



Universidade do Minho
Escola de Engenharia

Nuno Manuel Ferreira Braga
Modelos e Métodos de Optimização para Problemas de Planeamento e Gestão Integrada de Operações

Nuno Manuel Ferreira Braga

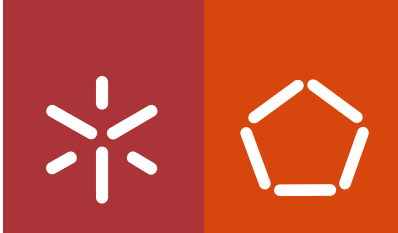
Modelos e Métodos de Optimização para Problemas de Planeamento e Gestão Integrada de Operações

FCT

Fundação para a Ciência e a Tecnologia

MINISTÉRIO DA EDUCAÇÃO E CIÊNCIA





Universidade do Minho
Escola de Engenharia

Nuno Manuel Ferreira Braga

**Modelos e Métodos de Optimização
para Problemas de Planeamento e
Gestão Integrada de Operações**

Tese de Doutoramento em Engenharia Industrial e de Sistemas

Trabalho realizado sob a orientação do

Professor Doutor Cláudio Manuel Martins Alves

e da

**Professora Doutora Rita Alexandra Santos
Gonçalves de Macedo**

setembro de 2016


DECLARACAO DE INTEGRIDADE

Declaro ter actuado com integridade na elaboração da presente tese. Confirmando que em todo o trabalho conducente à sua elaboração não recorri à prática de plágio ou a qualquer forma de falsificação de resultados.

Mais declaro que tomei conhecimento integral do Código de Conduta Ética da Universidade do Minho.

Universidade do Minho, 16 de Setembro de 2016

Nome completo: Nuno Manuel Ferreira Braga

Assinatura: 

Agradecimentos

Terminada agora esta Tese de Doutoramento, não gostaria de deixar de agradecer às pessoas que contribuíram para a sua conclusão. A ajuda dessas pessoas foi imprescindível nesta caminhada.

Em primeiro lugar gostaria de agradecer ao meu orientador Cláudio Alves, que foi imprescindível para a realização deste trabalho. A sua ajuda durante todo este caminho ajudou-me a crescer e foi essencial para alargar os horizontes, mudar de perspectiva e encontrar motivação nos momentos mais difíceis. Foram muitas as reuniões de “motivação” que me deram força para continuar. A sua ajuda, profissionalismo e disponibilidade foram fundamentais para finalizar esta tese.

Gostaria de agradecer a quem também contribuiu bastante para este trabalho, o Professor José Valério de Carvalho. A sua genialidade, simpatia, humildade, disponibilidade, ideias e sugestões ajudaram a ultrapassar dificuldades encontradas na resolução dos problemas. Foi um privilégio ter conhecido o Professor Valério, ainda mais ter trabalhado com ele. Um muito obrigado!

À minha co-orientadora Rita Macedo um obrigado pelas correcções e sugestões para o trabalho.

Por todo o apoio técnico e incansável ajuda na reparação do material informático, gostaria de agradecer ao Acácio Costa.

Para a conclusão deste trabalho, além da ajuda na parte científica, a ajuda dos amigos e família foi fundamental. Assim, cabe-me agradecer aos colegas de gabinete, Bruna, Telmo, Ana Cecília, Rui, Ana, Catarina e Mónica, pelos cafés e momentos de descontração que ajudaram a reiniciar o trabalho com mais força quando esta já quase não existia.

Nuno Alexandre, Alberto Barbosa, José Eduardo Ferreira, Daniela, Célia e Joana Rodrigues (na tua área quem está em último é o mais importante), obrigado

pelos jantares, cafés, saídas, motivação e momentos de descontração que ajudaram a esquecer os problemas e elevar o ânimo para continuar. Nuno Alexandre, agora sim, vamos poder andar de mota!

Para a pessoa que me acompanhou durante todo este tempo, a minha namorada Joana Rodrigues, obrigado por tudo! Obrigado pela tua paciência quando a motivação não era muita, pela alegria e sorrisos quando estamos juntos. Obrigado pelos teus *post-its* coloridos que me deixam sem reação mas que sem eles não teria chegado aqui. Por todo o carinho e incentivo para continuar a trabalhar. Obrigado!

Aos meus pais, que através de todo o apoio e sacrifício fizeram-me continuar acreditando que seria possível. O vosso apoio incondicional foi fundamental neste percurso.

Por fim gostaria de dedicar esta tese aos meus pais, José Braga e Maria Cândida e à minha namorada Joana♥Rodrigues. Estou em dívida para com vocês para sempre.

A todos vocês, um muito obrigado!

Universidade do Minho, 16 de Setembro de 2016

Este trabalho foi financiado por Fundos FEDER através do Programa Operacional Factores de Competitividade COMPETE e por Fundos Nacionais através da FCT Fundação para a Ciência e a Tecnologia no âmbito do projecto PTDC/EGE-GES/116676/2010 (Ref. COMPETE: FCOMP-01-0124-FEDER-020430).

FCT

Fundação para a Ciência e a Tecnologia

MINISTÉRIO DA CIÊNCIA, TECNOLOGIA E ENSINO SUPERIOR



Resumo

A resolução de problemas de optimização de processos no domínio da gestão de cadeias de abastecimento tende cada vez mais a ser feita numa perspectiva integrada. O objectivo é obter uma solução global que seja melhor do que aquela que é obtida quando os problemas são resolvidos de forma independente. Atendendo a este facto, esta tese tem como objectivo analisar e propor modelos de programação inteira para a resolução de três problemas práticos de optimização integrada em cadeias de abastecimento: o problema combinado de corte e escalonamento, o problema integrado de planeamento e escalonamento e o problema de encaminhamento de veículos com múltiplas rotas e gestão integrada de inventários.

Para o problema combinado de corte e escalonamento, foi proposto um modelo de afectação que foi reforçado através de desigualdades obtidas utilizando funções duais válidas. Um novo modelo pseudo-polinomial exacto de fluxos em rede foi também proposto. Este modelo foi posteriormente revisto recorrendo-se à agregação de intervalos de tempo, e resolvido através de heurísticas. Os testes computacionais demonstraram a qualidade das abordagens tanto em termos das soluções obtidas como dos tempos de resolução.

Para o problema integrado de planeamento e escalonamento, foi proposto um novo modelo pseudo-polinomial de fluxos em rede. Foram exploradas estratégias de redução para diminuir o número de restrições do modelo, cujos resultados superaram os obtidos com outros modelos existentes. Nas instâncias com intervalos de optimalidade elevados, foram testadas duas heurísticas utilizando diferentes tipos de critérios de parâmetros e ordenação. As heurísticas consistiram na fixação de variáveis segundo critérios alternativos, tendo como objectivo reduzir a dimensão do modelo, tornando-o mais fácil de resolver.

Por último, para o problema de encaminhamento de veículos com múltiplas rotas e gestão integrada de inventários, foi proposto outro modelo pseudo-polinomial de fluxos em rede, que implica a enumeração de todas as rotas válidas. Para analisar a qualidade deste modelo, foram testados diferentes parâmetros com o objectivo de se determinar a influência dos mesmos na construção das rotas e na solução final. Os testes computacionais realizados mostraram que é possível resolver até à optimalidade instâncias com até 50 clientes.

Abstract

The resolution of optimization problems in the field of supply chain management in an integrated perspective has been increasingly used. The goal is to get a global solution that is better than the one that is obtained when the problems are solved independently. With this in mind, integer programming models are proposed and analyzed in this thesis to solve three practical problems in an integrated optimization of supply chains: the combined cutting stock and scheduling problem, the integrated planning and scheduling problem and the integrated inventory routing problem with multiple routes.

For the combined cutting stock and scheduling problem, an assignment model is proposed and strengthened by using cutting planes obtained from dual-feasible functions. A new exact pseudo-polynomial network flow model is also proposed. This model is later revised using the aggregation of time intervals, and solved by heuristics. The quality of the tested approaches is confirmed by the computational results.

For the integrated planning and scheduling problem, a new pseudo-polynomial network flow model is also developed. In order to reduce the number of constraints in the model, reduction strategies were applied. This model was able to obtain better results than other existing ones from the literature. In instances with high optimality gaps two heuristics using different parameters and sorting criteria are tested. The heuristics consist in fixing variables with alternative criteria, aiming to reduce the model size and therefore making it easier to be solved.

Finally, for the integrated inventory routing problem with multiple routes, another pseudo-polynomial network flow model is proposed. This model is based on the enumeration of all feasible routes. To analyze the quality of this model, tests were performed on some parameters that were changed in order to determine their influence in the construction of these routes and in the final solution. The computational tests show that it is possible to solve to optimality instances with up to 50 customers.

Publicações

N. Braga, C. Alves, e J. Valério de Carvalho. Exact solution of combined cutting stock and scheduling problems, volume 682 of Lecture Notes in Economics and Mathematical Systems, capítulo 17, páginas 131–39. Springer International Publishing, 2016b

N. Braga, C. Alves, R. Macedo, e J. Valério de Carvalho. A Model-Based Heuristic for the Combined Cutting Stock and Scheduling Problem, volume 9156 of Lecture Notes in Computer Science, páginas 490–505. Springer International Publishing, 2015

J. Rietz, C. Alves, N. Braga, e J. Valério de Carvalho. An exact approach based on a new pseudo-polynomial network flow model for integrated planning and scheduling. Computers & Operations Research, 76:183–94, 2016

N. Braga, C. Alves, R. Macedo, e J. M. Valério de Carvalho. Combined cutting stock and scheduling: a matheuristic approach. International Journal of Innovative Computing and Applications, 7(3):135 – 146, 2016a

Conteúdo

1	Introdução geral	1
1.1	Contexto e motivação	1
1.2	Objectivos de investigação	4
1.3	Conteúdo da tese	5
2	Planeamento e gestão integrada de operações: um estado da arte	7
2.1	Introdução	8
2.2	Problemas integrados de corte e empacotamento	14
2.2.1	Problemas de corte e empacotamento	14
2.2.2	Problemas integrados de corte e empacotamento	17
2.3	Problemas integrados de planeamento e escalonamento	20
2.3.1	Escalonamento em máquinas paralelas	20
2.3.2	Problemas integrados de escalonamento	21
2.3.3	Problemas não integrados relacionados	26
2.4	Problemas de encaminhamento de veículos e gestão de inventários	28
2.4.1	Problemas de encaminhamento de veículos	28
2.4.2	Encaminhamento de veículos e gestão integrada de inventários	31
2.5	Conclusões	34
3	O problema combinado de corte e escalonamento	35
3.1	Introdução	36
3.2	Modelos de programação inteira da literatura	40
3.2.1	O modelo de Reinertsen e Vossen	40
3.2.2	O modelo de Arbib e Marinelli	41

3.3	Modelo de afectação	43
3.3.1	Reforço do modelo de afectação	44
3.4	Um método de planos de corte baseado no modelo de afectação	45
3.4.1	Funções duais válidas e desigualdades válidas	45
3.4.2	Um método de planos de corte	46
3.5	Resultados computacionais	47
3.6	Um novo modelo exacto de fluxo em rede pseudo-polinomial	51
3.7	Uma abordagem heurística baseada no modelo de fluxos em rede	53
3.7.1	Visão geral	53
3.7.2	Modelo de fluxos em rede revisto	54
3.7.3	Modelo de afectação no tempo	56
3.8	Resultados computacionais	58
3.9	Conclusões	65
4	O problema integrado de planeamento e escalonamento	67
4.1	Introdução	68
4.2	O problema integrado de planeamento e escalonamento	72
4.3	Um novo modelo pseudo-polinomial de fluxos em rede	75
4.3.1	Definição	75
4.3.2	Exemplo	80
4.4	Estratégias de redução	84
4.5	Dependência entre o número de variáveis e a sequência de tarefas	85
4.6	Resultados computacionais	87
4.6.1	Análise comparativa de resultados	87
4.6.2	Actualização dos resultados usando um computador recente	91
4.7	Abordagens heurísticas alternativas	95
4.7.1	Estratégias heurísticas	101
4.7.2	Resultados computacionais	105
4.8	Conclusões	110
5	O problema de encaminhamento de veículos com múltiplas rotas e gestão integrada de inventários	111

5.1	Introdução	111
5.2	O problema de encaminhamento de veículos com múltiplas rotas e gestão integrada de inventários	115
5.2.1	Definição	115
5.2.2	Dados e parâmetros	116
5.2.3	Exemplo de uma instância do problema	118
5.3	Um modelo pseudo-polinomial de fluxos em arcos	122
5.4	Resultados computacionais	124
5.5	Conclusões	130
6	Conclusão geral e trabalho futuro	135
6.1	Conclusão geral	135
6.2	Trabalho futuro	138
	Referências	141

Lista de Figuras

2.1	Cadeia de abastecimento genérica	9
2.2	Representação genérica da integração de problemas na indústria	10
2.3	(a) - Envio baseado no transporte, (b) - Envio baseado na procura, Exemplo de um conjunto de clientes e armazéns que devem satisfazer a procura dos clientes, Adaptado de Durbin e Hoffman (2008).	13
2.4	Grafo de fluxos gerados por uma instância de três itens, retirado de Valério de Carvalho (2002)	16
2.5	Estratégias para o problema integrado de planeamento e escalonamento, adaptado de Maravelias e Sung (2009). (a) problema resolvido de forma hierárquica, (b) problema resolvido através de métodos interactivos e (c) onde o problema é resolvido de forma integrada através da pesquisa global.	24
2.6	Exemplo de um problema de VRP com as rotas possíveis e rotas optimizadas.	29
2.7	Solução IRP para dois períodos.	33
3.1	Esquema da heurística para o modelo de fluxos em rede revisto	54
3.2	Exemplo de tempos agregados - padrões optimizados.	58
4.1	Instância do Exemplo 2.	75
4.2	Grafo (V, E_1) para a instância Exemplo 2 (sem arcos de desperdício)	79
4.3	Solução representada graficamente para a instância do Exemplo 4.3.2.	84
5.1	Representação da instância graficamente com o armazém, clientes e respectivas distâncias situadas a meia distância dos arcos.	119

5.2	Solução válida para a instância IRP. A meio de cada ligação entre cliente é representada a distância percorrida, assim como o volume da mercadoria do veículo no trajecto.	121
5.3	Solução IRP para três períodos.	124

Lista de Tabelas

2.1	Modelos para o problema de corte e empacotamento (Delorme et al., 2016)	17
3.1	Resultados computacionais	49
3.2	Resultados computacionais	50
3.3	Resultados computacionais com $D = 2\bar{C}_{max} + 1$	61
3.4	Resultados computacionais com $D = 1.5\bar{C}_{max} + 1$	62
3.5	Resultados computacionais com $D = 1.1\bar{C}_{max} + 1$	63
3.6	Resultados computacionais com $D = 1.05\bar{C}_{max} + 1$	64
4.1	Exemplo de valores de $\mu(v)$ para $t = 1$ e $t > 1$	78
4.2	Resultados computacionais por número de tarefas e períodos de tempo.	92
4.3	Resultados computacionais por número de tarefas e máquinas.	93
4.4	Tamanho médio do modelo de fluxos em rede (4.4)–(4.8).	94
4.5	Resultados computacionais por número de tarefas e períodos de tempo (instâncias A)	96
4.6	Resultados computacionais por número de tarefas e períodos de tempo (instâncias B)	97
4.7	Resultados computacionais por número de tarefas e períodos de tempo (instâncias C)	98
4.8	Resultados computacionais por número de tarefas e máquinas (instâncias A)	99
4.9	Resultados computacionais por número de tarefas e máquinas (instâncias B)	100

4.10	Resultados computacionais por número de tarefas e máquinas (instâncias C)	101
4.11	Resultados computacionais - Algoritmo 2 (parte I)	108
4.12	Resultados computacionais - Algoritmo 2 (parte II)	109
4.13	Resultados computacionais com $maxit = 10$ do Algoritmo 3	109
4.14	Resultados computacionais com $maxit = 15$ do Algoritmo 3	110
5.1	Exemplo de uma instância do problema de encaminhamento de veículos com múltiplas rotas e gestão integrada de inventários. (a) Dados da localização do Armazém, (b) Dados gerais do problema, (c) Dados dos clientes e (c) Dados das procuras dos clientes.	120
5.2	Valores da função objectivo para os três períodos.	122
5.3	Testes computacionais com veículos de capacidade $C = 10$	127
5.4	Testes computacionais com veículos de capacidade $C = 13$	127
5.5	Testes computacionais com veículos de capacidade $C = 10$ e número máximo de clientes a visitar $N_{max} = 3$	128
5.6	Testes computacionais com veículos de capacidade $C = 10$ e número máximo de clientes a visitar $N_{max} = 4$	129
5.7	Testes computacionais com veículos de capacidade $C = 10$ e número máximo de clientes a visitar $N_{max} = 5$	130
5.8	Testes computacionais com veículos de capacidade $C = 13$ e número máximo de clientes a visitar $N_{max} = 3$	131
5.9	Testes computacionais com veículos de capacidade $C = 13$ e número máximo de clientes a visitar $N_{max} = 4$	132
5.10	Testes computacionais com veículos de capacidade $C = 13$ e número máximo de clientes a visitar $N_{max} = 5$	132
5.11	Testes computacionais com veículos de capacidade $C = 20$ e número máximo de clientes a visitar $N_{max} = 3$	133
5.12	Testes computacionais com veículos de capacidade $C = 20$ e número máximo de clientes a visitar $N_{max} = 4$	133

5.13 Síntese das instâncias resolvidas até à optimalidade / instâncias em que o modelo é gerado / máximo de clientes.	134
5.14 Valores médios das Tabelas 5.3–5.12 para as instâncias 1-12.	134

Lista de abreviaturas

VRP	Vehicle Routing Problem
IRP	Inventory Routing Problem
VMI	Vendor Managed Inventory
PI	Programação Inteira
CSSP	Combined Cutting Stock and Scheduling Problem
FDV	Função Dual Válida

Capítulo 1

Introdução geral

Conteúdo

1.1	Contexto e motivação	1
1.2	Objectivos de investigação	4
1.3	Conteúdo da tese	5

1.1 Contexto e motivação

A resposta rápida e eficiente a problemas reais por parte de algoritmos capazes de os solucionar em empresas, tem vindo a ser cada vez mais uma necessidade constante e um objectivo. Os algoritmos baseados em modelos matemáticos, em particular na area de investigação operacional, permitem eliminar a subjectividade numa solução e, conseqüentemente, poupar dinheiro, tempo, recursos e aumentar a produtividade. Problemas normalmente encontrados em casos reais são tipicamente de difícil resolução uma vez que a quantidade de informação fornecida e requerida para o problema é elevada. O progresso tecnológico e o aumento da capacidade de computação permitem, uma rápida resolução de problemas cada vez mais complexos. Este aumento da capacidade computacional permite a integração de vários problemas, com o objectivo de obter uma melhor solução global. Problemas que eram resolvidos separadamente na indústria, podem agora ser resolvidos de forma integrada, uma vez que é possível

em tempo útil obter uma solução que permita responder rapidamente a um acontecimento não programado. Com este objectivo, métodos de resolução, tanto exactos como heurísticos, para estes problemas têm sido desenvolvidos, permitindo encontrar soluções rapidamente.

O objectivo desta tese é contribuir para a resolução de problemas integrados que estão presentes em cadeias de abastecimento da indústria. São propostos modelos de programação inteira, algoritmos exactos, heurísticas e meta-heurísticas para a sua resolução, com o objectivo de obter resultados satisfatórios em tempo útil para instâncias de tamanho considerável.

A procura pela satisfação do cliente tem vindo a ser um factor relevante nas decisões das empresas. Este facto dá origem à conjugação de problemas existentes com a procura de satisfação dos clientes. O problema de corte e empacotamento conjugado com datas de entrega, havendo penalidades pela entrega posterior à requerida pelo cliente, é um problema integrado de corte e empacotamento. Este problema foi recentemente abordado por Reinertsen e Vossen (2010) onde foram propostos algoritmos para a sua resolução. Mais recentemente Arbib e Marinelli (2014) implementaram um algoritmo de programação inteira exacto que utiliza geração de colunas e desigualdades válidas na resolução do problema. Este problema é abordado nesta tese sendo os métodos propostos comparados com o proposto por Arbib e Marinelli (2014).

O escalonamento de tarefas em máquinas tem um impacto importante na indústria, uma vez que a optimização deste problema pode reflectir-se numa redução tanto em termos monetários como de recursos utilizados. Estas tarefas podem ter prazos de entrega definidos pelos clientes, muitas vezes com penalidades pela sua realização antecipada ou posterior à data de entrega, havendo tarefas a executar em diferentes períodos de tempo. Neste caso, está-se perante o problema integrado de planeamento e escalonamento em máquinas paralelas. Esta tese tem como objectivo contribuir para a resolução do problema integrado de planeamento e escalonamento estudado recentemente por Kis e Kovács (2012). Será abordado em particular o problema onde existem máquinas paralelas e idênticas. A parte de planeamento consiste em determinar os períodos de tempo (tipicamente semanas) em que um determinado conjunto de tarefas deve ser processado, enquanto que a parte de afectação consiste em atribuir as

tarefas às máquinas disponíveis em cada período de tempo (numa base diária). Diferentes abordagens foram propostas para resolver problemas semelhantes. O trabalho realizado nesta tese é original, no sentido em que é proposto um modelo de programação inteira baseado numa formulação pseudo-polinomial diferente, que será resolvido de acordo com a taxonomia introduzida por Maravelias e Sung (2009). O modelo é resolvido de modo exacto, sem recorrer a qualquer método de decomposição que explora a sua estrutura. A abordagem apresentada será comparada exaustivamente com os resultados relatados em Kis e Kovács (2012). A principal contribuição desta tese baseia-se na descrição de um novo modelo detalhado com base numa formulação de fluxos em rede pseudo-polinomial para este problema integrado de planeamento e escalonamento.

A grande variedade de casos práticos na área dos transportes tem originado a procura da integração de problemas de transportes com outros problemas na cadeia de abastecimento. Embora a complexidade dos problemas de transportes não permita, em muitos casos, a integração de vários problemas o recente avanço tecnológico tem vindo a permitir solucionar os problemas utilizando algoritmos de programação inteira ou heurísticos. Nesse sentido, é proposto nesta tese um modelo de programação inteira capaz de resolver o problema integrado de encaminhamento de veículos com múltiplas rotas e gestão integrada de inventários. Este problema caracteriza-se por ter um conjunto de clientes que têm necessidades de produtos, com determinado volume, em diferentes períodos de tempo. Estes clientes encontram-se em diferentes locais, separados do armazém por uma dada distância. Existe um conjunto de veículos que terão que realizar as entregas aos clientes satisfazendo as suas procuras para os períodos. As entregas poderão ser efectuadas para períodos posteriores, com uma penalidade de custo de armazenagem. Um conjunto limitado de veículos com determinada capacidade poderá visitar um conjunto de clientes, retornar ao armazém e visitar um novo conjunto de clientes, desde que não ultrapasse o limite de distância percorrida. A solução do problema define as rotas de cada veículo para cada período, assim como as mercadorias a serem entregues em cada período.

Embora os problemas integrados sejam tipicamente complexos, a sua vantagem relativamente à sua resolução de forma separada traz vantagens competitivas para as

empresas, o que faz com que estes ganhem cada vez mais relevância. A crescente procura por algoritmos mais rápidos e eficientes motiva a investigação de novos algoritmos para a resolução de problemas integrados na cadeia de abastecimento.

1.2 Objectivos de investigação

O propósito geral desta tese é de encontrar algoritmos capazes de resolver problemas integrados, que são muitas vezes aplicados em casos reais na indústria. Para isso são propostos modelos de programação inteira e também heurísticas para a resolução dos problemas integrados de corte e empacotamento, de planeamento e escalonamento e de encaminhamento de veículos e gestão de inventário. Os objectivos detalhados para os problemas abordados são os seguintes:

- Desenvolvimento e análise teórica e computacional de novos modelos de programação inteira para o problema integrado de corte e empacotamento.
- Desenvolvimento de algoritmos de resolução baseados nos modelos desenvolvidos no ponto anterior.
- Implementação e teste dos algoritmos de resolução propostos. Análise do desempenho dos algoritmos através de testes computacionais extensivos.
- Pesquisa de métodos de reforço ou algoritmos capazes de fornecer soluções para casos em que as instâncias são de difícil resolução.
- Desenvolvimento de novos modelos de Programação Inteira para o problema integrado de planeamento e escalonamento em máquinas paralelas.
- Desenvolvimento de algoritmos de resolução baseados nos modelos desenvolvidos no ponto anterior.
- Implementação, teste e análise computacional dos algoritmos de resolução propostos.
- Revisão de literatura relativa a problemas integrados de encaminhamento de veículos e gestão de inventários.

- Análise teórica e experimental de modelos de Programação Inteira para estes problemas tendo em conta as características do contexto industrial onde ocorrem como a natureza das cadeias de abastecimento ou o tipo de transportes em causa.
- Desenvolvimento, implementação e teste para os modelos propostos.

1.3 Conteúdo da tese

Esta tese foi estruturada em 6 capítulos que englobam os objectivos de investigação acima enunciados:

- O conteúdo e motivação desta tese, assim como os objectivos de pesquisa são apresentados no capítulo actual (**Capítulo 1**). A estrutura desta tese é também descrita.
- No **Capítulo 2** é realizada uma revisão bibliográfica relativa aos problemas de planeamento e gestão integrada de operações, com foco principal nos problemas integrados de corte e empacotamento, de planeamento e escalonamento e de encaminhamento de veículos e gestão de inventário. Várias variantes destes problemas são abordadas superficialmente, assim como modelos e métodos de resolução.
- O problema integrado de corte e empacotamento é descrito no **Capítulo 3**. Neste capítulo são descritos alguns modelos da literatura que resolvem este problema. São propostos vários novos modelos de programação inteira, métodos exactos e heurísticos destacando-se o modelo de fluxo em rede pseudo-polinomial. Vários métodos de reforço dos modelos foram propostos, nomeadamente a heurística baseada no modelo de fluxos em rede. Os resultados dos testes computacionais realizados são apresentados no final do capítulo.
- No **Capítulo 4** é abordado o problema integrado de planeamento e escalonamento em máquinas paralelas. Para resolver este problema é proposto um modelo de programação inteira baseado no modelo de fluxos em rede pseudo-polinomial. São também descritos alguns métodos de redução de arcos com

objectivo de remover a simetria no modelo. Os testes computacionais com instâncias da literatura são comparados com os resultados obtidos por outros autores. Algumas abordagens heurísticas são descritas no mesmo capítulo, com o objectivo de resolver instâncias difíceis que o modelo de programação inteira não foi capaz de resolver. Os resultados destas heurísticas são também apresentados no final do capítulo.

- O problema de encaminhamento de veículos com múltiplas rotas e gestão integrada de inventários é apresentado no **Capítulo 5**. Para a resolução deste problema será descrito um modelo pseudo-polinomial baseado no modelo de fluxos em arcos. Detalhes da implementação são apresentados, assim como testes computacionais com instâncias geradas para várias combinações de parâmetros.
- Por fim, no **Capítulo 6**, são apresentadas umas considerações finais sobre o trabalho realizado, e indicam-se pistas para trabalho futuro.

Capítulo 2

Planeamento e gestão integrada de operações: um estado da arte

Conteúdo

2.1	Introdução	8
2.2	Problemas integrados de corte e empacotamento	14
2.2.1	Problemas de corte e empacotamento	14
2.2.2	Problemas integrados de corte e empacotamento	17
2.3	Problemas integrados de planeamento e escalonamento	20
2.3.1	Escalonamento em máquinas paralelas	20
2.3.2	Problemas integrados de escalonamento	21
2.3.3	Problemas não integrados relacionados	26
2.4	Problemas de encaminhamento de veículos e gestão de inventários . . .	28
2.4.1	Problemas de encaminhamento de veículos	28
2.4.2	Encaminhamento de veículos e gestão integrada de inventários .	31
2.5	Conclusões	34

2.1 Introdução

O aumento da competitividade entre as empresas e a procura constante por parte das mesmas pela redução dos custos globais, obrigam à procura constante de novos métodos de resolução eficazes para diversos problemas de optimização que ajudam na tomada de decisões.

O desenvolvimento de métodos eficientes de optimização confronta-se com dois grandes desafios: a dimensão dos problemas reais e a adequação dos problemas que são resolvidos à definição real dos problemas industriais. No domínio das cadeias de abastecimento, estes desafios são motivados por factores como a globalização ou o grau de integração entre os diferentes agentes. As empresas enfrentam pressões constantes para reduzir os seus custos e melhorar o seu serviço. Em consequência, são forçadas a manter baixos níveis de inventários ao longo da cadeia de abastecimento, a aumentar a sua capacidade de resposta, e a melhorar globalmente o seu desempenho operacional. Em todas as áreas funcionais de uma empresa, desde a compra de matéria prima, à produção até à entrega ao cliente, exige-se um alto grau de desempenho. Cada vez mais as empresas tentam interligar todas estas áreas funcionais com o objectivo de obter um melhor aproveitamento dos recursos, assim como a redução de custos. Para atingir esses objectivos, é fundamental que o planeamento de operações seja feito de forma integrada e coordenada ao longo das diferentes funções das empresas (Grossmann e Furman, 2009, Armstrong et al., 2008), uma vez que isto traz uma gestão dos recursos mais eficiente do que aquela conseguida através de optimizações locais dos problemas. Por seu lado, a optimização em tempo real permite responder à variabilidade que caracteriza qualquer sistema real. A optimização em tempo real envolve a re-optimização dos planos de operação com uma frequência muito elevada, de modo a considerar eventos não previstos e novos dados que só são conhecidos na fase de execução. O objectivo é reduzir os ciclos de planeamento de modo a incorporar novas informações em tempo real, e responder assim de forma eficiente a novos pedidos sem com isso comprometer o serviço aos clientes. Embora dependendo da dimensão da empresa, esta tarefa pode não ser trivial. Em casos de empresas que lidam diariamente com dezenas de encomendas ou processam centenas ou milhares

de produtos diferentes por dia (*p. ex.* indústria automóvel ou têxtil), tendo cadeias de abastecimento muito complexas, é necessário encontrar ferramentas que ajudem a lidar com estes problemas. Com a modernização tecnológica, têm surgido no mercado ferramentas que ajudam neste processo, embora nem sempre sejam as mais eficientes. *Softwares* de gestão normalmente não têm ferramentas de optimização e fazem unicamente a ligação entre departamentos. A crescente concorrência entre empresas obriga-as a procurar novos métodos que permitam obter vantagem competitiva em relação às suas concorrentes. Desta forma, além da necessidade do controlo interno da empresa, algumas vêm-se forçadas a interligar os seus processos com os seus fornecedores ou os seus clientes. Isto torna o controlo da cadeia de abastecimento muito mais complexo mas tipicamente mais vantajoso para estes parceiros. Neste caso, estas empresas têm a necessidade de trocar informações constantes e há uma ligação forte de confiança de forma a ser possível optimizar recursos das empresas parceiras. A Figura 2.1 mostra um exemplo de cadeia de abastecimento, desde a fase de investigação e desenvolvimento de um produto até à sua entrega ao cliente.

