



Universidade do Minho
Escola de Engenharia

António Jorge da Silva Leal

**Algoritmos de Investigação Operacional para um
problema de sequenciamento de projectos**

António Jorge da Silva Leal | **Algoritmos de Investigação Operacional para um problema de sequenciamento de projectos**

UMinho|2007

Setembro de 2007



Universidade do Minho

Escola de Engenharia

António Jorge da Silva Leal

Algoritmos de Investigação Operacional para um problema de sequenciamento de projectos

Tese de Mestrado em Engenharia
Área de Especialização em Engenharia Industrial

Trabalho realizado sob a orientação do
Professor Doutor Cláudio Manuel Martins Alves
Professor Doutor Filipe Pereira Pinto Cunha Alvelos

Setembro de 2007

É AUTORIZADA APENAS A CONSULTA DESTA TESE PARA
EFEITOS DE INVESTIGAÇÃO, MEDIANTE DECLARAÇÃO ESCRITA DO
INTERESSADO, QUE A TAL SE COMPROMETE

António Jorge da Silva Leal

Resumo

Os problemas de sequenciamento (ou escalonamento) de projectos com recursos escassos em que as actividades podem ser executadas segundo diferentes modos têm merecido um interesse crescente por parte da comunidade científica. Uma das justificações prende-se com a adequação desses problemas à realidade dos projectos. Tipicamente, é possível realizar as actividades de um projecto de forma mais rápida desde que sejam investidos mais recursos. Na terminologia anglo-saxónica, esses problemas são designados por *Multi-Mode Resource Constrained Project Scheduling Problem* (MRCPSP).

O objectivo do MRCPSP consiste em determinar o instante de tempo em que cada actividade deve ser iniciada e o modo que deve ser usado para realizar cada uma das actividades, sem que para isso sejam gastos mais recursos (renováveis e não renováveis) do que aqueles que estão disponíveis. Nesta dissertação, considera-se como objectivo a minimização da duração global do projecto, e não é permitida a interrupção das actividades.

Um dos objectivos desta dissertação foi de avaliar a qualidade de modelos alternativos para o MRCPSP. Para tal, recorreremos ao princípio da decomposição de Dantzig-Wolfe que aplicámos a um modelo compacto original descrito na literatura. Três novos modelos foram derivados usando esse princípio. Os resultados computacionais levados a cabo com base em instâncias da literatura permitiram aferir a qualidade dessas decomposições. Uma das decomposições provou ser de boa qualidade, obtendo-se a partir dela limites inferiores próximos do óptimo.

Nesta dissertação, também se investigou uma heurística construtiva para o cálculo de limites superiores (soluções válidas) para o MRCPSP em que existem recursos não renováveis. A heurística foi implementada numa linguagem de programação, e testada usando uma vez mais instâncias da literatura. Os testes revelaram o bom desempenho do algoritmo.

Abstract

Project scheduling problems with scarce resources where activities can be done in different modes have deserved an increasing interest from the scientific community. One may justify this situation by the adequacy of these problems to the reality of the real world projects. Typically, it is possible to execute the activities of a project faster if more resources are used for this purpose. This type of problems is formally known as the *Multi-Mode Resource Constrained Project Scheduling Problem* (MRCPSP).

The objective of the MRCPSP is to determine when to start each activity and which mode should be used for each activity, ensuring at the same time that the limit of resources (renewable or non renewable) is never exceeded. In this dissertation, we deal with the problem where one wants to minimize the total duration without interrupting any activity.

One of the goals of this dissertation was to evaluate the quality of alternative models for the MRCPSP. For that purpose, we used the principle of Dantzig-Wolfe decomposition that we applied to a compact and original model described in the literature. Three new models were derived using this principle. The computational tests realized with instances from the literature allowed to evaluate the quality of these decompositions. One of the decompositions proved to be of good quality. By using it, we were able to derive good lower bounds near from the optimum.

In this dissertation, we also investigated a constructive heuristic for the computation of upper bounds (feasible solutions) for the MRCPSP, for the case where there are non renewable resources. The heuristic was implemented in a programming language and tested using once more instances from literature. The experiments proved that the heuristic performs well.

Agradecimentos

Ao longo da elaboração desta dissertação foram muitas as pessoas que de alguma forma contribuíram para a mesma. Com elas fico em dívida e aqui lhes presto a minha gratidão.

Ao Prof. Doutor Cláudio Alves, orientador desta dissertação, por todo o seu empenho, não só científico mas também humano, pela paciência, motivação e dedicação que teve para comigo. Foi de facto um enorme prazer poder desenvolver este trabalho sob a sua coordenação e orientação.

Ao Prof. Doutor Filipe Alvelos, pela preciosa ajuda na parte das decomposições e pela boa disposição que sempre demonstrou.

Ao Sr. Eng.º Carlos Guimarães, sócio gerente da Auditene - Projectos e Auditorias Energéticas Lda, por toda a compreensão, apoio, disponibilidade e incentivo que me dedicou. Esta dissertação só foi possível graças à sua percepção de que o investimento na formação é um factor de desenvolvimento pessoal a nível profissional e social e, conseqüentemente, uma mais valia para as empresas.

A todos os meus amigos e colegas da Auditene, de forma particular ao Sr. Eng.º Vítor Naldinho, que com os seus esforços e profissionalismo me apoiaram e possibilitaram que me dedicasse a esta dissertação mesmo quando as solicitações profissionais eram mais exigentes.

À Filipa, minha esposa, por estar sempre disponível para me escutar, por me compreender, por encontrar sempre motivação, por realizar todas aquelas pequenas coisas que me permitiam estar apenas dedicado à dissertação.

Aos meus pais, família e amigos, pelo apoio incondicional que sempre me têm prestado, pela amizade, pelos conselhos e pela segurança de vida que me transmitem.

Índice

Resumo	1
Abstract.....	3
Agradecimentos	5
Índice	7
Lista de Figuras	9
Lista de Tabelas	11
1. Introdução.....	13
1.1 <i>A gestão de projectos</i>	13
1.2 <i>Definições</i>	23
1.3 <i>Estrutura da dissertação</i>	24
2. Problemas, modelos e algoritmos.....	27
2.1 <i>Introdução</i>	27
2.2 <i>Tipos de problemas</i>	28
2.3 <i>Algoritmos de resolução exacta para o MRCPSP</i>	35
2.3 <i>Métodos heurísticos para o MRCPSP</i>	37
2.4 <i>Conclusões</i>	40
3. Limites inferiores: novas decomposições de Dantzig-Wolfe	43
3.1 <i>Introdução</i>	43
3.2 <i>O princípio da decomposição de Dantzig-Wolfe</i>	44
3.3 <i>Modelo de Talbot</i>	48
3.4 <i>Primeira decomposição</i>	51
3.5 <i>Segunda decomposição</i>	53
3.6 <i>Terceira decomposição</i>	56
3.7 <i>Conclusões</i>	56
4. Limites superiores: uma nova heurística para o MRCPSP	59
4.1 <i>Introdução</i>	59
4.2 <i>Heurística de Boctor</i>	62

4.3 Extensão da heurística de Boctor	71
4.4 Implementação.....	86
4.5 Conclusões.....	87
5. Experiências computacionais	89
5.1 Introdução	89
5.2 Limites inferiores para o MRCPSP	91
5.3 Limites superiores para o MRCPSP.....	100
5.4 Conclusões.....	109
6. Conclusões.....	113
Referências	115

Lista de Figuras

Figura 1.1 – Grafo AON.....	17
Figura 1.2 – Relação de precedência.....	20
Figura 3.1 - Algoritmo de geração de colunas.....	46
Figura 3.2 - Estrutura angular em blocos	48
Figura 3.3 - Estrutura do modelo de Talbot	52
Figura 3.4 - Estrutura da primeira decomposição.....	54
Figura 3.5 - Estrutura da segunda decomposição	55
Figura 3.6 - Estrutura da terceira decomposição	57
Figura 4.1 – Grafo com 8 actividades (exemplo 4.1).....	66
Figura 4.2 – Grafo para $t = 0$ (exemplo 4.1).....	67
Figura 4.3 – Escalonamento parcial para $t = 0$ (exemplo 4.1).....	67
Figura 4.4 – Escalonamento parcial para $t = 2$ (exemplo 4.1).....	68
Figura 4.5 – Escalonamento parcial para $t = 3$ (exemplo 4.1).....	68
Figura 4.6 – Grafo para $t = 6$ (exemplo 4.1).....	69
Figura 4.7 – Escalonamento parcial para $t = 6$ (exemplo 4.1).....	69
Figura 4.8 – Escalonamento parcial para $t = 9$ (exemplo 4.1).....	70
Figura 4.9 – Escalonamento final (exemplo 4.1).....	70
Figura 4.10 – Grafo com 8 actividades (exemplo 4.2).....	75
Figura 4.11 – Grafo com valores de recursos mínimos e quantidade agregada q_i	77
Figura 4.12 – Grafo para $t = 0$ (exemplo 4.2).....	78
Figura 4.13 – Escalonamento parcial para $t = 0$ (exemplo 4.2).....	80
Figura 4.14 – Escalonamento parcial para $t = 2$ (exemplo 4.2).....	81
Figura 4.15 – Escalonamento parcial para $t = 3$ (exemplo 4.2).....	82
Figura 4.16 – Grafo para $t = 6$ (exemplo 4.2).....	83
Figura 4.17 – Escalonamento parcial para $t = 6$ (exemplo 4.2).....	84
Figura 4. 18 – Escalonamento parcial para $t = 7$ (exemplo 4.2).....	85
Figura 4. 19 – Escalonamento final (exemplo 4.2).....	86

Lista de Tabelas

Tabela 1.1 – Uma instância do MRCPSP	18
Tabela 1.2 – Relações de precedência (ver Figura 1.1).....	20
Tabela 4.1 – Instância com 8 actividades, 2 modos/actividade, 1 recurso renovável	66
Tabela 4.2 – Instância com 8 actividades, 2 modos/actividade, 1 recurso renovável e 2 recursos não renováveis.....	75
Tabela 4.3 – Níveis mínimos de recursos não renováveis gastos/actividade	76
Tabela 4.4 – Valores mínimos de recursos NR_1 e NR_2 consumidos/actividade	76
Tabela 5.1 – Limites inferiores $j10$	92
Tabela 5.2 – Limites inferiores $j12$	92
Tabela 5.3 – Limites inferiores $j14$	93
Tabela 5.4 – Limites inferiores $j16$	93
Tabela 5.5 – Limites inferiores $j18$	93
Tabela 5.6 – Limites inferiores $j20$	94
Tabela 5.7 – Limites inferiores $j30$	94
Tabela 5.8 – Limites inferiores $m1$	95
Tabela 5.9 – Limites inferiores $m2$	95
Tabela 5.10 – Limites inferiores $m3$	95
Tabela 5.11 – Limites inferiores $m4$	96
Tabela 5.12 – Limites inferiores $m5$	96
Tabela 5.13 – Limites inferiores $r1$	97
Tabela 5.14 – Limites inferiores $r2$	97
Tabela 5.15 – Limites inferiores $r3$	97
Tabela 5.16 – Limites inferiores $r4$	98
Tabela 5.17 – Limites inferiores $r5$	98
Tabela 5.18 – Limites inferiores $n0$	99
Tabela 5.19 – Limites inferiores $n1$	99
Tabela 5.20 – Limites inferiores $n2$	99
Tabela 5.21 – Limites inferiores $n3$	100

Tabela 5.22 – Limites superiores $j10$	101
Tabela 5.23 – Limites superiores $j12$	101
Tabela 5.24 – Limites superiores $j14$	101
Tabela 5.25 – Limites superiores $j16$	102
Tabela 5.26 – Limites superiores $j18$	102
Tabela 5.27 – Limites superiores $j20$	102
Tabela 5.28 – Limites superiores $j30$	103
Tabela 5.29 – Limites superiores $m1$	103
Tabela 5.30 – Limites superiores $m2$	104
Tabela 5.31 – Limites superiores $m3$	104
Tabela 5.32 – Limites superiores $m4$	104
Tabela 5.33 – Limites superiores $m5$	105
Tabela 5.34 – Limites superiores $r1$	106
Tabela 5.35 – Limites superiores $r2$	106
Tabela 5.36 – Limites superiores $r3$	106
Tabela 5.37 – Limites superiores $r4$	107
Tabela 5.38 – Limites superiores $r5$	107
Tabela 5.39 – Limites superiores $n0$	108
Tabela 5.40 – Limites superiores $n1$	108
Tabela 5.41 – Limites superiores $n2$	108
Tabela 5.42 – Limites superiores $n3$	109

Capítulo 1

Introdução

1.1 A gestão de projectos

A “*Gestão de Projectos*” tem evoluído ao longo dos anos. Desde tempos remotos que os homens realizam tarefas ou actividades que primeiramente idealizaram, conceberam e planearam, sejam elas de implementação simples ou complexa, como acontece por exemplo no caso da construção de pontes ou da montagem de aviões.

De facto a “*Gestão de Projectos*” esteve sempre inerente à vivência humana, mas a consciencialização e reconhecimento da sua importância, teve o seu grande apogeu na segunda metade do século XX, no período após a segunda guerra mundial. As fortes mudanças macroeconómicas operadas nos modelos de desenvolvimento das nações ocidentais, no sentido de permitirem mercados económicos abertos, caracterizados por forte incidência concorrencial, conduziram naturalmente a que as empresas procurassem dotar-se de meios, ferramentas e métodos que lhes permitissem ser mais produtivas e eficientes, ou seja, que lhes acrescentassem valor e dessa forma lhes garantissem sucesso.

Assim, se durante um largo período da história, a natureza do conceito de “gestão de projectos” estava fundamentalmente no domínio da arte, dependendo fortemente da

“vocação” do gestor para a acto de gerir, com a crescente pesquisa científica e o evoluir dos conhecimentos, a natureza do conceito situa-se actualmente numa simbiose arte-ciência, onde a natureza humana do gestor continua a ser um factor preponderante para o sucesso, mas servindo-se este cada vez mais de informação produzida e disponibilizada através de ferramentas e modelos matemáticos que o auxiliam na tomada das melhores decisões.

A “*Gestão de Projectos*” tem ainda uma natureza intrínseca de mudança, que se desenvolve em várias etapas até à concretização dessa mesma mudança, desde a concepção de uma ideia, passando pelo estabelecimento de um plano de execução de um conjunto de tarefas segundo uma determinada ordem e utilizando um conjunto de recursos disponíveis até à implementação e acompanhamento desse mesmo plano.

Tavares [50] define a “*Gestão de Projectos*” como sendo “o processo de conceber, preparar, organizar, dirigir e controlar as transformações necessárias para que se atinjam, a partir de um estado inicial S, os objectivos, definindo um novo estado S’, da forma mais conveniente”.

Numa abordagem considerada mais tradicional, a “*Gestão de Projectos*” tem por base o controlo de três variáveis fundamentais, que podem ser entendidas como três vértices de um triângulo, estando intimamente relacionadas. Essas variáveis são o *tempo*, o *custo* e o *âmbito/objectivo*.

O relacionamento existente entre as três variáveis é facilmente verificado, pois em regra, quanto mais alargado for o âmbito do projecto, mais tempo ele levará a ser implementado e os custos tenderão a ser maiores. Numa outra perspectiva, se se quiser implementar num período mais curto um determinado projecto, isso envolverá mais recursos e portanto resultará num custo maior. Por outro lado será necessário garantir que existem esses mesmos recursos, ou que existam condições para que a sua aplicação se traduza realmente numa diminuição do tempo de realização. Em suma, em cada projecto, existe sempre uma vertente financeira (custo), uma vertente organizacional (recursos humanos, materiais, espaço físico e temporal, entre outros) e uma vertente de controlo que garanta que os objectivos do projecto serão atingidos.

Numa outra abordagem, mais sistemática, o “Project Management Institute - PMI”, fundado em 1969 nos Estados Unidos para normalizar as técnicas e ferramentas a utilizar na gestão de projectos, enumera na sua publicação “A Guide to the Project Management Body of Knowledge (PMBOK) - 3ª Edition [40], cinco fases num projecto:

1. Iniciação
2. Planeamento
3. Execução
4. Monitorização e controlo
5. Encerramento

Associadas a estas fases, define ainda nove áreas de conhecimento que devem ser geridas no decurso do projecto:

- a. Integração do projecto
- b. Âmbito do projecto
- c. Tempo do projecto
- d. Custos do projecto
- e. Qualidade do projecto
- f. Recursos humanos do projecto
- g. Comunicações do projecto
- h. Riscos do projecto
- i. Aquisições do projecto

Claro que nem todos os projectos terão necessariamente que comportar todas as áreas acima indicadas, cabendo ao gestor do projecto adaptar o modelo ao seu projecto em concreto, de forma a minimizar o risco associado e a maximizar as garantias de cumprimento dos objectivos definidos.

Como já foi referido, a evolução dos conhecimentos nesta área tem conduzido ao desenvolvimento e aperfeiçoamento de diversas ferramentas que têm por função auxiliar o gestor do projecto na condução e controlo do mesmo, mas nunca substituí-lo.

Os primeiros modelos de auxílio utilizados foram o método “PERT” e o “Método do Caminho Crítico”. Com a generalização da utilização de computadores, e consequente desenvolvimento da capacidade de cálculo, surgiram diversos softwares que se tornaram ferramentas importantes no auxílio à gestão de projectos. De entre eles destaca-se uma aplicação: o Microsoft Project, amplamente utilizado a nível mundial.

O recurso a suportes informáticos permitiu melhorar substancialmente os modelos usados para representar determinada realidade, aproximando-os da mesma, bem como alargar a várias áreas o tipo de problemas estudados. Assim, a investigação científica e as aplicações práticas no domínio da Investigação Operacional, estendem-se por áreas de estudo muito vastas e diversificadas.

Uma das áreas mais estudadas é a dos problemas de escalonamento e a aplicabilidade deste tipo de problemas é muito alargado, indo desde o sector industrial ao sector dos serviços. Exemplos frequentes deste tipo de problemas são: o planeamento de equipamentos produtivos na indústria, conhecido na terminologia anglo-saxónica como “production scheduling”, o planeamento de obras na construção civil ou a afectação de recursos a um projecto, citando apenas alguns.

Brucker *et al.* [10] referem-se ainda à crescente importância deste tipo de problemas na actualidade, em virtude da generalizada tendência de adopção de metodologias “LEAN” e “MAKE TO ORDER” pelas empresas.

Os problemas de escalonamento de projectos são tratados no âmbito da Programação Inteira (PI) ou da Optimização Combinatória (OC). A sua resolução exacta ou mesmo a resolução aproximada continua a ser um desafio, dada a sua inerente complexidade. A maior parte desses problemas é de facto classificada como NP-difícil.

Genericamente, um problema de escalonamento de projectos é caracterizado pela existência de um conjunto de actividades, que formam o projecto, que devem ser realizadas atendendo a uma série de relações de precedência entre elas, com um objectivo operacional bem específico e determinado, como por exemplo a minimização da duração total do projecto ou a minimização de custos, e onde os recursos disponíveis para a realização das actividades são limitados.

O projecto poderá ser representado graficamente através de um grafo, onde os nodos representam as actividades e os arcos as relações de precedência que existem entre as diversas actividades. Este tipo de grafos é designado de AON (Activities On Nodes).

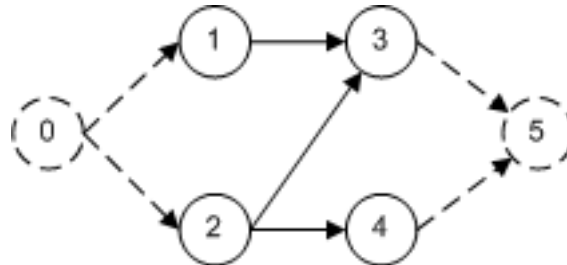


Figura 1.1 – Grafo AON

O grafo diz-se acíclico porque não existe a possibilidade de uma vez realizada uma actividade se voltar a executar essa mesma actividade. No exemplo acima, os nodos 0 e 5 representam actividades fictícias que têm um tempo de execução nulo e que não consomem recursos. A actividade 0 representa o início do projecto e precede todas aquelas actividades do projecto que não têm nenhuma actividade predecessora. A actividade 5 representa o final do projecto. É sucessora de todas as actividades que não são predecessoras de nenhuma outra actividade. Em qualquer algoritmo de resolução, e considerando os dados do exemplo acima, o facto de só faltar a actividade 5 para escalonar é um sinal de que todas as outras actividades foram devidamente escalonadas.

Esta dissertação abordará este tipo de problemas, mais concretamente, a forma mais geral do mesmo e que incide sobre a possibilidade de haver várias formas, ou modos, de executar tais actividades, sendo este problema designado na literatura anglo-saxónica por *Multi-Mode Resource Constrained Project Scheduling Problem* (MRCPSp).

Este tipo de problemas foi introduzido pela primeira vez em 1977 por Elmaghraby [21].

Na Figura 1.1 e Tabela 1.1, apresentamos uma instância do MRCPSp constituído por 4 actividades (a actividade 0 e a actividade 5 são fictícias e representam o início e a conclusão do projecto respectivamente, pelo que ambas apresentam duração igual a 0 e não consomem recursos), cada uma com dois modos possíveis de realização e onde se

indicam os respectivos consumos de recursos por modo. Neste caso existem dois tipos de recursos renováveis e dois de recursos não renováveis. Existem ainda 4 unidades do recurso renovável R_1 , 6 unidades do recurso renovável R_2 , 10 unidades de recurso não renovável N_1 e 8 unidades de recurso não renovável N_2 .

Actividade	Modo	Duração	Recursos Renováveis		Recursos Não Renováveis	
			R_1	R_2	N_1	N_2
0	0	0	0	0	0	0
1	1	2	2	3	2	2
	2	4	1	2	1	0
2	1	1	3	2	4	3
	2	3	1	1	2	1
3	1	2	3	2	3	3
	2	5	2	1	1	1
4	1	2	5	0	0	3
	2	3	2	2	0	1
5	0	0	0	0	0	0

Tabela 1.1 – Uma instância do MRCPSP

As actividades, muitas vezes designadas por “tarefas”, formam no seu conjunto o projecto. Caracterizam-se por terem associadas uma “duração” e um “custo”, podendo a “duração” ser de natureza discreta ou contínua no tempo, determinística ou estocástica conforme é ou não conhecido com certeza o tempo necessário à execução da actividade, respectivamente.

Relativamente ao “custo” este pode ser explicitado em termos monetários, ou ainda em termos de necessidades de quantidades de recursos alocados, verificando-se normalmente a existência de uma relação que traduz uma realidade esperada nos problemas de escalonamento. Assim, a uma duração menor estará sempre associado um “custo” maior.

Uma outra característica importante associada às actividades reside no facto de se admitir ou não a interrupção de actividades, ou seja, se uma actividade uma vez iniciada

pode ser interrompida para se executar uma outra actividade, sendo retomada mais tarde no mesmo ponto de execução em que se encontrava.

Uma actividade é também caracterizada por dois instantes de tempo de grande importância: o instante de tempo mais cedo e mais tarde em que pode ser terminada essa actividade. Se os dois valores forem diferentes, dizemos que a actividade possui uma folga. Essa noção de folga está directamente ligada ao conceito de caminho crítico. Voltaremos a ele mais adiante nesta secção.

Em alguns contextos, como é o caso por exemplo da indústria metalúrgica, existem períodos de tempo entre o arranque de duas actividades sucedâneas que deve ser verificado. Na terminologia anglo-saxónica, fala-se de *time lags*. Esses períodos garantem por exemplo que o metal fundido numa determinada actividade tenha tempo de arrefecer antes de ser manuseado em qualquer outra actividade.

Um problema próximo do MRCPS é o problema de compromisso tempo-custo (PCTC, ou *discrete time-cost tradeoff problem*, na terminologia anglo-saxónica) aplica-se a projectos nos quais a aplicação de mais capital permite reduzir o tempo de execução de determinadas actividades.

