0.5:0.25:1.5
        ev=0;
        for i11=1:4
          w11=W11(i11);
          T7=max(Y7,t6+w11/x11); % P=PY7
          U=max(0,T7-T); % P=PY7
          evU=expectedValue(U,PY7);
          TC(i11)=cl*evU; % Tardiness cost
        end % 4 values - Probability=P
        rc11=x11*expW(11);

```

```

        F1=TC+rcf+rc11; % total cost
        ev=expectedValue(F1,P);
        EV=[EV;x11,ev];
    end
m=min(EV(:,2));
k=max(find(EV(:,2)<=m));
x11star=EV(k,1);
v1star=EV(k,2);
    Best1=[Best1;t4,t5,t6,x11star,v1star];
end
end
end
clear X* Y* T7 TC PY7 EV U F1 t* ev* x11star rcf i11 w11 m k v1star
%['t4, t5, t6, x11star, ev1']
%Best1
% f1(t4,t5,t6) = min eV(x11*W11+cl*eV(U))
%           x11
%
% U=max(0,T7-T)
% T7=max(Y7,t6+w11/x11)
% Y7=max(t4+W9/x9,t5+W10/x10)

```

### A.3 mediumprojects2.m

```

% State is s2:(t2,t3,t4)
% decision variable is x7
% Global variables: Best2
Best2=[ ];
for t2=T2

```

```

X1=t2+W6/x6;
X3=t2+W5/x5;
for t3=T3
  for t4=T4
    X4=t4+W8/x8;
    T5n=calculateMax(X3,X4);
    PT5n=calculatePmax(T5n,X3,P,X4,P);
    pT5n=mass(PT5n);
    EV=[ ];
    for x7=0.5:0.25:1.5
      X2=t3+W7/x7;
      T6n=calculateMax(X1,X2);
      PT6n=calculatePmax(T6n,X1,P,X2,P);
      pT6n=mass(PT6n);
      F1=[ ];
      pF1=[ ];
      m=1;
      for i=1:size(T5n,2)
        for j=1:size(T6n,2)
          k=min( find( Best1(:,1)==t4 & abs(Best1(:,2)-T5n(i))<=stT5/2
                    & abs(Best1(:,3)-T6n(j))<=stT6/2 ) );
          F1(m)=Best1(k,5);
          pF1(m)=pT5n(i)*pT6n(j);
          m=m+1;
        end
      end
    end
    PF1=cum(pF1);
    evF1=expectedValue(F1,PF1);
    ev=x7*expW(7)+evF1;

```

```

        EV=[EV;x7,ev];
    end
    m=min(EV(:,2));
    k=max(find(EV(:,2)<=m));
    x7star=EV(k,1);
    v2star=EV(k,2);
    Best2=[Best2;t2,t3,t4,x7star,v2star];
end
end
end
clear X* EV T6n F1 IF2 PT6n PT5n PF1 pT6n pT5n pF1 ev* x7star t*
    k i m v2star
% ['t2, t3, t4, x7star, ev2']
% Best2
% f2(t2,t3,t4) = min eV(x7*W7+eV(f1(t4,T5,T6))
%           x7
%
% T5 = max(t2+W5/x5,t4+W8/x8)
% T6 = max(t2+W6/x6,t3+W7/x7)

```

## A.4 mediumprojects3.m

```

% State is s3:(t2)
% Decision variable is x4
% Global variables: Best3
Best3=[ ];
X2=W2/x2;
T4n=W3/x3;
for t2=T2

```

```

EV=[];
for x4=0.5:0.25:1.5
    X1=t2+W4/x4;
    T3n=calculateMax(X1,X2);
    PT3n=calculatePmax(T3n,X1,P,X2,P);
    pT3n=mass(PT3n);
    F2=[];
    pF2=[];
    m=1;
    for i=1:size(T3n,2)
        for j=1:size(T4n,2)
            k=min( find( Best2(:,1)==t2 & abs(Best2(:,2)-T3n(i))<=stT3/2 &
                abs(Best2(:,3)-T4n(j))<=stT4/2 ) );
            F2(m)=Best2(k,5);
            pF2(m)= pT3n(i)* p(j);
            m=m+1;
        end
    end
    PF2=cum(pF2);
    evF2=expectedValue(F2,PF2);
    ev=x4*expW(4)+evF2;
    EV=[EV;x4,ev];
end
m=min(EV(:,2));
k=max(find(EV(:,2)<=m));
x4star=EV(k,1);
v3star=EV(k,2);
Best3=[Best3;t2,x4star,v3star];
end

```

```

clear X* EV T3n pT3n PT3n sT4 pF2 PF2 F2 ev* x4star t2 i k m v3star
% ['t2, t4, x4star, ev3']
% Best3
% f3(t2) = min eV(x4*W4+ev(f2(t2,T3,T4))
%           x4
% T4 = W3/x3 (from mp.m)
% T3 = max(t2+W4/x4,W2/x2)

```

## A.5 mediumprojects4.m

```

% State is s4:t1=0
% Decision variable is x1
% Global variables: Best4
Best4=[ ];
EV=[ ];
for x1=0.5:0.25:1.5
    T2n=W1/x1;
    for i=1:4
        k=min(find(abs(Best3(:,1)-T2n(i))<=stT2/2));
        F3(i)=Best3(k,3);
    end
    evF3=expectedValue(F3,P);
    ev=x1*expW(1)+evF3;
    EV=[EV;x1,ev];
end
m=min(EV(:,2));
k=max(find(EV(:,2)<=m));
x1star=EV(k,1);
v4star=EV(k,2);

```

```

Best4=[x1star,v4star];
clear F* EV T2n x1star ev* i j k m v4star sT4
% f4(0) = min eV(x1*W1+eV(f3(T2))
%      x1
%
% T2 = W1/X1

```

## A.6 generateW.m

```

function w=generateW(lambda);
% For a given lambda (exponential distribution),
% it generates 4 values of W, each with p=0.25.
t(1)=0;
e(1)=1;
for i=1:3
    t(i+1)=log(e(i)-0.25)/(-lambda);
    e(i+1)=exp(-lambda*t(i+1));
    w(i)=((t(i)+1/lambda)*e(i)-(t(i+1)+1/lambda)*e(i+1))/0.25;
end
w(4)=(t(4)+1/lambda)*e(4)/0.25;

```

## A.7 expectedValue.m

```

function ev=expectedValue(X,P)
p=mass(P);
ev=sum(p.*X);          % ev=p1.X1 + p2.X2 + ...

```



## A.8 calculateTvalues.m

```

function [step,t]=calculateTvalues(it);
% Calculates Time Values within a given interval (it), and
% returns the time values (t) and the step used (step)
% The interval of values is divided by 40, otherwise too many points
n=round((it(2)-it(1))/40); % n-number of values considered.
                                % The greater the interval, the greater n
if n<4                            % (each 40 values - 1 interval)
    n=4;                            % If the interval is small, use 4 points (minimal)
elseif n>15                        % If it is large, use 15 points (maximal)
    n=15;
end
step=(it(2)-it(1))/(n-1);
for i=1:n
    t(i)=it(1)+(i-1)*step;
end

```

## A.9 calculateMax.m

```

function Y=calculateMax(X1,X2);
k=1;
s1=size(X1,2);
s2=size(X2,2);
for i=1:s1
    for j=1:s2
        Ybig(k)=max(X1(i),X2(j)); % the MAX can have repeated values
        k=k+1;
    end
end

```

```

end
Y=reduce(Ybig);      % takes off the repeated values

```

## A.10 calculatePmax.m

```

function P=calculatePmax(Y,X1,P1,X2,P2); % P(Y<=y)=P(X1<=y).P(X2<=y)
i=1;
dim=size(Y,2);
while i<=dim
    y=Y(i);
    c=cumProb(X1,P1,y)*cumProb(X2,P2,y);
    P(i)=c;
    i=i+1;
end

```

## A.11 cumProb.m

```

function cP=cumProb(X,Px,v) % P(W<=n)
k=max(find(X(:)<=(v+0.000000001)));
cP=Px(k);

```

## A.12 cum.m

```

function P=cum(p)
% Transforms probability mass function into cumulative probability
s=size(p,2);
P(1)=p(1);
for i=2:s
    P(i)=P(i-1)+p(i);
end

```

```
end
```

### A.13 reduce.m

```
function Y=reduce(Ybig);           %takes off all repeated elements
Y=[];
while ~isempty(Ybig)
    y=min(Ybig);
    Y=[Y,y];
    k=find(Ybig(:)==y);           % all indexes
    Ybig(k)=[ ];
end
```

### A.14 mass.m

```
function p=mass(P)                % transforms cumulative probability into
                                   probability mass function
s=size(P,2);
p(1)=P(1);
for i=2:s
    p(i)=P(i)-P(i-1);
end
```



# Apêndice B

## Versão 2

### B.1 dp.m

```
clear all
btime = clock;    % to measure time
global n cl T W expW P Net NL NLP ALP Y Tlim AF sc t1 pfa
t1=0;
P=[0.25 0.5 0.75 1];
pfa=3;    % n° of points of fixed activities (2=(0.75 1.25) or 3=(0.5 1 1.5))
inputNetwork;
decisionVars;
generateWs;
generateExpWs;
durations;    % Y (for any activitie, minimal and maximal duration)
generateTlimits; % Tlim (for any node, min. and max. realization times)
generateTvalues; % Tval (structure with two fields. Tval.st (step)
                % Tval.T (List of T values for the nodes)
AF=activitiesToFix(ALP);
AF=xFixedActivities(AF);
```

```

generateMainCode;
generateDps1Code;
for i=2:size(ALP,2)    % call for the states of the dp
    sc=i;            % state code
    generateDpsNCode;
end
rehash;              % to load new file: main.m into memory
main;
fprintf('\n\nElapsed time was %2.2f minutes\n',etime(clock,btime)/60);
beep
clear S btime t1 i sc

```

## B.2 inputNetwork.m

```

% global n cl T Net
Net=[];
lc=input('Do you want [L]oad an existing network or
         [C]reate a new one? : ','s');
if lc=='l' |lc=='L'
    s=input('What is the name of the network to load? : ','s');
    load(s);
elseif lc=='c' |lc=='C'
    n=input('How many activities: ');
    for i=1:n
        fprintf('\n\nActivitie %d',i);
        Net(1,i)=input('\nOrigin node no: ');
        Net(2,i)=input('Terminal node no: ');
        Net(3,i)=input('Lambda: ');
        Net(4,i)=input('Minimal resource level: ');
    end
end

```

```

    Net(5,i)=input('Maximal resource level: ');
end
cl=input('\nDelay cost per unit of time: ');
T=input('\nDue date of project: ');
s=input('\nSave Network As? : ','s');
save(s,'Net','n','cl','T');
end
clear lc s i

```

### B.3 decisionVars.m

```

NL=sumNodesLen;
NLP=nodesLP(NL);
ALP=activitiesLP(NLP);

```

### B.4 sumNodesLen.m

```

function NL=sumNodesLen;
% NL - Nodes Length
global Net n
nn=Net(2,n);
NL(1)=0; % NL1=0
for i=2:nn
    [Orig,]=originNodes(i);
    NL(i)=max(NL(Orig)+1); % NLj=max{NLi+1}; j=2,3,...,n
end % i->j

```

## B.5 originNodes.m

```
function [O,A]=originNodes(t);
% t - target
% for a target, find all origins
% A - activities (origin-target)
global Net
A=find(Net(2,:)==t);
O=Net(1,A);
```

## B.6 nodesLP.m

```
function NLP=nodesLP(NL);
% NLP-Nodes on the Longest Path; NL-Nodes Length
global Net n
i=Net(2,n); % Node n - last node
NLP=i; % Last node is always on the LP
while i>1
    [Orig,]=OriginNodes(i); % If a node as only one origin,
    if size(Orig,2)==1 % the origin is on the LP
        NLP=[Orig,NLP]; % Insert at beginning of NLP
    else
        k=find(NL==(NL(i)-1)); % If a node as more than one origin,
        if size(k,2)==1 % find the one that as a nl=nl(node)-1
            NLP=[k,NLP];
        else
            NLP=[k(1),NLP]; % If there is more than one origin with
        end % same node lenght choose one
    end % (for example the 1st one)
end
```



```

    i=NLP(1);
end

```

## B.7 activitiesLP.m

```

function ALP=activitiesLP(NLP);
% ALP-Activities on the Longest Path
% NLP-Nodes on the Longest Path
global Net n
nlp=size(NLP,2);
ALP=[]; % ALP are the activities that connect the NLP
for i=1:nlp-1
    k=find(Net(1,:)==NLP(i) & Net(2,:)==NLP(i+1));
    ALP=[ALP,k]; % Find origin NLP(i) and target NLP(i+1)
end % in Net. K is the activity no

```

## B.8 generateWs.m

```

for i=1:n % For all the activities generate W
    W(i,:)=generateW(Net(3,i)); % Net(3,i) contains the parameter
end % lambda of the exponential
% distribution

```

## B.9 generateW.m

```

function w=generateW(lambda);
% For a given lambda (exponential distribution),
% it generates 4 values of W, each with p=0.25.
t(1)=0;

```

```

e(1)=1;
for i=1:3
    t(i+1)=log(e(i)-0.25)/(-lambda);
    e(i+1)=exp(-lambda*t(i+1));
    w(i)=((t(i)+1/lambda)*e(i)-(t(i+1)+1/lambda)*e(i+1))/0.25;
end
w(4)=(t(4)+1/lambda)*e(4)/0.25;

```

## B.10 generateExpWs.m

```

for i=1:n                % For all the activities generate expW = 1/lambda
    expW(i)=1/Net(3,i);  % Net(3,i) contains the parameter
end                      % lambda of the exponential
                        % distribution

```

## B.11 durations.m

```

for i=1:n
    Y(1,i)=W(i,1)/Net(5,i); % Ymin=Wmin/Xmax
    Y(2,i)=W(i,4)/Net(4,i); % Ymax=Wmax/Xmin
end
clear i

```

## B.12 generateTlimits.m

```

% Tlim - Time interval (min,max) for the state
Tlim=[ ]; % Tlim stores the minimal and maximal
          % time of realization of each node.
Tlim(1,1)=0; % Tlim(1)=0

```

```

Tlim(2,1)=0;
for i=2:Net(2,n)
    [Orig,Act]=originNodes(i);                % Y=W/x (x=1) - duration
    Tlim(1,i)=max(Tlim(1,Orig)+Y(1,Act)); % Tlimj=max{Tlimi+Y}; j=2,...,m
    Tlim(2,i)=max(Tlim(2,Orig)+Y(2,Act));      % i->j
end
clear i Orig Act

```

## B.13 generateTvalues.m

```

% Constructs the Tval structure
% Tval.st : integer (step to jump between time values)
% Tval.T : array (all the time values the nodes are going to take)
Tval.st=0;
Tval.T=[ ];
for i=2:Net(2,n)-1
    [Tval(i).st,Tval(i).T]=calculateTvalues(Tlim(:,i));
end

```

## B.14 calculateTvalues.m

```

function [step,t]=calculateTvalues(it);
% Calculates Time Values within a given interval (it), and
% returns the time values (t) and the step used (step).
% The interval of values is divided by 40, otherwise too many points
n=round((it(2)-it(1))/40);          % n-number of values considered.
                                   % The greater the interval, the greater n
if n<4                               % (each 40 values - 1 interval)
    n=4;                             % If the interval is small, use 4 points (minimal)
end

```

```

elseif n>12      % If it is large, use 12 points (maximal)
    n=12;
end
step=(it(2)-it(1))/(n-1);
for i=1:n
    t(i)=it(1)+(i-1)*step;
end

```

## B.15 activitiesToFix.m

```

function AF=activitiesToFix(ALP)
% AF - Activities to be fixed are the
% activities that don't belong to the LP
global n
AF=1:n;      % Initialize AF with all activities in the Network
AF(ALP)=[ ]; % Remove from AF the activities in ALP

```

## B.16 xFixedActivities.m

```

function AF=xFixedActivities(AF)
global pfa;
j=1;      % pfa - n° of points (2 or 3)
for i=AF(1,:) % For all the fixed activities
    [x1,x2,x3]=calculateXs(i); % calculate the (2/3) values at which
    AF(2,j)=x1; % the resource should be fixed
    AF(3,j)=x2; % and store this values in line 2, 3 and 4
    if pfa==3
        AF(4,j)=x3;
    end
end

```

```

    j=j+1;
end

```

## B.17 calculateXs.m

```

function [x1,x2,x3]=calculateXs(i)
% pfa - n°of points of fixed activities
% For the activity i, calculates pda fixed
% values to give the resources xi1 and xi2 ...,
% e.g. if xi=[0.5...1.5], and pda=2, xi1=0.75 xi2=1.25.
% If pda=3 xi1=0.5, xi2=1, xi3=1.5
global Net pfa
xmin=Net(4,i);
xmax=Net(5,i);
if pfa==2
    v=(xmax-xmin)/4;      % in e.g. v=0.25
    x1=xmin+v;
    x2=xmax-v;
    x3=0;
elseif pfa==3
    x1=xmin;
    x2=(xmin+xmax)/2;
    x3=xmax;
end

```

## B.18 generateMainCode.m

```

% Writes to the file main.m the instruction that will
% after be called to optimize any network (via DP)

```

```

fm=fopen('main.m','w');
j=1;
fprintf(fm,'Res=[ ];\n');
X='X=';
for i=AF(1,:)
    % For every fixed activities, a for cycle is initiated.
    % This will allow to evaluate the expected cost for
    % every combination of the fixed variables
    % p.e. for x1=[0.75,1.25] ... for x2=[0.75,1.25] ...
    if pfa==2
        S=strcat('for x',int2str(i),'=[',num2str(AF(2,j))','',...
            num2str(AF(3,j)),'\n');
    elseif pfa==3
        S=strcat('for x',int2str(i),'=[',num2str(AF(2,j))','',...
            num2str(AF(3,j))','',num2str(AF(4,j)),'\n');
    end
    fprintf(fm,S);
    j=j+1;
    X=strcat(X,'x',int2str(i),'');
end
X(size(X,2))=']';
X=[X,'\n'];
% p.e. X=[0.75,0.75,1.25,1.25,1.25,1.25,1.25]
% is a vector with the fixed values of the activities
fprintf(fm,X);
% calculating the resource cost for the fixed activities
S=strcat('rc=calculateResourceCost(AF(1,:),X);\n');
fprintf(fm,S);
for i=1:size(ALP,2) % call for the states of the dp

```

```

    S=strcat('dps',int2str(i),';\n');
    fprintf(fm,S);
end
S=['Res=[Res;X,Best',int2str(sc),'];\n'];
fprintf(fm,S);
for i=AF(1,:)
    fprintf(fm,'end\n');
end
% Finds the minimum expected value
S='s=size(Res,2);\n';
S=[S,'m=min(Res(:,s));\n'];
S=[S,'k=max(find(Res(:,s)<=m));\n'];
% Prints result in the screen
S=[S,'fprintf("\nCurrent parameters");\n'];
S=[S,'fprintf("\nDue date (T) = %3d",T);\n'];
S=[S,'fprintf("\nCost of lateness (cl) = %2d\n\n",cl);\n'];
S=[S,'disp(' ');
for i=AF(1,:)
    S=[S,'x',int2str(i), ' '];
end
S=strcat(S,' x',int2str(ALP(1,1)), 'star ev',int2str(sc),'');\n'];
S=[S,'for i=1:size(Res,1)\n'];
S=[S,'fprintf(' ');
for i=AF(1,:)
    S=[S,'%6.2f '];
end
S=[S,'%6.2f %8.2f\n',Res(i,:);\n'];
S=[S,'end\n'];
S=[S,'fprintf("\nBest value was:\n");\n'];

```

```

S=[S,'disp(" ");
for i=AF(1,:)
    S=[S,'x',int2str(i),' '];
end
S=strcat(S,' x',int2str(ALP(1,1)), 'star ev',int2str(sc),'');\n';
S=[S,'fprintf("');
for i=AF(1,:)
    S=[S,'%%6.2f ');
end
S=[S,'%%6.2f %%8.2f\n',Res(k,:);\n'];
% -----
% Write result to a file (to export to excel)
S=[S,'fid = fopen("res.txt","w");\n'];
S=[S,'fprintf(fid,"\\nCurrent parameters");\n'];
S=[S,'fprintf(fid,"\\nDue date (T) = %%3d",T);\n'];
S=[S,'fprintf(fid,"\\nCost of lateness (cl) = %%2d\n\n",cl);\n'];
S=[S,'fprintf(fid,"');
for i=AF(1,:)
    S=[S,'x',int2str(i),' '];
end
S=[S,'x',int2str(ALP(1,1)), 'star, ev',int2str(sc),'\\n');\n'];
S=[S,'for i=1:size(Res,1)\n'];
S=[S,'fprintf(fid,"');
for i=AF(1,:)
    S=[S,'%%6.2f,'];
end
S=[S,'%%6.2f,%%8.2f\n',Res(i,:);\n'];
S=[S,'end\n'];
S=[S,'fprintf(fid,"\\nBest value was:\\n");\n'];

```



```

S=[S,'fprintf(fid,')'];
for i=AF(1,:)
    S=[S,'x',int2str(i),','];
end
S=[S,'x',int2str(ALP(1,1)),',star,ev',int2str(sc),'\n');
S=[S,'fprintf(fid,')'];
for i=AF(1,:)
    S=[S,'%%6.2f,'];
end
S=[S,'%%6.2f,%%8.2f\n',Res(k,:));\n'];
S=[S,'fclose(fid);\n'];
% -----
% Write result to a file (to read later into Matlab Workspace)
S=[S,'save workspace\n'];
fprintf(fm,S);
fprintf(fm,'clear x*');
fclose(fm);
clear fm i j S X

```

## B.19 generateDps1Code.m

```

% Writes to the file dps1.m the instruction that will
% after be called to optimize state 1 of any network (via DP)
% Find state times definition
% State 1 is defined by all node times that are origin of the last node
[OLN,A]=originNodes(Net(2,n)); % OLN - Origins of Last Node
a=ALP(size(ALP,2));          % Activity to optimize - last activity
o=Net(1,a);                  % of the longest path
k=find(OLN(:)==o);          % Eliminate from OLN origin of decision variable

```

```

OLN(k)=[ ];
A(k)=[ ];
S='Best1=[ ];\n';
for i=OLN
    if i~=1
        S=[S,'for t',int2str(i),'=Tval(',int2str(i),').T\n'];
    end
end
S=[S,'[Tn,PTn]=calculateMaxN('];
k=1;
for i=OLN
    j=A(k);
    S=[S,'t',int2str(i),'+W(',int2str(j),',:)/x',int2str(j),','];
    k=k+1;
end
S(size(S,2))=')';
S=[S,'];\n';
S=[S,'for t',int2str(o),'=Tval(',int2str(o),').T\n'];
S=[S,'EV=[ ];\n'];
st=(Net(5,a)-Net(4,a))/4;
S=[S,'for x',int2str(a),'=',num2str(Net(4,a)),',:',num2str(st),',:',num2str(Net(5,a)),'\n'];
S=[S,'ev=0;\n'];
S=[S,'for i=1:4\n'];
S=[S,'w=W(',int2str(a),',i);\n'];
S=[S,'TN=max(Tn,t',int2str(o),'+w/x',int2str(a),');\n']; % P=PY7
S=[S,'U=max(0,TN-T);\n']; % P=PY7
S=[S,'evU=expectedValue(U,PTn);\n'];
S=[S,'IC(i)=x',int2str(a),'*w+cl*evU;\n'];
% Intermediate cost = Delay cost + resource cost of last act.

```

```

% 4 values - Probability=P
S=[S,'end\n'];
S=[S,'F1=IC+rc;\n']; % total cost
S=[S,'ev=expectedValue(F1,P);\n'];
S=[S,'EV=[EV;x',int2str(a),'ev'];\n'];
S=[S,'end\n'];
S=[S,'m=min(EV(:,2));\n'];
S=[S,'k=max(find(EV(:,2)<=m));\n'];
S=[S,'x',int2str(a),'star=EV(k,1);\n'];
S=[S,'v1star=EV(k,2);\n'];
S=[S,'Best1=[Best1;'];
for i=OLN
    if i~=1
        S=[S,'t',int2str(i),''];
    end
end
S=[S,'t',int2str(o),'x',int2str(a),'star,v1star'];\n'];
for i=OLN
    if i~=1
        S=[S,'end\n'];
    end
end
S=[S,'end\n'];
fs1=fopen('dps1.m','w');
fprintf(fs1,S);
fclose(fs1);
OLS=[OLN,o]; % Origins of Last State
k=find(OLS(:)==1); % Eliminate from OLS node 1
OLS(k)=[ ]; % (t1=0 - no need to include)

```

```
clear OLN A fs1 a i j o k s t
```

## B.20 generateDpsNCode.m

```
% Writes to the file dpsn.m the instruction that will
% after be called to optimize state n (>1) of any network (via DP)
% Find state times definition
% State n is defined by all node times that are origin of the last state
OS=[ ]; % Origins of this State
for i=OLS % Origins of Last State
    [O,A]=originNodes(i);
    OS=[OS,O];
end
OS=reduce(OS);
k=find(OS(:)==1); % Eliminate from OS node 1 (t1=0-no need to include)
OS(k)=[ ];
a=ALP(size(ALP,2)-sc+1); % Activity to optimize
o=Net(1,a);
S=['Best',int2str(sc),'=[ ];\n'];
for i=OS
    S=[S,'for t',int2str(i),'=Tval(',int2str(i),').T\n'];
end
S=[S,'EV=[ ];\n'];
st=(Net(5,a)-Net(4,a))/4;
S=[S,'for x',int2str(a),'=',num2str(Net(4,a)),':',num2str(st),':',num2str(Net(5,a)),'\n'];
Tau=[ ]; % Values to input to the previous state
for i=OLS
    j=find(OS(:)==i);
    if isempty(j) % for the nodes on this state not in the last state
```

```

Tau=[Tau,i];
S=[S,['T',int2str(i),'PT',int2str(i)]=calculateMaxN(');
k=1;
[O,A]=originNodes(i);
for l=O % l - node
    m=A(k); % m - activity
    S=[S,'t',int2str(l),'+W(',int2str(m),',:)/x',int2str(m),','];
    k=k+1;
end
S(size(S,2)=='');
S=[S,'\n'];
S=[S,'pT',int2str(i),'=mass(PT',int2str(i),')\n'];
end
end
S=[S,'Fa=[ ];\n'];
S=[S,'pFa=[ ];\n'];
S=[S,'PFa=[ ];\n'];
S=[S,'m=1;\n'];
for i=Tau % Determination of the n°of elements of Ti
    S=[S,'sT',int2str(i),'=size(T',int2str(i),'2);\n'];
end
for i=Tau
    S=[S,'for i',int2str(i),'=1:sT',int2str(i),'\n'];
end
j=1;
S=[S,'k=min(find(');
for i=OLS
    k=find(Tau(:)==i);
    if isempty(k)

```

```

S=[S,'Best',int2str(sc-1),':',int2str(j),'==t',int2str(i),' & '];
else
S=[S,'abs(Best',int2str(sc-1),':',int2str(j),'-T',int2str(i),'(i',int2str(i))),
<=Tval('int2str(i)').st/2',' & '];
end
j=j+1;
end
S(size(S,2)-1)='';
S=[S,']\n'];
S=[S,'Fa(m)=Best',int2str(sc-1),'(k',int2str(j+1),')\n'];
S=[S,'pFa(m)='];
for i= Tau
S=[S,'pT',int2str(i),'(i',int2str(i),')*'];
end
S(size(S,2))='';
S=[S,']\n'];
S=[S,'m=m+1;\n'];
for i= Tau
S=[S,'end\n'];
end
S=[S,'PFa=cum(pFa);\n'];
S=[S,'evFa=expectedValue(Fa,PFa);\n'];
S=[S,'Fb=x',int2str(a),'*W(',int2str(a),',:)+evFa;\n'];
S=[S,'ev=expectedValue(Fb,P);\n'];
S=[S,'EV=[EV;x',int2str(a),',ev);\n'];
S=[S,'end\n'];
S=[S,'m=min(EV(:,2))\n'];
S=[S,'k=max(find(EV(:,2)<=m))\n'];
S=[S,'x',int2str(a),'star=EV(k,1);\n'];

```

```

S=[S,'vbstar=EV(k,2);\n'];
S=[S,'Best',int2str(sc),'=[Best',int2str(sc),','];
for i=OS
    S=[S,'t',int2str(i),','];
end
S=[S,'x',int2str(a),'star,vbstar'];\n'];
for i=OS
    S=[S,'end\n'];
end
fsn=fopen(['dps',int2str(sc),'.m'],'w');
fprintf(fsn,S);
fclose(fsn);
OLS=OS;           % Origins of Last State
k=find(OLS(:)==1); % Eliminate from OLS node 1
OLS(k)=[ ];      % (t1=0 - no need to include)
clear OS O A S Tau fsn a i j k l m o

```

## B.21 reduce.m

```

function Y=reduce(Ybig);           %takes off all repeated elements
Y=[];
while ~isempty(Ybig)
    y=min(Ybig);
    Y=[Y,y];
    k=find(Ybig(:)==y);           % all indexes
    Ybig(k)=[ ];
end

```

## B.22 calculateResourceCost.m

```
function rc=calculateResourceCost(Af,X)
% rc-resource cost for the fixed activities
global expW
rc=0;
j=1;
for i=Af
    rc=rc+X(j)*expW(i);
    j=j+1;
end
```

## B.23 cum.m

```
function P=cum(p)    % transforms probability mass function
                    % into cumulative probability
s=size(p,2);
P(1)=p(1);
for i=2:s
    P(i)=P(i-1)+p(i);
end
```

## B.24 calculateMaxN.m

```
function [Yt,PYt]=calculateMaxN(varargin)
% Calculates the max of a variable number of arguments
global P
n=nargin;
if n==1
    Yt=varargin{1};
```



```

    PYt=P;
    return
end
Yt=calculateMax(varargin{1},varargin{2});
PYt=calculatePmax(Yt,varargin{1},P,varargin{2},P);
i=3;
while i<=n
    Yt1=Yt;
    PYt1=PYt;
    Yt=calculateMax(Yt1,varargin{i});
    PYt=calculatePmax(Yt,Yt1,PYt1,varargin{i},P);
    i=i+1;
end

```

## B.25 calculateMax.m

```

function Y=calculateMax(X1,X2);
k=1;
s1=size(X1,2);
s2=size(X2,2);
for i=1:s1
    for j=1:s2
        Ybig(k)=max(X1(i),X2(j)); % the MAX can have repeated values
        k=k+1;
    end
end
end
Y=reduce(Ybig); % takes off the repeated values

```

**B.26 calculatePmax.m**

```

function P=calculatePmax(Y,X1,P1,X2,P2);
% P(Y<=y)=P(X1<=y).P(X2<=y)
i=1;
dim=size(Y,2);
while i<=dim
    y=Y(i);
    c=cumProb(X1,P1,y)*cumProb(X2,P2,y);
    P(i)=c;
    i=i+1;
end

```

**B.27 cumProb.m**

```

function cP=cumProb(W,P,n) % P(W<=n)
k=max(find(W(:)<=(n+0.000000001)));
cP=P(k);

```

**B.28 mass.m**

```

function p=mass(P) % transforms cumulative probability into
                  % probability mass function
s=size(P,2);
p(1)=P(1);
for i=2:s
    p(i)=P(i)-P(i-1);
end

```

## **B.29** expectedValue.m

```
function ev=expectedValue(X,P)
    p=mass(P);
    ev=sum(p.*X); % ev=p1.X1 + p2.X2 + ...
```



# Apêndice C

## Código gerado pela versão 2

### C.1 main.m

```
Res=[ ];
for x2=[0.5,1,1.5]
    for x3=[0.5,1,1.5]
        for x5=[0.5,1,1.5]
            for x6=[0.5,1,1.5]
                for x8=[0.5,1,1.5]
                    for x9=[0.5,1,1.5]
                        for x10=[0.5,1,1.5]
                            X=[x2,x3,x5,x6,x8,x9,x10];
                            rc=calculateResourceCost(AF(1,:),X);
                            dps1;
                            dps2;
                            dps3;
                            dps4;
                            Res=[Res;X,Best4];
                        end
                    end
                end
            end
        end
    end
end
```

```

        end
    end
end
end
end
end
end
end
s=size(Res,2);
m=min(Res(:,s));
k=max(find(Res(:,s)<=m));
fprintf('\nCurrent parameters');
fprintf('\nDue date (T) = %3d',T);
fprintf('\nCost of lateness (cl) = %2d\n\n',cl);
disp(' x2 x3 x5 x6 x8 x9 x10 x1star ev4');
for i=1:size(Res,1)
    fprintf('%6.2f %6.2f %6.2f %6.2f %6.2f %6.2f %6.2f %6.2f %8.2f\n',Res(i,:));
end
fprintf('\nBest value was:\n');
disp(' x2 x3 x5 x6 x8 x9 x10 x1star ev4');
fprintf('%6.2f %6.2f %6.2f %6.2f %6.2f %6.2f %6.2f %6.2f %8.2f\n',Res(k,:));
fid = fopen('res.txt','w');
fprintf(fid,'\nCurrent parameters');
fprintf(fid,'\nDue date (T) = %3d',T);
fprintf(fid,'\nCost of lateness (cl) = %2d\n\n',cl);
fprintf(fid,'x2,x3,x5,x6,x8,x9,x10,x1star,ev4\n');
for i=1:size(Res,1)
    fprintf(fid,'%6.2f,%6.2f,%6.2f,%6.2f,%6.2f,%6.2f,%6.2f,%6.2f,%8.2f\n',Res(i,:));
end
fprintf(fid,'\nBest value was:\n');
fprintf(fid,'x2,x3,x5,x6,x8,x9,x10,x1star,ev4\n');

```

```

fprintf(fid,'%6.2f,%6.2f,%6.2f,%6.2f,%6.2f,%6.2f,%6.2f,%6.2f,%8.2f\n',Res(k,:));
fclose(fid);
save workspace
clear x*

```

## C.2 *dps1.m*

```

Best1=[ ];
for t4=Tval(4).T
  for t5=Tval(5).T
    [Tn,PTn]=calculateMaxN(t4+W(9,+)/x9,t5+W(10,+)/x10);
    for t6=Tval(6).T
      EV=[ ];
      for x11=0.5:0.25:1.5
        ev=0;
        for i=1:4
          w=W(11,i);
          TN=max(Tn,t6+w/x11);
          U=max(0,TN-T);
          evU=expectedValue(U,PTn);
          IC(i)=x11*w+cl*evU;
        end
        F1=IC+rc;
        ev=expectedValue(F1,P);
        EV=[EV;x11,ev];
      end
      m=min(EV(:,2));
      k=max(find(EV(:,2)<=m));
      x11star=EV(k,1);
    end
  end
end

```

```

v1star=EV(k,2);
Best1=[Best1;t4,t5,t6,x11star,v1star];
end
end
end

```

### C.3 dps2.m

```

Best2=[ ];
for t2=Tval(2).T
for t3=Tval(3).T
for t4=Tval(4).T
EV=[ ];
for x7=0.5:0.25:1.5
[T5,PT5]=calculateMaxN(t2+W(5,+)/x5,t4+W(8,+)/x8);
pT5=mass(PT5);
[T6,PT6]=calculateMaxN(t2+W(6,+)/x6,t3+W(7,+)/x7);
pT6=mass(PT6);
Fa=[ ];
pFa=[ ];
PFa=[ ];
m=1;
sT5=size(T5,2);
sT6=size(T6,2);
for i5=1:sT5
for i6=1:sT6
k=min(find(Best1(:,1)==t4 & abs(Best1(:,2)-T5(i5))<=Tval(5).st/2
& abs(Best1(:,3)-T6(i6))<=Tval(6).st/2 ) );
Fa(m)=Best1(k,5);

```



```

        pFa(m)=pT5(i5)*pT6(i6);
        m=m+1;
    end
end
PFa=cum(pFa);
evFa=expectedValue(Fa,PFa);
Fb=x7*W(7,)+evFa;
ev=expectedValue(Fb,P);
EV=[EV;x7,ev];
end
m=min(EV(:,2));
k=max(find(EV(:,2)<=m));
x7star=EV(k,1);
vbstar=EV(k,2);
Best2=[Best2;t2,t3,t4,x7star,vbstar];
end
end
end

```

## C.4 dps3.m

```

Best3=[ ];
for t2=Tval(2).T
    EV=[ ];
    for x4=0.5:0.25:1.5
        [T3,PT3]=calculateMaxN(t1+W(2,)/x2,t2+W(4,)/x4);
        pT3=mass(PT3);
        [T4,PT4]=calculateMaxN(t1+W(3,)/x3);
        pT4=mass(PT4);
    end
end
end

```

```

Fa=[ ];
pFa=[ ];
PFa=[ ];
m=1;
sT3=size(T3,2);
sT4=size(T4,2);
for i3=1:sT3
    for i4=1:sT4
        k=min(find(Best2(:,1)==t2 & abs(Best2(:,2)-T3(i3))<=Tval(3).st/2
            &abs(Best2(:,3)-T4(i4))<=Tval(4).st/2 ) );
        Fa(m)=Best2(k,5);
        pFa(m)=pT3(i3)*pT4(i4);
        m=m+1;
    end
end
PFa=cum(pFa);
evFa=expectedValue(Fa,PFa);
Fb=x4*W(4,.)+evFa;
ev=expectedValue(Fb,P);
EV=[EV;x4,ev];
end
m=min(EV(:,2));
k=max(find(EV(:,2)<=m));
x4star=EV(k,1);
vbstar=EV(k,2);
Best3=[Best3;t2,x4star,vbstar];
end

```

## C.5 dps4.m