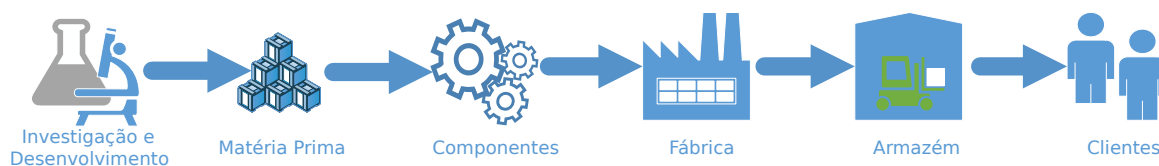


Figura 2.1: Cadeia de abastecimento genérica

Neste caso, é possível reconhecer a dimensão do problema no momento da sua integração com os vários departamentos ou estados de fabrico de um produto. A integração de todo o processo da cadeia de abastecimento num modelo de optimização tem como objectivos possíveis o melhor aproveitamento dos recursos ou a determinação das necessidades de materiais, muito embora a satisfação do cliente final seja sempre um objectivo ponderado em todo o processo. Na Figura 2.2 pode observar-se a integração de dois possíveis problemas: uma empresa tem que realizar o escalonamento da sua produção e organizar a entrega do seu produto aos clientes. Ao mesmo problema pode juntar-se a gestão das encomendas de matéria prima para a produção.

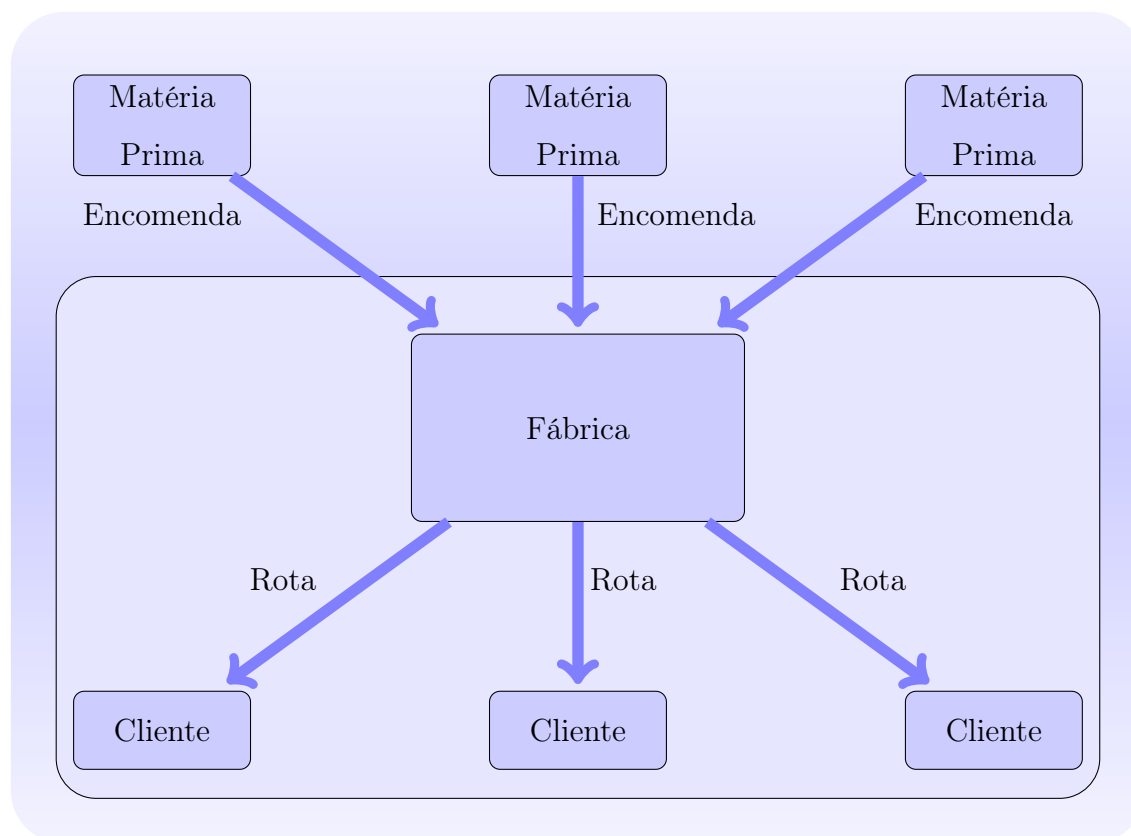


Figura 2.2: Representação genérica da integração de problemas na indústria

Apesar de tradicionalmente a produção e entrega serem efectuadas de forma separada (Li et al., 2016), sendo inicialmente programada a produção e posteriormente decidida a distribuição, os benefícios da optimização integrada de operações em tempo real são amplamente reconhecidos. Apesar da relevância prática destes problemas, a investigação em optimização em tempo real de problemas integrados ao nível operacional é muito recente (Armstrong et al., 2008, Bixby et al., 2006, Chang e Lee, 2004, Geismar et al., 2008, Durbin e Hoffman, 2008, Wang et al., 2005, Wang e Cheng, 2007, Wang e Lee, 2005), e foca-se essencialmente em métodos heurísticos. As abordagens desenvolvidas até ao momento não tiram partido dos progressos recentes das técnicas de Programação Inteira Mista. Estas técnicas, que têm beneficiado do progresso constante ao nível do hardware e software, têm sido aplicadas com sucesso na resolução de problemas de grande dimensão em vários sectores. Recentemente, vários autores descreveram aplicações bem sucedidas de métodos de programação inteira mista para

a resolução de problemas específicos de optimização integrada e em tempo real (Bixby et al., 2006, Durbin e Hoffman, 2008). Apesar desses avanços, ainda é necessário muito esforço de investigação para desenvolver métodos genéricos e eficientes de programação inteira mista que sejam capazes de responder aos diferentes desafios levantados por estes problemas.

No artigo de Grossmann e Furman (2009) são descritos vários exemplos da utilização de modelos de optimização para processos industriais em grande escala com problemas de escalonamento. Os autores defendem a necessidade crescente deste tipo de abordagem, dado que cada vez mais há a necessidade de utilizar ferramentas de apoio à decisão que possam analisar alternativas ou prever acções numa cadeia de abastecimento, tendo como objectivo a optimização global do sistema e também a satisfação do cliente. Neste artigo é possível encontrar vários casos descritos de modelos de programação inteira linear e não linear para a resolução de problemas de grandes dimensões na indústria, com grande relevo para a indústria petrolífera. Para resolver um problema integrado com distribuição e zero-inventário, em que a vida útil do produto é curta, Armstrong et al. (2008) utilizam heurísticas para fortalecer o limite inferior para acelerar a procura da solução na árvore de pesquisa. Tendo o produto (adesivo para painéis de madeira prensada) um ciclo de vida curto, e havendo apenas um veículo para realizar as entregas aos clientes, o objectivo é de maximizar o número de encomendas entregues. Uma vez que o produto considerado tem um ciclo de vida de apenas sete dias, Armstrong et al. (2008) consideram que não é permitido a existência de inventário. Um problema semelhante de escalonamento na produção e distribuição foi abordado por Abdul Rahim et al. (2014), e resolvido através de heurísticas e métodos exactos. Um trabalho idêntico foi realizado por Chen e Vairaktarakis (2005), onde os autores utilizam um modelo integrado numa cadeia de abastecimento de catering e também numa empresa de computadores. O objectivo é de reduzir ou eliminar o inventário, de forma a poder satisfazer os clientes no menor período de tempo possível. Os autores desenvolveram algoritmos exactos e heurísticas que demonstram obter soluções muito próximas da solução óptima. Nos modelos descritos em Chen e Vairaktarakis (2005), é integrado o escalonamento da produção das tarefas realizadas na indústria de processamento com a distribuição e entrega ao

cliente. Os testes computacionais demonstraram que em muitos casos há um benefício significativo com a integração destes problemas.

Para resolver um problema integrado de escalonamento e de distribuição de correio, Wang et al. (2005) utilizaram heurísticas com o objectivo de determinar a sequência em que o correio deverá ser entregue, de forma que a capacidade não utilizada do veículo seja minimizada. Deste modo, os autores definiram o problema como possuindo uma única máquina.

O artigo de Chang e Lee (2004) aborda o problema integrado de escalonamento da produção e posterior entrega do produto ao cliente, tendo em especial atenção a quantidade de espaço necessário durante a sua entrega. A função objectivo do seu problema considera a minimização do tempo de realização de todas as tarefas. Diversas variantes do problema foram abordadas por Chang e Lee (2004), que utilizaram heurísticas para resolver o problema em que existe uma única máquina para executar as tarefas consideradas, e um único veículo para entregar os produtos acabados, generalizando depois para o problema com máquinas paralelas. Um trabalho na mesma área foi desenvolvido por Li et al. (2005) que, recorrendo a um modelo de programação dinâmica, resolvem o problema especial que considera um único cliente, e generalizam depois para o caso com vários clientes e várias encomendas. Para diminuir a complexidade do problema, os autores consideram que as entregas deverão ser directas e que os veículos não possuem limite de capacidade. No artigo de Wang e Cheng (2007) é resolvido o problema integrado de escalonamento em máquinas paralelas com entrega, e como em Chang e Lee (2004) e Li et al. (2005) o problema tem como objectivo reduzir o tempo entre o processo de fabrico e as entregas ao cliente.

Durbin e Hoffman (2008) descrevem um conjunto de modelos de optimização que servem de base a uma ferramenta de apoio à decisão, utilizada para resolver um problema real da indústria cimenteira. A Figura 2.3 representa o modelo existente na empresa (a) e o novo modelo proposto (b). Como é possível verificar, o problema tornou-se muito mais complexo com a implementação do modelo proposto, uma vez que passa a ser possível haver transportes entre vários armazéns e vários clientes, o que não acontece em (a) onde um único veículo teria que efectuar a entrega a um único cliente, tendo que voltar ao armazém se necessário para efectuar uma segunda

entrega. Neste problema, foram consideradas várias restrições de transporte, assim como tempos de chegada da mercadoria ao cliente e também o horário de chegada dos condutores para realizar o transporte.

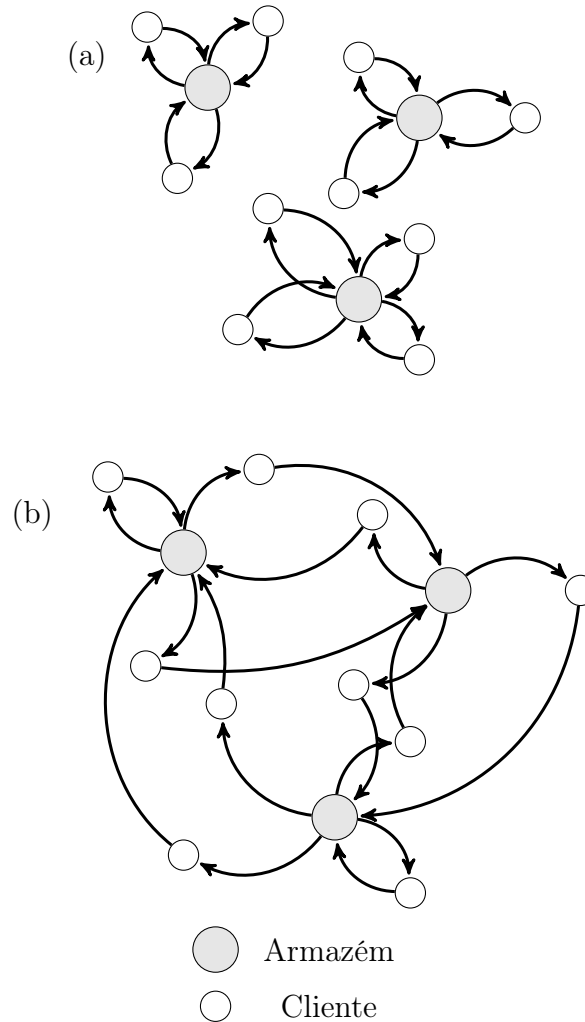


Figura 2.3: (a) - Envio baseado no transporte, (b) - Envio baseado na procura, Exemplo de um conjunto de clientes e armazéns que devem satisfazer a procura dos clientes, Adaptado de Durbin e Hoffman (2008).

Com o modelo proposto por Durbin e Hoffman (2008) foi possível obter intervalos médios de optimalidade de cerca de 1%. Li et al. (2016) recorreram ao uso de um algoritmo genético para resolver o problema integrado de produção numa máquina e entrega num ou múltiplos veículos. O problema considerado é multi-objectivo: minimizar o custo total das entregas e reduzir o tempo de espera dos clientes. Os autores (Li et al., 2016) definem que na solução haverá um conjunto de encomendas

a serem produzidas de forma a que todas possam caber no veículo e a que a rota que serve os clientes tenha o menor custo possível. Com esta abordagem foi possível obter soluções de boa qualidade em tempo razoável.

Um problema integrado de escalonamento da produção com transporte foi abordado por Wang e Lee (2005). Neste problema, um conjunto de produtos têm que ser produzidos e depois transportados para os clientes. Os autores abordam o escalonamento em máquinas paralelas e a escolha de diferentes modos de transporte. Diferentes abordagens para o problema foram propostas, sendo que a função objetivo considera a minimização do custo total de transporte, assim como dos atrasos na produção. Os autores Wang e Lee (2005) propuseram um algoritmo de partição e avaliação sucessivas, e um modelo matemático que resolvem com um *solver* comercial.

Embora complexa, a resolução de problemas integrados pode trazer grandes vantagens competitivas uma vez que fornece uma visão global de vários processos. A integração destes problemas poderá reduzir custos globais ao permitir melhorar o aproveitamento dos recursos, uma vez que os problemas são vistos de forma integrada e a tomada de decisão é mais precisa.

Assim, tem-se observado um aumento de contribuições ao nível da otimização de problemas que eram tradicionalmente decompostos e resolvidos separadamente. Isto tornou-se também possível graças aos avanços em termos de métodos de resolução, tanto heurísticos como exatos e, por outro lado, à crescente capacidade de resolução dos *solvers* comerciais de programação linear e inteira.

2.2 Problemas integrados de corte e empacotamento

2.2.1 Problemas de corte e empacotamento

O problema de corte e empacotamento é um problema \mathcal{NP} -difícil, que tem sido alvo de bastantes estudos devido à sua aplicação prática na indústria. Este problema consiste em determinar padrões de corte de forma a obter um conjunto de objetos (itens) I , com procuras b_i , $i \in I$, a partir de objetos maiores (rolos), de modo a reduzir os desperdícios, minimizando a quantidade de rolos usados. Este tipo de

problema poderá ser aplicado em várias indústrias como a indústria têxtil, de corte de aço, bobines, pele, papel, madeira ou vidro (Reinertsen e Vossen, 2010). Diferentes variantes deste problema foram já abordadas na literatura, e incluem problemas em uma (Alves e Valério de Carvalho, 2008), duas (Malaguti et al., 2014, Silva et al., 2010, Macedo et al., 2010) ou três dimensões (Martello et al., 2000, Chen et al., 1995). Este problema é muito similar ao problema de *bin-packing*, diferenciando-se apenas pelo facto do problema de corte e empacotamento ter tipicamente uma menor variedade de tipos de itens. Assim, considera-se que instâncias cujas procuras sejam globalmente muito baixas (próximas de 1) são instâncias do problema de empacotamento (*bin-packing*), enquanto que instâncias cujas procuras são mais elevadas são instâncias do problema de corte.

A necessidade de uma definição mais precisa dos problemas de corte foi sugerida por Dyckhoff (1990), que esquematizou e uniformizou as diferentes nomenclaturas da literatura da área de corte. O autor identifica diferentes características de problemas que agrupa em quatro tipos de problema. A divisão é efectuada segundo a dimensão do problema, tipo de afectação e ordenação dos itens. Posteriormente, Wäscher et al. (2007) propôs uma melhoria desta tipologia devido à crescente diversidade de problemas de corte. No artigo de Wäscher et al. (2007) os autores definem em mais detalhe as diferenças nos problemas, para fazer face ao crescente trabalho na área do corte e às crescentes necessidades por parte da indústria que foram obrigando à procura de novos problemas. Desta forma os autores identificam quais as áreas mais estudadas e aquelas para as quais há pouco trabalho desenvolvido.

Uma variante bem conhecida deste problema é o problema de corte e empacotamento com prazos de entrega. Este caracteriza-se por um conjunto de itens com uma procura e um prazo de entrega definidos, a ser cortados a partir de um conjunto de rolos da mesma dimensão (de quantidade infinita). O objectivo deste problema consiste em minimizar o uso de rolos e os atrasos de cortes de itens posteriores à data de entrega, estando-se assim perante um problema multi-objectivo.

O problema de corte e empacotamento tem vindo a ser alvo de diferentes abordagens de resolução, incluindo tanto técnicas heurísticas, que obtêm bons resultados em curtos espaços de tempo, como métodos exatos, cujos tempos de computação são ti-

picamente bastante superiores. No artigo de Valério de Carvalho (2002) são descritas várias abordagens de métodos de programação inteira para a resolução de problemas de corte e empacotamento. Neste último, são descritos modelos de fluxos em rede, capazes de resolver o problema de corte e empacotamento com rolos de diferentes tamanhos. Este modelo foi inicialmente proposto por Valério de Carvalho (1999) e foi posteriormente adaptado em Alves e Valério de Carvalho (2008) para problemas de corte e empacotamento com diferentes dimensões de rolos. A Figura 2.4 representa um grafo de fluxos em rede gerado a partir de uma instância retirada de um exemplo do artigo de Valério de Carvalho (2002), onde existem três itens de tamanhos $w = (4, 3, 2)$ e três tamanhos diferentes de rolos com dimensões $W = (9, 6, 5)$. Este grafo é representativo de um problema de corte e empacotamento com rolos de várias dimensões. No artigo de Brandão e Pedroso (2016), os autores utilizam a compressão do grafo para obter resultados mais rápidos que o modelo original de Valério de Carvalho (1999).

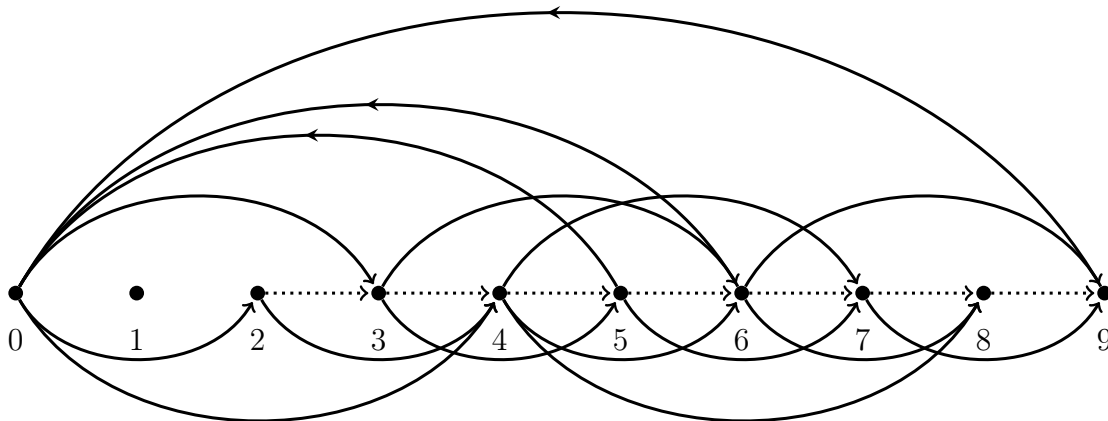


Figura 2.4: Grafo de fluxos gerados por uma instância de três itens, retirado de Valério de Carvalho (2002)

Um trabalho recente de Delorme et al. (2016) apresenta uma revisão da literatura sobre os mais importantes algoritmos utilizados para resolver o problema de corte. No seu artigo, Delorme et al. (2016) focam-se nos modelos mais eficientes da literatura a uma dimensão. Tendo abordado também métodos heurísticos, os autores testam

os modelos de programação inteira mais eficientes, comparando os seus resultados computacionais. A Tabela 2.1 resume os resultados dos algoritmos (mais importantes) testados por Delorme et al. (2016). Nos testes efectuados, o modelo que resolveu com mais eficiência as instâncias testadas foi o modelo de fluxos com compressão do grafo de Brandão e Pedroso (2016). O modelo de fluxos de Valério de Carvalho (1999) obtém resultados próximos de Brandão e Pedroso (2016). A diferença deve-se ao facto da compressão do grafo reduzir a simetria e o número de variáveis, tornando a resolução do modelo mais eficiente.

Algoritmo	Modelo	Autor
Partição e avaliação	MTP	Martello e Toth (1990)
	BISON	Scholl et al. (1997)
	CVRPSEP	Lysgaard et al. (2004)
Partição e geração de colunas	VANCE	Vance et al. (1994)
	BELOV	Belov e Scheithauer (2006b)
	SCIP-BP	Ryan e Foster (1981)
Pseudo-polinomial	ONECUT	Rao (1976), Dyckhoff (1981), Stadtler (1988)
	ARCFLOW	Valério de Carvalho (1999)
	DPFLOW	Cambazard e OSullivan (2010)
	VPSOLVER	Brandão e Pedroso (2016)
Outros	BASICILP	Martello e Toth (1990)
	CSTRPROG	Shaw (2004)

Tabela 2.1: Modelos para o problema de corte e empacotamento (Delorme et al., 2016)

2.2.2 Problemas integrados de corte e empacotamento

Embora o problema clássico de corte e empacotamento seja por si só de difícil resolução, situações reais encontradas por exemplo na indústria, fazem surgir novas variantes com características especiais, tipicamente de ainda maior complexidade.

Dell’Amico et al. (2012) aborda o problema de empacotamento (*bin packing*) com precedências (BPP-P), cujo objectivo é minimizar o número de rolos utilizados, sem

violam as relações de precedência entre os itens. A abordagem apresentada corresponde ao primeiro método exacto proposto na literatura para o problema de empacotamento com precedências. Os autores (Dell’Amico et al., 2012) propõem critérios de redução, limites inferiores, limites superiores baseados em métodos de pesquisa em vizinhança variável, e um algoritmo de partição e avaliação sucessivas. O mesmo problema foi abordado por Pereira (2016), onde foi proposta uma heurística baseada em métodos de programação dinâmica e um método exato que corresponde a um procedimento de enumeração modificado.

Um problema integrado de corte e empacotamento foi descrito por Arbib e Marinelli (2005) onde o objectivo consiste na redução de custos de produção, inventário e transporte. Este advém de um problema real da indústria do corte tendo o modelo proposto obtido uma redução de cerca de 43% do custo total.

Tipicamente, nos problemas de corte usuais, a ordem em que cada corte é efectuado é irrelevante uma vez que o objectivo é de se obter o menor número de peças de material a utilizar, minimizando-se assim os desperdícios. Contudo, devido à crescente concorrência no sector da indústria, a satisfação do cliente tem vindo a ser colocada em relevo. A consideração do factor do prazo de entrega definido pelo cliente gera uma nova variante do problema, o problema de corte e empacotamento com prazos de entrega, em que a ordem em que os cortes são efectuados tem relevância para o problema.

O problema de corte com prazos de entrega é um problema multi-objectivo, uma vez que o objectivo é o de minimizar o material despendido para satisfazer os pedidos e reduzir os atrasos dos prazos de entrega. Tipicamente as duas partes possuem o mesmo peso na função objectivo, o que significa que o uso de uma unidade de um rolo tem o mesmo peso que uma unidade de atraso. Contudo, alguns autores (Reinertsen e Vossen, 2010, Arbib e Marinelli, 2014) propõem o uso de diferentes pesos.

Li (1996) propôs heurísticas baseadas em modelos de programação linear e não linear para resolver o problema de corte e empacotamento a duas dimensões, com datas de lançamento e de entrega. Um algoritmo genético foi desenvolvido por Bennell et al. (2013) para o mesmo problema apenas com datas de entrega.

Este problema, mas a uma dimensão, foi também abordado por Reinertsen e Vossen (2010) que propuseram métodos capazes de resolver instâncias de tamanho industrial.

Em Arbib e Marinelli (2014), os autores propõem um modelo de programação inteira para o problema de corte e empacotamento unidimensional, com datas de entrega. Este modelo afecta padrões de corte a períodos de tempo de determinada duração. Dado o número exponencial de padrões de corte válidos e respectivas afecções aos períodos de tempo, os autores propõem um método de geração de colunas e um algoritmo de refinamento que actualiza iterativamente a duração dos períodos de tempo.

Bendraouche e Boudhar (2012) propõem heurísticas e limites inferiores para o problema de afectação de funcionários a tarefas em máquinas paralelas, tratado como um problema de corte e empacotamento unidimensional, com o objectivo de minimizar o tempo de execução das mesmas. Do mesmo modo, este problema pode ser aplicado na resolução de problemas em sistemas de computação (Baker e Coffman, 1996, Haned et al., 2012).

Halldórsson et al. (2003) referem que o problema de afectação de tarefas pode ser aplicado na distribuição de tráfego em redes de comunicação e em sessões de redes locais. A este problema podem ser adicionadas penalidades se os itens forem produzidos antes das respectivas datas de entrega, que correspondem a custos de inventário (Lasserre, 1992). Shabtay e Karhi (2012) abordam duas variantes do problema de produção e escalonamento em máquinas paralelas, uma variante *online* e *offline*. No problema *offline*, toda a informação está disponível antecipadamente, enquanto que no problema *online* podem considerar-se novas encomendas durante o processo de resolução, tornando naturalmente o mesmo mais difícil. Kis e Kovács (2012) abordam o problema integrado de escalonamento da produção em máquinas paralelas, propondo um algoritmo de partição e avaliação sucessivas com cortes. Lenstra et al. (1990) desenvolveram um algoritmo capaz de resolver o problema com um número arbitrário de máquinas.

2.3 Problemas integrados de planeamento e escalonamento

2.3.1 Escalonamento em máquinas paralelas

O problema de planeamento e escalonamento da produção em máquinas paralelas é caracterizado por ser um problema *NP-difícil*. Neste problema, existe um dado número de encomendas a produzir numa máquina e um conjunto de máquinas idênticas, tendo cada encomenda um prazo de entrega específico. O objectivo é minimizar o desperdício de material a utilizar e o tempo despendido a realizar as operações. Um planeamento e escalonamento eficientes são essenciais numa cadeia de abastecimento ou qualquer outra área funcional de uma empresa. O escalonamento poderá ser realizado em vários momentos na cadeia de abastecimento, muitas vezes de forma integrada. Vários exemplos de aplicações de problemas de escalonamento estão descritos em Pinedo (2005). No seu livro, o autor aborda diferentes métodos de optimização para definir o escalonamento em todo o processo industrial, desde a fase de compra até ao produto final ser entregue ao cliente.

Muito embora existam diversas variantes deste problema, tipicamente o objectivo é determinar em qual máquina, e em que ordem será realizado um determinado número de tarefas. Algumas abordagens para este problema e suas variantes foram descritas por Pinedo (2012).

Nos problemas de escalonamento em máquinas paralelas, pode considerar-se que estas poderão ser idênticas ou diferentes. No caso das máquinas serem diferentes, a dificuldade de resolução aumenta, uma vez que um novo grupo de restrições é adicionado para diferenciar as mesmas.

Outras variantes deste problema consideram prazos de entrega para as tarefas (podendo haver penalidades por execução das mesmas antes ou depois dos respectivos prazos) ou mesmo precedências entre tarefas.

Esta classe de problemas tem aplicações nos mais diversos contextos: determinar a produção para um conjunto de fábricas idênticas (Maravelias e Sung, 2009), atribuir funções a trabalhadores em paralelo (Bendraouche e Boudhar, 2012), aplica-

ções relativas a sistemas de computação (Baker e Coffman, 1996, Haned et al., 2012), distribuição de tráfego em redes (Halldórsson et al., 2003), entre outros.

Diversos métodos de resolução para esta classe de problemas, tanto exactos como heurísticos, foram já propostos na literatura (Senthilkumar, 2010, Chen et al., 1999, Pinedo, 2005, 2012, Westerlund e Isaksson, 1998, Giannelos e Georgiadis, 2001). Em Senthilkumar (2010), é proposta uma revisão da literatura de problemas relacionados com o escalonamento em máquinas paralelas e uniformes.

Um estudo de Maravelias e Sung (2009) sobre o planeamento da produção a médio prazo e escalonamento a curto prazo descreve várias abordagens para o problema. Os autores abordam a necessidade de integração da produção no escalonamento, justificando-a com a necessidade das fábricas responderem à flutuação da procura e à satisfação dos clientes. Alguns exemplos de abordagens de escalonamento para este problema são fornecidos pelos autores (Maravelias e Sung, 2009), e a sua utilidade para a indústria é demonstrada através de alguns exemplos de problemas e diferentes funções objectivos. Embora possam existir diferentes abordagens para o problema, tipicamente a função objectivo é a minimização do *makespan* (quantidade de tempo necessária para completar todas as tarefas), assim como a minimização dos custos de atrasos e avanços das tarefas.

2.3.2 Problemas integrados de escalonamento

A integração de vários problemas de escalonamento tem vindo a ser abordada nos últimos anos e aplicada a casos reais na indústria, gerando vantagens competitivas adicionais.