Uma actividade é realizada segundo um de vários modos possíveis. Para cada modo, existe um par “custo-duração” que determina o custo a pagar por realizar uma actividade com uma determinada duração. Essa função custo-duração é de natureza discreta. Tipicamente, para este tipo de problemas, existem duas funções objectivas: minimização do custo do projecto cumprindo uma data limite (*deadline problem*), e minimização da duração do projecto não excedendo um limite de capital (*budget problem*). Na literatura, o primeiro problema tem recebido mais atenção por parte da comunidade científica.

Na versão mais geral, e mais próxima da realidade dos projectos, uma actividade poderá ser executada de diferentes modos. Um modo é caracterizado por um determinado tempo de execução que depende do nível de recursos empregue, sendo que quanto mais recursos forem aplicados, mais rápida será a execução da actividade. Problemas deste

tipo são designados por problemas multi-modos. É essa classe de problemas que abordamos nesta dissertação.

Todas as actividades que formam um projecto terão de ser executadas para que o projecto possa ser concluído, mas a garantia da realização de todas as actividades só por si, não é condição suficiente para se atingir o objectivo específico que inicialmente foi traçado. Elas devem ser realizadas respeitando uma série de restrições à sua execução, como por exemplo a ordem pela qual serão executadas ou os recursos disponíveis para a sua execução.

Assim, a sequência lógica de realização das actividades traduz-se pelo estabelecimento das *relações de precedência*. Quando, por exemplo, o início de uma actividade j depende da conclusão de uma actividade i , diz-se que a actividade i precede a actividade j . Nos grafos de actividade nos nodos (activity on node – AON – na terminologia anglo-saxónica), cada uma destas relações dá origem a um arco.



Figura 1.2 – Relação de precedência

Actividade	Actividades Sucessoras
1	3
2	3, 4

Tabela 1.2 – Relações de precedência (ver Figura 1.1)

Os recursos utilizados na execução das actividades podem ser classificados atendendo à categoria, ao seu tipo ou ao valor associado. Assim, os recursos dividem-se em três categorias:

- *Recursos renováveis*: limitados apenas num dado período; após a sua utilização num dado período do horizonte de planeamento para a execução de uma determinada actividade, esses recursos voltam a estar disponíveis

para nova utilização na mesma ou noutra actividade. Exemplos de recursos renováveis são as máquinas ou os recursos humanos;

- *Recursos não renováveis*: limitados em todo o horizonte de planeamento do projecto; existe uma quantidade limitada de um determinado recurso que uma vez consumida não volta a estar disponível para outra utilização. O orçamento disponível para a execução de um projecto é um exemplo de recurso não renovável;
- *Recursos duplamente restritos*: recursos limitados num dado período e ao mesmo tempo no horizonte global de planeamento do projecto. Esses recursos podem sempre ser desdobrados em pelo menos um recurso renovável e num recurso não renovável. Um exemplo deste tipo de recursos poderá ser o dinheiro disponível para a realização global do projecto e que em cada período apenas está disponível um determinado montante.

O tipo de problemas está normalmente relacionado com o tipo de função objectivo que se pretende otimizar e com o tipo de restrições consideradas. Um projecto bem definido terá sempre um, ou mais, objectivos específicos, definidos no modelo através de um determinado critério que se pretende otimizar.

Os critérios podem ser muito diversos, mas os que têm sido mais estudados são o da minimização do tempo de conclusão do projecto, referido na literatura anglo-saxónica como *makespan minimization problem* e o da maximização do valor actual líquido (VAL). Existem muitos outros critérios de avaliação, entre eles a minimização do custo total ou a maximização da qualidade. Para este último, são significativos os trabalhos desenvolvidos por Icmeli e Rom [30] que, no decorrer de um estudo desenvolvido nos Estados Unidos, constataram que o objectivo mais importante para os gestores de projectos era de facto a maximização da qualidade do projecto.

Numa fase inicial o desenvolvimento de modelos incidiu fortemente sobre o escalonamento de equipamentos de produção na indústria. Os responsáveis pela gestão da produção tinham a necessidade de otimizar os tempos de processamento de cada um dos equipamentos disponíveis, e para tal foram desenvolvidos modelos que

otimizavam a utilização das máquinas, garantindo assim um melhor aproveitamento dos recursos. Este tipo de problema é conhecido como *job shop scheduling* ou ainda *machine scheduling*. Nos últimos anos, tem-se assistido também a um crescente interesse por parte da comunidade científica pelos problemas associados à gestão de projectos.

Antes de introduzirmos algumas definições e notação relacionada com o MRCPSP, apresentamos primeiro alguns conceitos comuns aos problemas de gestão de projectos. Destacamos em particular as noções de caminho crítico, e instante de tempo mais cedo e mais tarde em que pode ser terminada uma actividade.

O caminho crítico num projecto representa o conjunto ou sequência de actividades que implicarão atrasos na conclusão do projecto se não forem iniciadas num determinado instante de tempo. As actividades pertencentes ao caminho crítico não admitem atrasos na sua realização, ou seja, têm folga igual a 0. A folga calcula-se fazendo a diferença entre o instante de tempo mais tardio (LF - Latest Finish) e o instante de tempo mais cedo (EF – Early Finish) em que uma actividade pode ser terminada. A folga é uma medida da rigidez com que o escalonamento de uma actividade pode ser feito. As actividades pertencentes ao caminho crítico, com folga igual a 0, têm de ser escalonadas em determinados instantes, sob pena de se prolongar a conclusão do projecto, ao passo que as actividades que não pertencem ao caminho crítico e portanto admitem folga, permitem alguma liberdade na escolha do momento para serem iniciadas, sem que isso penalize a duração total do projecto.

O cálculo de EF para uma determinada actividade é feito percorrendo o grafo no sentido das actividades predecessoras para as actividades sucessoras (progressivo) e traduz o momento mais cedo em que a actividade poderá ser terminada. O cálculo tem por base a duração de realização das actividades predecessoras. Quando uma actividade tiver duas ou mais actividades predecessoras, o valor de EF será igual ao maior tempo de conclusão entre todas essas actividades sucessoras adicionado à sua própria duração (no modo mais rápido caso se trate de um problema MRCPSP). Com efeito, só após todas terem sido realizadas todas as actividades predecessoras é que será possível a actividade vir a ser iniciada.

O cálculo de LF é feito após se calcular os EF de todas as actividades do projecto, e depois de conhecer portanto o instante de tempo mais cedo em que poderá ser concluído o projecto. Como ponto de partida, considera-se que para a última actividade, o valor de LF é igual ao EF. De seguida, percorre-se o grafo, no sentido das actividades sucessoras para as predecessoras (regressivo), determinando o valor de LF para cada uma das actividades (a partir dos modos mais rápidos se se tratar do MRCPSP).

1.2 Definições

De modo a clarificar a apresentação dos próximos capítulos, resumimos de seguida os elementos característicos do MRCPSP, e introduzimos ainda alguns elementos de notação usados ao longo da dissertação.

O MRCPSP estudado nesta dissertação é um problema de optimização em qual se procura determinar o instante de tempo em que as actividades de um projecto devem ser iniciadas. Entre as actividades, existem relações de precedência que determinam quais as actividades que devem ter sido realizadas para se poder iniciar uma determinada actividade. Para realizar uma actividade, é necessário consumir recursos. Os recursos são escassos, e podem ser de dois tipos: renováveis e não renováveis (os recursos duplamente restritos são divididos na sua componente renovável e não renovável).

Existem diversas formas de se realizar uma actividade. A essas “formas” dá-se o nome de *modo de execução*, ou simplesmente modo. Um modo é completamente definido por uma duração da actividade, e um conjunto de valores de consumos de recursos. Em qualquer dos casos, não é permitido interromper uma actividade que tenha já sido iniciada.

O objectivo do problema é de encontrar o escalonamento óptimo das actividades de forma a concluir o projecto no mais curto espaço de tempo.

Ao longo da dissertação, as notações poderão eventualmente variar atendendo ao modelo específico que estaremos a descrever. No entanto, alguns elementos mantêm-se relativamente constantes. Para evitarmos repetições, enunciamo-los de seguida:

- M_i : Conjunto de modos possíveis para a actividade i ;
- F_i : Folga associada à actividade i (diferença entre os instantes de tempo mais tardio e mais cedo em que uma actividade poderá terminar; normalmente, é calculada tendo em conta os modos mais rápidos);
- $d_{ij(i)}$: Duração da actividade i quando usado o modo $j(i)$;
- d_{i1} : Duração da actividade i quando usado o modo 1 (para cada actividade, assumimos que os modos estão ordenados do mais rápido ao mais lento).

1.3 Estrutura da dissertação

A dissertação é composta por 6 capítulos incluindo este. Uma breve apresentação de cada capítulo é feita de seguida.

No Capítulo 2, apresentamos uma revisão de literatura relativa aos problemas de sequenciamento de projectos, e em particular ao MRCPSP. Exploramos os vários tipos de problemas dessa área, e enunciamos os principais métodos de resolução descritos na literatura.

No Capítulo 3 são introduzidos novos modelos de Programação Inteira para o MRCPSP baseados em decomposições de Dantzig-Wolfe de um modelo original e compacto. Esses modelos são usados para derivar limites inferiores de boa qualidade para o problema.

O Capítulo 4 é dedicado à pesquisa de limites superiores para o MRCPSP. Descrevemos uma heurística construtiva apresentada na literatura, e introduzimos uma extensão a essa heurística que permite resolver instâncias em que se considera a existência de recursos não renováveis

No Capítulo 5 são apresentados e analisados os testes computacionais que foram realizados para a validação das abordagens propostas. Os testes tiveram como suporte instâncias disponíveis publicamente.

Finalmente, no Capítulo 6, terminamos com algumas conclusões sobre o trabalho desenvolvido ao longo da dissertação.

Capítulo 2

Problemas, modelos e algoritmos

2.1 Introdução

Os problemas de sequenciamento de projectos com recursos escassos (*Resource Constrained Project Scheduling Problem*, ou RCPSP, na terminologia anglo-saxónica) são uma realidade no dia-a-dia dos gestores. Existe de facto uma grande diversidade de problemas que podem ser considerados como pertencendo a esta área, sendo que os métodos utilizados para o tratamento dos mesmos têm evoluído de forma muito rápida nos últimos anos, aproveitando o crescente desenvolvimento das tecnologias de informação e comunicação.

As diversas realidades encontradas pelas empresas e gestores conduzem a problemas muito diferenciados, os quais têm sido abordados de diferentes formas por parte dos membros da comunidade científica. Na área de escalonamento, poder-se-ão procurar diferentes objectivos como por exemplo a gestão dos recursos humanos, a gestão de matérias-primas, a minimização de custos de um determinado projecto ou ainda a maximização de lucros, podendo esses objectivos ser combinados com uma série de outros factores. Poderá haver ou não restrições na utilização de recursos, que por sua vez poderão ser de natureza diversa. Poderão existir diversos modos de execução das actividades ou apenas um. Cada uma dessas opções pode ser considerada em conjunto, dando origem como já dissemos a uma grande variedade de problemas.

O RCPSP consiste em definir o momento em que irão ser realizadas as actividades que constituem o projecto, assim como os recursos a associar à execução de cada uma

dessas actividades. Os recursos são naturalmente escassos (caso contrário, deixariam de fazer parte do problema). Têm de ser alocados às actividades de modo a permitir a conclusão do projecto, tendo em conta um determinado objectivo de optimização.

Como já referimos, os problemas do tipo RCPSP têm recebido uma atenção crescente por parte da comunidade científica em virtude das solicitações e aplicabilidades práticas nas realidades efectivas das empresas. Contudo, até a alguns anos atrás, o trabalho de investigação era desenvolvido de forma mais individualizada, sem que houvesse grande partilha de experiências e informação entre investigadores. O resultado foi uma notação e linguagem dispar que dificultava a comparação entre abordagens.

Com a generalização da partilha de informação através da Internet, e numa tentativa de normalizar conceitos usados nesta área, Brucker *et al.* [10] propuseram em 1999 uma notação genérica para problemas do tipo RCPSP, mas que foi no entanto contestada em alguns pontos por Herroelen *et al.* [29] poucos anos depois. Apesar de tudo, vários artigos publicados desde essa primeira data usam a notação proposta em [10].

Nesta secção, far-se-á uma breve caracterização e análise dos vários problemas e respectivos métodos de resolução que foram propostos na literatura na área dos problemas de escalonamento ou sequenciamento de projectos com recursos escassos (RCPSP), com particular incidência no caso mais geral em que existem vários modos para executar cada uma das actividades do projecto (MRCPSP).

2.2 Tipos de problemas

Os vários tipos de problemas de sequenciamento de projectos com recursos escassos definem-se através de uma combinação de diferentes parâmetros. Um deles, por exemplo, prende-se com a definição das próprias actividades que definem o projecto. Uma actividade poderá ter apenas um único modo de execução (problemas do tipo modo único, ou *single-mode*), ou vários modos de execução (problemas do tipo multi-modo, ou *multi-mode*). Outro parâmetro possível consiste no tipo de relação que existe entre a duração de uma actividade e o nível de recursos aplicados na sua execução.

O problema de sequenciamento poderá ser de natureza estocástica, baseado em informação pouco segura e à qual se associa uma determinada probabilidade de ocorrência, ou de natureza determinística, se toda a informação for perfeitamente conhecida. Os problemas que serão abordados no âmbito desta dissertação pertencem a este último grupo.

No que se refere ao tipo de função objectivo definida para o projecto, e que representa o objectivo geral que se pretende atingir ao resolver um determinado problema, a minimização da duração total do projecto (*makespan minimization*) tem sido o mais estudado até agora. Em problemas com este tipo de função objectivo, procura-se sequenciar as actividades do projecto de modo a garantir que o projecto termine o mais cedo possível. Outros tipos de função objectivo têm também sido usados e estudados.

O objectivo da maximização do valor actual líquido do projecto (*net present value maximization*) foi estudado pela primeira vez por Russell [42]. Icmeli e Rom [30] desenvolveram estudos muito precisos na área da qualidade dos projectos, e constataram através de um inquérito levado a cabo junto de gestores dos Estados Unidos que esse critério era um dos aspectos que os gestores mais valorizavam [31]. Na sequência desses trabalhos, definiram modelos em que a função objectivo consistia na maximização da qualidade do projecto. A minimização do custo global do projecto (*cost minimization*) tem sido outro critério amplamente utilizado que podemos encontrar em diversos artigos da literatura.

No artigo publicado por Brucker *et al.* [10], que constitui uma resenha do que havia sido feito na domínio do RCPSP até à data, os autores apontam também para alguns caminhos futuros de investigação, e apresentam de forma genérica os principais tipos de problemas que se enquadram nesta área. Uma classificação diferente é apresentada em Yang *et al.* [51], embora apenas difere ligeiramente da primeira. A classificação proposta apresenta 6 tipos de problemas do tipo RCPSP:

- os problemas com modos únicos (*single mode*);
- os problemas com vários modos de execução (*multi-mode*);
- os problemas de compromisso tempo-custo (*time-cost trade off*);

- os problemas com intervalos máximos e mínimos entre actividades (*minimum and maximum time lags*);
- os problemas com funções objective não regulares;
- os problemas com durações estocásticas para as actividades.

De seguida, exploramos cada um desses problemas.

Os problemas com um único modo de execução são considerados como sendo os problemas clássicos do tipo RCPSP. Nesses problemas, assume-se que existe apenas uma forma de realizar cada uma das actividades do projecto. Existe assim apenas uma duração possível e um único conjunto de valores que definem os recursos consumidos quando se executa a actividade. Cada um desses valores é perfeitamente conhecido. O problema é por isso do tipo determinístico. Esse problema poderá considerar ou não a possibilidade de interromper a execução de uma actividade, e retomá-la mais tarde. A variante mais estudada desse problema não admite a interrupção das actividades. Admite apenas recursos renováveis, e tem como objectivo a minimização da duração total do projecto. Na notação proposta por Brucker *et al.* [10], esse problema é designado por $PS |Prec| C_{max}$.

As técnicas mais utilizadas na resolução desta classe de problemas baseiam-se em técnicas de enumeração como o método de partição e avaliação sucessivas (*branch-and-bound*). A literatura descreve ainda alguns processos para cálculo de limites inferiores, e um número significativo de procedimentos de resolução heurística.

A alternativa mais óbvia à classe de problemas referida acima é aquela em que existe mais do que um simples modo de execução por actividade. Esses problemas são designados por multi-modo (MRCPSP), e tivemos já oportunidade de nos referir a eles anteriormente dado que são os problemas principais focados nesta dissertação. Os problemas do tipo *single-mode* são um caso especial do MRCPSP. Um resultado interessante acerca desse tipo de problemas foi derivado por Kolisch [33]. Segundo esse autor, o problema que consiste em encontrar uma solução válida para uma instância do MRCPSP com mais de 2 modos de execução e 2 recursos renováveis é um problema NP-completo.

A variante que aqui estudamos é aquela em a duração dos modos é conhecida à partida, estando essa duração directamente dependente do nível de recursos gastos (relação tempo-recursos). Uma relação alternativa poderá existir entre a duração dos modos e o custo dos modos. Essa relação define um problema próximo do MRCPSP que descreveremos mais adiante, e que é designado por problema de compromisso tempo-custo (*time-cost tradeoff problem*).

O objectivo final do MRCPSP consiste em determinar a afectação óptima dos modos às actividades, garantindo que o escalonamento (ou sequenciamento) é feito de tal maneira que as relações de precedência são respeitadas e os níveis de recursos não são excedidos. Na taxonomia definida por Brucker *et al.* [10], o MRCPSP é designado pelas siglas $MPS |Prec| C_{max}$.

Este tipo de problema tem recebido alguma atenção da comunidade científica, devido à sua proximidade com a realidade dos projectos e à sua inerente complexidade. Foram propostas algumas heurísticas para a sua resolução (essencialmente meta-heurísticas). O primeiro modelo matemático de que temos conhecimento foi proposto por Talbot em 1982 [49]. No Capítulo 3 desta dissertação, apresentaremos e discutiremos esse modelo.

Um dos problemas que partilha algumas semelhanças com o MRCPSP é o problema de compromisso tempo-custo (*time-cost tradeoff problem*). Na prática, a duração de um determinado projecto, isto é das actividades que o compõem, pode ser controlada variando o investimento que é feito. Assim, quanto mais dinheiro for gasto em materiais, recursos humanos ou outros elementos que possam considerados como recursos úteis à execução das tarefas, mais depressa se conseguirão realizar as actividades desse projecto e assim acelerar a conclusão do projecto. Podemos pensar na medida financeira associada aos níveis de recursos gastos em cada actividade, e estabelecer assim uma relação entre a duração da actividade e o valor líquido de dinheiro gasto para a sua execução. Claramente, quanto maior for o capital investido mais rápida será a conclusão de uma actividade. A relação inversa também é válida. Menores quantias de capital investido fazem com que as actividades sejam realizadas num maior espaço de tempo. Nos problemas de compromisso tempo-custo, não existe uma referência explícita à existência de recursos. Procura-se antes determinar o melhor sequenciamento das actividades de modo a respeitar uma restrição ao nível do capital

disponível, ou encontrar aquele sequenciamento que minimize o custo de execução do projecto.

Nessa classe de problemas e para cada actividade, definem-se modos de execução que mais não são do que pares “tempo-custo”. Tal como para o MRCPSP, um escalonamento válido define a forma como cada actividade deve ser executada (o seu modo de execução) e o instante de tempo em que essa actividade deve ser iniciada.

Em termos de função objectivo para o problema de compromisso tempo-custo, existem descritas na literatura as duas variantes seguintes:

- o caso em que se procura minimizar o custo total do projecto, considerando-se como sendo fixo o tempo máximo disponível para a sua execução (*deadline problem*);
- o caso em que se procura minimizar a duração global do projecto, considerando-se fixo o montante máximo de capital que pode ser aplicado (*budget problem*).

A primeira variante tem merecido mais atenção por parte da comunidade científica. Dada a importância e semelhanças deste problema com o caso do MRCPSP, analisamos de seguida um modelo específico para este problema.

Akkan *et al.* [3] descrevem um modelo para o problema de compromisso tempo-custo no caso em que a função objectivo consiste na minimização do custo total do projecto. A notação na qual se baseia é apresentada de seguida:

- $V = \{0, 1, \dots, n, n+1\}$: conjunto de actividades (a actividade 0 e $n+1$ são actividades fictícias que representam respectivamente o início e final do projecto);
- $E \subseteq V$: conjunto de actividades entre as quais existe uma relação de precedência;
- c_{jm} : custo de processamento da actividade j quando executada modo m ;

- EC_j : instante de tempo mais cedo em que a actividade j poderá ser terminada;
- LC_j : instante de tempo mais tardio em que a actividade j poderá ser terminada;
- T : prazo para a conclusão do projecto.

As variáveis de decisão desse modelo são de dois tipos. Por um lado, temos as variáveis C_j que representam o instante de tempo em que a actividade j é terminada, e as variáveis binárias x_{jm} que determinam se a actividade j é realizada ou não no modo m . As restrições do modelo garantem que cada actividade é realizada usando apenas um modo de execução, que as relações de precedência são satisfeitas, que o prazo de conclusão do projecto não é excedido e que as actividades do projecto começam a ser escalonadas a partir do instante de tempo $t=0$. O modelo para o problema de compromisso tempo-custo pode ser formulado como segue:

$$\min \sum_{j \in V} \sum_{m \in M_j} c_{jm} x_{jm}$$

sujeito a

$$\sum_{m \in M_j} x_{jm} = 1, \quad j \in V,$$

$$C_j - C_i - \sum_{m \in M_j} d_{jm} x_{jm} \geq 0, \quad (i,j) \in E,$$

$$C_{n+1} \leq T,$$

$$C_0 - \sum_{m \in M_j} d_{0m} x_{0m} \geq 0$$

$$x_{jm} \in \{0,1\}, \quad j \in V, \quad m \in M_j.$$

Este tipo de problema tem recebido grande atenção por parte da comunidade científica. A literatura descreve vários algoritmos exactos e métodos heurísticos que conduzem a boas soluções em tempos de computação razoáveis.

Ao nível dos algoritmos exactos, destacam-se os algoritmos de programação dinâmica combinados com a decomposição da estrutura em rede do projecto. Esta decomposição consiste em transformar sucessivamente a rede que define o projecto, de forma a que esta se assemelhe a uma rede do tipo “série-paralela”. Neste processo de sucessivas transformações, são identificados nodos de duplicação que permitem que se vá aproximando a rede em análise a uma rede do tipo “série-paralela”. Neste domínio, destacam-se os trabalhos de Robinson [41], Bein *et al.* [7], De *et al.* [14], Elmaghraby [22] e Demeulemeester *et al.* [19].

A procura de limites inferiores para este problema não tem sido muito explorada, conhecendo-se apenas um algoritmo introduzido por Akkan [1] e baseado em relaxação lagrangeana. No que se refere aos algoritmos de aproximação, Skutella [45] contribuiu com um método baseado na resolução exacta de relaxações lineares e em arredondamentos. Nesse domínio, são ainda de referir as contribuições de Deineko e Woeginger [17] que derivaram resultados teóricos de aproximação para esta classe de problemas. Ao nível das heurísticas refira-se a introduzida por Akkan [2] e que têm por base um método de pesquisa local.

Uma generalização do problemas de RCPSP consiste em considerar intervalos de tempo máximo e/ou mínimo entre o arranque de cada actividade. Um exemplo clássico que ilustra esse tipo de problemas ocorre na indústria metalúrgica. Nesses contextos o tratamento da matéria-prima (ferro fundido, por exemplo) pode ter de ser atrasado para além de certos limites, sob pena de inviabilizar a execução de uma actividade em que essa matéria-prima faz ofício de *input*.

Para a resolução deste tipo de problemas, destacam-se os trabalhos desenvolvidos por Bartush *et al.* [6], De Reyck [15], De Reyck e Herroelen [16], Schwindt [44] que se baseiam essencialmente em métodos de partição e avaliação. Mais recentes são os trabalhos apresentados por Heilmann [28]. Esse autor propôs uma heurística construtiva para resolver o problema em que recorre ao uso de regras de prioridade.