```

Best4=[ ];
EV=[ ];
for x1=0.5:0.25:1.5
    [T2,PT2]=calculateMaxN(t1+W(1,+)/x1);
    pT2=mass(PT2);
    Fa=[ ];
    pFa=[ ];
    PFa=[ ];
    m=1;
    sT2=size(T2,2);
    for i2=1:sT2
        k=min(find(abs(Best3(:,1)-T2(i2))<=Tval(2).st/2 ) );
        Fa(m)=Best3(k,3);
        pFa(m)=pT2(i2);
        m=m+1;
    end
    PFa=cum(pFa);
    evFa=expectedValue(Fa,PFa);
    Fb=x1*W(1,)+evFa;
    ev=expectedValue(Fb,P);
    EV=[EV;x1,ev];
end
m=min(EV(:,2));
k=max(find(EV(:,2)<=m));
x1star=EV(k,1);
vbstar=EV(k,2);
Best4=[Best4;x1star,vbstar];

```



# Apêndice D

## Resultados do programa

### D.1 Procura do óptimo para as várias combinações de $\mathcal{F}$

$x_2$	$x_3$	$x_5$	$x_6$	$x_8$	$x_9$	$x_{10}$	$x_1^*$	$ev_4$
0.50	0.50	0.50	0.50	0.50	0.50	0.50	1.00	589.70
0.50	0.50	0.50	0.50	0.50	0.50	1.00	1.00	581.96
0.50	0.50	0.50	0.50	0.50	0.50	1.50	1.00	582.31
0.50	0.50	0.50	0.50	0.50	1.00	0.50	1.00	477.49
0.50	0.50	0.50	0.50	0.50	1.00	1.00	1.00	465.52
0.50	0.50	0.50	0.50	0.50	1.00	1.50	1.00	464.44
0.50	0.50	0.50	0.50	0.50	1.50	0.50	1.00	472.78
0.50	0.50	0.50	0.50	0.50	1.50	1.00	1.00	458.99
0.50	0.50	0.50	0.50	0.50	1.50	1.50	1.00	457.18
0.50	0.50	0.50	0.50	1.00	0.50	0.50	1.00	553.81
0.50	0.50	0.50	0.50	1.00	0.50	1.00	1.00	549.11
0.50	0.50	0.50	0.50	1.00	0.50	1.50	1.00	550.59
0.50	0.50	0.50	0.50	1.00	1.00	0.50	1.00	426.45
0.50	0.50	0.50	0.50	1.00	1.00	1.00	1.00	418.73
0.50	0.50	0.50	0.50	1.00	1.00	1.50	1.00	419.26
0.50	0.50	0.50	0.50	1.00	1.50	0.50	1.25	415.23
0.50	0.50	0.50	0.50	1.00	1.50	1.00	1.25	405.13
0.50	0.50	0.50	0.50	1.00	1.50	1.50	1.25	404.58
0.50	0.50	0.50	0.50	1.50	0.50	0.50	1.00	548.86
0.50	0.50	0.50	0.50	1.50	0.50	1.00	1.00	546.55
0.50	0.50	0.50	0.50	1.50	0.50	1.50	1.00	548.82
0.50	0.50	0.50	0.50	1.50	1.00	0.50	1.25	413.84
0.50	0.50	0.50	0.50	1.50	1.00	1.00	1.25	407.84

$x_2$	$x_3$	$x_5$	$x_6$	$x_8$	$x_9$	$x_{10}$	$x_{11}^*$	$ev_4$
0.50	0.50	0.50	0.50	1.50	1.00	1.50	1.25	409.10
0.50	0.50	0.50	0.50	1.50	1.50	0.50	1.25	397.70
0.50	0.50	0.50	0.50	1.50	1.50	1.00	1.25	389.72
0.50	0.50	0.50	0.50	1.50	1.50	1.50	1.25	390.04
0.50	0.50	0.50	1.00	0.50	0.50	0.50	1.25	574.32
0.50	0.50	0.50	1.00	0.50	0.50	1.00	1.25	565.21
0.50	0.50	0.50	1.00	0.50	0.50	1.50	1.25	565.22
0.50	0.50	0.50	1.00	0.50	1.00	0.50	1.25	453.16
0.50	0.50	0.50	1.00	0.50	1.00	1.00	1.25	438.55
0.50	0.50	0.50	1.00	0.50	1.00	1.50	1.25	436.81
0.50	0.50	0.50	1.00	0.50	1.50	0.50	1.25	442.03
0.50	0.50	0.50	1.00	0.50	1.50	1.00	1.25	425.19
0.50	0.50	0.50	1.00	0.50	1.50	1.50	1.25	422.52
0.50	0.50	0.50	1.00	1.00	0.50	0.50	1.25	532.86
0.50	0.50	0.50	1.00	1.00	0.50	1.00	1.25	526.45
0.50	0.50	0.50	1.00	1.00	0.50	1.50	1.25	527.48
0.50	0.50	0.50	1.00	1.00	1.00	0.50	1.25	395.52
0.50	0.50	0.50	1.00	1.00	1.00	1.00	1.25	384.76
0.50	0.50	0.50	1.00	1.00	1.00	1.50	1.25	384.49
0.50	0.50	0.50	1.00	1.00	1.50	0.50	1.25	376.95
0.50	0.50	0.50	1.00	1.00	1.50	1.00	1.25	363.04
0.50	0.50	0.50	1.00	1.00	1.50	1.50	1.25	361.32
0.50	0.50	0.50	1.00	1.50	0.50	0.50	1.25	524.25
0.50	0.50	0.50	1.00	1.50	0.50	1.00	1.25	520.38
0.50	0.50	0.50	1.00	1.50	0.50	1.50	1.25	522.28
0.50	0.50	0.50	1.00	1.50	1.00	0.50	1.25	376.66
0.50	0.50	0.50	1.00	1.50	1.00	1.00	1.25	368.14
0.50	0.50	0.50	1.00	1.50	1.00	1.50	1.25	368.66
0.50	0.50	0.50	1.00	1.50	1.50	0.50	1.25	352.15
0.50	0.50	0.50	1.00	1.50	1.50	1.00	1.25	340.57
0.50	0.50	0.50	1.00	1.50	1.50	1.50	1.25	339.95
0.50	0.50	0.50	1.50	0.50	0.50	0.50	1.00	584.73
0.50	0.50	0.50	1.50	0.50	0.50	1.00	1.00	575.32
0.50	0.50	0.50	1.50	0.50	0.50	1.50	1.00	575.27
0.50	0.50	0.50	1.50	0.50	1.00	0.50	1.00	463.07
0.50	0.50	0.50	1.50	0.50	1.00	1.00	1.25	448.04
0.50	0.50	0.50	1.50	0.50	1.00	1.50	1.25	446.19
0.50	0.50	0.50	1.50	0.50	1.50	0.50	1.25	451.51
0.50	0.50	0.50	1.50	0.50	1.50	1.00	1.25	434.12
0.50	0.50	0.50	1.50	0.50	1.50	1.50	1.25	431.32
0.50	0.50	0.50	1.50	1.00	0.50	0.50	1.00	543.19
0.50	0.50	0.50	1.50	1.00	0.50	1.00	1.00	536.43
0.50	0.50	0.50	1.50	1.00	0.50	1.50	1.00	537.41
0.50	0.50	0.50	1.50	1.00	1.00	0.50	1.25	405.10
0.50	0.50	0.50	1.50	1.00	1.00	1.00	1.25	393.86
0.50	0.50	0.50	1.50	1.00	1.00	1.50	1.25	393.49

D.1. PROCURA DO ÓPTIMO PARA AS VÁRIAS COMBINAÇÕES DE F229

$x_2$	$x_3$	$x_5$	$x_6$	$x_8$	$x_9$	$x_{10}$	$x_1^*$	ev <sub>4</sub>
0.50	0.50	0.50	1.50	1.00	1.50	0.50	1.25	385.97
0.50	0.50	0.50	1.50	1.00	1.50	1.00	1.25	371.47
0.50	0.50	0.50	1.50	1.00	1.50	1.50	1.25	369.61
0.50	0.50	0.50	1.50	1.50	0.50	0.50	1.25	534.26
0.50	0.50	0.50	1.50	1.50	0.50	1.00	1.25	530.02
0.50	0.50	0.50	1.50	1.50	0.50	1.50	1.25	531.86
0.50	0.50	0.50	1.50	1.50	1.00	0.50	1.25	385.62
0.50	0.50	0.50	1.50	1.50	1.00	1.00	1.25	376.55
0.50	0.50	0.50	1.50	1.50	1.00	1.50	1.25	376.96
0.50	0.50	0.50	1.50	1.50	1.50	0.50	1.25	360.50
0.50	0.50	0.50	1.50	1.50	1.50	1.00	1.25	348.28
0.50	0.50	0.50	1.50	1.50	1.50	1.50	1.25	347.51
0.50	0.50	1.00	0.50	0.50	0.50	0.50	1.00	592.03
0.50	0.50	1.00	0.50	0.50	0.50	1.00	1.00	584.41
0.50	0.50	1.00	0.50	0.50	0.50	1.50	1.00	584.80
0.50	0.50	1.00	0.50	0.50	1.00	0.50	1.00	479.76
0.50	0.50	1.00	0.50	0.50	1.00	1.00	1.00	467.95
0.50	0.50	1.00	0.50	0.50	1.00	1.50	1.00	466.92
0.50	0.50	1.00	0.50	0.50	1.50	0.50	1.00	474.97
0.50	0.50	1.00	0.50	0.50	1.50	1.00	1.00	461.41
0.50	0.50	1.00	0.50	0.50	1.50	1.50	1.00	459.66
0.50	0.50	1.00	0.50	1.00	0.50	0.50	1.00	556.06
0.50	0.50	1.00	0.50	1.00	0.50	1.00	1.00	551.55
0.50	0.50	1.00	0.50	1.00	0.50	1.50	1.00	553.07
0.50	0.50	1.00	0.50	1.00	1.00	0.50	1.00	428.58
0.50	0.50	1.00	0.50	1.00	1.00	1.00	1.00	421.13
0.50	0.50	1.00	0.50	1.00	1.00	1.50	1.00	421.73
0.50	0.50	1.00	0.50	1.00	1.50	0.50	1.00	417.48
0.50	0.50	1.00	0.50	1.00	1.50	1.00	1.25	407.57
0.50	0.50	1.00	0.50	1.00	1.50	1.50	1.25	407.06
0.50	0.50	1.00	0.50	1.50	0.50	0.50	1.00	551.10
0.50	0.50	1.00	0.50	1.50	0.50	1.00	1.00	548.98
0.50	0.50	1.00	0.50	1.50	0.50	1.50	1.00	551.30
0.50	0.50	1.00	0.50	1.50	1.00	0.50	1.00	416.01
0.50	0.50	1.00	0.50	1.50	1.00	1.00	1.25	410.29
0.50	0.50	1.00	0.50	1.50	1.00	1.50	1.25	411.58
0.50	0.50	1.00	0.50	1.50	1.50	0.50	1.25	399.97
0.50	0.50	1.00	0.50	1.50	1.50	1.00	1.25	392.16
0.50	0.50	1.00	0.50	1.50	1.50	1.50	1.25	392.52
0.50	0.50	1.00	1.00	0.50	0.50	0.50	1.25	576.74
0.50	0.50	1.00	1.00	0.50	0.50	1.00	1.25	567.69
0.50	0.50	1.00	1.00	0.50	0.50	1.50	1.25	567.71
0.50	0.50	1.00	1.00	0.50	1.00	0.50	1.25	455.54
0.50	0.50	1.00	1.00	0.50	1.00	1.00	1.25	441.02
0.50	0.50	1.00	1.00	0.50	1.00	1.50	1.25	439.30
0.50	0.50	1.00	1.00	0.50	1.50	0.50	1.25	444.37

$x_2$	$x_3$	$x_5$	$x_6$	$x_8$	$x_9$	$x_{10}$	$x_1^*$	$ev_4$
0.50	0.50	1.00	1.00	0.50	1.50	1.00	1.25	427.65
0.50	0.50	1.00	1.00	0.50	1.50	1.50	1.25	425.01
0.50	0.50	1.00	1.00	1.00	0.50	0.50	1.25	535.23
0.50	0.50	1.00	1.00	1.00	0.50	1.00	1.25	528.91
0.50	0.50	1.00	1.00	1.00	0.50	1.50	1.25	529.97
0.50	0.50	1.00	1.00	1.00	1.00	0.50	1.25	397.83
0.50	0.50	1.00	1.00	1.00	1.00	1.00	1.25	387.21
0.50	0.50	1.00	1.00	1.00	1.00	1.50	1.25	386.97
0.50	0.50	1.00	1.00	1.00	1.50	0.50	1.25	379.22
0.50	0.50	1.00	1.00	1.00	1.50	1.00	1.25	365.48
0.50	0.50	1.00	1.00	1.00	1.50	1.50	1.25	363.80
0.50	0.50	1.00	1.00	1.50	0.50	0.50	1.25	526.62
0.50	0.50	1.00	1.00	1.50	0.50	1.00	1.25	522.84
0.50	0.50	1.00	1.00	1.50	0.50	1.50	1.25	524.77
0.50	0.50	1.00	1.00	1.50	1.00	0.50	1.25	378.98
0.50	0.50	1.00	1.00	1.50	1.00	1.00	1.25	370.59
0.50	0.50	1.00	1.00	1.50	1.00	1.50	1.25	371.15
0.50	0.50	1.00	1.00	1.50	1.50	0.50	1.25	354.41
0.50	0.50	1.00	1.00	1.50	1.50	1.00	1.25	343.01
0.50	0.50	1.00	1.00	1.50	1.50	1.50	1.25	342.44
0.50	0.50	1.00	1.50	0.50	0.50	0.50	1.00	587.01
0.50	0.50	1.00	1.50	0.50	0.50	1.00	1.00	577.76
0.50	0.50	1.00	1.50	0.50	0.50	1.50	1.00	577.76
0.50	0.50	1.00	1.50	0.50	1.00	0.50	1.00	465.23
0.50	0.50	1.00	1.50	0.50	1.00	1.00	1.00	450.48
0.50	0.50	1.00	1.50	0.50	1.00	1.50	1.25	448.68
0.50	0.50	1.00	1.50	0.50	1.50	0.50	1.25	453.80
0.50	0.50	1.00	1.50	0.50	1.50	1.00	1.25	436.57
0.50	0.50	1.00	1.50	0.50	1.50	1.50	1.25	433.80
0.50	0.50	1.00	1.50	1.00	0.50	0.50	1.00	545.34
0.50	0.50	1.00	1.50	1.00	0.50	1.00	1.00	538.84
0.50	0.50	1.00	1.50	1.00	0.50	1.50	1.00	539.88
0.50	0.50	1.00	1.50	1.00	1.00	0.50	1.25	407.34
0.50	0.50	1.00	1.50	1.00	1.00	1.00	1.25	396.29
0.50	0.50	1.00	1.50	1.00	1.00	1.50	1.25	395.97
0.50	0.50	1.00	1.50	1.00	1.50	0.50	1.25	388.15
0.50	0.50	1.00	1.50	1.00	1.50	1.00	1.25	373.89
0.50	0.50	1.00	1.50	1.00	1.50	1.50	1.25	372.09
0.50	0.50	1.00	1.50	1.50	0.50	0.50	1.00	536.45
0.50	0.50	1.00	1.50	1.50	0.50	1.00	1.00	532.46
0.50	0.50	1.00	1.50	1.50	0.50	1.50	1.25	534.34
0.50	0.50	1.00	1.50	1.50	1.00	0.50	1.25	387.87
0.50	0.50	1.00	1.50	1.50	1.00	1.00	1.25	378.98
0.50	0.50	1.00	1.50	1.50	1.00	1.50	1.25	379.44
0.50	0.50	1.00	1.50	1.50	1.50	0.50	1.25	362.68
0.50	0.50	1.00	1.50	1.50	1.50	1.00	1.25	350.70



D.1. PROCURA DO ÓPTIMO PARA AS VÁRIAS COMBINAÇÕES DE F231

$x_2$	$x_3$	$x_5$	$x_6$	$x_8$	$x_9$	$x_{10}$	$x_1^*$	ev <sub>4</sub>
0.50	0.50	1.00	1.50	1.50	1.50	1.50	1.25	349.99
0.50	0.50	1.50	0.50	0.50	0.50	0.50	1.00	594.53
0.50	0.50	1.50	0.50	0.50	0.50	1.00	1.00	586.91
0.50	0.50	1.50	0.50	0.50	0.50	1.50	1.00	587.30
0.50	0.50	1.50	0.50	0.50	1.00	0.50	1.00	482.26
0.50	0.50	1.50	0.50	0.50	1.00	1.00	1.00	470.45
0.50	0.50	1.50	0.50	0.50	1.00	1.50	1.00	469.42
0.50	0.50	1.50	0.50	0.50	1.50	0.50	1.00	477.47
0.50	0.50	1.50	0.50	0.50	1.50	1.00	1.00	463.91
0.50	0.50	1.50	0.50	0.50	1.50	1.50	1.00	462.16
0.50	0.50	1.50	0.50	1.00	0.50	0.50	1.00	558.56
0.50	0.50	1.50	0.50	1.00	0.50	1.00	1.00	554.05
0.50	0.50	1.50	0.50	1.00	0.50	1.50	1.00	555.57
0.50	0.50	1.50	0.50	1.00	1.00	0.50	1.00	431.08
0.50	0.50	1.50	0.50	1.00	1.00	1.00	1.00	423.63
0.50	0.50	1.50	0.50	1.00	1.00	1.50	1.00	424.23
0.50	0.50	1.50	0.50	1.00	1.50	0.50	1.00	419.98
0.50	0.50	1.50	0.50	1.00	1.50	1.00	1.25	410.07
0.50	0.50	1.50	0.50	1.00	1.50	1.50	1.25	409.56
0.50	0.50	1.50	0.50	1.50	0.50	0.50	1.00	553.60
0.50	0.50	1.50	0.50	1.50	0.50	1.00	1.00	551.48
0.50	0.50	1.50	0.50	1.50	0.50	1.50	1.00	553.80
0.50	0.50	1.50	0.50	1.50	1.00	0.50	1.00	418.51
0.50	0.50	1.50	0.50	1.50	1.00	1.00	1.25	412.79
0.50	0.50	1.50	0.50	1.50	1.00	1.50	1.25	414.08
0.50	0.50	1.50	0.50	1.50	1.50	0.50	1.25	402.47
0.50	0.50	1.50	0.50	1.50	1.50	1.00	1.25	394.66
0.50	0.50	1.50	0.50	1.50	1.50	1.50	1.25	395.02
0.50	0.50	1.50	1.00	0.50	0.50	0.50	1.25	579.24
0.50	0.50	1.50	1.00	0.50	0.50	1.00	1.25	570.19
0.50	0.50	1.50	1.00	0.50	0.50	1.50	1.25	570.21
0.50	0.50	1.50	1.00	0.50	1.00	0.50	1.25	458.04
0.50	0.50	1.50	1.00	0.50	1.00	1.00	1.25	443.52
0.50	0.50	1.50	1.00	0.50	1.00	1.50	1.25	441.80
0.50	0.50	1.50	1.00	0.50	1.50	0.50	1.25	446.87
0.50	0.50	1.50	1.00	0.50	1.50	1.00	1.25	430.15
0.50	0.50	1.50	1.00	0.50	1.50	1.50	1.25	427.51
0.50	0.50	1.50	1.00	1.00	0.50	0.50	1.25	537.73
0.50	0.50	1.50	1.00	1.00	0.50	1.00	1.25	531.41
0.50	0.50	1.50	1.00	1.00	0.50	1.50	1.25	532.47
0.50	0.50	1.50	1.00	1.00	1.00	0.50	1.25	400.33
0.50	0.50	1.50	1.00	1.00	1.00	1.00	1.25	389.71
0.50	0.50	1.50	1.00	1.00	1.00	1.50	1.25	389.47
0.50	0.50	1.50	1.00	1.00	1.50	0.50	1.25	381.72
0.50	0.50	1.50	1.00	1.00	1.50	1.00	1.25	367.98
0.50	0.50	1.50	1.00	1.00	1.50	1.50	1.25	366.30

$x_2$	$x_3$	$x_5$	$x_6$	$x_8$	$x_9$	$x_{10}$	$x_1^*$	$ev_4$
0.50	0.50	1.50	1.00	1.50	0.50	0.50	1.25	529.12
0.50	0.50	1.50	1.00	1.50	0.50	1.00	1.25	525.34
0.50	0.50	1.50	1.00	1.50	0.50	1.50	1.25	527.27
0.50	0.50	1.50	1.00	1.50	1.00	0.50	1.25	381.48
0.50	0.50	1.50	1.00	1.50	1.00	1.00	1.25	373.09
0.50	0.50	1.50	1.00	1.50	1.00	1.50	1.25	373.65
0.50	0.50	1.50	1.00	1.50	1.50	0.50	1.25	356.91
0.50	0.50	1.50	1.00	1.50	1.50	1.00	1.25	345.51
0.50	0.50	1.50	1.00	1.50	1.50	1.50	1.25	344.94
0.50	0.50	1.50	1.50	0.50	0.50	0.50	1.00	589.51
0.50	0.50	1.50	1.50	0.50	0.50	1.00	1.00	580.26
0.50	0.50	1.50	1.50	0.50	0.50	1.50	1.00	580.26
0.50	0.50	1.50	1.50	0.50	1.00	0.50	1.00	467.73
0.50	0.50	1.50	1.50	0.50	1.00	1.00	1.00	452.98
0.50	0.50	1.50	1.50	0.50	1.00	1.50	1.25	451.18
0.50	0.50	1.50	1.50	0.50	1.50	0.50	1.25	456.30
0.50	0.50	1.50	1.50	0.50	1.50	1.00	1.25	439.07
0.50	0.50	1.50	1.50	0.50	1.50	1.50	1.25	436.30
0.50	0.50	1.50	1.50	1.00	0.50	0.50	1.00	547.84
0.50	0.50	1.50	1.50	1.00	0.50	1.00	1.00	541.34
0.50	0.50	1.50	1.50	1.00	0.50	1.50	1.00	542.38
0.50	0.50	1.50	1.50	1.00	1.00	0.50	1.25	409.84
0.50	0.50	1.50	1.50	1.00	1.00	1.00	1.25	398.79
0.50	0.50	1.50	1.50	1.00	1.00	1.50	1.25	398.47
0.50	0.50	1.50	1.50	1.00	1.50	0.50	1.25	390.65
0.50	0.50	1.50	1.50	1.00	1.50	1.00	1.25	376.39
0.50	0.50	1.50	1.50	1.00	1.50	1.50	1.25	374.59
0.50	0.50	1.50	1.50	1.50	0.50	0.50	1.00	538.95
0.50	0.50	1.50	1.50	1.50	0.50	1.00	1.00	534.96
0.50	0.50	1.50	1.50	1.50	0.50	1.50	1.25	536.84
0.50	0.50	1.50	1.50	1.50	1.00	0.50	1.25	390.37
0.50	0.50	1.50	1.50	1.50	1.00	1.00	1.25	381.48
0.50	0.50	1.50	1.50	1.50	1.00	1.50	1.25	381.94
0.50	0.50	1.50	1.50	1.50	1.50	0.50	1.25	365.18
0.50	0.50	1.50	1.50	1.50	1.50	1.00	1.25	353.20
0.50	0.50	1.50	1.50	1.50	1.50	1.50	1.25	352.49
0.50	1.00	0.50	0.50	0.50	0.50	0.50	1.25	505.02
0.50	1.00	0.50	0.50	0.50	0.50	1.00	1.00	499.30
0.50	1.00	0.50	0.50	0.50	0.50	1.50	1.00	500.20
0.50	1.00	0.50	0.50	0.50	1.00	0.50	1.25	400.18
0.50	1.00	0.50	0.50	0.50	1.00	1.00	1.00	390.94
0.50	1.00	0.50	0.50	0.50	1.00	1.50	1.00	390.68
0.50	1.00	0.50	0.50	0.50	1.50	0.50	1.25	398.42
0.50	1.00	0.50	0.50	0.50	1.50	1.00	1.25	387.93
0.50	1.00	0.50	0.50	0.50	1.50	1.50	1.25	387.19
0.50	1.00	0.50	0.50	1.00	0.50	0.50	1.25	479.74

D.1. PROCURA DO ÓPTIMO PARA AS VÁRIAS COMBINAÇÕES DE  $F_{233}$

$x_2$	$x_3$	$x_5$	$x_6$	$x_8$	$x_9$	$x_{10}$	$x_1^*$	$ev_4$
0.50	1.00	0.50	0.50	1.00	0.50	1.00	1.25	477.09
0.50	1.00	0.50	0.50	1.00	0.50	1.50	1.00	479.10
0.50	1.00	0.50	0.50	1.00	1.00	0.50	1.25	363.30
0.50	1.00	0.50	0.50	1.00	1.00	1.00	1.25	358.45
0.50	1.00	0.50	0.50	1.00	1.00	1.50	1.25	359.84
0.50	1.00	0.50	0.50	1.00	1.50	0.50	1.25	356.53
0.50	1.00	0.50	0.50	1.00	1.50	1.00	1.25	350.08
0.50	1.00	0.50	0.50	1.00	1.50	1.50	1.25	350.66
0.50	1.00	0.50	0.50	1.50	0.50	0.50	1.25	481.43
0.50	1.00	0.50	0.50	1.50	0.50	1.00	1.25	480.28
0.50	1.00	0.50	0.50	1.50	0.50	1.50	1.25	482.85
0.50	1.00	0.50	0.50	1.50	1.00	0.50	1.25	360.57
0.50	1.00	0.50	0.50	1.50	1.00	1.00	1.25	356.55
0.50	1.00	0.50	0.50	1.50	1.00	1.50	1.25	358.34
0.50	1.00	0.50	0.50	1.50	1.50	0.50	1.25	350.56
0.50	1.00	0.50	0.50	1.50	1.50	1.00	1.25	345.08
0.50	1.00	0.50	0.50	1.50	1.50	1.50	1.25	346.19
0.50	1.00	0.50	1.00	0.50	0.50	0.50	1.25	478.99
0.50	1.00	0.50	1.00	0.50	0.50	1.00	1.25	471.58
0.50	1.00	0.50	1.00	0.50	0.50	1.50	1.25	472.05
0.50	1.00	0.50	1.00	0.50	1.00	0.50	1.25	362.15
0.50	1.00	0.50	1.00	0.50	1.00	1.00	1.25	349.88
0.50	1.00	0.50	1.00	0.50	1.00	1.50	1.25	348.88
0.50	1.00	0.50	1.00	0.50	1.50	0.50	1.25	353.11
0.50	1.00	0.50	1.00	0.50	1.50	1.00	1.25	339.32
0.50	1.00	0.50	1.00	0.50	1.50	1.50	1.25	337.66
0.50	1.00	0.50	1.00	1.00	0.50	0.50	1.25	445.81
0.50	1.00	0.50	1.00	1.00	0.50	1.00	1.25	441.62
0.50	1.00	0.50	1.00	1.00	0.50	1.50	1.25	443.25
0.50	1.00	0.50	1.00	1.00	1.00	0.50	1.25	316.42
0.50	1.00	0.50	1.00	1.00	1.00	1.00	1.25	308.87
0.50	1.00	0.50	1.00	1.00	1.00	1.50	1.25	309.58
0.50	1.00	0.50	1.00	1.00	1.50	0.50	1.25	301.12
0.50	1.00	0.50	1.00	1.00	1.50	1.00	1.25	291.38
0.50	1.00	0.50	1.00	1.00	1.50	1.50	1.25	291.08
0.50	1.00	0.50	1.00	1.50	0.50	0.50	1.25	444.34
0.50	1.00	0.50	1.00	1.50	0.50	1.00	1.25	441.91
0.50	1.00	0.50	1.00	1.50	0.50	1.50	1.25	444.22
0.50	1.00	0.50	1.00	1.50	1.00	0.50	1.25	308.17
0.50	1.00	0.50	1.00	1.50	1.00	1.00	1.25	302.23
0.50	1.00	0.50	1.00	1.50	1.00	1.50	1.25	303.63
0.50	1.00	0.50	1.00	1.50	1.50	0.50	1.25	288.77
0.50	1.00	0.50	1.00	1.50	1.50	1.00	1.25	281.11
0.50	1.00	0.50	1.00	1.50	1.50	1.50	1.25	281.67
0.50	1.00	0.50	1.50	0.50	0.50	0.50	1.25	488.91
0.50	1.00	0.50	1.50	0.50	0.50	1.00	1.25	481.16

$x_2$	$x_3$	$x_5$	$x_6$	$x_8$	$x_9$	$x_{10}$	$x_1^*$	$ev_4$
0.50	1.00	0.50	1.50	0.50	0.50	1.50	1.25	481.57
0.50	1.00	0.50	1.50	0.50	1.00	0.50	1.25	371.12
0.50	1.00	0.50	1.50	0.50	1.00	1.00	1.25	358.27
0.50	1.00	0.50	1.50	0.50	1.00	1.50	1.25	357.15
0.50	1.00	0.50	1.50	0.50	1.50	0.50	1.25	361.29
0.50	1.00	0.50	1.50	0.50	1.50	1.00	1.25	346.83
0.50	1.00	0.50	1.50	0.50	1.50	1.50	1.25	345.00
0.50	1.00	0.50	1.50	1.00	0.50	0.50	1.25	455.08
0.50	1.00	0.50	1.50	1.00	0.50	1.00	1.25	450.53
0.50	1.00	0.50	1.50	1.00	0.50	1.50	1.25	452.10
0.50	1.00	0.50	1.50	1.00	1.00	0.50	1.25	324.65
0.50	1.00	0.50	1.50	1.00	1.00	1.00	1.25	316.51
0.50	1.00	0.50	1.50	1.00	1.00	1.50	1.25	317.10
0.50	1.00	0.50	1.50	1.00	1.50	0.50	1.25	308.38
0.50	1.00	0.50	1.50	1.00	1.50	1.00	1.25	297.93
0.50	1.00	0.50	1.50	1.00	1.50	1.50	1.25	297.45
0.50	1.00	0.50	1.50	1.50	0.50	0.50	1.25	453.27
0.50	1.00	0.50	1.50	1.50	0.50	1.00	1.25	450.40
0.50	1.00	0.50	1.50	1.50	0.50	1.50	1.25	452.62
0.50	1.00	0.50	1.50	1.50	1.00	0.50	1.25	315.78
0.50	1.00	0.50	1.50	1.50	1.00	1.00	1.25	309.19
0.50	1.00	0.50	1.50	1.50	1.00	1.50	1.25	310.45
0.50	1.00	0.50	1.50	1.50	1.50	0.50	1.25	295.37
0.50	1.00	0.50	1.50	1.50	1.50	1.00	1.25	286.94
0.50	1.00	0.50	1.50	1.50	1.50	1.50	1.25	287.31
0.50	1.00	1.00	0.50	0.50	0.50	0.50	1.00	507.26
0.50	1.00	1.00	0.50	0.50	0.50	1.00	1.00	501.71
0.50	1.00	1.00	0.50	0.50	0.50	1.50	1.00	502.67
0.50	1.00	1.00	0.50	0.50	1.00	0.50	1.00	402.35
0.50	1.00	1.00	0.50	0.50	1.00	1.00	1.00	393.31
0.50	1.00	1.00	0.50	0.50	1.00	1.50	1.00	393.13
0.50	1.00	1.00	0.50	0.50	1.50	0.50	1.25	400.60
0.50	1.00	1.00	0.50	0.50	1.50	1.00	1.25	390.35
0.50	1.00	1.00	0.50	0.50	1.50	1.50	1.25	389.67
0.50	1.00	1.00	0.50	1.00	0.50	0.50	1.00	481.91
0.50	1.00	1.00	0.50	1.00	0.50	1.00	1.00	479.49
0.50	1.00	1.00	0.50	1.00	0.50	1.50	1.00	481.56
0.50	1.00	1.00	0.50	1.00	1.00	0.50	1.25	365.42
0.50	1.00	1.00	0.50	1.00	1.00	1.00	1.25	360.85
0.50	1.00	1.00	0.50	1.00	1.00	1.50	1.25	362.31
0.50	1.00	1.00	0.50	1.00	1.50	0.50	1.25	358.55
0.50	1.00	1.00	0.50	1.00	1.50	1.00	1.25	352.46
0.50	1.00	1.00	0.50	1.00	1.50	1.50	1.25	353.13
0.50	1.00	1.00	0.50	1.50	0.50	0.50	1.00	483.62
0.50	1.00	1.00	0.50	1.50	0.50	1.00	1.00	482.69
0.50	1.00	1.00	0.50	1.50	0.50	1.50	1.00	485.31

D.1. PROCURA DO ÓPTIMO PARA AS VÁRIAS COMBINAÇÕES DE F235

$x_2$	$x_3$	$x_5$	$x_6$	$x_8$	$x_9$	$x_{10}$	$x_1^*$	$ev_4$
0.50	1.00	1.00	0.50	1.50	1.00	0.50	1.25	362.69
0.50	1.00	1.00	0.50	1.50	1.00	1.00	1.25	358.95
0.50	1.00	1.00	0.50	1.50	1.00	1.50	1.25	360.80
0.50	1.00	1.00	0.50	1.50	1.50	0.50	1.25	352.59
0.50	1.00	1.00	0.50	1.50	1.50	1.00	1.25	347.46
0.50	1.00	1.00	0.50	1.50	1.50	1.50	1.25	348.65
0.50	1.00	1.00	1.00	0.50	0.50	0.50	1.25	481.33
0.50	1.00	1.00	1.00	0.50	0.50	1.00	1.25	474.04
0.50	1.00	1.00	1.00	0.50	0.50	1.50	1.25	474.54
0.50	1.00	1.00	1.00	0.50	1.00	0.50	1.25	364.41
0.50	1.00	1.00	1.00	0.50	1.00	1.00	1.25	352.31
0.50	1.00	1.00	1.00	0.50	1.00	1.50	1.25	351.36
0.50	1.00	1.00	1.00	0.50	1.50	0.50	1.25	355.29
0.50	1.00	1.00	1.00	0.50	1.50	1.00	1.25	341.74
0.50	1.00	1.00	1.00	0.50	1.50	1.50	1.25	340.13
0.50	1.00	1.00	1.00	1.00	0.50	0.50	1.25	448.06
0.50	1.00	1.00	1.00	1.00	0.50	1.00	1.25	444.05
0.50	1.00	1.00	1.00	1.00	0.50	1.50	1.25	445.73
0.50	1.00	1.00	1.00	1.00	1.00	0.50	1.25	318.55
0.50	1.00	1.00	1.00	1.00	1.00	1.00	1.25	311.27
0.50	1.00	1.00	1.00	1.00	1.00	1.50	1.25	312.05
0.50	1.00	1.00	1.00	1.00	1.50	0.50	1.25	303.15
0.50	1.00	1.00	1.00	1.00	1.50	1.00	1.25	293.76
0.50	1.00	1.00	1.00	1.00	1.50	1.50	1.25	293.55
0.50	1.00	1.00	1.00	1.50	0.50	0.50	1.25	446.58
0.50	1.00	1.00	1.00	1.50	0.50	1.00	1.25	444.34
0.50	1.00	1.00	1.00	1.50	0.50	1.50	1.25	446.70
0.50	1.00	1.00	1.00	1.50	1.00	0.50	1.25	310.29
0.50	1.00	1.00	1.00	1.50	1.00	1.00	1.25	304.63
0.50	1.00	1.00	1.00	1.50	1.00	1.50	1.25	306.10
0.50	1.00	1.00	1.00	1.50	1.50	0.50	1.25	290.79
0.50	1.00	1.00	1.00	1.50	1.50	1.00	1.25	283.49
0.50	1.00	1.00	1.00	1.50	1.50	1.50	1.25	284.14
0.50	1.00	1.00	1.50	0.50	0.50	0.50	1.00	491.12
0.50	1.00	1.00	1.50	0.50	0.50	1.00	1.00	483.58
0.50	1.00	1.00	1.50	0.50	0.50	1.50	1.00	484.04
0.50	1.00	1.00	1.50	0.50	1.00	0.50	1.25	373.28
0.50	1.00	1.00	1.50	0.50	1.00	1.00	1.25	360.68
0.50	1.00	1.00	1.50	0.50	1.00	1.50	1.25	359.62
0.50	1.00	1.00	1.50	0.50	1.50	0.50	1.25	363.37
0.50	1.00	1.00	1.50	0.50	1.50	1.00	1.25	349.22
0.50	1.00	1.00	1.50	0.50	1.50	1.50	1.25	347.47
0.50	1.00	1.00	1.50	1.00	0.50	0.50	1.25	457.23
0.50	1.00	1.00	1.50	1.00	0.50	1.00	1.25	452.94
0.50	1.00	1.00	1.50	1.00	0.50	1.50	1.25	454.57
0.50	1.00	1.00	1.50	1.00	1.00	0.50	1.25	326.64

$x_2$	$x_3$	$x_5$	$x_6$	$x_8$	$x_9$	$x_{10}$	$x_1^*$	$ev_4$
0.50	1.00	1.00	1.50	1.00	1.00	1.00	1.25	318.88
0.50	1.00	1.00	1.50	1.00	1.00	1.50	1.25	319.55
0.50	1.00	1.00	1.50	1.00	1.50	0.50	1.25	310.25
0.50	1.00	1.00	1.50	1.00	1.50	1.00	1.25	300.27
0.50	1.00	1.00	1.50	1.00	1.50	1.50	1.25	299.91
0.50	1.00	1.00	1.50	1.50	0.50	0.50	1.25	455.42
0.50	1.00	1.00	1.50	1.50	0.50	1.00	1.25	452.81
0.50	1.00	1.00	1.50	1.50	0.50	1.50	1.25	455.09
0.50	1.00	1.00	1.50	1.50	1.00	0.50	1.25	317.77
0.50	1.00	1.00	1.50	1.50	1.00	1.00	1.25	311.55
0.50	1.00	1.00	1.50	1.50	1.00	1.50	1.25	312.91
0.50	1.00	1.00	1.50	1.50	1.50	0.50	1.25	297.23
0.50	1.00	1.00	1.50	1.50	1.50	1.00	1.25	289.28
0.50	1.00	1.00	1.50	1.50	1.50	1.50	1.25	289.77
0.50	1.00	1.50	0.50	0.50	0.50	0.50	1.00	509.76
0.50	1.00	1.50	0.50	0.50	0.50	1.00	1.00	504.21
0.50	1.00	1.50	0.50	0.50	0.50	1.50	1.00	505.17
0.50	1.00	1.50	0.50	0.50	1.00	0.50	1.00	404.85
0.50	1.00	1.50	0.50	0.50	1.00	1.00	1.00	395.81
0.50	1.00	1.50	0.50	0.50	1.00	1.50	1.00	395.63
0.50	1.00	1.50	0.50	0.50	1.50	0.50	1.25	403.10
0.50	1.00	1.50	0.50	0.50	1.50	1.00	1.25	392.85
0.50	1.00	1.50	0.50	0.50	1.50	1.50	1.25	392.17
0.50	1.00	1.50	0.50	1.00	0.50	0.50	1.00	484.41
0.50	1.00	1.50	0.50	1.00	0.50	1.00	1.00	481.99
0.50	1.00	1.50	0.50	1.00	0.50	1.50	1.00	484.06
0.50	1.00	1.50	0.50	1.00	1.00	0.50	1.25	367.92
0.50	1.00	1.50	0.50	1.00	1.00	1.00	1.25	363.35
0.50	1.00	1.50	0.50	1.00	1.00	1.50	1.25	364.81
0.50	1.00	1.50	0.50	1.00	1.50	0.50	1.25	361.05
0.50	1.00	1.50	0.50	1.00	1.50	1.00	1.25	354.96
0.50	1.00	1.50	0.50	1.00	1.50	1.50	1.25	355.63
0.50	1.00	1.50	0.50	1.50	0.50	0.50	1.00	486.12
0.50	1.00	1.50	0.50	1.50	0.50	1.00	1.00	485.19
0.50	1.00	1.50	0.50	1.50	0.50	1.50	1.00	487.81
0.50	1.00	1.50	0.50	1.50	1.00	0.50	1.25	365.19
0.50	1.00	1.50	0.50	1.50	1.00	1.00	1.25	361.45
0.50	1.00	1.50	0.50	1.50	1.00	1.50	1.25	363.30
0.50	1.00	1.50	0.50	1.50	1.50	0.50	1.25	355.09
0.50	1.00	1.50	0.50	1.50	1.50	1.00	1.25	349.96
0.50	1.00	1.50	0.50	1.50	1.50	1.50	1.25	351.15
0.50	1.00	1.50	1.00	0.50	0.50	0.50	1.25	483.83
0.50	1.00	1.50	1.00	0.50	0.50	1.00	1.25	476.54
0.50	1.00	1.50	1.00	0.50	0.50	1.50	1.25	477.04
0.50	1.00	1.50	1.00	0.50	1.00	0.50	1.25	366.91
0.50	1.00	1.50	1.00	0.50	1.00	1.00	1.25	354.81

D.1. PROCURA DO ÓPTIMO PARA AS VÁRIAS COMBINAÇÕES DE F237

$x_2$	$x_3$	$x_5$	$x_6$	$x_8$	$x_9$	$x_{10}$	$x_1^*$	ev <sub>4</sub>
0.50	1.00	1.50	1.00	0.50	1.00	1.50	1.25	353.86
0.50	1.00	1.50	1.00	0.50	1.50	0.50	1.25	357.79
0.50	1.00	1.50	1.00	0.50	1.50	1.00	1.25	344.24
0.50	1.00	1.50	1.00	0.50	1.50	1.50	1.25	342.63
0.50	1.00	1.50	1.00	1.00	0.50	0.50	1.25	450.56
0.50	1.00	1.50	1.00	1.00	0.50	1.00	1.25	446.55
0.50	1.00	1.50	1.00	1.00	0.50	1.50	1.25	448.23
0.50	1.00	1.50	1.00	1.00	1.00	0.50	1.25	321.05
0.50	1.00	1.50	1.00	1.00	1.00	1.00	1.25	313.77
0.50	1.00	1.50	1.00	1.00	1.00	1.50	1.25	314.55
0.50	1.00	1.50	1.00	1.00	1.50	0.50	1.25	305.65
0.50	1.00	1.50	1.00	1.00	1.50	1.00	1.25	296.26
0.50	1.00	1.50	1.00	1.00	1.50	1.50	1.25	296.05
0.50	1.00	1.50	1.00	1.50	0.50	0.50	1.25	449.08
0.50	1.00	1.50	1.00	1.50	0.50	1.00	1.25	446.84
0.50	1.00	1.50	1.00	1.50	0.50	1.50	1.25	449.20
0.50	1.00	1.50	1.00	1.50	1.00	0.50	1.25	312.79
0.50	1.00	1.50	1.00	1.50	1.00	1.00	1.25	307.13
0.50	1.00	1.50	1.00	1.50	1.00	1.50	1.25	308.60
0.50	1.00	1.50	1.00	1.50	1.50	0.50	1.25	293.29
0.50	1.00	1.50	1.00	1.50	1.50	1.00	1.25	285.99
0.50	1.00	1.50	1.00	1.50	1.50	1.50	1.25	286.64
0.50	1.00	1.50	1.50	0.50	0.50	0.50	1.00	493.62
0.50	1.00	1.50	1.50	0.50	0.50	1.00	1.00	486.08
0.50	1.00	1.50	1.50	0.50	0.50	1.50	1.00	486.54
0.50	1.00	1.50	1.50	0.50	1.00	0.50	1.25	375.78
0.50	1.00	1.50	1.50	0.50	1.00	1.00	1.25	363.18
0.50	1.00	1.50	1.50	0.50	1.00	1.50	1.25	362.12
0.50	1.00	1.50	1.50	0.50	1.50	0.50	1.25	365.87
0.50	1.00	1.50	1.50	0.50	1.50	1.00	1.25	351.72
0.50	1.00	1.50	1.50	0.50	1.50	1.50	1.25	349.97
0.50	1.00	1.50	1.50	1.00	0.50	0.50	1.25	459.73
0.50	1.00	1.50	1.50	1.00	0.50	1.00	1.25	455.44
0.50	1.00	1.50	1.50	1.00	0.50	1.50	1.25	457.07
0.50	1.00	1.50	1.50	1.00	1.00	0.50	1.25	329.14
0.50	1.00	1.50	1.50	1.00	1.00	1.00	1.25	321.38
0.50	1.00	1.50	1.50	1.00	1.00	1.50	1.25	322.05
0.50	1.00	1.50	1.50	1.00	1.50	0.50	1.25	312.75
0.50	1.00	1.50	1.50	1.00	1.50	1.00	1.25	302.77
0.50	1.00	1.50	1.50	1.00	1.50	1.50	1.25	302.41
0.50	1.00	1.50	1.50	1.50	0.50	0.50	1.25	457.92
0.50	1.00	1.50	1.50	1.50	0.50	1.00	1.25	455.31
0.50	1.00	1.50	1.50	1.50	0.50	1.50	1.25	457.59
0.50	1.00	1.50	1.50	1.50	1.00	0.50	1.25	320.27
0.50	1.00	1.50	1.50	1.50	1.00	1.00	1.25	314.05
0.50	1.00	1.50	1.50	1.50	1.00	1.50	1.25	315.41

$x_2$	$x_3$	$x_5$	$x_6$	$x_8$	$x_9$	$x_{10}$	$x_{11}^*$	$ev_4$
0.50	1.00	1.50	1.50	1.50	1.50	0.50	1.25	299.73
0.50	1.00	1.50	1.50	1.50	1.50	1.00	1.25	291.78
0.50	1.00	1.50	1.50	1.50	1.50	1.50	1.25	292.27
0.50	1.50	0.50	0.50	0.50	0.50	0.50	1.25	487.42
0.50	1.50	0.50	0.50	0.50	0.50	1.00	1.00	482.72
0.50	1.50	0.50	0.50	0.50	0.50	1.50	1.00	483.77
0.50	1.50	0.50	0.50	0.50	1.00	0.50	1.25	387.22
0.50	1.50	0.50	0.50	0.50	1.00	1.00	1.25	379.48
0.50	1.50	0.50	0.50	0.50	1.00	1.50	1.00	379.51
0.50	1.50	0.50	0.50	0.50	1.50	0.50	1.25	387.26
0.50	1.50	0.50	0.50	0.50	1.50	1.00	1.25	378.17
0.50	1.50	0.50	0.50	0.50	1.50	1.50	1.25	377.85
0.50	1.50	0.50	0.50	1.00	0.50	0.50	1.25	465.45
0.50	1.50	0.50	0.50	1.00	0.50	1.00	1.25	464.49
0.50	1.50	0.50	0.50	1.00	0.50	1.50	1.25	466.84
0.50	1.50	0.50	0.50	1.00	1.00	0.50	1.25	354.17
0.50	1.50	0.50	0.50	1.00	1.00	1.00	1.25	351.54
0.50	1.50	0.50	0.50	1.00	1.00	1.50	1.25	353.42
0.50	1.50	0.50	0.50	1.00	1.50	0.50	1.25	348.08
0.50	1.50	0.50	0.50	1.00	1.50	1.00	1.25	344.11
0.50	1.50	0.50	0.50	1.00	1.50	1.50	1.25	345.46
0.50	1.50	0.50	0.50	1.50	0.50	0.50	1.25	472.55
0.50	1.50	0.50	0.50	1.50	0.50	1.00	1.25	472.33
0.50	1.50	0.50	0.50	1.50	0.50	1.50	1.25	474.97
0.50	1.50	0.50	0.50	1.50	1.00	0.50	1.25	359.06
0.50	1.50	0.50	0.50	1.50	1.00	1.00	1.25	356.84
0.50	1.50	0.50	0.50	1.50	1.00	1.50	1.25	358.92
0.50	1.50	0.50	0.50	1.50	1.50	0.50	1.25	351.35
0.50	1.50	0.50	0.50	1.50	1.50	1.00	1.25	347.86
0.50	1.50	0.50	0.50	1.50	1.50	1.50	1.25	349.47
0.50	1.50	0.50	1.00	0.50	0.50	0.50	1.25	458.26
0.50	1.50	0.50	1.00	0.50	0.50	1.00	1.25	451.58
0.50	1.50	0.50	1.00	0.50	0.50	1.50	1.25	452.08
0.50	1.50	0.50	1.00	0.50	1.00	0.50	1.25	344.76
0.50	1.50	0.50	1.00	0.50	1.00	1.00	1.25	333.97
0.50	1.50	0.50	1.00	0.50	1.00	1.50	1.25	333.21
0.50	1.50	0.50	1.00	0.50	1.50	0.50	1.25	338.34
0.50	1.50	0.50	1.00	0.50	1.50	1.00	1.25	326.13
0.50	1.50	0.50	1.00	0.50	1.50	1.50	1.25	324.87
0.50	1.50	0.50	1.00	1.00	0.50	0.50	1.25	426.65
0.50	1.50	0.50	1.00	1.00	0.50	1.00	1.25	424.61
0.50	1.50	0.50	1.00	1.00	0.50	1.50	1.25	426.71
0.50	1.50	0.50	1.00	1.00	1.00	0.50	1.25	300.92
0.50	1.50	0.50	1.00	1.00	1.00	1.00	1.25	296.52
0.50	1.50	0.50	1.00	1.00	1.00	1.50	1.25	297.97
0.50	1.50	0.50	1.00	1.00	1.50	0.50	1.25	286.99



D.1. PROCURA DO ÓPTIMO PARA AS VÁRIAS COMBINAÇÕES DE  $\mathcal{F}239$

$x_2$	$x_3$	$x_5$	$x_6$	$x_8$	$x_9$	$x_{10}$	$x_1^*$	$ev_4$
0.50	1.50	0.50	1.00	1.00	1.50	1.00	1.25	280.85
0.50	1.50	0.50	1.00	1.00	1.50	1.50	1.25	281.62
0.50	1.50	0.50	1.00	1.50	0.50	0.50	1.25	432.16
0.50	1.50	0.50	1.00	1.50	0.50	1.00	1.25	431.01
0.50	1.50	0.50	1.00	1.50	0.50	1.50	1.25	433.44
0.50	1.50	0.50	1.00	1.50	1.00	0.50	1.25	303.04
0.50	1.50	0.50	1.00	1.50	1.00	1.00	1.25	299.45
0.50	1.50	0.50	1.00	1.50	1.00	1.50	1.25	301.24
0.50	1.50	0.50	1.00	1.50	1.50	0.50	1.25	287.06
0.50	1.50	0.50	1.00	1.50	1.50	1.00	1.25	281.96
0.50	1.50	0.50	1.00	1.50	1.50	1.50	1.25	283.16
0.50	1.50	0.50	1.50	0.50	0.50	0.50	1.25	467.94
0.50	1.50	0.50	1.50	0.50	0.50	1.00	1.25	460.94
0.50	1.50	0.50	1.50	0.50	0.50	1.50	1.25	461.37
0.50	1.50	0.50	1.50	0.50	1.00	0.50	1.25	353.50
0.50	1.50	0.50	1.50	0.50	1.00	1.00	1.25	342.20
0.50	1.50	0.50	1.50	0.50	1.00	1.50	1.25	341.32
0.50	1.50	0.50	1.50	0.50	1.50	0.50	1.25	346.07
0.50	1.50	0.50	1.50	0.50	1.50	1.00	1.25	333.28
0.50	1.50	0.50	1.50	0.50	1.50	1.50	1.25	331.88
0.50	1.50	0.50	1.50	1.00	0.50	0.50	1.25	435.35
0.50	1.50	0.50	1.50	1.00	0.50	1.00	1.25	432.98
0.50	1.50	0.50	1.50	1.00	0.50	1.50	1.25	435.01
0.50	1.50	0.50	1.50	1.00	1.00	0.50	1.25	308.55
0.50	1.50	0.50	1.50	1.00	1.00	1.00	1.25	303.64
0.50	1.50	0.50	1.50	1.00	1.00	1.50	1.25	304.97
0.50	1.50	0.50	1.50	1.00	1.50	0.50	1.25	293.34
0.50	1.50	0.50	1.50	1.00	1.50	1.00	1.25	286.57
0.50	1.50	0.50	1.50	1.00	1.50	1.50	1.25	287.18
0.50	1.50	0.50	1.50	1.50	0.50	0.50	1.25	440.69
0.50	1.50	0.50	1.50	1.50	0.50	1.00	1.25	439.17
0.50	1.50	0.50	1.50	1.50	0.50	1.50	1.25	441.52
0.50	1.50	0.50	1.50	1.50	1.00	0.50	1.25	310.36
0.50	1.50	0.50	1.50	1.50	1.00	1.00	1.25	306.23
0.50	1.50	0.50	1.50	1.50	1.00	1.50	1.25	307.90
0.50	1.50	0.50	1.50	1.50	1.50	0.50	1.25	293.08
0.50	1.50	0.50	1.50	1.50	1.50	1.00	1.25	287.32
0.50	1.50	0.50	1.50	1.50	1.50	1.50	1.25	288.36
0.50	1.50	1.00	0.50	0.50	0.50	0.50	1.00	489.58
0.50	1.50	1.00	0.50	0.50	0.50	1.00	1.00	485.09
0.50	1.50	1.00	0.50	0.50	0.50	1.50	1.00	486.23
0.50	1.50	1.00	0.50	0.50	1.00	0.50	1.00	389.28
0.50	1.50	1.00	0.50	0.50	1.00	1.00	1.00	381.80
0.50	1.50	1.00	0.50	0.50	1.00	1.50	1.00	381.95
0.50	1.50	1.00	0.50	0.50	1.50	0.50	1.25	389.28
0.50	1.50	1.00	0.50	0.50	1.50	1.00	1.25	380.55

$x_2$	$x_3$	$x_5$	$x_6$	$x_8$	$x_9$	$x_{10}$	$x_{11}^*$	$ev_4$
0.50	1.50	1.00	0.50	0.50	1.50	1.50	1.25	380.32
0.50	1.50	1.00	0.50	1.00	0.50	0.50	1.00	467.52
0.50	1.50	1.00	0.50	1.00	0.50	1.00	1.00	466.89
0.50	1.50	1.00	0.50	1.00	0.50	1.50	1.25	469.31
0.50	1.50	1.00	0.50	1.00	1.00	0.50	1.25	356.11
0.50	1.50	1.00	0.50	1.00	1.00	1.00	1.25	353.89
0.50	1.50	1.00	0.50	1.00	1.00	1.50	1.25	355.88
0.50	1.50	1.00	0.50	1.00	1.50	0.50	1.25	349.87