Um exemplo prático de aplicação de um problema integrado de escalonamento é descrito em Westerlund e Isaksson (1998), onde os autores o abordaram na indústria de corte de papel com o objectivo de otimizar o desempenho reduzindo o desperdício efectuado pelos cortes. O problema consiste em, por um lado reduzir o desperdício efectuado nos padrões de corte, e por outro escalonar esses padrões a máquinas. Um problema similar na indústria do papel é descrito em Giannelos e Georgiadis (2001), com o objectivo de determinar que padrões de corte efectuar num conjunto de máquinas paralelas, reduzindo o desperdício. Os autores propõem uma heurística

para resolver o problema. Num estudo mais recente (Arbib et al., 2012), também na área do corte, é utilizada a pesquisa tabu para resolver um problema de minimização de desperdícios resultantes do corte e do tempo de operação.

A otimização integrada das operações nas empresas contribui para reduzir os custos e melhorar o seu desempenho. Por isso, muitos autores têm apoiado a ideia de uma maior coordenação entre planeamento e escalonamento, desde a aquisição de matéria prima até à entrega do produto final ao cliente (Grossmann e Furman, 2009). Alguns dos estudos descritos na literatura consideram a integração das operações ao longo de diferentes áreas funcionais. Em Armstrong et al. (2008), Chang e Lee (2004), Geismar et al. (2008), Wang e Cheng (2007), por exemplo, os autores abordam a otimização integrada de escalonamento de máquinas e entregas. Enquanto que o escalonamento das máquinas diz respeito à área de produção, as entregas estão relacionadas com a área da distribuição. No presente trabalho, explora-se o problema integrado de planeamento e escalonamento, em que todas as decisões são tomadas no âmbito da mesma área funcional, ou seja, na área de produção. O problema foi estudado recentemente por Kis e Kovács (2012).

Um dos principais objectivos do planeamento da produção a médio prazo é estabelecer metas específicas de produção para as instalações das empresas. Por outro lado, o escalonamento a curto prazo diz respeito à afectação das tarefas às unidades de produção e a sua consequente sequenciação, de modo a cumprir as metas definidas na fase de planeamento (Maravelias e Sung, 2009). A interligação entre planeamento e escalonamento baseia-se essencialmente nas metas de produção, que são simultaneamente os *outputs* da fase de planeamento e os *inputs* da fase de escalonamento. Estas metas de produção determinam tanto a viabilidade como a optimização dos escalonamentos finais.

Como consequência, a abordagem padrão que consiste em resolver os problemas de planeamento e escalonamento separadamente levanta essencialmente duas questões. As metas de produção podem ser impossíveis de satisfazer tendo em conta os recursos disponíveis, e mesmo quando os problemas de escalonamento são possíveis, a solução global de optimização em termos do seu custo de produção ou *makespan* dificilmente é alcançado. A única forma de superar esses problemas é resolver estes dois problemas

de uma forma coordenada.

Problemas integrados de planeamento e escalonamento são definidos ao longo de um horizonte de planeamento que é posteriormente dividido em períodos de tempo mais curtos. Conforme resumido em Maravelias e Sung (2009), estes problemas envolvem um conjunto de itens que têm de ser produzidos para satisfazer a procura dos clientes em determinados períodos de tempo. A procura poderá ser satisfeita, total ou parcialmente através de inventários. Neste caso, existem custos de armazenamento que se aplicam a cada item, e restrições que asseguram um equilíbrio entre níveis de produção, inventário e procura. A produção é também naturalmente alcançada sob restrições de recursos. As decisões estão relacionadas com as quantidades de itens que têm de ser produzidas e que existem em inventário em cada período de tempo, ao longo de todo o horizonte de planeamento. O objectivo é a minimização dos custos totais de inventário e de produção.

Maravelias e Sung (2009) dividiram as abordagens propostas na literatura para os problemas integrados de planeamento e escalonamento em três grupos. A primeira abordagem de modelação consiste na formulação do problema através de um modelo de escalonamento detalhado, usando variáveis de planeamento e escalonamento ligadas através de um conjunto de restrições, ou recorrendo a uma formulação monolítica em que as variáveis de planeamento são substituídas por variáveis de escalonamento. Um exemplo deste caso pode ser encontrado em Joly et al. (2002), para o planeamento da produção em refinarias de petróleo. Joly et al. (2002) formularam um modelo não-linear e um modelo inteiro misto que foram resolvidos directamente utilizando *solvers* de optimização. Uma formulação monolítica foi explorada recentemente por Kis e Kovács (2012). Os autores resolveram o problema directamente com um *solver* de optimização. Além disso, elaboraram uma abordagem de decomposição hierárquica que resolveram utilizando um algoritmo de partição e avaliação sucessivas com cortes.

Na Figura 2.5 está representada a classificação segundo Maravelias e Sung (2009) para os problemas integrados de planeamento e escalonamento. Como representado na figura, o modelo é decomposto num sub-problema mestre (alto nível) que é utilizado para determinar as metas de produção e num sub-problema escravo (baixo nível) que é utilizado para determinar o escalonamento detalhado. Se o fluxo de informação

é só no sentido do problema mestre para o sub-problema, então obtém-se um problema hierárquico (a). Se o fluxo de informação é nos dois sentidos, o problema é interactivo (b). Se o problema é resolvido sem recurso a métodos de decomposição, então trata-se de um problema de pesquisa global resolvido de forma integrada (c). A abordagem proposta nesta tese insere-se neste último grupo (c), uma vez que é resolvido o problema de forma integrada.

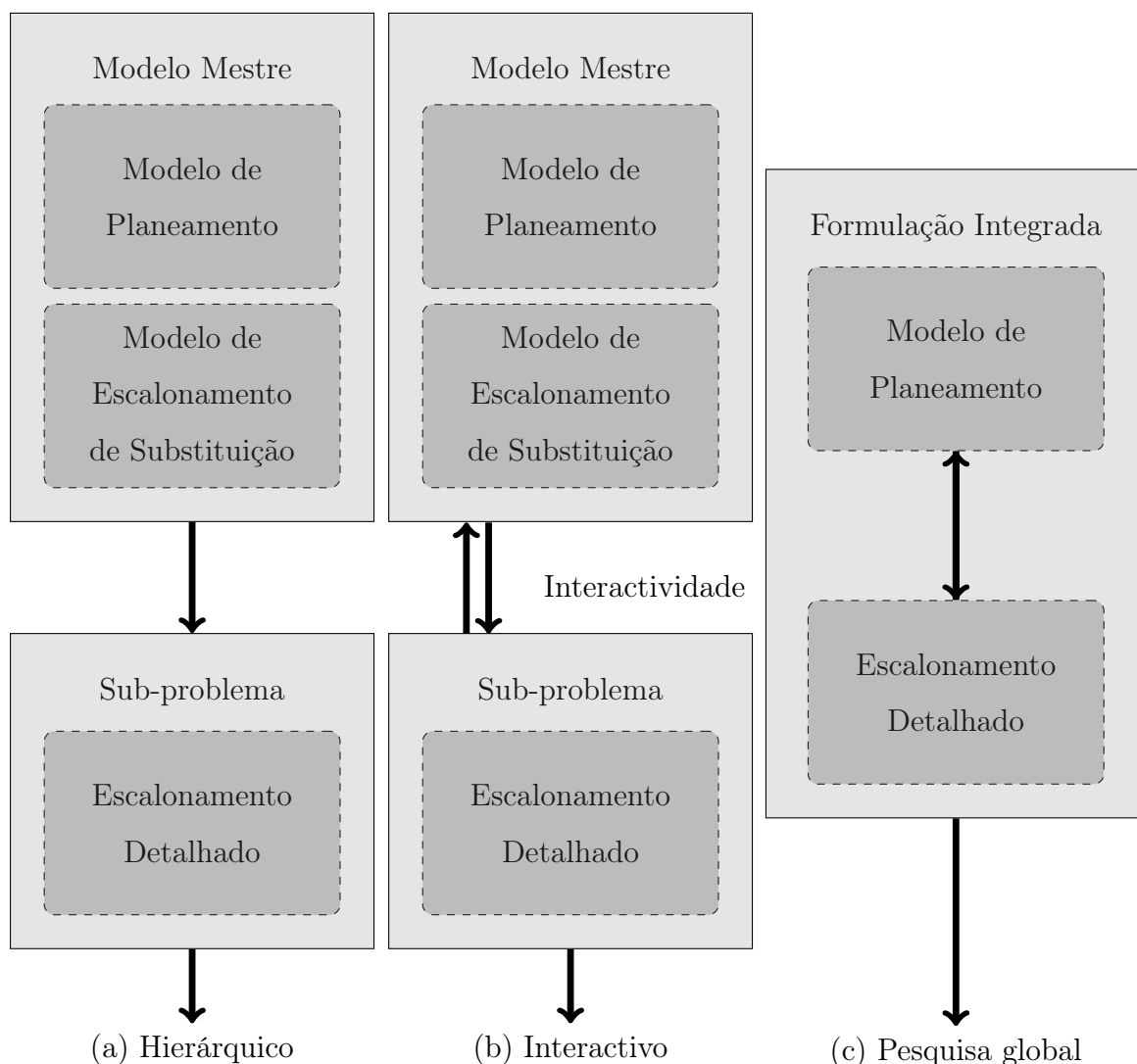


Figura 2.5: Estratégias para o problema integrado de planeamento e escalonamento, adaptado de Maravelias e Sung (2009). (a) problema resolvido de forma hierárquica, (b) problema resolvido através de métodos interactivos e (c) onde o problema é resolvido de forma integrada através da pesquisa global.

A contrapartida das abordagens anteriores é que os modelos aumentam significativamente de tamanho para problemas de grande escala, tornando a sua resolução até à optimalidade um verdadeiro desafio computacional.

Uma alternativa para fazer face à dificuldade de resolução destes problemas consiste em utilizar uma aproximação ao modelo original, quer considerando uma versão relaxada do modelo, quer agregando-o de forma a reduzir o seu número de variáveis e restrições (Maravelias e Sung, 2009). A utilização de relaxações foi considerada em Roe et al. (2005), Maravelias (2006), enquanto que exemplos de abordagens baseadas em modelos agregados podem ser encontrados em Bassett et al. (1996), Váncza et al. (2004).

Um segundo grupo de abordagens descritas em Maravelias e Sung (2009) baseia-se na computação *offline* de restrições que caracterizam o espaço de soluções válidas no modelo detalhado de escalonamento e na estimativa de custos de produção. Um exemplo deste tipo de abordagem pode ser encontrada em Sung e Maravelias (2007).

Os principais métodos propostos na literatura para problemas integrados de planeamento e escalonamento podem ser divididos em métodos de decomposição hierárquicos, métodos iterativos e métodos de pesquisa global (*full-space methods* na terminologia anglo-saxónica) (Maravelias e Sung, 2009).

Nos métodos hierárquicos, o problema é dividido num problema mestre e subproblemas de escalonamento de baixo nível. O problema mestre é usado, por exemplo, para determinar as metas de produção e o subconjunto de tarefas que será passado para os subproblemas de escalonamento. As restrições de escalonamento detalhadas não são consideradas no problema mestre. O trabalho descrito por Amaro e Barbosa-Póvoa (2008) ilustra esse tipo de abordagens. Métodos de horizonte rolante (*rolling horizon*) são considerados em Maravelias e Sung (2009) como abordagens de decomposição hierárquicas. A ideia é resolver os modelos de escalonamento detalhado para alguns dos períodos iniciais, enquanto o plano para os restantes períodos é determinado pela resolução de um modelo agregado com um menor nível de detalhe. Os dados de entrada são então actualizados para o horizonte de planeamento seguinte, e o processo é repetido. O principal problema com estes métodos hierárquicos é que pode ser impossível cumprir as metas de produção estabelecidas pelo problema

mestre. A alternativa é considerar um ciclo de *feedback* entre o problema mestre e os subproblemas de escalonamento para superar esta potencial inviabilidade ou sub-óptimalidade. Este *feedback* pode ser feito pela adição de desigualdades válidas para cortar soluções do problema mestre que levam a soluções inválidas nos subproblemas de escalonamento. Esta abordagem foi seguida por exemplo, em Kis e Kovács (2012). Em Li e Ierapetritou (2009) foi utilizada uma abordagem similar.

Em Erdirik-Dogan e Grossmann (2008), os autores exploraram o problema de planeamento e escalonamento de estágio único, multi-produtos e com unidades paralelas. O problema tem custos de transição e atrasos gerados por mudanças de um produto para outro. Dada a dificuldade em resolver o seu modelo de programação para horizontes de tempo longos, Erdirik-Dogan e Grossmann (2008) recorreram a um método de resolução iterativa que visa reduzir progressivamente o desvio de optimalidade do seu problema integrado de planeamento e escalonamento.

O terceiro grupo de métodos de solução identificadas por Maravelias e Sung (2009) são os chamados métodos de pesquisa global que consistem em resolver diretamente o modelo de escalonamento detalhado completo. Esta resolução pode ser feita utilizando métodos padrão como em Joly et al. (2002) ou utilizando heurísticas como em Yan e Zhang (2007). Note-se que essas abordagens são, de longe, as menos exploradas para a resolução de problemas integrados de planeamento e escalonamento.

Como referido acima, os problemas integrados de planeamento e escalonamento incluem dois subproblemas de otimização padrão (o problema escalonamento em máquinas paralelas e o problema de corte de empacotamento uni-dimensional). Em Kis e Kovács (2012), os autores exploraram também uma formulação monolítica. Apesar dos seus resultados serem promissores, as suas abordagens encontram algumas dificuldades na resolução do problema integrado, em particular nas instâncias maiores e quando todas as tarefas são longas no que diz respeito à duração dos períodos de tempo.

2.3.3 Problemas não integrados relacionados

O problema de escalonamento em máquinas paralelas e o problema de corte e empacotamento a uma dimensão são dois problemas \mathcal{NP} -difíceis bem conhecidos. Os pro-

blemas de empacotamento a uma dimensão foram discutidos em detalhe mais acima. Nesta secção, é realizada uma breve revisão relativa aos problemas de escalonamento.

Os problemas de escalonamento representam um campo importante e abrangente no domínio da investigação operacional e das ciências da computação, que recebeu muita atenção ao longo dos últimos anos. Por exemplo, Carlier e Néron (2000) lidaram com o problema de escalonamento de projetos com recursos limitados, que consiste no escalonamento de forma otimizada de um conjunto de actividades que requerem quantidades variáveis de um determinado recurso. As actividades têm datas de lançamento e tempos de processamento, o objectivo é minimizar o *makespan*. Embora cada actividade necessite de uma quantidade constante do recurso, esta pode ser diferente entre as actividades. Além disso, pode haver precedências entre as actividades, e uma actividade não pode terminar após a sua data de entrega. No problema de escalonamento estudado nesta tese, as tarefas também possuem datas de lançamento e datas de entrega, mas não há precedências entre as mesmas. Além disso, as tarefas podem ser executadas antes ou depois das suas datas de entrega, com um custo de penalidade. O objectivo consiste em minimizar o custo total das penalidades.

Mais recentemente, Su et al. (2013) resolveram um problema de escalonamento em máquinas com diferentes velocidades e disponibilidade limitada, e onde o objectivo consiste em minimizar tanto o *makespan* como o atraso máximo. Os autores modelam o problema como um problema de fluxos de custo mínimo num grafo. Detienne (2014) resolveu diversas variantes do problema de escalonamento em máquinas que podem não estar disponíveis em determinados períodos de tempo. O autor considerou diferentes casos com tarefas retomáveis, não retomáveis e semi-retomáveis. Enquanto as tarefas semi-retomáveis precisam de um determinado tempo de *setup* após terem sido interrompidas, as tarefas não retomáveis devem ser reiniciadas. Sadykov e Wolsey (2006) exploraram um problema de escalonamento em que ambos os tempos de processamento e os custos das tarefas dependem do equipamento que é usado. Além disso, todas as tarefas devem ser concluídas entre a data de lançamento e data de entrega. Os autores propuseram várias abordagens de programação inteira, programação por restrições e geração de colunas.

Yang et al. (2012) usaram um algoritmo genético melhorado para resolver um

problema de escalonamento de *job shop* com datas de lançamento, datas de entrega, prazos finais e custos associados à precocidade e atraso das tarefas. Os autores consideraram o caso em que as tarefas podem ser concluídas após a sua data de entrega com uma dada penalidade e em que os prazos finais devem ser estritamente satisfeitos.

Kis e Kovács (2012) abordaram recentemente um problema de planeamento de produção para o qual propõem um algoritmo de partição e avaliação sucessivas com cortes. O modelo de Bendraouche e Boudhar (2012) utiliza grafos onde os vértices representam tarefas a serem executadas em duas máquinas.

Em Haned et al. (2012) é proposto um modelo de programação dinâmica para duas máquinas idênticas testando-se diferentes cenários com o objectivo de minimizar o desperdício na produção e diminuir o tempo de execução. Lenstra et al. (1990) desenvolveram um algoritmo para resolver o problema com um número arbitrário de máquinas. No problema de escalonamento da produção com máquinas paralelas em modo *online*, existe ainda pouco trabalho desenvolvido embora Shabtay e Karhi (2012) tenham recentemente proposto um algoritmo para o resolver. No mesmo artigo é proposta a resolução do mesmo problema em modo *offline*.

2.4 Problemas de encaminhamento de veículos e gestão de inventários

2.4.1 Problemas de encaminhamento de veículos

O problema de encaminhamento de veículos (*Vehicle Routing Problem*, VRP) é um problema complexo. Mesmo na sua forma mais simples, é o problema de transportes mais estudado na literatura, com um elevado número de aplicações práticas em problemas reais. Vários autores (Toth e Vigo, 2001, 2014, Caceres-Cruz et al., 2014, Coelho et al., 2014) dedicaram-se a este problema, apresentando revisões da literatura do mesmo onde analisam diferentes variantes.

Este problema caracteriza-se por um conjunto de bens a serem entregues a clientes, a partir de um ou mais armazéns, através de uma frota de veículos. A solução para o problema de encaminhamento de veículos define as rotas dos veículos, assim como as

quantidades de mercadoria a transportar em cada uma delas, de forma a que todas as necessidades dos clientes sejam satisfeitas e que o custo global da operação seja minimizado.

As rotas dos veículos são definidas através de caminhos possíveis entre os armazéns e todos os clientes. Na Figura 2.6 é possível observar um exemplo onde existe um armazém e vários clientes que deverão ser visitados. O conjunto de arcos traçado define os possíveis caminhos que os veículos podem percorrer (a). Em (b) tem-se uma representação da solução óptima para o problema, estando definido o conjunto de rotas a percorrer por cada veículo de forma a satisfazer todos os clientes e minimizando o custo global de transporte.

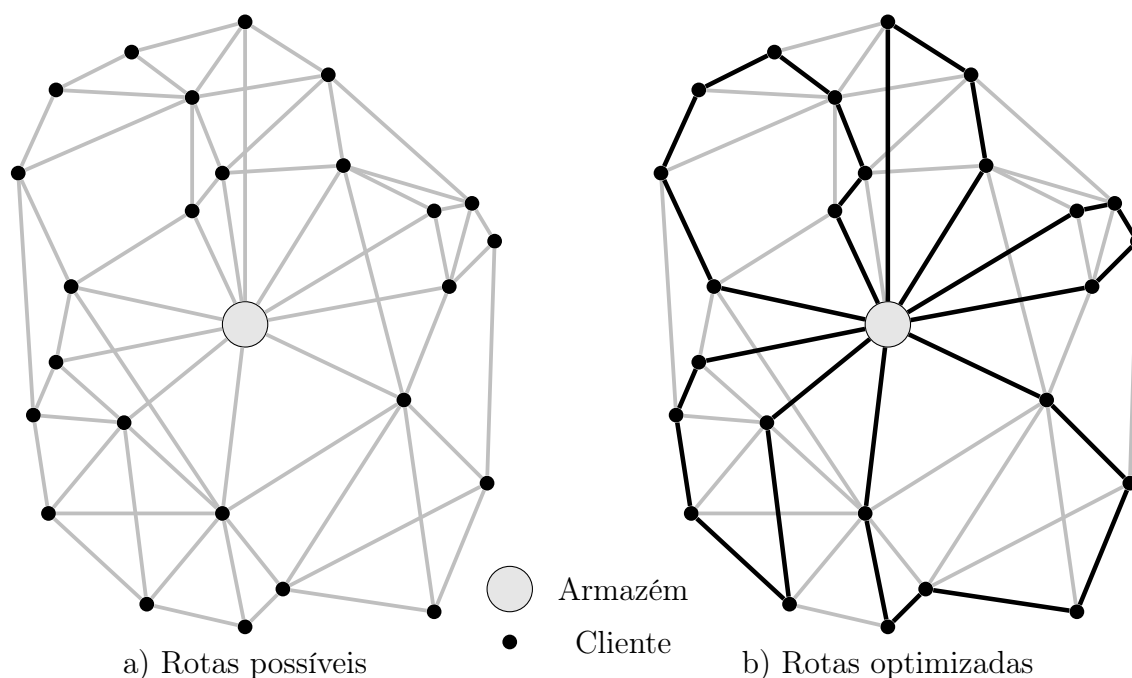


Figura 2.6: Exemplo de um problema de VRP com as rotas possíveis e rotas otimizadas.

Existem diversas variantes para o problema de encaminhamento de veículos, que muitas vezes dependem da aplicação prática em casos reais. É possível, por exemplo, definir que um cliente poderá receber a sua mercadoria de forma faseada ou de uma só vez. Em alguns problemas, a entrega da mercadoria fora do período definido pelo cliente (janela temporal) não é permitida, enquanto que noutros esta pode ser permitida, incorrendo-se em penalidades. No que diz respeito aos veículos, estes podem ser

homogéneos ou heterogéneos. Esta diferença poderá verificar-se em termos de capacidade de carga, de tipo de veículo, dimensão, existência de compartimentos especializados para diferentes mercadorias ou custo operativo. Neste problema, pode também considerar-se que um veículo possa partir de um determinado armazém, realizar as entregas aos clientes e terminar num outro armazém. Para a utilização dos veículos são necessários condutores, que obedecem a restrições impostas para sua segurança, como sendo as horas de viagem, assim como tempo de descanso. O escalonamento de condutores para este problema pode ser realizado de forma integrada.

Nesta classe de problemas podem considerar-se diferentes funções objectivo. Tipicamente, esta define-se pela minimização do custo global de transporte da mercadoria através das rotas seleccionadas, que satisfazem a procura de todos os clientes. O custo pode ser determinado pela distância percorrida, assim como pelo uso do veículo a percorrer a rota e o custo do condutor. Outra possível função objectivo diz respeito ao balanceamento das rotas, ou seja a minimização das diferenças das distâncias percorridas por todos os veículos para que, por exemplo, todos os condutores tenham um idêntico período de trabalho. Pode ainda considerar-se, entre outros, a minimização do número de veículos necessários, ou ainda, pondo-se o foco no cliente, a minimização das penalidades associadas aos atrasos nas entregas.

Múltiplas abordagens, tanto heurísticas como exactas, para a resolução do problema de encaminhamento de veículos foram propostos na literatura. Laporte e Nohrbert (1987) descrevem métodos para fornecer soluções iniciais. Toth e Vigo (2001) propõem modelos para algumas variantes do problema. Caceres-Cruz et al. (2014) realizam um estudo sobre o problema abordando diferentes variantes e suas resoluções, focando-se em casos práticos. Em Toth e Vigo (2014), são abordados de forma exhaustiva os problemas de encaminhamento de veículos e suas variantes, assim como os seus métodos de resolução mais recentes, tanto exactos como heurísticos. No estudo de Baldacci et al. (2012), os autores abordam o problema de encaminhamento de veículos com restrições de capacidade dos veículos e com janelas temporais e propõem algoritmos exactos ao nível do estado da arte para a sua resolução.

Um trabalho de Archetti e Speranza (2014) faz uma revisão de abordagens do tipo *matheurísticas* (abordagens híbridas que combinam os pontos fortes das meta-

heurísticas e dos métodos exatos) para problemas de encaminhamento de veículos. Os autores propõem uma classificação em três classes dessas matheurísticas: *abordagens de decomposição*, *heurísticas de melhoria*, e *abordagens baseadas em geração de colunas e algoritmos de partição e avaliação sucessivas*. Mais recentemente, em Labadie et al. (2016) é feita uma revisão da literatura sobre meta-heurísticas aplicadas aos problemas de encaminhamento de veículos.

2.4.2 Encaminhamento de veículos e gestão integrada de inventários

A otimização integrada dos processos de uma cadeia de abastecimento, desde a extração das matérias-primas até à entrega do produto ao cliente final, representa mais um factor diferenciador nos mercados cada vez mais competitivos. Nesta óptica, as empresas têm vindo a recorrer a sistemas avançados de apoio à decisão, que as ajudam a lidar com situações cada vez mais complexas.

Um exemplo de problema integrado que surge na indústria é o problema de encaminhamento de veículos com gestão integrada de inventários (*Inventory Routing Problem*, IRP).

Tradicionalmente, os problemas de encaminhamento de veículos e a gestão de inventários (*Vendor Managed Inventory*, VMI) são tratados nas empresas de forma separada. No VMI, o fornecedor é responsável pelas decisões de gestão de inventário dos produtos nos seus clientes (Bertazzi et al., 2005, Archetti et al., 2007, Andersson et al., 2010, Schmid et al., 2013).

O objectivo, em que se considera simultaneamente o problema de encaminhamento de veículos e gestão de inventários, é de otimizar a gestão da frota de veículos, baixando paralelamente os níveis de inventário na cadeia de abastecimento, o que se traduz numa redução do preço final dos produtos.

O encaminhamento de veículos e gestão integrada de inventários tem como objectivo minimizar o custo total das operações de transporte de mercadorias do fornecedor para o cliente, de forma a que este mantenha um inventário que lhe permita satisfazer as suas procuras a cada período de um dado horizonte de planeamento. Esta aborda-

gem pode ser utilizada em qualquer cadeia de abastecimento que envolva o transporte de mercadoria, seja este marítimo, terrestre, aéreo ou ferroviário, entre outros. Andersson et al. (2010) propõem uma classificação e revisão da literatura desta classe de problemas.

A Figura 2.7 representa uma solução para um exemplo genérico de um problema de encaminhamento de veículos e gestão integrada de inventários. Na Figura 2.7 pode verificar-se a diferença gerada pelo problema de encaminhamento de veículos e gestão integrada de inventários. Neste caso, os clientes representados possuem uma procura para períodos diferentes de determinado produto que deverá ser satisfeita, representada nas figuras como $[d_x^y, d_x^{y+1}]$, onde d_x^y é a procura do cliente x no período y . Na Figura 2.7a estão representadas as rotas para satisfazer as procuras dos clientes no primeiro período, que em alguns casos, podem também satisfazer a procura para o segundo período. As entregas das procuras por cliente estão representadas na Figura 2.7 a negrito. Da mesma forma, a Figura 2.7b apresenta o segundo período com as rotas para satisfazer as entregas nesse período. Neste caso, procuras não satisfeitas do período anterior não poderão ser entregues posteriormente. Na Figura 2.7 é possível verificar que, por exemplo, o cliente 3 é visitado uma só vez na rota realizada no período 1, tendo a sua procura sido entregue para os dois períodos. Na mesma rota, são visitados os clientes 2, 3, 4 e 5. No período seguinte, à excepção do cliente 3, todos os outros são novamente visitados por outras rotas.

Mirzapour Al-e hashem e Rekik (2013) introduzem o conceito de “logística verde” em problemas de IRP, considerando níveis de emissão de gases de efeito de estufa para além dos custos de transporte na selecção dos veículos.

Mjirda et al. (2014) resolvem o problema de IRP em duas fases. Inicialmente, resolvem o problema de encaminhamento de veículos para cada um dos períodos para obter uma solução inicial do problema, sem ter em consideração o nível de inventário. Na segunda fase, melhoram a solução inicial minimizando o custo de transporte e de inventário. Nas duas fases são utilizadas heurísticas e modelos de programação linear para determinar o número de produtos a recolher a cada fornecedor em cada período. Uma formulação de programação inteira mista é proposta por Archetti et al. (2007), que para reforçar o modelo utilizam desigualdades válidas.

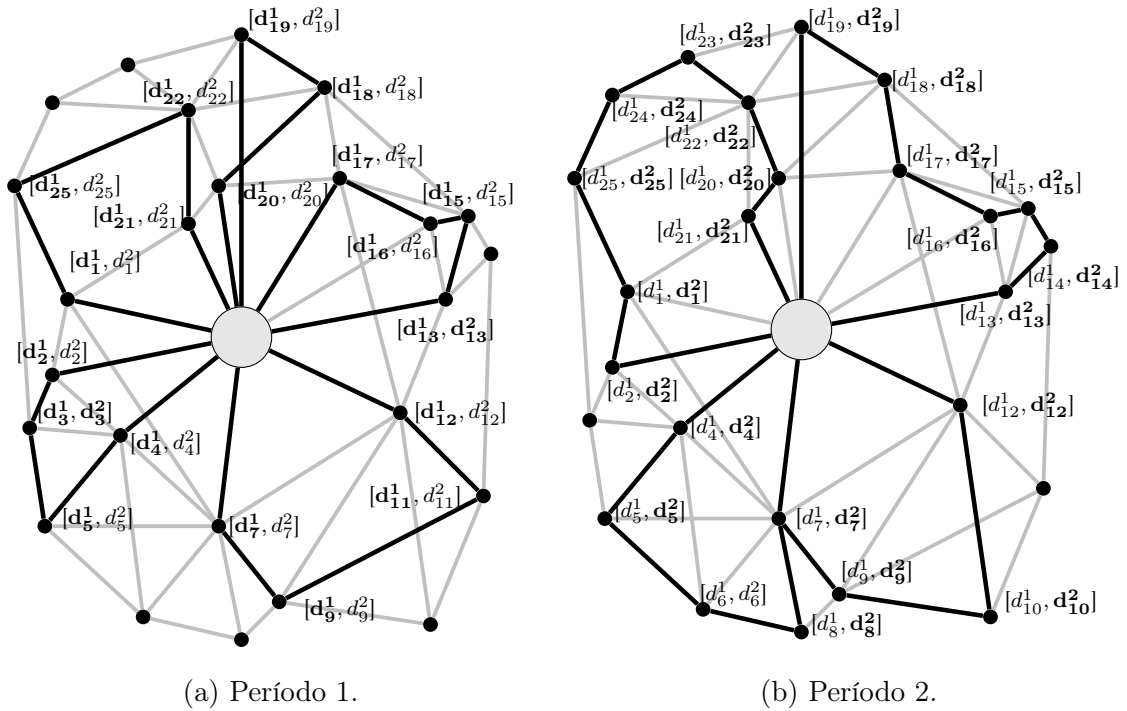


Figura 2.7: Solução IRP para dois períodos.