A literatura descreve ainda algumas abordagens para problemas em que as funções que se pretendem otimizar não têm um comportamento regular. Nos problemas do tipo RCPSP, uma função é dita regular sempre que o custo aumenta quanto mais rápida for a execução de uma actividade. Em [10], os autores analisam em detalhe dois casos: o problema de nivelamento de recursos (*resource leveling problem*) e o problema do valor líquido actual (*net present value problem*).

A última classe de problemas do tipo RCPSP é aquela em que as durações das actividades não são conhecidas completamente à partida. Trata-se dos problemas estocásticos. Na realidade empresarial, e na maioria dos casos, a determinação da duração de cada actividade raramente é um dado bem conhecido à partida (não é determinística). Para essa situação, concorrem vários factores que por não serem bem conhecidos, ou por serem incontrolláveis, conduzem a que as durações de execução de cada actividade tenham de ser estimadas com base em funções de probabilidades de ocorrência. Da mesma forma, os consumos de recursos são dados por estimativas.

Se esse tipo de problemas é na verdade mais representativo da realidade que se pretende modelar, por outro lado a sua complexidade é bem maior comparativamente com os modelos determinísticos, conduzindo quase sempre a tempos de processamento bastante elevados.

2.3 Algoritmos de resolução exacta para o MRCPSP

Um algoritmo dito exacto procura encontrar a solução óptima para um problema de optimização. No domínio dos problemas do tipo MRCPSP, os algoritmos exactos que são referenciados são essencialmente algoritmos de pesquisa e enumeração baseados no método de partição e avaliação sucessivas (*branch-and-bound*). Entre eles, destacamos os seguintes:

- algoritmos baseados em árvores de precedências (*precedence tree*);

- algoritmos baseados no conceito de alternativas de modo e atraso (*mode and delay alternatives*);
- algoritmos baseados no conceito de alternativas de modo e extensão (*mode and extension alternatives*).

Estes três tipos de algoritmos têm em comum o facto de construírem escalonamentos parciais, apresentando como uma diferença bem vincada o facto de o primeiro algoritmo fazer uma análise actividade a actividade, enquanto que os outros dois, fazem uma análise baseada em conjuntos de actividades.

Um dos primeiros algoritmos exactos para o MRCPSP foi introduzido por Patterson *et al.* [39] e usa o conceito que foi entretanto designado por *árvore de precedências*. Outras versões melhoradas foram mais tarde introduzidas por Sprecher [46], e posteriormente por Sprecher and Drexl [47].

Esse método consiste em construir ao longo de uma árvore de pesquisa diferentes escalonamentos, escolhendo nos ramos da árvore pares modo-actividade que possam ser escalonados assim como instante de tempo mais cedo em que esse escalonamento pode ser realizado. Um ramo da árvore termina quando se atinge a actividade fictícia que representa o fim do projecto. Fazendo *backtracking* ao longo da árvore, constroem-se escalonamentos alternativos que podem ser comparados entre eles com base no valor obtido para a função objectivo. Um caminho que leva da raiz da árvore até a um nodo do nível mais baixo corresponde a uma sequência válida de actividades, isto é, que respeita as relações de precedência.

Os algoritmos baseados no conceito de alternativas de modo e atraso foram propostos originalmente por Sprecher *et al.* [48]. Essas abordagens consistem na realidade em adaptações de outros métodos propostos na literatura para resolver o RCPS ([11], [18]).

Esses métodos são igualmente baseados na construção de uma árvore de pesquisa na qual cada nodo está agora associado a um determinado instante de tempo. Em cada nodo, escalonam-se temporariamente actividades cujas antecessoras tenham já sido concluídas. Esse escalonamento é feito escolhendo entre vários pares modos-actividade

possíveis. Se alguma restrição de recurso tiver sido violada com esse escalonamento, calculam-se conjuntos de actividades (entre aquelas que acabaram de ser escalonadas) a que se dá o nome de alternativas de atraso. Na prática, esses conjuntos representam as actividades que podem tornar válido o escalonamento parcial realizado no nodo da árvore se forem todas atrasadas. Entre os vários conjuntos, escolhe-se um dos conjuntos *mínimos*, isto é um conjunto que é tal que se for removida uma qualquer actividade desse conjunto, o atraso das actividades restantes não será suficiente para garantir a validade do escalonamento parcial.

O terceiro tipo de algoritmo baseado em alternativas de modo e extensão foi proposto por Hartmann and Drexel em [26]. Esse algoritmo é muito parecido com o anterior. A diferença reside no facto de não ser permitido efectuar escalonamentos parciais nos nodos da árvore de pesquisa que viole restrições de recursos. O conceito de extensão referido no nome do algoritmo consiste num conjunto de pares modos-actividades escalonáveis num determinado nodo, e que estendem o escalonamento parcial do nodo no nível superior sem que nenhuma restrição ligadas aos recursos seja violada.

De notar que não encontramos na literatura referências onde fossem estudadas decomposições de Dantzig-Wolfe para o MRCPSP. Os algoritmos exactos para esse problema não diferem substancialmente uns dos outros. Como já referimos, o método de partição e avaliação sucessivas foi amplamente usada para esse propósito.

2.3 Métodos heurísticos para o MRCPSP

Os resultados de investigação no campo dos métodos heurísticos para o MRCPSP são bastante mais numerosos do que aqueles relacionados com os métodos exactos. Essa realidade justifica-se pela complexidade inerente ao MRCPSP, mas também pelo facto de muitas vezes as heurísticas conduzirem em tempos computacionais aceitáveis a soluções aproximadas de boa qualidade, que podem mesmo em alguns casos coincidir com a solução óptima.

As abordagens heurísticas propostas para o MRCPSP podem naturalmente ser subdivididas em dois grandes grupos:

- as heurísticas propriamente ditas em que os escalonamentos são feitos geralmente usando um conjunto de regras de prioridade (em [35], essas heurísticas são designadas por métodos *X-pass*);
- as meta-heurísticas entre as quais se destacam os métodos de pesquisa tabu, os métodos de pesquisa por arrefecimento simulado ou ainda os algoritmos genéticos.

Nas heurísticas, a construção do escalonamento é feita usualmente de forma parcial. Consistem num processo de geração de escalonamentos para uma ou mais actividades, e em regras de prioridade para a selecção das actividades que irão ser escalonadas a seguir. Para gerar os escalonamentos, é normal distinguir entre duas abordagens diferentes. A primeira baseia-se nas actividades, e é designada por escalonamento em série. Em cada passo da heurística, as actividades escalonáveis são aquelas cujos predecessores já terminaram. Realizam escalonamentos óptimos se o problema não tiver recursos associados. A segunda abordagem considera antes os instantes de tempo, e é designada por escalonamento em paralelo. As actividades são escalonadas num instante de tempo t se as suas predecessoras tiverem já acabado, e se houver recursos para tal.

As heurísticas do tipo *X-pass*, em que o escalonamento das actividades é feito com base em regras de prioridade, baseiam-se em passagens do início para o fim (ou vice-versa) no grafo de actividades, podendo ser feita uma única passagem ou várias passagens.

As regras de prioridade podem ser definidas de muitas formas e existe na literatura actual uma grande diversidade de regras referenciadas. Estas são utilizadas para classificar as actividades e os respectivos modos, no conjunto de actividades escalonáveis, de forma a se poder definir uma ordem de prioridade para o escalonamento no nível ou instante de tempo em análise.

Boctor [8] desenvolveu um estudo envolvendo 7 regras de prioridade para as actividades e 3 regras para os modos, numa combinação de 21 tipos de procedimentos diferentes para efectuar o escalonamento, sendo este feito actividade a actividade. Mais tarde, o mesmo autor apresenta um outro estudo [9] onde apresenta uma heurística em que o escalonamento das actividades não é feito de forma individual, mas considerando

antes conjuntos de actividades. No Capítulo 4 desta dissertação, voltaremos em detalhe nos trabalhos desse autor.

Heilmann [27] apresentou um algoritmo para o caso mais geral do MRCPSP (para além de se verificar a existência de tempos mínimos entre os inícios das actividades, existem também tempos máximos entre o início das mesmas) baseado em 3 regras de prioridade para a selecção das actividades e 3 regras de prioridade para a selecção dos respectivos modos.

Nos modelos mais simplistas, utiliza-se apenas um esquema de geração do escalonamento e uma regra de prioridade para as actividades. Contudo, em algumas abordagens, consideram-se probabilidades na selecção das regras de prioridade. De acordo com a forma como essas probabilidades são calculadas, variam também os tipos de heurísticas. Na terminologia anglo-saxónica, essas abordagens são designadas por *sampling methods* ([20], [43], [12]), e foram usadas essencialmente para resolver o RCPSP, podendo ser também estendidas ao caso em que existe diversos modos de execução.

As meta-heurísticas foram também usadas para resolver o MRCPSP. Kolish e Drexler [34] desenvolveram uma heurística baseada em pesquisa local e que tem em consideração os níveis de recursos não renováveis (o que nem sempre acontece com as outras heurísticas). Para além dessas, encontram-se na literatura algoritmos baseados em métodos de pesquisa tabu, de arrefecimento simulado e algoritmos genéticos.

De Reyck e Herroelen [16] propuseram um método de pesquisa local e outra de pesquisa tabu para resolver o MRCPSP com diversas restrições adicionais. A sua abordagem divide o problema em duas partes. Depois de ter determinado uma afectação possível dos modos às actividades, prossegue com o escalonamento das actividades resolvendo um problema clássico do tipo RCPSP. Entre os métodos que testaram, os melhores resultados foram obtidos com a pesquisa tabu. Um dos inconvenientes dos métodos propostos observada com base nas experiências computacionais levadas a cabo foi o facto de algumas vezes os algoritmos falharem em encontrar soluções válidas para as instâncias.

Em [32] os autores resolvem o MRCPSP através de um método de arrefecimento simulado. O algoritmo é apresentado segundo duas versões, com e sem função de penalização, sendo que a primeira retorna os melhores resultados. Os autores testaram ainda três formas de gerar as vizinhanças. No final, reportam resultados obtidos com diversas instâncias.

Em [24] é descrito um algoritmo genético para o MRCPSP baseado em listas de precedências e afectação de modos. Diversas variantes são discutidas no artigo. O autor compara a sua abordagem com outras heurísticas, e conclui que a sua abordagem é melhor em termos de aproximação ao óptimo.

2.4 Conclusões

Os problemas de escalonamento de projectos com recursos escassos (RCPSP) têm grande aplicabilidade prática e traduzem uma realidade em que existe um conjunto de actividades para serem executadas (actividades essas que constituem um projecto), com relações de precedência entre elas e com a utilização de recursos que são escassos para a execução das actividades.

Existe uma grande variedade de problemas que podem ser agrupados nessa área. Nesta classe de problemas, destacamos os problemas do tipo multi-modo (MRCPSP) que são objecto de estudo nesta dissertação. Neste tipo de problemas, procura-se um escalonamento onde se indica quando é que determinada actividade deve ser executada e em que modo, por forma a atingir o objectivo geral do problema (como por exemplo a minimização da duração total do projecto).

Como acontece com outros problemas de optimização, as abordagens de resolução dividem-se em duas partes: os métodos de resolução exacta, e os métodos heurísticos. Na resolução exacta do MRCPSP, destacam-se os algoritmos de pesquisa e enumeração baseados em métodos de partição e avaliação sucessivas. Os métodos heurísticos para o MRCPSP (essencialmente meta-heurísticas) produzem soluções aproximadas em tempos computacionais muito aceitáveis. De entre estas, as meta-heurísticas que já apresentam algum nível de aprendizagem são as que conduzem a melhores resultados.

Contudo, os tempos de resolução são mais elevados do que para as heurísticas construtivas (do tipo *X-pass*, por exemplo) que se baseiam em simples regras de prioridade.

Capítulo 3

Limites inferiores: novas decomposições de Dantzig-Wolfe

3.1 Introdução

Um dos objectivos desta dissertação foi avaliar a qualidade de modelos alternativos para o MRCPSP. Os modelos analisados aqui são de Programação Inteira. A sua qualidade é medida tendo em conta o valor do limite inferior associado às correspondentes relaxações lineares. As relaxações lineares são resolvidas exactamente através do método de geração de colunas que descreveremos mais adiante.

Nos últimos anos, a área da Programação Inteira registou progressos notáveis tanto ao nível da definição de modelos analíticos, como no domínio das técnicas de resolução. O horizonte de aplicação dessas técnicas e modelos é vasto. Encontramos contribuições em domínios como a logística, a gestão de operações e a gestão de projectos. Neste capítulo, debruçamo-nos em duas técnicas de Programação Inteira que estão directamente interligadas: o método de decomposição de Dantzig-Wolfe que consiste numa reformulação de modelos compactos, e o método de geração de colunas que tem por objectivo resolver de forma eficiente os modelos decompostos.

Na próxima secção, começamos por introduzir os conceitos teóricos subjacentes. Na Secção 3.3, apresentamos em detalhe o modelo de Talbot [49] que está na base das decomposições exploradas nesta dissertação. Nas três secções seguintes, descrevemos as decomposições que analisamos. Os resultados computacionais estão descritos no

Capítulo 5 juntamente com os resultados da heurística que descreveremos no próximo capítulo.

3.2 O princípio da decomposição de Dantzig-Wolfe

O princípio da decomposição de Dantzig-Wolfe foi formalizado nos anos 60 por George Dantzig (um dos pioneiros da Programação Linear) e Philip Wolfe [13], mas na realidade o método de geração de colunas ao qual está directamente associado tinha sido já usada por Ford e Fulkerson [23] para resolver problemas de fluxo multicomodidade. Uma das motivações para aplicação da decomposição de Dantzig-Wolfe em Programação Inteira (cujo potencial começou a ser explorado em anos recentes [5]), é a obtenção de modelos que sejam potencialmente mais fortes que os modelos originais, ou seja, que originem melhores limites inferiores (em problemas de minimização).

Para definirmos formalmente o princípio da decomposição de Dantzig-Wolfe, consideramos a seguinte representação genérica de um modelo de Programação Linear (do tipo minimização):

$$\begin{aligned} \min \quad & cx \\ \text{s. a} \quad & \\ & Ax \geq b, \\ & x \in X, \\ & x \in \mathfrak{R}_+^n. \end{aligned}$$

A decomposição de Dantzig-Wolfe aplicada a esse modelo de Programação Linear consiste em separar o modelo em partes: algumas restrições ficam explicitamente no modelo, enquanto outras são removidas, passando a ser expressas de um modo alternativo. A título de exemplo, considere que retiramos do modelo acima o segundo conjunto de restrições. O espaço de soluções válidas definido por essas restrições passa a ser expresso como uma soma de uma combinação convexa de pontos extremos com uma combinação não negativa de raios extremos de X . Temos assim o seguinte:

$$X = \left\{ x \in \mathfrak{R}_+^n : \sum_{i \in I} \lambda_i X_i + \sum_{j \in J} \mu_j R_j, \sum_{i \in I} \lambda_i = 1, \lambda_i \geq 0, \forall i, \mu_j \geq 0, \forall j \right\},$$

sendo I o conjunto de pontos extremos de X , e J o conjunto de raios extremos de X . Se X for um conjunto fechado, então qualquer ponto que pertença a esse espaço poderá ser expresso como uma combinação convexa dos seus pontos extremos. Nessa condição, o conjunto X passará a ser expresso como segue:

$$X = \left\{ x \in \mathfrak{R}_+^n : \sum_{i \in I} \lambda_i X_i, \sum_{i \in I} \lambda_i = 1, \lambda_i \geq 0 \right\}.$$

As variáveis do modelo reformulado passam a ser os λ_i (e os μ_j se X não for um conjunto fechado). Tipicamente, o número de restrições diminui: retiram-se as restrições que definem X , e acrescentam-se as restrições de convexidade. No entanto, o número de variáveis (ou colunas) aumenta exponencialmente. Isso deve-se ao facto do conjunto de pontos e raios extremos de um poliedro ser exponencial.

Em termos de modelos que resultam da decomposição de Dantzig-Wolfe do modelo compacto original, é usual falar-se em problema mestre (ou problema principal), e subproblema (ou eventualmente subproblemas se a decomposição for múltipla). O problema mestre é o problema que permanece com as restrições originais do modelo compacto, e no qual é reformulada a definição das variáveis para passarem a ser uma combinação de pontos e raios extremos de X . É o modelo com um número exponencial de colunas. O subproblema está ligado às restrições que saíram do modelo compacto original. O subproblema é de facto um problema de optimização pela razão seguinte. Dado que o problema mestre tem um número exponencial de colunas, não é viável a enumeração completa de todas as colunas que o compõem para qualquer instância de média ou grande dimensão. Como tal, quando se pretende resolver um modelo desse tipo, apenas um conjunto restrito de colunas é enumerado à partida. A esse problema, dá-se o nome de problema mestre restrito. O subproblema é o problema que é resolvido para identificar colunas atractivas (com potencial para melhorar o valor da função objectivo do problema mestre) que não tenham estejam no problema mestre. Esta interacção entre problema mestre restrito e subproblema define o algoritmo de geração

de colunas. Para identificar uma coluna atractiva, usa-se o valor das variáveis duais do problema mestre restrito. O objectivo do subproblema consiste em identificar uma coluna com custo reduzido negativo, isto é, uma coluna atractiva. De notar que todas as colunas que constam do problema mestre restrito tem custo reduzido nulo ou positivo, dado que o problema foi já optimizado.

O algoritmo de geração de colunas pode ser esquematizado da forma representada na figura que se segue.

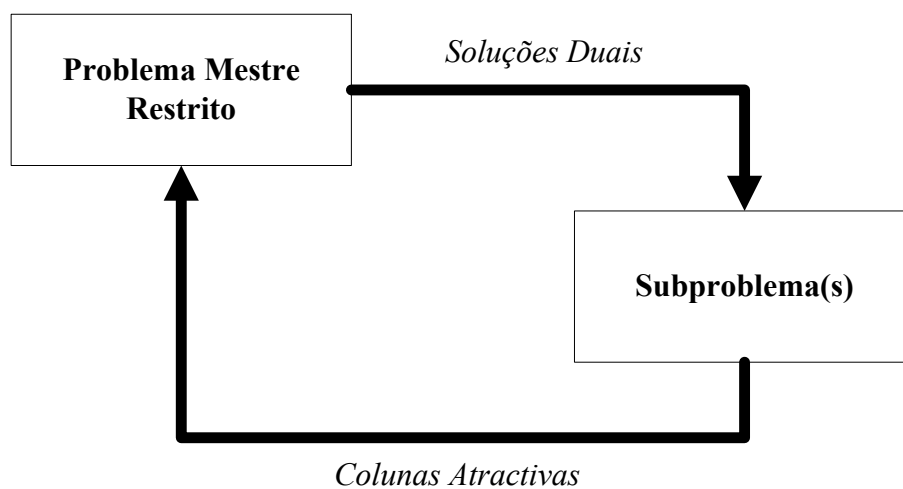


Figura 3.1 - Algoritmo de geração de colunas

Uma das questões que se levantam quando se pretende resolver um modelo reformulado usando o método de geração de colunas é a forma como deve ser inicializado o problema mestre restrito. A forma mais simples consiste em considerar apenas uma variável (artificial) de custo muito elevado e que garanta que o problema tenha solução. Além disso, podem-se adicionar ao primeiro problema mestre restrito um conjunto de colunas que correspondam a uma solução heurística para o problema. Na maior parte dos casos, isso permite acelerar a convergência do método dado que o limite superior (solução óptima do problema mestre restrito com um conjunto parcial de colunas) com que se inicia o processo é de melhor qualidade.

Um dos aspectos importantes ligados ao método de decomposição de Dantzig-Wolfe prende-se com qualidade dos modelos que dele podem resultar. Com efeito, se o

subproblema não tiver a propriedade da integralidade (resolver a sua relaxação linear não garante que a solução seja inteira), e se no processo de geração de colunas, esses subproblemas forem resolvidos até encontrar uma solução ótima inteira, então o limite inferior dado pelo problema mestre restrito poderá ser melhor que o limite inferior do modelo compacto original. Na realidade, excluem-se soluções fracionárias que eram consideradas no modelo original, restringindo mais fortemente o espaço de soluções válidas. Essa é uma propriedade bem conhecida, da qual muitos investigadores tentaram já tirar partido para desenvolver modelos fortes para problemas de optimização difíceis. E é tendo em atenção essa propriedade que definimos as decomposições que descreveremos a seguir.

De notar que para resolver o problema mestre até à optimalidade, pode ser necessário resolver um grande número de subproblemas. É por esse motivo que as decomposições devem ser tais que os subproblemas sejam relativamente fáceis de resolver. Se assim não for, a resolução do problema mestre poderá ser excessivamente lenta. Resolver um problema até obter soluções ótimas inteiras pode trazer benefícios em termos da qualidade dos limites obtidos, mas se essa resolução for demasiadamente exigente em termos computacionais, o balanço poderá muito bem ser negativo. Os subproblemas (ou dito de outra forma, as restrições que o definem no modelo compacto) devem preferencialmente ter uma estrutura que permita que essa resolução possa ser feita da forma mais eficiente possível.

A noção de estrutura é muito importante quando se trata de usar o método de decomposição de Dantzig-Wolfe. Com efeito, é frequente encontrar problemas que possam ser formulados usando modelos com uma estrutura particular. Muitas vezes, esses conjuntos de restrições definem outros problemas bem conhecidos como é o caso do problema de caminho mais curto, por exemplo. Quando os modelos compactos originais têm um grande número de restrições, a probabilidade disso acontecer é ainda maior. Isso acontece com os problemas de gestão de projectos, e em particular com o MRCPS. Por vezes, é possível identificar nos próprios modelos blocos independentes de restrições ligados apenas por algumas restrições e que podem ser usados para definir decomposições múltiplas. Essa estrutura é conhecida por estrutura angular em blocos, e pode ser representada graficamente como na Figura 3.2.

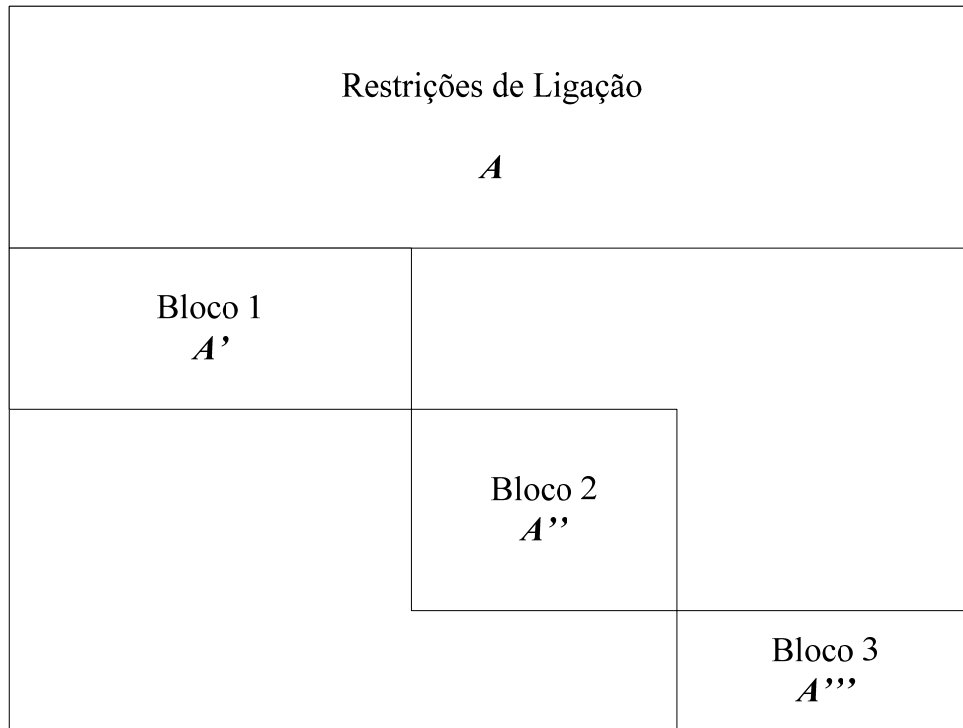


Figura 3.2 - Estrutura angular em blocos

Nesta dissertação, exploramos diferentes decomposições para o MRCPSP usando os princípios resumidos acima. O objectivo é o de aferir a qualidade de decomposições alternativas para o MRCPSP. Tratam-se de decomposições originais para as quais não encontramos nenhum registo na literatura e que poderão contribuir à resolução exacta desse problema. Como base dos trabalhos, usámos o modelo compacto enunciado por Talbot [49]. Esse modelo possui uma estrutura angular em blocos, com blocos que podem definir diferentes problemas de optimização. Na secção seguinte, descrevemos em detalhe o modelo de Talbot.