0.50	1.50	1.00	0.50	1.00	1.50	1.00	1.25	346.43
0.50	1.50	1.00	0.50	1.00	1.50	1.50	1.25	347.91
0.50	1.50	1.00	0.50	1.50	0.50	0.50	1.00	474.63
0.50	1.50	1.00	0.50	1.50	0.50	1.00	1.25	474.73
0.50	1.50	1.00	0.50	1.50	0.50	1.50	1.25	477.44
0.50	1.50	1.00	0.50	1.50	1.00	0.50	1.25	360.99
0.50	1.50	1.00	0.50	1.50	1.00	1.00	1.25	359.19
0.50	1.50	1.00	0.50	1.50	1.00	1.50	1.25	361.37
0.50	1.50	1.00	0.50	1.50	1.50	0.50	1.25	353.14
0.50	1.50	1.00	0.50	1.50	1.50	1.00	1.25	350.18
0.50	1.50	1.00	0.50	1.50	1.50	1.50	1.25	351.93
0.50	1.50	1.00	1.00	0.50	0.50	0.50	1.25	460.52
0.50	1.50	1.00	1.00	0.50	0.50	1.00	1.25	454.02
0.50	1.50	1.00	1.00	0.50	0.50	1.50	1.25	454.56
0.50	1.50	1.00	1.00	0.50	1.00	0.50	1.25	346.91
0.50	1.50	1.00	1.00	0.50	1.00	1.00	1.25	336.37
0.50	1.50	1.00	1.00	0.50	1.00	1.50	1.25	335.67
0.50	1.50	1.00	1.00	0.50	1.50	0.50	1.25	340.37
0.50	1.50	1.00	1.00	0.50	1.50	1.00	1.25	328.51
0.50	1.50	1.00	1.00	0.50	1.50	1.50	1.25	327.34
0.50	1.50	1.00	1.00	1.00	0.50	0.50	1.25	428.77
0.50	1.50	1.00	1.00	1.00	0.50	1.00	1.25	427.01
0.50	1.50	1.00	1.00	1.00	0.50	1.50	1.25	429.18
0.50	1.50	1.00	1.00	1.00	1.00	0.50	1.25	302.85
0.50	1.50	1.00	1.00	1.00	1.00	1.00	1.25	298.87
0.50	1.50	1.00	1.00	1.00	1.00	1.50	1.25	300.42
0.50	1.50	1.00	1.00	1.00	1.50	0.50	1.25	288.78
0.50	1.50	1.00	1.00	1.00	1.50	1.00	1.25	283.16
0.50	1.50	1.00	1.00	1.00	1.50	1.50	1.25	284.07
0.50	1.50	1.00	1.00	1.50	0.50	0.50	1.25	434.28
0.50	1.50	1.00	1.00	1.50	0.50	1.00	1.25	433.41
0.50	1.50	1.00	1.00	1.50	0.50	1.50	1.25	435.91
0.50	1.50	1.00	1.00	1.50	1.00	0.50	1.25	304.98
0.50	1.50	1.00	1.00	1.50	1.00	1.00	1.25	301.80
0.50	1.50	1.00	1.00	1.50	1.00	1.50	1.25	303.69
0.50	1.50	1.00	1.00	1.50	1.50	0.50	1.25	288.85
0.50	1.50	1.00	1.00	1.50	1.50	1.00	1.25	284.28
0.50	1.50	1.00	1.00	1.50	1.50	1.50	1.25	285.61

D.1. PROCURA DO ÓPTIMO PARA AS VÁRIAS COMBINAÇÕES DE  $\mathcal{F}241$

$x_2$	$x_3$	$x_5$	$x_6$	$x_8$	$x_9$	$x_{10}$	$x_1^*$	$ev_4$
0.50	1.50	1.00	1.50	0.50	0.50	0.50	1.00	470.05
0.50	1.50	1.00	1.50	0.50	0.50	1.00	1.25	463.35
0.50	1.50	1.00	1.50	0.50	0.50	1.50	1.25	463.85
0.50	1.50	1.00	1.50	0.50	1.00	0.50	1.25	355.50
0.50	1.50	1.00	1.50	0.50	1.00	1.00	1.25	344.57
0.50	1.50	1.00	1.50	0.50	1.00	1.50	1.25	343.78
0.50	1.50	1.00	1.50	0.50	1.50	0.50	1.25	347.93
0.50	1.50	1.00	1.50	0.50	1.50	1.00	1.25	335.62
0.50	1.50	1.00	1.50	0.50	1.50	1.50	1.25	334.34
0.50	1.50	1.00	1.50	1.00	0.50	0.50	1.25	437.33
0.50	1.50	1.00	1.50	1.00	0.50	1.00	1.25	435.35
0.50	1.50	1.00	1.50	1.00	0.50	1.50	1.25	437.47
0.50	1.50	1.00	1.50	1.00	1.00	0.50	1.25	310.29
0.50	1.50	1.00	1.50	1.00	1.00	1.00	1.25	305.94
0.50	1.50	1.00	1.50	1.00	1.00	1.50	1.25	307.41
0.50	1.50	1.00	1.50	1.00	1.50	0.50	1.25	294.89
0.50	1.50	1.00	1.50	1.00	1.50	1.00	1.25	288.82
0.50	1.50	1.00	1.50	1.00	1.50	1.50	1.25	289.62
0.50	1.50	1.00	1.50	1.50	0.50	0.50	1.25	442.67
0.50	1.50	1.00	1.50	1.50	0.50	1.00	1.25	441.53
0.50	1.50	1.00	1.50	1.50	0.50	1.50	1.25	443.98
0.50	1.50	1.00	1.50	1.50	1.00	0.50	1.25	312.11
0.50	1.50	1.00	1.50	1.50	1.00	1.00	1.25	308.53
0.50	1.50	1.00	1.50	1.50	1.00	1.50	1.25	310.34
0.50	1.50	1.00	1.50	1.50	1.50	0.50	1.25	294.63
0.50	1.50	1.00	1.50	1.50	1.50	1.00	1.25	289.58
0.50	1.50	1.00	1.50	1.50	1.50	1.50	1.25	290.80
0.50	1.50	1.50	0.50	0.50	0.50	0.50	1.00	492.08
0.50	1.50	1.50	0.50	0.50	0.50	1.00	1.00	487.59
0.50	1.50	1.50	0.50	0.50	0.50	1.50	1.00	488.73
0.50	1.50	1.50	0.50	0.50	1.00	0.50	1.00	391.78
0.50	1.50	1.50	0.50	0.50	1.00	1.00	1.00	384.30
0.50	1.50	1.50	0.50	0.50	1.00	1.50	1.00	384.45
0.50	1.50	1.50	0.50	0.50	1.50	0.50	1.25	391.78
0.50	1.50	1.50	0.50	0.50	1.50	1.00	1.25	383.05
0.50	1.50	1.50	0.50	0.50	1.50	1.50	1.25	382.82
0.50	1.50	1.50	0.50	1.00	0.50	0.50	1.00	470.02
0.50	1.50	1.50	0.50	1.00	0.50	1.00	1.00	469.39
0.50	1.50	1.50	0.50	1.00	0.50	1.50	1.25	471.81
0.50	1.50	1.50	0.50	1.00	1.00	0.50	1.25	358.61
0.50	1.50	1.50	0.50	1.00	1.00	1.00	1.25	356.39
0.50	1.50	1.50	0.50	1.00	1.00	1.50	1.25	358.38
0.50	1.50	1.50	0.50	1.00	1.50	0.50	1.25	352.37
0.50	1.50	1.50	0.50	1.00	1.50	1.00	1.25	348.93
0.50	1.50	1.50	0.50	1.00	1.50	1.50	1.25	350.41
0.50	1.50	1.50	0.50	1.50	0.50	0.50	1.00	477.13

$x_2$	$x_3$	$x_5$	$x_6$	$x_8$	$x_9$	$x_{10}$	$x_1^*$	$ev_4$
0.50	1.50	1.50	0.50	1.50	0.50	1.00	1.25	477.23
0.50	1.50	1.50	0.50	1.50	0.50	1.50	1.25	479.94
0.50	1.50	1.50	0.50	1.50	1.00	0.50	1.25	363.49
0.50	1.50	1.50	0.50	1.50	1.00	1.00	1.25	361.69
0.50	1.50	1.50	0.50	1.50	1.00	1.50	1.25	363.87
0.50	1.50	1.50	0.50	1.50	1.50	0.50	1.25	355.64
0.50	1.50	1.50	0.50	1.50	1.50	1.00	1.25	352.68
0.50	1.50	1.50	0.50	1.50	1.50	1.50	1.25	354.43
0.50	1.50	1.50	1.00	0.50	0.50	0.50	1.25	463.02
0.50	1.50	1.50	1.00	0.50	0.50	1.00	1.25	456.52
0.50	1.50	1.50	1.00	0.50	0.50	1.50	1.25	457.06
0.50	1.50	1.50	1.00	0.50	1.00	0.50	1.25	349.41
0.50	1.50	1.50	1.00	0.50	1.00	1.00	1.25	338.87
0.50	1.50	1.50	1.00	0.50	1.00	1.50	1.25	338.17
0.50	1.50	1.50	1.00	0.50	1.50	0.50	1.25	342.87
0.50	1.50	1.50	1.00	0.50	1.50	1.00	1.25	331.01
0.50	1.50	1.50	1.00	0.50	1.50	1.50	1.25	329.84
0.50	1.50	1.50	1.00	1.00	0.50	0.50	1.25	431.27
0.50	1.50	1.50	1.00	1.00	0.50	1.00	1.25	429.51
0.50	1.50	1.50	1.00	1.00	0.50	1.50	1.25	431.68
0.50	1.50	1.50	1.00	1.00	1.00	0.50	1.25	305.35
0.50	1.50	1.50	1.00	1.00	1.00	1.00	1.25	301.37
0.50	1.50	1.50	1.00	1.00	1.00	1.50	1.25	302.92
0.50	1.50	1.50	1.00	1.00	1.50	0.50	1.25	291.28
0.50	1.50	1.50	1.00	1.00	1.50	1.00	1.25	285.66
0.50	1.50	1.50	1.00	1.00	1.50	1.50	1.25	286.57
0.50	1.50	1.50	1.00	1.50	0.50	0.50	1.25	436.78
0.50	1.50	1.50	1.00	1.50	0.50	1.00	1.25	435.91
0.50	1.50	1.50	1.00	1.50	0.50	1.50	1.25	438.41
0.50	1.50	1.50	1.00	1.50	1.00	0.50	1.25	307.48
0.50	1.50	1.50	1.00	1.50	1.00	1.00	1.25	304.30
0.50	1.50	1.50	1.00	1.50	1.00	1.50	1.25	306.19
0.50	1.50	1.50	1.00	1.50	1.50	0.50	1.25	291.35
0.50	1.50	1.50	1.00	1.50	1.50	1.00	1.25	286.78
0.50	1.50	1.50	1.00	1.50	1.50	1.50	1.25	288.11
0.50	1.50	1.50	1.50	0.50	0.50	0.50	1.00	472.55
0.50	1.50	1.50	1.50	0.50	0.50	1.00	1.25	465.85
0.50	1.50	1.50	1.50	0.50	0.50	1.50	1.25	466.35
0.50	1.50	1.50	1.50	0.50	1.00	0.50	1.25	358.00
0.50	1.50	1.50	1.50	0.50	1.00	1.00	1.25	347.07
0.50	1.50	1.50	1.50	0.50	1.00	1.50	1.25	346.28
0.50	1.50	1.50	1.50	0.50	1.50	0.50	1.25	350.43
0.50	1.50	1.50	1.50	0.50	1.50	1.00	1.25	338.12
0.50	1.50	1.50	1.50	0.50	1.50	1.50	1.25	336.84
0.50	1.50	1.50	1.50	1.00	0.50	0.50	1.25	439.83
0.50	1.50	1.50	1.50	1.00	0.50	1.00	1.25	437.85

D.1. PROCURA DO ÓPTIMO PARA AS VÁRIAS COMBINAÇÕES DE  $\mathcal{F}243$

$x_2$	$x_3$	$x_5$	$x_6$	$x_8$	$x_9$	$x_{10}$	$x_1^*$	$ev_4$
0.50	1.50	1.50	1.50	1.00	0.50	1.50	1.25	439.97
0.50	1.50	1.50	1.50	1.00	1.00	0.50	1.25	312.79
0.50	1.50	1.50	1.50	1.00	1.00	1.00	1.25	308.44
0.50	1.50	1.50	1.50	1.00	1.00	1.50	1.25	309.91
0.50	1.50	1.50	1.50	1.00	1.50	0.50	1.25	297.39
0.50	1.50	1.50	1.50	1.00	1.50	1.00	1.25	291.32
0.50	1.50	1.50	1.50	1.00	1.50	1.50	1.25	292.12
0.50	1.50	1.50	1.50	1.50	0.50	0.50	1.25	445.17
0.50	1.50	1.50	1.50	1.50	0.50	1.00	1.25	444.03
0.50	1.50	1.50	1.50	1.50	0.50	1.50	1.25	446.48
0.50	1.50	1.50	1.50	1.50	1.00	0.50	1.25	314.61
0.50	1.50	1.50	1.50	1.50	1.00	1.00	1.25	311.03
0.50	1.50	1.50	1.50	1.50	1.00	1.50	1.25	312.84
0.50	1.50	1.50	1.50	1.50	1.50	0.50	1.25	297.13
0.50	1.50	1.50	1.50	1.50	1.50	1.00	1.25	292.08
0.50	1.50	1.50	1.50	1.50	1.50	1.50	1.25	293.30
1.00	0.50	0.50	0.50	0.50	0.50	0.50	1.00	593.86
1.00	0.50	0.50	0.50	0.50	0.50	1.00	1.00	586.12
1.00	0.50	0.50	0.50	0.50	0.50	1.50	1.00	586.48
1.00	0.50	0.50	0.50	0.50	1.00	0.50	1.00	481.66
1.00	0.50	0.50	0.50	0.50	1.00	1.00	1.00	469.68
1.00	0.50	0.50	0.50	0.50	1.00	1.50	1.00	468.60
1.00	0.50	0.50	0.50	0.50	1.50	0.50	1.00	476.95
1.00	0.50	0.50	0.50	0.50	1.50	1.00	1.00	463.16
1.00	0.50	0.50	0.50	0.50	1.50	1.50	1.00	461.34
1.00	0.50	0.50	0.50	1.00	0.50	0.50	1.00	557.98
1.00	0.50	0.50	0.50	1.00	0.50	1.00	1.00	553.28
1.00	0.50	0.50	0.50	1.00	0.50	1.50	1.00	554.76
1.00	0.50	0.50	0.50	1.00	1.00	0.50	1.00	430.62
1.00	0.50	0.50	0.50	1.00	1.00	1.00	1.00	422.90
1.00	0.50	0.50	0.50	1.00	1.00	1.50	1.00	423.43
1.00	0.50	0.50	0.50	1.00	1.50	0.50	1.25	419.39
1.00	0.50	0.50	0.50	1.00	1.50	1.00	1.25	409.29
1.00	0.50	0.50	0.50	1.00	1.50	1.50	1.25	408.74
1.00	0.50	0.50	0.50	1.50	0.50	0.50	1.00	553.02
1.00	0.50	0.50	0.50	1.50	0.50	1.00	1.00	550.71
1.00	0.50	0.50	0.50	1.50	0.50	1.50	1.00	552.99
1.00	0.50	0.50	0.50	1.50	1.00	0.50	1.25	418.01
1.00	0.50	0.50	0.50	1.50	1.00	1.00	1.25	412.01
1.00	0.50	0.50	0.50	1.50	1.00	1.50	1.25	413.27
1.00	0.50	0.50	0.50	1.50	1.50	0.50	1.25	401.87
1.00	0.50	0.50	0.50	1.50	1.50	1.00	1.25	393.88
1.00	0.50	0.50	0.50	1.50	1.50	1.50	1.25	394.20
1.00	0.50	0.50	1.00	0.50	0.50	0.50	1.25	578.49
1.00	0.50	0.50	1.00	0.50	0.50	1.00	1.25	569.38
1.00	0.50	0.50	1.00	0.50	0.50	1.50	1.25	569.39

$x_2$	$x_3$	$x_5$	$x_6$	$x_8$	$x_9$	$x_{10}$	$x_1^*$	$ev_4$
1.00	0.50	0.50	1.00	0.50	1.00	0.50	1.25	457.33
1.00	0.50	0.50	1.00	0.50	1.00	1.00	1.25	442.72
1.00	0.50	0.50	1.00	0.50	1.00	1.50	1.25	440.97
1.00	0.50	0.50	1.00	0.50	1.50	0.50	1.25	446.20
1.00	0.50	0.50	1.00	0.50	1.50	1.00	1.25	429.36
1.00	0.50	0.50	1.00	0.50	1.50	1.50	1.25	426.69
1.00	0.50	0.50	1.00	1.00	0.50	0.50	1.25	537.03
1.00	0.50	0.50	1.00	1.00	0.50	1.00	1.25	530.61
1.00	0.50	0.50	1.00	1.00	0.50	1.50	1.25	531.65
1.00	0.50	0.50	1.00	1.00	1.00	0.50	1.25	399.68
1.00	0.50	0.50	1.00	1.00	1.00	1.00	1.25	388.93
1.00	0.50	0.50	1.00	1.00	1.00	1.50	1.25	388.65
1.00	0.50	0.50	1.00	1.00	1.50	0.50	1.25	381.12
1.00	0.50	0.50	1.00	1.00	1.50	1.00	1.25	367.20
1.00	0.50	0.50	1.00	1.00	1.50	1.50	1.25	365.48
1.00	0.50	0.50	1.00	1.50	0.50	0.50	1.25	528.41
1.00	0.50	0.50	1.00	1.50	0.50	1.00	1.25	524.54
1.00	0.50	0.50	1.00	1.50	0.50	1.50	1.25	526.45
1.00	0.50	0.50	1.00	1.50	1.00	0.50	1.25	380.83
1.00	0.50	0.50	1.00	1.50	1.00	1.00	1.25	372.30
1.00	0.50	0.50	1.00	1.50	1.00	1.50	1.25	372.83
1.00	0.50	0.50	1.00	1.50	1.50	0.50	1.25	356.32
1.00	0.50	0.50	1.00	1.50	1.50	1.00	1.25	344.73
1.00	0.50	0.50	1.00	1.50	1.50	1.50	1.25	344.12
1.00	0.50	0.50	1.50	0.50	0.50	0.50	1.00	588.90
1.00	0.50	0.50	1.50	0.50	0.50	1.00	1.00	579.49
1.00	0.50	0.50	1.50	0.50	0.50	1.50	1.00	579.44
1.00	0.50	0.50	1.50	0.50	1.00	0.50	1.00	467.23
1.00	0.50	0.50	1.50	0.50	1.00	1.00	1.25	452.21
1.00	0.50	0.50	1.50	0.50	1.00	1.50	1.25	450.36
1.00	0.50	0.50	1.50	0.50	1.50	0.50	1.25	455.68
1.00	0.50	0.50	1.50	0.50	1.50	1.00	1.25	438.29
1.00	0.50	0.50	1.50	0.50	1.50	1.50	1.25	435.48
1.00	0.50	0.50	1.50	1.00	0.50	0.50	1.00	547.36
1.00	0.50	0.50	1.50	1.00	0.50	1.00	1.00	540.60
1.00	0.50	0.50	1.50	1.00	0.50	1.50	1.00	541.57
1.00	0.50	0.50	1.50	1.00	1.00	0.50	1.25	409.26
1.00	0.50	0.50	1.50	1.00	1.00	1.00	1.25	398.03
1.00	0.50	0.50	1.50	1.00	1.00	1.50	1.25	397.65
1.00	0.50	0.50	1.50	1.00	1.50	0.50	1.25	390.13
1.00	0.50	0.50	1.50	1.00	1.50	1.00	1.25	375.64
1.00	0.50	0.50	1.50	1.00	1.50	1.50	1.25	373.77
1.00	0.50	0.50	1.50	1.50	0.50	0.50	1.25	538.42
1.00	0.50	0.50	1.50	1.50	0.50	1.00	1.25	534.19
1.00	0.50	0.50	1.50	1.50	0.50	1.50	1.25	536.02
1.00	0.50	0.50	1.50	1.50	1.00	0.50	1.25	389.79

D.1. PROCURA DO ÓPTIMO PARA AS VÁRIAS COMBINAÇÕES DE  $\mathcal{F}245$

$x_2$	$x_3$	$x_5$	$x_6$	$x_8$	$x_9$	$x_{10}$	$x_1^*$	$ev_4$
1.00	0.50	0.50	1.50	1.50	1.00	1.00	1.25	380.71
1.00	0.50	0.50	1.50	1.50	1.00	1.50	1.25	381.13
1.00	0.50	0.50	1.50	1.50	1.50	0.50	1.25	364.67
1.00	0.50	0.50	1.50	1.50	1.50	1.00	1.25	352.45
1.00	0.50	0.50	1.50	1.50	1.50	1.50	1.25	351.68
1.00	0.50	1.00	0.50	0.50	0.50	0.50	1.00	596.20
1.00	0.50	1.00	0.50	0.50	0.50	1.00	1.00	588.58
1.00	0.50	1.00	0.50	0.50	0.50	1.50	1.00	588.96
1.00	0.50	1.00	0.50	0.50	1.00	0.50	1.00	483.92
1.00	0.50	1.00	0.50	0.50	1.00	1.00	1.00	472.12
1.00	0.50	1.00	0.50	0.50	1.00	1.50	1.00	471.08
1.00	0.50	1.00	0.50	0.50	1.50	0.50	1.00	479.13
1.00	0.50	1.00	0.50	0.50	1.50	1.00	1.00	465.58
1.00	0.50	1.00	0.50	0.50	1.50	1.50	1.00	463.82
1.00	0.50	1.00	0.50	1.00	0.50	0.50	1.00	560.23
1.00	0.50	1.00	0.50	1.00	0.50	1.00	1.00	555.71
1.00	0.50	1.00	0.50	1.00	0.50	1.50	1.00	557.24
1.00	0.50	1.00	0.50	1.00	1.00	0.50	1.00	432.74
1.00	0.50	1.00	0.50	1.00	1.00	1.00	1.00	425.30
1.00	0.50	1.00	0.50	1.00	1.00	1.50	1.00	425.90
1.00	0.50	1.00	0.50	1.00	1.50	0.50	1.00	421.64
1.00	0.50	1.00	0.50	1.00	1.50	1.00	1.25	411.73
1.00	0.50	1.00	0.50	1.00	1.50	1.50	1.25	411.23
1.00	0.50	1.00	0.50	1.50	0.50	0.50	1.00	555.27
1.00	0.50	1.00	0.50	1.50	0.50	1.00	1.00	553.15
1.00	0.50	1.00	0.50	1.50	0.50	1.50	1.00	555.47
1.00	0.50	1.00	0.50	1.50	1.00	0.50	1.00	420.18
1.00	0.50	1.00	0.50	1.50	1.00	1.00	1.25	414.46
1.00	0.50	1.00	0.50	1.50	1.00	1.50	1.25	415.75
1.00	0.50	1.00	0.50	1.50	1.50	0.50	1.25	404.13
1.00	0.50	1.00	0.50	1.50	1.50	1.00	1.25	396.32
1.00	0.50	1.00	0.50	1.50	1.50	1.50	1.25	396.69
1.00	0.50	1.00	1.00	0.50	0.50	0.50	1.25	580.91
1.00	0.50	1.00	1.00	0.50	0.50	1.00	1.25	571.86
1.00	0.50	1.00	1.00	0.50	0.50	1.50	1.25	571.88
1.00	0.50	1.00	1.00	0.50	1.00	0.50	1.25	459.71
1.00	0.50	1.00	1.00	0.50	1.00	1.00	1.25	445.19
1.00	0.50	1.00	1.00	0.50	1.00	1.50	1.25	443.46
1.00	0.50	1.00	1.00	0.50	1.50	0.50	1.25	448.54
1.00	0.50	1.00	1.00	0.50	1.50	1.00	1.25	431.82
1.00	0.50	1.00	1.00	0.50	1.50	1.50	1.25	429.18
1.00	0.50	1.00	1.00	1.00	0.50	0.50	1.25	539.40
1.00	0.50	1.00	1.00	1.00	0.50	1.00	1.25	533.08
1.00	0.50	1.00	1.00	1.00	0.50	1.50	1.25	534.14
1.00	0.50	1.00	1.00	1.00	1.00	0.50	1.25	401.99
1.00	0.50	1.00	1.00	1.00	1.00	1.00	1.25	391.38

$x_2$	$x_3$	$x_5$	$x_6$	$x_8$	$x_9$	$x_{10}$	$x_{11}^*$	$ev_4$
1.00	0.50	1.00	1.00	1.00	1.00	1.50	1.25	391.14
1.00	0.50	1.00	1.00	1.00	1.50	0.50	1.25	383.38
1.00	0.50	1.00	1.00	1.00	1.50	1.00	1.25	369.64
1.00	0.50	1.00	1.00	1.00	1.50	1.50	1.25	367.97
1.00	0.50	1.00	1.00	1.50	0.50	0.50	1.25	530.79
1.00	0.50	1.00	1.00	1.50	0.50	1.00	1.25	527.01
1.00	0.50	1.00	1.00	1.50	0.50	1.50	1.25	528.94
1.00	0.50	1.00	1.00	1.50	1.00	0.50	1.25	383.14
1.00	0.50	1.00	1.00	1.50	1.00	1.00	1.25	374.75
1.00	0.50	1.00	1.00	1.50	1.00	1.50	1.25	375.31
1.00	0.50	1.00	1.00	1.50	1.50	0.50	1.25	358.58
1.00	0.50	1.00	1.00	1.50	1.50	1.00	1.25	347.17
1.00	0.50	1.00	1.00	1.50	1.50	1.50	1.25	346.60
1.00	0.50	1.00	1.50	0.50	0.50	0.50	1.00	591.18
1.00	0.50	1.00	1.50	0.50	0.50	1.00	1.00	581.93
1.00	0.50	1.00	1.50	0.50	0.50	1.50	1.00	581.92
1.00	0.50	1.00	1.50	0.50	1.00	0.50	1.00	469.40
1.00	0.50	1.00	1.50	0.50	1.00	1.00	1.00	454.65
1.00	0.50	1.00	1.50	0.50	1.00	1.50	1.25	452.85
1.00	0.50	1.00	1.50	0.50	1.50	0.50	1.25	457.97
1.00	0.50	1.00	1.50	0.50	1.50	1.00	1.25	440.74
1.00	0.50	1.00	1.50	0.50	1.50	1.50	1.25	437.97
1.00	0.50	1.00	1.50	1.00	0.50	0.50	1.00	549.51
1.00	0.50	1.00	1.50	1.00	0.50	1.00	1.00	543.01
1.00	0.50	1.00	1.50	1.00	0.50	1.50	1.00	544.04
1.00	0.50	1.00	1.50	1.00	1.00	0.50	1.25	411.51
1.00	0.50	1.00	1.50	1.00	1.00	1.00	1.25	400.46
1.00	0.50	1.00	1.50	1.00	1.00	1.50	1.25	400.13
1.00	0.50	1.00	1.50	1.00	1.50	0.50	1.25	392.32
1.00	0.50	1.00	1.50	1.00	1.50	1.00	1.25	378.06
1.00	0.50	1.00	1.50	1.00	1.50	1.50	1.25	376.25
1.00	0.50	1.00	1.50	1.50	0.50	0.50	1.00	540.61
1.00	0.50	1.00	1.50	1.50	0.50	1.00	1.00	536.63
1.00	0.50	1.00	1.50	1.50	0.50	1.50	1.25	538.51
1.00	0.50	1.00	1.50	1.50	1.00	0.50	1.25	392.04
1.00	0.50	1.00	1.50	1.50	1.00	1.00	1.25	383.15
1.00	0.50	1.00	1.50	1.50	1.00	1.50	1.25	383.61
1.00	0.50	1.00	1.50	1.50	1.50	0.50	1.25	366.85
1.00	0.50	1.00	1.50	1.50	1.50	1.00	1.25	354.87
1.00	0.50	1.00	1.50	1.50	1.50	1.50	1.25	354.15
1.00	0.50	1.50	0.50	0.50	0.50	0.50	1.00	598.70
1.00	0.50	1.50	0.50	0.50	0.50	1.00	1.00	591.08
1.00	0.50	1.50	0.50	0.50	0.50	1.50	1.00	591.46
1.00	0.50	1.50	0.50	0.50	1.00	0.50	1.00	486.42
1.00	0.50	1.50	0.50	0.50	1.00	1.00	1.00	474.62
1.00	0.50	1.50	0.50	0.50	1.00	1.50	1.00	473.58



D.1. PROCURA DO ÓPTIMO PARA AS VÁRIAS COMBINAÇÕES DE  $\mathcal{F}247$

$x_2$	$x_3$	$x_5$	$x_6$	$x_8$	$x_9$	$x_{10}$	$x_1^*$	$ev_4$
1.00	0.50	1.50	0.50	0.50	1.50	0.50	1.00	481.63
1.00	0.50	1.50	0.50	0.50	1.50	1.00	1.00	468.08
1.00	0.50	1.50	0.50	0.50	1.50	1.50	1.00	466.32
1.00	0.50	1.50	0.50	1.00	0.50	0.50	1.00	562.73
1.00	0.50	1.50	0.50	1.00	0.50	1.00	1.00	558.21
1.00	0.50	1.50	0.50	1.00	0.50	1.50	1.00	559.74
1.00	0.50	1.50	0.50	1.00	1.00	0.50	1.00	435.24
1.00	0.50	1.50	0.50	1.00	1.00	1.00	1.00	427.80
1.00	0.50	1.50	0.50	1.00	1.00	1.50	1.00	428.40
1.00	0.50	1.50	0.50	1.00	1.50	0.50	1.00	424.14
1.00	0.50	1.50	0.50	1.00	1.50	1.00	1.25	414.23
1.00	0.50	1.50	0.50	1.00	1.50	1.50	1.25	413.73
1.00	0.50	1.50	0.50	1.50	0.50	0.50	1.00	557.77
1.00	0.50	1.50	0.50	1.50	0.50	1.00	1.00	555.65
1.00	0.50	1.50	0.50	1.50	0.50	1.50	1.00	557.97
1.00	0.50	1.50	0.50	1.50	1.00	0.50	1.00	422.68
1.00	0.50	1.50	0.50	1.50	1.00	1.00	1.25	416.96
1.00	0.50	1.50	0.50	1.50	1.00	1.50	1.25	418.25
1.00	0.50	1.50	0.50	1.50	1.50	0.50	1.25	406.63
1.00	0.50	1.50	0.50	1.50	1.50	1.00	1.25	398.82
1.00	0.50	1.50	0.50	1.50	1.50	1.50	1.25	399.19
1.00	0.50	1.50	1.00	0.50	0.50	0.50	1.25	583.41
1.00	0.50	1.50	1.00	0.50	0.50	1.00	1.25	574.36
1.00	0.50	1.50	1.00	0.50	0.50	1.50	1.25	574.38
1.00	0.50	1.50	1.00	0.50	1.00	0.50	1.25	462.21
1.00	0.50	1.50	1.00	0.50	1.00	1.00	1.25	447.69
1.00	0.50	1.50	1.00	0.50	1.00	1.50	1.25	445.96
1.00	0.50	1.50	1.00	0.50	1.50	0.50	1.25	451.04
1.00	0.50	1.50	1.00	0.50	1.50	1.00	1.25	434.32
1.00	0.50	1.50	1.00	0.50	1.50	1.50	1.25	431.68
1.00	0.50	1.50	1.00	1.00	0.50	0.50	1.25	541.90
1.00	0.50	1.50	1.00	1.00	0.50	1.00	1.25	535.58
1.00	0.50	1.50	1.00	1.00	0.50	1.50	1.25	536.64
1.00	0.50	1.50	1.00	1.00	1.00	0.50	1.25	404.49
1.00	0.50	1.50	1.00	1.00	1.00	1.00	1.25	393.88
1.00	0.50	1.50	1.00	1.00	1.00	1.50	1.25	393.64
1.00	0.50	1.50	1.00	1.00	1.50	0.50	1.25	385.88
1.00	0.50	1.50	1.00	1.00	1.50	1.00	1.25	372.14
1.00	0.50	1.50	1.00	1.00	1.50	1.50	1.25	370.47
1.00	0.50	1.50	1.00	1.50	0.50	0.50	1.25	533.29
1.00	0.50	1.50	1.00	1.50	0.50	1.00	1.25	529.51
1.00	0.50	1.50	1.00	1.50	0.50	1.50	1.25	531.44
1.00	0.50	1.50	1.00	1.50	1.00	0.50	1.25	385.64
1.00	0.50	1.50	1.00	1.50	1.00	1.00	1.25	377.25
1.00	0.50	1.50	1.00	1.50	1.00	1.50	1.25	377.81
1.00	0.50	1.50	1.00	1.50	1.50	0.50	1.25	361.08

$x_2$	$x_3$	$x_5$	$x_6$	$x_8$	$x_9$	$x_{10}$	$x_1^*$	$ev_4$
1.00	0.50	1.50	1.00	1.50	1.50	1.00	1.25	349.67
1.00	0.50	1.50	1.00	1.50	1.50	1.50	1.25	349.10
1.00	0.50	1.50	1.50	0.50	0.50	0.50	1.00	593.68
1.00	0.50	1.50	1.50	0.50	0.50	1.00	1.00	584.43
1.00	0.50	1.50	1.50	0.50	0.50	1.50	1.00	584.42
1.00	0.50	1.50	1.50	0.50	1.00	0.50	1.00	471.90
1.00	0.50	1.50	1.50	0.50	1.00	1.00	1.00	457.15
1.00	0.50	1.50	1.50	0.50	1.00	1.50	1.25	455.35
1.00	0.50	1.50	1.50	0.50	1.50	0.50	1.25	460.47
1.00	0.50	1.50	1.50	0.50	1.50	1.00	1.25	443.24
1.00	0.50	1.50	1.50	0.50	1.50	1.50	1.25	440.47
1.00	0.50	1.50	1.50	1.00	0.50	0.50	1.00	552.01
1.00	0.50	1.50	1.50	1.00	0.50	1.00	1.00	545.51
1.00	0.50	1.50	1.50	1.00	0.50	1.50	1.00	546.54
1.00	0.50	1.50	1.50	1.00	1.00	0.50	1.25	414.01
1.00	0.50	1.50	1.50	1.00	1.00	1.00	1.25	402.96
1.00	0.50	1.50	1.50	1.00	1.00	1.50	1.25	402.63
1.00	0.50	1.50	1.50	1.00	1.50	0.50	1.25	394.82
1.00	0.50	1.50	1.50	1.00	1.50	1.00	1.25	380.56
1.00	0.50	1.50	1.50	1.00	1.50	1.50	1.25	378.75
1.00	0.50	1.50	1.50	1.50	0.50	0.50	1.00	543.11
1.00	0.50	1.50	1.50	1.50	0.50	1.00	1.00	539.13
1.00	0.50	1.50	1.50	1.50	0.50	1.50	1.25	541.01
1.00	0.50	1.50	1.50	1.50	1.00	0.50	1.25	394.54
1.00	0.50	1.50	1.50	1.50	1.00	1.00	1.25	385.65
1.00	0.50	1.50	1.50	1.50	1.00	1.50	1.25	386.11
1.00	0.50	1.50	1.50	1.50	1.50	0.50	1.25	369.35
1.00	0.50	1.50	1.50	1.50	1.50	1.00	1.25	357.37
1.00	0.50	1.50	1.50	1.50	1.50	1.50	1.25	356.65
1.00	1.00	0.50	0.50	0.50	0.50	0.50	1.25	509.19
1.00	1.00	0.50	0.50	0.50	0.50	1.00	1.00	503.47
1.00	1.00	0.50	0.50	0.50	0.50	1.50	1.00	504.37
1.00	1.00	0.50	0.50	0.50	1.00	0.50	1.25	404.35
1.00	1.00	0.50	0.50	0.50	1.00	1.00	1.00	395.11
1.00	1.00	0.50	0.50	0.50	1.00	1.50	1.00	394.84
1.00	1.00	0.50	0.50	0.50	1.50	0.50	1.25	402.59
1.00	1.00	0.50	0.50	0.50	1.50	1.00	1.25	392.10
1.00	1.00	0.50	0.50	0.50	1.50	1.50	1.25	391.36
1.00	1.00	0.50	0.50	1.00	0.50	0.50	1.25	483.90
1.00	1.00	0.50	0.50	1.00	0.50	1.00	1.25	481.26
1.00	1.00	0.50	0.50	1.00	0.50	1.50	1.00	483.26
1.00	1.00	0.50	0.50	1.00	1.00	0.50	1.25	367.47
1.00	1.00	0.50	0.50	1.00	1.00	1.00	1.25	362.62
1.00	1.00	0.50	0.50	1.00	1.00	1.50	1.25	364.00
1.00	1.00	0.50	0.50	1.00	1.50	0.50	1.25	360.69
1.00	1.00	0.50	0.50	1.00	1.50	1.00	1.25	354.25

D.1. PROCURA DO ÓPTIMO PARA AS VÁRIAS COMBINAÇÕES DE  $\mathcal{F}249$

$x_2$	$x_3$	$x_5$	$x_6$	$x_8$	$x_9$	$x_{10}$	$x_1^*$	$ev_4$
1.00	1.00	0.50	0.50	1.00	1.50	1.50	1.25	354.83
1.00	1.00	0.50	0.50	1.50	0.50	0.50	1.25	485.60
1.00	1.00	0.50	0.50	1.50	0.50	1.00	1.25	484.45
1.00	1.00	0.50	0.50	1.50	0.50	1.50	1.25	487.02
1.00	1.00	0.50	0.50	1.50	1.00	0.50	1.25	364.74
1.00	1.00	0.50	0.50	1.50	1.00	1.00	1.25	360.72
1.00	1.00	0.50	0.50	1.50	1.00	1.50	1.25	362.50
1.00	1.00	0.50	0.50	1.50	1.50	0.50	1.25	354.73
1.00	1.00	0.50	0.50	1.50	1.50	1.00	1.25	349.25
1.00	1.00	0.50	0.50	1.50	1.50	1.50	1.25	350.35
1.00	1.00	0.50	1.00	0.50	0.50	0.50	1.25	483.16
1.00	1.00	0.50	1.00	0.50	0.50	1.00	1.25	475.75
1.00	1.00	0.50	1.00	0.50	0.50	1.50	1.25	476.22
1.00	1.00	0.50	1.00	0.50	1.00	0.50	1.25	366.31
1.00	1.00	0.50	1.00	0.50	1.00	1.00	1.25	354.05
1.00	1.00	0.50	1.00	0.50	1.00	1.50	1.25	353.05
1.00	1.00	0.50	1.00	0.50	1.50	0.50	1.25	357.27
1.00	1.00	0.50	1.00	0.50	1.50	1.00	1.25	343.48
1.00	1.00	0.50	1.00	0.50	1.50	1.50	1.25	341.82
1.00	1.00	0.50	1.00	1.00	0.50	0.50	1.25	449.98
1.00	1.00	0.50	1.00	1.00	0.50	1.00	1.25	445.78
1.00	1.00	0.50	1.00	1.00	0.50	1.50	1.25	447.42
1.00	1.00	0.50	1.00	1.00	1.00	0.50	1.25	320.59
1.00	1.00	0.50	1.00	1.00	1.00	1.00	1.25	313.04
1.00	1.00	0.50	1.00	1.00	1.00	1.50	1.25	313.75
1.00	1.00	0.50	1.00	1.00	1.50	0.50	1.25	305.29
1.00	1.00	0.50	1.00	1.00	1.50	1.00	1.25	295.55
1.00	1.00	0.50	1.00	1.00	1.50	1.50	1.25	295.24
1.00	1.00	0.50	1.00	1.50	0.50	0.50	1.25	448.50
1.00	1.00	0.50	1.00	1.50	0.50	1.00	1.25	446.08
1.00	1.00	0.50	1.00	1.50	0.50	1.50	1.25	448.38
1.00	1.00	0.50	1.00	1.50	1.00	0.50	1.25	312.34
1.00	1.00	0.50	1.00	1.50	1.00	1.00	1.25	306.40
1.00	1.00	0.50	1.00	1.50	1.00	1.50	1.25	307.80
1.00	1.00	0.50	1.00	1.50	1.50	0.50	1.25	292.93
1.00	1.00	0.50	1.00	1.50	1.50	1.00	1.25	285.28
1.00	1.00	0.50	1.00	1.50	1.50	1.50	1.25	285.83
1.00	1.00	0.50	1.50	0.50	0.50	0.50	1.25	493.08
1.00	1.00	0.50	1.50	0.50	0.50	1.00	1.25	485.33
1.00	1.00	0.50	1.50	0.50	0.50	1.50	1.25	485.73
1.00	1.00	0.50	1.50	0.50	1.00	0.50	1.25	375.29
1.00	1.00	0.50	1.50	0.50	1.00	1.00	1.25	362.43
1.00	1.00	0.50	1.50	0.50	1.00	1.50	1.25	361.32
1.00	1.00	0.50	1.50	0.50	1.50	0.50	1.25	365.46
1.00	1.00	0.50	1.50	0.50	1.50	1.00	1.25	351.00
1.00	1.00	0.50	1.50	0.50	1.50	1.50	1.25	349.17

$x_2$	$x_3$	$x_5$	$x_6$	$x_8$	$x_9$	$x_{10}$	$x_1^*$	$ev_4$
1.00	1.00	0.50	1.50	1.00	0.50	0.50	1.25	459.25
1.00	1.00	0.50	1.50	1.00	0.50	1.00	1.25	454.69
1.00	1.00	0.50	1.50	1.00	0.50	1.50	1.25	456.26
1.00	1.00	0.50	1.50	1.00	1.00	0.50	1.25	328.81
1.00	1.00	0.50	1.50	1.00	1.00	1.00	1.25	320.68
1.00	1.00	0.50	1.50	1.00	1.00	1.50	1.25	321.26
1.00	1.00	0.50	1.50	1.00	1.50	0.50	1.25	312.55
1.00	1.00	0.50	1.50	1.00	1.50	1.00	1.25	302.10
1.00	1.00	0.50	1.50	1.00	1.50	1.50	1.25	301.62
1.00	1.00	0.50	1.50	1.50	0.50	0.50	1.25	457.43
1.00	1.00	0.50	1.50	1.50	0.50	1.00	1.25	454.56
1.00	1.00	0.50	1.50	1.50	0.50	1.50	1.25	456.79
1.00	1.00	0.50	1.50	1.50	1.00	0.50	1.25	319.94
1.00	1.00	0.50	1.50	1.50	1.00	1.00	1.25	313.35
1.00	1.00	0.50	1.50	1.50	1.00	1.50	1.25	314.62
1.00	1.00	0.50	1.50	1.50	1.50	0.50	1.25	299.53
1.00	1.00	0.50	1.50	1.50	1.50	1.00	1.25	291.11
1.00	1.00	0.50	1.50	1.50	1.50	1.50	1.25	291.48
1.00	1.00	1.00	0.50	0.50	0.50	0.50	1.00	511.42
1.00	1.00	1.00	0.50	0.50	0.50	1.00	1.00	505.88
1.00	1.00	1.00	0.50	0.50	0.50	1.50	1.00	506.84
1.00	1.00	1.00	0.50	0.50	1.00	0.50	1.00	406.52
1.00	1.00	1.00	0.50	0.50	1.00	1.00	1.00	397.47
1.00	1.00	1.00	0.50	0.50	1.00	1.50	1.00	397.30
1.00	1.00	1.00	0.50	0.50	1.50	0.50	1.25	404.77
1.00	1.00	1.00	0.50	0.50	1.50	1.00	1.25	394.52
1.00	1.00	1.00	0.50	0.50	1.50	1.50	1.25	393.84
1.00	1.00	1.00	0.50	1.00	0.50	0.50	1.00	486.08
1.00	1.00	1.00	0.50	1.00	0.50	1.00	1.00	483.66
1.00	1.00	1.00	0.50	1.00	0.50	1.50	1.00	485.72
1.00	1.00	1.00	0.50	1.00	1.00	0.50	1.25	369.59
1.00	1.00	1.00	0.50	1.00	1.00	1.00	1.25	365.02
1.00	1.00	1.00	0.50	1.00	1.00	1.50	1.25	366.47
1.00	1.00	1.00	0.50	1.00	1.50	0.50	1.25	362.72
1.00	1.00	1.00	0.50	1.00	1.50	1.00	1.25	356.63
1.00	1.00	1.00	0.50	1.00	1.50	1.50	1.25	357.30
1.00	1.00	1.00	0.50	1.50	0.50	0.50	1.00	487.78
1.00	1.00	1.00	0.50	1.50	0.50	1.00	1.00	486.85
1.00	1.00	1.00	0.50	1.50	0.50	1.50	1.00	489.48
1.00	1.00	1.00	0.50	1.50	1.00	0.50	1.25	366.86
1.00	1.00	1.00	0.50	1.50	1.00	1.00	1.25	363.12
1.00	1.00	1.00	0.50	1.50	1.00	1.50	1.25	364.97
1.00	1.00	1.00	0.50	1.50	1.50	0.50	1.25	356.75
1.00	1.00	1.00	0.50	1.50	1.50	1.00	1.25	351.62
1.00	1.00	1.00	0.50	1.50	1.50	1.50	1.25	352.82
1.00	1.00	1.00	1.00	0.50	0.50	0.50	1.25	485.49

D.1. PROCURA DO ÓPTIMO PARA AS VÁRIAS COMBINAÇÕES DE F251

$x_2$	$x_3$	$x_5$	$x_6$	$x_8$	$x_9$	$x_{10}$	$x_1^*$	ev <sub>4</sub>
1.00	1.00	1.00	1.00	0.50	0.50	1.00	1.25	478.20
1.00	1.00	1.00	1.00	0.50	0.50	1.50	1.25	478.70
1.00	1.00	1.00	1.00	0.50	1.00	0.50	1.25	368.58
1.00	1.00	1.00	1.00	0.50	1.00	1.00	1.25	356.48
1.00	1.00	1.00	1.00	0.50	1.00	1.50	1.25	355.53
1.00	1.00	1.00	1.00	0.50	1.50	0.50	1.25	359.46
1.00	1.00	1.00	1.00	0.50	1.50	1.00	1.25	345.90
1.00	1.00	1.00	1.00	0.50	1.50	1.50	1.25	344.30
1.00	1.00	1.00	1.00	1.00	0.50	0.50	1.25	452.23
1.00	1.00	1.00	1.00	1.00	0.50	1.00	1.25	448.22
1.00	1.00	1.00	1.00	1.00	0.50	1.50	1.25	449.89
1.00	1.00	1.00	1.00	1.00	1.00	0.50	1.25	322.71
1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.25	315.44
1.00	1.00	1.00	1.00	1.00	1.00	1.50	1.25	316.22
1.00	1.00	1.00	1.00	1.00	1.50	0.50	1.25	307.31
1.00	1.00	1.00	1.00	1.00	1.50	1.00	1.25	297.93
1.00	1.00	1.00	1.00	1.00	1.50	1.50	1.25	297.71
1.00	1.00	1.00	1.00	1.50	0.50	0.50	1.25	450.75
1.00	1.00	1.00	1.00	1.50	0.50	1.00	1.25	448.51
1.00	1.00	1.00	1.00	1.50	0.50	1.50	1.25	450.86
1.00	1.00	1.00	1.00	1.50	1.00	0.50	1.25	314.46
1.00	1.00	1.00	1.00	1.50	1.00	1.00	1.25	308.80
1.00	1.00	1.00	1.00	1.50	1.00	1.50	1.25	310.27
1.00	1.00	1.00	1.00	1.50	1.50	0.50	1.25	294.96
1.00	1.00	1.00	1.00	1.50	1.50	1.00	1.25	287.65
1.00	1.00	1.00	1.00	1.50	1.50	1.50	1.25	288.30
1.00	1.00	1.00	1.50	0.50	0.50	0.50	1.00	495.29
1.00	1.00	1.00	1.50	0.50	0.50	1.00	1.00	487.75
1.00	1.00	1.00	1.50	0.50	0.50	1.50	1.00	488.21
1.00	1.00	1.00	1.50	0.50	1.00	0.50	1.25	377.45
1.00	1.00	1.00	1.50	0.50	1.00	1.00	1.25	364.84
1.00	1.00	1.00	1.50	0.50	1.00	1.50	1.25	363.79
1.00	1.00	1.00	1.50	0.50	1.50	0.50	1.25	367.53
1.00	1.00	1.00	1.50	0.50	1.50	1.00	1.25	353.39
1.00	1.00	1.00	1.50	0.50	1.50	1.50	1.25	351.64
1.00	1.00	1.00	1.50	1.00	0.50	0.50	1.25	461.40
1.00	1.00	1.00	1.50	1.00	0.50	1.00	1.25	457.10
1.00	1.00	1.00	1.50	1.00	0.50	1.50	1.25	458.73
1.00	1.00	1.00	1.50	1.00	1.00	0.50	1.25	330.81
1.00	1.00	1.00	1.50	1.00	1.00	1.00	1.25	323.04
1.00	1.00	1.00	1.50	1.00	1.00	1.50	1.25	323.72
1.00	1.00	1.00	1.50	1.00	1.50	0.50	1.25	314.41
1.00	1.00	1.00	1.50	1.00	1.50	1.00	1.25	304.44
1.00	1.00	1.00	1.50	1.00	1.50	1.50	1.25	304.07
1.00	1.00	1.00	1.50	1.50	0.50	0.50	1.25	459.59
1.00	1.00	1.00	1.50	1.50	0.50	1.00	1.25	456.97

$x_2$	$x_3$	$x_5$	$x_6$	$x_8$	$x_9$	$x_{10}$	$x_{11}^*$	$ev_4$
1.00	1.00	1.00	1.50	1.50	0.50	1.50	1.25	459.26
1.00	1.00	1.00	1.50	1.50	1.00	0.50	1.25	321.94
1.00	1.00	1.00	1.50	1.50	1.00	1.00	1.25	315.72
1.00	1.00	1.00	1.50	1.50	1.00	1.50	1.25	317.08
1.00	1.00	1.00	1.50	1.50	1.50	0.50	1.25	301.40
1.00	1.00	1.00	1.50	1.50	1.50	1.00	1.25	293.45
1.00	1.00	1.00	1.50	1.50	1.50	1.50	1.25	293.93
1.00	1.00	1.50	0.50	0.50	0.50	0.50	1.00	513.92
1.00	1.00	1.50	0.50	0.50	0.50	1.00	1.00	508.38
1.00	1.00	1.50	0.50	0.50	0.50	1.50	1.00	509.34
1.00	1.00	1.50	0.50	0.50	1.00	0.50	1.00	409.02
1.00	1.00	1.50	0.50	0.50	1.00	1.00	1.00	399.97
1.00	1.00	1.50	0.50	0.50	1.00	1.50	1.00	399.80
1.00	1.00	1.50	0.50	0.50	1.50	0.50	1.25	407.27
1.00	1.00	1.50	0.50	0.50	1.50	1.00	1.25	397.02
1.00	1.00	1.50	0.50	0.50	1.50	1.50	1.25	396.34
1.00	1.00	1.50	0.50	1.00	0.50	0.50	1.00	488.58
1.00	1.00	1.50	0.50	1.00	0.50	1.00	1.00	486.16
1.00	1.00	1.50	0.50	1.00	0.50	1.50	1.00	488.22
1.00	1.00	1.50	0.50	1.00	1.00	0.50	1.25	372.09
1.00	1.00	1.50	0.50	1.00	1.00	1.00	1.25	367.52
1.00	1.00	1.50	0.50	1.00	1.00	1.50	1.25	368.97
1.00	1.00	1.50	0.50	1.00	1.50	0.50	1.25	365.22
1.00	1.00	1.50	0.50	1.00	1.50	1.00	1.25	359.13
1.00	1.00	1.50	0.50	1.00	1.50	1.50	1.25	359.80
1.00	1.00	1.50	0.50	1.50	0.50	0.50	1.00	490.28
1.00	1.00	1.50	0.50	1.50	0.50	1.00	1.00	489.35
1.00	1.00	1.50	0.50	1.50	0.50	1.50	1.00	491.98
1.00	1.00	1.50	0.50	1.50	1.00	0.50	1.25	369.36
1.00	1.00	1.50	0.50	1.50	1.00	1.00	1.25	365.62
1.00	1.00	1.50	0.50	1.50	1.00	1.50	1.25	367.47
1.00	1.00	1.50	0.50	1.50	1.50	0.50	1.25	359.25
1.00	1.00	1.50	0.50	1.50	1.50	1.00	1.25	354.12
1.00	1.00	1.50	0.50	1.50	1.50	1.50	1.25	355.32
1.00	1.00	1.50	1.00	0.50	0.50	0.50	1.25	487.99
1.00	1.00	1.50	1.00	0.50	0.50	1.00	1.25	480.70
1.00	1.00	1.50	1.00	0.50	0.50	1.50	1.25	481.20
1.00	1.00	1.50	1.00	0.50	1.00	0.50	1.25	371.08
1.00	1.00	1.50	1.00	0.50	1.00	1.00	1.25	358.98
1.00	1.00	1.50	1.00	0.50	1.00	1.50	1.25	358.03
1.00	1.00	1.50	1.00	0.50	1.50	0.50	1.25	361.96
1.00	1.00	1.50	1.00	0.50	1.50	1.00	1.25	348.40
1.00	1.00	1.50	1.00	0.50	1.50	1.50	1.25	346.80
1.00	1.00	1.50	1.00	1.00	0.50	0.50	1.25	454.73
1.00	1.00	1.50	1.00	1.00	0.50	1.00	1.25	450.72
1.00	1.00	1.50	1.00	1.00	0.50	1.50	1.25	452.39

D.1. PROCURA DO ÓPTIMO PARA AS VÁRIAS COMBINAÇÕES DE  $\mathcal{F}253$

$x_2$	$x_3$	$x_5$	$x_6$	$x_8$	$x_9$	$x_{10}$	$x_1^*$	$ev_4$
1.00	1.00	1.50	1.00	1.00	1.00	0.50	1.25	325.21
1.00	1.00	1.50	1.00	1.00	1.00	1.00	1.25	317.94
1.00	1.00	1.50	1.00	1.00	1.00	1.50	1.25	318.72
1.00	1.00	1.50	1.00	1.00	1.50	0.50	1.25	309.81
1.00	1.00	1.50	1.00	1.00	1.50	1.00	1.25	300.43
1.00	1.00	1.50	1.00	1.00	1.50	1.50	1.25	300.21
1.00	1.00	1.50	1.00	1.50	0.50	0.50	1.25	453.25