Um modelo de programação inteira mista não linear é proposto por Aghezzaf et al. (2006), que utilizam abordagens aproximadas baseadas em métodos de geração de colunas para resolver o problema.

Bertazzi et al. (2005) propõem abordagens heurísticas baseadas em duas decomposições diferentes de um problema IRP. Abdul Rahim et al. (2014) resolvem o problema com procura estocástica com um único armazém e um único produto, e utilizam uma formulação inteira mista, que é decomposta em dois sub-problemas, sendo um o problema de inventário e o segundo de transporte. Para uma aproximação da solução óptima foi utilizada uma relaxação Lagrangeana. No estudo de Archetti e Speranza (2016) foi realizada uma comparação dos dois tipos de variantes do problema integrado de encaminhamento de veículos com gestão de inventários. No seu estudo, Archetti e Speranza (2016) fazem uma comparação entre o modelo em que o inventário é gerido pelo fornecedor e o modelo em que o inventário é gerido pelo retalhista, concluindo que a primeira variante traz vantagens em relação à variante em que o inventário é gerido pelo retalhista. Em Bertazzi e Speranza (2012), são abordadas diversas variantes do problema, descrevendo algoritmos a nível de estado da arte para a sua resolução. Um

estudo mais recente dos mesmos autores Bertazzi e Speranza (2012), apresenta alguns casos especiais relativos ao problema assim como as suas abordagens de resolução.

Testes comparativos com modelos a nível de estado da arte foram realizados em (Archetti et al., 2014). No seu artigo, Archetti et al. (2014) apresentam modelos da literatura, assim como um novo modelo, testando-os em instâncias da literatura. Estes modelos são reforçados utilizando várias desigualdades válidas e algoritmos de partição e avaliação sucessivas com cortes.

2.5 Conclusões

Problemas que eram anteriormente resolvidos separadamente começam agora a ser resolvidos de forma integrada. A resolução destes problemas no contexto da indústria traz vantagens competitivas uma vez que a solução do problema integrado conduz tipicamente a uma solução de melhor qualidade do ponto de vista global. No entanto, estes problemas são mais difíceis de resolver, uma vez que incorporam outros problemas que são por si só de elevada complexidade.

Dadas as vantagens que existem na resolução destes problemas integrados, em particular nas cadeias de abastecimento, têm surgido na literatura diversas abordagens para os mesmos. O objectivo é desenvolver modelos e métodos eficientes. Nesse âmbito, destacam-se os modelos e métodos de programação inteira, incluindo as heurísticas que podem permitir obter soluções de boa qualidade de forma rápida. Neste capítulo, apresentámos as abordagens propostas na literatura para os problemas que são explorados nesta tese. Nos próximos capítulos, será retomada a análise de cada um desses problemas, apresentando e analisando abordagens novas baseadas essencialmente em métodos de programação inteira.

Capítulo 3

O problema combinado de corte e escalonamento ¹

Conteúdo

3.1	Introdução	36
3.2	Modelos de programação inteira da literatura	40
3.2.1	O modelo de Reinertsen e Vossen	40
3.2.2	O modelo de Arbib e Marinelli	41
3.3	Modelo de afectação	43
3.3.1	Reforço do modelo de afectação	44

¹Os resultados deste capítulo foram publicados nos artigos:

- N. Braga, C. Alves, R. Macedo, J. Valério de Carvalho, A model-based heuristic for the combined cutting stock and scheduling problem, *Lecture Notes in Computer Science*, vol. 9156, pp. 490-505, 2015 (Braga et al. (2016b));
- N. Braga, C. Alves, J. Valério de Carvalho, Exact solution of combined cutting stock and scheduling problems, *Lecture Notes in Economics and Mathematical Systems*, vol. 682, pp. 131-139, 2016 (Braga et al. (2015));
- N. Braga, C. Alves, R. Macedo, e J. M. Valerio de Carvalho. Combined cutting stock and scheduling: a matheuristic approach. *International Journal of Innovative Computing and Applications*, 7(3):135–146, 2016.

3.4	Um método de planos de corte baseado no modelo de afectação	45
3.4.1	Funções duais válidas e desigualdades válidas	45
3.4.2	Um método de planos de corte	46
3.5	Resultados computacionais	47
3.6	Um novo modelo exacto de fluxo em rede pseudo-polinomial	51
3.7	Uma abordagem heurística baseada no modelo de fluxos em rede	53
3.7.1	Visão geral	53
3.7.2	Modelo de fluxos em rede revisto	54
3.7.3	Modelo de afectação no tempo	56
3.8	Resultados computacionais	58
3.9	Conclusões	65

3.1 Introdução

O problema de corte é um problema de optimização combinatoria bem conhecido na literatura, que pertence à classe de problemas de corte e empacotamento. Tem aplicações em diversos contextos, tais como a área das ciências da computação, telecomunicações, indústrias têxteis, metalúrgica, do couro, papel, madeira ou vidro, entre outros (Suliman, 2001). O problema padrão consiste em encontrar uma forma de cortar um dado conjunto de objectos mais pequenos (itens) a partir de um conjunto de objectos maiores (rolos), de modo a que o número utilizado destes últimos seja minimizado. Embora o problema seja \mathcal{NP} -difícil, foram apresentadas muitas contribuições na literatura, incluindo métodos exactos, procedimentos para encontrar limites inferiores e abordagens heurísticas (alguns exemplos podem ser encontrados em Gilmore e Gomory (1961, 1963), Dyckhoff (1981), Scholl et al. (1997), Fekete e Schepers (2001), Degraeve e Peeters (2003), Perrot (2005)). Graças aos progressos na resolução de instâncias de grande escala do problema padrão, muitas variantes deste problema têm vindo a ser exploradas, incluindo várias restrições encontradas em contextos reais industriais.

Devido à sua utilidade e natureza, os problemas de corte e empacotamento têm sido um grande tema de investigação desde há muitos anos. A tipologia de Wäscher et al.

(2007) é uma boa ilustração desse facto. Para além de proporcionar um amplo sistema de classificação para estes problemas, identifica também um grande conjunto de contribuições para diferentes variantes do problema padrão. Estas variantes incluem diferentes características dos itens e rolos e diferentes tipos de funções objectivo. Em geral, o objectivo deste problema é de minimizar o material total usado para satisfazer a procura dos clientes, isto é, minimizar o desperdício (Reinertsen e Vossen, 2010). Este problema pode também ser visto como um problema de maximização, uma vez que diminuindo o desperdício, haverá um aumento do lucro (Reinertsen e Vossen, 2010). Gilmore e Gomory (1961) definiram um modelo de programação linear, que se baseia na enumeração de padrões de corte potenciais, em que o problema foi resolvido com geração de colunas.

No problema de corte padrão, a ordem em que os cortes são efectuados não é importante. No entanto, se se assumir que as datas de entrega devem ser consideradas, é necessário adicionar essa informação ao problema. Um novo tipo de problema é então criado: um problema de corte com prazos de entrega. Reinertsen e Vossen (2010) propuseram um novo modelo de optimização para o problema de corte com prazos de entrega, considerando uma variante do problema que tem em conta na função objectivo o escalonamento relacionado com a existência de prazos de entrega dos itens. Este problema foi também abordado recentemente por Arbib e Marinelli (2014).

No problema de corte combinado com prazos de entrega abordado nesta tese, considera-se um conjunto I de itens unidimensionais ($|I| = n$), com tamanhos designados por w_i , $i \in I$ e uma procura correspondente b_i , $i \in I$. Estes itens têm de ser cortados a partir de rolos de dimensão padrão W , cuja disponibilidade é assumida como sendo ilimitada.

Uma vez que as abordagens concebidas neste trabalho são independentes do nível de procura, será daqui em diante referido apenas o caso mais geral de corte e empacotamento. A parte de afectação do problema é baseada no facto de que os prazos de entrega são impostos para cada tipo de item. Consequentemente, para cada item $i \in I$, há uma data de entrega designada por d_i . Do ponto de vista da afectação, os itens podem ser vistos como tarefas, cuja entrega está prevista, no máximo, até à d_i -

ésima unidade de tempo. Entregas após as datas limite são toleradas, mas com uma penalidade na função objectivo. Além disso, assume-se que cortar um rolo demora exactamente uma unidade de tempo. Assim, o tempo necessário para completar a tarefa relacionada com um item i corresponde ao número de rolos que tenham sido cortados até o último rolo no qual o item i é cortado. A função objectivo do problema considera dois termos: um relacionado com o desperdício total e outro relacionado com o atraso total. Aqui, será seguida a abordagem de Arbib e Marinelli (2014), considerando-se que cada termo tem o mesmo peso na função objectivo.

Um problema semelhante foi explorado por Li (1996) para o caso com duas dimensões, com diferentes comprimentos de rolos. O autor considera que existem datas de lançamento e de entrega, e propõe heurísticas baseadas em modelos de programação linear para encontrar soluções válidas para o problema. Como apontado por Arbib e Marinelli (2014), os modelos descritos em Li (1996) não garantem que a solução encontrada seja a solução óptima global para o problema. Em Reinertsen e Vossen (2010), os autores abordam o problema unidimensional com datas de entrega e sem datas de lançamento. Os autores (Reinertsen e Vossen, 2010) descrevem um modelo de Programação Inteira (PI), que é uma extensão do modelo de geração de colunas para o problema de corte. Os autores (Reinertsen e Vossen, 2010) utilizam um modelo de afectação para atribuir padrões de corte a períodos de tempo. As variáveis estão associadas a padrões indexados pelo período de tempo. O horizonte de planeamento é dividido em n períodos de tempo distintos. Em seguida, a tarefa correspondente ao item i só pode ser executada até ao i -ésimo período de tempo. Em Arbib e Marinelli (2014), os autores demonstram uma vez mais que essa abordagem pode levar a soluções sub-óptimas, sempre que uma solução baseada no conceito de menor data de lançamento primeiro (*early due date first*) não for a solução óptima.

A primeira formulação exacta para o problema combinado de corte e escalonamento estudado neste capítulo (CSSP para *Combined Cutting Stock and Scheduling Problem*, na terminologia anglo-saxónica) foi proposta em Arbib e Marinelli (2014). Como em Reinertsen e Vossen (2010), os autores consideram uma formulação em que as variáveis estão relacionadas com padrões de corte, indexados pelo período de tempo. Adicionalmente, usam variáveis que representam o nível do inventário de um

item num determinado período de tempo, combinando-as com equações de equilíbrio para identificar situações de atraso. O horizonte de planeamento é dividido num número fixo de períodos de tempo, e um item i só pode ser afectado até ao i -ésimo período. Os autores resolvem o problema através de procedimentos de geração de colunas, onde o subproblema é um problema do caminho mais curto num grafo modificado. Para lidar com o tamanho do modelo, os autores descrevem um procedimento de divisão de períodos de tempo que permite começar com períodos de tempo maiores, agrupando vários períodos no mesmo conjunto de padrões da mesma dimensão que o tempo agrupado. Esta estratégia consiste em resolver de forma iterativa o modelo, com um menor número de períodos de tempo. Os resultados computacionais fornecidos pelos autores ilustram a dificuldade inerente do problema.

Em Braga et al. (2016b), é explorada uma formulação de afectação exata e compacta para uma versão do *CSSP* semelhante à discutida em Reinertsen e Vossen (2010) e Arbib e Marinelli (2014). O modelo é reforçado aplicando-se diferentes desigualdades válidas. Algumas destas desigualdades são geradas através da utilização de funções duais válidas, que são aplicadas às restrições de mochila do modelo original. É descrito um procedimento de planos de corte dependente dessas funções duais válidas, sendo o seu desempenho analisado através de testes computacionais em instâncias de referência para diferentes conjuntos de parâmetros.

Em Braga et al. (2015), é descrito um novo modelo pseudo-polinomial para o problema abordado neste capítulo. O modelo descrito por Braga et al. (2015) e baseado no modelo de fluxos em rede é revisto com o objectivo de reduzir o número de períodos associados ao problema e assim reduzir a complexidade do mesmo. No modelo revisto, ao invés de serem utilizados explicitamente todos os períodos de tempo do horizonte de planeamento, os mesmos são agregados. Como o modelo revisto fornece uma solução cujo valor é sobrestimado, é utilizado um modelo auxiliar de afectação para construir uma solução válida cujo valor seja inferior.

Este capítulo está organizado da seguinte forma. Na Secção 3.2, são apresentados os modelos de programação inteira ao nível do estado-da-arte para o problema analisado neste capítulo. Posteriormente, na Secção 3.3 é apresentado um novo modelo de afectação para o problema, assim como estratégias de reforço com o objectivo de

acelerar a convergência para uma solução. Ao modelo da Secção 3.3 foram adicionados planos de corte através de funções duais válidas. Essas funções são apresentadas na Secção 3.4. Os resultados dos respectivos testes computacionais são discutidos na Secção 3.5. Um novo modelo pseudo-polinomial de fluxos em rede é descrito na Secção 3.6. Na secção 3.7, é proposta uma alteração ao modelo a partir da qual é definida uma abordagem heurística. Os testes computacionais associados a essa abordagem são descritos na Secção 3.8. O capítulo termina com uma breve síntese e conclusões.

3.2 Modelos de programação inteira da literatura

3.2.1 O modelo de Reinertsen e Vossen

O modelo de programação inteira proposto por Reinertsen e Vossen (2010) é uma extensão do modelo de geração de colunas de Gilmore e Gomory (1961) para o problema padrão de corte de empacotamento. O modelo atribui padrões de corte a períodos de tempo. O horizonte de planeamento é dividido em exatamente n períodos de tempo, sendo n o número de diferentes tipos de itens (tarefas) disponíveis. O modelo impõe a restrição adicional de que um item i só pode ser cortado até ao i -ésimo período de tempo. Note-se que, sem perda de generalidade, assume-se que os itens são ordenados por ordem crescente dos seus valores de data de entrega. Como demonstrado em Arbib e Marinelli (2014), devido a esta exigência específica, o modelo resultante não é exato em geral, mas apenas nos casos em que a solução para o problema não tem atrasos nas entregas. O modelo de Reinertsen e Vossen (2010) é representado da seguinte forma:

$$\min. \quad \sum_{p \in P} \sum_{k=1}^n x_{pk} + \sum_{i=1}^n M_i y_i \quad (3.1)$$

$$\text{s.a.} \quad \sum_{p \in P} \sum_{k=1}^i x_{pk} \leq d_i + y_i, \quad i = 1, \dots, n, \quad (3.2)$$

$$\sum_{p \in P} \sum_{k=1}^i a_{ip} x_{pk} \geq b_i, \quad i = 1, \dots, n, \quad (3.3)$$

$$x_{pk} \geq 0 \text{ e inteiro, } \quad p \in P, \quad k = 1, \dots, n, \quad (3.4)$$

$$y_i \geq 0 \text{ e inteiro, } i = 1, \dots, n. \quad (3.5)$$

A partir de cada padrão $p \in P$ são cortados a_{ip} itens de tipo i . As variáveis x_{pk} representam o número de rolos cortados de acordo com o padrão p (do conjunto de possíveis padrões de corte P) no período de tempo k , com $k = 1, \dots, n$. O atraso associado a um item i é representado pelas variáveis y_i . Essas variáveis tomam o valor 0 se o item correspondente é cortado dentro do seu limite de entrega d_i . Se um item i é cortado após a sua data de entrega, a variável y_i terá um valor que é igual ao número de rolos cortados após d_i até ao último rolo a partir do qual um item i é cortado. As restrições (3.2) definem as variáveis y_i . As restrições (3.3) correspondem às restrições de procura. A função objectivo (3.2) considera o uso de material assim como a afectação, penalizando o atraso total. Em (3.1), o peso relacionado com o atraso de um determinado item i é representado por M_i .

Como referido em Reinertsen e Vossen (2010), se os prazos de entrega são idênticos para todos os itens e estes prazos de entrega não são restritivos, (*i.e.* os seus valores são grandes), o modelo (3.1)–(3.5) é equivalente ao modelo de Gilmore e Gomory (1961) para o problema de corte e empacotamento.

3.2.2 O modelo de Arbib e Marinelli

Arbib e Marinelli (2014) descrevem um modelo de programação inteira alternativa para o CSSP. Como em Reinertsen e Vossen (2010), esse modelo atribui padrões de corte a períodos de tempo com uma determinada duração. Os autores dividiram o horizonte de planeamento D em $n + 1$ períodos de tempo, sendo que o último instante de tempo coincide com a data de entrega de um item (assume-se novamente que os itens estão ordenados por ordem crescente de datas de entrega): $[d_0, d_1]$, $[d_1, d_2]$, \dots , $[d_n, d_{n+1}]$, com $d_0 = 0$ e $d_{n+1} = D$. O modelo é definido da seguinte forma:

$$\min. \quad \sum_{p \in P} \sum_{k=1}^{n+1} x_p^k + \sum_{i=1}^n \sum_{k>i} (d_k - d_{k-1}) y_i^k \quad (3.6)$$

$$\text{s.a.} \quad \sum_{p \in P} x_p^k \leq d_k - d_{k-1}, \quad k = 1, \dots, n + 1, \quad (3.7)$$

$$s_i^{k-1} + \sum_{p \in P} a_{ip} x_p^k = s_i^k + b_i^k, \quad i = 1, \dots, n, \quad k = 1, \dots, n+1, \quad (3.8)$$

$$u_i^k y_i^k + s_i^{k-1} \geq 0, \quad i = 1, \dots, n, \quad k = i+1, \dots, n+1, \quad (3.9)$$

$$x_p^k \geq 0 \text{ e inteiro}, \quad p \in P, \quad k = 1, \dots, n+1, \quad (3.10)$$

$$y_i^k \in \{0, 1\}, \quad i = 1, \dots, n, \quad k = i+1, \dots, n+1 \quad (3.11)$$

$$s_i^k \text{ inteiro}, \quad i = 1, \dots, n, \quad k = i, \dots, n+1. \quad (3.12)$$

Como em (3.1)–(3.5), o número de padrões de corte válidos é representado por P . As variáveis x_p^k representam o número de rolos cortados de acordo com o padrão $p \in P$ no período de tempo k . As variáveis s_i^k determinam o nível de inventário de um item i no período de tempo k , enquanto as variáveis y_i^k estão definidas apenas para os períodos de tempo posteriores ao prazo de entrega k .

A função objectivo (3.6) é composta por um termo relacionado com o total de material utilizado e por um termo relacionado com a soma dos atrasos de entrega dos itens. Note-se que este último sobrestima o atraso de um item i , uma vez que considera que os padrões de corte que são cortados no período de tempo k estão escalonados para o último instante de tempo deste período. O número total de rolos necessários para cortar os padrões escalonados para o período de tempo k não pode exceder a duração deste período de tempo, uma vez que é necessária uma unidade de tempo para cortar um rolo, seja qual for o padrão de corte considerado. Essa restrição é definida por (3.7). Do ponto de vista geral, o parâmetro b_i^k denota a procura de um item i que deverá ser entregue no período de tempo k . Neste caso $b_i^k = b_i$ se $k = i$, e 0 no caso contrário. A restrição (3.8) força as quantidades de itens a serem satisfeitas, pois possibilita que um item i seja cortado mais tarde do que a sua data de entrega, definindo a variável s_i^k correspondente para um valor negativo. Neste caso, as variáveis correspondentes y_i^k são definidas como 1 através das restrições (3.9), em que u_i^k é um limite superior do número de itens i que pode estar em atraso no período de tempo k .

Dado que o número de padrões de corte válidos e a sua correspondente afectação a períodos de tempo é exponencial, este modelo só pode ser resolvido através de procedimentos dinâmicos de geração de colunas para instâncias de média ou grande

escala de CSSP. Além disso, este modelo requer um processo de refinamento adicional, tal como descrito em Arbib e Marinelli (2014) que iterativamente actualiza a duração dos períodos de tempo para atingir a solução óptima do problema.

3.3 Modelo de afectação

Seja T a duração do horizonte de planeamento. O CSSP pode ser formulado com variáveis x_i^t que representam o número de itens de tamanho $i = 1, \dots, n$ que são cortados no período de tempo $t = 1, \dots, T$. A duração do período é assumida como sendo igual a 1, existindo portanto uma relação de um para um entre os períodos de tempo e os rolos que são cortados. As variáveis binárias z^t , $t = 1, \dots, T$, determinam se um rolo é usado ou não no período de tempo t . O atraso de um item $i = 1, \dots, n$ é medido através das variáveis binárias y_i^t , $t = d_i + 1, \dots, T$, que são apenas definidas desde a sua data de entrega mais um período de tempo até ao final do horizonte de planeamento. As variáveis y_i^t tomam o valor 1 se o item i é cortado no período de tempo t . Portanto, se $y_i^t = 1$ para algum $i = 1, \dots, n$, e $t = d_i + 1, \dots, T$, o item ficará atrasado pelo menos $t - d_i$ unidades de tempo. Estas variáveis podem ser relacionados com as variáveis y_i^k usadas por Arbib e Marinelli (2014). O modelo de afectação é definido da forma seguinte:

$$\min. \quad \sum_{t=1}^T z^t + \sum_{i=1}^n \sum_{t=d_i+1}^T y_i^t \quad (3.13)$$

$$\text{s.a.} \quad \sum_{t=1}^T x_i^t = b_i, \quad i = 1, \dots, n, \quad (3.14)$$

$$\sum_{i=1}^n w_i x_i^t \leq W z^t, \quad i = 1, \dots, n, \quad t = 1, \dots, T, \quad (3.15)$$

$$x_i^t + y_i^{t+1} \leq L_i^{\min} y_i^t, \quad i = 1, \dots, n, \quad t = d_i + 1, \dots, T, \quad (3.16)$$

$$z^t \in \{0, 1\}, \quad t = 1, \dots, T, \quad (3.17)$$

$$x_i^t \geq 0 \text{ e inteiro}, \quad i = 1, \dots, n, \quad t = 1, \dots, T, \quad (3.18)$$

$$y_i^t \in \{0, 1\}, \quad i = 1, \dots, n, \quad t = d_i + 1, \dots, T. \quad (3.19)$$

Como mencionado acima, a função objectivo (3.13) é composta por um termo relativo

ao desperdício total e outro ao atraso total. As restrições (3.14) garantem a satisfação da procura de cada item i . As restrições (3.15) não permitem que o tamanho do padrão exceda o comprimento do rolo, enquanto as restrições (3.16) definem as variáveis y_i^t . Os valores L_i^{min} para $i = 1, \dots, n$, podem ser definidos da forma seguinte:

$$L_i^{min} = \min \left\{ b_i, \left\lceil \frac{W}{w_i} \right\rceil + 1 \right\}. \quad (3.20)$$

3.3.1 Reforço do modelo de afectação

A relaxação linear de (3.13)–(3.19) pode ser reforçada através de desigualdades válidas (3.21)–(3.24). Uma vez que uma solução válida nunca usará menos rolos do que os necessários para satisfazer a procura de todos os itens, poderá considerar-se a inequação:

$$\sum_{t=1}^T z^t \geq K^{min}, \quad (3.21)$$

onde K^{min} é o limite inferior do número de rolos calculados por exemplo pelas funções duais válidas (Clautiaux et al., 2010). Afectar um item a um rolo ou padrão (ou período de tempo) implica a utilização de um rolo nesse período de tempo e desta forma tem-se:

$$z^t \geq y_i^t, \quad i = 1, \dots, n, \quad t = d_i + 1, \dots, T. \quad (3.22)$$

O uso de um rolo no período $t + 1$, $t = 1, \dots, T - 1$ implica ter-se utilizado um rolo no período t , e assim:

$$z^t \geq z^{t+1}, \quad t = 1, \dots, T. \quad (3.23)$$

Restrições similares podem aplicar-se às variáveis y_i^t , o que se traduz nas seguintes desigualdades:

$$y_i^t \geq y_i^{t+1}, \quad t = d_i + 1, \dots, T. \quad (3.24)$$

3.4 Um método de planos de corte baseado no modelo de afectação

3.4.1 Funções duais válidas e desigualdades válidas

Uma função $f : [0, 1] \rightarrow [0, 1]$ é uma função dual válida (FDV) se para qualquer conjunto finito $\{x_i \in \mathbb{R}_+ : i \in J\}$ de números reais não negativos o seguinte é válido:

$$\sum_{i \in J} x_i \leq 1 \implies \sum_{i \in J} f(x_i) \leq 1. \quad (3.25)$$

As funções duais válidas têm vindo a ser utilizadas essencialmente para derivar limites inferiores para problemas clássicos de corte e empacotamento. Em Clautiaux et al. (2010), os autores demonstram que estas funções podem também ser utilizadas para derivar desigualdades válidas para modelos de PI com restrições de mochila.

Para reforçar a formulação (3.13)–(3.19), a FDV (3.26) foi aplicada às restrições de mochila (3.15). Seja f uma FDV. As seguintes desigualdades derivadas de (3.15) aplicando f são desigualdades válidas para a envolvente convexa inteira associada a (3.13)–(3.19):

$$\sum_{i=1}^n f\left(\frac{w_i}{W}\right) x_i^t \leq 1, \quad i = 1, \dots, n, \quad t = 1, \dots, T. \quad (3.26)$$

Entre todas as FDV, as funções maximais são aquelas que produzem resultados não-dominadas. Nestes testes, foram utilizadas as seguintes funções maximais descritas em Clautiaux et al. (2010): $f_{FS,1}$ (propostas inicialmente por Fekete e Schepers (2001)), com $k \in \mathbb{N} \setminus \{0\}$, e $f_{CCM,1}$ (descrito em Carlier et al. (2007)), com $C \in \mathbb{R}$ e $C \geq 1$.

$$\bullet f_{FS,1}(x; k) = \begin{cases} x, & \text{se } (k+1) * x \in \mathbb{N}, \\ \lfloor (k+1)x \rfloor / k, & \text{caso contrário.} \end{cases}$$

$$\bullet f_{CCM,1}(x; C) = \begin{cases} \lfloor Cx \rfloor / \lfloor C \rfloor, & \text{se } x < 1/2, \\ 1/2, & \text{se } x = 1/2, \\ 1 - f_{CCM,1}(1 - x), & \text{se } x > 1/2. \end{cases}$$

3.4.2 Um método de planos de corte

A partir das definições anteriores, foi definido o seguinte método de planos de corte para (3.13)–(3.19).

Algoritmo 1: Método de planos de corte

Resolver a relaxação linear de (3.13)–(3.19) (seja (z^*, x^*, y^*) a solução óptima);

enquanto os critérios de paragem não forem satisfeitos **faça**

Seja f^k uma FDV com parâmetro γ ;

para um dado intervalo de γ e um dado conjunto de restrições de sacco de mochila (3.15) **faça**

se a desigualdade (3.26) for violada para f^γ, x^* e a restrição actual

(3.15) **então**

adicionar esta restrição (3.26) à relaxação linear de (3.13)–(3.19);

fim

fim

Resolver novamente a relaxação linear de (3.13)–(3.19) (seja (z^*, x^*, y^*) a solução óptima);

fim

Para controlar o número de desigualdades geradas, pode limitar-se a gama de parâmetros das FDV assim como o número de restrições de mochila (3.15) a partir das quais as desigualdades válidas são derivadas. Além disso, os critérios de paragem podem incluir, por exemplo, o número de desigualdades a adicionar à relaxação linear ou o número máximo de iterações permitido sem uma melhoria significativa do valor da função de objectivo (com base num valor pré-definido).

3.5 Resultados computacionais

Para avaliar o desempenho das abordagens elaboradas neste trabalho e, em particular, do método de planos de corte descrito na Seção 3.4, foi realizado um conjunto de experiências computacionais em instâncias de referência da literatura. As instâncias consideradas correspondem a um subconjunto das instâncias de corte e empacotamento apresentadas em Arbib e Marinelli (2014), para as quais foram gerados prazos de entrega. Os valores de T (horizonte de planejamento) considerados correspondem aos descritos por estes autores. Um conjunto de 40 instâncias divididas em dois grupos de 20 foi gerado deste modo. Estes grupos, embora possuam o mesmo valor de T , possuem prazos de entrega diferentes (Arbib e Marinelli (2014)). Os testes foram executados num PC com um processador i7 com 3.5 GHz e 32 GB de RAM. Para as sub-rotinas de otimização, recorrendo ao CPLEX 12.5.

Nas Tabelas 3.1–3.2 são descritos os valores dos limites inferiores fornecidos pela relaxação linear de (3.13)–(3.19), tendo em conta os seguintes cenários:

- A : Relaxação linear (3.13)–(3.19) sem cortes;
- B : Relaxação linear (3.13)–(3.19) com cortes (3.21);
- C : idêntico a B incluindo os cortes obtidos pelo método descrito na Seção 3.4;
- D : idêntico a C incluindo os cortes (3.22)–(3.22).