3.3 Modelo de Talbot

O modelo que apresentamos nesta secção, e no qual se baseiam as nossas decomposições, foi proposto por Talbot [49] em 1982. Trata-se de um modelo de Programação Inteira com variáveis que são exclusivamente binárias.

Alguns elementos de notação foram já descritos no capítulo introdutório desta dissertação. A notação que falta para descrevermos o modelo é apresentada a seguir:

- N : número total de actividades no projecto,
- P : conjunto de pares de actividades entre as quais existe uma relação de precedência; $(i,j) \in P$, se e só se a actividade j não pode ser iniciada sem a actividade i ter terminado,
- R_{kt} : nível de recurso renovável k ($k=1,\dots,K$) disponível no instante t ,
- $NR_{k't}$: nível de recurso não renovável k' ($k'=1,\dots,K'$) disponível no instante de tempo t ,
- r_{ikm} : quantidade de recurso renovável k consumido pela actividade i quando executada no seu modo m ,
- $nr_{ik'm}$: quantidade de recurso não renovável k' consumido pela actividade i quando executada no seu modo m ,
- E_i : instante de tempo mais cedo em que a actividade i pode ser terminada,
- L_i : instante de tempo mais tardio em que a actividade i deve ser terminada (de modo a não atrasar a conclusão do projecto).

O instante de tempo mais cedo em que uma actividade pode ser terminada (E_i) e o instante de tempo mais tardio em que deveria ser terminado (L_i) são calculados usando o método do caminho crítico e as durações dos modos mais rápidos para cada uma das actividades. Para L_N , o instante de tempo mais tardio em que o projecto deve ser terminado, escolhe-se um limite superior que poderá ser obtido usando uma heurística, ou então a soma das durações dos modos mais lentos de cada actividade.

As variáveis de decisão do modelo designam-se por x_{itm} , com $i=1,\dots,N$, $t=0,\dots,L_N$, e $m=1,\dots,M_i$. Uma variável x_{itm} será igual a 1 se e só se a actividade i terminar no instante de tempo t e for executada usando o seu modo m . Tomará o valor 0 caso contrário. Dado que não é permitida a interrupção de actividades, só existe um instante de tempo em que actividade pode terminar, e assim para cada actividade i apenas uma destas variáveis pode tomar o valor 1.

O modelo de Talbot é formulado da forma seguinte:

$$\min \sum_{m=1}^{M_N} \sum_{t=E_N}^{L_N} tx_{Ntm} \quad (1)$$

sujeito a

$$\sum_{m=1}^{M_i} \sum_{t=E_i}^{L_i} x_{itm} = 1, \quad i = 1, \dots, N, \quad (2)$$

$$-\sum_{m=1}^{M_i} \sum_{t=E_i}^{L_i} tx_{itm} + \sum_{m=1}^{M_j} \sum_{t=E_j}^{L_j} (t - d_{jm})x_{jtm} \geq 0, \quad (i,j) \in P, \quad (3)$$

$$\sum_{i=1}^N \sum_{m=1}^{M_i} \sum_{q=t}^{t+d_{im}-1} r_{ikm}x_{iqm} \leq R_{kt}, \quad k = 1, \dots, K, t = 1, \dots, L_N, \quad (4)$$

$$\sum_{i=1}^N \sum_{m=1}^{M_i} \sum_{t=E_i}^{L_i} nr_{ik'm}x_{itm} \leq NR_{k'}, \quad k' = 1, \dots, K', \quad (5)$$

$$x_{itm} \in \{0,1\}, \quad i = 1, \dots, N, t = 0, \dots, L_N, m = 1, \dots, M_i. \quad (6)$$

O objectivo do problema é expresso através da função objectivo (1), e consiste em determinar o instante de tempo mais próximo de $t=0$ em que o projecto pode ser terminado. As restrições (2) obrigam a que todas as actividades sejam realizadas dentro da janela temporal que lhe estão associadas (definidas com base no caminho crítico, como referido anteriormente) em qualquer um (mas só um) dos modos existentes. As restrições (3) garantem que as relações de precedências entre actividades relacionadas são de facto respeitadas. As restrições (4) garantem que não são gastos mais recursos renováveis do que aqueles disponíveis no instante de tempo t . As restrições (5) estão relacionadas com os recursos não renováveis disponíveis, e portanto não dependem de nenhum instante de tempo. Os recursos duplamente restritos podem facilmente ser incorporados neste modelo. Basta para isso considerá-los tanto nas restrições (4) como nas restrições (5). Finalmente, as restrições (6) definem o domínio das variáveis de decisão.

A título de exemplo, e para ilustrarmos a estrutura do modelo de Talbot, consideramos um subconjunto de actividades de um projecto com índices 0, 1 e 2, um período de planeamento de 5 unidades de tempo, 2 recursos renováveis e 2 recursos não renováveis. Os consumos e disponibilidade de cada recurso não são relevantes para o nosso exemplo. Considere apenas que a actividade 0 precede a actividade 2, e que por sua vez a actividade 2 precede a actividade 1. O modelo de Talbot pode ser representado esquematicamente conforme apresentado na Figura 3.3. Os diversos blocos do modelo são aí postos em evidência. As restrições (5) relativas aos recursos não renováveis formam um bloco único, enquanto as restantes formam blocos nem sempre independentes.

As decomposições que definimos tiram partido da estrutura do modelo de Talbot. Nas próximas secções, apresentamo-las usando o mesmo exemplo de base que aquele ilustrado na Figura 3.3.

3.4 Primeira decomposição

A primeira decomposição que analisámos consiste em considerar no subproblema os conjuntos de restrições (2), (3) e (5) do modelo de Talbot, isto é, as restrições que obrigam a que todas as actividades sejam executadas, as restrições relacionadas com as relações de precedência entre actividades e as restrições associadas aos recursos não renováveis. No problema mestre, mantêm-se as restrições do modelo compacto original que dizem respeito aos recursos renováveis.

		0	0	0	1	1	1	2	2	2	2	0	0	1	1	2	2	2	0	0	1	1	2	2	0	0	1	1	2	2	0	0	1	1	2	2	0	0	1	1	2	2	0	0	1	1	2	2			
		Actividade																																																	
		t																																																	
		Modo																																																	
(2)	Actividade 0 deve ser feita																																																		
	Actividade 1 deve ser feita																																																		
	Actividade 2 deve ser feita																																																		
(3)	Actividade 0 precede actividade 2																																																		
	Actividade 2 precede actividade 1																																																		
	Recurso renovável 0 em $t=0$																																																		
(4)	Recurso renovável 1 em $t=0$																																																		
	Recurso renovável 0 em $t=1$																																																		
	Recurso renovável 1 em $t=1$																																																		
	Recurso renovável 0 em $t=2$																																																		
	Recurso renovável 1 em $t=2$																																																		
	Recurso renovável 0 em $t=3$																																																		
(5)	Recurso renovável 1 em $t=3$																																																		
	Recurso renovável 0 em $t=4$																																																		
(5)	Recurso não renovável 0																																																		
	Recurso não renovável 1																																																		

Figura 3.3 - Estrutura do modelo de Talbot

O subproblema é um problema de escalonamento de actividades com vários modos de execução e em que a selecção dos modos pode ser influenciada pela existência de

recursos não renováveis. No problema mestre, determina-se o escalonamento final que satisfaz as restrições relacionadas com os recursos renováveis. Os subproblemas têm alguma complexidade, mas dado que não possuem a propriedade da integralidade, podemos esperar obter com essa decomposição limites inferiores de melhor qualidade que aqueles obtidos com a relaxação linear do modelo de Talbot.

A Figura 3.4 ilustra essa decomposição a partir do exemplo introduzido na secção anterior.

3.5 Segunda decomposição

Na segunda decomposição, mantemos no problema mestre as restrições (2) que obrigam a que as actividades sejam todas executadas, e as restrições (3) associadas às relações de precedência entre actividades. No subproblema consideram-se em conjunto as restrições relacionadas com os recursos renováveis e não renováveis. O subproblema é mais complexo que um problema de mochila (*knapsack problem*), e não possui a propriedade da integralidade. Novamente, o limite inferior dado pela relaxação linear do problema mestre será sempre maior ou igual que aquele obtido com a relaxação linear do modelo de Talbot.

Na Figura 3.5, apresentamos um esquema dessa decomposição usando o quadro associado ao modelo de Talbot para a pequena instância descrita na Secção 3.3.

RESTRIÇÕES

		Actividade																													
		<i>t</i>																													
		Modo																													
(4)	Recurso renovável 0 em $t=0$	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	2	2	2	2	2	2	2	2	2		
	Recurso renovável 0 em $t=1$	0	0	1	1	2	3	4	4	0	0	1	1	2	2	3	3	4	4	0	1	1	2	2	3	3	4	4	4	4	
	Recurso renovável 0 em $t=2$	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1
	Recurso renovável 0 em $t=3$	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	2	2	2	2	2	2	2	2	2	2
	Recurso renovável 0 em $t=4$	0	0	1	1	2	2	3	3	4	4	0	0	1	1	2	2	3	3	4	4	0	0	1	1	2	2	3	3	4	4
	Recurso renovável 1 em $t=0$	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1
	Recurso renovável 1 em $t=1$	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	2	2	2	2	2	2	2	2	2
	Recurso renovável 1 em $t=2$	0	0	1	1	2	2	3	3	4	4	0	0	1	1	2	2	3	3	4	4	0	0	1	1	2	2	3	3	4	4
(2)	Actividade 0 deve ser feita	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1
	Actividade 1 deve ser feita	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	2	2	2	2	2	2	2	2	2	2
	Actividade 2 deve ser feita	0	0	1	1	2	2	3	3	4	4	0	0	1	1	2	2	3	3	4	4	0	0	1	1	2	2	3	3	4	4
	Actividade 0 precede actividade 2	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1
	Actividade 2 precede actividade 1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
(5)	Recurso não renovável 0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
	Recurso não renovável 1	0	0	1	1	2	2	3	3	4	4	0	0	1	1	2	2	3	3	4	4	0	0	1	1	2	2	3	3	4	4

Figura 3.4 - Estrutura associada à primeira decomposição

RESTRICÇÕES		Actividade																			
		t	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
		Modo	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0
(2)	Actividade 0 deve ser feita																				
	Actividade 1 deve ser feita																				
	Actividade 2 deve ser feita																				
(3)	Actividade 0 precede actividade 2																				
	Actividade 2 precede actividade 1																				
(4)	Recurso renovável 0 em $t=0$																				
	Recurso renovável 0 em $t=1$																				
	Recurso renovável 0 em $t=2$																				
	Recurso renovável 0 em $t=3$																				
	Recurso renovável 0 em $t=4$																				
	Recurso renovável 1 em $t=0$																				
	Recurso renovável 1 em $t=1$																				
Recurso renovável 1 em $t=2$																					
Recurso renovável 1 em $t=3$																					
Recurso renovável 1 em $t=4$																					
(5)	Recurso não renovável 0																				
	Recurso não renovável 1																				
		Problema																			
		Subproblema																			

Figura 3.5 - Estrutura associada à segunda decomposição

3.6 Terceira decomposição

A terceira e última decomposição que consideramos no âmbito desta dissertação consiste num problema mestre no qual se mantêm as restrições (3) relacionadas com as relações de precedência entre actividades, e as restrições (5) associadas aos recursos não renováveis que formam o bloco único ilustrado nas figuras acima. O subproblema é definido através das restrições (2) do modelo original de Talbot e das restrições (4) associadas aos recursos renováveis. Com um subproblema desse tipo, obriga-se a que todas as actividades sejam escolhidas, assim como um modo de execução e um instante de tempo para a sua finalização de modo que nenhuma restrição de recurso renovável seja violada. Essas restrições definem um problema de mochila com restrições adicionais. Não possui a propriedade da integralidade, podendo por isso vir a melhorar a qualidade do limite inferior. No problema mestre, procuram-se escalonamentos em que as relações de precedência sejam satisfeitas, garantindo que o valor total de recursos não renováveis consumidos não exceda a capacidade desses recursos.

Novamente, ilustramos através da Figura 3.6 a estrutura do modelo de Talbot com base na qual essa decomposição assenta.

3.7 Conclusões

Neste capítulo, apresentamos a componente desta dissertação ligada ao cálculo de limites inferiores para o MRCPSP. Esse cálculo baseia-se na resolução de modelos de Programação Linear obtidos através da decomposição de Dantzig-Wolfe de um modelo original compacto. Esse modelo compacto foi apresentado por Talbot [49]. As decomposições são originais, não existindo na literatura nenhum registo que mencione o uso de tais modelos. Na primeira parte do capítulo, introduzimos os conceitos teóricos que sustentam a decomposição de Dantzig-Wolfe e o método de geração de colunas, que é o método de resolução usado normalmente para resolver esses modelos.

RESTRICÇÕES

	0	1	2	0	1	1	2	2	0	1	1	2	2	0	0	1	1	2	2	2	2	2	3	3	3	3	3	3	4	4	4	4				
(3)	Actividade 0 precede actividade 2	Actividade 2 precede actividade 1																																		
(5)	Recurso não renovável 0	Recurso não renovável 1																																		
(2)	Actividade 0 deve ser feita	Actividade 1 deve ser feita	Actividade 3 deve ser feita																																	
(4)	Recurso renovável 0 em $t=0$	Recurso renovável 1 em $t=0$	Recurso renovável 0 em $t=1$	Recurso renovável 1 em $t=1$	Recurso renovável 0 em $t=2$	Recurso renovável 1 em $t=2$	Recurso renovável 0 em $t=3$	Recurso renovável 1 em $t=3$	Recurso renovável 0 em $t=4$	Recurso renovável 1 em $t=4$																										

Actividade
 t

Modo

0 0 1 1 2 2 0 0 1 1 2 2 0 0 1 1 2 2 0 0 1 1 2 2 0 0 1 1 2 2 0 0 1 1 2 2

0 0 0 0 0 1 1 1 1 1 1 1 2 2 2 2 2 2 2 2 2 2 2 2 3 3 3 3 3 3 3 3 4 4 4 4

0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1

Figura 3.6: Estrutura associada à terceira decomposição

O objectivo deste capítulo foi apenas de introduzir os novos modelos, enquadrando-os no âmbito da teoria dos métodos de geração de colunas. No Capítulo 5, apresentamos resultados computacionais em instâncias da literatura que permitem aferir da qualidade de cada um desses modelos.

Capítulo 4

Limites superiores: uma nova heurística para o MRCPSP

4.1 Introdução

No MRCPSP, existem dois tipos de decisão a tomar para cada uma das actividades que constituem o projecto:

- 1) a escolha do momento em que deve ser iniciada a sua execução;
- 2) a escolha do modo em que deve ser executada.

Essas decisões são tomadas na perspectiva da optimização do objectivo geral do problema. Lembramos que no âmbito desta dissertação, o objectivo que consideramos é o da minimização da duração global do projecto. Por outro lado, a tomada destas decisões são também naturalmente limitadas pelas restrições relativas aos recursos, renováveis ou não renováveis, e às relações de precedência existentes entre as diversas actividades que constituem o projecto. No nosso caso, consideramos ainda a restrição de não interrupção das actividades que obriga a que uma actividade seja executada sem qualquer paragem pelo meio.

No Capítulo 2, tivemos já oportunidade de referir a dificuldade inerente associada à resolução exacta deste tipo de problemas. Atendendo a esta realidade, é natural que a definição de heurísticas se torne uma solução viável para esse tipo de problemas. Em problemas de minimização, as heurísticas permitem obter limites superiores para o valor do óptimo. Esse limite está associado a uma solução válida, mas frequentemente não óptima. Apenas a disponibilidade de um limite inferior permite aferir a qualidade da solução dada pela heurística. No caso extremo, quando os dois limites coincidem, a solução é claramente uma das soluções óptimas do problema.

A literatura descreve vários métodos heurísticos para o MRCPSP. Esses métodos são essencialmente meta-heurísticas. As meta-heurísticas conduzem globalmente a bons resultados, mas requerem tempos de processamento elevados e modelos mais complexos. As heurísticas simples (do tipo construtivo, por exemplo) têm muito interesse devido à rapidez com que podem potencialmente gerar soluções para o problema. Essas heurísticas permitem uma boa relação entre qualidade da solução e tempo de processamento. Contudo, são muito poucas as heurísticas simples disponíveis para o MRCPSP.

As heurísticas descritas na literatura baseiam-se em regras de prioridade que se aplicam no processo de selecção das actividades, e de selecção dos modos. O escalonamento das actividades é feito sequencialmente, sendo processada uma actividade de cada vez. Caso tenha tido sucesso no escalonamento da actividade em análise, a heurística prossegue com a análise de uma outra actividade.

De entre os vários trabalhos publicados neste domínio, Boctor [8] publicou em 1993 um estudo onde analisou sete regras de prioridade para a selecção das actividades combinando-as com três regras de prioridade para a selecção dos modos. Tanto as regras relativas às actividades como aquelas associadas à selecção dos modos são amplamente usadas em outros trabalhos. O total de combinações possíveis ascende a 21, podendo cada uma delas dar origem a uma heurística diferente.

As sete regras de prioridade consideradas para a selecção das actividades foram as seguintes:

- a) MIN SLK: a actividade com menor valor de folga (o conceito de folga está directamente ligado à noção de caminho crítico, e será introduzido com mais pormenor mais adiante);
- b) MIN LFT: a actividade com tempo de fim mais próximo de t (o instante de tempo em análise);
- c) MAX NIS: a actividade com maior número de actividades sucessoras imediatas;
- d) MAX RWK: a actividade cuja soma da sua duração e das durações de todas as suas sucessoras é maior;
- e) MAX PTM: actividade com maior duração;
- f) MIN PTM: actividade com menor duração;
- g) MAX CAN: actividade seleccionada com base no número de candidatos subsequentes.

Para a selecção dos modos de execução das actividades, as três regras de prioridade usadas foram as seguintes:

- a) SFM: modo com duração mais curta;
- b) LCR: modo com menor valor da medida que mede o quanto crítico é o uso dos recursos (essa medida traduz, para cada recurso, o máximo rácio entre o pico de utilização de recursos e o montante disponível de cada recurso);
- c) LRP: modo com menor valor de proporção de recursos consumidos.

Para analisar a performance de cada uma das 21 heurísticas resultantes, Boctor resolveu com cada uma delas um conjunto de 240 instâncias. Observou que a regra SFM de selecção de modos combinada com a regra MIN SLK para a selecção das actividades conduzia a melhores resultados na maioria dos casos.

Contudo, numa publicação posterior [9] Boctor apresenta contudo uma nova heurística, baseada numa técnica inovadora e cujos resultados revelaram ser muito interessantes. A heurística que desenvolvemos baseia-se nos trabalhos de Boctor, estendendo-os para o caso mais geral do MRCPSP em que existem recursos não renováveis. Na secção seguinte, começamos por apresentar em detalhe os elementos que caracterizam a heurística de Boctor, prosseguindo com as inovações introduzidas no âmbito desta

dissertação. Na Secção 4.4, discutimos algumas questões relacionadas com a implementação em C da heurística.

4.2 Heurística de Boctor

Enquanto que as heurísticas apresentadas até então se baseavam no escalonamento individualizado de actividades, ou seja, em cada momento apenas uma actividade é analisada e passível de ser escalonada, Boctor [9] sugere que o escalonamento das actividades em cada instante de tempo seja feito com base em blocos ou conjuntos de actividades, escalonando o bloco que tiver o melhor valor para uma determinada medida que descreveremos mais adiante.

Num determinado instante de tempo t , para cada uma das actividades passíveis de serem escalonadas (cujas actividades antecessoras já foram totalmente concluída), é associado um modo de execução. O par resultante é designado por combinação actividade-modo. Um bloco consiste num conjunto de combinações actividade-modo que podem ser escalonadas no instante de tempo t sem violar nenhuma das restrições relativas aos recursos renováveis, assim como nenhuma relação de precedência. Uma combinação actividade-modo pode ser adicionada a um bloco já existente se não comprometer nenhuma dessas restrições.

A forma de construção dos blocos ou conjuntos de actividades tem subjacente um novo conceito introduzido por Boctor em [9]: o conceito de *dominância* entre blocos escalonáveis de actividades. Assim, uma dada combinação de actividades é *dominada* por outra se uma ou ambas as condições seguintes for verificada:

- 1) O segundo bloco inclui todas as combinações actividade-modo do primeiro bloco e ainda mais algumas combinações (nesse caso, é claramente melhor escalonar em t o segundo bloco visto que a taxa de aproveitamento dos recursos é obviamente melhor);
- 2) O segundo bloco é idêntico ao primeiro (contém as mesmas combinações actividade-modo), excepto no modo de execução de uma das actividades

que tem uma duração inferior ao modo escolhido para a mesma actividade no primeiro bloco (o segundo bloco permite pelo menos uma das situações seguintes: escalonar um conjunto de actividades idêntico às dos primeiro bloco em menos tempo, e/ou libertar recursos mais rapidamente aumentando assim a probabilidade de escalonar mais cedo um outro bloco de actividades).

Um bloco de actividades é construído adicionando-se combinações actividade-modo, e tendo em atenção a restrição relativa aos recursos renováveis disponíveis no instante de tempo t . Assim, num dado instante de tempo t e para um conjunto de actividades passíveis de serem escalonadas em t , poder-se-ão ter vários blocos de combinações actividade-modo que não são dominados por nenhum outro bloco. A escolha do bloco que deve ser escalonado em t é feita usando uma medida de atraso que o respectivo escalonamento irá induzir no projecto. O bloco com menor valor de atraso induzido é obviamente aquele que é escalonado no instante t . Essa escolha é feita apenas entre os blocos que não são dominados.

Antes de descrevermos a medida de atraso descrita por Boctor em [9], introduzimos de seguida alguma notação adicional. Por razões de clareza, a notação que usamos é idêntica àquela usada por Boctor em [9] (a restante notação foi introduzida no capítulo de Introdução desta dissertação):

- t : Instante de tempo no qual se pretende fazer o escalonamento;
- t_{ij} : Instante de tempo mais próximo de t no qual a actividade i poderá ser escalonada no modo j
- A : Actividades que estão em condições de serem escalonadas em t (todas as actividades antecessoras foram escalonadas até t , e as actividades têm modos associados que não consomem mais recursos do que aqueles que estão disponíveis em t);
- S : Conjunto de actividades tal que $S \subseteq A$;
- $j_{(i)}$: Modo escolhido para a actividade $i \in A$;
- C : Combinações actividade-modo associadas às actividades contidas em S .