1.00	1.00	1.50	1.00	1.50	0.50	1.00	1.25	451.01
1.00	1.00	1.50	1.00	1.50	0.50	1.50	1.25	453.36
1.00	1.00	1.50	1.00	1.50	1.00	0.50	1.25	316.96
1.00	1.00	1.50	1.00	1.50	1.00	1.00	1.25	311.30
1.00	1.00	1.50	1.00	1.50	1.00	1.50	1.25	312.77
1.00	1.00	1.50	1.00	1.50	1.50	0.50	1.25	297.46
1.00	1.00	1.50	1.00	1.50	1.50	1.00	1.25	290.15
1.00	1.00	1.50	1.00	1.50	1.50	1.50	1.25	290.80
1.00	1.00	1.50	1.50	0.50	0.50	0.50	1.00	497.79
1.00	1.00	1.50	1.50	0.50	0.50	1.00	1.00	490.25
1.00	1.00	1.50	1.50	0.50	0.50	1.50	1.00	490.71
1.00	1.00	1.50	1.50	0.50	1.00	0.50	1.25	379.95
1.00	1.00	1.50	1.50	0.50	1.00	1.00	1.25	367.34
1.00	1.00	1.50	1.50	0.50	1.00	1.50	1.25	366.29
1.00	1.00	1.50	1.50	0.50	1.50	0.50	1.25	370.03
1.00	1.00	1.50	1.50	0.50	1.50	1.00	1.25	355.89
1.00	1.00	1.50	1.50	0.50	1.50	1.50	1.25	354.14
1.00	1.00	1.50	1.50	1.00	0.50	0.50	1.25	463.90
1.00	1.00	1.50	1.50	1.00	0.50	1.00	1.25	459.60
1.00	1.00	1.50	1.50	1.00	0.50	1.50	1.25	461.23
1.00	1.00	1.50	1.50	1.00	1.00	0.50	1.25	333.31
1.00	1.00	1.50	1.50	1.00	1.00	1.00	1.25	325.54
1.00	1.00	1.50	1.50	1.00	1.00	1.50	1.25	326.22
1.00	1.00	1.50	1.50	1.00	1.50	0.50	1.25	316.91
1.00	1.00	1.50	1.50	1.00	1.50	1.00	1.25	306.94
1.00	1.00	1.50	1.50	1.00	1.50	1.50	1.25	306.57
1.00	1.00	1.50	1.50	1.50	0.50	0.50	1.25	462.09
1.00	1.00	1.50	1.50	1.50	0.50	1.00	1.25	459.47
1.00	1.00	1.50	1.50	1.50	0.50	1.50	1.25	461.76
1.00	1.00	1.50	1.50	1.50	1.00	0.50	1.25	324.44
1.00	1.00	1.50	1.50	1.50	1.00	1.00	1.25	318.22
1.00	1.00	1.50	1.50	1.50	1.00	1.50	1.25	319.58
1.00	1.00	1.50	1.50	1.50	1.50	0.50	1.25	303.90
1.00	1.00	1.50	1.50	1.50	1.50	1.00	1.25	295.95
1.00	1.00	1.50	1.50	1.50	1.50	1.50	1.25	296.43
1.00	1.50	0.50	0.50	0.50	0.50	0.50	1.25	491.58
1.00	1.50	0.50	0.50	0.50	0.50	1.00	1.00	486.88
1.00	1.50	0.50	0.50	0.50	0.50	1.50	1.00	487.93
1.00	1.50	0.50	0.50	0.50	1.00	0.50	1.25	391.39

$x_2$	$x_3$	$x_5$	$x_6$	$x_8$	$x_9$	$x_{10}$	$x_{11}^*$	$ev_4$
1.00	1.50	0.50	0.50	0.50	1.00	1.00	1.25	383.64
1.00	1.50	0.50	0.50	0.50	1.00	1.50	1.00	383.68
1.00	1.50	0.50	0.50	0.50	1.50	0.50	1.25	391.42
1.00	1.50	0.50	0.50	0.50	1.50	1.00	1.25	382.34
1.00	1.50	0.50	0.50	0.50	1.50	1.50	1.25	382.02
1.00	1.50	0.50	0.50	1.00	0.50	0.50	1.25	469.62
1.00	1.50	0.50	0.50	1.00	0.50	1.00	1.25	468.66
1.00	1.50	0.50	0.50	1.00	0.50	1.50	1.25	471.01
1.00	1.50	0.50	0.50	1.00	1.00	0.50	1.25	358.34
1.00	1.50	0.50	0.50	1.00	1.00	1.00	1.25	355.70
1.00	1.50	0.50	0.50	1.00	1.00	1.50	1.25	357.59
1.00	1.50	0.50	0.50	1.00	1.50	0.50	1.25	352.25
1.00	1.50	0.50	0.50	1.00	1.50	1.00	1.25	348.28
1.00	1.50	0.50	0.50	1.00	1.50	1.50	1.25	349.63
1.00	1.50	0.50	0.50	1.50	0.50	0.50	1.25	476.72
1.00	1.50	0.50	0.50	1.50	0.50	1.00	1.25	476.50
1.00	1.50	0.50	0.50	1.50	0.50	1.50	1.25	479.14
1.00	1.50	0.50	0.50	1.50	1.00	0.50	1.25	363.22
1.00	1.50	0.50	0.50	1.50	1.00	1.00	1.25	361.00
1.00	1.50	0.50	0.50	1.50	1.00	1.50	1.25	363.09
1.00	1.50	0.50	0.50	1.50	1.50	0.50	1.25	355.52
1.00	1.50	0.50	0.50	1.50	1.50	1.00	1.25	352.03
1.00	1.50	0.50	0.50	1.50	1.50	1.50	1.25	353.64
1.00	1.50	0.50	1.00	0.50	0.50	0.50	1.25	462.43
1.00	1.50	0.50	1.00	0.50	0.50	1.00	1.25	455.75
1.00	1.50	0.50	1.00	0.50	0.50	1.50	1.25	456.25
1.00	1.50	0.50	1.00	0.50	1.00	0.50	1.25	348.93
1.00	1.50	0.50	1.00	0.50	1.00	1.00	1.25	338.14
1.00	1.50	0.50	1.00	0.50	1.00	1.50	1.25	337.37
1.00	1.50	0.50	1.00	0.50	1.50	0.50	1.25	342.51
1.00	1.50	0.50	1.00	0.50	1.50	1.00	1.25	330.30
1.00	1.50	0.50	1.00	0.50	1.50	1.50	1.25	329.04
1.00	1.50	0.50	1.00	1.00	0.50	0.50	1.25	430.81
1.00	1.50	0.50	1.00	1.00	0.50	1.00	1.25	428.78
1.00	1.50	0.50	1.00	1.00	0.50	1.50	1.25	430.88
1.00	1.50	0.50	1.00	1.00	1.00	0.50	1.25	305.09
1.00	1.50	0.50	1.00	1.00	1.00	1.00	1.25	300.69
1.00	1.50	0.50	1.00	1.00	1.00	1.50	1.25	302.13
1.00	1.50	0.50	1.00	1.00	1.50	0.50	1.25	291.15
1.00	1.50	0.50	1.00	1.00	1.50	1.00	1.25	285.01
1.00	1.50	0.50	1.00	1.00	1.50	1.50	1.25	285.78
1.00	1.50	0.50	1.00	1.50	0.50	0.50	1.25	436.32
1.00	1.50	0.50	1.00	1.50	0.50	1.00	1.25	435.18
1.00	1.50	0.50	1.00	1.50	0.50	1.50	1.25	437.61
1.00	1.50	0.50	1.00	1.50	1.00	0.50	1.25	307.21
1.00	1.50	0.50	1.00	1.50	1.00	1.00	1.25	303.62



D.1. PROCURA DO ÓPTIMO PARA AS VÁRIAS COMBINAÇÕES DE  $\mathcal{F}255$

$x_2$	$x_3$	$x_5$	$x_6$	$x_8$	$x_9$	$x_{10}$	$x_1^*$	$ev_4$
1.00	1.50	0.50	1.00	1.50	1.00	1.50	1.25	305.41
1.00	1.50	0.50	1.00	1.50	1.50	0.50	1.25	291.23
1.00	1.50	0.50	1.00	1.50	1.50	1.00	1.25	286.13
1.00	1.50	0.50	1.00	1.50	1.50	1.50	1.25	287.33
1.00	1.50	0.50	1.50	0.50	0.50	0.50	1.25	472.11
1.00	1.50	0.50	1.50	0.50	0.50	1.00	1.25	465.11
1.00	1.50	0.50	1.50	0.50	0.50	1.50	1.25	465.54
1.00	1.50	0.50	1.50	0.50	1.00	0.50	1.25	357.67
1.00	1.50	0.50	1.50	0.50	1.00	1.00	1.25	346.37
1.00	1.50	0.50	1.50	0.50	1.00	1.50	1.25	345.49
1.00	1.50	0.50	1.50	0.50	1.50	0.50	1.25	350.23
1.00	1.50	0.50	1.50	0.50	1.50	1.00	1.25	337.45
1.00	1.50	0.50	1.50	0.50	1.50	1.50	1.25	336.04
1.00	1.50	0.50	1.50	1.00	0.50	0.50	1.25	439.51
1.00	1.50	0.50	1.50	1.00	0.50	1.00	1.25	437.15
1.00	1.50	0.50	1.50	1.00	0.50	1.50	1.25	439.18
1.00	1.50	0.50	1.50	1.00	1.00	0.50	1.25	312.72
1.00	1.50	0.50	1.50	1.00	1.00	1.00	1.25	307.81
1.00	1.50	0.50	1.50	1.00	1.00	1.50	1.25	309.14
1.00	1.50	0.50	1.50	1.00	1.50	0.50	1.25	297.51
1.00	1.50	0.50	1.50	1.00	1.50	1.00	1.25	290.73
1.00	1.50	0.50	1.50	1.00	1.50	1.50	1.25	291.35
1.00	1.50	0.50	1.50	1.50	0.50	0.50	1.25	444.86
1.00	1.50	0.50	1.50	1.50	0.50	1.00	1.25	443.33
1.00	1.50	0.50	1.50	1.50	0.50	1.50	1.25	445.69
1.00	1.50	0.50	1.50	1.50	1.00	0.50	1.25	314.53
1.00	1.50	0.50	1.50	1.50	1.00	1.00	1.25	310.40
1.00	1.50	0.50	1.50	1.50	1.00	1.50	1.25	312.07
1.00	1.50	0.50	1.50	1.50	1.50	0.50	1.25	297.25
1.00	1.50	0.50	1.50	1.50	1.50	1.00	1.25	291.49
1.00	1.50	0.50	1.50	1.50	1.50	1.50	1.25	292.52
1.00	1.50	1.00	0.50	0.50	0.50	0.50	1.00	493.74
1.00	1.50	1.00	0.50	0.50	0.50	1.00	1.00	489.26
1.00	1.50	1.00	0.50	0.50	0.50	1.50	1.00	490.39
1.00	1.50	1.00	0.50	0.50	1.00	0.50	1.00	393.45
1.00	1.50	1.00	0.50	0.50	1.00	1.00	1.00	385.97
1.00	1.50	1.00	0.50	0.50	1.00	1.50	1.00	386.12
1.00	1.50	1.00	0.50	0.50	1.50	0.50	1.25	393.45
1.00	1.50	1.00	0.50	0.50	1.50	1.00	1.25	384.72
1.00	1.50	1.00	0.50	0.50	1.50	1.50	1.25	384.49
1.00	1.50	1.00	0.50	1.00	0.50	0.50	1.00	471.69
1.00	1.50	1.00	0.50	1.00	0.50	1.00	1.00	471.05
1.00	1.50	1.00	0.50	1.00	0.50	1.50	1.25	473.48
1.00	1.50	1.00	0.50	1.00	1.00	0.50	1.25	360.27
1.00	1.50	1.00	0.50	1.00	1.00	1.00	1.25	358.05
1.00	1.50	1.00	0.50	1.00	1.00	1.50	1.25	360.04

$x_2$	$x_3$	$x_5$	$x_6$	$x_8$	$x_9$	$x_{10}$	$x_{11}^*$	$ev_4$
1.00	1.50	1.00	0.50	1.00	1.50	0.50	1.25	354.03
1.00	1.50	1.00	0.50	1.00	1.50	1.00	1.25	350.60
1.00	1.50	1.00	0.50	1.00	1.50	1.50	1.25	352.08
1.00	1.50	1.00	0.50	1.50	0.50	0.50	1.00	478.79
1.00	1.50	1.00	0.50	1.50	0.50	1.00	1.25	478.90
1.00	1.50	1.00	0.50	1.50	0.50	1.50	1.25	481.60
1.00	1.50	1.00	0.50	1.50	1.00	0.50	1.25	365.16
1.00	1.50	1.00	0.50	1.50	1.00	1.00	1.25	363.35
1.00	1.50	1.00	0.50	1.50	1.00	1.50	1.25	365.54
1.00	1.50	1.00	0.50	1.50	1.50	0.50	1.25	357.30
1.00	1.50	1.00	0.50	1.50	1.50	1.00	1.25	354.35
1.00	1.50	1.00	0.50	1.50	1.50	1.50	1.25	356.09
1.00	1.50	1.00	1.00	0.50	0.50	0.50	1.25	464.68
1.00	1.50	1.00	1.00	0.50	0.50	1.00	1.25	458.19
1.00	1.50	1.00	1.00	0.50	0.50	1.50	1.25	458.73
1.00	1.50	1.00	1.00	0.50	1.00	0.50	1.25	351.08
1.00	1.50	1.00	1.00	0.50	1.00	1.00	1.25	340.54
1.00	1.50	1.00	1.00	0.50	1.00	1.50	1.25	339.84
1.00	1.50	1.00	1.00	0.50	1.50	0.50	1.25	344.53
1.00	1.50	1.00	1.00	0.50	1.50	1.00	1.25	332.68
1.00	1.50	1.00	1.00	0.50	1.50	1.50	1.25	331.51
1.00	1.50	1.00	1.00	1.00	0.50	0.50	1.25	432.93
1.00	1.50	1.00	1.00	1.00	0.50	1.00	1.25	431.18
1.00	1.50	1.00	1.00	1.00	0.50	1.50	1.25	433.35
1.00	1.50	1.00	1.00	1.00	1.00	0.50	1.25	307.02
1.00	1.50	1.00	1.00	1.00	1.00	1.00	1.25	303.04
1.00	1.50	1.00	1.00	1.00	1.00	1.50	1.25	304.59
1.00	1.50	1.00	1.00	1.00	1.50	0.50	1.25	292.94
1.00	1.50	1.00	1.00	1.00	1.50	1.00	1.25	287.33
1.00	1.50	1.00	1.00	1.00	1.50	1.50	1.25	288.24
1.00	1.50	1.00	1.00	1.50	0.50	0.50	1.25	438.44
1.00	1.50	1.00	1.00	1.50	0.50	1.00	1.25	437.58
1.00	1.50	1.00	1.00	1.50	0.50	1.50	1.25	440.08
1.00	1.50	1.00	1.00	1.50	1.00	0.50	1.25	309.14
1.00	1.50	1.00	1.00	1.50	1.00	1.00	1.25	305.97
1.00	1.50	1.00	1.00	1.50	1.00	1.50	1.25	307.86
1.00	1.50	1.00	1.00	1.50	1.50	0.50	1.25	293.01
1.00	1.50	1.00	1.00	1.50	1.50	1.00	1.25	288.44
1.00	1.50	1.00	1.00	1.50	1.50	1.50	1.25	289.78
1.00	1.50	1.00	1.50	0.50	0.50	0.50	1.00	474.21
1.00	1.50	1.00	1.50	0.50	0.50	1.00	1.25	467.52
1.00	1.50	1.00	1.50	0.50	0.50	1.50	1.25	468.01
1.00	1.50	1.00	1.50	0.50	1.00	0.50	1.25	359.67
1.00	1.50	1.00	1.50	0.50	1.00	1.00	1.25	348.73
1.00	1.50	1.00	1.50	0.50	1.00	1.50	1.25	347.95
1.00	1.50	1.00	1.50	0.50	1.50	0.50	1.25	352.10

D.1. PROCURA DO ÓPTIMO PARA AS VÁRIAS COMBINAÇÕES DE F257

$x_2$	$x_3$	$x_5$	$x_6$	$x_8$	$x_9$	$x_{10}$	$x_1^*$	$ev_4$
1.00	1.50	1.00	1.50	0.50	1.50	1.00	1.25	339.79
1.00	1.50	1.00	1.50	0.50	1.50	1.50	1.25	338.50
1.00	1.50	1.00	1.50	1.00	0.50	0.50	1.25	441.49
1.00	1.50	1.00	1.50	1.00	0.50	1.00	1.25	439.51
1.00	1.50	1.00	1.50	1.00	0.50	1.50	1.25	441.64
1.00	1.50	1.00	1.50	1.00	1.00	0.50	1.25	314.46
1.00	1.50	1.00	1.50	1.00	1.00	1.00	1.25	310.11
1.00	1.50	1.00	1.50	1.00	1.00	1.50	1.25	311.58
1.00	1.50	1.00	1.50	1.00	1.50	0.50	1.25	299.05
1.00	1.50	1.00	1.50	1.00	1.50	1.00	1.25	292.99
1.00	1.50	1.00	1.50	1.00	1.50	1.50	1.25	293.78
1.00	1.50	1.00	1.50	1.50	0.50	0.50	1.25	446.84
1.00	1.50	1.00	1.50	1.50	0.50	1.00	1.25	445.70
1.00	1.50	1.00	1.50	1.50	0.50	1.50	1.25	448.15
1.00	1.50	1.00	1.50	1.50	1.00	0.50	1.25	316.28
1.00	1.50	1.00	1.50	1.50	1.00	1.00	1.25	312.70
1.00	1.50	1.00	1.50	1.50	1.00	1.50	1.25	314.50
1.00	1.50	1.00	1.50	1.50	1.50	0.50	1.25	298.80
1.00	1.50	1.00	1.50	1.50	1.50	1.00	1.25	293.75
1.00	1.50	1.00	1.50	1.50	1.50	1.50	1.25	294.96
1.00	1.50	1.50	0.50	0.50	0.50	0.50	1.00	496.24
1.00	1.50	1.50	0.50	0.50	0.50	1.00	1.00	491.76
1.00	1.50	1.50	0.50	0.50	0.50	1.50	1.00	492.89
1.00	1.50	1.50	0.50	0.50	1.00	0.50	1.00	395.95
1.00	1.50	1.50	0.50	0.50	1.00	1.00	1.00	388.47
1.00	1.50	1.50	0.50	0.50	1.00	1.50	1.00	388.62
1.00	1.50	1.50	0.50	0.50	1.50	0.50	1.25	395.95
1.00	1.50	1.50	0.50	0.50	1.50	1.00	1.25	387.22
1.00	1.50	1.50	0.50	0.50	1.50	1.50	1.25	386.99
1.00	1.50	1.50	0.50	1.00	0.50	0.50	1.00	474.19
1.00	1.50	1.50	0.50	1.00	0.50	1.00	1.00	473.55
1.00	1.50	1.50	0.50	1.00	0.50	1.50	1.25	475.98
1.00	1.50	1.50	0.50	1.00	1.00	0.50	1.25	362.77
1.00	1.50	1.50	0.50	1.00	1.00	1.00	1.25	360.55
1.00	1.50	1.50	0.50	1.00	1.00	1.50	1.25	362.54
1.00	1.50	1.50	0.50	1.00	1.50	0.50	1.25	356.53
1.00	1.50	1.50	0.50	1.00	1.50	1.00	1.25	353.10
1.00	1.50	1.50	0.50	1.00	1.50	1.50	1.25	354.58
1.00	1.50	1.50	0.50	1.50	0.50	0.50	1.00	481.29
1.00	1.50	1.50	0.50	1.50	0.50	1.00	1.25	481.40
1.00	1.50	1.50	0.50	1.50	0.50	1.50	1.25	484.10
1.00	1.50	1.50	0.50	1.50	1.00	0.50	1.25	367.66
1.00	1.50	1.50	0.50	1.50	1.00	1.00	1.25	365.85
1.00	1.50	1.50	0.50	1.50	1.00	1.50	1.25	368.04
1.00	1.50	1.50	0.50	1.50	1.50	0.50	1.25	359.80
1.00	1.50	1.50	0.50	1.50	1.50	1.00	1.25	356.85

$x_2$	$x_3$	$x_5$	$x_6$	$x_8$	$x_9$	$x_{10}$	$x_1^*$	$ev_4$
1.00	1.50	1.50	0.50	1.50	1.50	1.50	1.25	358.59
1.00	1.50	1.50	1.00	0.50	0.50	0.50	1.25	467.18
1.00	1.50	1.50	1.00	0.50	0.50	1.00	1.25	460.69
1.00	1.50	1.50	1.00	0.50	0.50	1.50	1.25	461.23
1.00	1.50	1.50	1.00	0.50	1.00	0.50	1.25	353.58
1.00	1.50	1.50	1.00	0.50	1.00	1.00	1.25	343.04
1.00	1.50	1.50	1.00	0.50	1.00	1.50	1.25	342.34
1.00	1.50	1.50	1.00	0.50	1.50	0.50	1.25	347.03
1.00	1.50	1.50	1.00	0.50	1.50	1.00	1.25	335.18
1.00	1.50	1.50	1.00	0.50	1.50	1.50	1.25	334.01
1.00	1.50	1.50	1.00	1.00	0.50	0.50	1.25	435.43
1.00	1.50	1.50	1.00	1.00	0.50	1.00	1.25	433.68
1.00	1.50	1.50	1.00	1.00	0.50	1.50	1.25	435.85
1.00	1.50	1.50	1.00	1.00	1.00	0.50	1.25	309.52
1.00	1.50	1.50	1.00	1.00	1.00	1.00	1.25	305.54
1.00	1.50	1.50	1.00	1.00	1.00	1.50	1.25	307.09
1.00	1.50	1.50	1.00	1.00	1.50	0.50	1.25	295.44
1.00	1.50	1.50	1.00	1.00	1.50	1.00	1.25	289.83
1.00	1.50	1.50	1.00	1.00	1.50	1.50	1.25	290.74
1.00	1.50	1.50	1.00	1.50	0.50	0.50	1.25	440.94
1.00	1.50	1.50	1.00	1.50	0.50	1.00	1.25	440.08
1.00	1.50	1.50	1.00	1.50	0.50	1.50	1.25	442.58
1.00	1.50	1.50	1.00	1.50	1.00	0.50	1.25	311.64
1.00	1.50	1.50	1.00	1.50	1.00	1.00	1.25	308.47
1.00	1.50	1.50	1.00	1.50	1.00	1.50	1.25	310.36
1.00	1.50	1.50	1.00	1.50	1.50	0.50	1.25	295.51
1.00	1.50	1.50	1.00	1.50	1.50	1.00	1.25	290.94
1.00	1.50	1.50	1.00	1.50	1.50	1.50	1.25	292.28
1.00	1.50	1.50	1.50	0.50	0.50	0.50	1.00	476.71
1.00	1.50	1.50	1.50	0.50	0.50	1.00	1.25	470.02
1.00	1.50	1.50	1.50	0.50	0.50	1.50	1.25	470.51
1.00	1.50	1.50	1.50	0.50	1.00	0.50	1.25	362.17
1.00	1.50	1.50	1.50	0.50	1.00	1.00	1.25	351.23
1.00	1.50	1.50	1.50	0.50	1.00	1.50	1.25	350.45
1.00	1.50	1.50	1.50	0.50	1.50	0.50	1.25	354.60
1.00	1.50	1.50	1.50	0.50	1.50	1.00	1.25	342.29
1.00	1.50	1.50	1.50	0.50	1.50	1.50	1.25	341.00
1.00	1.50	1.50	1.50	1.00	0.50	0.50	1.25	443.99
1.00	1.50	1.50	1.50	1.00	0.50	1.00	1.25	442.01
1.00	1.50	1.50	1.50	1.00	0.50	1.50	1.25	444.14
1.00	1.50	1.50	1.50	1.00	1.00	0.50	1.25	316.96
1.00	1.50	1.50	1.50	1.00	1.00	1.00	1.25	312.61
1.00	1.50	1.50	1.50	1.00	1.00	1.50	1.25	314.08
1.00	1.50	1.50	1.50	1.00	1.50	0.50	1.25	301.55
1.00	1.50	1.50	1.50	1.00	1.50	1.00	1.25	295.49
1.00	1.50	1.50	1.50	1.00	1.50	1.50	1.25	296.28

D.1. PROCURA DO ÓPTIMO PARA AS VÁRIAS COMBINAÇÕES DE  $\mathcal{F}259$

$x_2$	$x_3$	$x_5$	$x_6$	$x_8$	$x_9$	$x_{10}$	$x_1^*$	$ev_4$
1.00	1.50	1.50	1.50	1.50	0.50	0.50	1.25	449.34
1.00	1.50	1.50	1.50	1.50	0.50	1.00	1.25	448.20
1.00	1.50	1.50	1.50	1.50	0.50	1.50	1.25	450.65
1.00	1.50	1.50	1.50	1.50	1.00	0.50	1.25	318.78
1.00	1.50	1.50	1.50	1.50	1.00	1.00	1.25	315.20
1.00	1.50	1.50	1.50	1.50	1.00	1.50	1.25	317.00
1.00	1.50	1.50	1.50	1.50	1.50	0.50	1.25	301.30
1.00	1.50	1.50	1.50	1.50	1.50	1.00	1.25	296.25
1.00	1.50	1.50	1.50	1.50	1.50	1.50	1.25	297.46
1.50	0.50	0.50	0.50	0.50	0.50	0.50	1.25	596.26
1.50	0.50	0.50	0.50	0.50	0.50	1.00	1.25	588.52
1.50	0.50	0.50	0.50	0.50	0.50	1.50	1.25	588.87
1.50	0.50	0.50	0.50	0.50	1.00	0.50	1.25	483.89
1.50	0.50	0.50	0.50	0.50	1.00	1.00	1.25	471.89
1.50	0.50	0.50	0.50	0.50	1.00	1.50	1.25	470.79
1.50	0.50	0.50	0.50	0.50	1.50	0.50	1.25	478.81
1.50	0.50	0.50	0.50	0.50	1.50	1.00	1.25	464.94
1.50	0.50	0.50	0.50	0.50	1.50	1.50	1.25	463.08
1.50	0.50	0.50	0.50	1.00	0.50	0.50	1.25	560.20
1.50	0.50	0.50	0.50	1.00	0.50	1.00	1.25	555.46
1.50	0.50	0.50	0.50	1.00	0.50	1.50	1.25	556.93
1.50	0.50	0.50	0.50	1.00	1.00	0.50	1.25	432.32
1.50	0.50	0.50	0.50	1.00	1.00	1.00	1.25	424.53
1.50	0.50	0.50	0.50	1.00	1.00	1.50	1.25	425.04
1.50	0.50	0.50	0.50	1.00	1.50	0.50	1.25	420.74
1.50	0.50	0.50	0.50	1.00	1.50	1.00	1.25	410.48
1.50	0.50	0.50	0.50	1.00	1.50	1.50	1.25	409.87
1.50	0.50	0.50	0.50	1.50	0.50	0.50	1.25	555.12
1.50	0.50	0.50	0.50	1.50	0.50	1.00	1.25	552.75
1.50	0.50	0.50	0.50	1.50	0.50	1.50	1.25	555.00
1.50	0.50	0.50	0.50	1.50	1.00	0.50	1.25	419.58
1.50	0.50	0.50	0.50	1.50	1.00	1.00	1.25	413.20
1.50	0.50	0.50	0.50	1.50	1.00	1.50	1.25	414.37
1.50	0.50	0.50	0.50	1.50	1.50	0.50	1.25	403.00
1.50	0.50	0.50	0.50	1.50	1.50	1.00	1.25	394.53
1.50	0.50	0.50	0.50	1.50	1.50	1.50	1.25	394.74
1.50	0.50	0.50	1.00	0.50	0.50	0.50	1.25	580.46
1.50	0.50	0.50	1.00	0.50	0.50	1.00	1.25	571.27
1.50	0.50	0.50	1.00	0.50	0.50	1.50	1.25	571.25
1.50	0.50	0.50	1.00	0.50	1.00	0.50	1.25	459.04
1.50	0.50	0.50	1.00	0.50	1.00	1.00	1.25	444.11
1.50	0.50	0.50	1.00	0.50	1.00	1.50	1.25	442.28
1.50	0.50	0.50	1.00	0.50	1.50	0.50	1.25	447.52
1.50	0.50	0.50	1.00	0.50	1.50	1.00	1.25	430.27
1.50	0.50	0.50	1.00	0.50	1.50	1.50	1.25	427.48
1.50	0.50	0.50	1.00	1.00	0.50	0.50	1.25	538.70

$x_2$	$x_3$	$x_5$	$x_6$	$x_8$	$x_9$	$x_{10}$	$x_{11}^*$	$ev_4$
1.50	0.50	0.50	1.00	1.00	0.50	1.00	1.25	532.18
1.50	0.50	0.50	1.00	1.00	0.50	1.50	1.25	533.18
1.50	0.50	0.50	1.00	1.00	1.00	0.50	1.25	400.98
1.50	0.50	0.50	1.00	1.00	1.00	1.00	1.25	389.85
1.50	0.50	0.50	1.00	1.00	1.00	1.50	1.25	389.48
1.50	0.50	0.50	1.00	1.00	1.50	0.50	1.25	381.90
1.50	0.50	0.50	1.00	1.00	1.50	1.00	1.25	367.52
1.50	0.50	0.50	1.00	1.00	1.50	1.50	1.25	365.69
1.50	0.50	0.50	1.00	1.50	0.50	0.50	1.25	529.99
1.50	0.50	0.50	1.00	1.50	0.50	1.00	1.25	525.93
1.50	0.50	0.50	1.00	1.50	0.50	1.50	1.25	527.77
1.50	0.50	0.50	1.00	1.50	1.00	0.50	1.25	381.62
1.50	0.50	0.50	1.00	1.50	1.00	1.00	1.25	372.58
1.50	0.50	0.50	1.00	1.50	1.00	1.50	1.25	373.03
1.50	0.50	0.50	1.00	1.50	1.50	0.50	1.25	356.50
1.50	0.50	0.50	1.00	1.50	1.50	1.00	1.25	344.48
1.50	0.50	0.50	1.00	1.50	1.50	1.50	1.25	343.75
1.50	0.50	0.50	1.50	0.50	0.50	0.50	1.25	591.14
1.50	0.50	0.50	1.50	0.50	0.50	1.00	1.25	581.67
1.50	0.50	0.50	1.50	0.50	0.50	1.50	1.25	581.61
1.50	0.50	0.50	1.50	0.50	1.00	0.50	1.25	468.98
1.50	0.50	0.50	1.50	0.50	1.00	1.00	1.25	453.55
1.50	0.50	0.50	1.50	0.50	1.00	1.50	1.25	451.62
1.50	0.50	0.50	1.50	0.50	1.50	0.50	1.25	456.97
1.50	0.50	0.50	1.50	0.50	1.50	1.00	1.25	439.13
1.50	0.50	0.50	1.50	0.50	1.50	1.50	1.25	436.20
1.50	0.50	0.50	1.50	1.00	0.50	0.50	1.25	549.02
1.50	0.50	0.50	1.50	1.00	0.50	1.00	1.25	542.22
1.50	0.50	0.50	1.50	1.00	0.50	1.50	1.25	543.18
1.50	0.50	0.50	1.50	1.00	1.00	0.50	1.25	410.53
1.50	0.50	0.50	1.50	1.00	1.00	1.00	1.25	398.90
1.50	0.50	0.50	1.50	1.00	1.00	1.50	1.25	398.42
1.50	0.50	0.50	1.50	1.00	1.50	0.50	1.25	390.86
1.50	0.50	0.50	1.50	1.00	1.50	1.00	1.25	375.87
1.50	0.50	0.50	1.50	1.00	1.50	1.50	1.25	373.90
1.50	0.50	0.50	1.50	1.50	0.50	0.50	1.25	539.97
1.50	0.50	0.50	1.50	1.50	0.50	1.00	1.25	535.54
1.50	0.50	0.50	1.50	1.50	0.50	1.50	1.25	537.31
1.50	0.50	0.50	1.50	1.50	1.00	0.50	1.25	390.51
1.50	0.50	0.50	1.50	1.50	1.00	1.00	1.25	380.90
1.50	0.50	0.50	1.50	1.50	1.00	1.50	1.25	381.24
1.50	0.50	0.50	1.50	1.50	1.50	0.50	1.25	364.76
1.50	0.50	0.50	1.50	1.50	1.50	1.00	1.25	352.09
1.50	0.50	0.50	1.50	1.50	1.50	1.50	1.25	351.21
1.50	0.50	1.00	0.50	0.50	0.50	0.50	1.00	598.67
1.50	0.50	1.00	0.50	0.50	0.50	1.00	1.25	591.00

D.1. PROCURA DO ÓPTIMO PARA AS VÁRIAS COMBINAÇÕES DE  $\mathcal{F}261$

$x_2$	$x_3$	$x_5$	$x_6$	$x_8$	$x_9$	$x_{10}$	$x_1^*$	$ev_4$
1.50	0.50	1.00	0.50	0.50	0.50	1.50	1.25	591.36
1.50	0.50	1.00	0.50	0.50	1.00	0.50	1.25	486.27
1.50	0.50	1.00	0.50	0.50	1.00	1.00	1.25	474.36
1.50	0.50	1.00	0.50	0.50	1.00	1.50	1.25	473.28
1.50	0.50	1.00	0.50	0.50	1.50	0.50	1.25	481.15
1.50	0.50	1.00	0.50	0.50	1.50	1.00	1.25	467.40
1.50	0.50	1.00	0.50	0.50	1.50	1.50	1.25	465.57
1.50	0.50	1.00	0.50	1.00	0.50	0.50	1.25	562.57
1.50	0.50	1.00	0.50	1.00	0.50	1.00	1.25	557.93
1.50	0.50	1.00	0.50	1.00	0.50	1.50	1.25	559.42
1.50	0.50	1.00	0.50	1.00	1.00	0.50	1.25	434.62
1.50	0.50	1.00	0.50	1.00	1.00	1.00	1.25	426.98
1.50	0.50	1.00	0.50	1.00	1.00	1.50	1.25	427.52
1.50	0.50	1.00	0.50	1.00	1.50	0.50	1.25	423.00
1.50	0.50	1.00	0.50	1.00	1.50	1.00	1.25	412.92
1.50	0.50	1.00	0.50	1.00	1.50	1.50	1.25	412.35
1.50	0.50	1.00	0.50	1.50	0.50	0.50	1.25	557.50
1.50	0.50	1.00	0.50	1.50	0.50	1.00	1.25	555.22
1.50	0.50	1.00	0.50	1.50	0.50	1.50	1.25	557.49
1.50	0.50	1.00	0.50	1.50	1.00	0.50	1.25	421.88
1.50	0.50	1.00	0.50	1.50	1.00	1.00	1.25	415.65
1.50	0.50	1.00	0.50	1.50	1.00	1.50	1.25	416.85
1.50	0.50	1.00	0.50	1.50	1.50	0.50	1.25	405.25
1.50	0.50	1.00	0.50	1.50	1.50	1.00	1.25	396.97
1.50	0.50	1.00	0.50	1.50	1.50	1.50	1.25	397.23
1.50	0.50	1.00	1.00	0.50	0.50	0.50	1.25	582.88
1.50	0.50	1.00	1.00	0.50	0.50	1.00	1.25	573.75
1.50	0.50	1.00	1.00	0.50	0.50	1.50	1.25	573.75
1.50	0.50	1.00	1.00	0.50	1.00	0.50	1.25	461.42
1.50	0.50	1.00	1.00	0.50	1.00	1.00	1.25	446.57
1.50	0.50	1.00	1.00	0.50	1.00	1.50	1.25	444.77
1.50	0.50	1.00	1.00	0.50	1.50	0.50	1.25	449.86
1.50	0.50	1.00	1.00	0.50	1.50	1.00	1.25	432.73
1.50	0.50	1.00	1.00	0.50	1.50	1.50	1.25	429.97
1.50	0.50	1.00	1.00	1.00	0.50	0.50	1.25	541.07
1.50	0.50	1.00	1.00	1.00	0.50	1.00	1.25	534.65
1.50	0.50	1.00	1.00	1.00	0.50	1.50	1.25	535.67
1.50	0.50	1.00	1.00	1.00	1.00	0.50	1.25	403.29
1.50	0.50	1.00	1.00	1.00	1.00	1.00	1.25	392.30
1.50	0.50	1.00	1.00	1.00	1.00	1.50	1.25	391.97
1.50	0.50	1.00	1.00	1.00	1.50	0.50	1.25	384.16
1.50	0.50	1.00	1.00	1.00	1.50	1.00	1.25	369.95
1.50	0.50	1.00	1.00	1.00	1.50	1.50	1.25	368.18
1.50	0.50	1.00	1.00	1.50	0.50	0.50	1.25	532.36
1.50	0.50	1.00	1.00	1.50	0.50	1.00	1.25	528.40
1.50	0.50	1.00	1.00	1.50	0.50	1.50	1.25	530.26

$x_2$	$x_3$	$x_5$	$x_6$	$x_8$	$x_9$	$x_{10}$	$x_{11}^*$	$ev_4$
1.50	0.50	1.00	1.00	1.50	1.00	0.50	1.25	383.93
1.50	0.50	1.00	1.00	1.50	1.00	1.00	1.25	375.03
1.50	0.50	1.00	1.00	1.50	1.00	1.50	1.25	375.51
1.50	0.50	1.00	1.00	1.50	1.50	0.50	1.25	358.76
1.50	0.50	1.00	1.00	1.50	1.50	1.00	1.25	346.91
1.50	0.50	1.00	1.00	1.50	1.50	1.50	1.25	346.23
1.50	0.50	1.00	1.50	0.50	0.50	0.50	1.25	593.53
1.50	0.50	1.00	1.50	0.50	0.50	1.00	1.25	584.14
1.50	0.50	1.00	1.50	0.50	0.50	1.50	1.25	584.10
1.50	0.50	1.00	1.50	0.50	1.00	0.50	1.25	471.31
1.50	0.50	1.00	1.50	0.50	1.00	1.00	1.25	456.00
1.50	0.50	1.00	1.50	0.50	1.00	1.50	1.25	454.10
1.50	0.50	1.00	1.50	0.50	1.50	0.50	1.25	459.26
1.50	0.50	1.00	1.50	0.50	1.50	1.00	1.25	441.57
1.50	0.50	1.00	1.50	0.50	1.50	1.50	1.25	438.69
1.50	0.50	1.00	1.50	1.00	0.50	0.50	1.25	551.35
1.50	0.50	1.00	1.50	1.00	0.50	1.00	1.25	544.68
1.50	0.50	1.00	1.50	1.00	0.50	1.50	1.25	545.66
1.50	0.50	1.00	1.50	1.00	1.00	0.50	1.25	412.77
1.50	0.50	1.00	1.50	1.00	1.00	1.00	1.25	401.33
1.50	0.50	1.00	1.50	1.00	1.00	1.50	1.25	400.90
1.50	0.50	1.00	1.50	1.00	1.50	0.50	1.25	393.04
1.50	0.50	1.00	1.50	1.00	1.50	1.00	1.25	378.29
1.50	0.50	1.00	1.50	1.00	1.50	1.50	1.25	376.38
1.50	0.50	1.00	1.50	1.50	0.50	0.50	1.25	542.30
1.50	0.50	1.00	1.50	1.50	0.50	1.00	1.25	537.99
1.50	0.50	1.00	1.50	1.50	0.50	1.50	1.25	539.79
1.50	0.50	1.00	1.50	1.50	1.00	0.50	1.25	392.76
1.50	0.50	1.00	1.50	1.50	1.00	1.00	1.25	383.34
1.50	0.50	1.00	1.50	1.50	1.00	1.50	1.25	383.72
1.50	0.50	1.00	1.50	1.50	1.50	0.50	1.25	366.94
1.50	0.50	1.00	1.50	1.50	1.50	1.00	1.25	354.51
1.50	0.50	1.00	1.50	1.50	1.50	1.50	1.25	353.68
1.50	0.50	1.50	0.50	0.50	0.50	0.50	1.00	601.17
1.50	0.50	1.50	0.50	0.50	0.50	1.00	1.25	593.50
1.50	0.50	1.50	0.50	0.50	0.50	1.50	1.25	593.86
1.50	0.50	1.50	0.50	0.50	1.00	0.50	1.25	488.77
1.50	0.50	1.50	0.50	0.50	1.00	1.00	1.25	476.86
1.50	0.50	1.50	0.50	0.50	1.00	1.50	1.25	475.78
1.50	0.50	1.50	0.50	0.50	1.50	0.50	1.25	483.65
1.50	0.50	1.50	0.50	0.50	1.50	1.00	1.25	469.90
1.50	0.50	1.50	0.50	0.50	1.50	1.50	1.25	468.07
1.50	0.50	1.50	0.50	1.00	0.50	0.50	1.25	565.07
1.50	0.50	1.50	0.50	1.00	0.50	1.00	1.25	560.43
1.50	0.50	1.50	0.50	1.00	0.50	1.50	1.25	561.92
1.50	0.50	1.50	0.50	1.00	1.00	0.50	1.25	437.12



D.1. PROCURA DO ÓPTIMO PARA AS VÁRIAS COMBINAÇÕES DE F263

$x_2$	$x_3$	$x_5$	$x_6$	$x_8$	$x_9$	$x_{10}$	$x_1^*$	$ev_4$
1.50	0.50	1.50	0.50	1.00	1.00	1.00	1.25	429.48
1.50	0.50	1.50	0.50	1.00	1.00	1.50	1.25	430.02
1.50	0.50	1.50	0.50	1.00	1.50	0.50	1.25	425.50
1.50	0.50	1.50	0.50	1.00	1.50	1.00	1.25	415.42
1.50	0.50	1.50	0.50	1.00	1.50	1.50	1.25	414.85
1.50	0.50	1.50	0.50	1.50	0.50	0.50	1.25	560.00
1.50	0.50	1.50	0.50	1.50	0.50	1.00	1.25	557.72
1.50	0.50	1.50	0.50	1.50	0.50	1.50	1.25	559.99
1.50	0.50	1.50	0.50	1.50	1.00	0.50	1.25	424.38
1.50	0.50	1.50	0.50	1.50	1.00	1.00	1.25	418.15
1.50	0.50	1.50	0.50	1.50	1.00	1.50	1.25	419.35
1.50	0.50	1.50	0.50	1.50	1.50	0.50	1.25	407.75
1.50	0.50	1.50	0.50	1.50	1.50	1.00	1.25	399.47
1.50	0.50	1.50	0.50	1.50	1.50	1.50	1.25	399.73
1.50	0.50	1.50	1.00	0.50	0.50	0.50	1.25	585.38
1.50	0.50	1.50	1.00	0.50	0.50	1.00	1.25	576.25
1.50	0.50	1.50	1.00	0.50	0.50	1.50	1.25	576.25
1.50	0.50	1.50	1.00	0.50	1.00	0.50	1.25	463.92
1.50	0.50	1.50	1.00	0.50	1.00	1.00	1.25	449.07
1.50	0.50	1.50	1.00	0.50	1.00	1.50	1.25	447.27
1.50	0.50	1.50	1.00	0.50	1.50	0.50	1.25	452.36
1.50	0.50	1.50	1.00	0.50	1.50	1.00	1.25	435.23
1.50	0.50	1.50	1.00	0.50	1.50	1.50	1.25	432.47
1.50	0.50	1.50	1.00	1.00	0.50	0.50	1.25	543.57
1.50	0.50	1.50	1.00	1.00	0.50	1.00	1.25	537.15
1.50	0.50	1.50	1.00	1.00	0.50	1.50	1.25	538.17
1.50	0.50	1.50	1.00	1.00	1.00	0.50	1.25	405.79
1.50	0.50	1.50	1.00	1.00	1.00	1.00	1.25	394.80
1.50	0.50	1.50	1.00	1.00	1.00	1.50	1.25	394.47
1.50	0.50	1.50	1.00	1.00	1.50	0.50	1.25	386.66
1.50	0.50	1.50	1.00	1.00	1.50	1.00	1.25	372.45
1.50	0.50	1.50	1.00	1.00	1.50	1.50	1.25	370.68
1.50	0.50	1.50	1.00	1.50	0.50	0.50	1.25	534.86
1.50	0.50	1.50	1.00	1.50	0.50	1.00	1.25	530.90
1.50	0.50	1.50	1.00	1.50	0.50	1.50	1.25	532.76
1.50	0.50	1.50	1.00	1.50	1.00	0.50	1.25	386.43
1.50	0.50	1.50	1.00	1.50	1.00	1.00	1.25	377.53
1.50	0.50	1.50	1.00	1.50	1.00	1.50	1.25	378.01
1.50	0.50	1.50	1.00	1.50	1.50	0.50	1.25	361.26
1.50	0.50	1.50	1.00	1.50	1.50	1.00	1.25	349.41
1.50	0.50	1.50	1.00	1.50	1.50	1.50	1.25	348.73
1.50	0.50	1.50	1.50	0.50	0.50	0.50	1.25	596.03
1.50	0.50	1.50	1.50	0.50	0.50	1.00	1.25	586.64
1.50	0.50	1.50	1.50	0.50	0.50	1.50	1.25	586.60
1.50	0.50	1.50	1.50	0.50	1.00	0.50	1.25	473.81
1.50	0.50	1.50	1.50	0.50	1.00	1.00	1.25	458.50

$x_2$	$x_3$	$x_5$	$x_6$	$x_8$	$x_9$	$x_{10}$	$x_1^*$	$ev_4$
1.50	0.50	1.50	1.50	0.50	1.00	1.50	1.25	456.60
1.50	0.50	1.50	1.50	0.50	1.50	0.50	1.25	461.76
1.50	0.50	1.50	1.50	0.50	1.50	1.00	1.25	444.07
1.50	0.50	1.50	1.50	0.50	1.50	1.50	1.25	441.19
1.50	0.50	1.50	1.50	1.00	0.50	0.50	1.25	553.85
1.50	0.50	1.50	1.50	1.00	0.50	1.00	1.25	547.18
1.50	0.50	1.50	1.50	1.00	0.50	1.50	1.25	548.16
1.50	0.50	1.50	1.50	1.00	1.00	0.50	1.25	415.27
1.50	0.50	1.50	1.50	1.00	1.00	1.00	1.25	403.83
1.50	0.50	1.50	1.50	1.00	1.00	1.50	1.25	403.40
1.50	0.50	1.50	1.50	1.00	1.50	0.50	1.25	395.54
1.50	0.50	1.50	1.50	1.00	1.50	1.00	1.25	380.79
1.50	0.50	1.50	1.50	1.00	1.50	1.50	1.25	378.88
1.50	0.50	1.50	1.50	1.50	0.50	0.50	1.25	544.80
1.50	0.50	1.50	1.50	1.50	0.50	1.00	1.25	540.49
1.50	0.50	1.50	1.50	1.50	0.50	1.50	1.25	542.29
1.50	0.50	1.50	1.50	1.50	1.00	0.50	1.25	395.26
1.50	0.50	1.50	1.50	1.50	1.00	1.00	1.25	385.84
1.50	0.50	1.50	1.50	1.50	1.00	1.50	1.25	386.22
1.50	0.50	1.50	1.50	1.50	1.50	0.50	1.25	369.44
1.50	0.50	1.50	1.50	1.50	1.50	1.00	1.25	357.01
1.50	0.50	1.50	1.50	1.50	1.50	1.50	1.25	356.18
1.50	1.00	0.50	0.50	0.50	0.50	0.50	1.25	510.68
1.50	1.00	0.50	0.50	0.50	0.50	1.00	1.25	505.00
1.50	1.00	0.50	0.50	0.50	0.50	1.50	1.25	505.93
1.50	1.00	0.50	0.50	0.50	1.00	0.50	1.25	405.83
1.50	1.00	0.50	0.50	0.50	1.00	1.00	1.25	396.59
1.50	1.00	0.50	0.50	0.50	1.00	1.50	1.25	396.34
1.50	1.00	0.50	0.50	0.50	1.50	0.50	1.25	403.77
1.50	1.00	0.50	0.50	0.50	1.50	1.00	1.25	393.19
1.50	1.00	0.50	0.50	0.50	1.50	1.50	1.25	392.41
1.50	1.00	0.50	0.50	1.00	0.50	0.50	1.25	485.35
1.50	1.00	0.50	0.50	1.00	0.50	1.00	1.25	482.65
1.50	1.00	0.50	0.50	1.00	0.50	1.50	1.25	484.65
1.50	1.00	0.50	0.50	1.00	1.00	0.50	1.25	368.57
1.50	1.00	0.50	0.50	1.00	1.00	1.00	1.25	363.64
1.50	1.00	0.50	0.50	1.00	1.00	1.50	1.25	365.00
1.50	1.00	0.50	0.50	1.00	1.50	0.50	1.25	361.30
1.50	1.00	0.50	0.50	1.00	1.50	1.00	1.25	354.76
1.50	1.00	0.50	0.50	1.00	1.50	1.50	1.25	355.29
1.50	1.00	0.50	0.50	1.50	0.50	0.50	1.25	487.04
1.50	1.00	0.50	0.50	1.50	0.50	1.00	1.25	485.83
1.50	1.00	0.50	0.50	1.50	0.50	1.50	1.25	488.39
1.50	1.00	0.50	0.50	1.50	1.00	0.50	1.25	365.83
1.50	1.00	0.50	0.50	1.50	1.00	1.00	1.25	361.50
1.50	1.00	0.50	0.50	1.50	1.00	1.50	1.25	363.22

D.1. PROCURA DO ÓPTIMO PARA AS VÁRIAS COMBINAÇÕES DE  $\mathcal{F}265$

$x_2$	$x_3$	$x_5$	$x_6$	$x_8$	$x_9$	$x_{10}$	$x_1^*$	$ev_4$
1.50	1.00	0.50	0.50	1.50	1.50	0.50	1.25	355.24
1.50	1.00	0.50	0.50	1.50	1.50	1.00	1.25	349.37
1.50	1.00	0.50	0.50	1.50	1.50	1.50	1.25	350.37
1.50	1.00	0.50	1.00	0.50	0.50	0.50	1.25	484.60
1.50	1.00	0.50	1.00	0.50	0.50	1.00	1.25	477.12
1.50	1.00	0.50	1.00	0.50	0.50	1.50	1.25	477.57
1.50	1.00	0.50	1.00	0.50	1.00	0.50	1.25	367.44
1.50	1.00	0.50	1.00	0.50	1.00	1.00	1.25	354.88
1.50	1.00	0.50	1.00	0.50	1.00	1.50	1.25	353.82
1.50	1.00	0.50	1.00	0.50	1.50	0.50	1.25	357.86
1.50	1.00	0.50	1.00	0.50	1.50	1.00	1.25	343.69
1.50	1.00	0.50	1.00	0.50	1.50	1.50	1.25	341.92
1.50	1.00	0.50	1.00	1.00	0.50	0.50	1.25	450.99
1.50	1.00	0.50	1.00	1.00	0.50	1.00	1.25	446.72
1.50	1.00	0.50	1.00	1.00	0.50	1.50	1.25	448.33
1.50	1.00	0.50	1.00	1.00	1.00	0.50	1.25	321.14
1.50	1.00	0.50	1.00	1.00	1.00	1.00	1.25	313.28
1.50	1.00	0.50	1.00	1.00	1.00	1.50	1.25	313.92
1.50	1.00	0.50	1.00	1.00	1.50	0.50	1.25	305.11
1.50	1.00	0.50	1.00	1.00	1.50	1.00	1.25	294.99
1.50	1.00	0.50	1.00	1.00	1.50	1.50	1.25	294.58
1.50	1.00	0.50	1.00	1.50	0.50	0.50	1.25	449.50
1.50	1.00	0.50	1.00	1.50	0.50	1.00	1.25	446.93
1.50	1.00	0.50	1.00	1.50	0.50	1.50	1.25	449.19
1.50	1.00	0.50	1.00	1.50	1.00	0.50	1.25	312.52
1.50	1.00	0.50	1.00	1.50	1.00	1.00	1.25	306.14
1.50	1.00	0.50	1.00	1.50	1.00	1.50	1.25	307.46
1.50	1.00	0.50	1.00	1.50	1.50	0.50	1.25	292.32
1.50	1.00	0.50	1.00	1.50	1.50	1.00	1.25	284.16
1.50	1.00	0.50	1.00	1.50	1.50	1.50	1.25	284.59
1.50	1.00	0.50	1.50	0.50	0.50	0.50	1.25	494.52
1.50	1.00	0.50	1.50	0.50	0.50	1.00	1.25	486.69
1.50	1.00	0.50	1.50	0.50	0.50	1.50	1.25	487.08
1.50	1.00	0.50	1.50	0.50	1.00	0.50	1.25	376.38
1.50	1.00	0.50	1.50	0.50	1.00	1.00	1.25	363.20
1.50	1.00	0.50	1.50	0.50	1.00	1.50	1.25	362.00
1.50	1.00	0.50	1.50	0.50	1.50	0.50	1.25	365.98
1.50	1.00	0.50	1.50	0.50	1.50	1.00	1.25	351.10
1.50	1.00	0.50	1.50	0.50	1.50	1.50	1.25	349.16
1.50	1.00	0.50	1.50	1.00	0.50	0.50	1.25	460.21
1.50	1.00	0.50	1.50	1.00	0.50	1.00	1.25	455.58
1.50	1.00	0.50	1.50	1.00	0.50	1.50	1.25	457.13
1.50	1.00	0.50	1.50	1.00	1.00	0.50	1.25	329.31
1.50	1.00	0.50	1.50	1.00	1.00	1.00	1.25	320.83
1.50	1.00	0.50	1.50	1.00	1.00	1.50	1.25	321.34
1.50	1.00	0.50	1.50	1.00	1.50	0.50	1.25	312.29

$x_2$	$x_3$	$x_5$	$x_6$	$x_8$	$x_9$	$x_{10}$	$x_{11}^*$	$ev_4$
1.50	1.00	0.50	1.50	1.00	1.50	1.00	1.25	301.42
1.50	1.00	0.50	1.50	1.00	1.50	1.50	1.25	300.83
1.50	1.00	0.50	1.50	1.50	0.50	0.50	1.25	458.38
1.50	1.00	0.50	1.50	1.50	0.50	1.00	1.25	455.36
1.50	1.00	0.50	1.50	1.50	0.50	1.50	1.25	457.53
1.50	1.00	0.50	1.50	1.50	1.00	0.50	1.25	320.02
1.50	1.00	0.50	1.50	1.50	1.00	1.00	1.25	312.99