As colunas $D1$ a $D4$ estendem os resultados anteriores para diferentes parâmetros do método de planos de corte discutidos na Seção 3.4. Sejam δ , max_{cuts} e it , respectivamente, o menor valor de violação de uma desigualdade (3.26) que desencadeia a sua inserção no modelo, o número máximo de cortes adicionados entre duas resoluções consecutivas da relaxação linear, e o número máximo de iterações que são permitidas sem uma melhoria do valor da solução da relaxação linear de valor superior a 0.001. Para os cenários C e D acima mencionados, os valores de (δ, max_{cuts}, it) considerados são $(0.1, 20, 20)$. As colunas $D1$ a $D4$ da Tabela 3.1 correspondem ao cenário D com os seguintes pares de valores para (δ, max_{cuts}, it) :

- $D1 : (\delta, max_{cuts}, it) = (0.1, 20, 50);$
- $D2 : (\delta, max_{cuts}, it) = (0.1, 50, 50);$
- $D3 : (\delta, max_{cuts}, it) = (0.01, 20, 20);$
- $D4 : (\delta, max_{cuts}, it) = (0.25, 20, 20).$

As colunas z_{RL} e t_{RL} das Tabelas 3.1–3.2 identificam respectivamente o valor da solução óptima e o seu tempo de computação no cenário correspondente. Note-se que foram utilizadas as duas funções duais válidas descritas na Secção 3.4 sem qualquer restrição sobre o domínio dos seus parâmetros. Em vez disso, foram utilizados os parâmetros (δ, max_{cuts}, it) para controlar o número de cortes adicionados à relaxação linear.

Enquanto a inequação (3.21) conduz a uma pequena melhoria no valor do limite inferior fornecido pela relaxação linear de (3.13)–(3.19), o impacto do procedimento de planos de corte descrito na Secção 3.4 é muito mais significativo. Em média, o valor do limite inferior aumentou quase 4,7% para o primeiro conjunto de instâncias, e 11,1% para o segundo conjunto (cenário A contra o cenário C). As desigualdades (3.22)–(3.24) não têm impacto sobre o primeiro conjunto de instâncias mas, quando combinadas com o procedimento de plano de corte, a melhoria média do limite inferior chega aos 15%. Isto pode ser explicado pelo facto dos planos de corte baseados nas FDV se concentrarem nas restrições de mochila do modelo e em particular nas variáveis x_i^t , enquanto que as desigualdades (3.22)–(3.24) se concentram nas restantes variáveis do modelo, completando assim o procedimento de planos de corte. Considerando os parâmetros (δ, max_{cuts}, it) , os melhores resultados em termos de qualidade dos limites inferiores são alcançados com o cenário $D2$ para ambos os conjuntos de instâncias. A correspondente melhoria média no valor do limite inferior é de respectivamente 8,9% e 19,2% para o primeiro e segundo conjuntos de instâncias. A melhoria máxima no primeiro conjunto de instâncias sobe para 30,9% e para 38,1% no segundo conjunto.

<i>Inst.</i>	<i>n</i>	<i>T</i>	A		B		C		D		D1		D2		D3		D4	
			<i>z_{RL}</i>	<i>t_{RL}</i>	<i>z_{RL}</i>	<i>t_{RL}</i>	<i>z_{RL}</i>	<i>t_{RL}</i>	<i>z_{RL}</i>	<i>t_{RL}</i>	<i>z_{RL}</i>	<i>t_{RL}</i>	<i>z_{RL}</i>	<i>t_{RL}</i>	<i>z_{RL}</i>	<i>t_{RL}</i>	<i>z_{RL}</i>	<i>t_{RL}</i>
1	20	481	264,83	0,23	267,90	4,15	268,133	13,60	328,00	275,07	328,00	299,55	267,762	6,55	311,00	274,30		
2	20	815	503,76	0,53	516,90	48,28	516,91	97,58	516,91	100,50	516,91	124,76	516,569	82,86	516,91	198,59		
3	20	703	397,35	0,47	407,72	10,86	400,908	13,06	407,03	49,16	407,03	61,33	407,027	55,79	400,80	53,01		
4	20	825	502,37	0,70	520,22	30,32	520,249	45,25	520,25	45,43	520,25	46,74	520,258	26,04	564,18	779,44		
5	20	619	368,24	0,30	376,41	12,67	370,196	12,05	389,75	124,44	378,46	89,35	370,196	9,46	369,44	18,28		
6	20	787	511,45	0,54	520,17	14,82	513,519	20,35	520,31	70,11	520,42	72,90	513,545	17,68	513,98	102,68		
7	20	675	370,53	0,41	371,60	14,67	386,397	278,75	403,00	717,63	403,00	917,87	371,759	20,73	371,11	40,69		
8	20	851	486,53	0,57	537,15	17,80	536,199	32,66	537,32	49,14	637,00	424,05	537,323	38,98	537,29	103,51		
9	20	683	390,71	0,44	424,04	42,24	409,967	58,43	409,97	55,27	423,70	99,96	409,967	51,17	449,00	377,05		
10	20	727	473,04	0,47	483,07	7,03	483,134	17,18	483,13	18,80	483,13	23,18	483,134	18,01	515,89	381,80		
11	20	475	279,35	1,85	280,32	1,09	280,409	2,94	280,41	4,52	280,51	4,97	280,409	2,87	280,41	3,51		
12	20	875	511,12	0,57	560,73	17,46	560,971	37,30	560,98	50,65	560,69	44,67	560,727	31,82	560,82	63,40		
13	20	687	366,33	0,44	371,70	10,50	371,664	14,30	471,00	692,77	471,00	613,30	371,752	16,71	438,30	873,37		
14	20	703	441,90	0,44	443,04	10,57	452,59	177,12	452,67	169,03	462,42	296,38	442,96	10,67	442,21	6,66		
15	20	747	473,52	0,44	547,50	103,29	547,5	242,32	547,50	272,47	547,50	280,47	501,221	51,08	525,75	497,08		
16	20	645	355,41	0,42	364,30	7,30	369,66	33,31	369,74	42,53	369,68	39,66	364,204	14,05	414,00	888,98		
17	20	711	475,20	0,44	496,56	5,12	496,723	9,53	504,72	33,62	515,49	73,90	496,742	9,09	522,88	275,17		
18	20	535	339,61	2,37	340,25	0,25	365,39	75,35	365,46	150,99	365,46	148,69	343,075	38,85	340,34	26,41		
19	20	661	419,26	0,37	423,04	0,40	451,10	25,98	450,85	62,51	450,94	69,80	435,591	37,66	423,11	28,06		
20	20	773	495,68	0,53	517,07	14,11	517,106	31,33	517,17	36,08	536,02	217,57	517,166	28,34	517,39	180,54		
<i>média</i>			421,31	0,63	441,14	23,68	439,68	60,83	451,81	151,04	458,88	197,45	435,57	28,42	450,74	258,63		

Tabela 3.1: Resultados computacionais

Inst.	n	T	A		B		C		D		D1		D2		D3		D4	
			z_{RL}	t_{RL}	z_{RL}	t_{RL}	z_{RL}	t_{RL}	z_{RL}	t_{RL}	z_{RL}	t_{RL}	z_{RL}	t_{RL}	z_{RL}	t_{RL}	z_{RL}	t_{RL}
21	20	481	290,51	0,31	292,71	0,33	338,60	28,28	337,67	29,70	350,93	70,67	350,93	227,76	337,67	44,70	343,79	144,28
22	20	815	528,57	0,40	530,57	0,44	582,00	10,13	590,11	42,00	607,83	165,07	607,84	201,02	587,73	29,67	589,91	145,16
23	20	703	441,44	0,29	444,21	0,29	501,35	5,47	536,83	68,07	537,07	119,97	537,08	278,43	484,22	15,63	537,04	191,49
24	20	825	545,66	0,43	554,58	0,47	645,00	13,92	579,75	22,68	659,19	214,13	659,19	883,62	587,89	50,36	659,19	287,84
25	20	619	387,29	0,33	388,49	0,36	431,51	27,24	442,65	70,25	442,65	66,47	442,65	192,47	443,28	126,16	431,88	71,46
26	20	787	544,06	0,35	544,45	0,35	633,00	10,66	651,86	68,03	651,92	85,62	651,86	262,19	651,86	256,39	651,86	67,80
27	20	675	407,52	0,36	407,84	0,41	439,50	37,43	447,12	93,20	447,12	85,33	447,12	183,44	448,52	240,55	445,45	49,00
28	20	851	494,68	0,42	538,39	0,51	543,02	6,87	568,76	18,84	683,16	463,53	683,22	817,41	568,50	18,58	669,82	231,24
29	20	683	419,91	0,34	424,27	0,42	469,95	4,92	529,93	78,09	529,93	85,13	529,93	272,00	529,93	269,44	529,93	106,40
30	20	727	538,88	0,35	540,01	0,38	603,00	6,32	613,84	49,56	613,86	81,28	613,86	249,34	613,84	78,63	613,84	95,99
31	20	475	303,46	0,24	304,42	0,26	320,03	6,02	329,24	24,08	329,24	22,54	329,24	68,63	330,69	64,04	329,11	21,54
32	20	875	581,92	0,33	616,93	0,46	715,00	13,37	736,32	105,90	736,32	121,21	736,32	370,26	648,76	27,87	736,32	81,51
33	20	687	427,85	0,26	428,88	0,29	445,15	3,05	536,77	81,78	536,77	71,00	536,77	213,52	536,77	322,55	536,72	51,73
34	20	703	471,20	0,32	471,40	0,35	489,10	16,62	503,76	15,16	531,32	76,08	531,32	216,78	531,55	142,37	503,47	37,80
35	20	747	496,21	0,42	504,09	0,46	509,68	3,82	581,83	131,04	581,83	122,02	581,66	510,64	584,13	462,79	580,15	156,05
36	20	645	409,30	0,39	416,26	0,37	468,25	8,49	475,79	57,32	475,79	67,06	475,79	255,54	475,64	134,07	475,79	161,29
37	20	711	498,86	0,41	516,78	0,45	525,87	8,07	555,42	18,77	611,17	170,04	610,86	623,89	555,50	13,68	561,52	66,63
38	20	535	356,18	0,33	356,82	0,36	400,90	23,31	381,81	16,19	408,37	141,29	408,37	243,16	381,19	11,71	399,64	62,48
39	20	661	449,39	0,33	453,17	0,36	465,30	4,09	511,60	87,38	511,60	230,48	511,60	325,48	498,75	26,20	505,83	97,91
40	20	773	536,03	0,34	541,31	0,38	614,00	8,84	625,24	71,83	625,24	287,96	635,24	246,81	571,30	14,15	635,24	257,57
<i>média</i>			456,45	0,35	463,78	0,38	507,01	12,35	526,81	57,49	543,56	137,34	544,04	332,12	518,39	117,47	536,82	119,26

Tabela 3.2: Resultados computacionais

Adicionalmente, o primeiro conjunto de instâncias foi resolvido até à integralidade, com um limite de tempo de 3600 segundos para o método de partição e avaliação, de forma semelhante à abordagem seguida em Arbib e Marinelli (2014). Para este efeito, foi utilizado o procedimento de plano de corte com as definições do cenário D . Após o limite de tempo, o valor do melhor limite inferior encontrado foi melhor em 13,3% em relação ao limite na raiz da árvore de pesquisa. Por outro lado, o intervalo de integralidade médio foi de 20,4%, um valor que está em linha com os valores descritos em Arbib e Marinelli (2014) e obtidos através de um modelo compacto.

3.6 Um novo modelo exacto de fluxo em rede pseudo-polinomial

O CSSP pode ser formulado usando um modelo de fluxo em rede pseudo-polinomial definido num grafo $G = (V, A)$ onde V representa o conjunto de vértices e A o conjunto de arcos. Os vértices em V representam posições dentro dos rolos a partir da posição mais à esquerda 0 até ao vértice W mais à direita, *i.e.* $V = \{0, 1, 2, \dots, W\}$. O conjunto de arcos A é dividido em D conjuntos A^t , $t = 1, \dots, D$, com D sendo a duração do horizonte de planeamento. Um arco (r, s) em A^t representa um item de tamanho $s - r$ a ser cortado na posição i de um rolo no instante de tempo t . Os desperdícios nos rolos são representados através de arcos de comprimento unitário. O CSSP pode ser formulado como um problema de fluxo de custo mínimo com restrições adicionais. A definição formal desse modelo é indicada a seguir.

$$\min. \sum_{t=1}^D z^t + \sum_{i=1}^n \sum_{t=d_i+1}^D y_i^t \quad (3.27)$$

sujeito a

$$-\sum_{(r,s) \in A^t} x_{rs}^t + \sum_{(s,u) \in A^t} x_{su}^t = \begin{cases} z^t, & \text{if } s = 0, \\ -z^t, & \text{if } s = W, s = 0, \dots, W, t = 1, \dots, D, \\ 0, & \text{caso contrário.} \end{cases} \quad (3.28)$$

$$\sum_{(r,r+w_i) \in A^t} \sum_{t=1}^D x_{r,r+w_i}^t = b_i, \quad i = 1, \dots, n, \quad (3.29)$$

$$\sum_{(r,r+w_i) \in A^t} x_{rs}^t \leq L_i^{\min} y_i^t, \quad \forall (r,s) \in A^t, t = d_i + 1, \dots, D, \quad (3.30)$$

$$y_i^{t+1} \leq y_i^t, \quad i = 1, \dots, n, \quad t = d_i + 1, \dots, D, \quad (3.31)$$

$$z^{t+1} \leq z^t, \quad t = 1, \dots, D - 1, \quad (3.32)$$

$$x_{rs}^t \in \{0, 1\}, \quad \forall (r,s) \in A^t, t = 1, \dots, D, \quad (3.33)$$

$$y_i^t \in \{0, 1\}, \quad i = d_i + 1, \dots, n, t = 1, \dots, D, \quad (3.34)$$

$$z^t \in \{0, 1\}, \quad t = 1, \dots, D. \quad (3.35)$$

As variáveis de fluxo x_{rs}^t indicam se um item de tamanho $s - r$ é cortado a partir da posição r de um rolo no período de tempo t , enquanto as variáveis y_i^t tomam o valor de 1 se pelo menos um item i é cortado no instante de tempo t ou mais tarde para $t > d_i$, ou 0 caso contrário. O atraso de um item i é representado como a soma das variáveis y_i^t de $d_i + 1$ até D , *i.e.* $\sum_{t=(d_i+1)}^D y_i^t$. As variáveis z^t , $t = 1, \dots, D$, indicam se um rolo está a ser cortado no instante de tempo t , e, portanto, $\sum_{t=1}^D z^t$ corresponde ao número total de rolos usados. A função objectivo (3.27) considera tanto o uso de material total como o atraso total relacionado com o escalonamento. As restrições (3.28) asseguram a conservação do fluxo ao longo do grafo G . As restrições de procura são representadas através de (3.29). As restrições (3.30) forçam as variáveis y_i^t a tomar o valor de 1 se um item i é cortado no instante de tempo t . Neste último caso, as restrições (3.31) permitem a ativação de todas variáveis $y_i^{t'}$ de $t' = t$ até $d_i + 1$. Nas restrições (3.30), L_i^{\min} representa o limite superior do número de itens i que podem ser cortados num rolo, como descrito em (3.20). Conforme referido anteriormente, é necessária exactamente uma unidade de tempo para cortar um rolo. Assim, o corte de um rolo num instante de tempo t significa que outros rolos foram cortados anteriormente até t . As restrições (3.32) apoiam esta definição, proibindo um rolo de ser cortados no instante de tempo $t + 1$ se nenhum rolo foi cortado no período t .

O modelo (3.27)–(3.35) difere de (3.1)–(3.5) e (3.6)–(3.12) no sentido em que não está explicitamente baseado na afectação dos padrões de corte, mas na atribuição directa de itens i a períodos de tempo. O modelo é exato, uma vez que não impõe

restrições sobre os períodos de tempo em que cada item pode ser cortado, como acontece em Reinertsen e Vossen (2010). Além disso, o seu tamanho é pseudo-polinomial, e, portanto, é de tratamento mais fácil do que os outros dois modelos de programação inteira descritos em Reinertsen e Vossen (2010) e Arbib e Marinelli (2014).

3.7 Uma abordagem heurística baseada no modelo de fluxos em rede

3.7.1 Visão geral

Nesta secção, é descrito um procedimento para a resolução do CSSP baseado no modelo (3.27)–(3.35) que tira partido da sua estrutura e dimensão para gerar soluções válidas de boa qualidade. A abordagem baseia-se numa versão revista deste modelo de fluxos em rede que é de dimensão mais pequena do que (3.27)–(3.35), e que é resolvido até à optimalidade usando um *solver* de optimização. Esta versão revista agrega instantes de tempo utilizados em (3.27)–(3.35) em períodos de tempo de uma determinada duração. A solução não define uma afectação precisa para as operações de corte, uma vez que os cortes estão afectados por período agregado, e a ordem pela qual estes cortes devem ser realizados dentro do período de tempo é livre. Como consequência, o atraso associado a cada item é avaliado pelo seu valor sobrestimado dentro de cada período de tempo e para cada item. Assim, a solução do modelo de fluxos em rede revisto fornece um limite superior para o CSSP, uma vez que o seu valor é forçosamente maior ou igual que o valor da solução de qualquer afectação válida dos cortes dentro dos prazos determinados. Daqui em diante, este limite superior será referido como limite superior sobrestimado.

Após a resolução do modelo de fluxos revisto, são calculadas as afectações precisas através de um procedimento que consiste na resolução de diferentes modelos de afectação relacionados com os períodos de tempo. Deste modo, serão definidos os instantes de tempo em que os padrões gerados pelo modelo de fluxos de rede revisto devem ser cortados. A solução obtida para o CSSP com este processo de afectação de tempo permite uma estimação correcta do atraso correspondente, calculando-se assim

um limite superior para o problema que é menor do que o obtido com o modelo de fluxo de rede revisto.

A Figura 3.1 apresenta um esquema desta abordagem heurística baseado no modelo, em que P representa uma instância do CSSP. Nas próximas subsecções, será formalmente apresentado o modelo de fluxos em rede revisto, assim como a formulação de afectação no tempo acima referidos.

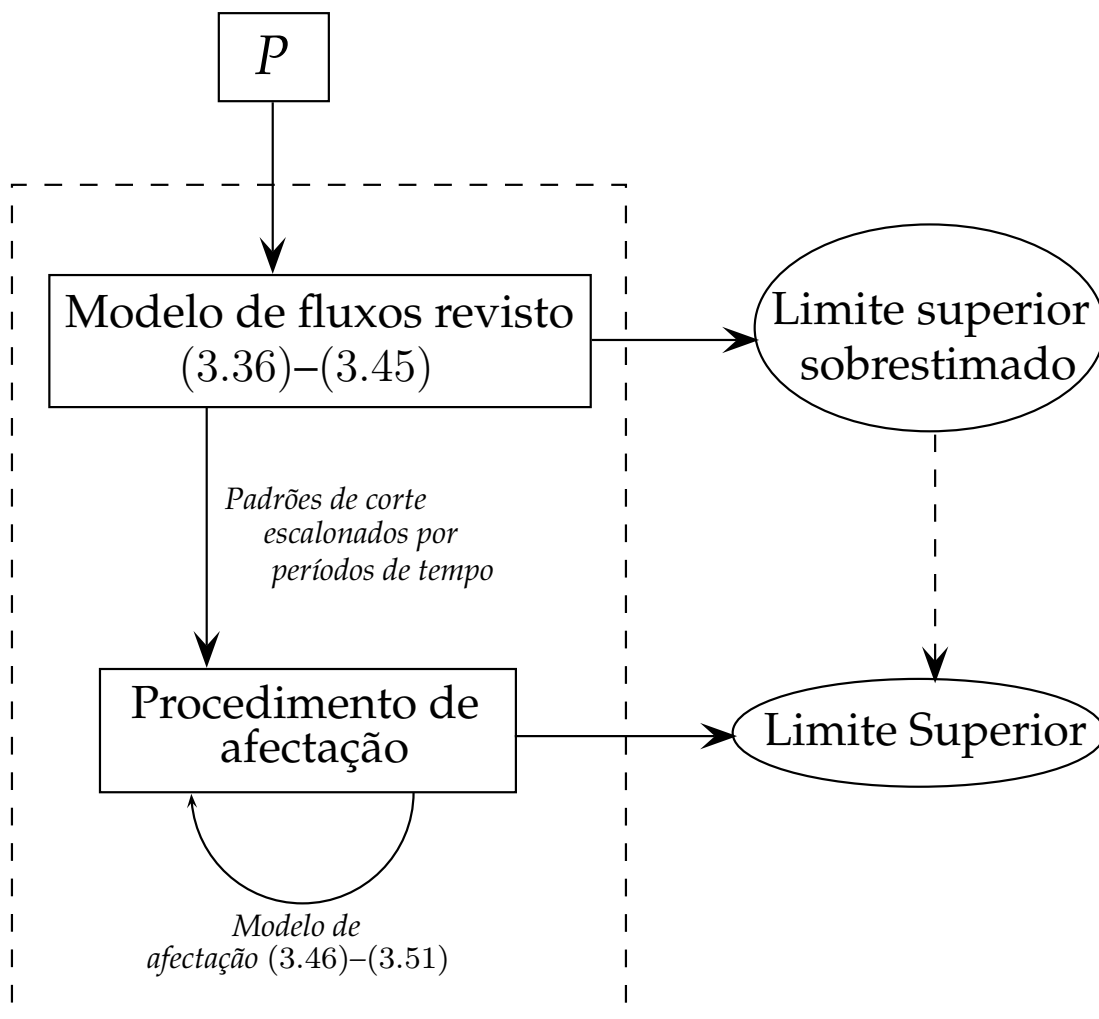


Figura 3.1: Esquema da heurística para o modelo de fluxos em rede revisto

3.7.2 Modelo de fluxos em rede revisto

O modelo de fluxos em rede (3.27)–(3.35) considera períodos de tempo de uma unidade de duração. Para reduzir o seu tamanho, podem ser usados períodos de tempo

mais longos através da agregação de diferentes instantes de tempo consecutivos. Designamos por T o número de períodos que são considerados. De notar que T pode assumir qualquer valor e, portanto, não tem de ser restrito ao número de itens diferentes, nem o último instante de tempo tem que coincidir com o prazo de entrega de um determinado item. As variáveis de fluxo do modelo de fluxos em rede revisto são semelhantes às de (3.27)–(3.35), à excepção das variáveis de fluxo passarem a ser definidas como inteiras. Seja $[a_{t-1}, a_t]$ o período de tempo t , sendo a_t o último instante de tempo do período t . Tem-se $a_0 = 0$ e $a_T = D$, onde D representa a duração do horizonte de planeamento. O modelo de fluxos em rede revisto é formalmente definido como se segue.

$$\min. \sum_{t=1}^T z^t + \sum_{i=1}^n \sum_{t=t_i}^T (a_t - a_{t-1}) y_i^t \quad (3.36)$$

sujeito a

$$-\sum_{(r,s) \in A^t} x_{rs}^t + \sum_{(s,u) \in A^t} x_{su}^t = \begin{cases} z^t, & \text{if } s = 0, \\ -z^t, & \text{if } s = W, s = 0, \dots, W, t = 1, \dots, T, \\ 0, & \text{caso contrário} \end{cases} \quad (3.37)$$

$$\sum_{(r,r+w_i) \in A^t} \sum_{t=1}^T x_{r,r+w_i}^t = b_i, \quad i = 1, \dots, n, \quad (3.38)$$

$$\sum_{(r,r+w_i) \in A^t} x_{rs}^t \leq L_i^{\min} y_i^t, \quad \forall (r,s) \in A^t, t = t_i, \dots, T, \quad (3.39)$$

$$y_i^{t+1} \leq y_i^t, \quad i = 1, \dots, n, \quad t = t_i, \dots, T, \quad (3.40)$$

$$z^t \leq a_t - a_{t-1}, \quad t = 1, \dots, T, \quad (3.41)$$

$$z^t \geq (a_t - a_{t-1}) y_i^{t+1}, \quad i = 1, \dots, n, t = t_i, \dots, T - 1, \quad (3.42)$$

$$x_{rs}^t \geq 0 \text{ e inteiro}, \quad \forall (r,s) \in A^t, t = 1, \dots, T, \quad (3.43)$$

$$y_i^t \in \{0, 1\}, \quad i = 1, \dots, n, t = t_i, \dots, T, \quad (3.44)$$

$$z^t \geq 0 \text{ e inteiro}, \quad t = 1, \dots, T. \quad (3.45)$$

A definição de (3.36)–(3.45) é semelhante em muitos aspectos à definição de (3.27)–(3.35). Uma das diferenças chave encontra-se na definição do atraso associado a um

item i . Na função objectivo (3.36), a contribuição para o atraso total de um item i que é cortado num determinado período t é igual à sua duração. Este valor pode sobrestimar o atraso total de um item i pois considera-se implicitamente que a última instância desse item é sempre cortada no último instante de tempo do período. O valor t_i representa o período de tempo que inclui a data de entrega d_i do item i , e pode ser definido da seguinte forma: $t_i = \min\{t' : a_{t'} \geq d_i, t' = 1, \dots, T\}$. As variáveis z^t tornam-se também variáveis inteiras representando o número de rolos no período de tempo t . As restrições (3.41) asseguram que não são usados mais do que $a_t - a_{t-1}$ rolos no período de tempo t , uma vez que é preciso exactamente uma unidade de tempo para cortar um rolo. As restrições (3.42) são uma extensão das restrições (3.32) para o caso da possibilidade de mais do que um rolo poder ser cortado num determinado período de tempo, e de os períodos de tempo poderem não ter todos a mesma duração.

A desvantagem do modelo (3.36)–(3.45) é que, enquanto os padrões de corte são claramente determinados, a afectação das operações de corte correspondentes pode não ser completamente definida, uma vez que pode não ser evidente a ordem pela qual os padrões de corte devem ser executados dentro de um período de tempo. Para ultrapassar esta dificuldade, recorreu-se a um procedimento de afectação no tempo que se baseia no modelo de programação inteira descrito na próxima secção e que recebe como entrada os padrões de corte gerados por (3.36)–(3.45). Este procedimento determina a melhor afectação desses padrões de corte dentro de cada período de tempo agregado.

3.7.3 Modelo de afectação no tempo

Seja P' o conjunto de padrões de corte gerados através da resolução do modelo (3.36)–(3.45) para um determinado período de tempo t' de duração T' definido por $[a_{t'-1}, a_{t'}]$, e seja α_{ip} o número de itens i no padrão p . Note-se que, se $|P'| < T'$, o valor de T' passa a ser igual a $|P'|$, tornando-se a definição do período de tempo resultante $[a_{t'-1}, a_{t'-1} + |P'|]$. Para afectar os padrões de corte de P' do período de tempo t' de modo a minimizar o atraso total associado aos itens, foi considerada a seguinte formulação de afectação com restrições adicionais:

$$\min \sum_{i=1}^n y_i \quad (3.46)$$

sujeito a

$$\sum_{t=a_{t'}-1}^{a_{t'}} \lambda_{pt} = 1, \quad p \in P', \quad (3.47)$$

$$\sum_{p \in P'} \lambda_{pt} = 1, \quad t = a_{t'-1}, a_{t'-1} + 1, \dots, a_{t'}, \quad (3.48)$$

$$y_i \geq (t - d_i) \lambda_{pt}, \quad i = 1, \dots, n, p \in P' \text{ de tal modo que } \alpha_{ip} > 0,$$

$$t = a_{t'-1}, a_{t'-1} + 1, \dots, a_{t'}, \text{ de tal modo que } t > d_i, \quad (3.49)$$

$$\lambda_{pt} \in \{0, 1\}, \quad p \in P', t = a_{t'-1}, a_{t'-1} + 1, \dots, a_{t'}, \quad (3.50)$$

$$y_i \geq 0 \text{ e inteiro}, \quad i = 1, \dots, n. \quad (3.51)$$

As variáveis de afectação λ_{pt} determinam se o padrão de corte p é atribuído ou não ao período de tempo t (de duração unitária), enquanto que as variáveis y_i representam o atraso do item i . A função objectivo (3.46) considera a minimização do atraso total associado aos itens. As restrições (3.47) e (3.48) são restrições de afectação que garantem, respectivamente, que todos os padrões de corte sejam atribuídos a um instante de tempo em t' , e todos os instantes de tempo tenham exactamente um padrão de corte atribuído. As restrições (3.49) forçam as variáveis y_i a ter um valor igual ao maior atraso das diferentes instâncias do item i cortadas neste período de tempo t' . Juntamente com a função objectivo (3.46), estas restrições asseguram que o atraso de cada item incluído nos padrões de P' seja correctamente avaliado.

Exemplo 1. Considere uma solução do modelo anterior (3.36)–(3.45), que inclui itens de tamanhos 3, 5 e 7 e rolos de dimensão $W = 10$. Considere-se, em particular, um período de tempo t' de duração 3, ao qual são afectados na solução os dois últimos itens de cada um dos tamanhos considerados ($x_{0,7}^{t'}, x_{7,10}^{t'} = 2$ e $x_{0,5}^{t'}, x_{0,10}^{t'} = 1$). Suponha-se que as datas de entrega dos itens de tamanho 3 e 7 correspondem ao último instante do período t' e que a data de entrega dos itens de tamanho 5 corresponde ao primeiro instante de tempo do período $t' + 1$. Uma vez que o modelo (3.36)–(3.45) sobrestima o atraso dos itens, o atraso considerado pelo modelo de fluxos revisto em

relação ao item 5 é de três instantes de tempo.

É possível corrigir o atraso deste item escolhendo uma boa afectação de padrões aos instantes agregados no período t' . A Figura 3.2 mostra uma afectação proposta pelo modelo (3.46)–(3.51), que minimiza os atrasos neste período. Neste caso, o atraso total considerado neste período passa a ser igual a 1, uma vez que todos os itens de tamanho 5 foram terminados no primeiro instante do período agregado, ou seja, apenas um período após o correspondente prazo de entrega.

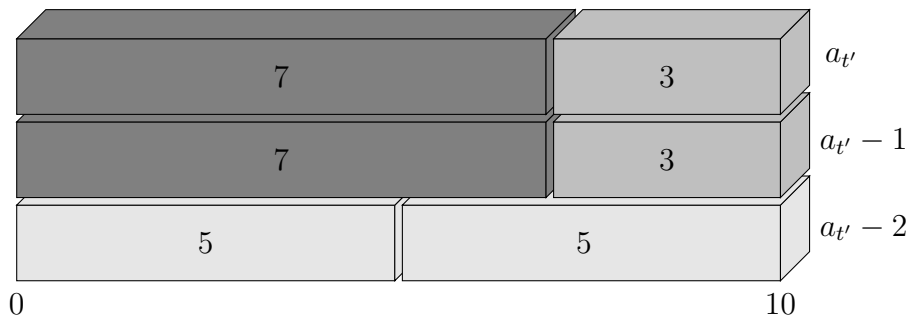


Figura 3.2: Exemplo de tempos agregados - padrões otimizados.