Se uma actividade não for escalonada no seu modo mais rápido, poderá eventualmente atrasar o projecto global. Para sabermos se existe ou não atraso quando não é escolhido o modo mais rápido, temos de ter em atenção a folga associada à actividade, e o modo em que planeamos escalonar a actividade. A título de exemplo, considere uma actividade cujo modo mais rápido tem uma duração de 1 unidade de tempo e uma folga de 3 unidades de tempo. Se planearmos escalonar essa actividade num modo que demora 5 unidades de tempo, iremos atrasar o projecto global de 1 unidade de tempo (5-1-3). Boctor representa formalmente esse atraso usando o parâmetro $a_{ij(i)}$, com $i \in S$, e define-o como segue:

$$a_{ij(i)} = \max \{0, d_{ij(i)} - d_{i1} - F_i\}.$$

Ao não escalonarmos um conjunto de actividades que poderia ser escalonado no momento t , podemos também estar a atrasar a conclusão do projecto. Não podemos à priori determinar o valor exacto desse atraso dado que no momento t não sabemos quando irão ser escalonadas essas actividades. No entanto, é possível determinar um limite inferior para esse valor. Para uma actividade $i \in A-S$, Boctor define esse atraso usando o parâmetro b_i como segue:

$$b_i = \max \{0, \min_{j \in M_i} (t_{ij} - t + d_{ij} - d_{i1} - F_i)\}.$$

De notar que em [9], a fórmula para b_i apresenta um erro. No segundo termo da expressão, não é subtraída a quantidade t aos restantes parâmetros. Em [9], o valor de b_i é dado como segue:

$$b_i = \max \{0, \min_{j \in M_i} (t_{ij} + d_{ij} - d_{i1} - F_i)\}.$$

Para sabermos se existe atraso ou não, é necessário comparar a duração do modo escolhido para a actividade i com a duração do modo padrão (o mais rápido). A diferença constitui já um primeiro “atraso” relativamente às medidas que são calculadas inicialmente (e que dependem sempre da duração dos modos mais rápidos). Da mesma forma, t_{ij} representa o instante de tempo em que a actividade i pode ser escalonada usando o modo j atendendo aos recursos disponíveis. Se t_{ij} for diferente de t , novamente teremos aí um “atraso” igual a $t_{ij} - t$. Finalmente, se a soma desses dois “atrasos” for

compensada pela folga associada à actividade i , então o projecto não sofrerá nenhum atraso. Caso contrário, a quantidade positiva igual à diferença entre a soma desses “atrasos” e a folga F_i será uma medida do atraso induzido no projecto.

Finalmente, essas duas medidas são usadas para determinar o atraso mínimo I_S em que se incorre quando são escalonadas as actividades de S no momento t :

$$I_S = \max \{0, \max_{i \in S} (a_{ij(i)}), \max_{i \in A-S} (b_i)\}.$$

Claramente, o bloco de actividades S que será escalonado no momento t será aquele que tiver menor valor I_S associado.

A heurística de Boctor é do tipo construtivo. Em cada iteração, começa com a construção de blocos não-dominados, prosseguindo com o cálculo da medida de atraso I_S para cada bloco, seguido da escolha do bloco com menor valor I_S e da actualização dos níveis de recursos disponíveis no horizonte de planeamento. O valor de t é incrementado e o processo repete-se até que todas as actividades tenham sido escalonadas. De notar que a heurística termina sempre com uma solução válida, o que não acontece necessariamente quando existem recursos não renováveis tal como veremos mais adiante.

O exemplo seguinte ilustra os passos associados à execução da heurística de Boctor.

Exemplo 4.1

Considere a Figura 4.1 e a Tabela 4.1 em que se representa uma instância constituída por 8 actividades (as actividades 0 e 9 são actividades fictícias e correspondem respectivamente ao início e término do projecto), dois modos por actividade e apenas um tipo de recurso renovável (R_1). O nível máximo de recurso disponível por período é de 5 unidades.

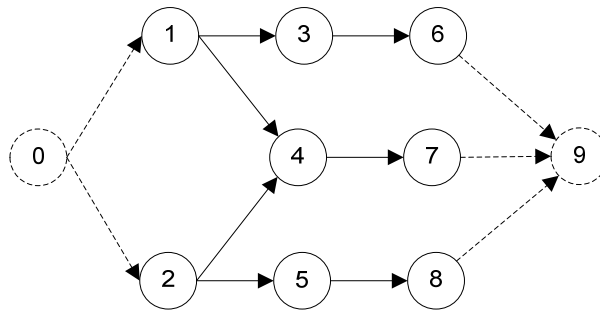


Figura 4.1 – Grafo com 8 actividades (exemplo 4.1)

Actividade	0	1		2		3		4		5		6		7		8		9
Modo	---	1	2	1	2	1	2	1	2	1	2	1	2	1	2	1	2	---
Duração	0	2	4	1	3	3	4	3	4	3	4	1	3	2	4	2	4	0
Recurso R_t	0	3	1	4	1	5	3	3	2	2	1	4	2	5	3	3	1	0

Tabela 4.1 – Instância com 8 actividades, 2 modos/actividade, 1 recurso renovável

Em cada instante de tempo em análise, utilizando os modos mais rápidos para cada actividade e através de uma passagem progressiva e outra regressiva na rede, determinam-se os tempos de término (EF e LF, respectivamente o instante de tempo mais cedo e mais tardio em que uma actividade pode ser terminada) e respectiva folga (F) de cada actividade. Os valores de EF e LF aparecem nas caixas junto aos nodos do grafo que representa o projecto. Os passos da heurística estão descritos de seguida, para cada instante de tempo em que é tomada uma decisão. O conjunto A representa o conjunto de actividades escalonáveis no instante de tempo considerado. O bloco não-dominado i de actividades é representado por C_i . A disponibilidade de recurso renovável i no instante de tempo t é designada por R_{it} .

. $t = 0$

$R_{1,0} = 5$ (quantidade disponível do recurso renovável 1 no instante 0);

$A = \{1, 2\}$ (actividades escalonáveis).

Blocos não-dominados:

$$C_1 = \{(1,1); (2,2)\};$$

$$C_2 = \{(1,2); (2,1)\}.$$

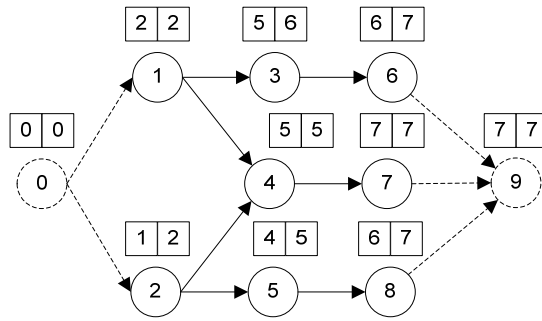


Figura 4.2 – Grafo para $t = 0$ (exemplo 4.1)

Temos assim que:

$$a_{2,2} = \max\{0, 3-1-1\} = \max\{0, 1\} = 1;$$

$$a_{1,2} = \max\{0, 4-2-0\} = \max\{0, 2\} = 2;$$

e

$$I_1 = \max\{0,1\} = 1;$$

$$I_2 = \max\{0,2\} = 2.$$

Dado que $I_1 < I_2$ o bloco a escalonar será o bloco C_1 . O escalonamento parcial resultante é o seguinte:



Figura 4.3 – Escalonamento parcial para $t = 0$ (exemplo 4.1)

. $t = 2$

$$R_{1,2} = 4; \quad A = \{3\}.$$

Blocos não-dominados: $C_1 = \{(3,2)\}$.

Como existe apenas um bloco não-dominado, o bloco C_1 será então escalonado, obtendo se:

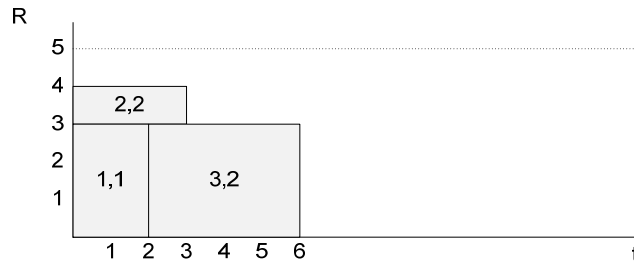


Figura 4.4 – Escalonamento parcial para $t = 2$ (exemplo 4.1)

. $t = 3$

$$R_{I,3} = 2; \quad A = \{4,5\}.$$

Blocos não-dominados: $C_I = \{(4,2)\}$.

Mais uma vez, como existe apenas um bloco não-dominado, o bloco C_I será então escalonado, obtendo-se:

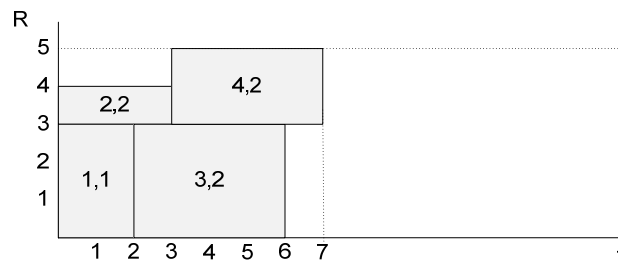


Figura 4.5 – Escalonamento parcial para $t = 3$ (exemplo 4.1)

. $t = 6$

$$R_{I,6} = 3; \quad A = \{5, 6\}.$$

Blocos não-dominados:

$$C_1 = \{(5,1)\};$$

$$C_2 = \{(5,2);(6,2)\}.$$

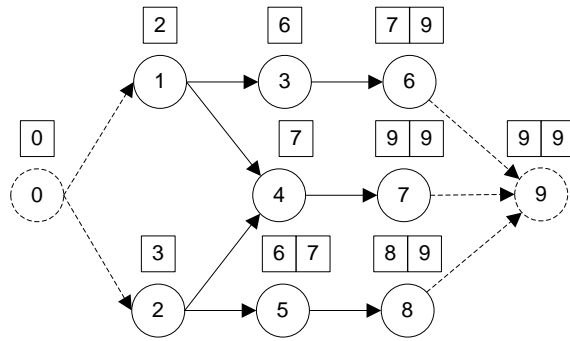


Figura 4.6 – Grafo para $t = 6$ (exemplo 4.1)

Temos assim que:

$$b_6 = \max \{0, \min(9-6+1-1-2, 7-6+3-1-2)\} = \max \{0, \min(1,1)\} = 1;$$

$$a_{5,2} = \max \{0, 4-3-1\} = \max \{0, 0\} = 0;$$

$$a_{6,2} = \max \{0, 3-1-2\} = \max \{0, 0\} = 0.$$

e:

$$I_1 = \max \{0,1\} = 1;$$

$$I_2 = \max \{0,0\} = 0.$$

Dado que $I_2 < I_1$ o bloco a escalonar será o bloco C_2 , obtendo-se:

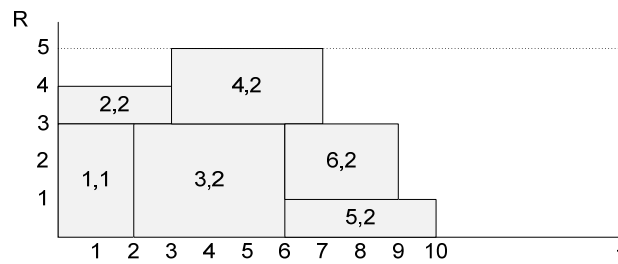


Figura 4.7 – Escalonamento parcial para $t = 6$ (exemplo 4.1)

$t = 7$

$$R_{1,7} = 2 \quad A = \{7\}$$

Blocos não-dominados: $C_1 = \emptyset$

. $t = 9$

$R_{I,9} = 4; \quad A = \{7\}.$

Blocos não-dominados: $C_I = \{(7,2)\}.$

Mais uma vez, como existe apenas um bloco não-dominado, o bloco C_I será então escalonado, obtendo-se:

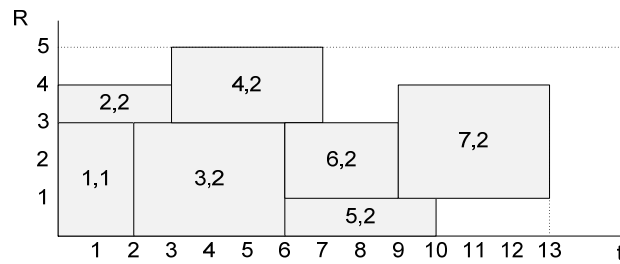


Figura 4.8 – Escalonamento parcial para $t = 9$ (exemplo 4.1)

. $t = 10$

$R_{I,10} = 2; \quad A = \{8\}.$

Blocos não-dominados: $C_I = \{(8,2)\}$

Finalmente, para a última actividade como existe apenas um bloco não-dominado, o bloco C_I será então escalonado, obtendo-se a solução final do problema:

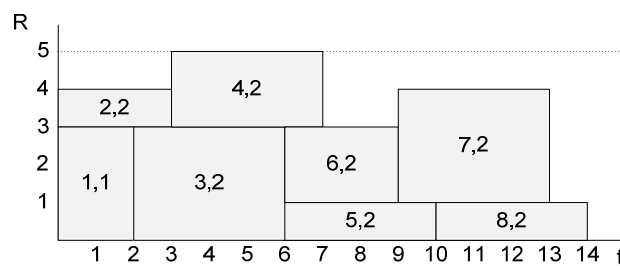


Figura 4.9 – Escalonamento final (exemplo 4.1)

Pelo que seriam necessárias 14 unidades de tempo para realizar todas as actividades deste projecto.

No domínio dos problemas de escalonamento de projectos, a heurística desenvolvida por Boctor revela-se interessante no sentido em que assenta numa forma diferente de escalonar as actividades. Essa heurística tem contudo uma limitação óbvia: não

considera a existência de recursos não renováveis. O tratamento desse tipo de recursos está longe de ser trivial.

Com efeito, quando aplicada a instâncias em que existem recursos não renováveis, a heurística de Boctor pode simplesmente falhar em encontrar uma solução válida. O escalonamento das actividades deve ter em conta esse factor, e contemplar procedimentos que permitam desfazer escalonamentos já realizados, re-escalando as actividades afectadas em modos e instantes de tempo diferentes. Na secção seguinte, exploramos em detalhe uma alternativa ao método proposto por Boctor para instâncias onde se considera a existência de recursos não renováveis.

4.3 Extensão da heurística de Boctor

Ainda que numa primeira abordagem se possa pensar que contemplar a restrição relativa à existência de recursos não renováveis na heurística de Boctor seja imediato e simples, quando se aprofunda um pouco mais o estudo percebe-se rapidamente que tal não acontece. A consideração da restrição relativa à existência de recursos não renováveis obriga a que, em cada instante de tempo, se esteja a verificar que o escalonamento efectuado nesse instante permite às actividades que ainda não foram escalonadas que o possam vir a ser. Para tal, é necessário garantir a disponibilidade de recursos para as próximas actividades a escalonar, não esquecendo contudo o objectivo de optimização subjacente sempre que se procede a um escalonamento. De seguida, apresentamos as novidades introduzidas na heurística de Boctor de modo a ter em conta a existência de recursos não renováveis.

Quando existem recursos não renováveis, ao escalonar um conjunto de actividades no instante de tempo t , podemos estar a inviabilizar o uso de alguns modos para algumas actividades não escalonadas. Tipicamente, isto acontece nos últimos estágios do escalonamento, quando já faltam poucas actividades por escalonar, e os recursos não renováveis começam a escassear. Para uma actividade i não escalonável no instante de tempo t , podemos estar a induzir um atraso o qual pode ser medido através da seguinte expressão:

$$c_i = d_{ij'} - d_{i1} - F_i,$$

sendo d_{ij} , a duração do modo j' que é o modo mais rápido no qual pode ser realizado a actividade i se as actividades de S forem escalonadas no instante de tempo t . Essa medida refere-se apenas às actividades que não pertencem ao conjunto A , e para as quais ainda não estamos em medida de dizer quando poderão ser escalonadas dado existirem relações de precedência ainda não satisfeitas.

Para o cálculo do atraso induzido no projecto, se as actividades de S forem escalonadas no instante de tempo t , mantemos inalterada a expressão referente ao parâmetro $a_{ij(i)}$, mas introduzimos a seguinte alteração na expressão relativa ao parâmetro b_i :

$$b_i = \max \{0, \min_{j \in M_i'} (t_{ij} - t + d_{i,j} - d_{i,l} - F_j)\},$$

em que M_i' representa agora o subconjunto de modos que podem ser usados para escalonar a actividade i , tendo em consideração os níveis de recursos não renováveis disponíveis se forem escalonadas as actividades de S no instante de tempo t .

A medida do atraso I_S induzido no projecto se as actividades de S forem escalonadas no instante de tempo t , é dada agora pela seguinte expressão:

$$I_S = \max \{0, \max_{i \in S} (a_{ij(i)}), \max_{i \in A-S} (b_i), \max_{i \in I-A} (c_i)\},$$

sendo I o conjunto de actividades ainda não escalonadas, e que não se encontram em condições de o ser no instante de tempo t . Novamente, o conjunto de actividades a escalonar em t será aquele com menor valor I_S associado.

Como foi já referido anteriormente, o facto de se considerar a existência de recursos não renováveis implica que tenhamos de verificar constantemente se ainda é possível escalonar todas as actividades. Para poder ter em conta esse dado, definimos para cada actividade um vector $[N_1, N_2, \dots]$ com tantas entradas como o número de recursos não renováveis existentes. Cada entrada representa o nível mínimo de cada recurso não renovável necessário para a actividade e as suas sucessoras poderem ser executadas. Esses valores são calculados para cada recurso não renovável tendo em conta os valores mínimos de consumo de recursos dos modos para os respectivos recursos. À última

actividade (fictícia) está associado o vector nulo $[0, 0, \dots]$. Esses vectores são calculados da mesma forma que os instantes de tempo mais cedo e mais tarde em que pode começar uma actividade, isto é, usando um procedimento de recorrência para trás.

Claramente, ao escolher apenas os valores mínimos de consumo entre os modos para cada recurso não renovável, estamos a perder informação. Dito de outra forma, mesmo que o escalonamento respeite os valores contidos nos vectores descritos acima, isso não garante por si só que o escalonamento seja todo feito com sucesso. Para reduzir a probabilidade disso acontecer, introduzimos uma nova medida: o valor acumulado mínimo q_i de recurso não renovável necessário à execução de uma actividade i . A actividade fictícia final tem associado o valor 0. Para cada actividade, somam-se os valores de consumo de cada modo e escolhe-se a menor soma entre esses valores. De seguida, usando um processo de recorrência para trás semelhante aos anteriormente descritos, calculam-se os valores mínimos acumulados para cada actividade.

É importante notar que os dois tipos de medida apresentados acima são usados aquando da construção dos blocos válidos e não dominados de actividades. Com efeito, um bloco que comprometa a execução do projecto é simplesmente rejeitado do conjunto de blocos escalonáveis no instante de tempo t . Para validar um bloco, compara-se o nível de recursos não renováveis que sobraria se se escalonasse esse bloco no instante de tempo t com os níveis mínimos de recursos necessários às actividades que ainda não foram escalonadas. Entre essas últimas, escolhem-se aquelas que serão escalonadas em primeiro lugar atendendo às restrições de precedência. O exemplo que apresentamos mais abaixo ilustra em detalhe esses passos.

Quando se consideram essas duas medidas, a probabilidade de o escalonamento falhar reduz-se substancialmente. As experiências computacionais realizadas no âmbito desta dissertação confirmam essa afirmação. Contudo, a probabilidade de uma falha ocorrer não desaparece completamente. Quando isso acontece (o escalonamento não pode ser feito até ao fim por ausência de recursos), recorremos ao chamado *backplanning*.

O *backplanning* é necessário sempre que existam actividades por escalonar, e não existe disponibilidade de recursos não renováveis que permita a execução dessas mesmas actividades em qualquer um dos modos existentes para a sua realização. Nessas

condições, o projecto não pode simplesmente ser concluído. No âmbito do nosso estudo, definimos o seguinte processo de *backplanning*:

- Para a última actividade i escalonada, analisamos o modo $j(i)$ em que foi escalonada:
 - se essa actividade tiver sido escalonada num modo $j(i) < M_i$, ou seja, num modo diferente do mais lento, então, para essa actividade, removem-se os modos compreendidos entre o modo mais rápido e o modo $j(i)$, no qual a actividade tinha sido escalonada;
 - se a actividade tiver sido escalonada no modo mais lento, ou seja, $j(i) = M_i$, então passamos à análise de uma outra actividade escalonada no mesmo instante de tempo t e que não tenha sido ainda analisada, ou então a uma outra actividade escalonada imediatamente antes da actividade i . O processo repete-se para a nova actividade escolhida;
 - O procedimento repete-se até que seja encontrada uma actividade que não tenha sido escalonada no seu modo mais lento. Designamos por t_i o instante de tempo em que começa essa actividade;
- Uma vez determinada a actividade na qual se irão retirar os modos mais rápidos até $j(i)$, anulamos o escalonamento feito do instante de tempo t_i para a frente, e repetimos a heurística a partir desse mesmo instante de tempo t_i .

O processo de *backplanning* falha quando todas as actividades tiverem sido escalonadas nos seus modos mais lentos, mas mesmo assim a disponibilidade de recursos não permite que o projecto seja devidamente concluído. Como veremos mais adiante, essa situação ocorre muitas poucas vezes.

Exemplo 4.2

Considere a Figura 4.10 e a Tabela 4.2 em que se representa uma instância semelhante àquela usada no Exemplo 4.1 constituída por 8 actividades, dois modos por actividade, apenas um tipo de recurso renovável (R_I) cujo nível máximo de recurso disponível por

período é de 5 unidades e ainda dois recursos não renováveis (NR_1 e NR_2), sendo de 15 unidades a quantidade total disponível de cada um desses recursos.

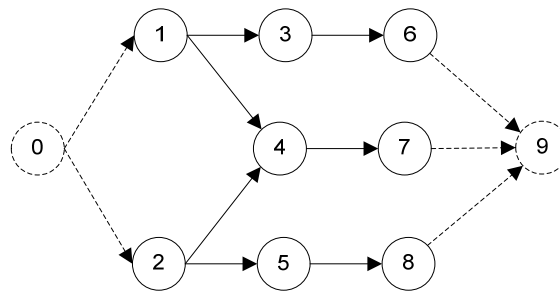


Figura 4.10 – Grafo com 8 actividades (exemplo 4.2)

Actividade	0	1		2		3		4		5		6		7		8		9
Modo	---	1	2	1	2	1	2	1	2	1	2	1	2	1	2	1	2	---
Duração	0	2	4	1	3	3	4	3	4	3	4	1	3	2	4	2	4	0
Recurso R_1	0	3	1	4	1	5	3	3	2	2	1	4	2	5	3	3	1	0
Recurso N_1	0	3	2	1	1	2	1	3	7	2	1	0	0	4	4	7	1	0
Recurso N_2	0	2	0	1	0	0	0	2	0	3	1	2	10	4	3	2	1	0

Tabela 4.2 – Instância com 8 actividades, 2 modos/actividade, 1 recurso renovável e 2 recursos não renováveis

Na aplicação desta nova heurística, começa-se por se calcular para cada actividade, os valores mínimos de cada recurso não renovável $[N_1, N_2]$ e a quantidade mínima agregada de recursos não renováveis q_i . Essas quantidades mantêm-se constantes ao longo do processo, pelo que são calculados uma única vez. Os valores obtidos para essas medidas são obtidos em vários passos descritos a seguir.

1.º Passo

De forma a posteriormente se poderem calcular as valores mínimos $[N_1, N_2]$ para cada actividade, determina-se primeiro a quantidade mínima de cada recurso não renovável necessária à execução de cada actividade. Para cada actividade e para cada recurso é determinado o valor mínimo de recurso de entre todos os modos possíveis para executar a actividade.

Actividade	Nível mínimo de recurso NR ₁ consumido	Nível mínimo de recurso NR ₂ consumido
1	2	0
2	1	0
3	1	0
4	3	0
5	1	1
6	0	2
7	4	3
8	1	1

Tabela 4.3 – Níveis mínimos de recursos não renováveis gastos/actividade

2.º Passo

De forma a posteriormente se poder calcular a quantidade q_i para cada actividade i , somam-se para cada modo de cada actividade as quantidades de cada recurso e escolhe-se para cada actividade o valor mínimo.

Actividade	Modo	Recurso NR ₁	Recurso NR ₂	Σ	Min
1	1	3	2	5	2
	2	2	0	2	
2	1	1	1	2	1
	2	1	0	1	
3	1	2	0	2	1
	2	1	0	1	
4	1	3	2	5	5
	2	7	0	7	
5	1	2	3	5	2
	2	1	1	2	
6	1	0	2	2	2
	2	0	10	10	
7	1	4	4	8	7
	2	4	3	7	
8	1	7	2	9	2
	2	1	1	2	

Tabela 4.4 – Valores mínimos de recursos NR₁ e NR₂ consumidos/actividade

Após a execução de cada um destes passos, pode-se então determinar, através de um processo de recorrência para trás, as janelas de níveis mínimos de cada recurso não renovável $[N_1, N_2]$ e a quantidade de recurso agregado q_i para cada actividade i . A Figura 4.11 mostra o grafo com esses valores para a instância deste exemplo.