1.50	1.00	0.50	1.50	1.50	1.00	1.50	1.25	314.16
1.50	1.00	0.50	1.50	1.50	1.50	0.50	1.25	298.80
1.50	1.00	0.50	1.50	1.50	1.50	1.00	1.25	289.86
1.50	1.00	0.50	1.50	1.50	1.50	1.50	1.25	290.09
1.50	1.00	1.00	0.50	0.50	0.50	0.50	1.25	513.02
1.50	1.00	1.00	0.50	0.50	0.50	1.00	1.25	507.46
1.50	1.00	1.00	0.50	0.50	0.50	1.50	1.25	508.42
1.50	1.00	1.00	0.50	0.50	1.00	0.50	1.25	408.09
1.50	1.00	1.00	0.50	0.50	1.00	1.00	1.25	399.02
1.50	1.00	1.00	0.50	0.50	1.00	1.50	1.25	398.82
1.50	1.00	1.00	0.50	0.50	1.50	0.50	1.25	405.95
1.50	1.00	1.00	0.50	0.50	1.50	1.00	1.25	395.61
1.50	1.00	1.00	0.50	0.50	1.50	1.50	1.25	394.89
1.50	1.00	1.00	0.50	1.00	0.50	0.50	1.25	487.60
1.50	1.00	1.00	0.50	1.00	0.50	1.00	1.25	485.08
1.50	1.00	1.00	0.50	1.00	0.50	1.50	1.25	487.13
1.50	1.00	1.00	0.50	1.00	1.00	0.50	1.25	370.68
1.50	1.00	1.00	0.50	1.00	1.00	1.00	1.25	366.04
1.50	1.00	1.00	0.50	1.00	1.00	1.50	1.25	367.47
1.50	1.00	1.00	0.50	1.00	1.50	0.50	1.25	363.32
1.50	1.00	1.00	0.50	1.00	1.50	1.00	1.25	357.13
1.50	1.00	1.00	0.50	1.00	1.50	1.50	1.25	357.76
1.50	1.00	1.00	0.50	1.50	0.50	0.50	1.25	489.28
1.50	1.00	1.00	0.50	1.50	0.50	1.00	1.25	488.26
1.50	1.00	1.00	0.50	1.50	0.50	1.50	1.25	490.86
1.50	1.00	1.00	0.50	1.50	1.00	0.50	1.25	367.94
1.50	1.00	1.00	0.50	1.50	1.00	1.00	1.25	363.90
1.50	1.00	1.00	0.50	1.50	1.00	1.50	1.25	365.68
1.50	1.00	1.00	0.50	1.50	1.50	0.50	1.25	357.25
1.50	1.00	1.00	0.50	1.50	1.50	1.00	1.25	351.74
1.50	1.00	1.00	0.50	1.50	1.50	1.50	1.25	352.84
1.50	1.00	1.00	1.00	0.50	0.50	0.50	1.25	486.94
1.50	1.00	1.00	1.00	0.50	0.50	1.00	1.25	479.58
1.50	1.00	1.00	1.00	0.50	0.50	1.50	1.25	480.06
1.50	1.00	1.00	1.00	0.50	1.00	0.50	1.25	369.71
1.50	1.00	1.00	1.00	0.50	1.00	1.00	1.25	357.31
1.50	1.00	1.00	1.00	0.50	1.00	1.50	1.25	356.29
1.50	1.00	1.00	1.00	0.50	1.50	0.50	1.25	360.04
1.50	1.00	1.00	1.00	0.50	1.50	1.00	1.25	346.10

D.1. PROCURA DO ÓPTIMO PARA AS VÁRIAS COMBINAÇÕES DE F267

$x_2$	$x_3$	$x_5$	$x_6$	$x_8$	$x_9$	$x_{10}$	$x_1^*$	$ev_4$
1.50	1.00	1.00	1.00	0.50	1.50	1.50	1.25	344.40
1.50	1.00	1.00	1.00	1.00	0.50	0.50	1.25	453.23
1.50	1.00	1.00	1.00	1.00	0.50	1.00	1.25	449.15
1.50	1.00	1.00	1.00	1.00	0.50	1.50	1.25	450.81
1.50	1.00	1.00	1.00	1.00	1.00	0.50	1.25	323.26
1.50	1.00	1.00	1.00	1.00	1.00	1.00	1.25	315.67
1.50	1.00	1.00	1.00	1.00	1.00	1.50	1.25	316.39
1.50	1.00	1.00	1.00	1.00	1.50	0.50	1.25	307.13
1.50	1.00	1.00	1.00	1.00	1.50	1.00	1.25	297.36
1.50	1.00	1.00	1.00	1.00	1.50	1.50	1.25	297.05
1.50	1.00	1.00	1.00	1.50	0.50	0.50	1.25	451.74
1.50	1.00	1.00	1.00	1.50	0.50	1.00	1.25	449.36
1.50	1.00	1.00	1.00	1.50	0.50	1.50	1.25	451.67
1.50	1.00	1.00	1.00	1.50	1.00	0.50	1.25	314.63
1.50	1.00	1.00	1.00	1.50	1.00	1.00	1.25	308.54
1.50	1.00	1.00	1.00	1.50	1.00	1.50	1.25	309.93
1.50	1.00	1.00	1.00	1.50	1.50	0.50	1.25	294.34
1.50	1.00	1.00	1.00	1.50	1.50	1.00	1.25	286.54
1.50	1.00	1.00	1.00	1.50	1.50	1.50	1.25	287.06
1.50	1.00	1.00	1.50	0.50	0.50	0.50	1.25	496.80
1.50	1.00	1.00	1.50	0.50	0.50	1.00	1.25	489.13
1.50	1.00	1.00	1.50	0.50	0.50	1.50	1.25	489.56
1.50	1.00	1.00	1.50	0.50	1.00	0.50	1.25	378.54
1.50	1.00	1.00	1.50	0.50	1.00	1.00	1.25	365.61
1.50	1.00	1.00	1.50	0.50	1.00	1.50	1.25	364.48
1.50	1.00	1.00	1.50	0.50	1.50	0.50	1.25	368.05
1.50	1.00	1.00	1.50	0.50	1.50	1.00	1.25	353.49
1.50	1.00	1.00	1.50	0.50	1.50	1.50	1.25	351.63
1.50	1.00	1.00	1.50	1.00	0.50	0.50	1.25	462.36
1.50	1.00	1.00	1.50	1.00	0.50	1.00	1.25	457.98
1.50	1.00	1.00	1.50	1.00	0.50	1.50	1.25	459.60
1.50	1.00	1.00	1.50	1.00	1.00	0.50	1.25	331.30
1.50	1.00	1.00	1.50	1.00	1.00	1.00	1.25	323.19
1.50	1.00	1.00	1.50	1.00	1.00	1.50	1.25	323.80
1.50	1.00	1.00	1.50	1.00	1.50	0.50	1.25	314.14
1.50	1.00	1.00	1.50	1.00	1.50	1.00	1.25	303.75
1.50	1.00	1.00	1.50	1.00	1.50	1.50	1.25	303.29
1.50	1.00	1.00	1.50	1.50	0.50	0.50	1.25	460.53
1.50	1.00	1.00	1.50	1.50	0.50	1.00	1.25	457.77
1.50	1.00	1.00	1.50	1.50	0.50	1.50	1.25	460.01
1.50	1.00	1.00	1.50	1.50	1.00	0.50	1.25	322.01
1.50	1.00	1.00	1.50	1.50	1.00	1.00	1.25	315.35
1.50	1.00	1.00	1.50	1.50	1.00	1.50	1.25	316.62
1.50	1.00	1.00	1.50	1.50	1.50	0.50	1.25	300.65
1.50	1.00	1.00	1.50	1.50	1.50	1.00	1.25	292.19
1.50	1.00	1.00	1.50	1.50	1.50	1.50	1.25	292.55

$x_2$	$x_3$	$x_5$	$x_6$	$x_8$	$x_9$	$x_{10}$	$x_1^*$	$ev_4$
1.50	1.00	1.50	0.50	0.50	0.50	0.50	1.25	515.52
1.50	1.00	1.50	0.50	0.50	0.50	1.00	1.25	509.96
1.50	1.00	1.50	0.50	0.50	0.50	1.50	1.25	510.92
1.50	1.00	1.50	0.50	0.50	1.00	0.50	1.25	410.59
1.50	1.00	1.50	0.50	0.50	1.00	1.00	1.25	401.52
1.50	1.00	1.50	0.50	0.50	1.00	1.50	1.25	401.32
1.50	1.00	1.50	0.50	0.50	1.50	0.50	1.25	408.45
1.50	1.00	1.50	0.50	0.50	1.50	1.00	1.25	398.11
1.50	1.00	1.50	0.50	0.50	1.50	1.50	1.25	397.39
1.50	1.00	1.50	0.50	1.00	0.50	0.50	1.25	490.10
1.50	1.00	1.50	0.50	1.00	0.50	1.00	1.25	487.58
1.50	1.00	1.50	0.50	1.00	0.50	1.50	1.25	489.63
1.50	1.00	1.50	0.50	1.00	1.00	0.50	1.25	373.18
1.50	1.00	1.50	0.50	1.00	1.00	1.00	1.25	368.54
1.50	1.00	1.50	0.50	1.00	1.00	1.50	1.25	369.97
1.50	1.00	1.50	0.50	1.00	1.50	0.50	1.25	365.82
1.50	1.00	1.50	0.50	1.00	1.50	1.00	1.25	359.63
1.50	1.00	1.50	0.50	1.00	1.50	1.50	1.25	360.26
1.50	1.00	1.50	0.50	1.50	0.50	0.50	1.25	491.78
1.50	1.00	1.50	0.50	1.50	0.50	1.00	1.25	490.76
1.50	1.00	1.50	0.50	1.50	0.50	1.50	1.25	493.36
1.50	1.00	1.50	0.50	1.50	1.00	0.50	1.25	370.44
1.50	1.00	1.50	0.50	1.50	1.00	1.00	1.25	366.40
1.50	1.00	1.50	0.50	1.50	1.00	1.50	1.25	368.18
1.50	1.00	1.50	0.50	1.50	1.50	0.50	1.25	359.75
1.50	1.00	1.50	0.50	1.50	1.50	1.00	1.25	354.24
1.50	1.00	1.50	0.50	1.50	1.50	1.50	1.25	355.34
1.50	1.00	1.50	1.00	0.50	0.50	0.50	1.25	489.44
1.50	1.00	1.50	1.00	0.50	0.50	1.00	1.25	482.08
1.50	1.00	1.50	1.00	0.50	0.50	1.50	1.25	482.56
1.50	1.00	1.50	1.00	0.50	1.00	0.50	1.25	372.21
1.50	1.00	1.50	1.00	0.50	1.00	1.00	1.25	359.81
1.50	1.00	1.50	1.00	0.50	1.00	1.50	1.25	358.79
1.50	1.00	1.50	1.00	0.50	1.50	0.50	1.25	362.54
1.50	1.00	1.50	1.00	0.50	1.50	1.00	1.25	348.60
1.50	1.00	1.50	1.00	0.50	1.50	1.50	1.25	346.90
1.50	1.00	1.50	1.00	1.00	0.50	0.50	1.25	455.73
1.50	1.00	1.50	1.00	1.00	0.50	1.00	1.25	451.65
1.50	1.00	1.50	1.00	1.00	0.50	1.50	1.25	453.31
1.50	1.00	1.50	1.00	1.00	1.00	0.50	1.25	325.76
1.50	1.00	1.50	1.00	1.00	1.00	1.00	1.25	318.17
1.50	1.00	1.50	1.00	1.00	1.00	1.50	1.25	318.89
1.50	1.00	1.50	1.00	1.00	1.50	0.50	1.25	309.63
1.50	1.00	1.50	1.00	1.00	1.50	1.00	1.25	299.86
1.50	1.00	1.50	1.00	1.00	1.50	1.50	1.25	299.55
1.50	1.00	1.50	1.00	1.50	0.50	0.50	1.25	454.24

D.1. PROCURA DO ÓPTIMO PARA AS VÁRIAS COMBINAÇÕES DE F269

$x_2$	$x_3$	$x_5$	$x_6$	$x_8$	$x_9$	$x_{10}$	$x_1^*$	$ev_4$
1.50	1.00	1.50	1.00	1.50	0.50	1.00	1.25	451.86
1.50	1.00	1.50	1.00	1.50	0.50	1.50	1.25	454.17
1.50	1.00	1.50	1.00	1.50	1.00	0.50	1.25	317.13
1.50	1.00	1.50	1.00	1.50	1.00	1.00	1.25	311.04
1.50	1.00	1.50	1.00	1.50	1.00	1.50	1.25	312.43
1.50	1.00	1.50	1.00	1.50	1.50	0.50	1.25	296.84
1.50	1.00	1.50	1.00	1.50	1.50	1.00	1.25	289.04
1.50	1.00	1.50	1.00	1.50	1.50	1.50	1.25	289.56
1.50	1.00	1.50	1.50	0.50	0.50	0.50	1.25	499.30
1.50	1.00	1.50	1.50	0.50	0.50	1.00	1.25	491.63
1.50	1.00	1.50	1.50	0.50	0.50	1.50	1.25	492.06
1.50	1.00	1.50	1.50	0.50	1.00	0.50	1.25	381.04
1.50	1.00	1.50	1.50	0.50	1.00	1.00	1.25	368.11
1.50	1.00	1.50	1.50	0.50	1.00	1.50	1.25	366.98
1.50	1.00	1.50	1.50	0.50	1.50	0.50	1.25	370.55
1.50	1.00	1.50	1.50	0.50	1.50	1.00	1.25	355.99
1.50	1.00	1.50	1.50	0.50	1.50	1.50	1.25	354.13
1.50	1.00	1.50	1.50	1.00	0.50	0.50	1.25	464.86
1.50	1.00	1.50	1.50	1.00	0.50	1.00	1.25	460.48
1.50	1.00	1.50	1.50	1.00	0.50	1.50	1.25	462.10
1.50	1.00	1.50	1.50	1.00	1.00	0.50	1.25	333.80
1.50	1.00	1.50	1.50	1.00	1.00	1.00	1.25	325.69
1.50	1.00	1.50	1.50	1.00	1.00	1.50	1.25	326.30
1.50	1.00	1.50	1.50	1.00	1.50	0.50	1.25	316.64
1.50	1.00	1.50	1.50	1.00	1.50	1.00	1.25	306.25
1.50	1.00	1.50	1.50	1.00	1.50	1.50	1.25	305.79
1.50	1.00	1.50	1.50	1.50	0.50	0.50	1.25	463.03
1.50	1.00	1.50	1.50	1.50	0.50	1.00	1.25	460.27
1.50	1.00	1.50	1.50	1.50	0.50	1.50	1.25	462.51
1.50	1.00	1.50	1.50	1.50	1.00	0.50	1.25	324.51
1.50	1.00	1.50	1.50	1.50	1.00	1.00	1.25	317.85
1.50	1.00	1.50	1.50	1.50	1.00	1.50	1.25	319.12
1.50	1.00	1.50	1.50	1.50	1.50	0.50	1.25	303.15
1.50	1.00	1.50	1.50	1.50	1.50	1.00	1.25	294.69
1.50	1.00	1.50	1.50	1.50	1.50	1.50	1.25	295.05
1.50	1.50	0.50	0.50	0.50	0.50	0.50	1.25	493.07
1.50	1.50	0.50	0.50	0.50	0.50	1.00	1.25	488.39
1.50	1.50	0.50	0.50	0.50	0.50	1.50	1.25	489.48
1.50	1.50	0.50	0.50	0.50	1.00	0.50	1.25	392.87
1.50	1.50	0.50	0.50	0.50	1.00	1.00	1.25	385.11
1.50	1.50	0.50	0.50	0.50	1.00	1.50	1.25	385.14
1.50	1.50	0.50	0.50	0.50	1.50	0.50	1.25	392.47
1.50	1.50	0.50	0.50	0.50	1.50	1.00	1.25	383.23
1.50	1.50	0.50	0.50	0.50	1.50	1.50	1.25	382.87
1.50	1.50	0.50	0.50	1.00	0.50	0.50	1.25	471.04
1.50	1.50	0.50	0.50	1.00	0.50	1.00	1.25	469.99

$x_2$	$x_3$	$x_5$	$x_6$	$x_8$	$x_9$	$x_{10}$	$x_{11}^*$	$ev_4$
1.50	1.50	0.50	0.50	1.00	0.50	1.50	1.25	472.33
1.50	1.50	0.50	0.50	1.00	1.00	0.50	1.25	359.24
1.50	1.50	0.50	0.50	1.00	1.00	1.00	1.25	356.49
1.50	1.50	0.50	0.50	1.00	1.00	1.50	1.25	358.35
1.50	1.50	0.50	0.50	1.00	1.50	0.50	1.25	352.42
1.50	1.50	0.50	0.50	1.00	1.50	1.00	1.25	348.31
1.50	1.50	0.50	0.50	1.00	1.50	1.50	1.25	349.61
1.50	1.50	0.50	0.50	1.50	0.50	0.50	1.25	478.14
1.50	1.50	0.50	0.50	1.50	0.50	1.00	1.25	477.83
1.50	1.50	0.50	0.50	1.50	0.50	1.50	1.25	480.45
1.50	1.50	0.50	0.50	1.50	1.00	0.50	1.25	364.12
1.50	1.50	0.50	0.50	1.50	1.00	1.00	1.25	361.67
1.50	1.50	0.50	0.50	1.50	1.00	1.50	1.25	363.70
1.50	1.50	0.50	0.50	1.50	1.50	0.50	1.25	355.64
1.50	1.50	0.50	0.50	1.50	1.50	1.00	1.25	351.86
1.50	1.50	0.50	0.50	1.50	1.50	1.50	1.25	353.40
1.50	1.50	0.50	1.00	0.50	0.50	0.50	1.25	463.85
1.50	1.50	0.50	1.00	0.50	0.50	1.00	1.25	457.08
1.50	1.50	0.50	1.00	0.50	0.50	1.50	1.25	457.55
1.50	1.50	0.50	1.00	0.50	1.00	0.50	1.25	349.90
1.50	1.50	0.50	1.00	0.50	1.00	1.00	1.25	338.87
1.50	1.50	0.50	1.00	0.50	1.00	1.50	1.25	338.04
1.50	1.50	0.50	1.00	0.50	1.50	0.50	1.25	342.75
1.50	1.50	0.50	1.00	0.50	1.50	1.00	1.25	330.22
1.50	1.50	0.50	1.00	0.50	1.50	1.50	1.25	328.88
1.50	1.50	0.50	1.00	1.00	0.50	0.50	1.25	431.59
1.50	1.50	0.50	1.00	1.00	0.50	1.00	1.25	429.45
1.50	1.50	0.50	1.00	1.00	0.50	1.50	1.25	431.52
1.50	1.50	0.50	1.00	1.00	1.00	0.50	1.25	305.19
1.50	1.50	0.50	1.00	1.00	1.00	1.00	1.25	300.53
1.50	1.50	0.50	1.00	1.00	1.00	1.50	1.25	301.92
1.50	1.50	0.50	1.00	1.00	1.50	0.50	1.25	290.26
1.50	1.50	0.50	1.00	1.00	1.50	1.00	1.25	283.80
1.50	1.50	0.50	1.00	1.00	1.50	1.50	1.25	284.49
1.50	1.50	0.50	1.00	1.50	0.50	0.50	1.25	437.09
1.50	1.50	0.50	1.00	1.50	0.50	1.00	1.25	435.80
1.50	1.50	0.50	1.00	1.50	0.50	1.50	1.25	438.20
1.50	1.50	0.50	1.00	1.50	1.00	0.50	1.25	307.13
1.50	1.50	0.50	1.00	1.50	1.00	1.00	1.25	303.22
1.50	1.50	0.50	1.00	1.50	1.00	1.50	1.25	304.94
1.50	1.50	0.50	1.00	1.50	1.50	0.50	1.25	290.11
1.50	1.50	0.50	1.00	1.50	1.50	1.00	1.25	284.64
1.50	1.50	0.50	1.00	1.50	1.50	1.50	1.25	285.74
1.50	1.50	0.50	1.50	0.50	0.50	0.50	1.25	473.53
1.50	1.50	0.50	1.50	0.50	0.50	1.00	1.25	466.42
1.50	1.50	0.50	1.50	0.50	0.50	1.50	1.25	466.82



D.1. PROCURA DO ÓPTIMO PARA AS VÁRIAS COMBINAÇÕES DE F271

$x_2$	$x_3$	$x_5$	$x_6$	$x_8$	$x_9$	$x_{10}$	$x_1^*$	$ev_4$
1.50	1.50	0.50	1.50	0.50	1.00	0.50	1.25	358.59
1.50	1.50	0.50	1.50	0.50	1.00	1.00	1.25	347.02
1.50	1.50	0.50	1.50	0.50	1.00	1.50	1.25	346.08
1.50	1.50	0.50	1.50	0.50	1.50	0.50	1.25	350.39
1.50	1.50	0.50	1.50	0.50	1.50	1.00	1.25	337.26
1.50	1.50	0.50	1.50	0.50	1.50	1.50	1.25	335.77
1.50	1.50	0.50	1.50	1.00	0.50	0.50	1.25	440.22
1.50	1.50	0.50	1.50	1.00	0.50	1.00	1.25	437.74
1.50	1.50	0.50	1.50	1.00	0.50	1.50	1.25	439.74
1.50	1.50	0.50	1.50	1.00	1.00	0.50	1.25	312.73
1.50	1.50	0.50	1.50	1.00	1.00	1.00	1.25	307.55
1.50	1.50	0.50	1.50	1.00	1.00	1.50	1.25	308.82
1.50	1.50	0.50	1.50	1.00	1.50	0.50	1.25	296.50
1.50	1.50	0.50	1.50	1.00	1.50	1.00	1.25	289.38
1.50	1.50	0.50	1.50	1.00	1.50	1.50	1.25	289.91
1.50	1.50	0.50	1.50	1.50	0.50	0.50	1.25	445.56
1.50	1.50	0.50	1.50	1.50	0.50	1.00	1.25	443.88
1.50	1.50	0.50	1.50	1.50	0.50	1.50	1.25	446.20
1.50	1.50	0.50	1.50	1.50	1.00	0.50	1.25	314.34
1.50	1.50	0.50	1.50	1.50	1.00	1.00	1.25	309.88
1.50	1.50	0.50	1.50	1.50	1.00	1.50	1.25	311.48
1.50	1.50	0.50	1.50	1.50	1.50	0.50	1.25	296.00
1.50	1.50	0.50	1.50	1.50	1.50	1.00	1.25	289.85
1.50	1.50	0.50	1.50	1.50	1.50	1.50	1.25	290.79
1.50	1.50	1.00	0.50	0.50	0.50	0.50	1.25	495.32
1.50	1.50	1.00	0.50	0.50	0.50	1.00	1.25	490.83
1.50	1.50	1.00	0.50	0.50	0.50	1.50	1.25	491.97
1.50	1.50	1.00	0.50	0.50	1.00	0.50	1.25	395.01
1.50	1.50	1.00	0.50	0.50	1.00	1.00	1.25	387.51
1.50	1.50	1.00	0.50	0.50	1.00	1.50	1.25	387.61
1.50	1.50	1.00	0.50	0.50	1.50	0.50	1.25	394.48
1.50	1.50	1.00	0.50	0.50	1.50	1.00	1.25	385.61
1.50	1.50	1.00	0.50	0.50	1.50	1.50	1.25	385.33
1.50	1.50	1.00	0.50	1.00	0.50	0.50	1.25	473.16
1.50	1.50	1.00	0.50	1.00	0.50	1.00	1.25	472.39
1.50	1.50	1.00	0.50	1.00	0.50	1.50	1.25	474.80
1.50	1.50	1.00	0.50	1.00	1.00	0.50	1.25	361.17
1.50	1.50	1.00	0.50	1.00	1.00	1.00	1.25	358.84
1.50	1.50	1.00	0.50	1.00	1.00	1.50	1.25	360.80
1.50	1.50	1.00	0.50	1.00	1.50	0.50	1.25	354.20
1.50	1.50	1.00	0.50	1.00	1.50	1.00	1.25	350.62
1.50	1.50	1.00	0.50	1.00	1.50	1.50	1.25	352.06
1.50	1.50	1.00	0.50	1.50	0.50	0.50	1.25	480.25
1.50	1.50	1.00	0.50	1.50	0.50	1.00	1.25	480.23
1.50	1.50	1.00	0.50	1.50	0.50	1.50	1.25	482.92
1.50	1.50	1.00	0.50	1.50	1.00	0.50	1.25	366.05

$x_2$	$x_3$	$x_5$	$x_6$	$x_8$	$x_9$	$x_{10}$	$x_{11}^*$	$ev_4$
1.50	1.50	1.00	0.50	1.50	1.00	1.00	1.25	364.02
1.50	1.50	1.00	0.50	1.50	1.00	1.50	1.25	366.16
1.50	1.50	1.00	0.50	1.50	1.50	0.50	1.25	357.41
1.50	1.50	1.00	0.50	1.50	1.50	1.00	1.25	354.18
1.50	1.50	1.00	0.50	1.50	1.50	0.50	1.25	355.85
1.50	1.50	1.00	1.00	0.50	0.50	0.50	1.25	466.11
1.50	1.50	1.00	1.00	0.50	0.50	1.00	1.25	459.51
1.50	1.50	1.00	1.00	0.50	0.50	1.50	1.25	460.03
1.50	1.50	1.00	1.00	0.50	1.00	0.50	1.25	352.04
1.50	1.50	1.00	1.00	0.50	1.00	1.00	1.25	341.27
1.50	1.50	1.00	1.00	0.50	1.00	1.50	1.25	340.51
1.50	1.50	1.00	1.00	0.50	1.50	0.50	1.25	344.77
1.50	1.50	1.00	1.00	0.50	1.50	1.00	1.25	332.60
1.50	1.50	1.00	1.00	0.50	1.50	1.50	1.25	331.35
1.50	1.50	1.00	1.00	1.00	0.50	0.50	1.25	433.70
1.50	1.50	1.00	1.00	1.00	0.50	1.00	1.25	431.84
1.50	1.50	1.00	1.00	1.00	0.50	1.50	1.25	433.99
1.50	1.50	1.00	1.00	1.00	1.00	0.50	1.25	307.11
1.50	1.50	1.00	1.00	1.00	1.00	1.00	1.25	302.88
1.50	1.50	1.00	1.00	1.00	1.00	1.50	1.25	304.37
1.50	1.50	1.00	1.00	1.00	1.50	0.50	1.25	292.04
1.50	1.50	1.00	1.00	1.00	1.50	1.00	1.25	286.11
1.50	1.50	1.00	1.00	1.00	1.50	1.50	1.25	286.94
1.50	1.50	1.00	1.00	1.50	0.50	0.50	1.25	439.21
1.50	1.50	1.00	1.00	1.50	0.50	1.00	1.25	438.20
1.50	1.50	1.00	1.00	1.50	0.50	1.50	1.25	440.67
1.50	1.50	1.00	1.00	1.50	1.00	0.50	1.25	309.05
1.50	1.50	1.00	1.00	1.50	1.00	1.00	1.25	305.56
1.50	1.50	1.00	1.00	1.50	1.00	1.50	1.25	307.39
1.50	1.50	1.00	1.00	1.50	1.50	0.50	1.25	291.89
1.50	1.50	1.00	1.00	1.50	1.50	1.00	1.25	286.95
1.50	1.50	1.00	1.00	1.50	1.50	1.50	1.25	288.19
1.50	1.50	1.00	1.50	0.50	0.50	0.50	1.25	475.69
1.50	1.50	1.00	1.50	0.50	0.50	1.00	1.25	468.83
1.50	1.50	1.00	1.50	0.50	0.50	1.50	1.25	469.30
1.50	1.50	1.00	1.50	0.50	1.00	0.50	1.25	360.58
1.50	1.50	1.00	1.50	0.50	1.00	1.00	1.25	349.38
1.50	1.50	1.00	1.50	0.50	1.00	1.50	1.25	348.54
1.50	1.50	1.00	1.50	0.50	1.50	0.50	1.25	352.24
1.50	1.50	1.00	1.50	0.50	1.50	1.00	1.25	339.60
1.50	1.50	1.00	1.50	0.50	1.50	1.50	1.25	338.23
1.50	1.50	1.00	1.50	1.00	0.50	0.50	1.25	442.19
1.50	1.50	1.00	1.50	1.00	0.50	1.00	1.25	440.10
1.50	1.50	1.00	1.50	1.00	0.50	1.50	1.25	442.20
1.50	1.50	1.00	1.50	1.00	1.00	0.50	1.25	314.46
1.50	1.50	1.00	1.50	1.00	1.00	1.00	1.25	309.85

D.1. PROCURA DO ÓPTIMO PARA AS VÁRIAS COMBINAÇÕES DE F273

$x_2$	$x_3$	$x_5$	$x_6$	$x_8$	$x_9$	$x_{10}$	$x_1^*$	$ev_4$
1.50	1.50	1.00	1.50	1.00	1.00	1.50	1.25	311.25
1.50	1.50	1.00	1.50	1.00	1.50	0.50	1.25	298.03
1.50	1.50	1.00	1.50	1.00	1.50	1.00	1.25	291.64
1.50	1.50	1.00	1.50	1.00	1.50	1.50	1.25	292.34
1.50	1.50	1.00	1.50	1.50	0.50	0.50	1.25	447.53
1.50	1.50	1.00	1.50	1.50	0.50	1.00	1.25	446.25
1.50	1.50	1.00	1.50	1.50	0.50	1.50	1.25	448.66
1.50	1.50	1.00	1.50	1.50	1.00	0.50	1.25	316.07
1.50	1.50	1.00	1.50	1.50	1.00	1.00	1.25	312.18
1.50	1.50	1.00	1.50	1.50	1.00	1.50	1.25	313.91
1.50	1.50	1.00	1.50	1.50	1.50	0.50	1.25	297.53
1.50	1.50	1.00	1.50	1.50	1.50	1.00	1.25	292.11
1.50	1.50	1.00	1.50	1.50	1.50	1.50	1.25	293.23
1.50	1.50	1.50	0.50	0.50	0.50	0.50	1.25	497.82
1.50	1.50	1.50	0.50	0.50	0.50	1.00	1.25	493.33
1.50	1.50	1.50	0.50	0.50	0.50	1.50	1.25	494.47
1.50	1.50	1.50	0.50	0.50	1.00	0.50	1.25	397.51
1.50	1.50	1.50	0.50	0.50	1.00	1.00	1.25	390.01
1.50	1.50	1.50	0.50	0.50	1.00	1.50	1.25	390.11
1.50	1.50	1.50	0.50	0.50	1.50	0.50	1.25	396.98
1.50	1.50	1.50	0.50	0.50	1.50	1.00	1.25	388.11
1.50	1.50	1.50	0.50	0.50	1.50	1.50	1.25	387.83
1.50	1.50	1.50	0.50	1.00	0.50	0.50	1.25	475.66
1.50	1.50	1.50	0.50	1.00	0.50	1.00	1.25	474.89
1.50	1.50	1.50	0.50	1.00	0.50	1.50	1.25	477.30
1.50	1.50	1.50	0.50	1.00	1.00	0.50	1.25	363.67
1.50	1.50	1.50	0.50	1.00	1.00	1.00	1.25	361.34
1.50	1.50	1.50	0.50	1.00	1.00	1.50	1.25	363.30
1.50	1.50	1.50	0.50	1.00	1.50	0.50	1.25	356.70
1.50	1.50	1.50	0.50	1.00	1.50	1.00	1.25	353.12
1.50	1.50	1.50	0.50	1.00	1.50	1.50	1.25	354.56
1.50	1.50	1.50	0.50	1.50	0.50	0.50	1.25	482.75
1.50	1.50	1.50	0.50	1.50	0.50	1.00	1.25	482.73
1.50	1.50	1.50	0.50	1.50	0.50	1.50	1.25	485.42
1.50	1.50	1.50	0.50	1.50	1.00	0.50	1.25	368.55
1.50	1.50	1.50	0.50	1.50	1.00	1.00	1.25	366.52
1.50	1.50	1.50	0.50	1.50	1.00	1.50	1.25	368.66
1.50	1.50	1.50	0.50	1.50	1.50	0.50	1.25	359.91
1.50	1.50	1.50	0.50	1.50	1.50	1.00	1.25	356.68
1.50	1.50	1.50	0.50	1.50	1.50	1.50	1.25	358.35
1.50	1.50	1.50	1.00	0.50	0.50	0.50	1.25	468.61
1.50	1.50	1.50	1.00	0.50	0.50	1.00	1.25	462.01
1.50	1.50	1.50	1.00	0.50	0.50	1.50	1.25	462.53
1.50	1.50	1.50	1.00	0.50	1.00	0.50	1.25	354.54
1.50	1.50	1.50	1.00	0.50	1.00	1.00	1.25	343.77
1.50	1.50	1.50	1.00	0.50	1.00	1.50	1.25	343.01

$x_2$	$x_3$	$x_5$	$x_6$	$x_8$	$x_9$	$x_{10}$	$x_{11}^*$	$ev_4$
1.50	1.50	1.50	1.00	0.50	1.50	0.50	1.25	347.27
1.50	1.50	1.50	1.00	0.50	1.50	1.00	1.25	335.10
1.50	1.50	1.50	1.00	0.50	1.50	1.50	1.25	333.85
1.50	1.50	1.50	1.00	1.00	0.50	0.50	1.25	436.20
1.50	1.50	1.50	1.00	1.00	0.50	1.00	1.25	434.34
1.50	1.50	1.50	1.00	1.00	0.50	1.50	1.25	436.49
1.50	1.50	1.50	1.00	1.00	1.00	0.50	1.25	309.61
1.50	1.50	1.50	1.00	1.00	1.00	1.00	1.25	305.38
1.50	1.50	1.50	1.00	1.00	1.00	1.50	1.25	306.87
1.50	1.50	1.50	1.00	1.00	1.50	0.50	1.25	294.54
1.50	1.50	1.50	1.00	1.00	1.50	1.00	1.25	288.61
1.50	1.50	1.50	1.00	1.00	1.50	1.50	1.25	289.44
1.50	1.50	1.50	1.00	1.50	0.50	0.50	1.25	441.71
1.50	1.50	1.50	1.00	1.50	0.50	1.00	1.25	440.70
1.50	1.50	1.50	1.00	1.50	0.50	1.50	1.25	443.17
1.50	1.50	1.50	1.00	1.50	1.00	0.50	1.25	311.55
1.50	1.50	1.50	1.00	1.50	1.00	1.00	1.25	308.06
1.50	1.50	1.50	1.00	1.50	1.00	1.50	1.25	309.89
1.50	1.50	1.50	1.00	1.50	1.50	0.50	1.25	294.39
1.50	1.50	1.50	1.00	1.50	1.50	1.00	1.25	289.45
1.50	1.50	1.50	1.00	1.50	1.50	1.50	1.25	290.69
1.50	1.50	1.50	1.50	0.50	0.50	0.50	1.25	478.19
1.50	1.50	1.50	1.50	0.50	0.50	1.00	1.25	471.33
1.50	1.50	1.50	1.50	0.50	0.50	1.50	1.25	471.80
1.50	1.50	1.50	1.50	0.50	1.00	0.50	1.25	363.08
1.50	1.50	1.50	1.50	0.50	1.00	1.00	1.25	351.88
1.50	1.50	1.50	1.50	0.50	1.00	1.50	1.25	351.04
1.50	1.50	1.50	1.50	0.50	1.50	0.50	1.25	354.74
1.50	1.50	1.50	1.50	0.50	1.50	1.00	1.25	342.10
1.50	1.50	1.50	1.50	0.50	1.50	1.50	1.25	340.73
1.50	1.50	1.50	1.50	1.00	0.50	0.50	1.25	444.69
1.50	1.50	1.50	1.50	1.00	0.50	1.00	1.25	442.60
1.50	1.50	1.50	1.50	1.00	0.50	1.50	1.25	444.70
1.50	1.50	1.50	1.50	1.00	1.00	0.50	1.25	316.96
1.50	1.50	1.50	1.50	1.00	1.00	1.00	1.25	312.35
1.50	1.50	1.50	1.50	1.00	1.00	1.50	1.25	313.75
1.50	1.50	1.50	1.50	1.00	1.50	0.50	1.25	300.53
1.50	1.50	1.50	1.50	1.00	1.50	1.00	1.25	294.14
1.50	1.50	1.50	1.50	1.00	1.50	1.50	1.25	294.84
1.50	1.50	1.50	1.50	1.50	0.50	0.50	1.25	450.03
1.50	1.50	1.50	1.50	1.50	0.50	1.00	1.25	448.75
1.50	1.50	1.50	1.50	1.50	0.50	1.50	1.25	451.16

*D.1. PROCURA DO ÓPTIMO PARA AS VÁRIAS COMBINAÇÕES DE  $\mathcal{F}275$*

$\mathbf{x}_2$	$\mathbf{x}_3$	$\mathbf{x}_5$	$\mathbf{x}_6$	$\mathbf{x}_8$	$\mathbf{x}_9$	$\mathbf{x}_{10}$	$\mathbf{x}_1^*$	$\mathbf{ev}_4$
1.50	1.50	1.50	1.50	1.50	1.00	0.50	1.25	318.57
1.50	1.50	1.50	1.50	1.50	1.00	1.00	1.25	314.68
1.50	1.50	1.50	1.50	1.50	1.00	1.50	1.25	316.41
1.50	1.50	1.50	1.50	1.50	1.50	0.50	1.25	300.03
1.50	1.50	1.50	1.50	1.50	1.50	1.00	1.25	294.61
1.50	1.50	1.50	1.50	1.50	1.50	1.50	1.25	295.73

**O melhor valor foi:**

$\mathbf{x}_2$	$\mathbf{x}_3$	$\mathbf{x}_5$	$\mathbf{x}_6$	$\mathbf{x}_8$	$\mathbf{x}_9$	$\mathbf{x}_{10}$	$\mathbf{x}_1^*$	$\mathbf{ev}_4$
0.50	1.50	0.50	1.00	1.00	1.50	1.00	1.25	280.85

## D.2 Política óptima para o estágio 1

$t_4$	$t_5$	$t_6$	$x_{11}^*$	$ev_1$
1.8261	4.1086	5.0977	0.5000	159.0983
1.8261	4.1086	48.6597	1.0000	162.2233
1.8261	4.1086	92.2217	1.5000	318.4000
1.8261	4.1086	135.7836	1.5000	536.2098
1.8261	4.1086	179.3456	1.5000	754.0197
1.8261	4.1086	222.9076	1.5000	971.8295
1.8261	4.1086	266.4695	1.5000	1189.6394
1.8261	56.7731	5.0977	0.5000	167.7280
1.8261	56.7731	48.6597	1.0000	170.8530
1.8261	56.7731	92.2217	1.5000	318.4000
1.8261	56.7731	135.7836	1.5000	536.2098
1.8261	56.7731	179.3456	1.5000	754.0197
1.8261	56.7731	222.9076	1.5000	971.8295
1.8261	56.7731	266.4695	1.5000	1189.6394
1.8261	109.4376	5.0977	0.5000	410.7294
1.8261	109.4376	48.6597	0.5000	410.7294
1.8261	109.4376	92.2217	0.7500	412.8410
1.8261	109.4376	135.7836	1.5000	536.2098
1.8261	109.4376	179.3456	1.5000	754.0197
1.8261	109.4376	222.9076	1.5000	971.8295
1.8261	109.4376	266.4695	1.5000	1189.6394
1.8261	162.1020	5.0977	0.5000	674.0518
1.8261	162.1020	48.6597	0.5000	674.0518
1.8261	162.1020	92.2217	0.5000	674.0518
1.8261	162.1020	135.7836	0.5000	674.9672
1.8261	162.1020	179.3456	1.5000	754.0197
1.8261	162.1020	222.9076	1.5000	971.8295
1.8261	162.1020	266.4695	1.5000	1189.6394
1.8261	214.7665	5.0977	0.5000	1189.6394
1.8261	214.7665	48.6597	0.5000	937.3741
1.8261	214.7665	92.2217	0.5000	937.3741
1.8261	214.7665	135.7836	0.5000	937.3741
1.8261	214.7665	179.3456	0.5000	937.3741
1.8261	214.7665	222.9076	1.5000	977.0104
1.8261	214.7665	266.4695	1.5000	1189.6394
33.0346	4.1086	5.0977	0.5000	198.1091
33.0346	4.1086	48.6597	1.0000	201.2341
33.0346	4.1086	92.2217	1.5000	322.9540
33.0346	4.1086	135.7836	1.5000	536.2098
33.0346	4.1086	179.3456	1.5000	754.0197
33.0346	4.1086	222.9076	1.5000	971.8295
33.0346	4.1086	266.4695	1.5000	1189.6394
33.0346	56.7731	5.0977	0.5000	205.3107
33.0346	56.7731	48.6597	1.0000	208.4357

$t_4$	$t_5$	$t_6$	$x_{11}^*$	$ev_1$
33.0346	56.7731	92.2217	1.5000	322.9540
33.0346	56.7731	135.7836	1.5000	536.2098
33.0346	56.7731	179.3456	1.5000	754.0197
33.0346	56.7731	222.9076	1.5000	971.8295
33.0346	56.7731	266.4695	1.5000	1189.6394
33.0346	109.4376	5.0977	0.5000	410.7294
33.0346	109.4376	48.6597	0.5000	410.7294
33.0346	109.4376	92.2217	0.7500	412.8410
33.0346	109.4376	135.7836	1.5000	536.2098
33.0346	109.4376	179.3456	1.5000	754.0197
33.0346	109.4376	222.9076	1.5000	971.8295
33.0346	109.4376	266.4695	1.5000	1189.6394
33.0346	162.1020	5.0977	0.5000	674.0518
33.0346	162.1020	48.6597	0.5000	674.0518
33.0346	162.1020	92.2217	0.5000	674.0518
33.0346	162.1020	135.7836	0.5000	674.9672
33.0346	162.1020	179.3456	1.5000	754.0197
33.0346	162.1020	222.9076	1.5000	971.8295
33.0346	162.1020	266.4695	1.5000	1189.6394
33.0346	214.7665	5.0977	0.5000	937.3741
33.0346	214.7665	48.6597	0.5000	937.3741
33.0346	214.7665	92.2217	0.5000	937.3741
33.0346	214.7665	135.7836	0.5000	937.3741
33.0346	214.7665	179.3456	0.5000	937.3741
33.0346	214.7665	222.9076	1.5000	977.0104
33.0346	214.7665	266.4695	1.5000	1189.6394
64.2432	4.1086	5.0977	0.5000	290.3132
64.2432	4.1086	48.6597	0.7500	292.0313
64.2432	4.1086	92.2217	1.5000	361.0760
64.2432	4.1086	135.7836	1.5000	536.2098
64.2432	4.1086	179.3456	1.5000	754.0197
64.2432	4.1086	222.9076	1.5000	971.8295
64.2432	4.1086	266.4695	1.5000	1189.6394
64.2432	56.7731	5.0977	0.5000	291.7614
64.2432	56.7731	48.6597	0.7500	293.4406
64.2432	56.7731	92.2217	1.5000	361.0760
64.2432	56.7731	135.7836	1.5000	536.2098
64.2432	56.7731	179.3456	1.5000	754.0197
64.2432	56.7731	222.9076	1.5000	971.8295
64.2432	56.7731	266.4695	1.5000	1189.6394
64.2432	109.4376	5.0977	0.5000	428.7606
64.2432	109.4376	48.6597	0.5000	428.7606
64.2432	109.4376	92.2217	0.7500	430.7349
64.2432	109.4376	135.7836	1.5000	536.2098
64.2432	109.4376	179.3456	1.5000	754.0197

$t_4$	$t_5$	$t_6$	$x_{11}^*$	$ev_1$
64.2432	109.4376	222.9076	1.5000	971.8295
64.2432	109.4376	266.4695	1.5000	1189.6394
64.2432	162.1020	5.0977	0.5000	674.0518
64.2432	162.1020	48.6597	0.5000	674.0518
64.2432	162.1020	92.2217	0.5000	674.0518
64.2432	162.1020	135.7836	0.5000	674.9672
64.2432	162.1020	179.3456	1.5000	754.0197
64.2432	162.1020	222.9076	1.5000	971.8295
64.2432	162.1020	266.4695	1.5000	1189.6394
64.2432	214.7665	5.0977	0.5000	937.3741
64.2432	214.7665	48.6597	0.5000	937.3741
64.2432	214.7665	92.2217	0.5000	937.3741
64.2432	214.7665	135.7836	0.5000	937.3741
64.2432	214.7665	179.3456	0.5000	937.3741
64.2432	214.7665	222.9076	1.5000	977.0104
64.2432	214.7665	266.4695	1.5000	1189.6394
95.4518	4.1086	5.0977	0.5000	446.3561
95.4518	4.1086	48.6597	0.5000	446.3561
95.4518	4.1086	92.2217	1.0000	451.9436
95.4518	4.1086	135.7836	1.5000	563.4441
95.4518	4.1086	179.3456	1.5000	754.0197
95.4518	4.1086	222.9076	1.5000	971.8295
95.4518	4.1086	266.4695	1.5000	1189.6394
95.4518	56.7731	5.0977	0.5000	446.3561
95.4518	56.7731	48.6597	0.5000	446.3561
95.4518	56.7731	92.2217	1.0000	451.9436
95.4518	56.7731	135.7836	1.5000	563.4441
95.4518	56.7731	179.3456	1.5000	754.0197
95.4518	56.7731	222.9076	1.5000	971.8295
95.4518	56.7731	266.4695	1.5000	1189.6394
95.4518	109.4376	5.0977	0.5000	477.3394
95.4518	109.4376	48.6597	0.5000	477.3394
95.4518	109.4376	92.2217	0.7500	479.1764
95.4518	109.4376	135.7836	1.5000	563.4441
95.4518	109.4376	179.3456	1.5000	754.0197
95.4518	109.4376	222.9076	1.5000	971.8295
95.4518	109.4376	266.4695	1.5000	1189.6394
95.4518	162.1020	5.0977	0.5000	674.0518
95.4518	162.1020	48.6597	0.5000	674.0518
95.4518	162.1020	92.2217	0.5000	674.0518
95.4518	162.1020	135.7836	0.5000	674.9672
95.4518	162.1020	179.3456	1.5000	754.0197
95.4518	162.1020	222.9076	1.5000	971.8295
95.4518	162.1020	266.4695	1.5000	1189.6394
95.4518	214.7665	5.0977	0.5000	937.3741



$t_4$	$t_5$	$t_6$	$x_{11}^*$	$ev_1$
95.4518	214.7665	48.6597	0.5000	937.3741
95.4518	214.7665	92.2217	0.5000	937.3741
95.4518	214.7665	135.7836	0.5000	937.3741
95.4518	214.7665	179.3456	0.5000	937.3741
95.4518	214.7665	222.9076	1.5000	977.0104
95.4518	214.7665	266.4695	1.5000	1189.6394

### D.3 Política óptima para o estágio 2

$t_2$	$t_3$	$t_4$	$x_7^*$	$ev_2$
0.9130	2.0543	1.8261	1.2500	204.8755
0.9130	2.0543	33.0346	1.2500	248.9305
0.9130	2.0543	64.2432	0.7500	394.4870
0.9130	2.0543	95.4518	0.7500	551.8620
0.9130	37.1640	1.8261	1.2500	244.5522
0.9130	37.1640	33.0346	1.2500	278.7319
0.9130	37.1640	64.2432	1.2500	411.4685
0.9130	37.1640	95.4518	0.7500	568.0638
0.9130	72.2736	1.8261	1.0000	406.1858
0.9130	72.2736	33.0346	1.0000	409.6013
0.9130	72.2736	64.2432	1.0000	464.3149
0.9130	72.2736	95.4518	1.0000	577.0860
0.9130	107.3832	1.8261	1.0000	569.5432
0.9130	107.3832	33.0346	1.0000	570.6817
0.9130	107.3832	64.2432	1.0000	588.9195
0.9130	107.3832	95.4518	1.0000	649.4745
16.5173	2.0543	1.8261	1.2500	245.3751
16.5173	2.0543	33.0346	1.2500	278.3413
16.5173	2.0543	64.2432	0.7500	401.2228
16.5173	2.0543	95.4518	0.7500	552.1203
16.5173	37.1640	1.8261	1.2500	274.3410
16.5173	37.1640	33.0346	1.2500	300.2041
16.5173	37.1640	64.2432	1.2500	417.9944
16.5173	37.1640	95.4518	0.7500	568.2360
16.5173	72.2736	1.8261	1.0000	406.1858
16.5173	72.2736	33.0346	1.0000	409.6013
16.5173	72.2736	64.2432	1.0000	464.3149
16.5173	72.2736	95.4518	1.0000	577.0860
16.5173	107.3832	1.8261	1.0000	569.5432
16.5173	107.3832	33.0346	1.0000	570.6817
16.5173	107.3832	64.2432	1.0000	588.9195
16.5173	107.3832	95.4518	1.0000	649.4745
32.1216	2.0543	1.8261	1.2500	249.4064
32.1216	2.0543	33.0346	1.2500	278.7319
32.1216	2.0543	64.2432	0.7500	401.3277
32.1216	2.0543	95.4518	0.7500	552.1203
32.1216	37.1640	1.8261	1.2500	277.0715
32.1216	37.1640	33.0346	1.2500	300.2041
32.1216	37.1640	64.2432	1.2500	417.9944
32.1216	37.1640	95.4518	0.7500	568.2360
32.1216	72.2736	1.8261	1.0000	406.1858
32.1216	72.2736	33.0346	1.0000	409.6013
32.1216	72.2736	64.2432	1.0000	464.3149

$t_2$	$t_3$	$t_4$	$x_7^*$	$ev_2$
32.1216	72.2736	95.4518	1.0000	577.0860
32.1216	107.3832	1.8261	1.0000	569.5432
32.1216	107.3832	33.0346	1.0000	570.6817
32.1216	107.3832	64.2432	1.0000	588.9195
32.1216	107.3832	95.4518	1.0000	649.4745
47.7259	2.0543	1.8261	1.2500	286.2931
47.7259	2.0543	33.0346	0.7500	305.0096
47.7259	2.0543	64.2432	0.7500	407.8536
47.7259	2.0543	95.4518	0.7500	552.3786
47.7259	37.1640	1.8261	1.2500	304.7365
47.7259	37.1640	33.0346	1.2500	321.6763
47.7259	37.1640	64.2432	1.2500	424.5203
47.7259	37.1640	95.4518	0.7500	568.4082
47.7259	72.2736	1.8261	1.0000	406.1858
47.7259	72.2736	33.0346	1.0000	409.6013
47.7259	72.2736	64.2432	1.0000	464.3149
47.7259	72.2736	95.4518	1.0000	577.0860
47.7259	107.3832	1.8261	1.0000	569.5432
47.7259	107.3832	33.0346	1.0000	570.6817
47.7259	107.3832	64.2432	1.0000	588.9195
47.7259	107.3832	95.4518	1.0000	649.4745

#### D.4 Política óptima para o estágio 3

$t_2$	$x_4^*$	$ev_3$
0.9130	0.7500	251.4763
16.5173	1.0000	295.0074
32.1216	1.5000	307.3878
47.7259	1.0000	373.0500

#### D.5 Política óptima para o estágio 4

$x_1^*$	$ev_4$
1.2500	274.8591



# Apêndice E

## Aproximação 1 (Versão 1)

Neste apêndice vão ser apresentados os procedimentos que sofreram alterações, relativamente à versão 1 sem aproximações (apêndice A).

### E.1 mp.m