3.8 Resultados computacionais

Para avaliar o desempenho da heurística baseada no modelo de fluxos em rede, foi realizado um conjunto de testes computacionais em instâncias de referência da literatura descritas em Arbib e Marinelli (2014). As instâncias têm 20 itens de tamanhos w_i , $i = 1, \dots, n$, que variam no intervalo $[100, 7000]$, e rolos de comprimento $W = 10000$. Os valores das procuras b_i dos itens pertencem ao conjunto $[1, 100]$. As datas de entrega dos itens são definidas da seguinte forma: $d_1 = \dots = d_t = \frac{\overline{C}_{max}}{2}$ e $d_{t+1} = \dots = d_n$, com $t = \lfloor \frac{n}{2} \rfloor$ e \overline{C}_{max} sendo o *makespan* de uma solução que minimiza o número de rolos usados. Vinte instâncias diferentes foram geradas desta forma. Os testes foram executados num PC com um processador i7 com 3,5 GHz e 32 GB de RAM. Além disso, foi utilizado o CPLEX 12.5 para implementar algumas das sub-rotinas de optimização.

Um dos objectivos destas experiências computacionais é o de analisar o impacto de alguns parâmetros importantes na qualidade das soluções fornecidas por esta abor-

dagem. Foi comparado, em particular, o desempenho da heurística para números diferentes de períodos de tempo T no modelo (3.36)–(3.45), e para diferentes durações do horizonte de planeamento D . Nas Tabelas 3.3–3.6, descrevem-se os resultados para dois valores de T , *i.e.* $T \in \{20, 40\}$, e quatro valores de D com $D = \bar{C}_{max} + \alpha \bar{C}_{max} + 1$ e $\alpha \in \{0.05, 0.1, 0.5, 1\}$. Ambos os parâmetros determinam o tamanho do modelo (3.36)–(3.45). Quanto maior for o seu valor, maior será o tamanho desse modelo. Para $T = 20$, foi utilizado um limite de tempo de 3300 segundos para resolver o modelo de fluxos em rede revisto (3.36)–(3.45), e de 300 segundos para o modelo de afectação de tempo (3.46)–(3.51). Para $T = 40$, estes tempos foram de, respectivamente, 3500 e 100 segundos. Em ambos os casos, o tempo de processamento total foi limitado a 1 hora. O significado das colunas nas Tabelas 3.3–3.6 é o seguinte:

Inst.: número da instância;

T: número de períodos de tempo do modelo de fluxos em rede revisto (3.36)–(3.45);

best_lb: melhor limite inferior conhecido para a instância;

ub: limite superior fornecido pelo modelo correspondente dentro do tempo limite;

t_{ub}: tempo necessário para atingir o melhor limite superior fornecido pelo modelo de fluxos em rede revisto (3.36)–(3.45);

gap: intervalo de optimalidade associado à solução fornecida pelo modelo de fluxos em rede revisto (3.36)–(3.45);

t: tempo (em segundos) necessário para obter um limite superior do modelo correspondente;

final gap: intervalo de optimalidade entre a solução fornecida pelo modelo de afectação no tempo (3.46)–(3.51) e o melhor limite inferior do problema, *i.e.* $final\ gap = (ub - best_lb)/best_lb$, com *ub* sendo o valor da melhor solução encontrada pelo modelo de afectação no tempo.

Uma entrada “-” nas Tabelas 3.3-3.6 significa que o CPLEX não conseguiu encontrar uma solução inteira para o modelo de fluxos em rede revisto.

As diferenças médias entre os limites superiores fornecidos por esta abordagem e os melhores limites inferiores conhecidas são em geral baixas, geralmente inferior a 6%. A menor diferença média, de 3,88%, é obtida com $T = 20$ e $D = 1.5\bar{C}_{max} + 1$. Num número significativo de casos, a heurística fornece uma solução óptima para o problema. É importante notar que a heurística não começa com uma solução válida encontrada através de um método diferente. Assim, os resultados aqui apresentados ilustram apenas a capacidade desta abordagem em gerar boas soluções incumbentes para este problema.

O modelo de fluxos em rede revisto (3.36)–(3.45) torna-se naturalmente mais difícil de resolver à medida em que o valor de T aumenta. Consequentemente, o intervalo de optimalidade das soluções obtidas resolvendo este modelo com *CPLEX* tende a aumentar com T . Este resultado parece ter um impacto sobre a qualidade da solução final. A média do desvio de optimalidade entre o melhor limite superior conhecido e o melhor limite inferior conhecido para o problema são pequenos para $T = 20$ e para $T = 40$. O mesmo acontece com o valor de D : os intervalos de optimalidade das soluções do modelo de fluxos em rede revisto (3.36)–(3.45) são geralmente menores quando o valor de D diminui.

O número de modelos de afectação no tempo (3.46)–(3.51) que são resolvidos aumenta com T , enquanto que o tamanho destes modelos diminui quando T aumenta. No entanto, o modelo de afectação no tempo é resolvido até à optimalidade em poucos milissegundos para ambos $T = 20$ e $T = 40$. Como os resultados das Tabelas 3.3-3.6 mostram, este modelo ajuda a melhorar a solução obtida com o modelo de fluxo em rede revisto ao calcular uma afectação precisa dos padrões ao longo do tempo, reduzindo assim o valor sobrestimado do atraso total feito pelo modelo de fluxos em rede revisto.

<i>Inst.</i>	<i>T</i>	<i>best</i> <i>lb</i>	Modelo de fluxos em rede revisto (3.36)–(3.45)				Modelo de afectação (3.46)–(3.51)		<i>final</i> <i>gap</i>
			<i>ub</i>	<i>t_{ub}</i>	<i>gap</i>	<i>t</i>	<i>ub</i>	<i>t</i>	
1	20	240	240	38,80	0,00	38,80	240	0,00	0,00
2		488	548	1305,75	0,00	2948,68	538	0,10	10,25
3		383	423	97,25	11,48	3304,74	407	0,06	6,27
4		475	518	82,91	0,00	175,69	498	0,09	4,84
5		309	309	22,04	0,00	22,48	309	0,00	0,00
6		421	436	64,43	0,00	395,63	436	0,04	3,56
7		348	364	598,73	0,00	2485,34	349	0,02	0,29
8		451	451	36,18	0,00	100,32	451	0,03	0,00
9		514	580	2812,27	4,47	3304,01	570	0,11	10,89
10		384	416	416,77	4,39	3307,80	403	0,04	4,95
11		257	280	1957,64	10,37	3301,70	280	0,04	8,95
12		503	518	53,73	0,00	235,30	506	0,02	0,60
13		399	424	36,32	0,00	420,39	419	0,06	5,01
14		416	462	689,43	11,95	3302,98	441	0,06	6,01
15		391	405	40,37	0,00	782,03	391	0,01	0,00
16		362	382	30,55	0,00	185,29	382	0,08	5,52
17		435	483	1684,70	0,00	2987,75	479	0,09	10,11
18		267	267	1615,56	0,00	1910,44	267	0,00	0,00
19		332	351	1993,93	5,90	3302,20	332	0,00	0,00
20		497	542	494,00	0,00	745,31	530	0,05	6,64
1	40	240	240	289,55	0,00	290,75	240	0,00	0,00
2		488	544	696,35	15,44	3502,77	543	0,00	11,27
3		383	404	1669,50	8,88	3501,47	402	0,02	4,96
4		475	505	630,00	0,00	630,15	498	0,05	4,84
5		309	309	1170,56	0,00	1170,56	309	0,00	0,00
6		421	436	182,50	0,00	1460,40	431	0,03	2,38
7		348	353	448,38	1,58	3503,98	349	0,02	0,29
8		451	451	173,00	0,00	2007,54	451	0,01	0,00
9		514	582	2537,88	24,59	3503,00	581	0,05	13,04
10		384	406	2651,07	6,14	3502,38	404	0,03	5,21
11		257	337	3500,00	24,53	3502,39	333	0,05	29,57
12		503	504	705,12	0,00	985,02	504	0,02	0,20
13		399	416	923,80	0,00	1892,92	415	0,03	4,01
14		416	463	2411,17	11,97	3501,66	445	0,05	6,97
15		391	393	331,08	0,00	331,08	391	0,00	0,00
16		362	372	197,81	0,00	574,92	372	0,03	2,76
17		435	496	3500,00	13,54	3505,28	494	0,05	13,56
18		267	-	-	-	-	-	-	-
19		332	351	3336,00	5,88	3504,08	332	0,01	0,00
20		497	554	1368,25	11,26	3501,75	554	0,03	11,47

Tabela 3.3: Resultados computacionais com $D = 2\overline{C}_{max} + 1$

<i>Inst.</i>	<i>T</i>	<i>best</i> <i>lb</i>	Modelo de fluxos em rede revisto (3.36)–(3.45)				Modelo de afectação (3.46)–(3.51)		<i>final</i> <i>gap</i>
			<i>ub</i>	<i>t_{ub}</i>	<i>gap</i>	<i>t</i>	<i>ub</i>	<i>t</i>	
			1	20	240	240	97,63	0,00	
2		488	548	1010,83	0,00	3286,98	535	0,09	9,63
3		383	412	268,45	1,17	3303,79	407	0,06	6,27
4		475	505	35,46	0,00	123,70	498	0,09	4,84
5		309	309	21,11	0,00	21,54	309	0,00	0,00
6		421	436	49,18	0,00	948,90	432	0,06	2,61
7		348	353	246,75	0,00	2971,29	349	0,01	0,29
8		451	451	150,00	0,00	703,05	451	0,03	0,00
9		514	569	1753,09	0,91	3304,38	564	0,11	9,73
10		384	405	750,18	4,17	3304,61	403	0,03	4,95
11		257	280	3300,00	9,08	3301,46	270	0,04	5,06
12		503	516	819,64	5,43	3308,27	516	0,09	2,58
13		399	424	578,45	0,00	741,89	420	0,05	5,26
14		416	451	2050,00	4,66	3305,00	440	0,07	5,77
15		391	393	302,45	0,00	587,69	391	0,01	0,00
16		362	377	39,54	0,00	639,21	377	0,06	4,14
17		435	483	330,06	1,04	1487,23	478	0,08	9,89
18		267	267	1153,00	0,00	1262,22	267	0,00	0,00
19		332	341	1178,36	3,12	3303,21	332	0,00	0,00
20		497	530	485,50	0,00	512,46	530	0,04	6,64
1	40	240	240	223,96	0,00	223,96	240	0,00	0,00
2		488	552	2870,10	15,65	3502,97	551	0,05	12,91
3		383	405	1723,26	6,23	3504,86	405	0,04	5,74
4		475	499	393,73	0,00	613,31	498	0,05	4,84
5		309	309	51,14	0,00	52,01	309	0,00	0,00
6		421	436	2408,82	4,77	3507,28	431	0,03	2,38
7		348	353	600,93	1,55	3509,89	349	0,01	0,29
8		451	451	420,00	3,15	3503,87	451	0,01	0,00
9		514	613	1065,82	21,80	3502,70	610	0,06	18,68
10		384	395	2784,60	2,73	3502,47	391	0,02	1,82
11		257	330	3500,00	22,93	3502,41	326	0,01	26,85
12		503	504	768,55	6,85	3508,00	504	0,02	0,20
13		399	416	2496,82	0,00	2923,29	415	0,03	4,01
14		416	435	2851,00	6,33	3504,26	429	0,03	3,13
15		391	393	440,63	0,00	1913,24	391	0,01	0,00
16		362	372	212,60	0,00	1473,37	372	0,02	2,76
17		435	482	2955,72	8,14	3502,28	478	0,04	9,89
18		267	267	3335,56	0,02	3503,22	267	0,00	0,00
19		332	351	3500,00	5,71	3502,38	332	0,01	0,00
20		497	554	2996,00	11,63	3501,40	554	0,03	11,47

Tabela 3.4: Resultados computacionais com $D = 1.5\overline{C}_{max} + 1$

<i>Inst.</i>	<i>T</i>	<i>best lb</i>	Modelo de fluxos em rede revisto (3.36)-(3.45)				Modelo de afectação (3.46)-(3.51)		<i>final gap</i>
			<i>ub</i>	<i>t_{ub}</i>	<i>gap</i>	<i>t</i>	<i>ub</i>	<i>t</i>	
1	20	240	240	109,90	0,00	109,90	240	0,45	0,00
2		488	538	659,60	0,00	2308,64	537	0,09	10,04
3		383	407	206,92	0,00	1080,52	407	0,03	6,27
4		475	498	27,49	0,00	89,09	498	0,56	4,84
5		309	309	22,22	0,00	22,63	309	0,02	0,00
6		421	431	52,93	0,00	117,18	431	0,04	2,38
7		348	349	71,08	0,00	102,51	349	0,00	0,29
8		451	451	69,20	0,00	637,35	451	0,03	0,00
9		514	581	3265,18	4,57	3301,02	579	0,09	12,65
10		384	403	1204,99	4,32	3303,21	403	0,03	4,95
11		257	270	3300,00	5,75	3301,40	268	0,03	4,28
12		503	516	176,73	0,00	1659,84	515	0,05	2,39
13		399	420	910,95	0,00	936,44	420	0,05	5,26
14		416	465	692,26	12,04	3300,97	465	0,06	11,78
15		391	391	49,51	0,00	49,51	391	0,02	0,00
16		362	373	18,94	0,00	48,81	372	0,03	2,76
17		435	478	3054,19	3,60	3300,92	478	0,06	9,89
18		267	267	1601,07	0,00	1490,50	267	0,00	0,00
19		332	333	3543,27	0,79	3302,92	332	0,00	0,00
20		497	533	893,21	0,00	1131,80	532	0,03	7,04
1	40	240	240	367,44	0,00	295,82	240	0,00	0,00
2		488	553	2834,88	15,21	3502,69	552	0,04	13,11
3		383	419	2028,25	9,36	3501,59	409	0,04	6,79
4		475	498	773,36	0,00	1466,09	498	0,05	4,84
5		309	309	67,02	0,00	67,02	309	0,00	0,00
6		421	431	194,48	0,00	961,68	431	0,03	2,38
7		348	349	859,28	0,00	1358,65	349	0,02	0,29
8		451	451	1593,36	3,46	3505,67	451	0,01	0,00
9		514	632	2153,93	24,03	3501,74	627	0,07	21,98
10		384	394	3500,00	2,45	3502,50	393	0,02	2,34
11		257	282	3500,00	9,81	3502,60	282	0,03	9,73
12		503	504	367,67	0,00	1774,61	503	0,01	0,00
13		399	416	1823,73	0,90	3500,64	415	0,03	4,01
14		416	432	3082,00	5,67	3501,94	429	0,03	3,13
15		391	391	772,95	0,00	774,01	391	0,00	0,00
16		362	372	126,00	0,00	2344,11	372	0,02	2,76
17		435	488	3040,86	9,48	3502,11	484	0,02	11,26
18		267	-	-	-	-	-	-	-
19		332	332	560,67	0,50	3508,38	332	0,01	0,00
20		497	598	2278,94	21,65	3501,56	598	0,03	20,32

Tabela 3.5: Resultados computacionais com $D = 1.1\bar{C}_{max} + 1$

<i>Inst.</i>	<i>T</i>	<i>best</i> <i>lb</i>	Modelo de fluxos em rede revisto (3.36)–(3.45)				Modelo de afectação (3.46)–(3.51)		<i>final</i> <i>gap</i>
			<i>ub</i>	<i>t_{ub}</i>	<i>gap</i>	<i>t</i>	<i>ub</i>	<i>t</i>	
1	20	240	240	41,25	0,00	41,25	240	0,00	0,00
2		488	538	858,00	0,00	2629,48	535	0,09	9,63
3		383	407	826,00	0,00	2331,42	406	0,05	6,01
4		475	498	204,82	0,00	253,09	498	0,08	4,84
5		309	309	149,37	0,00	149,37	309	0,00	0,00
6		421	431	24,00	0,00	299,18	431	0,04	2,38
7		348	349	460,09	0,00	1024,76	349	0,01	0,29
8		451	451	287,50	0,00	524,09	451	0,03	0,00
9		514	565	2018,29	0,00	2831,39	561	0,10	9,14
10		384	403	2250,55	4,32	3301,79	403	0,03	4,95
11		257	268	3300,00	5,04	3301,39	268	0,03	4,28
12		503	516	496,10	0,00	2183,01	515	0,06	2,39
13		399	421	125,50	0,00	714,25	420	0,05	5,26
14		416	441	2117,27	7,26	3301,07	435	0,07	4,57
15		391	391	93,26	0,00	93,26	391	0,00	0,00
16		362	372	21,64	0,00	23,17	372	0,03	2,76
17		435	478	869,83	0,00	1032,35	478	0,04	9,89
18		267	267	2983,05	0,00	2983,05	267	0,00	0,00
19		332	332	147,57	0,00	2037,29	332	0,01	0,00
20		497	563	1275,33	0,00	3353,01	562	0,08	13,08
1	40	240	240	189,07	0,00	189,07	240	0,00	0,00
2		488	535	3289,82	11,62	3501,30	535	0,03	9,63
3		383	405	2485,00	6,34	3502,41	404	0,04	5,48
4		475	498	888,80	0,00	942,96	498	0,05	4,84
5		309	309	1611,05	0,00	1612,18	309	0,00	0,00
6		421	431	512,80	3,17	3503,66	431	0,01	2,38
7		348	349	1130,50	0,69	3501,30	349	0,00	0,29
8		451	451	381,18	3,12	3502,69	451	0,01	0,00
9		514	639	1838,57	24,62	3504,59	639	0,05	24,32
10		384	392	3065,47	1,86	3502,17	391	0,02	1,82
11		257	1194	3502,35	78,70	3502,35	1192	0,06	363,81
12		503	504	515,25	0,00	936,44	504	0,02	0,20
13		399	416	1736,50	8,45	3502,55	413	0,01	3,51
14		416	442	3432,80	8,14	3501,62	442	0,03	6,25
15		391	391	661,00	0,00	662,64	391	0,00	0,00
16		362	371	99,00	0,00	2830,68	371	0,02	2,49
17		435	483	2790,00	7,88	3502,44	483	0,03	11,03
18		267	-	-	-	-	-	-	-
19		332	332	315,00	0,50	3502,27	332	0,00	0,00
20		497	635	2559,93	24,45	3501,08	627	0,05	26,16

Tabela 3.6: Resultados computacionais com $D = 1.05\overline{C}_{max} + 1$

3.9 Conclusões

Neste capítulo é introduzido um modelo de afectação onde foram utilizadas estratégias de reforço com o objectivo de obter soluções de melhor qualidade. Foram utilizadas funções duais válidas, assim como outras desigualdades válidas. É descrito também um novo modelo de programação inteira pseudo-polinomial para o CSSP, a partir do qual foi derivada uma abordagem heurística para a sua resolução. Foi descrito um modelo revisto e mais compacto, acompanhado de um modelo de afectação no tempo que permite uma definição clara de um plano de afectação dos planos de corte. Foi realizado um estudo computacional em instâncias de referência da literatura para avaliar o impacto de alguns parâmetros sobre o desempenho da heurística correspondente. Os resultados mostram que a heurística pode gerar soluções de boa qualidade com pequenos intervalos de optimalidade para a maioria destas instâncias.

Capítulo 4

O problema integrado de planeamento e escalonamento ¹

Conteúdo

4.1	Introdução	68
4.2	O problema integrado de planeamento e escalonamento	72
4.3	Um novo modelo pseudo-polinomial de fluxos em rede	75
4.3.1	Definição	75
4.3.2	Exemplo	80
4.4	Estratégias de redução	84
4.5	Dependência entre o número de variáveis e a sequência de tarefas	85
4.6	Resultados computacionais	87
4.6.1	Análise comparativa de resultados	87
4.6.2	Actualização dos resultados usando um computador recente	91
4.7	Abordagens heurísticas alternativas	95
4.7.1	Estratégias heurísticas	101
4.7.2	Resultados computacionais	105

¹Os resultados deste capítulo foram publicados no artigo: J. Rietz, C. Alves, N. Braga, e J. Valério de Carvalho, An exact approach based on a new pseudo-polynomial network flow model for integrated planning and scheduling, Computers and Operations Research, 76:183–194, 2016 (Rietz et al. (2016)).

4.8	Conclusões	110
-----	----------------------	-----

4.1 Introdução

A importância da optimização integrada nas cadeias de abastecimento é reconhecida pela maioria dos actores do sector logístico e do transporte. Optimizar as operações ao longo de diferentes áreas funcionais das empresas contribui para a redução de custos e para a melhoria do desempenho. Por isso, muitos autores têm apoiado na literatura a ideia de uma coordenação mais ampla entre o planeamento e o escalonamento desde a aquisição das matérias primas ao planeamento da entrega (Grossmann e Furman, 2009). Do ponto de vista computacional, a resolução de problemas práticos de optimização integrada é um verdadeiro desafio. A maioria das abordagens desenvolvidas para este tipo de problemas foram propostas nos últimos anos (Armstrong et al., 2008, Chang e Lee, 2004, Geismar et al., 2008, Wang e Cheng, 2007).

Neste capítulo, é abordado o problema integrado de planeamento e escalonamento em máquinas paralelas e idênticas descrito em Kis e Kovács (2012). O problema é dividido em duas partes que são resolvidas simultaneamente de uma forma integrada. A primeira parte consiste em determinar as tarefas que serão processadas em cada período de tempo dentro de um determinado horizonte de planeamento. Esta é a parte de planeamento do problema. A segunda parte consiste em atribuir as tarefas às máquinas disponíveis em cada período de tempo, de acordo com as correspondentes datas de lançamento e de modo a que todas as tarefas sejam processadas até o final do período de tempo. Uma penalidade aplica-se sempre que uma tarefa é efectuada antes ou depois da sua data de entrega. O objectivo do problema é determinar o plano e afectações de tarefas às máquinas que minimizam os custos totais relacionados com essas penalidades.

Os problemas de planeamento e escalonamento podem abranger decisões estratégicas de longo prazo e/ou decisões táticas de médio prazo. A título de exemplo, podem envolver decisões como a construção de instalações que são depois usadas para produzir uma carteira de produtos adaptadas aos pedidos e exigências dos clientes. Como descrito em Maravelias e Sung (2009), essas decisões de longo e médio prazo estão

fortemente interligadas, e devem por isso ser tomadas de uma forma integrada. Na prática, a resolução dos problemas de optimização integrada é um verdadeiro desafio. Conforme descrito em Shobrys e White (2000), essa integração enfrenta dificuldades que não são exclusivas do domínio da gestão da cadeia de abastecimento.

Uma vez que a definição de um plano detalhado para um horizonte de tempo muito longo não é normalmente possível, a abordagem à resolução destes problemas passa muitas vezes por resolver o problema para um horizonte de tempo mais curto, e em seguida actualizar o respetivo plano considerando a informação dos períodos seguintes. Assim, em vez de se definir um plano para vários meses, pode definir-se um horizonte de tempo de algumas semanas e, depois da semana terminada, deslocar-se o mesmo para a semana seguinte. Esta abordagem foi utilizada em Persson e Göthe-Lundgren (2005) para a optimização de rotas de navios, onde os mesmos transportam betume e outros derivados de petróleo a partir de um conjunto de refinarias para depósitos onde são armazenados.

Em Erdirik-Dogan e Grossmann (2008), os autores exploram um problema de produção química definida ao longo de vários períodos de tempo. O problema tem custos de transição e atrasos relacionados com a mudança de um produto para outro. Os autores resolveram o problema de planeamento e escalonamento correspondente por decomposição do mesmo num problema de planeamento a alto nível e num problema de escalonamento a baixo nível. Uma revisão sistemática dos problemas de escalonamento na indústria química é feita em Baldea e Harjunkoski (2014).

O problema integrado de planeamento e escalonamento abordado neste trabalho foi discutido pela primeira vez em Kis e Kovács (2012). De um modo geral, o horizonte de planeamento é dividido em períodos de tempo. Para cada período de tempo, há um conjunto de máquinas idênticas com um tempo de trabalho limitado. Além disso, há um conjunto de tarefas com datas de lançamento, datas de entrega, tempos de processamento e penalidades para cumprir uma tarefa anterior ou posterior à sua data de entrega. Cada tarefa tem de ser atribuída a um período de tempo e uma máquina, de tal modo a que a soma das penalidades seja minimizada. Uma vez que nenhuma das tarefas pode ser dividida, nem de um período de tempo para outro, nem entre máquinas, o problema contém um problema de empacotamento a uma dimensão

(bin-packing problem) como subproblema.

Em Kis e Kovács (2012), os autores descreveram duas abordagens de decomposição e uma formulação matemática monolítica para este problema. As abordagens baseadas em decomposição contam com um modelo de programação inteira que representa a parte de planeamento do problema. Para garantir a validade do escalonamento, os autores recorrem a planos de corte, enquanto o método de partição e avaliação com cortes (*branch-and-cut*) é usado para encontrar soluções inteiras. Apesar dos resultados promissores obtidos por estes autores, as abordagens encontram algumas dificuldades na resolução deste problema.

Como mencionado acima, o problema integrado de planeamento e escalonamento combina problemas \mathcal{NP} -difíceis de optimização combinatória: o problema do escalonamento em máquinas paralelas e o problema de empacotamento. Os problemas de escalonamento têm sido amplamente estudados no quadro da investigação operacional e das ciências da computação ao longo dos anos. Por exemplo, Baldea e Harjunkoski (2014) abordaram o problema de escalonamento de projetos com recursos limitados, que consiste em escalonar um conjunto de actividades que requerem quantidades variáveis de um determinado recurso. As actividades têm datas de lançamento e tempos de processamento, sendo que o objectivo é minimizar o material utilizado. Embora cada actividade requeira uma quantidade constante do recurso, este montante pode ser diferente entre as actividades. Além disso, pode haver precedências entre as actividades e as datas de entrega devem ser cumpridas exactamente (a actividade não pode terminar após a sua data de entrega). Em contraste, não há precedências no problema considerado neste trabalho. As tarefas podem ser terminadas com atraso, incorrendo-se nesse caso num custo que é proporcional a esse atraso. Da mesma forma, uma tarefa pode ser concluída antes da sua data de entrega, mas também com uma dada penalidade.

Mais recentemente, Su et al. (2013) resolveram um problema de escalonamento em máquinas com diferentes velocidades e disponibilidade limitada, e onde o objectivo consiste em minimizar tanto o material utilizado como o atraso máximo. Os autores formulam o problema como um problema de fluxo de custo mínimo numa rede. Detienne (2014) resolveu diversas variantes do problema de escalonamento em máquinas

que podem não estar disponíveis em determinados períodos de tempo. O autor considerou diferentes casos que diferem em função da forma segundo a qual uma tarefa interrompida por ser retomada. Algumas tarefas precisam de um tempo de *setup* após terem sido interrompidas. Outras devem ser reiniciadas desde o início. Sadykov e Wolsey (2006) exploram um problema de escalonamento em que ambos os tempos de processamento e os custos das tarefas dependem do equipamento que é usado. Além disso, todas as tarefas têm de ser concluídas entre a sua data de lançamento e de entrega. Os autores propõem várias abordagens algorítmicas com base em programação inteira, programação por restrições e geração de colunas. Yang et al. (2012) usa um algoritmo genético melhorado para resolver um problema de escalonamento de tarefas com datas de lançamento, datas de entrega, prazos e custos associados ao atraso ou antecipação das tarefas.

O problema de empacotamento é um problema $\mathcal{NP} - \text{difícil}$ (Garey e Johnson, 1978). Uma revisão de algumas heurísticas simples e do seu desempenho no pior caso pode ser consultada em Hochbaum (1997). Limites inferiores para estes problemas podem ser calculados de forma eficiente, por exemplo, com funções duais válidas, como demonstrado em Clautiaux et al. (2010). Métodos exactos, incluindo algoritmos de partição e corte com geração de colunas (*branch-and-cut-and-price*) foram também descritos em Belov (2003), Belov e Scheithauer (2006a), Alves e Valério de Carvalho (2008). Resultados teóricos importantes sobre a qualidade dos modelos de programação inteira para o problema de corte e empacotamento a uma dimensão têm também sido propostos na literatura.

Neste trabalho, o objectivo é de contribuir para a resolução eficiente de um problema particular de planeamento e escalonamento descrito na literatura por Kis e Kovács (2012). É proposto e analisado um novo modelo de programação inteira para esse problema, e comparados os resultados obtidos com esse modelo com os descritos por Kis e Kovács (2012).

Na secção 4.2, é descrito formalmente o problema e introduzida a notação utilizada durante todo o capítulo para este problema. O Exemplo 2 apresenta o conteúdo de uma instância do problema. A contribuição para a resolução deste problema é apresentada na secção 4.3 onde é definido o modelo de programação inteira (4.4)–

(4.8) de fluxos em rede. No Exemplo 3 são demonstrados sumariamente os fluxos gerados pelo grafo da instância do Exemplo 2. Na secção 4.3.2 estão representadas as restrições do problema através das variáveis, segundo a notação apresentada no modelo (4.4)–(4.8), com objectivo de auxiliar na sua compreensão. As estratégias de redução de arcos são apresentada na secção 4.4. Os resultados computacionais podem ser consultados na secção 4.6 que realizam uma comparação justa com os de Kis e Kovács (2012). Os testes efectuados utilizando um computador recente são descritos na secção 4.6.2. As abordagens heurísticas (secções 4.7 e 4.7.1) são descritas com o objectivo de resolver as instâncias de difícil resolução em que o modelo (4.4)–(4.8) teve dificuldade de se aproximar do óptimo. Os resultados computacionais com estas heurísticas são descritos na secção 4.7.2. Por fim, é apresentada uma breve conclusão na secção 4.8.

4.2 O problema integrado de planeamento e escalonamento

No problema integrado de planeamento e escalonamento abordado neste trabalho, é dado um conjunto de M máquinas idênticas e um conjunto de N tarefas. O planeamento e escalonamento é efectuado ao longo de um horizonte de planeamento que é dividido em τ períodos de tempo de igual dimensão $P \in \mathbb{N}$. Seja $T = \{1, \dots, \tau\}$. As tarefas têm tempos de processamento $p_j \in \mathbb{N} \setminus \{0\}$, datas de lançamento $r_j \in T$, prazos de entrega $d_j \geq r_j$ com $d_j \in T$ e penalidades $e_j, \ell_j \in \mathbb{N}$ por terminar a tarefa $j \in N$ mais cedo ou mais tarde que o prazo de entrega, respectivamente.