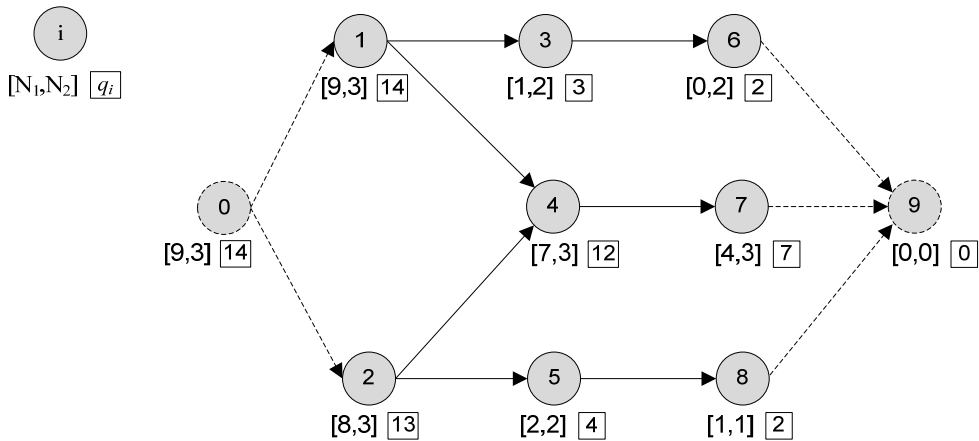


Figura 4.11 – Grafo com valores de recursos mínimos $[N_1, N_2]$ e quantidade agregada q_i

Na determinação da quantidade mínima q_i de uma actividade com duas ou mais actividades imediatamente sucessoras, alerta-se para o facto de não se poder somar os valores dos intervalos das actividades imediatamente sucessoras, tomando-se antes o maior desses valores, visto que aquando do escalonamento parcial não se sabe se os caminhos do nodo em causa para a frente serão divergentes ou se por ventura voltarão a unir-se num dado nodo.

Observando a Figura 4.11 é possível afirmar, por exemplo, que no instante em que se escalonar a actividade 1, tem que haver no mínimo 9 unidades do recurso não renovável NR_1 , 3 unidades de recurso não renovável NR_2 e que a soma de ambos os recursos não renováveis terá de ser no mínimo de 14 unidades, sob pena de o projecto não poder ser escalonado com sucesso por falta de recursos não renováveis.

Após se terem determinado as janelas de níveis de recursos mínimos $[N_1, N_2]$ e a quantidade mínima q_i em cada actividade, a heurística passa ao escalonamento das actividades, sendo que, à semelhança da heurística de Boctor, para cada instante de tempo em análise são calculadas as janelas de tempo relativas aos instantes de término

mais cedo e de término mais tarde [EF, LF] de cada actividade. Temos assim os escalonamentos parciais descritos de seguida.

$t = 0$

$R_{1,0} = 5$ (quantidade disponível do recurso renovável 1 no instante $t = 0$);

$NR_{1,0} = 15$ (quantidade disponível do recurso não renovável 1 no instante $t = 0$);

$NR_{2,0} = 15$ (quantidade disponível do recurso não renovável 2 no instante $t = 0$);

$A = \{1, 2\}$ (actividades escalonáveis no instante $t = 0$).

A forma de determinação dos blocos não-dominados é quase semelhante à heurística de Boctor, acrescentando-se contudo as regras de validação para os recursos não renováveis. Para tal executam-se os passos seguintes.

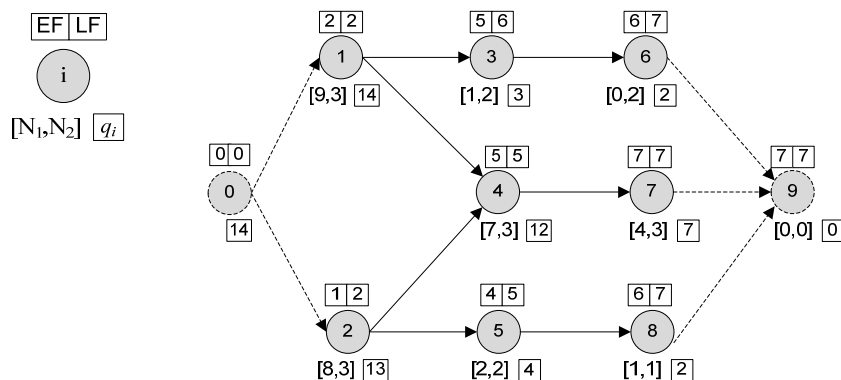


Figura 4.12 – Grafo para $t = 0$ (exemplo 4.2)

a. Blocos não-dominados

$$C_1 = \{(1,1) ; (2,2)\};$$

$$C_2 = \{(1,2) ; (2,1)\}.$$

b. Análise dos recursos não renováveis

Caso se opte por escalonar o bloco C_1 , ter-se-ia após o escalonamento os seguintes valores de recursos não renováveis disponíveis para as restantes actividades:

$$[11,13]$$

sendo que a quantidade total de recursos disponíveis seria:

$$11 + 13 = 24$$

Comparando com os níveis mínimos exigidos nas actividades imediatamente sucessoras, temos que:

$$[11,13] > [1,2] \quad (\text{actividade 3})$$

$$[11,13] > [7,3] \quad (\text{actividade 4})$$

$$[11,13] > [2,2] \quad (\text{actividade 5})$$

Para simplificar a apresentação, consideramos que $[a,b] > [c,d]$ se $a > b$ e $c > d$.

Relativamente aos valores q_i , chegamos às seguintes conclusões:

$$24 > 3 \quad (\text{actividade 3})$$

$$24 > 12 \quad (\text{actividade 4})$$

$$24 > 4 \quad (\text{actividade 5})$$

Pelo que o bloco C_1 é considerado válido.

Procedendo de forma análoga para o bloco C_2 , obtêm-se as seguintes relações:

$$[12,14] > [1,2] \quad (\text{actividade 3})$$

$$[12,14] > [7,3] \quad (\text{actividade 4})$$

$$[12,14] > [2,2] \quad (\text{actividade 5})$$

$$26 > 3 \quad (\text{actividade 3})$$

$$26 > 12 \quad (\text{actividade 4})$$

$$26 > 4 \quad (\text{actividade 5})$$

Pelo que o bloco C_2 é considerado válido.

c. Blocos não-dominados e válidos

$$C_1 = \{(1,1); (2,2)\};$$

$$C_2 = \{(1,2); (2,1)\}.$$

Após a determinação dos blocos não-dominados e válidos, prossegue-se com o cálculo dos vários coeficientes, $a_{ij(i)}$, b_i e c_i , de forma a determinar-se o menor valor de I_S .

$$a_{11} = \max\{0, 2-2-0\} = \max\{0, 0\} = 0;$$

$$a_{22} = \max\{0, 3-1-1\} = \max\{0, 1\} = 1;$$

$$a_{12} = \max\{0, 4-2-0\} = \max\{0, 2\} = 2;$$

$$a_{21} = \max\{0, 1-1-1\} = \max\{0, -1\} = 0;$$

$$c_i \leq 0, \text{ para } \forall i \in I-A.$$

Temos assim que:

$$I_1 = \max\{0, a_{11}, a_{22}, c_i\} = \max\{0, 0, 1, 0\} = 1;$$

$$I_2 = \max\{0, a_{12}, a_{21}, c_i\} = \max\{0, 2, 0, 0\} = 2.$$

Dado que $I_1 < I_2$ o bloco a escalonar será o bloco C_1 , obtendo-se:



Figura 4.13 – Escalonamento parcial para $t = 0$ (exemplo 4.2)

. $t = 2$

$$R_{1,2} = 4; \quad NR_{1,2} = 11; \quad NR_{2,2} = 13; \quad A = \{3\}.$$

Blocos não-dominados: $C_I = \{(3,2)\}$.

Teste aos níveis mínimos de recursos não renováveis:

$[10,13] > [7,3]$	(actividade 4)
$[10,13] > [2,2]$	(actividade 5)
$[10,13] > [0,2]$	(actividade 6)
$23 > 12$	(actividade 4)
$23 > 4$	(actividade 5)
$23 > 2$	(actividade 6)

Blocos não-dominados e válidos: $C_I = \{(3,2)\}$.

Pelo que dado que só existe um bloco, o escalonamento é imediato.

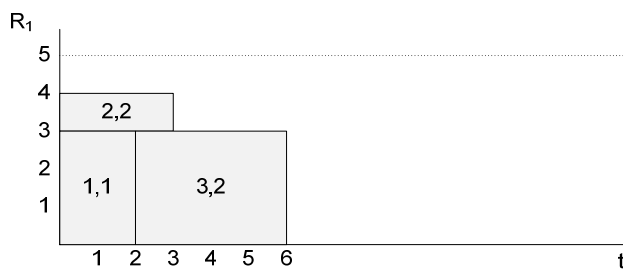


Figura 4.14 – Escalonamento parcial para $t = 2$ (exemplo 4.2)

. $t = 3$

$$R_{1,3} = 2; \quad NR_{1,3} = 10; \quad NR_{2,3} = 13; \quad A = \{4,5\}.$$

Blocos não-dominados:

$$C_I = \{(4,2)\};$$

$$C_2 = \{(5,1)\}.$$

Teste aos níveis mínimos de recursos não renováveis para o bloco C_I :

$$[3,13] > [2,2] \quad (\text{actividade 5})$$

$$[3,13] > [0,2] \quad (\text{actividade 6})$$

$$\mathbf{[3,13] > [4,3] \quad (\text{falso}) \quad (\text{actividade 7})}$$

Como falha a validação para o recurso não renovável NR_1 na quantidade mínima necessária da actividade 7, o bloco C_1 é considerado inválido.

Teste aos níveis mínimos de recursos não renováveis para o bloco C_2 :

$$[8,10] > [2,2] \quad (\text{actividade 4})$$

$$[8,10] > [0,2] \quad (\text{actividade 6})$$

$$[8,10] > [1,1] \quad (\text{actividade 8})$$

$$18 > 12 \quad (\text{actividade 4})$$

$$18 > 2 \quad (\text{actividade 6})$$

$$18 > 2 \quad (\text{actividade 8})$$

Blocos não-dominados e válidos: $C_2 = \{(5,1)\}$.

Pelo que o bloco C_2 é o bloco escolhido para escalonar, obtendo-se:

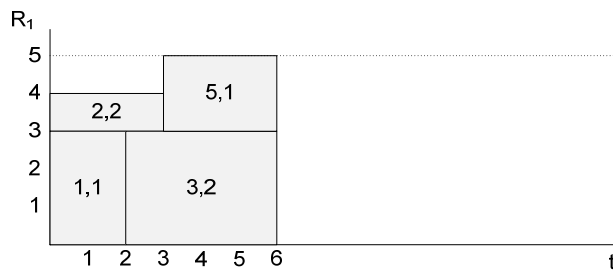


Figura 4.15 – Escalonamento parcial para $t = 3$ (exemplo 4.2)

. $t = 6$

$$R_{1,6} = 5; \quad NR_{1,6} = 8; \quad NR_{2,6} = 10; \quad A = \{4,6,8\}.$$

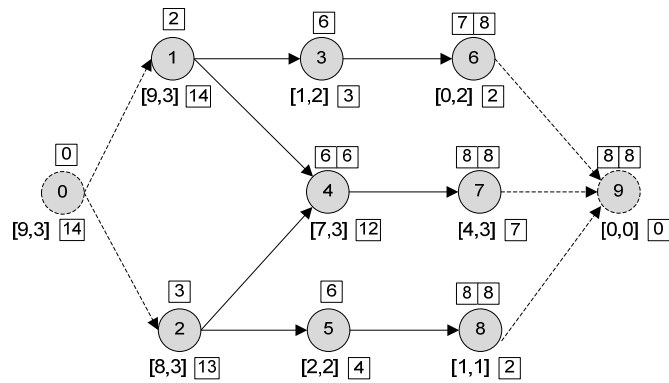


Figura 4.16 – Grafo para $t = 6$ (exemplo 4.2)

Blocos não-dominados:

$$C_1 = \{(4,1) ; (8,2)\};$$

$$C_2 = \{(6,1) ; (8,2)\};$$

$$C_3 = \{(8,1)\}.$$

Teste aos níveis mínimos de recursos não renováveis para o bloco C_1 :

$$[4,7] > [0,2] \quad (\text{atividade } 6)$$

$$[4,7] > [4,3] \quad (\text{atividade } 7)$$

$$11 > 2 \quad (\text{atividade } 6)$$

$$11 > 7 \quad (\text{atividade } 7)$$

Teste aos níveis mínimos de recursos não renováveis para o bloco C_2 :

$$[7,7] > [7,3] \quad (\text{atividade } 4)$$

$$14 > 12 \quad (\text{atividade } 4)$$

Teste aos níveis mínimos de recursos não renováveis para o bloco C_3 :

$$[1,8] > [7,3] \quad (\text{falso}) \quad (\text{atividade } 4)$$

Como falha a validação para o recurso não renovável N_1 na quantidade mínima necessária da atividade 4, o bloco C_3 é considerado inválido.

Blocos não-dominados e válidos:

$$C_1 = \{(4,1);(8,2)\}$$

$$C_2 = \{(6,1);(8,2)\}$$

Cálculo dos coeficientes:

$$a_{41} = \max\{0, 3-3-0\} = \max\{0, 0\} = 0;$$

$$a_{82} = \max\{0, 4-2-0\} = \max\{0, 2\} = 2;$$

$$a_{61} = \max\{0, 1-1-1\} = \max\{0, -1\} = 0$$

$$b_4 = \max\{0, \min(7-6+3-3-0, 7-6+4-3-0)\} = \max\{0, \min(1,2)\} = 1;$$

$$b_6 = \max\{0, \min(9-6+1-1-1, 9-6+3-1-1)\} = \max\{0, \min(2,4)\} = 2;$$

$$c_i \leq 0, \text{ para } \forall i \in I-A.$$

donde:

$$I_1 = \max\{0, a_{41}, a_{82}, b_6, c_7\} = \max\{0, 0, 2, 2, 0\} = 2;$$

$$I_2 = \max\{0, a_{61}, a_{82}, b_4, c_7\} = \max\{0, 0, 2, 1, 0\} = 2.$$

Dado que $I_1 = I_2$, o atraso introduzido na conclusão do projecto será o mesmo independentemente da escolha, pelo que para efeitos de ilustração optou-se por escalonar o bloco C_2 , obtendo-se:

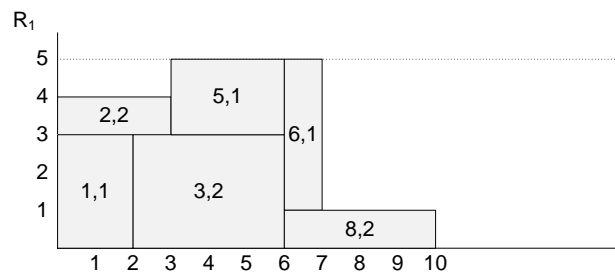


Figura 4.17 – Escalonamento parcial para $t = 6$ (exemplo 4.2)

. $t = 7$

$$R_{1,7} = 4; \quad NR_{1,7} = 7; \quad NR_{2,7} = 7; \quad A = \{4\}.$$

Blocos não-dominados: $C_I = \{(4,1)\}$.

Teste aos níveis mínimos de recursos não renováveis para o bloco C_I :

$$[4,5] \geq [4,3] \quad (\text{actividade 7})$$

$$9 > 7 \quad (\text{actividade 7})$$

Blocos não-dominados e válidos: $C_I = \{(4,1)\}$.

Mais uma vez, dado que só existe um bloco, o escalonamento é imediato.

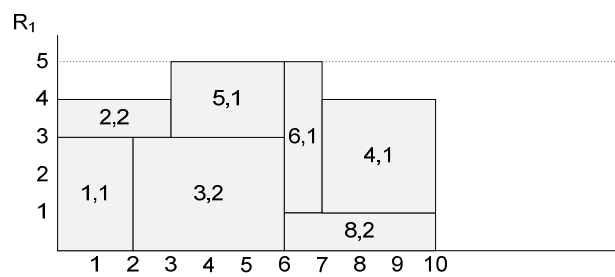


Figura 4. 18 – Escalonamento parcial para $t = 7$ (exemplo 4.2)

. $t = 10$

$$R_{1,10} = 5; \quad NR_{1,10} = 4; \quad NR_{2,10} = 5; \quad A = \{7\}.$$

Blocos não-dominados: $C_I = \{(7,1)\}$.

Teste aos níveis mínimos de recursos não renováveis:

$$[0,1] \geq [0,0] \quad (\text{actividade 9})$$

$$1 > 0 \quad (\text{actividade 9})$$

Blocos não-dominados e válidos: $C_I = \{(7,1)\}$.

Finalizando-se então o escalonamento de todas as actividades do projecto.

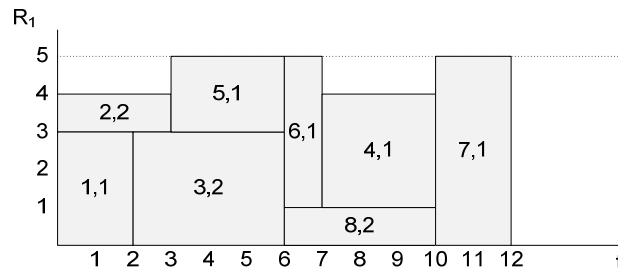


Figura 4. 19 – Escalonamento final (exemplo 4.2)

4.4 Implementação

A heurística para o MRCPSP com recursos não renováveis descrita acima foi implementada de raiz na linguagem de programação C++. A aplicação foi desenvolvida no ambiente Microsoft Visual Studio 6.0 que, entre outras linguagens, suporta a linguagem de programação Visual C++. A nossa implementação consiste antes num conjunto de classes e métodos (funções) que implementam as diversas operações relacionadas com a gestão de dados (leitura de dados e impressão de resultados) e com as rotinas e subrotinas definidas ou usadas pela própria heurística.

Um dos aspectos importantes do código que desenvolvemos prende-se com as estruturas de dados. Como foi referido ao longo desta dissertação, uma instância do MRCPSP é definida com base num conjunto importante de informação. Muita dessa informação precisa de ser acedida várias vezes no decurso da heurística, e por essa razão, tentámos na medida do possível usar estruturas de dados que facilitassem esses acessos. As relações de precedência, por exemplo, são armazenadas em matrizes quadradas (mesmo número de linhas e colunas) indexadas usando o mesmo índice que as actividades. Essa matriz contém valores binários que indicam se existe ou não uma relação entre duas actividades. A informação referente aos modos de execução das actividades é guardada num vector em que cada posição i aponta para uma matriz com a informação completa sobre a actividade i . De notar ainda a complexidade ligada à gestão da informação associada aos blocos não dominados de actividade. Essa informação é gerada dinamicamente, e como tal, as estruturas são também elas dinâmicas. O espaço é alocado à medida que as combinações actividade-modo e os próprios blocos vão sendo criados.

Uma das funções mais elementares da aplicação consiste na extracção de dados a partir de um ficheiro de texto que descreve a instância, e sua correspondente conversão nas estruturas de dados referidas acima. O formato de referência é aquele usado na PSPLIB [37]. Para além disso, a aplicação implementa as rotinas necessárias ao cálculo dos instantes de tempo mais cedo e mais tardio em que uma actividade deve ser terminada, assim como as rotinas de cálculo dos valores mínimos de recursos não renováveis necessários à execução de uma actividade e suas sucessoras. Os grandes passos da heurística e o processo de construção de blocos não dominados foram implementados em rotinas isoladas. No total, a aplicação tem cerca de 1500 linhas de código.

4.5 Conclusões

As heurísticas constituem um método simples e expedito de encontrar rapidamente soluções para problemas complexos como é o caso do MRCPSP.

A maioria das heurísticas conhecidas para o MRCPSP baseiam-se em métodos construtivos utilizando regras de prioridade para a selecção das actividades e dos modos, construindo escalonamentos parciais ao longo do tempo até se atingir o escalonamento final do projecto. Normalmente o escalonamento é feito actividade a actividade, ou seja, em cada iteração apenas se escalona uma actividade.

Num estudo efectuado por Boctor [8] este constatou, através de testes computacionais em que aplicou 21 combinações constituídas a partir de 7 regras de prioridade para a selecção das actividades e 3 regras de prioridade para a selecção dos modos, todas amplamente documentadas na literatura, que a combinação da regra SFM (modo com duração mais curta) de selecção de modos com a regra MIN SLK (actividade com menor valor de folga) para a selecção das actividades era a que conduzia, normalmente a melhores resultados.

Em 1996, Boctor [9] apresentou contudo uma heurística para problemas MRCPSP que se revelou bastante interessante, com bons resultados à época e que apresentava um conceito totalmente diferente das demais, onde as actividades em vez de serem

escalonadas de forma individual, eram escalonadas por blocos. No entanto esta heurística tem um grande constrangimento, visto que considera apenas a existência de recursos renováveis, não sendo admitida a existência de recursos não renováveis, o que penaliza gravemente o seu universo de aplicação.

Assim, no âmbito deste trabalho, tendo por base essa heurística de Boctor, foi desenvolvida uma extensão da mesma por forma a que ela considere a existência de recursos não renováveis. A execução desta extensão não é no entanto trivial, visto que é necessário estar constantemente a avaliar se os níveis de cada recurso não renovável permitem ainda a execução de todas as actividades que constituem o projecto.

Em relação à heurística de Boctor foram introduzidos três novos parâmetros que têm por função a referida avaliação dos níveis de recursos não renováveis. O primeiro parâmetro consiste num vector $[N_1, N_2, \dots]$ correspondente ao nível mínimo de cada recurso não renovável (N_1, N_2, \dots) que deve ser garantido em cada actividade de forma a ainda ser possível completar o projecto. O segundo parâmetro refere-se à quantidade acumulada mínima q_i de recursos não renováveis que deve ser garantida em cada actividade, que caso não se verifique impede a conclusão do escalonamento por falta de recursos. O terceiro parâmetro introduzido foi uma medida de atraso c_i para as actividades que ainda não foram escalonadas e que não o podem ser no instante t em análise.

Na nova heurística foi ainda introduzido um mecanismo de *backplanning* para melhorar a sua performance. Assim, se em determinado instante do escalonamento se verificar que já não é possível escalonar todas as actividades por falta de recurso, procede-se a um desescalonamento regressivo das últimas actividades escalonadas, à remoção de modos de execução em algumas actividades e posteriormente retoma-se o escalonamento das actividades.

No próximo capítulo são apresentados os testes computacionais que se realizaram para avaliar o desempenho desta nova heurística.

Capítulo 5

Experiências computacionais

5.1 Introdução

Os testes computacionais que apresentamos neste capítulo debruçam-se sobre os dois tipos de abordagens estudados nesta dissertação: a procura de limites inferiores para o problema usando decomposições do modelo original de Talbot [49], e o cálculo de limites superiores usando uma heurística construtiva para o MRCPSP com recursos não renováveis inspirada na heurística de Boctor [9].

A revisão de literatura efectuada permitiu-nos constatar que os investigadores que estudaram o MRCPSP testaram as suas abordagens usando maioritariamente instâncias próprias. Esta situação causa dificuldades sempre que alguém quer comparar os seus resultados com aqueles obtidos usando outras abordagens. No sentido de tentar estabelecer uma base de trabalho alargada e disponível para qualquer utilizador, foi criada e disponibilizada por Kolisch e Sprecher [37] uma biblioteca de instâncias, designada de PSPLIB, que se encontra disponível na Internet em <http://129.187.106.231/psplib>. Essa biblioteca contempla diversos problemas de escalonamento, e possui alguma informação relevante nomeadamente no que toca à disponibilidade de soluções óptimas para cada instância. Os testes que realizámos no

âmbito desta dissertação tiveram como suporte as instâncias constantes na PSPLIB, e em particular aquelas relativas ao MRCPSP.