```
% mp.m (performs a 3 factorial experiment
% for every fixed variable xi (0.5(Lo),1.0(Medium),
% 1.5(Hi))calculating x1star and the expected value of
% cost for the project with 4 states (ev4))
btime = clock;    % to measure time
% Global variables: lambda, expW, p, P, cl, T, Wi
% Lambda is the parameter of W
lambda=[0.1,0.12,0.05,0.08,0.2,0.04,0.03,0.04,0.024,0.15,0.16];
expW=1./lambda;
p=[0.25 0.25 0.25 0.25];          % probability of Wi
P=[0.25 0.5 0.75 1];            % cumulative probability of Wi
cl=5;                            % delay cost per unit time
T=65;                            % due date
% Fixed W's at their means
```

```
W1=generateW(lambda(1));
W2=expW(2);
W3=expW(3);
W4=generateW(lambda(4));
W5=expW(5);
W6=expW(6);
W7=generateW(lambda(7));
W8=expW(8);
W9=expW(9);
W10=expW(10);
W11=generateW(lambda(11));
It2min=W1(1)/1.5;
It2max=W1(4)/0.5;
It2=[It2min,It2max];
[stT2,T2]=calculateTvalues(It2);
It3min=It2min+W4(1)/1.5;
It3max=It2max+W4(4)/0.5;
It3=[It3min,It3max];
[stT3,T3]=calculateTvalues(It3);
It4min=W3/1.5;
It4max=W3/0.5;
It4=[It4min,It4max];
[stT4,T4]=calculateTvalues(It4);
It5min=It4min+W8/1.5;
It5max=It4max+W8/0.5;
It5=[It5min,It5max];
[stT5,T5]=calculateTvalues(It5);
It6min=It3min+W7(1)/1.5;
It6max=It3max+W7(4)/0.5;
```