O objectivo do problema consiste em determinar o período de tempo $t \in T$ em que cada tarefa será efectuada, e, em seguida, atribuir as tarefas às máquinas, de modo a que a soma dos tempos de processamento das tarefas atribuídas a uma dada máquina não exceda o tempo fixo P . É assumido ainda que:

$$P \geq \max \{p_j : j \in N\}.$$

Além disso, cada tarefa deve ser concluída sem interrupção na mesma máquina e

no mesmo período de tempo em que foi iniciada. A penalidade que se aplica a uma tarefa $j \in N$ feita no período de tempo $t \in T$ é definido como segue:

$$w_t^j := e_j \times \max\{0, d_j - t\} + \ell_j \times \max\{0, t - d_j\}. \quad (4.1)$$

O objectivo do problema consiste em determinar a afectação das tarefas aos períodos de tempo e o respetivo escalonamento nas máquinas que minimiza a soma de todas das penalidades.

Uma instância para o problema estudado neste capítulo possui os seguintes dados de entrada, que são representados por $(|M|, |N|, P, \tau)$ e por $|N|$ conjuntos $(r_j, d_j, p_j, e_j, \ell_j)$ para as tarefas $j \in N$. O primeiro conjunto de dados fornece informação relativa à limitação dos recursos e o segundo informação sobre as tarefas a efectuar nesses recursos limitados. Além disso, assume-se que:

- $|M|, |N|, P, \tau > 0$,
- $N = \{1, \dots, |N|\}$,
- $1 \leq r_j \leq d_j \leq \tau$, para todo $j \in N$, e
- $p_j \leq P$, para todo $j \in N$.

Exemplo 2. *De acordo com a notação introduzida acima, os seguintes dados definem uma instância do problema integrado de planeamento e escalonamento:*

$(M $	$ N $	P	$\tau)$					
2	10	18	3					
				$(r_j$	d_j	p_j	e_j	$\ell_j)$
	1	1	12	0	60			
	1	1	9	0	45			
	1	1	7	0	35			
	1	1	4	0	20			
	1	1	3	0	15			
	1	2	10	3	50			
	2	2	9	0	45			
	2	2	6	0	30			
	2	2	6	0	30			
	2	2	4	0	20			

A instância é composta por duas máquinas e dez tarefas. Todas as tarefas, exceto a sexta, devem ser concluídas no mesmo período em que são disponibilizadas. Assim, a penalidade e_j não tem impacto nestas tarefas. Pelo contrário, se a sexta tarefa for efectuada no primeiro período, a penalidade terá um valor de 3 unidades. Para esta instância, se uma tarefa for executada mais tarde do que a sua correspondente data de entrega, a penalidade em que se incorre é muito mais elevada. De notar que todas as tarefas, à excepção da sexta, não podem ser realizadas antes da sua data de entrega, uma vez que a data de lançamento e a data de entrega coincidem. Nesta instância observa-se que as penalidades associadas às entregas posteriores à data de entrega (ℓ_j) são bastante elevadas comparativamente à penalidade e_j (entrega antecipada). Isto representa uma maior valorização do facto das tarefas serem efectuadas dentro dos prazos.

A Figura 4.1 ilustra a instância do Exemplo 2 dos recursos disponíveis. Neste caso, existem duas máquinas paralelas (M1 e M2). O tempo total de processamento das tarefas atribuídas a cada uma destas máquinas em cada período não pode exceder 18 unidades (P). No máximo, estão disponíveis $\tau = 3$ períodos de tempo em cada

máquina para realizar as 10 tarefas consideradas na instância do Exemplo 2.

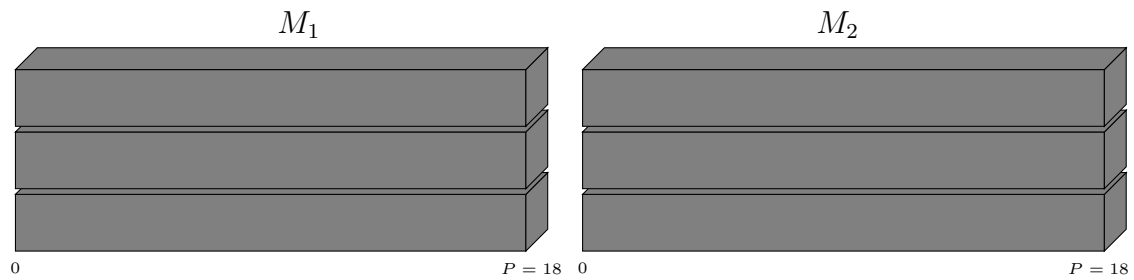


Figura 4.1: Instância do Exemplo 2.

Afectar tarefas a máquinas num determinado período de tempo $t \in T$ é equivalente a resolver um problema de corte a uma dimensão, e verificar se a respetiva função objectivo não excede o número $|M|$ de máquinas disponíveis. Assim, os modelos propostos na literatura para o problema de corte podem ser utilizados para avaliar se um subconjunto de tarefas atribuídas a um determinado período de tempo t podem efectivamente ser executadas nas máquinas disponíveis naquele período. Seja

$$N_t := \{j \in N : r_j \leq t\}$$

o conjunto das tarefas que podem ser executadas no período de tempo t . Obviamente, apenas as tarefas cuja data de lançamento é anterior (ou simultânea) ao período de tempo t poderão ser agendadas para esse período de tempo.

4.3 Um novo modelo pseudo-polinomial de fluxos em rede

4.3.1 Definição

Nesta secção, é apresentado um novo modelo de fluxos em rede para o problema integrado de planeamento e escalonamento descrito acima. O modelo é definido num grafo que é composto por um conjunto de vértices $V := \{0, \dots, P\}$ que representam instantes de tempo associados às máquinas. O instante de tempo inicial corresponde

ao vértice 0 do grafo, enquanto o último instante de tempo é representado pelo vértice P , que é também o tempo máximo de processamento disponível numa máquina no período de tempo t ($t \in T$). O conjunto de todos os arcos possíveis é representado por E . Para cada período de tempo $t \in T$, é definido um subconjunto de arcos designados por E_t , tal que:

$$E = E_0 \bigcup_{t \in T} E_t$$

O conjunto de arcos E_0 refere-se ao tempo não utilizado numa máquina,

A cada arco $(a, a + p_j) \in E_t$ está associada uma variável binária x_{aj}^t no modelo de fluxos. Essa variável toma o valor 1 se a tarefa j é realizada no período de tempo $t \in T$ com início no instante de tempo a , e 0 caso contrário. Para representar uma máquina que não é usada durante todo um período de tempo, recorre-se a arcos de desperdício que vão desde um vértice $a \in V$ até ao vértice final P do grafo. As variáveis associadas a estes arcos são variáveis inteiras (não necessariamente binárias) que são designadas por $x_{a0}^t \in \mathbb{N}$, com $a \in \{0, 1, \dots, P - 1\}$. Se x_{a0}^t for superior a 0, então pelo menos uma máquina não é utilizada no período t entre os instantes de tempo a e P desse período.

De notar que, no modelo proposto neste capítulo assume-se que as tarefas estão numeradas de 1 a $|N|$. O índice 0 é usado para representar instantes de tempo em que uma ou mais máquinas não são efectivamente usadas.

Um arco $(a, b) \in E_t$ corresponde a uma tarefa $j \in N_t$ com $b - a = p_j$. Dado que poderão existir tarefas $i, j \in N$ com $i \neq j$ com o mesmo tempo de processamento, no modelo proposto neste capítulo optou-se por associar as tarefas aos arcos do grafo, em vez dos vértices do grafo representarem o início e fim do processamento de uma tarefa, como acontece por exemplo nos modelos analisados no capítulo anterior. Consequentemente, poderão existir no grafo vários arcos entre os mesmos pares de vértices, referentes a diferentes tarefas $i, j \in N$, mas com a mesma duração de tempo de processamento ($p_i = p_j$). A única exceção são os arcos de desperdício, que são únicos.

A relação entre o problema integrado de planeamento e escalonamento e o pro-

blema de corte a uma dimensão foi já referida acima. Nos problemas de corte, é usual considerarem-se apenas padrões de corte que são maximais, *i.e.* padrões em que o desperdício é menor que o tamanho do item mais pequeno a cortar. Essa abordagem é válida em casos em que é permitido o excesso de produção. No problema integrado de planeamento e escalonamento, produzir em excesso significa que a mesma tarefa é efectuada mais de uma vez, o que por definição não é permitido. Por outro lado, efectuar mais de uma vez a mesma tarefa conduziria naturalmente a um aumento das penalidades (4.1). Consequentemente, no modelo apresentado a seguir, são também considerados padrões que sejam não maximais. Para isso, recorre-se ao subconjunto de arcos E_0 que já foi apresentado acima, e que é definido da seguinte forma:

$$E_0 := \{(a, P) : a \in \{0, \dots, P - 1\}\}.$$

De notar que um padrão poderá ser associado a diferentes caminhos no grafo. Uma solução para o problema poderá assim ter várias representações no modelo de fluxos, induzindo simetria no espaço de soluções. Para reduzir essa simetria, sem com isso afectar a solução óptima do problema, são impostas restrições adicionais à forma como são definidos os arcos do grafo. Essas restrições definem uma certa ordem entre as tarefas executadas nas máquinas, e permitem também diminuir o número de arcos nos conjuntos E_t , $t \in T$. Considere-se os arcos $(a, b), (b, c) \in E_t$ tal que $a < b < c$. Impõe-se que, a tarefa associada ao arco (b, c) tenha um índice no conjunto N que é superior ao menor índice das tarefas cujos respectivos arcos terminam no vértice b . A definição formal dessa regra é apresentada a seguir.

O conjunto de vértices a serem considerados em cada período de tempo $t \in T$ é o seguinte:

$$V_t := \left\{ v \in V : v = \sum_{j \in N_t} p_j \times a_j, a_j \in \{0, 1\} \right\}, \quad (4.2)$$

O vértice v pertence a V_t se existir uma combinação válida de tarefas $j \in N_t$ cujo tempo total de processamento seja igual a v . De acordo com Scheithauer (2008), é possível garantir a unicidade do caminho associado a um determinado padrão rede-

finindo os conjuntos de arcos E_t como segue. Seja $\mu(v)$ o menor índice das tarefas cujos arcos terminam no vértice v , *i.e.*

$$\mu(v) := \begin{cases} 0, & v = 0, \\ \min \{j \in N_t : v - p_j \in V_t, j > \mu(v - p_j)\}, & v \in V_t \setminus \{0\}. \end{cases} \quad (4.3)$$

Usando essa definição, pode redefinir-se o conjunto de arcos E_t da forma seguinte para induzir a ordem pretendida entre as tarefas que são processadas nas máquinas num determinado período de tempo t :

$$E_t := \{(a, b) \in V_t : \exists j \in N_t \text{ com } b - a = p_j \text{ e } j > \mu(a)\}.$$

Exemplo 3. *Considere-se a instância descrita no Exemplo 2. Têm-se que $V = \{0, 1, \dots, 18\}$, $N_1 = \{1, \dots, 6\}$ e $N_2 = N_3 = \{1, \dots, 10\}$. De acordo com a definição (4.2), o conjunto de vértices do grafo é tal que: $V_1 = \{0, 3, 4, 7, 9, 10, \dots, 17\}$ e $V_2 = V_3 = \{0, 3, 4, 6, 7, \dots, 18\}$. A título de exemplo, note-se que nenhuma combinação de tarefas resultará num tempo total de processamento igual a 5. A definição (4.3) resulta nos seguintes valores para $\mu(v)$:*

v	3	4	6	7	8	9	10	11	12	13	14	15	16	17	18
$\mu(v)$ (para $t = 1$)	5	4	-	3	-	2	5	4	1	4	5	5	3	6	-
$\mu(v)$ (para $t > 1$)	5	4	8	3	10	2	5	4	1	4	5	5	3	6	7

Tabela 4.1: Exemplo de valores de $\mu(v)$ para $t = 1$ e $t > 1$.

Para $t = 1$, o caminho desde o vértice 0 até ao vértice 3 ou 4 é único. Por outro lado, existem dois caminhos até ao vértice 7: $0 \rightarrow 7$ (associado à execução da terceira tarefa) e $0 \rightarrow 4 \rightarrow 7$ (tarefa 4 seguida pela tarefa 5). O caminho $0 \rightarrow 3 \rightarrow 7$ é eliminado do grafo, uma vez que o arco $3 \rightarrow 7$ associado à tarefa 4 não pertence ao conjunto de arcos válidos porque $4 \leq \mu(3)$.

O grafo (V, E_1) é apresentado na Figura 4.2. Os 19 arcos de E_1 são: $(0, 3)$, $(0, 4)$, $(0, 7)$, $(0, 9)$, $(0, 10)$, $(0, 12)$, $(3, 13)$, $(4, 7)$, $(4, 14)$, $(7, 10)$, $(7, 11)$, $(7, 17)$, $(9, 12)$, $(9, 13)$, $(9, 16)$, $(11, 14)$, $(12, 15)$, $(12, 16)$ e $(13, 16)$.

Os conjuntos E_2 e E_3 contêm, adicionalmente, os seguintes 33 arcos, incluindo alguns arcos entre o mesmo par de vértices:

(0, 9), (0, 6), (0, 6), (0, 4), (3, 12), (3, 9), (3, 9), (3, 7), (4, 13), (4, 10), (4, 10), (4, 8), (6, 12), (6, 10), (7, 16), (7, 13), (7, 13), (7, 11), (9, 18), (9, 15), (9, 15), (9, 13), (10, 16), (10, 16), (10, 14), (11, 17), (11, 17), (11, 15), (12, 18), (12, 18), (12, 16), (13, 17) e (14, 18).

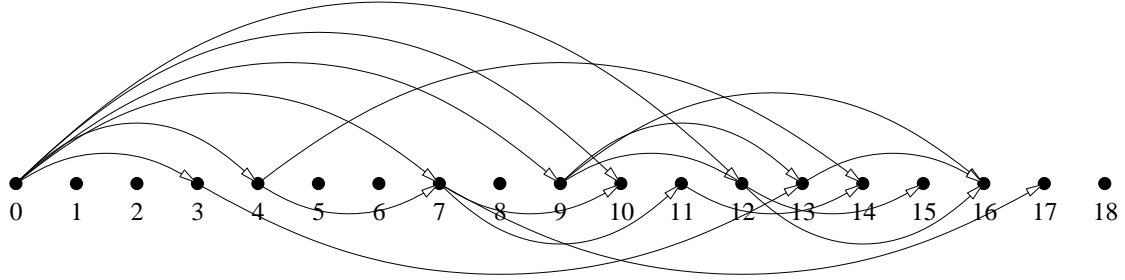


Figura 4.2: Grafo (V, E_1) para a instância Exemplo 2 (sem arcos de desperdício)

A repetição de tarefas não é possível, uma vez que implica penalidades adicionais. Assim, é necessário adicionar arcos de desperdício ao grafo. No exemplo da Figura 4.2, existem arcos de desperdício E_0 que não estão representados na figura.

Um caminho entre os vértices 0 e P do grafo corresponde a uma afectação válida de tarefas a uma máquina no período $t \in T$. Por sua vez, um fluxo no grafo (V, E) corresponde ao número de vezes que uma afectação é usada. Os valores associados às variáveis x_{aj}^t representam numericamente o valor desses fluxos. O modelo de fluxos em rede é definido formalmente como segue:

$$\min \quad w := \sum_{j \in N} \sum_{t \in T} w_t^j \times \left(\sum_{(a, a+p_j) \in E_t} x_{aj}^t \right) \quad (4.4)$$

$$\text{sujeito a} \quad \sum_{(a-p_j, a) \in E_t} x_{a-p_j, j}^t - \sum_{(a, a+p_i) \in E_t} x_{ai}^t - x_{a0}^t = 0, \quad \forall a \in V_t \setminus \{0, P\}, t \in T, \quad (4.5)$$

$$\sum_{t \in T} \sum_{(a, a+p_j) \in E_t} x_{aj}^t = 1, \quad \forall j \in N, \quad (4.6)$$

$$x_{00}^t + \sum_{(0, p_j) \in E_t} x_{0j}^t = |M|, \quad \forall t \in T, \quad (4.7)$$

$$x_{aj}^t \in \{0, 1\} \text{ e } x_{a0}^t \in \mathbb{N}, \quad a \in \{0, \dots, P-1\}, \quad t \in T, \quad j \in N_t. \quad (4.8)$$

A função objectivo (4.4) do modelo corresponde à minimização do total das penalidades associadas à execução das tarefas antes ou depois das suas datas de entrega. As restrições (4.5) garantem a conservação de fluxo ao longo do grafo. As restrições (4.6) garantem que todas as tarefas são executadas uma e uma única vez, enquanto as restrições (4.7) limitam a $|M|$ o número de máquinas que são usadas em cada período de tempo.

O modelo (4.4)–(4.8) tem um número pseudo-polinomial de restrições e variáveis. Existem $|N| + \tau$ restrições (4.6) e (4.7), enquanto que o número de restrições de conservação de fluxo (4.5) depende dos conjuntos V_t , $t \in T$. Claramente, a dimensão do conjunto $V = \{0, \dots, P\}$ é pseudo-polinomial em ordem aos dados do problema. O mesmo acontece com V_t , tendo em conta a sua definição (4.2). A complexidade associada ao cálculo de V_t é inferior a $\mathcal{O}(|N| \times P)$ e, por isso, derivar as restrições de conservação de fluxo (4.5) requer um esforço computacional que é também ele pseudo-polinomial. No entanto, apesar do tamanho do modelo ser pseudo-polinomial, é importante notar que o esforço necessário para o resolver não é necessariamente pseudo-polinomial.

4.3.2 Exemplo

Considere o problema integrado de planeamento e escalonamento definido no Exemplo 2. Para esta instância, existem $19 + 2 \times 52 = 123$ variáveis binárias x_{aj}^t . As restrições (4.7) podem ser representadas como se segue:

$$\begin{aligned} x_{00}^1 + x_{01}^1 + \dots + x_{06}^1 &= 2 \\ x_{00}^2 + x_{01}^2 + \dots + x_{0,10}^2 &= 2 \\ x_{00}^3 + x_{01}^3 + \dots + x_{0,10}^3 &= 2 \end{aligned}$$

Para clarificar a apresentação das restrições de conservação de fluxos (4.5), as restrições são ordenadas por ordem crescente do vértice. Depois de eliminar restrições

redundantes associadas aos vértices $a > 15$, obtêm-se as seguintes restrições para $t = 1$ e $t = 2$:

$$\begin{aligned}
x_{05}^1 - x_{30}^1 - x_{36}^1 &= 0 \\
x_{04}^1 - x_{40}^1 - x_{45}^1 - x_{46}^1 &= 0 \\
x_{03}^1 + x_{45}^1 - x_{70}^1 - x_{74}^1 - x_{75}^1 - x_{76}^1 &= 0 \\
x_{02}^1 - x_{90}^1 - x_{93}^1 - x_{94}^1 - x_{95}^1 &= 0 \\
x_{06}^1 + x_{75}^1 - x_{10,0}^1 &= 0 \\
x_{74}^1 - x_{11,0}^1 - x_{11,5}^1 &= 0 \\
x_{01}^1 + x_{95}^1 - x_{12,0}^1 - x_{12,4}^1 - x_{12,5}^1 &= 0 \\
x_{94}^1 + x_{36}^1 - x_{13,0}^1 - x_{13,5}^1 &= 0 \\
x_{11,5}^1 + x_{46}^1 - x_{14,0}^1 &= 0 \\
x_{12,5}^1 - x_{15,0}^1 &= 0
\end{aligned}$$

e

$$\begin{aligned}
x_{05}^2 - x_{30}^2 - x_{36}^2 - x_{37}^2 - \cdots - x_{3,10}^2 &= 0 \\
x_{04}^2 + x_{0,10}^2 - x_{40}^2 - x_{45}^2 - x_{46}^2 - \cdots - x_{4,10}^2 &= 0 \\
x_{08}^2 + x_{09}^2 - x_{60}^2 - x_{69}^2 - x_{6,10}^2 &= 0 \\
x_{03}^2 + x_{45}^2 + x_{3,10}^2 - x_{70}^2 - x_{74}^2 - x_{75}^2 - \cdots - x_{7,10}^2 &= 0 \\
x_{4,10}^2 - x_{80}^2 &= 0 \\
x_{02}^2 + x_{38}^2 + x_{39}^2 + x_{07}^2 - x_{90}^2 - x_{93}^2 - x_{94}^2 - x_{95}^2 - x_{97}^2 - x_{98}^2 - x_{99}^2 - x_{9,10}^2 &= 0 \\
x_{06}^2 + x_{75}^2 + x_{48}^2 + x_{49}^2 + x_{6,10}^2 - x_{10,0}^2 - x_{10,8}^2 - x_{10,9}^2 - x_{10,10}^2 &= 0 \\
x_{74}^2 + x_{7,10}^2 - x_{11,0}^2 - x_{11,5}^2 - x_{11,8}^2 - x_{11,9}^2 - x_{11,10}^2 &= 0 \\
x_{01}^2 + x_{95}^2 + x_{37}^2 + x_{69}^2 - x_{12,0}^2 - x_{12,4}^2 - x_{12,5}^2 - x_{12,8}^2 - x_{12,9}^2 - x_{12,10}^2 &= 0 \\
x_{94}^2 + x_{36}^2 + x_{47}^2 + x_{78}^2 + x_{79}^2 + x_{9,10}^2 - x_{13,0}^2 - x_{13,5}^2 - x_{13,10}^2 &= 0 \\
x_{46}^2 + x_{11,5}^2 + x_{10,10}^2 - x_{14,0}^2 - x_{14,10}^2 &= 0
\end{aligned}$$

$$x_{98}^2 + x_{99}^2 + x_{11,10}^2 + x_{12,5}^2 - x_{15,0}^2 = 0$$

Para $t = 3$, as restrições são semelhantes ao caso $t = 2$. As restrições (4.6) são definidas como segue:

$$\begin{aligned} \sum_{t=1}^3 x_{01}^t &= 1 \\ \sum_{t=1}^3 x_{02}^t &= 1 \\ \sum_{t=1}^3 (x_{03}^t + x_{93}^t) &= 1 \\ \sum_{t=1}^3 (x_{04}^t + x_{12,4}^t + x_{94}^t + x_{74}^t) &= 1 \\ \sum_{t=1}^3 (x_{05}^t + x_{45}^t + x_{75}^t + x_{95}^t + \sum_{a=11}^{13} x_{a5}^t) &= 1 \\ \sum_{t=1}^3 (x_{06}^t + x_{76}^t + x_{46}^t + x_{36}^t) &= 1 \\ \sum_{t=2}^3 (x_{07}^t + x_{97}^t + x_{77}^t + x_{47}^t + x_{37}^t) &= 1 \\ \sum_{t=2}^3 (x_{08}^t + x_{38}^t + x_{48}^t + x_{78}^t + \sum_{a=9}^{12} x_{a8}^t) &= 1 \\ \sum_{t=2}^3 (x_{09}^t + x_{39}^t + x_{49}^t + x_{69}^t + x_{79}^t + \sum_{a=9}^{12} x_{a9}^t) &= 1 \\ \sum_{t=2}^3 (x_{0,10}^t + x_{3,10}^t + x_{4,10}^t + x_{6,10}^t + x_{7,10}^t + \sum_{a=9}^{14} x_{a,10}^t) &= 1 \end{aligned}$$

A função objectivo (4.4) corresponde a:

$$\begin{aligned} &3 \times (x_{06}^1 + x_{76}^1 + x_{46}^1 + x_{36}^1) + 60x_{01}^2 + 120x_{01}^3 + 45x_{02}^2 + 90x_{02}^3 + 35x_{03}^2 + \\ &70x_{03}^3 + 35x_{93}^2 + 70x_{93}^3 + \\ &20 \times (x_{04}^2 + x_{12,4}^2 + x_{94}^2 + x_{74}^2) + 40 \times (x_{04}^3 + x_{12,4}^3 + x_{94}^3 + x_{74}^3) + \end{aligned}$$

$$\begin{aligned}
& 15 \times (x_{05}^2 + x_{45}^2 + x_{75}^2 + x_{95}^2 + x_{11,5}^2 + x_{12,5}^2 + x_{13,5}^2) + \\
& 30 \times (x_{05}^3 + x_{45}^3 + x_{75}^3 + x_{95}^3 + x_{11,5}^3 + x_{12,5}^3 + x_{13,5}^3) + \\
& 50 \times (x_{06}^3 + x_{76}^3 + x_{46}^3 + x_{36}^3) + 45 \times (x_{07}^3 + x_{97}^3 + x_{77}^3 + x_{47}^3 + x_{37}^3) + \\
& 30 \times (x_{08}^3 + x_{38}^3 + x_{48}^3 + x_{78}^3 + \sum_{a=9}^{12} (x_{a8}^3 + x_{a9}^3) + x_{09}^3 + x_{39}^3 + x_{49}^3 + x_{69}^3 + x_{79}^3) + \\
& 20 \times (x_{0,10}^3 + x_{3,10}^3 + x_{4,10}^3 + x_{6,10}^3 + x_{7,10}^3 + \sum_{a=9}^{14} x_{a,10}^3).
\end{aligned}$$

O valor da solução óptima associada a esta instância é igual a 35, e corresponde a:

$$x_{01}^1 = x_{02}^1 = x_{93}^1 = x_{12,4}^1 = x_{06}^2 = x_{07}^2 = x_{95}^2 = x_{10,8}^2 = x_{12,9}^2 = x_{0,10}^3 = 1.$$

Todas as outras variáveis binárias são iguais a zero. A primeira e segunda tarefa são executadas em máquinas diferentes a partir do instante 0 do primeiro período. A terceira e quarta tarefa são executadas no mesmo período, enquanto que todas as outras tarefas, exceto a última, são executadas no segundo período. A décima tarefa é executada no terceiro período de tempo. A máquina em que cada tarefa é realizada não é explicitamente definida. No entanto, através da variável x_{95}^2 (cujo valor é 1 na solução óptima), é possível determinar que a tarefa 5 é iniciada logo após a tarefa 7 na mesma máquina devido à duração de cada uma destas tarefas. Com base na solução do modelo (4.4)–(4.8), é fácil determinar um plano completo de afectação das tarefas às máquinas.

A solução do Exemplo 4.3.2 pode ser representada através da Figura 4.3 onde estão agora atribuídas tarefas às máquinas. Ao contrário da Figura 4.1, na Figura 4.3 são apresentadas as tarefas atribuídas às máquinas por unidade de tempo e sua sequência. Na figura, estão representadas todas as tarefas. Cada máquina (M_1 e M_2) possui um conjunto de tarefas atribuídas segundo as variáveis do modelo (4.4)–(4.8) escalonadas a cada período de tempo.

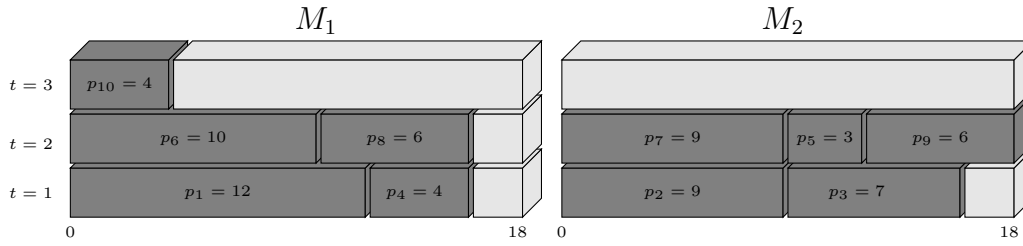


Figura 4.3: Solução representada graficamente para a instância do Exemplo 4.3.2.

4.4 Estratégias de redução

Em Rietz et al. (2016), foram propostas diferentes estratégias para simplificar o modelo de fluxos em rede (4.4)–(4.8) usando o conceito de *grid point*. O objectivo é eliminar vértices do grafo que não sejam necessários, e reduzir assim o número de variáveis do modelo. Os *grid points* foram propostos inicialmente para o problema de saco de mochila. Uma descrição completa do conceito de *grid point* pode ser encontrada em Scheithauer (2008). Considere-se uma instância do problema de saco de mochila com uma capacidade L , e com um conjunto I de itens de comprimento $\ell_i > 0$, $i \in I$, disponíveis em número ilimitado. O conjunto S dos *grid points* é composto pelas combinações lineares não negativas dos comprimentos dos itens, de modo a que a capacidade L da mochila não seja excedida. Formalmente, tem-se

$$S := \{s \leq L \mid s = \sum_{i \in I} \ell_i \times x_i, x_i \in \mathbb{N}\}.$$

Nos casos em que itens de comprimento ℓ_i não podem ser usados mais de $b_i \in \mathbb{N}$ vezes, o conjunto S de *grid points* passa a ser definido como segue:

$$S := \{s \leq L \mid s = \sum_{i \in I} \ell_i \times x_i, x_i \in \mathbb{N}, x_i \leq b_i\}. \quad (4.9)$$

No modelo de fluxos em rede (4.4)–(4.8), os conjuntos V_t , $t \in T$, definidos em (4.2), são também eles conjuntos de *grid points*.

Em Rietz et al. (2016), foram propostas duas estratégias que permitem reduzir o número de vértices do grafo e, conseqüentemente, o número de restrições do modelo

(4.4)–(4.8). Essas estratégias assentam essencialmente na correspondência que existe entre o conjunto de *grid points* V_t no intervalo $[0, P]$, $t \in T$, e os *grid points* associados ao problema de mochila definidos através de (4.9). Desse modo, potenciais vértices do grafo que não correspondam a *grid points* do conjunto S poderão ser eliminados dos conjuntos V_t . Neste capítulo, essas estratégias serão designadas por *Restrição I e II* ao conjunto de *grid points*, respetivamente. Para que a análise do modelo seja completa, são apresentados mais abaixo os resultados computacionais associados ao modelo de fluxos com e sem o recurso a essas estratégias de simplificação.