Na secção da PSPLIB relativa ao MRCPSP, encontramos vários conjuntos de instâncias (cada conjunto tem um número elevado de instâncias, sempre superior a 300 por conjunto). Essas instâncias são obtidas variando alguns dos seus parâmetros. Podemos agrupar essas instâncias como se segue.

a. Conjuntos de instâncias onde varia o número de actividades:

1. j10 – 10 actividades, 3 modos/act., 2 rec. renováveis, 2 rec. não renováveis;
2. j12 – 12 actividades, 3 modos/act., 2 rec. renováveis, 2 rec. não renováveis;
3. j14 – 14 actividades, 3 modos/act., 2 rec. renováveis, 2 rec. não renováveis;
4. j16 – 16 actividades, 3 modos/act., 2 rec. renováveis, 2 rec. não renováveis;
5. j18 – 18 actividades, 3 modos/act., 2 rec. renováveis, 2 rec. não renováveis;
6. j20 – 20 actividades, 3 modos/act., 2 rec. renováveis, 2 rec. não renováveis;
7. j30 – 30 actividades, 3 modos/act., 2 rec. renováveis, 2 rec. não renováveis.

b. Conjuntos de instâncias onde se varia o número de modos:

1. m1 – 16 actividades, 1 modo/act., 2 rec. renováveis, 2 rec. não renováveis;
2. m2 – 16 actividades, 2 modos/ act., 2 rec. renováveis, 2 rec. não renováveis;
3. m4 – 16 actividades, 4 modos/ act., 2 rec. renováveis, 2 rec. não renováveis;
4. m5 – 16 actividades, 5 modos/ act., 2 rec. renováveis, 2 rec. não renováveis.

Nota: o conjunto de instâncias “j16” complementa este grupo; corresponde aos mesmos problemas que os anteriores, mas com 3 modos/actividade.

c. Conjuntos de instâncias onde se varia o número de recursos renováveis:

1. r1 – 16 actividades, 3 modos/act., 1 rec. renovável, 2 rec. não renováveis;
2. r3 – 16 actividades, 3 modos/ act., 3 rec. renováveis, 2 rec. não renováveis;
3. r4 – 16 actividades, 3 modos/ act., 4 rec. renováveis, 2 rec. não renováveis;
4. r5 – 16 actividades, 3 modos/ act., 5 rec. renováveis, 2 rec. não renováveis.

Nota: novamente, o conjunto de instâncias “j16” complementa este grupo; corresponde aos mesmos problemas que os anteriores, mas com 2 recursos renováveis.

d. Conjuntos de instâncias onde se varia o número de recursos não renováveis:

1. n0 – 16 actividades, 3 modos/act., 2 rec. renovável, 0 rec. não renováveis;
2. n1 – 16 actividades, 3 modos/ act., 2 rec. renováveis, 1 rec. não renováveis;
3. n3 – 16 actividades, 3 modos/ act., 2 rec. renováveis, 3 rec. não renováveis.

Nota: o conjunto de instâncias “j16” complementa este grupo, visto que corresponde a ter a mesma situação dos problemas anteriores, mas com 2 recursos não renováveis.

Para cada uma das instâncias de cada conjunto são facultados os valores das soluções óptimas, com excepção do conjunto de instâncias “j30”. Nesse caso, são facultados apenas limites superiores obtidos através de uma heurística.

Nas secções seguintes, começa-se pela explicitação dos resultados relativos aos limites inferiores obtidos com as várias decomposições descritas no capítulo 3, seguem-se os resultados relativos aos limites superiores obtidos com a heurística apresentada no capítulo 4 e por fim faz-se uma análise e conclusão relativamente aos mesmos.

5.2 Limites inferiores para o MRCPSP

Nesta secção, apresentamos os resultados obtidos com as decomposições que descrevemos no Capítulo 3. Resolveram-se as relaxações lineares para diferentes instâncias da PSPLIB usando o software *ADDiNg* – Automatic Dantzig-Wolfe Decomposition for Integer Column Generation [4]. Os testes foram efectuados usando um PC equipado com um Pentium IV com 1,7GHz e 512Mb RAM.

Atendendo aos tempos de computação necessários para resolver cada uma das decomposições (e em especial a primeira decomposição), restringimos a 10 o número de

instâncias resolvidas por cada conjunto da PSPLIB (grupo de instâncias do MRCPSP). Foi imposto um limite máximo de tempo para a resolução de 480 segundos.

Apresentamos de seguida os resultados organizados de acordo com o agrupamento de instâncias definido acima. Indicamos o valor das soluções óptimas obtidas para cada instância. Na última linha de cada tabela, indicamos o número de instâncias para as quais o intervalo de integralidade é nulo.

a. Conjuntos de instâncias onde varia o número de actividades

• Conjunto de instâncias *j10*

Instâncias	Ótimo	1ª Dec	2ª Dec	3ª Dec	Rel. Lin
j1010_1.mm	17	17	17	17	17
j1011_1.mm	20	20	20	20	20
j1012_1.mm	15	15	15	15	15
j1013_1.mm	24	24	24	24	24
j1014_1.mm	16	16	14.301	14.301	14.301
j1015_1.mm	13	13	12	12	12
j1016_1.mm	13	13	13	13	13
j1018_1.mm	15	15	12	12	12
j1019_1.mm	13	13	13	13	13
j1020_1.mm	12	12	12	12	12
Coinc. Ótimo		100%	70%	70%	70%

Tabela 5.1 – Limites inferiores *j10*

• Conjunto de instâncias *j12*

Instâncias	Ótimo	1ª Dec	2ª Dec	3ª Dec	Rel. Lin
j1210_1.mm	20	20	18	18	18
j1211_1.mm	17	17	17	17	17
j1212_1.mm	30	30	30	30	30
j1214_1.mm	16	16	16	16	16
j1215_1.mm	21	19.2	19	19	19
j1216_1.mm	17	17	16	16	16
j1217_1.mm	17	15.026	14.162	14.162	14.162
j1218_1.mm	25	22.556	22.170	22.170	22.170
j1219_1.mm	17	17	17	17	17
j1220_1.mm	17	17	17	17	17
Coinc. Ótimo		70%	50%	50%	50%

Tabela 5.2 – Limites inferiores *j12*

• Conjunto de instâncias j14

Instâncias	Ótimo	1ª Dec	2ª Dec	3ª Dec	Rel. Lin
j1410_1.mm	16	16	16	16	16
j1411_1.mm	23	23	21	21	21
j1412_1.mm	18	18	18	18	18
j1415_1.mm	25	25	25	25	25
j1416_1.mm	20	19	18	18	18
j1417_1.mm	26	19.2	18.541	19.5	18.541
j1418_1.mm	22	21	21	21	21
j1419_1.mm	16	16	16	16	16
j1420_1.mm	21	21	21	21	21
j1421_1.mm	27	21.387	21	21	21
Coinc. Ótimo		60%	50%	50%	50%

Tabela 5.3 – Limites inferiores j14

• Conjunto de instâncias j16

Instâncias	Ótimo	1ª Dec	2ª Dec	3ª Dec	Rel. Lin
j1610_1.mm	22	22	22	22	22
j1611_1.mm	25	25	24	24	24
j1612_1.mm	20	20	20	20	20
j1613_1.mm	30	-	23	23	23
j1614_1.mm	21	-	21	21	21
j1615_1.mm	23	23	23	23	23
j1616_1.mm	24	24	24	24	24
j1617_1.mm	19	17	17	17	17
j1618_1.mm	27	25	25	25	25
j1619_1.mm	22	22	22	22	22
Coinc. Ótimo		60%	60%	60%	60%

Tabela 5.4 – Limites inferiores j16

• Conjunto de instâncias j18

Instâncias	Ótimo	1ª Dec	2ª Dec	3ª Dec	Rel. Lin
j1810_1.mm	25	-	25	25	25
j1811_1.mm	24	24	24	24	24
j1812_1.mm	29	29	29	29	29
j1813_1.mm	31	-	26	26	26
j1814_1.mm	17	-	15.252	15.251	15.251
j1815_1.mm	19	19	19	19	19
j1816_1.mm	30	30	30	30	30
j1817_1.mm	25	23.615	23	23	23
j1818_1.mm	21	21	21	21	21
j1819_1.mm	20	20	20	20	20
Coinc. Ótimo		60%	70%	70%	70%

Tabela 5.5 – Limites inferiores j18

• Conjunto de instâncias j20

Instâncias	Optimo	1ª Dec	2ª Dec	3ª Dec	Rel. Lin
j2010_1.mm	18	18	18	18	18
j2011_1.mm	21	21	21	21	21
j2012_1.mm	21	-	18	18	18
j2014_1.mm	25	23.037	23	23	23
j2015_1.mm	25	25	25	25	25
j2016_1.mm	24	24	24	24	24
j2018_1.mm	30	30	30	30	30
j2019_1.mm	25	25	25	25	25
j2020_1.mm	25	25	25	25	25
j2021_1.mm	33	-	28	28	28
Coinc. Optimo		70%	70%	70%	70%

Tabela 5.6 – Limites inferiores j20

• Conjunto de instâncias j30

Instâncias	Heur.	1ª Dec	2ª Dec	3ª Dec	Rel. Lin
j3010_1.mm	26	-	26	26	26
j3011_1.mm	35	35	35	35	35
j3012_1.mm	29	29	29	29	29
j3013_1.mm	37	-	36	36	36
j3014_1.mm	37	-	37	37	37
j3015_1.mm	40	40	40	40	40
j3016_1.mm	27	27	27	27	27
j3017_1.mm	34	33	33	33	33
j3018_1.mm	28	28	28	28	28
j3019_1.mm	35	35	35	35	35
Coinc. Heur.		60%	80%	80%	80%

Tabela 5.7 – Limites inferiores j30

Os resultados obtidos permitem verificar que, em regra, a primeira decomposição produz limites inferiores de melhor qualidade. Contudo, dado que o subproblema é mais complexo, a sua resolução é bastante mais lenta que as outras, não conseguindo por vezes calcular uma solução válida dentro do tempo limite estabelecido.

De notar ainda que as outras duas decomposições dão sempre o mesmo resultado que a relaxação linear do modelo de Talbot para esse conjunto de instâncias.

b. Conjuntos de instâncias onde se varia o número de modos

• Conjunto de instâncias $m1$

Instâncias	Ótimo	1ª Dec	2ª Dec	3ª Dec	Rel. Lin
m110_1.mm	26	25	25	25	25
m111_1.mm	28	28	28	28	28
m112_1.mm	33	33	33	33	33
m113_1.mm	45	35.400	34.359	34.083	34.083
m114_1.mm	41	35.050	34	34	34
m115_1.mm	39	39	39	39	39
m116_1.mm	40	40	40	40	40
m117_1.mm	43	33.258	32.670	32.670	32.670
m118_1.mm	35	32.938	32.063	32.063	32.062
m119_1.mm	41	41	41	41	41
Coinc. Ótimo		50%	50%	50%	50%

Tabela 5.8 – Limites inferiores $m1$

• Conjunto de instâncias $m2$

Instâncias	Ótimo	1ª Dec	2ª Dec	3ª Dec	Rel. Lin
m210_1.mm	34	31.275	31	31	31
m211_1.mm	27	27	27	27	27
m212_1.mm	20	20	19	19	19
m213_1.mm	34	-	24.845	24.653	24.653
m214_1.mm	26	26	26	26	26
m215_1.mm	28	28	28	28	28
m216_1.mm	30	30	30	30	30
m217_1.mm	41	31.110	30.5	30.5	30.5
m218_1.mm	24	23.857	23	23	23
m219_1.mm	26	26	26	26	26
Coinc. Ótimo		60%	50%	50%	50%

Tabela 5.9 – Limites inferiores $m2$

• Conjunto de instâncias $m3$

Instâncias	Ótimo	1ª Dec	2ª Dec	3ª Dec	Rel. Lin
j1610_1.mm	22	22	22	22	22
j1611_1.mm	25	25	24	24	24
j1612_1.mm	20	20	20	20	20
j1613_1.mm	30	-	23	23	23
j1614_1.mm	21	-	21	21	21
j1615_1.mm	23	23	23	23	23
j1616_1.mm	24	24	24	24	24
j1617_1.mm	19	17	17	17	17
j1618_1.mm	27	25	25	25	25
j1619_1.mm	22	22	22	22	22
Coinc. Ótimo		60%	60%	60%	60%

Tabela 5.10 – Limites inferiores $m3$

• Conjunto de instâncias m4

Instâncias	Ótimo	1ª Dec	2ª Dec	3ª Dec	Rel. Lin
m410_1.mm	14	14	14	14	14
m411_1.mm	18	18	18	18	18
m412_1.mm	14	14	14	14	14
m414_1.mm	23	-	21	21	21
m415_1.mm	16	-	16	16	16
m416_1.mm	14	14	13.095	13.095	13.095
m417_1.mm	24	24	24	24	24
m418_1.mm	15	15	15	15	15
m419_1.mm	19	19	19	19	19
m420_1.mm	20	20	20	20	20
Coinc. Ótimo		80%	80%	80%	80%

Tabela 5.11 – Limites inferiores m4

• Conjunto de instâncias m5

Instâncias	Ótimo	1ª Dec	2ª Dec	3ª Dec	Rel. Lin
m510_1.mm	15	14.368	14	14	14
m511_1.mm	16	16	15	15	15
m512_1.mm	15	15	15	15	15
m513_1.mm	27	-	20.399	20.398	20.398
m514_1.mm	19	17.217	17	17	17
m515_1.mm	13	13	13	13	13
m516_1.mm	12	12	11	11	11
m518_1.mm	13	13	13	13	13
m519_1.mm	13	13	13	13	13
m520_1.mm	15	15	15	15	15
Coinc. Ótimo		70%	50%	50%	50%

Tabela 5.12 – Limites inferiores m5

Mais uma vez a primeira decomposição é aquela que consegue melhores resultados, sendo interessante verificar que mesmo com o aumento do número de modos por actividade a decomposição apresenta boa qualidade, parecendo não ser afectada com este incremento.

As restantes decomposições mantêm o comportamento verificado com as instâncias anteriores. A sua resolução é mais rápida, mas o limite inferior é sempre igual ao que é obtido com a relaxação linear do modelo de Talbot.

c. Conjuntos de instâncias onde se varia o número de recursos renováveis

• Conjunto de instâncias r1

Instâncias	Ótimo	1ª Dec	2ª Dec	3ª Dec	Rel. Lin
r110_1.mm	17	17	17	17	17
r111_1.mm	16	16	16	16	16
r112_1.mm	27	27	27	27	27
r113_1.mm	27	-	23	23	23
r114_1.mm	29	29	29	29	29
r115_1.mm	23	-	20.393	20.393	20.393
r116_1.mm	27	27	27	27	27
r117_1.mm	29	28.083	27.262	27.275	27.205
r118_1.mm	18	18	18	18	18
r119_1.mm	16	16	16	16	16
Coinc. Ótimo		70%	70%	70%	70%

Tabela 5.13 – Limites inferiores r1

• Conjunto de instâncias r2

Instâncias	Ótimo	1ª Dec	2ª Dec	3ª Dec	Rel. Lin
j1610_1.mm	22	22	22	22	22
j1611_1.mm	25	25	24	24	24
j1612_1.mm	20	20	20	20	20
j1613_1.mm	30	-	23	23	23
j1614_1.mm	21	-	21	21	21
j1615_1.mm	23	23	23	23	23
j1616_1.mm	24	24	24	24	24
j1617_1.mm	19	17	17	17	17
j1618_1.mm	27	25	25	25	25
j1619_1.mm	22	22	22	22	22
Coinc. Ótimo		60%	60%	60%	60%

Tabela 5.14 – Limites inferiores r2

• Conjunto de instâncias r3

Instâncias	Ótimo	1ª Dec	2ª Dec	3ª Dec	Rel. Lin
r310_1.mm	28	26	26	26	26
r311_1.mm	20	20	20	20	20
r312_1.mm	14	14	14	14	14
r313_1.mm	35	-	27	27	27
r314_1.mm	28	-	24	24	24
r315_1.mm	21	-	19	19	19
r316_1.mm	23	23	22.240	22.240	22.240
r317_1.mm	32	29	29	29	29
r318_1.mm	18	18	18	18	18
r319_1.mm	18	18	18	18	18
Coinc. Ótimo		50%	40%	40%	40%

Tabela 5.15 – Limites inferiores r3

• Conjunto de instâncias r4

Instâncias	Ótimo	1ª Dec	2ª Dec	3ª Dec	Rel. Lin
r410_1.mm	30	-	25.151	25.151	25.151
r411_1.mm	20	19	19	19	19
r412_1.mm	21	21	20	20	20
r414_1.mm	26	-	23	23	23
r415_1.mm	19	18	18	18	18
r416_1.mm	19	19	19	19	19
r417_1.mm	33	23.125	22.389	22.395	22.389
r418_1.mm	24	24	24	24	24
r419_1.mm	16	16	16	16	16
r420_1.mm	25	25	25	25	25
Coinc. Ótimo		50%	40%	40%	40%

Tabela 5.16 – Limites inferiores r4

• Conjunto de instâncias r5

Instâncias	Ótimo	1ª Dec	2ª Dec	3ª Dec	Rel. Lin
r510_1.mm	20	-	18	18	18
r511_1.mm	21	20.4	20	20	20
r512_1.mm	20	20	20	20	20
r513_1.mm	37	-	22.384	22.282	22.282
r514_1.mm	21	17.572	17	17	17
r515_1.mm	22	21	21	21	21
r516_1.mm	16	16	15	15	15
r519_1.mm	20	20	20	20	20
r520_1.mm	25	25	25	25	25
r521_1.mm	43	-	25.619	25.618	25.618
Coinc. Ótimo		40%	30%	30%	30%

Tabela 5.17 – Limites inferiores r5

A análise dos resultados obtidos quando se varia o número de recursos renováveis considerados em cada instância permite constatar que existe em todas as decomposições uma acentuada degradação do desempenho das mesmas com o aumento do número de recursos.

d. Conjuntos de instâncias onde se varia o número de recursos não renováveis

• Conjunto de instâncias n0

Instâncias	Ótimo	1ª Dec	2ª Dec	3ª Dec	Rel. Lin
n010_1.mm	19	17.353	17.181	17	17
n011_1.mm	16	16	16	16	16
n012_1.mm	21	21	21	21	21
n013_1.mm	36	-	24.504	24.168	24.168
n014_1.mm	22	19.401	19	19	19
n015_1.mm	15	15	15	15	15
n016_1.mm	24	24	24	24	24
n017_1.mm	31	19.503	18.676	18.658	18.406
n018_1.mm	17	16	16	16	16
n019_1.mm	14	14	14	14	14
Coinc. Ótimo		50%	50%	50%	50%

Tabela 5.18 – Limites inferiores n0

• Conjunto de instâncias n1

Instâncias	Ótimo	1ª Dec	2ª Dec	3ª Dec	Rel. Lin
n110_1.mm	17	17	17	17	17
n111_1.mm	29	29	29	29	29
n112_1.mm	26	26	25	25	25
n114_1.mm	31	28.471	27.076	27.076	27.076
n115_1.mm	22	22	22	22	22
n116_1.mm	19	19	19	19	19
n117_1.mm	21	18.729	18	18	18
n118_1.mm	20	20	20	20	20
n119_1.mm	20	20	20	20	20
n120_1.mm	14	14	14	14	14
Coinc. Ótimo		80%	70%	70%	70%

Tabela 5.19 – Limites inferiores n1

• Conjunto de instâncias n2

Instâncias	Ótimo	1ª Dec	2ª Dec	3ª Dec	Rel. Lin
j1610_1.mm	22	22	22	22	22
j1611_1.mm	25	25	24	24	24
j1612_1.mm	20	20	20	20	20
j1613_1.mm	30	-	23	23	23
j1614_1.mm	21	-	21	21	21
j1615_1.mm	23	23	23	23	23
j1616_1.mm	24	24	24	24	24
j1617_1.mm	19	17	17	17	17
j1618_1.mm	27	25	25	25	25
j1619_1.mm	22	22	22	22	22
Coinc. Ótimo		60%	60%	60%	60%

Tabela 5.20 – Limites inferiores n2

• Conjunto de instâncias n3

Instâncias	Ótimo	1ª Dec	2ª Dec	3ª Dec	Rel. Lin
n310_1.mm	19	19	19	19	19
n311_1.mm	19	-	17	17	17
n312_1.mm	25	25	24	24	24
n315_1.mm	21	21	21	21	21
n316_1.mm	15	15	13.251	13.251	13.251
n317_1.mm	27	23.388	23.089	23.089	23.089
n318_1.mm	28	28	28	28	28
n319_1.mm	19	19	19	19	19
n320_1.mm	23	23	23	23	23
n321_1.mm	29	-	23	23	23
Coinc. Ótimo		70%	50%	50%	50%

Tabela 5.21 – Limites inferiores n3

O aumento do número de recursos não renováveis por instância não tem grande influência no qualidade das decomposições, sendo mesmo de registrar a boa qualidade da primeira decomposição.

5.3 Limites superiores para o MRCPSP

A segunda parte dos nossos testes computacionais incidiu na nossa extensão da heurística de Boctor descrita no Capítulo 4. Usámos as instâncias da PSPLIB, sendo que desta vez resolvemos todas as instâncias de cada conjunto, dada a rapidez com que é executada a heurística. O PC usado é o mesmo que o usado para obter os resultados da secção anterior.

Os resultados obtidos são listados nas tabelas seguintes. Nessas tabelas, indicamos o número de instâncias para as quais a heurística obteve uma solução, e comparamos a qualidade das soluções obtidas com o valor das soluções óptimas.

a. Conjuntos de instâncias onde varia o número de actividades

• Conjunto de instâncias j10

Nº de instâncias	Com solução	Sem solução
536	524	12
	97.8%	2.2%

Afastamento em relação ao óptimo						Total
0%	<=10%	10%<x<=25%	25%<x<=50%	50%<x<=100%	>100%	
97	54	105	178	77	13	524
18.1%	10.1%	19.6%	33.2%	14.4%	2.4%	97.8%

Tabela 5.22 – Limites superiores j10

• Conjunto de instâncias j12

Nº de instâncias	Com solução	Sem solução
547	536	11
	98.0%	2.0%

Afastamento em relação ao óptimo						Total
0%	<=10%	10%<x<=25%	25%<x<=50%	50%<x<=100%	>100%	
115	39	122	168	84	8	536
21.0%	7.1%	22.3%	30.7%	15.4%	1.5%	98.0%

Tabela 5.23 – Limites superiores j12

• Conjunto de instâncias j14

Nº de instâncias	Com solução	Sem solução
551	538	13
	97.6%	2.4%

Afastamento em relação ao óptimo						Total
0%	<=10%	10%<x<=25%	25%<x<=50%	50%<x<=100%	>100%	
106	39	111	172	99	11	538
19.2%	7.1%	20.1%	31.2%	18.0%	2.0%	97.6%

Tabela 5.24 – Limites superiores j14

• Conjunto de instâncias j16

Nº de instâncias	Com solução	Sem solução
550	535	15
	97.3%	2.7%

Afastamento em relação ao ótimo						Total
0%	<=10%	10%<x<=25%	25%<x<=50%	50%<x<=100%	>100%	
108	36	100	166	112	13	535
19.6%	6.5%	18.2%	30.2%	20.4%	2.4%	97.3%

Tabela 5.25 – Limites superiores j16

• Conjunto de instâncias j18

Nº de instâncias	Com solução	Sem solução
552	544	8
	98.6%	1.4%

Afastamento em relação ao ótimo						Total
0%	<=10%	10%<x<=25%	25%<x<=50%	50%<x<=100%	>100%	
109	35	100	168	119	13	544
19.7%	6.3%	18.1%	30.4%	21.6%	2.4%	98.6%

Tabela 5.26 – Limites superiores j18

• Conjunto de instâncias j20

Nº de instâncias	Com solução	Sem solução
554	543	11
	98.0%	2.0%

Afastamento em relação ao ótimo						Total
0%	<=10%	10%<x<=25%	25%<x<=50%	50%<x<=100%	>100%	
101	26	124	153	123	16	543
18.2%	4.7%	22.4%	27.6%	22.2%	2.9%	98.0%

Tabela 5.27 – Limites superiores j20

• Conjunto de instâncias j30

Nº de instâncias	Com solução	Sem solução
550	536	14
	97.5%	2.5%

Afastamento em relação ao óptimo (heurística)						Total
0%	<=10%	10%<x<=25%	25%<x<=50%	50%<x<=100%	>100%	
109	32	75	155	145	20	536
19.8%	5.8%	13.6%	28.2%	26.4%	3.6%	97.5%

Tabela 5.28 – Limites superiores j30

A nova heurística apresenta uma taxa de resolução (instâncias para as quais se obtém uma solução) muito boa para os conjuntos de instâncias onde se varia o número de actividades, sendo sempre superior a 97%.