```

disp(' x2 x3 x5 x6 x8 x9 x10 x1star ev4');
for i=1:size(Res,1)
    fprintf('%6.2f %6.2f %6.2f %6.2f %6.2f %6.2f %6.2f %6.2f %8.2f\n',Res(i,:));
end
disp('Best value was:');
disp(' x2 x3 x5 x6 x8 x9 x10 x1star ev4');
fprintf('%6.2f %6.2f %6.2f %6.2f %6.2f %6.2f %6.2f %6.2f %8.2f\n',Res(k,:));
% -----
% Write result to a file (to export to excel)
fid = fopen('resBestAll.txt','w');
fprintf(fid,'\nCurrent parameters');
fprintf(fid,'\nDue date (T) = %3d',T);
fprintf(fid,'\nCost of lateness (cl) = %2d\n\n',cl);
fprintf(fid,'x2,x3,x5,x6,x8,x9,x10,x1star,ev4\n');
for i=1:size(Res,1)
    fprintf(fid,'%6.2f,%6.2f,%6.2f,%6.2f,%6.2f,%6.2f,%6.2f,%6.2f,%8.2f\n',Res(i,:));
end
fprintf(fid,'\nBest value was:\n');
fprintf(fid,'x2,x3,x5,x6,x8,x9,x10,x1star,ev4\n');
fprintf(fid,'%6.2f,%6.2f,%6.2f,%6.2f,%6.2f,%6.2f,%6.2f,%6.2f,%8.2f',Res(k,:));
fclose(fid);
% -----
% Write Best1 to a file (to export to excel)
fid = fopen('resBest1.txt','w');
fprintf(fid,'t2,t4,t6,x11star,ev1\n');
for i=1:size(Best1,1)
    fprintf(fid,'%6.4f,%6.4f,%6.4f,%6.4f,%8.4f\n',Best1(i,:));
end
fclose(fid);

```



```

% -----
% Write Best2 to a file (to export to excel)
fid = fopen('resBest2.txt','w');
fprintf(fid,'t2,t3,t4,x7star,ev2\n');
for i=1:size(Best2,1)
    fprintf(fid,'%6.4f,%6.4f,%6.4f,%6.4f,%8.4f\n',Best2(i,:));
end
fclose(fid);
% -----
% Write Best3 to a file (to export to excel)
fid = fopen('resBest3.txt','w');
fprintf(fid,'t2,t4,x4star,ev3\n');
for i=1:size(Best3,1)
    fprintf(fid,'%6.4f,%6.4f,%6.4f,%8.4f\n',Best3(i,:));
end
fclose(fid);
% -----
% Write Best4 to a file (to export to excel)
fid = fopen('resBest4.txt','w');
fprintf(fid,'x1star,ev4\n');
for i=1:size(Best4,1)
    fprintf(fid,'%6.4f,%8.4f\n',Best4(i,:));
end
fclose(fid);
% -----
% Write result to a file (to read later into Matlab Workspace)
save resultBestAll T cl Best1 Best2 Best3 Best4 Res
fprintf('\n\nElapsed time was %2.2f minutes\n',etime(clock,btime)/60);
clear W* I* T* P x* st* p i cl m k btime lambda

```

```
beep
```

## E.2 mediumprojects1.m

```
% State is s1:(t4,t5,t6)
% Global variables: Best1
% Decision variable is x11
Best1=[ ];
% resource cost for activities whose resource level is fixed
rcf=x2*W2+x3*W3+x5*W5+x6*W6+x8*W8+x9*W9+x10*W10;
for t4=T4
for t5=T5
    % Calculation of Y7: %Y7=max(t4+W9/x9,t5+W10/x10)
    a1=t4+W9/x9;    % This is a number now, not a random variable
    a2=t5+W10/x10;
    y7=max(a1,a2);
    % Calculation of T7, U , ...
    for t6=T6
        EV=[ ];
        for x11=0.5:0.25:1.5
            ev=0;
            for i11=1:4
                w11=W11(i11);
                t7=max(y7,t6+w11/x11);
                u=max(0,t7-T);
                TC(i11)=cl*u;    % Tardiness cost
            end % 4 values - Probability=P
            rc11=x11*expW(11);
            F1=TC+rcf+rc11; % total cost
```

```

        ev=expectedValue(F1,P);
        EV=[EV;x11,ev];
    end
    m=min(EV(:,2));
    k=max(find(EV(:,2)<=m));
    x11star=EV(k,1);
    v1star=EV(k,2);
    Best1=[Best1;t4,t5,t6,x11star,v1star];
end
end
end
clear TC EV F1 a* t* ev* u y7 x11star rcf rc11 i11 w11 m k v1star
%['t4, t5, t6, x11star, ev1']
%Best1
% f1(t4,t5,t6) = min eV(x11*W11+cl*eV(U))
% x11
%
% U=max(0,T7-T)
% T7=max(Y7,t6+w11/x11)
% Y7=max(t4+W9/x9,t5+W10/x10)

```

### E.3 mediumprojects2.m

```

% State is s2:(t2,t3,t4)
% decision variable is x7
% Global variables: Best2
Best2=[ ];
for t2=T2
    a1=t2+W6/x6; % this is now a value, not a random variable

```

```

a3=t2+W5/x5;
for t3=T3
  for t4=T4
    a4=t4+W8/x8;
    t5n=max(a3,a4);
    EV=[ ];
    for x7=0.5:0.25:1.5
      A2=t3+W7/x7;
      T6n=max(a1,A2);
      F1=[ ];
      m=1;
      for i=1:size(T6n,2)
        k=min( find( Best1(:,1)==t4 & abs(Best1(:,2)-t5n)<=stT5/2 &
          abs(Best1(:,3)-T6n(i))<=stT6/2 ) );
        F1(m)=Best1(k,5);
        m=m+1;
      end
      evF1=expectedValue(F1,P);
      ev=x7*expW(7)+evF1;
      EV=[EV;x7,ev];
    end
    m=min(EV(:,2));
    k=max(find(EV(:,2)<=m));
    x7star=EV(k,1);
    v2star=EV(k,2);
    Best2=[Best2;t2,t3,t4,x7star,v2star];
  end
end
end
end

```

```

clear A2 EV T6n F1 a* ev* x7star t* k i m v2star
% ['t2, t3, t4, x7star, ev2']
% Best2
% f2(t2,t3,t4) = min eV(x7*W7+eV(f1(t4,T5,T6))
% x7
%
% T5 = max(t2+W5/x5,t4+W8/x8)
% T6 = max(t2+W6/x6,t3+W7/x7)

```

## E.4 mediumprojects3.m

```

% State is s3:(t2)
% Decision variable is x4
% Global variables: Best3
Best3=[ ];
a2=W2/x2;          % this is now a value, not a random variable
t4n=W3/x3;
for t2=T2
    EV=[ ];
    for x4=0.5:0.25:1.5
        A1=t2+W4/x4;
        T3n=max(A1,a2);
        F2=[ ];
        m=1;
        for i=1:size(T3n,2)
            k=min( find( Best2(:,1)==t2 & abs(Best2(:,2)-T3n(i))<=stT3/2 &
                abs(Best2(:,3)-t4n)<=stT4/2 ) );
            F2(m)=Best2(k,5);
            m=m+1;
        end
    end
end

```

```

end
evF2=expectedValue(F2,P);
ev=x4*expW(4)+evF2;
EV=[EV;x4,ev];
end
m=min(EV(:,2));
k=max(find(EV(:,2)<=m));
x4star=EV(k,1);
v3star=EV(k,2);
Best3=[Best3;t2,x4star,v3star];
end
clear A1 EV T3n F2 a2 ev* x4star t* i k m v3star
% ['t2, t4, x4star, ev3']
% Best3
% f3(t2) = min eV(x4*W4+ev(f2(t2,T3,T4))
% x4
% T4 = W3/x3 (from mp.m)
% T3 = max(t2+W4/x4,W2/x2)

```

## E.5 mediumprojects4.m

```

% State is s4:t1=0
% Decision variable is x1
% Global variables: Best4
Best4=[ ];
EV=[ ];
for x1=0.5:0.25:1.5
T2n=W1/x1;
for i=1:4

```

```

    k=min(find(abs(Best3(:,1)-T2n(i))<=stT2/2));
    F3(i)=Best3(k,3);
end
evF3=expectedValue(F3,P);
ev=x1*expW(1)+evF3;
EV=[EV;x1,ev];
end
m=min(EV(:,2));
k=max(find(EV(:,2)<=m));
x1star=EV(k,1);
v4star=EV(k,2);
Best4=[x1star,v4star];
clear F* EV T2n x1star ev* i j k m v4star sT4
% f4(0) = min eV(x1*W1+eV(f3(T2))
% x1
%
% T2 = W1/X1

```





# Apêndice F

## Aproximação 1 (Versão 2)

Neste apêndice vão ser apresentados os procedimentos que sofreram alterações, relativamente à versão 2 sem aproximações (apêndice B).

### F.1 dp.m

```
% dp.m
clear all

btime = clock; % to measure time

global n cl T W expW P Net NL NLP ALP Y Tlim AF sc t1 pfa

t1=0;

P=[0.25 0.5 0.75 1];

pfa=3; % n°of points of fixed activities (2=(0.75 1.25) or 3=(0.5 1 1.5))

W=struct('w',{ }); % new

inputNetwork;

decisionVars;

generateExpWs;

generateWs;

durations; % Y (for any activitie, minimal and maximal duration)

generateTlimits; % Tlim (for any node, minimal and maximal
```

```

                                % realization times)
generateTvalues; % Tval (structure with two fields. Tval.st (step)
% Tval.T (List of T values for the nodes)
AF=activitiesToFix(ALP);
AF=xFixedActivities(AF);
sc=size(ALP,2);
generateMainCode;
generateDps1Code;
for i=2:size(ALP,2) % call for the states of the dp
    sc=i; % state code
    generateDpsNCode;
end
rehash; % to load new file: main.m into memory
main;
fprintf('\n\nElapsed time was %2.2f minutes\n',etime(clock,btime)/60);
beep
clear S btime t1 i sc

```

## F.2 generateDps1Code.m

```

% generateDps1Code.m
% Writes to the file dps1.m the instruction that will
% after be called to optimize state 1 of any network (via DP)
% Find state times definition
% State 1 is defined by all node times that are origin of the last node
[OLN,A]=originNodes(Net(2,n)); % OLN - Origins of Last Node
a=ALP(size(ALP,2)); % Activity to optimize - last activity
o=Net(1,a); % of the longest path
k=find(OLN(:)==o); % Eliminate from OLN origin of decision variable

```

```

OLN(k)=[ ];
A(k)=[ ];
S='Best1=[ ];\n';
for i=OLN
    if i~=1
        S=[S,'for t',int2str(i),'=Tval(',int2str(i),').T\n'];
    end
end
S=[S,'[Tn,PTn]=calculateMaxN('];
k=1;
for i=OLN
    j=A(k);
    S=[S,'t',int2str(i),'+W(',int2str(j),').w/x',int2str(j),',,'];
    k=k+1;
end
S(size(S,2))=')';
S=[S,'];\n'];
S=[S,'for t',int2str(o),'=Tval(',int2str(o),').T\n'];
S=[S,'EV=[ ];\n'];
st=(Net(5,a)-Net(4,a))/4;
S=[S,'for x',int2str(a),'=',num2str(Net(4,a)),',:',num2str(st),',:',num2str(Net(5,a)),'\n'];
S=[S,'ev=0;\n'];
S=[S,'for i=1:4\n'];
S=[S,'w=W(',int2str(a),').w(i);\n'];    % new
S=[S,'TN=max(Tn,t',int2str(o),'+w/x',int2str(a),');\n'];    % P=PY7
S=[S,'U=max(0,TN-T);\n'];    % P=PY7
S=[S,'evU=expectedValue(U,PTn);\n'];
S=[S,'IC(i)=x',int2str(a),'*w+cl*evU;\n'];
% Intermediate cost = Delay cost + resource cost of last act.

```

```

% 4 values - Probability=P
S=[S,'end\n'];
S=[S,'F1=IC+rc;\n']; % total cost
S=[S,'ev=expectedValue(F1,P);\n'];
S=[S,'EV=[EV;x',int2str(a),'ev'];\n'];
S=[S,'end\n'];
S=[S,'m=min(EV(:,2));\n'];
S=[S,'k=max(find(EV(:,2)<=m));\n'];
S=[S,'x',int2str(a),'star=EV(k,1);\n'];
S=[S,'v1star=EV(k,2);\n'];
S=[S,'Best1=[Best1;'];
for i=OLN
    if i~=1
        S=[S,'t',int2str(i),''];
    end
end
S=[S,'t',int2str(o),'x',int2str(a),'star,v1star'];\n'];
for i=OLN
    if i~=1
        S=[S,'end\n'];
    end
end
S=[S,'end\n'];
fs1=fopen('dps1.m','w');
fprintf(fs1,S);
fclose(fs1);
OLS=[OLN,o]; % Origins of Last State
k=find(OLS(:)==1); % Eliminate from OLS node 1 (t1=0-no need to include)
OLS(k)=[ ];

```

```
clear OLN A fs1 a i j o k s st
```

### F.3 generateDpsNCode.m

```
% generateDpsNCode.m
% Writes to the file dpsn.m the instruction that will
% after be called to optimize state n (>1) of any network (via DP)
% Find state times definition
% State n is defined by all node times that are origin of the last state
OS=[ ]; % Origins of this State
for i=OLS % Origins of Last State
    [O,A]=originNodes(i);
    OS=[OS,O];
end
OS=reduce(OS);
k=find(OS(:)==1); % Eliminate from OS node 1 (t1=0 - no need to include)
OS(k)=[ ];
a=ALP(size(ALP,2)-sc+1); % Activity to optimize
o=Net(1,a);
S=['Best',int2str(sc),' [ ];\n'];
for i=OS
    S=[S,'for t',int2str(i),'=Tval(',int2str(i),').T\n'];
end
S=[S,'EV=[ ];\n'];
st=(Net(5,a)-Net(4,a))/4;
S=[S,'for x',int2str(a),'=',num2str(Net(4,a)),':',num2str(st),':',
    num2str(Net(5,a)),'\n'];
Tau=[ ]; % Values to input to the previous state
for i=OLS
```

```

j=find(OS(:)==i);
if isempty(j) % for the nodes on this state not in the last state
    Tau=[Tau,i];
    S=[S,[T',int2str(i),'PT',int2str(i),']=calculateMaxN(')];
    k=1;
    [O,A]=originNodes(i);
    for l=O % l - node
        m=A(k); % m - activity      % ——new——
        q=find(ALP==m);
        if isempty(q) % Variavel fixa
            S=[S,'t',int2str(l),'+W(',int2str(m),').w/x',int2str(m),',,'];
        else % Variavel de decisao
            S=[S,'t',int2str(l),'+W(',int2str(m),').w(:)"/x',int2str(m),',,'];
        end
        k=k+1;
    end
    S(size(S,2))=')';
    S=[S,';\n'];
    S=[S,'pT',int2str(i),']=mass(PT',int2str(i),')\n'];
end
end
S=[S,'Fa=[ ];\n'];
S=[S,'pFa=[ ];\n'];
S=[S,'PFa=[ ];\n'];
S=[S,'m=1;\n'];
for i=Tau % Determination of the n° of elements of Ti
    S=[S,'sT',int2str(i),']=size(T',int2str(i),',2)\n'];
end
for i=Tau

```

```

    S=[S,'for i',int2str(i),'=1:sT',int2str(i),'\n'];
end
j=1;
S=[S,'k=min(find(');
for i=OLS
    k=find(Tau(:)==i);
    if isempty(k)
        S=[S,'Best',int2str(sc-1),':',int2str(j),')=t',int2str(i),' & '];
    else
        S=[S,'abs(Best',int2str(sc-1),':',int2str(j),')-T',int2str(i),'(i',int2str(i),'))<=
            Tval(',int2str(i),').st/2+0.000000001', ' & '];
    end
    j=j+1;
end
S(size(S,2)-1)='';
S=[S,'];\n'];
S=[S,'Fa(m)=Best',int2str(sc-1),'(k',int2str(j+1),')'];\n'];
S=[S,'pFa(m)='];
for i=Tau
    S=[S,'pT',int2str(i),'(i',int2str(i),')*'];
end % new
S(size(S,2))='';
S=[S,'];\n'];
S=[S,'m=m+1;\n'];
for i=Tau
    S=[S,'end\n'];
end
S=[S,'PFa=cum(pFa);\n'];
S=[S,'evFa=expectedValue(Fa,PFa);\n'];

```

```

S=[S,'Fb=x',int2str(a),'*W(',int2str(a),').w(:)'+evFa;\n'];
S=[S,'ev=expectedValue(Fb,P);\n'];
S=[S,'EV=[EV;x',int2str(a),',ev);\n'];
S=[S,'end\n'];
S=[S,'m=min(EV(:,2));\n'];
S=[S,'k=max(find(EV(:,2)<=m));\n'];
S=[S,'x',int2str(a),',star=EV(k,1);\n'];
S=[S,'vbstar=EV(k,2);\n'];
S=[S,'Best',int2str(sc),]=[Best',int2str(sc),'];
for i=OS
    S=[S,'t',int2str(i),'];
end
S=[S,'x',int2str(a),',star,vbstar);\n'];
for i=OS
    S=[S,'end\n'];
end
fsn=fopen(['dps',int2str(sc),'.m'],'w');
fprintf(fsn,S);
fclose(fsn);
OLS=OS; % Origins of Last State
k=find(OLS(:)==1); % Eliminate from OLS node 1 (t1=0-no need to include)
OLS(k)=[ ];
clear OS O A S Tau fsn a i j k l m o q

```

## F.4 generateWs.m

```

% generateWs.m
for i=1:n
    k=find(ALP==i);

```



```

if isempty(k) % Variavel fixa
    W(i).w=expW(i); % For all the fix activities W=Wmean
else % Variavel de decisao % For all the decision activities
    W(i).w=generateW(Net(3,i)); % generate 4 W's
end % Net(3,i) contains the parameter
end % lambda of the exponential
clear k % distribution

```

## F.5 durations.m

```

% durations.m
for i=1:n
    k=find(ALP==i);
    if isempty(k) % Variavel fixa
        Y(1,i)=W(i).w/Net(5,i); % Ymin=W/Xmax
        Y(2,i)=W(i).w/Net(4,i); % Ymax=W/Xmin
    else % Variavel de decisao
        Y(1,i)=W(i).w(1)/Net(5,i); % Ymin=Wmin/Xmax
        Y(2,i)=W(i).w(4)/Net(4,i); % Ymax=Wmax/Xmin
    end
end
clear i k

```

## F.6 calculateMaxN.m

```

function [Yt,PYt]=calculateMaxN(varargin)
% Calculates the max of a variable number of arguments
global P
n=nargin;

```

```

if n==1
    Yt=varargin{1};
    if size(Yt,2)==1 % new
        PYt=1; % new
    else % new
        PYt=P;
    end % new
    return
end
V1=varargin{1};
V2=varargin{2};
if size(V1,2)==1 % new
    PV1=1; % new
else % new
    PV1=P; % new
end % new
if size(V2,2)==1 % new
    PV2=1; % new
else % new
    PV2=P; % new
end % new
Yt=calculateMax(V1,V2);
PYt=calculatePmax(Yt,V1,PV1,V2,PV2);
i=3;
while i<=n
    Yt1=Yt;
    PYt1=PYt;
    V1=varargin{1}; % new
    if size(V1,2)==1 % new

```

```
        PV1=1; % new
    else % new
        PV1=P; % new
    end % new
    Yt=calculateMax(Yt1,V1);
    PYt=calculatePmax(Yt,Yt1,PYt1,V1,PV1);
    i=i+1;
end
```



# Apêndice G

## Aproximação 2 (Versão 1)

Neste apêndice vão ser apresentados os procedimentos que sofreram alterações, relativamente à versão 1 sem aproximações (apêndice A).

### G.1 mp.m

```
% mp.m (performs a 3 factorial experiment
% for every fixed variable xi (0.5(Lo),1.0(Medium),
% 1.5(Hi))calculating x1star and the expected value of
% cost for the project with 4 states (ev4))
btime = clock;          % to measure time
% Global variables: lambda, expW, p, P, cl, T, Wi
% Lambda is the parameter of W
lambda=[0.1,0.12,0.05,0.08,0.2,0.04,0.03,0.04,0.024,0.15,0.16];
expW=1./lambda;
p=[0.25 0.25 0.25 0.25];      % probability of Wi
P=[0.25 0.5 0.75 1];        % cumulative probability of Wi
cl=5;                        % delay cost per unit time
T=65;                        % due date
% All W's at their means
```

```
W1=expW(1);
W2=expW(2);
W3=expW(3);
W4=expW(4);
W5=expW(5);
W6=expW(6);
W7=expW(7);
W8=expW(8);
W9=expW(9);
W10=expW(10);
W11=expW(11);
It2min=W1/1.5;
It2max=W1/0.5;
It2=[It2min,It2max];
[stT2,T2]=calculateTvalues(It2);
It3min=It2min+W4/1.5;
It3max=It2max+W4/0.5;
It3=[It3min,It3max];
[stT3,T3]=calculateTvalues(It3);
It4min=W3/1.5;
It4max=W3/0.5;
It4=[It4min,It4max];
[stT4,T4]=calculateTvalues(It4);
It5min=It4min+W8/1.5;
It5max=It4max+W8/0.5;
It5=[It5min,It5max];
[stT5,T5]=calculateTvalues(It5);
It6min=It3min+W7/1.5;
It6max=It3max+W7/0.5;
```



```

disp(' x2 x3 x5 x6 x8 x9 x10 x1star ev4');
for i=1:size(Res,1)
    fprintf('%6.2f %6.2f %6.2f %6.2f %6.2f %6.2f %6.2f %6.2f %8.2f\n',Res(i,:));
end
disp('Best value was:');
disp(' x2 x3 x5 x6 x8 x9 x10 x1star ev4');
fprintf('%6.2f %6.2f %6.2f %6.2f %6.2f %6.2f %6.2f %6.2f %8.2f\n',Res(k,:));
% -----
% Write result to a file (to export to excel)
fid = fopen('resBestAll.txt','w');
fprintf(fid,'\nThis is a search for x10=[0.75:0.5:1.25] (around 1)\n\n',cl);
fprintf(fid,'\nCurrent parameters');
fprintf(fid,'\nDue date (T) = %3d',T);
fprintf(fid,'\nCost of lateness (cl) = %2d\n\n',cl);
fprintf(fid,'x2,x3,x5,x6,x8,x9,x10,x1star,ev4\n');
for i=1:size(Res,1)
    fprintf(fid,'%6.2f,%6.2f,%6.2f,%6.2f,%6.2f,%6.2f,%6.2f,%6.2f,%8.2f\n',Res(i,:));
end
fprintf(fid,'\nBest value was:\n');
fprintf(fid,'x2,x3,x5,x6,x8,x9,x10,x1star,ev4\n');
fprintf(fid,'%6.2f,%6.2f,%6.2f,%6.2f,%6.2f,%6.2f,%6.2f,%6.2f,%8.2f',Res(k,:));
fclose(fid);
% -----
% Write Best1 to a file (to export to excel)
fid = fopen('resBest1.txt','w');
fprintf(fid,'t2,t4,t6,x11star,ev1\n');
for i=1:size(Best1,1)
    fprintf(fid,'%6.4f,%6.4f,%6.4f,%6.4f,%8.4f\n',Best1(i,:));
end

```



```

fclose(fid);
% -----
% Write Best2 to a file (to export to excel)
fid = fopen('resBest2.txt','w');
fprintf(fid,'t2,t3,t4,x7star,ev2\n');
for i=1:size(Best2,1)
    fprintf(fid,'%6.4f,%6.4f,%6.4f,%6.4f,%8.4f\n',Best2(i,:));
end
fclose(fid);
% -----
% Write Best3 to a file (to export to excel)
fid = fopen('resBest3.txt','w');
fprintf(fid,'t2,t4,x4star,ev3\n');
for i=1:size(Best3,1)
    fprintf(fid,'%6.4f,%6.4f,%6.4f,%8.4f\n',Best3(i,:));
end
fclose(fid);
% -----
% Write Best4 to a file (to export to excel)
fid = fopen('resBest4.txt','w');
fprintf(fid,'x1star,ev4\n');
for i=1:size(Best4,1)
    fprintf(fid,'%6.4f,%8.4f\n',Best4(i,:));
end
fclose(fid);
% -----
% Write result to a file (to read later into Matlab Workspace)
save resultBestAll T cl Best1 Best2 Best3 Best4 Res
fprintf('\n\nElapsed time was %2.2f minutes\n',etime(clock,btime)/60);

```

```
clear W* I* T* P x* st* p i cl m k btime lambda
beep
```

## G.2 mediumprojects1.m

```
% State is s1:(t4,t5,t6)
% Global variables: Best1
% Decision variable is x11
Best1=[ ];
% resource cost for activities whose resource level is fixed
rcf=x2*W2+x3*W3+x5*W5+x6*W6+x8*W8+x9*W9+x10*W10;
for t4=T4
for t5=T5
% Calculation of Y7: Y7=max(t4+W9/x9,W10/x10+max(t2+W5/x5,t4+W8/x8))
%Y7=max(t4+W9/x9,t5+W10/x10)
a1=t4+W9/x9; % This is a number now, not a random variable
a2=t5+W10/x10;
y7=max(a1,a2);
% Calculation of T7, U , ...
for t6=T6
F1=[ ];
for x11=0.5:0.25:1.5
ev=0;
t7=max(y7,t6+W11/x11);
u=max(0,t7-T);
tc=cl*u; % Tardiness cost
rc11=x11*W11;
f1=tc+rcf+rc11; % total cost
F1=[F1;x11,f1];
```

```

        end
        m=min(F1(:,2));
        k=max(find(F1(:,2)<=m)));
        x11star=F1(k,1);
        v1star=F1(k,2);
        Best1=[Best1;t4,t5,t6,x11star,v1star];
    end
end
clear F1 a* t* u y7 f1 x11star rcf rc11 i11 w11 tc m k v1star
%['t4, t5, t6, x11star, ev1']
%Best1
% f1(t4,t5,t6) = min eV(x11*W11+cl*eV(U))
% x11
%
% U=max(0,T7-T)
% T7=max(Y7,t6+w11/x11)
% Y7=max(t4+W9/x9,t5+W10/x10)

```

### G.3 mediumprojects2.m

```

% State is s2:(t2,t3,t4)
% decision variable is x7
% Global variables: Best2
Best2=[ ];
for t2=T2
    a1=t2+W6/x6;          % this is now a value, not a random variable
    a3=t2+W5/x5;
    for t3=T3

```

```

for t4=T4
    a4=t4+W8/x8;
    t5n=max(a3,a4);
    F2=[ ];
    for x7=0.5:0.25:1.5
        a2=t3+W7/x7;
        t6n=max(a1,a2);
        k=min( find( Best1(:,1)==t4 & abs(Best1(:,2)-t5n)<=stT5/2 &
            abs(Best1(:,3)-t6n)<=stT6/2 ) );
        f1=Best1(k,5);
        f2=x7*W7+f1;
        F2=[F2;x7,f2];
    end
    end
    m=min(F2(:,2));
    k=max(find(F2(:,2)<=m));
    x7star=F2(k,1);
    v2star=F2(k,2);
    Best2=[Best2;t2,t3,t4,x7star,v2star];
end
end
end
clear F2 a* f* x7star t* k m v2star
% ['t2, t3, t4, x7star, ev2']
% Best2
% f2(t2,t3,t4) = min eV(x7*W7+eV(f1(t4,T5,T6))
% x7
%
% T5 = max(t2+W5/x5,t4+W8/x8)
% T6 = max(t2+W6/x6,t3+W7/x7)

```

## G.4 mediumprojects3.m

```

% State is s3:(t2)
% Decision variable is x4
% Global variables: Best3
Best3=[ ];
a2=W2/x2;          % this is now a value, not a random variable
t4n=W3/x3;
for t2=T2
F3=[ ];
  for x4=0.5:0.25:1.5
    a1=t2+W4/x4;
    t3n=max(a1,a2);
    k=min( find( Best2(:,1)==t2 & abs(Best2(:,2)-t3n)<=stT3/2 &
              abs(Best2(:,3)-t4n)<=stT4/2 ) );
    f2=Best2(k,5);
    f3=x4*W4+f2;
    F3=[F3;x4,f3];
  end
  m=min(F3(:,2));
  k=max(find(F3(:,2)<=m));
  x4star=F3(k,1);
  v3star=F3(k,2);
  Best3=[Best3;t2,x4star,v3star];
end
clear F3 a* x4star t* f* k m v3star
% ['t2, t4, x4star, ev3']
% Best3
% f3(t2) = min eV(x4*W4+ev(f2(t2,T3,T4))

```

```

% x4
% T4 = W3/x3 (from mp.m)
% T3 = max(t2+W4/x4,W2/x2)

```

## G.5 mediumprojects4.m

```

% State is s4:t1=0
% Decision variable is x1
% Global variables: Best4
Best4=[ ];
F4=[ ];
for x1=0.5:0.25:1.5
    t2n=W1/x1;
    k=min(find(abs(Best3(:,1)-t2n)<=stT2/2));
    f3=Best3(k,3);
    f4=x1*W1+f3;
    F4=[F4;x1,f4];
end
m=min(F4(:,2));
k=max(find(F4(:,2)<=m));
x1star=F4(k,1);
v4star=F4(k,2);
Best4=[x1star,v4star];
clear F* F4 x1star ev* t2n f* k m v4star
% f4(0) = min eV(x1*W1+eV(f3(T2))
% x1
%
% T2 = W1/X1

```

# Apêndice H

## Aproximação 2 (Versão 2)

Neste apêndice vão ser apresentados os procedimentos que sofreram alterações, relativamente à versão 2 sem aproximações (apêndice B).

### H.1 dp.m

```
% dp.m
clear all

btime = clock; % to measure time

global n cl T W expW P Net NL NLP ALP Y Tlim AF sc t1 pfa
t1=0;
P=[0.25 0.5 0.75 1];
pfa=3; % n°of points of fixed activities (2=(0.75 1.25) or 3=(0.5 1 1.5))
inputNetwork;
decisionVars;
% generateWs;
generateExpWs;
W=expW; %new
durations; % Y (for any activitie, minimal and maximal duration) - changed
generateTlimits; % Tlim (for any node, minimal and maximal
```

```

        % realization times)
generateTvalues; % Tval (structure with two fields. Tval.st (step)
        % Tval.T (List of T values for the nodes)
AF=activitiesToFix(ALP);
AF=xFixedActivities(AF);
sc=size(ALP,2);
generateMainCode;
generateDps1Code;
for i=2:size(ALP,2) % call for the states of the dp
    sc=i; % state code
    generateDpsNCode;
end
rehash; % to load new file: main.m into memory
main;
fprintf('\n\nElapsed time was %2.2f minutes\n',etime(clock,btime)/60);
beep
clear S btime t1 i sc

```

## H.2 generateDps1Code.m

```

% generateDps1Code.m
% Writes to the file dps1.m the instruction that will
% after be called to optimize state 1 of any network (via DP)
% Find state times definition
% State 1 is defined by all node times that are origin of the last node
[OLN,A]=originNodes(Net(2,n)); % OLN - Origins of Last Node
a=ALP(size(ALP,2)); % Activity to optimize - last activity
o=Net(1,a); % of the longest path
k=find(OLN(:)==o); % Eliminate from OLN origin of decision variable

```



```

OLN(k)=[ ];
A(k)=[ ];
S='Best1=[ ];\n';
for i=OLN
    if i~=1
        S=[S,'for t',int2str(i),'=Tval(',int2str(i),').T\n'];
    end
end
S=[S,['Tn,PTn]=calculateMaxN('); %pode ficar igual - nao ligar ao PTn
k=1;
for i=OLN
    j=A(k);
    S=[S,'t',int2str(i),'+W(',int2str(j),')/x',int2str(j),',,']; % W(4) e nao W(4,:)
    k=k+1;
end
S(size(S,2))=')';
S=[S,;\n'];
S=[S,'for t',int2str(o),'=Tval(',int2str(o),').T\n'];
S=[S,'EV=[ ];\n'];
st=(Net(5,a)-Net(4,a))/4;
S=[S,'for x',int2str(a),'=',num2str(Net(4,a)),',:',num2str(st),':',
    num2str(Net(5,a)),'\n'];
S=[S,'ev=0;\n'];
S=[S,'w=W(',int2str(a),');\n'];
S=[S,'TN=max(Tn,t',int2str(o),'+w/x',int2str(a),');\n']; % P=PY7
S=[S,'U=max(0,TN-T);\n']; % P=PY7
S=[S,'IC=x',int2str(a),'*w+cl*U;\n'];
% Intermediate cost = Delay cost + resource cost of last act.
% 4 values - Probability=P

```

```

S=[S,'F1=IC+rc;\n']; % total cost
S=[S,'EV=[EV;x',int2str(a),'F1];\n'];
S=[S,'end\n'];
S=[S,'m=min(EV(:,2));\n'];
S=[S,'k=max(find(EV(:,2)<=m));\n'];
S=[S,'x',int2str(a),'star=EV(k,1);\n'];
S=[S,'v1star=EV(k,2);\n'];
S=[S,'Best1=[Best1;'];
for i=OLN
    if i~=1
        S=[S,'t',int2str(i),''];
    end
end
S=[S,'t',int2str(o),'x',int2str(a),'star,v1star'];\n'];
for i=OLN
    if i~=1
        S=[S,'end\n'];
    end
end
S=[S,'end\n'];
fs1=fopen('dps1.m','w');
fprintf(fs1,S);
fclose(fs1);
OLS=[OLN,o]; % Origins of Last State
k=find(OLS(:)==1); % Eliminate from OLS node 1 (t1=0-no need to include)
OLS(k)=[ ]; %NOVO
clear OLN A fs1 a i j o k s st

```

### H.3 generateDpsNCode.m

```

% generateDpsNCode.m
% Writes to the file dpsn.m the instruction that will
% after be called to optimize state n (>1) of any network (via DP)
% Find state times definition
% State n is defined by all node times that are origin of the last state
OS=[]; % Origins of this State
for i=OLS % Origins of Last State
    [O,A]=originNodes(i);
    OS=[OS,O];
end
OS=reduce(OS);
k=find(OS(:)==1); % Eliminate from OS node 1 (t1=0 - no need to include)
OS(k)=[ ];
a=ALP(size(ALP,2)-sc+1); % Activity to optimize
o=Net(1,a);
S=['Best',int2str(sc),'=[ ];\n'];
for i=OS
    S=[S,'for t',int2str(i),'=Tval(',int2str(i),').T\n'];
end
S=[S,'EV=[ ];\n'];
st=(Net(5,a)-Net(4,a))/4;
S=[S,'for x',int2str(a),'=',num2str(Net(4,a)),',:',num2str(st),':',
    num2str(Net(5,a)),'\n'];
Tau=[]; % Values to input to the previous state
for i=OLS
    j=find(OS(:)==i);
    if isempty(j) % for the nodes on this state not in the last state

```

```

Tau=[Tau,i];
S=[S,['T',int2str(i),'PT',int2str(i)]=calculateMaxN(')];
k=1;
[O,A]=originNodes(i);
for l=O % l - node
    m=A(k); % m - activity
    S=[S,'t',int2str(l),'+W(',int2str(m),')/x',int2str(m),','];
    k=k+1;
end
S(size(S,2))=');
S=[S,;\n'];
end
end
S=[S,'Fa=[ ];\n'];
j=1;
S=[S,'k=min(find(');
for i=OLS
    k=find(Tau(:)==i);
    if isempty(k)
        S=[S,'Best',int2str(sc-1),'(:',int2str(j),')=t',int2str(i),' & '];
    else
        S=[S,'abs(Best',int2str(sc-1),'(:',int2str(j),')-T',int2str(i),'(1))<=
            Tval(',int2str(i),').st/2+0.000000001', & '];
    end
    j=j+1;
end
S(size(S,2)-1)=');
S=[S,;\n'];
S=[S,'Fa=Best',int2str(sc-1),'(k',int2str(j+1),');\n'];

```

```

S=[S,'Fb=x',int2str(a),'*W(',int2str(a),')+Fa;\n']; %evFa->Fa
S=[S,'EV=[EV;x',int2str(a),',Fb];\n']; %ev ->Fb
S=[S,'end\n'];
S=[S,'m=min(EV(:,2));\n'];
S=[S,'k=max(find(EV(:,2)<=m));\n'];
S=[S,'x',int2str(a),'star=EV(k,1);\n'];
S=[S,'vstar=EV(k,2);\n'];
S=[S,'Best',int2str(sc),'=[Best',int2str(sc),','];
for i=OS
    S=[S,'t',int2str(i),','];
end
S=[S,'x',int2str(a),'star,vstar];\n'];
for i=OS
    S=[S,'end\n'];
end
fsn=fopen(['dps',int2str(sc),'.m'],'w');
fprintf(fsn,S);
fclose(fsn);
OLS=OS; % Origins of Last State
k=find(OLS(:)==1); % Eliminate from OLS node 1 (t1=0-no need to include)
OLS(k)=[ ]; %NOVO
clear OS O A S Tau fsn a i j k l m o

```

## H.4 durations.m

```

% durations.m
for i=1:n
    Y(1,i)=W(i)/Net(5,i); % Ymin=W/Xmax
    Y(2,i)=W(i)/Net(4,i); % Ymax=W/Xmin

```

```

end
clear i

```

## H.5 calculateMaxN.m

```

function [Yt,PYt]=calculateMaxN(varargin)
% Calculates the max of a variable number of arguments
global P
PYt=1;
n=nargin;
if n==1
    Yt=varargin{1};
    %PYt=P;
    return
end
Yt=calculateMax(varargin{1},varargin{2});
%PYt=calculatePmax(Yt,varargin{1},P,varargin{2},P);
i=3;
while i<=n
    Yt1=Yt;
    %PYt1=PYt;
    Yt=calculateMax(Yt1,varargin{i});
    %PYt=calculatePmax(Yt,Yt1,PYt1,varargin{i},P);
    i=i+1;
end

```

# Apêndice I

## Algoritmo “Electromagnético”

### I.1 EM

EM( $m$ , MAXITER, LSITER,  $\delta$ )

1. *Initialize*( )
2.  $iteration \leftarrow 1$
3. **while**  $iteration < MAXITER$  **do**
4.      $Local(LSITER, \delta)$
5.      $F \leftarrow CalcF( )$
6.      $Move(F)$
7.      $iteration \leftarrow iteration + 1$
8. **end while**

### I.2 Initialize

Initialize( )

1. **for**  $i = 1$  **to**  $m$  **do**
2.     **for**  $k = l$  **to**  $n$  **do**
3.          $\lambda \leftarrow U(0, 1)$
4.          $x_k^i \leftarrow l_k + \lambda(u_k - l_k)$

5. **end for**
6. Calculate  $f(x^i)$
7. **end for**
8.  $x^{best} \leftarrow \arg \min\{f(x^i), \forall i\}$

### I.3 Local

Local(LSITER,  $\delta$ )

1.  $counter \leftarrow 1$
2.  $Length \leftarrow \delta(\max_k\{u_k - l_k\})$
3. **for**  $i = 1$  **to**  $m$  **do**
4.     **for**  $k = 1$  **to**  $n$  **do**
5.          $\lambda_1 \leftarrow U(0, 1)$
6.         **while**  $counter < LSITER$  **do**
7.              $y \leftarrow x^i$
8.              $\lambda_2 \leftarrow U(0, 1)$
9.             **if**  $\lambda_1 > 0.5$  **then**
10.                  $y_k \leftarrow y_k + \lambda_2(Length)$
11.             **else**
12.                  $y_k \leftarrow y_k - \lambda_2(Length)$
13.             **end if**
14.             **if**  $f(y) < f(x^i)$  **then**
15.                  $x^i \leftarrow y$
16.              $counter \leftarrow LSITER - 1$
17.             **end if**
18.              $counter \leftarrow counter + 1$
19.         **end while**
20.     **end for**
21. **end for**



22.  $x_{best} \leftarrow \arg \min\{f(x^i), i = 1, 2, \dots, m\}$

## I.4 CalcF

CalcF( ):F

1. **for**  $i = 1$  **to**  $m$  **do**
2.      $q \leftarrow \exp\left(-n \frac{f(x^i) - f(x^{best})}{\sum_{k=1}^m (f(x^k) - f(x^{best}))}\right)$
3.      $F^i \leftarrow 0$
4. **end for**
5. **for**  $i = 1$  **to**  $m$  **do**
6.     **for**  $j = 1$  **to**  $m$  **do**
7.         **if**  $f(x^j) < f(x^i)$  **then**
8.              $F^i \leftarrow F^i + (x^j - x^i) \frac{q^i q^j}{\|x^j - x^i\|^2}$  {Attraction}
9.         **else**
10.              $F^i \leftarrow F^i - (x^j - x^i) \frac{q^i q^j}{\|x^j - x^i\|^2}$  {Repulsion}
11.         **end if**
12.     **end for**
13. **end for**

## I.5 Move

Move(F)

1. **for**  $i = 1$  **to**  $m$  **do**
2.     **if**  $i \neq best$  **then**
3.          $\lambda \leftarrow U(0, 1)$
4.          $F^i \leftarrow \frac{F^i}{\|F^i\|}$
5.         **for**  $k = 1$  **to**  $n$  **do**
6.             **if**  $F_k^i > 0$  **then**
7.                  $x_k^i \leftarrow x_k^i + \lambda F_k^i (u_k - x_k^i)$

8.           **else**
9.            $x_k^i \leftarrow x_k^i + \lambda F_k^i(x_k^i - l_k)$
10.          **end if**
11.         **end for**
12.         **end if**
13.         **end for**

# Apêndice J

## Algoritmo EM - RCPSP

Os procedimentos que vão ser apresentados a seguir foram adaptados dos originais, pelo que agradecemos aos seus autores a sua disponibilização [BirbilFang00]. Foram desenvolvidos de raiz TotalCost.m, Cpm.m, OriginNodes.m e MeanRes.m.

### J.1 GOfunc.m

```
btime = clock; % to measure time
global lbda T cl W GLOBALCOUNT Net n
% Net - Each column is an activity
Net =[1 1 1 2 2 2 3 4 4 5 6;... % Origin node
      2 3 4 3 5 6 6 5 7 7 7]; % Target node
lbda=[0.1,0.12,0.05,0.08,0.2,0.04,0.03,0.04,0.024,0.15,0.16];
T=65;
cl=5;
n=11;
Params = struct('Optimum', 0.0, ...
               'FunctionName', strvcat('totalcost'), ...
               'OutputFileName', 'func.out', ...
               'UpperBounds', [1.5 1.5 1.5 1.5 1.5 1.5 1.5 1.5 1.5 1.5], ...
```

```

'LowerBounds', [0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5], ...
'OptimumCoords', [1 1 1 1 1 1 1 1 1 1], ...
'NumOfIterations', 275, ...
'NumOfRuns', 100, ...
'n', 11, ...
'm', 15, ...
'LocalIterations', 1, ...
'delta', 0.05, ...
'pertpar', 0.25, ...
'graph', 0);
[best,Results] = EMVIS(Params);
save Workspace
fprintf('\n\nElapsed time was %2.2f minutes\n',etime(clock,btime)/60);
beep

```

## J.2 EMVIS.m

```

% *****
% The general scheme of the algorithm.
% The parameters is fed as a structure
% called Params.
% *****
function [best,Results] = EMVIS(Params)
global W
Results = struct('Ws', zeros(Params.n, 1), ...
    'Xs', zeros(Params.n, 1), ...
    'f', 0);
% The structures for the particles called
% ions and the structure for the current

```

```

% best particle are determined. Also
% the parameters are assigned to the new
% structure Par
[ions, best, Par] = SetStruct(Params);
% Output file is opened.
fout = fopen(Par.OUTNAME, 'w');
% Function Name is written
fprintf(fout, '\n\n%s\n\n', Par.FUNCNAME);
for i=1 : Par.NUMOFRUNS
    fprintf(fout, '%d.Run : \n', i);
    % Random Seed is initiated.
    rand('state',sum(100*clock));
    % Points are randomly generated
    % from the feasible region.
    [ions, best] = Initialize(ions, best, Par);
    for j=1 : Par.MAXGEN
        % If user wants to follow algorithm visually
        % she/he can set the parameter (Par.GRAPH) to 1
        % NOTE: This is allowed if the dimension is 2.
        if (Par.GRAPH)
            Draw(ions, best, Par);
        end
        % The calculation of total force
        [ions] = CalcF(ions, best, Par);
        % The points are moved in the direction of total force.
        [ions, best] = Move(ions, best, Par);
        % Simple local search is applied to current best point.
        [ions, best] = Local(ions, best, Par);
    end
end

```

```

indx=best.index;
%for k=1 : Par.N
    % fprintf (fout, 'x%d = %f\n', k, ions(indx).coord(k));
%end
%fprintf (fout, 'Best Value: %f\n', best.objvalue);
%fprintf (fout, 'Count:%d\n\n', best.count);
Results(i).Ws=W;
Results(i).Xs=ions(indx).coord;
Results(i).f=best.objvalue;
end
fclose(fout);

```

### J.3 SetStruct.m

```

% *****
% The structures for the points (ions),
% current best point (best), and for the
% parameters (Par) are constructed here.
% *****
function [ions, best, Par] = SetStruct(Params)
% The structure for the points.
ions(1:Params.m) = struct('Fi', zeros(Params.n,1), ... % Total force
    'Q', 0, ... % Charge
    'coord', zeros(Params.n,1), ... % Coordinated
    'objvalue', realmax); % Objective Function Value
% The structure for the current best point
best = struct('count', 0, ... % The number of function evaluations
    'index', 0, ... % Index of the current best point
    'objvalue', realmax); % Objective function value of the current best point.

```

```

% The structure for the parameters
Par = struct('OPTIMUM', realmax, ... % Known optimum
'FUNCNAME', '', ... % Name of the function
'OUTNAME', '', ... % Output file name
'UB', zeros(Params.n, 1), ... % Upper bounds on each dimension
'LB', zeros(Params.n, 1), ... % Lower bounds on each dimension
'OPTCOORDS', zeros(Params.n, 1), % The coords of the optimum point.
'MAXGEN', 0, ... % Maximum number of iterations allowed
'NUMOFRUNS', 0, ... % Number of runs on each function
'N', 0, ... % Dimension of the problem
'M', 0, ... % Number of points in the population
'LS', 0, ... % Number of Local Search iterations
'DELTA', 0, ... % Local search parameter
'PERTPAR', 0, ... % Perturbation parameter
'GRAPH', 1); % If 0 nothing is drawn,
                % otherwise a visual demonstration is shown.

% NOTE: The dimension should be 2.
% The parameters are assigned to Par structure.
Par.OPTIMUM = Params.Optimum;
Par.FUNCNAME = Params.FunctionName;
Par.OUTNAME = Params.OutputFileName;
Par.UB = Params.UpperBounds;
Par.LB = Params.LowerBounds;
Par.OPTCOORDS = Params.OptimumCoords;
Par.MAXGEN = Params.NumOfIterations;
Par.NUMOFRUNS = Params.NumOfRuns;
Par.N = Params.n;
Par.M = Params.m;
Par.LS = Params.LocalIterations;

```

```

Par.DELTA = Params.delta;
Par.PERTPAR = Params.pertpar;
Par.GRAPH = Params.graph;

```

## J.4 Initialize.m

```

% *****
% Initialization of the algorithm.
% The points are sampled from the
% feasible region.
% *****
function [ions, best] = Initialize(ions, best, Par)
global GLOBALCOUNT lbda W
GLOBALCOUNT = 0;
Z=rand(1,Par.N); % generates n random uniformly
                 % distributed values between 0 and 1
for k=1:Par.N
    W(k)=-log(1-Z(k))/lbda(k);
end
for i=1 : Par.M
    for j=1 : Par.N
        range = Par.UB(j) - Par.LB(j);
        ions(i).coord(j) = Par.LB(j) + rand*range;
    end
    [ions(i).objvalue]=feval(Par.FUNCNAME,ions(i).coord);
end
% The current best point index and the objective
% function value are stored.
[best.objvalue best.index] = min([ions.objvalue]);

```



## J.5 TotalCost.m

```
function c = totalcost(X)
    global W T cl n
    global GLOBALCOUNT
    GLOBALCOUNT = GLOBALCOUNT + 1;
    c=0;
    for i=1:n
        c=c+X(i)*W(i);
        Y(i)=W(i)/X(i);
    end
    tf=cpm(Y);
    c=c+cl*max(0,tf-T);
```

## J.6 Cpm.m

```
function T7=cpm(Y)
    % cpm.m Gives the time of realization of the
    % last node (T7) of the network using cpm
    % Y is the duration of the activities.
    global Net n
    TNode=[]; % TNode stores the time of realization of each node
                % The last one is T7
    TNode(1)=0; % TNode1=0
    for i=2:Net(2,n) [Orig,Act]=originNodes1(Net,i); % Y=W/x (x=1) - duration
    TNode(i)=max(TNode(Orig)+Y(Act)); % TNodej=max{TNodei+Y}; j=2,...,n
    end % i->j
    M=size(TNode,2);
    T7=TNode(M);
```

## J.7 OriginNodes.m

```
function [O,A]=originNodes1(Net1,t);
% t - target
% for a target, find all origins
% A - activities (origin-target)
A=find(Net1(2,:)==t);
O=Net1(1,A);
```

## J.8 CalcF.m

```
% *****
% The total force on each point is
% calculated here.
% *****
function [ions] = CalcF(ions, best, Par)
MINDIST = 1.0e-10;
totdif = 0.0;
temp = 0.0;
dist = 0.0;
aij = zeros(Par.N, 1);
farthest = -1.0;
tempvec = zeros(Par.N, 1);
% The point which is the farthest from the current best
% point is selected to be the perturbed point.
for i=1 : Par.M
% The total deviation from the current best objective
% function value. This is used in the calculation of
% charges below.
```

```

    totdif = totdif + ions(i).objvalue-best.objvalue;
  if (i~=best.index)
    aij = ions(best.index).coord - ions(i).coord;
    dist = norm(aij);
    if (dist >farthest)
      farthest = dist;
      p = i;
    end
  end
end
end

```

## J.9 Move.m

```

% *****
% The points except the current best point
% are moved in the direction of the total
% force applied on them.
% *****
function [ions, best] = Move(ions, best, Par)
global GLOBALCOUNT;
nom = 0.0;
Tiny = 0.0;
for i=1 : Par.M
  if (i~=best.index)
    Tiny = rand;
    % Normalize the total force on each point
    % to maintain the feasibility.
    nom = norm(ions(i).Fi);
    ions(i).Fi = (1.0/nom)*ions(i).Fi;
  end
end

```

```

for j=1 : Par.N
    if (ions(i).Fi(j) >= 0)
        ions(i).coord(j) = ions(i).coord(j) +
            Tiny*ions(i).Fi(j)*(Par.UB(j) - ions(i).coord(j));
    else
        ions(i).coord(j) = ions(i).coord(j) +
            Tiny*ions(i).Fi(j)*(ions(i).coord(j) - Par.LB(j));
    end
end
end
ions(i).objvalue = feval(Par.FUNCNAME, ions(i).coord);
% Keep the record of best number of function evaluations.
if (ions(i).objvalue < best.objvalue)
    best.count = GLOBALCOUNT;
end
end
end
end
% Update the current best point and the current best
% objective function value.
[best.objvalue best.index] = min([ions.objvalue]);

```

## J.10 Local.m

```

% *****
% Simple local search algorithm.
% This is applied to the current best
% point only.
% *****
function [ions, best] = Local(ions, best, Par)
global GLOBALCOUNT;

```

```

% A temporary particle is constructed to store
% the information about the current best
% point.
TempIon = struct('Fi', zeros(Par.N,1), ...
'Q', 0, ...
'coord', zeros(Par.N,1), ...
'objvalue', realmax);
Threshold = 0.0;
Tiny = 0.0;
Tiny1 = 0.0;
TempVec = zeros(Par.N);
TempVec = Par.UB - Par.LB;
Threshold = max(TempVec);
Threshold = Threshold * Par.DELTA;
% Current best point is assigned to be point i.
i=best.index;
for j=1 : Par.N
    count = 0;
    Tiny1 = rand;
    while (count <Par.LS)
        TempIon = ions(i);
        Tiny = rand;
        FLAG = 1;
        % Make a tiny increment in the corresponding dimension
        if (Tiny1 >0.5)
            if (TempIon.coord(j)+Tiny*Threshold <Par.UB(j))
                TempIon.coord(j) = TempIon.coord(j) + Tiny*Threshold;
            else
                FLAG = 0;

```

```

    end
    % Make a tiny decrement in the corresponding dimension
else
    if (TempIon.coord(j)-Tiny*Threshold >Par.LB(j))
        TempIon.coord(j) = TempIon.coord(j) - Tiny*Threshold;
    else
        FLAG = 0;
    end
end
end
% Update the best values.
if (FLAG)
    TempIon.objvalue = feval(Par.FUNCNAME, TempIon.coord);
    if (TempIon.objvalue <best.objvalue)
        ions(i) = TempIon;
        best.objvalue = TempIon.objvalue;
        best.count = GLOBALCOUNT;
    else
        count = Par.LS - 1;
    end
end
count = count + 1;;
end
end

```

## J.11 MeanRes.m

```

Ws1=[];
for i=1:size(Results,2)
    Ws1(i,:)=Results(i).Ws;

```

```
end
Wm=[ ];
for i=1:11
    Wm=[Wm,mean(Ws1(:,i))];
end
Wm
F=[Results.f];
fm=mean(F)
Xs=[Results.Xs];
Xm=[ ];
for i=1:11
    Xm=[Xm,mean(Xs(i,:))];
end
Xm
```