Da análise das várias tabelas (Tabela 5.22 a Tabela 5.28), conclui-se ainda que a heurística se mostra insensível ao incremento sucessivo do número de actividades, visto que o nível de instâncias com solução se mantém praticamente constante ao longo dos vários conjuntos de instâncias testados. Mesmo a percentagem de soluções obtidas pela heurística e que coincidiram com o valor óptimo das instâncias manteve-se sempre muito estável, com uma taxa a rondar os 20%.

b. Conjuntos de instâncias onde se varia o número de modos

• Conjunto de instâncias m1

Nº de instâncias	Com solução	Sem solução
640	640	0
	100.0%	0.0%

Afastamento em relação ao óptimo (heurística)						Total
0%	<=10%	10%<x<=25%	25%<x<=50%	50%<x<=100%	>100%	
299	193	135	13	0	0	640
46.7%	30.2%	21.1%	2.0%	0.0%	0.0%	100.0%

Tabela 5.29 – Limites superiores m1

• Conjunto de instâncias m2

Nº de instâncias	Com solução	Sem solução
481	465	16
	96.7%	3.3%

Afastamento em relação ao ótimo (heurística)						Total
0%	<=10%	10%<x<=25%	25%<x<=50%	50%<x<=100%	>100%	
90	62	130	144	38	1	465
18.7%	12.9%	27.0%	29.9%	7.9%	0.2%	96.7%

Tabela 5.30 – Limites superiores m2

• Conjunto de instâncias m3

Nº de instâncias	Com solução	Sem solução
550	535	15
	97.3%	2.7%

Afastamento em relação ao ótimo (heurística)						Total
0%	<=10%	10%<x<=25%	25%<x<=50%	50%<x<=100%	>100%	
108	36	100	166	112	13	535
19.6%	6.5%	18.2%	30.2%	20.4%	2.4%	97.3%

Tabela 5.31 – Limites superiores m3

• Conjunto de instâncias m4

Nº de instâncias	Com solução	Sem solução
555	533	22
	96.0%	4.0%

Afastamento em relação ao ótimo (heurística)						Total
0%	<=10%	10%<x<=25%	25%<x<=50%	50%<x<=100%	>100%	
100	32	80	155	138	28	533
18.0%	5.8%	14.4%	27.9%	24.9%	5.0%	96.0%

Tabela 5.32 – Limites superiores m4

• Conjunto de instâncias m5

Nº de instâncias	Com solução	Sem solução
558	530	28
	95.0%	5.0%

Afastamento em relação ao óptimo (heurística)						Total
0%	<=10%	10%<x<=25%	25%<x<=50%	50%<x<=100%	>100%	
94	30	62	149	169	26	530
16.8%	5.4%	11.1%	26.7%	30.3%	4.7%	95.0%

Tabela 5.33 – Limites superiores m5

Os resultados obtidos aplicando a heurística ao conjunto de instâncias onde se varia o número de modos por actividade são bastante satisfatórios. À semelhança do que acontecia com os conjuntos associados a variações do número de actividades, a heurística apresenta uma boa taxa de resolução para os vários conjuntos de instâncias, com um mínimo de 95% quando o nº de modos/actividade é de 5.

O desempenho da heurística parece, no entanto, ser influenciada pelo incremento do número de modos por actividade, já que, ainda que ligeiramente, a percentagem de instâncias com solução apresenta uma ligeira tendência decrescente.

Realce-se ainda os bons resultados obtidos para o conjunto de instâncias com apenas 1 modo/actividade, onde se obteve uma taxa de resolução das instâncias de 100% e uma taxa de soluções coincidentes com a solução óptima de 46,7%.

c. Conjuntos de instâncias onde se varia o número de recursos renováveis

• Conjunto de instâncias r1

Nº de instâncias	Com solução	Sem solução
553	546	7
	98.7%	1.3%

Afastamento em relação ao ótimo (heurística)						Total
0%	<=10%	10%<x<=25%	25%<x<=50%	50%<x<=100%	>100%	
126	35	100	168	104	13	546
22.8%	6.3%	18.1%	30.4%	18.8%	2.4%	98.7%

Tabela 5.34 – Limites superiores r1

• Conjunto de instâncias r2

Nº de instâncias	Com solução	Sem solução
550	535	15
	97.3%	2.7%

Afastamento em relação ao ótimo (heurística)						Total
0%	<=10%	10%<x<=25%	25%<x<=50%	50%<x<=100%	>100%	
108	36	100	166	112	13	535
19.6%	6.5%	18.2%	30.2%	20.4%	2.4%	97.3%

Tabela 5.35 – Limites superiores r2

• Conjunto de instâncias r3

Nº de instâncias	Com solução	Sem solução
557	537	20
	96.4%	3.6%

Afastamento em relação ao ótimo (heurística)						Total
0%	<=10%	10%<x<=25%	25%<x<=50%	50%<x<=100%	>100%	
94	33	118	177	104	11	537
16.9%	5.9%	21.2%	31.8%	18.7%	2.0%	96.4%

Tabela 5.36 – Limites superiores r3

• Conjunto de instâncias r4

Nº de instâncias	Com solução	Sem solução
552	539	13
	97.6%	2.4%

Afastamento em relação ao óptimo (heurística)						Total
0%	<=10%	10%<x<=25%	25%<x<=50%	50%<x<=100%	>100%	
89	33	138	167	98	14	539
16.1%	6.0%	25.0%	30.3%	17.8%	2.5%	97.6%

Tabela 5.37 – Limites superiores r4

• Conjunto de instâncias r5

Nº de instâncias	Com solução	Sem solução
546	536	10
	98.2%	1.8%

Afastamento em relação ao óptimo (heurística)						Total
0%	<=10%	10%<x<=25%	25%<x<=50%	50%<x<=100%	>100%	
83	44	112	167	113	17	536
15.2%	8.1%	20.5%	30.6%	20.7%	3.1%	98.2%

Tabela 5.38 – Limites superiores r5

As tabelas 5.34 a 5.38 apresentam os resultados obtidos com a heurística quando aplicada aos conjuntos de instâncias onde se varia o número de recursos renováveis.

Mais uma vez, a taxa de resolução das instâncias é bastante elevada e muito estável com a variação do nº de recursos, apresentando mínimo de 96,4% para o conjunto de instâncias com 3 recursos renováveis. No que se refere à percentagem de soluções coincidentes com a solução óptima, ela situa-se entre os 23,1% (1 recurso renovável) e os 15,5% (5 recursos renováveis), apresentando ainda uma tendência decrescente, como seria de esperar visto que a complexidade das instâncias vai crescendo com o aumento do número de recursos renováveis considerados.

d. Conjuntos de instâncias onde se varia o número de recursos não renováveis

• Conjunto de instâncias n0

Nº de instâncias	Com solução	Sem solução
470	470	0
	100.0%	0.0%

Afastamento em relação ao óptimo (heurística)						Total
0%	<=10%	10%<x<=25%	25%<x<=50%	50%<x<=100%	>100%	
201	59	102	88	19	1	470
42.8%	12.6%	21.7%	18.7%	4.0%	0.2%	100.0%

Tabela 5.39 – Limites superiores n0

• Conjunto de instâncias n1

Nº de instâncias	Com solução	Sem solução
637	631	6
	99.1%	0.9%

Afastamento em relação ao óptimo (heurística)						Total
0%	<=10%	10%<x<=25%	25%<x<=50%	50%<x<=100%	>100%	
99	52	155	202	121	2	631
15.5%	8.2%	24.3%	31.7%	19.0%	0.3%	99.1%

Tabela 5.40 – Limites superiores n1

• Conjunto de instâncias n2

Nº de instâncias	Com solução	Sem solução
550	535	15
	97.3%	2.7%

Afastamento em relação ao óptimo (heurística)						Total
0%	<=10%	10%<x<=25%	25%<x<=50%	50%<x<=100%	>100%	
108	36	100	166	112	13	535
19.6%	6.5%	18.2%	30.2%	20.4%	2.4%	97.3%

Tabela 5.41 – Limites superiores n2

• Conjunto de instâncias n3

Nº de instâncias	Com solução	Sem solução
600	534	66
	89.0%	11.0%

Afastamento em relação ao óptimo (heurística)						Total
0%	<=10%	10%<x<=25%	25%<x<=50%	50%<x<=100%	>100%	
99	35	104	165	118	13	534
16.5%	5.8%	17.3%	27.5%	19.7%	2.2%	89.0%

Tabela 5.42 – Limites superiores n3

Os resultados obtidos com a heurística quando aplicada ao conjunto de instâncias onde se varia o número de recursos não renováveis são apresentados nas tabelas 5.39 a 5.42.

A percentagem de instâncias para as quais a heurística consegue obter uma solução é amplamente satisfatória, revelando no entanto que à medida que se incrementam o número de recursos não renováveis considerados nas instâncias, o desempenho da heurística parece ser afectado. De facto, para o conjunto de instâncias com 2 recursos não renováveis obteve-se uma taxa de sucesso na resolução das instâncias de 97,3%, mas quando se considerou a existência de 3 recursos não renováveis, a taxa de resolução baixou para 89%, sendo mesmo o valor mais baixo de entre todas os testes que se efectuaram com a heurística.

No conjunto de instâncias em que não se considera a existência de recursos não renováveis (Tabela 5.39), a heurística obtém solução para todas as instâncias, conseguindo ainda uma taxa de coincidência da solução obtida com a solução óptima de 42,8%.

5.4 Conclusões

Os testes computacionais realizados no âmbito desta dissertação tiveram por objectivo avaliar a qualidade das três decomposições apresentadas no Capítulo 3. Essa avaliação

foi feita comparando o valor do limite inferior dado pelas correspondentes relaxações lineares e o valor da relaxação linear do modelo de Talbot. Por outro lado, procuramos avaliar o desempenho da nova heurística apresentada no Capítulo 4 para o cálculo de limites superiores (soluções válidas mas não necessariamente ótimas) para o MRCPSP.

As instâncias utilizadas para a realização dos testes foram extraídas da PSPLIB, uma biblioteca de instâncias de problemas do tipo RCPSP disponível na Internet. Para o MRCPSP existem vários conjuntos de instâncias, encontrando-se estes agrupados pelas características das instâncias neles contidas. Assim, têm-se 7 conjuntos onde se varia o número de actividades por instância, desde 10 actividades até 30 actividades, 5 conjuntos onde se varia o número de modos, de 1 a 5 modos/actividade, 5 conjuntos onde se varia o número de recursos renováveis, de 1 a 5, e finalmente 4 conjuntos onde se varia o número de recursos não renováveis, de 0 a 3 recursos não renováveis por instância.

Os testes realizados para as decomposições tiveram como imposição um tempo máximo de 480 segundos para pesquisa da solução e foram utilizadas apenas 10 instâncias de cada conjunto. A partir dos resultados obtidos para as decomposições constata-se que a primeira decomposição apresentada é a que normalmente produz melhores resultados. Contudo é também aquela que necessita de mais tempo de computação até atingir o óptimo. Em alguns casos, o tempo de processamento imposto como limite foi excedido na resolução dessa decomposição. Com a segunda e terceira decomposições, foi sempre possível encontrar o óptimo dentro do limite de tempo. Contudo, o valor do limite inferior é muitas vezes igual ao valor óptimo do modelo de Talbot.

Relativamente à nova heurística, um dos aspectos a destacar é a capacidade dessa heurística em obter soluções válidas para um problema complexo como o MRCPSP. De facto, a percentagem de resolução com sucesso das instâncias é quase sempre próxima dos 97%. Um outro aspecto importante prende-se com o número de instâncias para as quais a heurística obteve uma solução equivalente à solução óptima. Esse valor ronda os 20%, atingindo mesmo 46,7% no caso em que é considerado apenas um modo por actividade e 42,8% no caso de não existirem recursos não renováveis.

Em termos de desempenho, e para as instâncias testadas, verifica-se uma ligeira degradação com o aumento do número de modos por actividade ou dos recursos, sendo mais vincada essa degradação no caso do aumento do número de recursos não renováveis.

Capítulo 6

Conclusões

O interesse na área dos problemas de sequenciamento (ou escalonamento) de projectos com recursos escassos e com vários modos de execução para cada actividade (conhecido na literatura por Multi-Mode Resource Constrained Scheduling Problem - MRCPSP) tem vindo a crescer de forma significativa nos últimos anos. Este interesse resulta da necessidade que as empresas têm de melhorar o seu desempenho, na busca de maior eficiência e competitividade. Na prática, o MRCPSP modela de forma mais adequada a realidade dos projectos. Por outro lado, dada a inerente complexidade associada a este problema, ele tem sido encarado como um verdadeiro desafio por parte dos investigadores.

Como acontece com muitos outros problemas de optimização, os métodos de resolução que foram desenvolvidos e testados para estes problemas dividem-se em dois grandes grupos: os métodos de resolução exacta e os métodos heurísticos. Os métodos exactos são essencialmente baseados em algoritmos de partição e avaliação sucessivas. Tipicamente, os tempos de computação necessários para resolver instâncias reais são elevados, apesar dos resultados de investigação apontarem para melhorias progressivas. Os métodos heurísticos produzem soluções próximas das soluções óptimas, e requerem tempos de computação bastante inferiores. Neste campo, destacam-se as meta-heurísticas como os algoritmos genéticos, por exemplo. Neste momento, existem ainda poucos resultados relativamente às heurísticas mais simples de tipo construtivo.

O objectivo desta dissertação foi duplo. Por um lado, procurámos avaliar a qualidade de modelos alternativos para o MRCPSP. A qualidade de um modelo de Programação

Inteira depende do valor da sua relaxação linear, e sobretudo na distância entre esse valor e o ótimo inteiro (o chamado intervalo de integralidade). Para aferir essa qualidade, as relaxações lineares devem ser resolvidas exactamente através por exemplo do método Simplex. Os modelos investigados nesta dissertação resultam de decomposições de Dantzig-Wolfe de um modelo original e compacto introduzido por Talbot. Foram testados três modelos. Entre eles, a decomposição que deu origem a um subproblema mais complexo foi aquela que deu também os melhores valores em termos de limites inferiores. Apresentamos um vasto leque de resultados computacionais para instâncias da literatura. De notar que as decomposições analisadas nesta dissertação nunca foram exploradas na literatura.

Na segunda parte desta dissertação, explorámos uma heurística construtiva para o MRCPSPP com recursos não renováveis. A heurística foi implementada na linguagem de programação C, e foi testada também em instâncias da literatura. Os resultados obtidos foram bons, na medida em que para muitas instâncias foi possível encontrar uma solução válida e também ótima. Em alguns casos, a heurística poderá falhar em encontrar um escalonamento válido para as actividades. Segundo as experiências que realizámos, essa situação ocorre num número muito reduzido de ocasiões.

Referências

- [1] Akkan C, “A lagrangian heuristic for the discrete time-cost tradeoff problem for activity-on-arc project networks”, Working paper, Koç University, Istanbul, 1998.
- [2] Akkan C, “Iterated local search algorithms for the discrete time-cost tradeoff problem”, Working paper, Koç University, Istanbul, 1999.
- [3] Akkan C, Drexl A, Kimms A, “Network decomposition-based benchmark results for the discrete time-cost tradeoff problem”, *European Journal of Operational Research* 165: 339-358, 2005.
- [4] Alvelos F, “Branch-and-price and multicommodity flows”, PhD. Dissertation, Universidade do Minho, 2005.
- [5] Barnhart C, Johnson E, Nemhauser G, Savelsbergh M, and Vance P, “Branch-and-price: Column generation for solving huge integer programs”, *Operations Research* 46(3):316–329, 1998.
- [6] Bartusch M, Möhring R.H, Radermacher F.J, “Scheduling project networks with resource constraints and time windows”, *Annals of Operations Research* 16: 201-240, 1988.
- [7] Bein W.W, Kamburowski J, Stallmann M, “Optimal reduction of two-terminal directed acyclic graphs”, *SIAM Journal of Computing* 21: 1112-1129, 1992.

- [8] Boctor F.F, “Heuristics for scheduling projects with resource restrictions and several resource-duration modes”, *International Journal of Production Research* 31: 2547-58, 1993.
- [9] Boctor F.F, “A new and efficient heuristic for scheduling projects with resource restrictions and multiple execution modes”, *European Journal of Operational Research* 90: 349-361, 1996.
- [10] Brucker P, Drexl A, Mohring R, Neumann K, Pesch E, “Resource-constrained project scheduling: notation, classification, models, and methods”. *European Journal of Operational Research* 112: 3-41, 1999.
- [11] Christofides N, Alvarez-Valdes R, Tamarit J. M, “Project scheduling with resource constraints: a branch and bound approach”, *European Journal of Operational Research* 29: 262–273, 1987.
- [12] Coelho J, Tavares L, “Comparative analysis of metaheuristics for the resource constrained project scheduling problem”, Technical report, Department of Civil Engineering, Instituto Superior Técnico, Portugal, 2003.
- [13] Dantzig G.B, Wolfe P, “Decomposition principle for linear programs”, *Operations Research* 8: 101-111, 1960.
- [14] De P, Dunne E.J, Ghosh J.B, Wells C.E, “Discrete time-cost tradeoff problem revisited”, *European Journal of Operational Research* 81: 225-238, 1995.
- [15] De Reyck B, “Scheduling projects with generalized precedence relations: Exact and heuristic procedures”, PhD. Dissertation, Katholieke Universiteit Leuven, 1998.
- [16] De Reyck B, Herroelen W, “The multi-mode resource-constrained project scheduling problem with generalized precedence relations”, *European Journal of Operational Research* 119: 538-556, 1999.

- [17] Deineko V.G, Woeginger G.J, “Hardness of approximation on the discrete time-cost tradeoff problem”, *Operations Research Letters* 29: 207-210, 2001.
- [18] Demeulemeester E, Herroelen W, “A branch-and-bound procedure for the multiple resource-constrained project scheduling problem”, *Management Science* 38:1803-1818, 1992.
- [19] Deumelemeester E, Herroelen W, Elmaghraby S.E, “Optimal procedures for the discrete time/cost trade-off problem in project networks”, *European Journal of Operational Research* 88: 50-68, 1996.
- [20] Drexl A, Grünewald J, “Nonpreemptive multi-mode resource-constrained project scheduling”, *IIE Transactions* 25: 74-81, 1993.
- [21] Elmaghraby S.E, “Activity Networks: Project planning and control by network models”, Wiley, New York, 1977.
- [22] Elmaghraby S.E, “Resource allocation via dynamic programming in activity networks”, *European Journal of Operational Research* 88: 50-86, 1992.
- [23] Ford L.R, Fulkerson D.R, “A suggested computation for maximal multicommodity network flows”, *Management Science*, 5:97-101, 1958.
- [24] Hartmann S, “Project scheduling with multiple modes: a genetic algorithm”, *Annals of Operational Research* 102: 111–135, 2001.
- [25] Hartmann S, Kolisch R, “Experimental evaluation of state-of-the-art heuristics for the resource-constrained project scheduling problem”, *European Journal of Operational Research* 127: 394–407, 2000.
- [26] Hartmann S, Drexl A, “Project scheduling with multiple modes: a comparison of exact algorithms”. *Networks*, to appear

- [27] Heilmann R, “Resource-constrained project scheduling: a heuristic for the multi-mode case”, *OR Spektrum* 23: 335-357, 2001.
- [28] Heilmann R, “A branch-and-bound procedure for the multi-mode resource-constrained project scheduling problem with minimum and maximum time lags”, *European Journal of Operational Research* 144: 348-365, 2003.
- [29] Herroelen W, Demeulemeester E, Reyck B, “A note on the paper Resource-constrained project scheduling: notation, classification, models, and methods by Brucker et al.”, *European Journal of Operational Research* 128: 679-688, 2001.
- [30] Icmeli O, Rom W, “Ensuring quality in resource constrained project scheduling”, *European Journal of Operational Research* 103: 483-96, 1997.
- [31] Icmeli O, Rom W, “Analysis of the characteristics of projects in diverse industries”, *Journal of Operations Management* 16: 43-61, 1998.
- [32] Józefowska J, Mika M, Rózycki R, Waligóra G, Weglarz J, “Simulated annealing for multi-mode resource-constrained project scheduling”, *Annals of Operational Research* 102: 137–155, 2001.
- [33] Kolisch R “Project scheduling under resource constraints – efficient heuristics for several problem classes” *Physica*, Heidelberg, 1995.
- [34] Kolisch R, Drexel A, “Local search for nonpreemptive multi-mode resource-constrained project scheduling”, *IIE Transactions* 29: 987-99, 1997.
- [35] Kolisch R, Hartmann S, “Experimental investigation of heuristics for resource-constrained project scheduling: An update”, *European Journal of Operational Research* 174: 23–37, 2006.
- [36] Kolisch R, Padman R, “An integrated survey of deterministic project scheduling”, *OMEGA International Journal of Management Science* 29: 249–272, 2001.

- [37] Kolisch, R, Sprecher A, “PSPLIB - A Project scheduling problem library”, European Journal of Operational Research 96: 205-216, 1996.
- [38] Özdamar L, Ulusoy G, “A local constraint based analysis approach to project scheduling under general resource constraints”, European Journal of Operational Research 79: 287-98, 1994.
- [39] Patterson J.H, Slowinski R, Talbot F.B, Weglarz J, “An algorithm for a general class of precedence and resource constrained scheduling problems”, Elsevier, Amsterdam, The Netherlands 3-28, 1989.
- [40] Project Management Institute, “A Guide to the Project Management Body of Knowledge (PMBOK® Guide)”, Third Edition, ISBN-10:193069945X, ISBN-13: 978-1930699458, 2004.
- [41] Robinson D.R, “A dynamic programming solution to the cost-time tradeoff for CPM”, Management Science 22: 158-166, 1975.
- [42] Russell A.H, “Cash flows in networks”, Management Science 16(5): 357-73, 1970.
- [43] Schirmer A, Riesenberg S, “Parameterized heuristics for project scheduling – biased random sampling methods”, Technical Report 456, Manuskripte aus den Instituten für Betriebswirtschaftslehre der Universität Kiel, Germany, 1997.
- [44] Schwindt C, “Verfahren zur Lösung des ressourcenbeschränkten Projektdauerminimierungsproblems mit planungsabhängigen” Zeitfenstern, Shaker, Aachen, 1998.
- [45] Skutella M, “Approximation algorithms for the discrete time-cost tradeoff problem” in: Proceedings of the Eighth Annual ACM-SIAM Symposium on Discrete Algorithms, New Orleans, LA, 501-508, 1997.

- [46] Sprecher A, “Resource-constrained project scheduling: Exact methods for the multi-mode case”, *Lecture Notes in Economics and Mathematical Systems* 409 Springer, Berlin, 1994.
- [47] Sprecher A, Drexl A, “Solving multi-mode resource-constrained project scheduling problems by a simple, general and powerful sequencing algorithm. Part I: Theory”, *Manuskripte aus den Instituten für Betriebswirtschaftslehre* 385, University of Kiel, Germany, 1996.
- [48] Sprecher A, Hartmann S, Drexl A, “An exact algorithm for project scheduling with multiple modes”, *OR Spektrum* 19: 195–203, 1997.
- [49] Talbot F.B, “Resource-constrained project scheduling with time-resource tradeoffs: the non-preemptive case”, *Management Science* 28: 1197-1210, 1982.
- [50] Tavares L, “A review of the contribution of operational research to project management”, *European Journal of Operational Research* 136: 1-18, 2002.
- [51] Yang B, Geunes J, O’Brian W.J, “Resource-constrained project scheduling: past work and new directions”, *Research Report 2001-6*, Department of Industrial and Systems Engineering, University of Florida, 2001.