4.5 Dependência entre o número de variáveis e a sequência de tarefas

A ordenação das tarefas de uma instância do problema integrado de planeamento e escalonamento pelo seu tempo de processamento pode ter um forte impacto no número de arcos no grafo, e conseqüentemente, no número de variáveis do modelo (4.4)–(4.8). Para ilustrar esse efeito, a instância do Exemplo 2 será novamente utilizada.

Para cada tarefa $j \in N_t$, existe um arco $(0, p_j)$. Além disso, é possível mostrar que os outros arcos do modelo dependem da ordem das tarefas na instância. Se as tarefas forem ordenadas por ordem decrescente do seu tempo de processamento, obtém-se então os seguintes arcos adicionais no primeiro período:

$$(4, 7), (7, 11), (7, 10), (9, 16), (9, 13), (9, 12), (10, 17),$$

$$(10, 14), (10, 13), (11, 14), (12, 16), (12, 15), (13, 16), (14, 17).$$

No total, existem 20 arcos. No segundo e terceiro períodos de tempo, têm-se 10 arcos $(0, p_j)$ com $j \in N$ e, adicionalmente, os seguintes 38 arcos:

$$(4, 8), (4, 7), (6, 12), (6, 10), (6, 10), (6, 9), (7, 13), (7, 13), (7, 11), (7, 11), (7, 10), (8, 11),$$

(9, 18), (9, 16), (9, 15), (9, 15), (9, 13), (9, 13), (9, 12), (10, 17), (10, 16), (10, 16),
 (10, 14), (10, 14), (10, 13), (11, 15), (11, 14), (12, 18), (12, 18),
 (12, 16), (12, 16), (12, 15), (13, 17), (13, 17), (13, 16), (14, 18), (14, 17) e (15, 18).

Nos 3 períodos de tempo, existem no total 116 variáveis.

Se for considerado que as tarefas são ordenadas pela ordem inversa, além dos arcos $(0, p_j)$, $j \in N_t$, obtêm-se os seguintes arcos adicionais no primeiro período:

(3, 7), (3, 10), (3, 12), (3, 13), (3, 15), (4, 11), (4, 13), (4, 14), (4, 16), (7, 14),
 (7, 16) e (7, 17).

Nos restantes períodos de tempo, obtêm-se os seguintes arcos adicionais:

(3, 7), (3, 7), (3, 9), (3, 9), (3, 10), (3, 12), (3, 12), (3, 13), (3, 15), (4, 8), (4, 10), (4, 10),
 (4, 11), (4, 13), (4, 13), (4, 14), (4, 16), (6, 12), (6, 13), (6, 15), (6, 15), (6, 16),
 (6, 18), (7, 11), (7, 13), (7, 13), (7, 14), (7, 16), (7, 16), (7, 17), (8, 14), (8, 14),
 (8, 15), (8, 17), (8, 17), (8, 18), (9, 15), (9, 16), (9, 18), (9, 18),
 (10, 16), (10, 17), (11, 17), (11, 17) e (11, 18).

Existe assim um total de $18 + 2 \times 55 = 128$ variáveis.

De notar que o número de restrições do modelo não depende tão fortemente da ordem das tarefas. Cada vértice $v \in V_t$, $0 < v \leq P - \min\{p_j : j \in N_t\}$ gera uma restrição de conservação de fluxo no modelo (4.4)–(4.8).

4.6 Resultados computacionais

4.6.1 Análise comparativa de resultados

Para avaliar o desempenho do modelo (4.4)–(4.8), foi realizado um extenso conjunto de testes computacionais, usando-se as instâncias de Kis e Kovács (2012). Nesta secção, são apresentados os resultados obtidos que são posteriormente comparados com os resultados obtidos através das três abordagens propostas por Kis e Kovács. A primeira abordagem é baseada na resolução directa de um modelo de programação inteira monolítico. As outras duas são abordagens de decomposição hierárquica que recorrem a métodos de planos de corte. Essas duas abordagens diferem precisamente nos cortes que são aplicados. Daqui em diante, essas abordagens serão designadas por *monolítica*, *hierárquica H1* e *hierárquica H2*, tal como em Kis e Kovács (2012).

Nestes testes computacionais, foram utilizadas 405 instâncias que foram geradas de forma aleatória por Kis e Kovács (2012). As instâncias são divididas em três conjuntos A, B e C, que diferem essencialmente nos comprimentos relativos dos tempos de processamento das tarefas. O conjunto A contém apenas tarefas com pequenos tempos de processamento, *i.e.* com $p_j \leq P/3$ para cada tarefa $j \in N$, enquanto que o conjunto C tem apenas tarefas com tempos de processamento elevados, *i.e.* com $p_j > P/3$, $j \in N$. No conjunto B, metade das tarefas têm tempos de processamento pequenos, enquanto que a outra metade tem tempos de processamento elevados. Para todas as instâncias, o tempo P disponível em cada máquina foi definido para 100, como em Kis e Kovács (2012). Estes conjuntos de instâncias foram depois divididas em três subclasses de acordo com o número de tarefas $|N|$. Para o conjunto A, tem-se $|N| \in \{100, 150, 200\}$. Para os conjuntos B e C, tem-se respectivamente $|N| \in \{50, 100, 150\}$ e $|N| \in \{40, 60, 80\}$. Dentro de cada subconjunto, os grupos de instâncias foram geradas variando o número de máquinas $|M|$ (com $|M| \in \{2, 6, 10\}$), e o número de períodos de tempo τ_0 . Para garantir a validade das instâncias, Kis e Kovács (2012) consideraram períodos de tempos adicionais onde as tarefas podem terminar, fazendo:

$$\tau := \tau_0 + \left\lceil 2 \times \sum_{j \in N} p_j / (P \times |M|) \right\rceil, \quad (4.10)$$

com $\tau_0 \in \{2, 6, 10\}$. Para cada combinação de parâmetros $|N|$, $|M|$ e τ_0 , foram geradas 5 instâncias. Para garantir a compatibilidade dos resultados obtidos com os resultados de Kis e Kovács (2012), estes testes foram também executados num computador equivalente com processador Intel Core 2 Quad de 2.33 GHz e 3 GB de RAM. Foi ainda utilizada a versão 12.2 do CPLEX para implementar algumas das sub-rotinas de optimização.

As Tabelas 4.2 e 4.3 apresentam a média dos resultados obtidos usando o modelo de fluxos (4.4)–(4.8) segundo 3 cenários diferentes. No primeiro cenário, foi considerado o modelo (4.4)–(4.8) conforme foi definido na Secção 4.3, *i.e.* sem restrições ao conjunto dos *grid points*. Nos segundo e terceiro cenários, é considerado o modelo de fluxo em rede com as simplificações obtidas, respectivamente, restringindo os conjuntos de *grid points* através das Restrição I e Restrição II referidas na Secção 4.4. Na Tabela 4.2, são apresentados os resultados médios obtidos por número de tarefas e períodos de tempo como em Kis e Kovács (2012), *i.e.* agrupando as instâncias com o mesmo valor de $|N|$ e τ_0 . Cada linha dessas tabelas refere-se aos resultados obtidos para grupos de 15 instâncias. Na Tabela 4.3, os resultados são apresentados por número de tarefas e máquinas, *i.e.* para instâncias com o mesmo valor de $|N|$ e $|M|$. O significado das colunas nessas tabelas é o seguinte:

- $|N|$: número de tarefas;
- $|M|$: número de máquinas;
- τ_0 : comprimento nominal do horizonte de planeamento, com τ definido segundo (4.10);
- *opt*: número de instâncias resolvidas até à optimalidade;
- *gap*: intervalo médio de optimalidade, *i.e.*
 $(\text{melhor limite superior} - \text{melhor limite inferior}) / (\text{melhor limite superior})$;

- t_p : tempo médio de computação necessário para construir o correspondente modelo de fluxos em rede (4.4)–(4.8) (em segundos);
- t : tempo médio de resolução (em segundos).

Para cada instância, foi definido um tempo limite de computação de 1200 segundos. Nos casos em que a instância não é resolvida até à optimalidade, o tempo total corresponde ao tempo de processamento até a execução ser parada.

Adicionalmente, na Tabela 4.4, é apresentado o tamanho médio do modelo (4.4)–(4.8) para cada grupo de instâncias e cenários. O objectivo é mostrar de que modo o tamanho do modelo aumenta com os diferentes conjuntos de instâncias que foram consideradas, e o potencial impacto sobre a qualidade das soluções obtidas. As colunas r e v da Tabela 4.4 indicam, respectivamente, o número médio de restrições e variáveis no modelo (4.4)–(4.8). Visto que o número $|M|$ de máquinas disponíveis não influencia o tamanho do modelo, os diferentes casos $|M| \in \{2, 6, 10\}$ são agregados nesta tabela. Cada linha descreve assim os resultados obtidos para grupos de 15 instâncias com o mesmo número $|N|$ de tarefas e de valor de τ_0 , e para os três cenários considerados nas Tabelas 4.2 e 4.3.

Um total de 50% das instâncias do conjunto A foram resolvidas até à optimalidade com o modelo (4.4)–(4.8), e sem restrições ao conjunto de *grid points*. O intervalo médio de optimalidade é de 23%. Como se pode ver na Tabela 4.4, para estas instâncias, a restrição ao conjunto de *grid points* teve um impacto pequeno no tamanho do modelo. Consequentemente, as soluções obtidas através do modelo (4.4)–(4.8) foram praticamente as mesmas independentemente do número de restrições e variáveis. Comparando os resultados com os obtidos por Kis e Kovács (2012), conclui-se que o desempenho do modelo apresentado neste capítulo é equivalente ao da abordagem *monolítica* de Kis e Kovács (2012), em termos de instâncias resolvidas até à optimalidade e tempos de computação. Por outro lado, os métodos *hierárquica H1* e *hierárquica H2* destes autores superaram o modelo de fluxos (4.4)–(4.8) nos três cenários, uma vez que conseguiram resolver mais instâncias até à optimalidade. No entanto, é importante notar que para podermos avaliar com clareza o modelo de fluxos em rede (4.4)–(4.8), não foram utilizados quaisquer tipos de cortes que poderiam ser

usados para fortalecer o mesmo. Tal como acontece nos três métodos descritos em Kis e Kovács (2012), os resultados obtidos através do modelo (4.4)–(4.8) melhoraram com o aumento do número de máquinas. O mesmo não acontece com o aumento do comprimento do horizonte de planeamento.

Para o conjunto B, os resultados superaram claramente os obtidos em Kis e Kovács (2012), conforme ilustrado nas Tabelas 4.2 e 4.3. No melhor caso, foram resolvidas 89 instâncias em 135, enquanto que a melhor abordagem de Kis e Kovács (2012) conseguiu resolver apenas 58. O tempo de computação médio correspondente foi de 460,04 segundos no melhor caso, enquanto que foram necessários 715,5 segundos no melhor caso descrito por estes autores. Além disso, o intervalo de optimalidade para estas instâncias é pequeno. O intervalo médio para todas estas instâncias, e nos três cenários, foi sempre inferior a 6%, não ultrapassando os 2% para o melhor cenário. A restrição ao conjunto de *grid points* não teve nenhum impacto nestas instâncias. Na verdade, pode-se observar na Tabela 4.4 que, enquanto o número de restrições do modelo (4.4)–(4.8) diminui ligeiramente com a restrição ao conjunto de *grid points*, o número de variáveis aumenta, por vezes, o que prejudica a qualidade das soluções obtidas. A razão para este aumento pode ser explicado pelo facto de não se terem ordenado as tarefas segundo o seu tempo de processamento. Conforme foi discutido na Secção 4.5, se as tarefas tivessem sido ordenadas, o número de variáveis em cada um dos três cenários podia ter sido reduzido. A importância da ordem das tarefas sobre o número de variáveis ficou aparente apenas ao resolver-se as instâncias da classe B. Com efeito, as instâncias dessa classe têm uma estrutura especial. As primeiras $|N|/2$ tarefas têm tempos de processamento pequenos, enquanto que a segunda metade tem tempos de processamento elevados. Pelo contrário, as instâncias das classes A e C não possuem esse tipo de características.

As Tabelas 4.2 e 4.3 mostram que foi possível resolver todas as instâncias do conjunto C de forma rápida em cada um dos três cenários, enquanto que a melhor abordagem descrita em Kis e Kovács (2012) só conseguiu resolver 115 dessas instâncias com tempos médios de 202,1 segundos. Em quase todas as instâncias, foi possível encontrar a solução óptima em menos de 1 segundo. O tempo máximo de computação necessário para resolver uma instância do conjunto C foi de 2,61 segundos. Estas foram

também as instâncias em que as restrições sobre os conjuntos de *grid points* tiveram o maior impacto, com uma clara redução no número de restrições e variáveis do modelo (4.4)–(4.8).

Como já foi referido, o tamanho do modelo de fluxos em rede (4.4)–(4.8) depende do número de vértices e arcos do grafo. Para as instâncias usadas nos testes, tem-se que $P = 100$ e como tal poderá haver no máximo $P - 1 = 99$ vértices intermédios, e conseqüentemente, no máximo 99τ restrições de conservação de fluxo (4.5). Claramente, o modelo torna-se mais difícil de resolver à medida que o valor de P aumenta. O número de arcos é igual ao número de variáveis binárias x_{ai}^t em (4.4)–(4.8). Para as instâncias da classe C, significa que existem no máximo $\mathcal{O}(|N|^2 \times \tau)$ arcos, porque no máximo apenas duas tarefas podem ser executadas num determinado período de tempo em cada máquina. Em contraste, podem existir até $\mathcal{O}(|N| \times P \times \tau)$ arcos nos conjuntos A e B. De notar que as restrições aos conjuntos de *grid points* produzem melhores resultados quando os tempos de processamento das tarefas não são muito pequenos.

4.6.2 Actualização dos resultados usando um computador recente

Nesta secção, são descritos os resultados obtidos nos conjuntos A, B e C de instâncias descritas na secção anterior, utilizando agora um computador recente e software de optimização actualizado. Os testes foram realizados num computador com um processador i7 de 3.5 GHz e 32 GB de RAM. Para as sub-rotinas de optimização, foi usada a versão 12.6.1 do CPLEX.

Para estes testes, foi considerado um conjunto maior de instâncias, fornecidas por Tamas Kis (primeiro autor do artigo Kis e Kovács (2012)). De seguida são apresentadas as características destas instâncias. O tempo disponível em cada máquina é fixado em $P = 100$ para todas as instâncias. No conjunto A de instâncias, foram gerados 5 grupos com $|N| \in \{100, 150, 200, 250, 300\}$. O conjunto B contém um grupo adicional de instâncias com $|N| = 50$. Para o conjunto C, tem-se $|N| \in \{40, 60, 80, 100\}$. O número de máquinas que foi considerado é tal que $|M| \in \{2, 6, 10\}$, e o número τ_0 de

Restrições aos conjuntos de <i>grid points</i>														
<i>conjunto</i>	$ N $	τ_0	<i>Sem restrições</i>				<i>Restrição I</i>				<i>Restrição II</i>			
			<i>opt</i>	<i>gap</i>	t_p	t	<i>opt</i>	<i>gap</i>	t_p	t	<i>opt</i>	<i>gap</i>	t_p	t
A	100	2	9	0,06	0,11	512,06	9	0,12	0,12	500,66	9	0,10	0,12	504,20
	100	6	10	0,20	0,14	403,82	10	0,17	0,15	404,29	10	0,14	0,16	404,39
	100	10	10	0,04	0,17	404,52	10	0,07	0,18	404,38	10	0,02	0,19	404,11
	150	2	5	0,18	0,25	933,91	5	0,12	0,25	873,93	5	0,14	0,26	965,33
	150	6	8	0,21	0,30	609,43	8	0,21	0,31	599,29	7	0,23	0,31	655,89
	150	10	10	0,27	0,35	418,68	10	0,27	0,35	415,10	10	0,28	0,35	416,67
	200	2	1	0,37	0,47	1174,63	0	0,31	0,46	1200,85	0	0,35	0,46	1201,18
	200	6	5	0,41	0,56	809,17	5	0,45	0,51	807,87	5	0,43	0,51	808,78
	200	10	10	0,30	0,54	466,41	10	0,30	0,58	482,75	10	0,30	0,59	455,61
<i>total/média conj. A</i>			68	0,23	0,32	636,96	67	0,23	0,33	632,12	66	0,22	0,33	646,24
B	50	2	15	0,00	0,03	34,91	15	0,00	0,04	103,71	15	0,00	0,05	14,99
	50	6	15	0,00	0,04	36,54	15	0,00	0,05	51,89	15	0,00	0,05	21,01
	50	10	15	0,00	0,04	6,82	15	0,00	0,06	9,32	15	0,00	0,06	8,18
	100	2	11	0,00	0,13	439,37	9	0,00	0,17	653,75	11	0,00	0,19	419,79
	100	6	8	0,01	0,14	574,73	7	0,01	0,20	661,42	8	0,01	0,21	599,21
	100	10	10	0,01	0,16	412,56	9	0,02	0,21	489,19	10	0,01	0,24	412,31
	150	2	4	0,03	0,30	950,19	2	0,12	0,41	1120,44	6	0,08	0,44	893,79
	150	6	3	0,10	0,33	1003,96	2	0,18	0,49	1094,90	3	0,11	0,48	1032,09
	150	10	8	0,05	0,35	681,32	5	0,24	0,47	807,50	5	0,18	0,54	810,01
<i>total/média conj. B</i>			89	0,02	0,17	460,04	79	0,06	0,23	554,68	88	0,04	0,25	467,93
C	40	2	15	0,00	0,00	0,11	15	0,00	0,00	0,06	15	0,00	0,00	0,09
	40	6	15	0,00	0,00	0,05	15	0,00	0,00	0,07	15	0,00	0,00	0,06
	40	10	15	0,00	0,00	0,03	15	0,00	0,00	0,06	15	0,00	0,00	0,03
	60	2	15	0,00	0,01	0,36	15	0,00	0,01	0,19	15	0,00	0,01	0,32
	60	6	15	0,00	0,01	0,10	15	0,00	0,01	0,11	15	0,00	0,00	0,07
	60	10	15	0,00	0,00	0,07	15	0,00	0,01	0,23	15	0,00	0,00	0,05
	80	2	15	0,00	0,02	1,25	15	0,00	0,01	0,48	15	0,00	0,02	1,02
	80	6	15	0,00	0,01	0,25	15	0,00	0,01	0,60	15	0,00	0,01	0,19
	80	10	15	0,00	0,01	0,15	15	0,00	0,02	0,40	15	0,00	0,00	0,10
<i>total/média conj. C</i>			135	0,00	0,01	0,26	135	0,00	0,01	0,24	135	0,00	0,01	0,22

Tabela 4.2: Resultados computacionais por número de tarefas e períodos de tempo.

		Restrições aos conjuntos de <i>grid points</i>												
		Sem restrições				Restrição I				Restrição II				
conjunto	$ N $	$ M $	<i>opt</i>	<i>gap</i>	t_p	<i>t</i>	<i>opt</i>	<i>gap</i>	t_p	<i>t</i>	<i>opt</i>	<i>gap</i>	t_p	<i>t</i>
A	100	2	0	0,30	0,23	1200,68	0	0,36	0,25	1200,73	0	0,26	0,25	1200,59
	100	6	14	0,00	0,11	112,09	14	0,00	0,11	100,47	14	0,00	0,12	103,89
	100	10	15	0,00	0,09	7,63	15	0,00	0,09	8,13	15	0,00	0,09	8,23
	150	2	0	0,61	0,51	1205,04	0	0,61	0,52	1202,57	0	0,63	0,52	1204,30
	150	6	8	0,05	0,22	614,31	8	0,01	0,23	606,06	7	0,01	0,24	656,55
	150	10	15	0,00	0,16	142,68	15	0,00	0,17	79,68	15	0,00	0,17	177,05
	200	2	0	0,75	0,96	1203,98	0	0,75	0,90	1202,06	0	0,77	0,91	1201,96
	200	6	5	0,28	0,37	855,55	5	0,30	0,39	873,13	5	0,32	0,38	846,63
	200	10	11	0,05	0,25	390,69	10	0,02	0,28	416,27	10	0,01	0,27	416,98
<i>total/média conj. A</i>			68	0,23	0,32	636,96	67	0,23	0,33	632,12	66	0,22	0,33	646,24
B	50	2	15	0,00	0,06	75,36	15	0,00	0,08	160,61	15	0,00	0,09	41,00
	50	6	15	0,00	0,03	2,00	15	0,00	0,04	3,13	15	0,00	0,04	2,03
	50	10	15	0,00	0,02	0,91	15	0,00	0,03	1,19	15	0,00	0,03	1,14
	100	2	2	0,02	0,26	1134,81	1	0,02	0,35	1174,74	1	0,02	0,38	1166,95
	100	6	12	0,00	0,10	276,44	9	0,01	0,14	534,66	13	0,00	0,15	246,49
	100	10	15	0,00	0,07	15,41	15	0,00	0,10	94,96	15	0,00	0,11	17,87
	150	2	0	0,16	0,61	1201,37	0	0,48	0,86	1201,55	0	0,33	0,91	1201,52
	150	6	3	0,03	0,22	1075,50	0	0,05	0,30	1201,13	1	0,03	0,33	1123,86
	150	10	12	0,00	0,15	358,60	9	0,00	0,20	620,16	13	0,00	0,22	410,52
<i>total/média conj. B</i>			89	0,02	0,17	460,04	79	0,06	0,23	554,68	88	0,04	0,25	467,93
C	40	2	15	0,00	0,00	0,11	15	0,00	0,00	0,11	15	0,00	0,00	0,09
	40	6	15	0,00	0,00	0,05	15	0,00	0,00	0,05	15	0,00	0,00	0,06
	40	10	15	0,00	0,00	0,03	15	0,00	0,00	0,03	15	0,00	0,00	0,03
	60	2	15	0,00	0,01	0,36	15	0,00	0,01	0,39	15	0,00	0,01	0,32
	60	6	15	0,00	0,01	0,10	15	0,00	0,01	0,09	15	0,00	0,00	0,07
	60	10	15	0,00	0,00	0,07	15	0,00	0,00	0,05	15	0,00	0,00	0,05
	80	2	15	0,00	0,02	1,25	15	0,00	0,02	1,11	15	0,00	0,02	1,02
	80	6	15	0,00	0,01	0,25	15	0,00	0,01	0,23	15	0,00	0,01	0,19
	80	10	15	0,00	0,01	0,15	15	0,00	0,01	0,13	15	0,00	0,00	0,10
<i>total/média conj. C</i>			135	0,00	0,01	0,26	135	0,00	0,01	0,24	135	0,00	0,01	0,22

Tabela 4.3: Resultados computacionais por número de tarefas e máquinas.

Restrições aos conjuntos de <i>grid points</i>								
<i>conjunto</i>	$ N $	τ_0	<i>Sem restrições</i>		<i>Restrição I</i>		<i>Restrição II</i>	
			r	v	r	v	r	v
A	100	2	1181	81548	1181	82095	1181	82095
	100	6	1616	103922	1615	104954	1610	104938
	100	10	1977	123427	1973	123562	1956	123509
	150	2	1689	180727	1689	179790	1689	179790
	150	6	2116	215687	2115	215219	2115	215217
	150	10	2443	240592	2442	240610	2437	240598
	200	2	2187	315735	2187	315603	2187	315603
	200	6	2533	352543	2533	351680	2533	351680
	200	10	2899	389060	2898	390902	2896	390896
B	50	2	1341	22414	1326	29323	1299	29249
	50	6	1667	26718	1612	34027	1550	33843
	50	10	2027	30807	1983	40513	1867	40177
	100	2	2442	93408	2441	122809	2437	122807
	100	6	2869	103723	2860	137961	2831	137887
	100	10	3238	114680	3206	149710	3166	149596
	150	2	3588	216419	3588	280167	3588	280167
	150	6	4002	228261	3999	302381	3994	302370
	150	10	4395	246416	4387	320895	4364	320819
C	40	2	284	1804	205	1467	174	1434
	40	6	350	2358	265	2004	219	1884
	40	10	356	2329	254	2003	225	1937
	60	2	522	4867	388	4116	324	3887
	60	6	587	5534	456	4809	388	4601
	60	10	634	6497	487	5633	431	5490
	80	2	778	9576	616	8338	525	7951
	80	6	837	11167	664	9555	560	8936
	80	10	950	12084	759	10590	658	10315

Tabela 4.4: Tamanho médio do modelo de fluxos em rede (4.4)–(4.8).

períodos de tempo foi definido como segue: $\tau_0 \in \{2, 6, 10\}$. No total, foram geradas 675 instâncias.

Tal como na Secção 4.6, o modelo de fluxos em rede (4.4)–(4.8) foi testado sob três cenários diferentes, conforme ilustrado nas Tabelas 4.5 a 4.10. Tal como nos testes anteriores, o primeiro cenário corresponde à resolução do modelo (4.4)–(4.8) conforme foi definido na Secção 4.3, *i.e.* sem restrições ao conjunto dos *grid points*. No segundo e terceiro cenários, foram considerados o modelo de fluxos em rede com as Restrições I e II ao conjunto dos *grid points*.

No total, 40% das instâncias do conjunto A são resolvidas até à optimalidade com o modelo (4.4)–(4.8) sem restrições ao conjunto de *grid points*. O intervalo médio de optimalidade é igual a 25% se forem consideradas todas as 225 instâncias deste conjunto, embora diminua para apenas 10% em média se só forem consideradas as instâncias do conjunto A resolvidas por Kis e Kovács (2012). Os resultados melhoraram ligeiramente em termos do número de instâncias resolvidas até à optimalidade e em tempo de computação quando as restrições ao conjunto de *grid points* são aplicadas. A redução no tempo de computação obtido através da aplicação destas restrições atingiu um máximo de 88%.

No conjunto de instâncias B, o modelo de fluxos supera claramente as abordagens descritas em Kis e Kovács (2012), conforme ilustrado nas Tabelas 4.6 e 4.9, tal como já acontecia nos testes descritos na Secção anterior. No melhor caso, foram resolvidas 101 instâncias das 135 usadas em Kis e Kovács (2012), enquanto que a melhor abordagem desses autores resolveu apenas 58. Além disso, o intervalo médio de optimalidade para essas instâncias foi muito pequeno. O intervalo médio foi sempre inferior a 0,9%. As Tabelas 4.7 e 4.10 ilustram os resultados obtidos para as instâncias da classe C. Todas as instâncias são resolvidas até à optimalidade, incluindo as maiores com $|N| = 100$.

4.7 Abordagens heurísticas alternativas

O modelo de fluxos em rede apresentado acima permite resolver até à optimalidade muitas instâncias de média dimensão do problema integrado de planeamento e escalonamento. Nos casos em que o valor óptimo não é atingido, as instâncias são resolvidas

Restrições aos conjuntos de <i>grid points</i>										
$ N $	τ_0	Sem restrições			Restrição I			Restrição II		
		<i>opt</i>	<i>gap</i>	<i>t</i>	<i>opt</i>	<i>gap</i>	<i>t</i>	<i>opt</i>	<i>gap</i>	<i>t</i>
100	2	10	0,00	406,73	10	0,00	410,15	10	0,00	409,56
100	6	10	0,01	401,94	10	0,01	401,97	10	0,01	402,01
100	10	10	0,01	402,56	11	0,01	346,26	10	0,02	402,10
150	2	6	0,08	780,82	6	0,09	781,07	6	0,05	786,54
150	6	9	0,10	507,87	9	0,09	501,82	8	0,10	571,62
150	10	10	0,04	405,38	10	0,03	405,26	10	0,08	406,07
200	2	2	0,18	1150,12	3	0,18	1147,22	3	0,17	1127,86
200	6	5	0,23	802,93	5	0,23	803,29	7	0,22	700,59
200	10	10	0,28	426,85	10	0,23	438,92	10	0,24	420,05
250	2	0	0,33	1200,90	1	0,34	1130,63	2	0,39	1076,90
250	6	5	0,47	846,67	5	0,53	944,92	4	0,55	911,96
250	10	7	0,29	711,78	7	0,37	697,31	7	0,30	704,79
300	2	0	0,50	1201,46	0	0,51	1201,39	0	0,52	1201,29
300	6	0	0,62	1201,33	1	0,70	1133,21	1	0,64	1179,94
300	10	5	0,60	821,56	5	0,60	815,98	5	0,56	810,88
<i>total/média</i>		89	0,25	751,26	93	0,26	743,96	93	0,26	740,81

Tabela 4.5: Resultados computacionais por número de tarefas e períodos de tempo (instâncias A)

com intervalos de optimalidade que são muito reduzidos. As instâncias de referência que foram utilizadas nestes testes têm um número de itens $|N| \in [100, 300]$. Para essas instâncias, o número de arcos gerados no modelo de fluxos (4.4)–(4.7) tende a ser muito elevado. Para melhorar a resolução das instâncias mais difíceis, foram desenvolvidas e implementadas abordagens heurísticas alternativas que são descritas nesta secção.

Entre as instâncias que foram resolvidas, observa-se que as instâncias para as quais é mais difícil reduzir o intervalo de optimalidade são as instâncias com um número de tarefas $|N|$ superior a 100 e/ou aquelas cuja duração das tarefas p_i é pequena. Nestes dois casos, o modelo (4.4)–(4.7) tem um número elevado de arcos, o que dificulta a sua resolução através de um *solver* de optimização.

Na literatura, foram propostas diferentes heurísticas para resolver problemas in-

Restrições aos conjuntos de <i>grid points</i>											
N	τ_0	Sem restrições			Restrição I			Restrição II			
		<i>opt</i>	<i>gap</i>	<i>t</i>	<i>opt</i>	<i>gap</i>	<i>t</i>	<i>opt</i>	<i>gap</i>	<i>t</i>	
50	2	15	0,00	6,72	15	0,00	19,42	15	0,00	6,50	
50	6	15	0,00	4,27	15	0,00	8,04	15	0,00	4,60	
50	10	15	0,00	2,13	15	0,00	3,43	15	0,00	2,90	
100	2	14	0,00	159,99	10	0,00	465,74	14	0,00	229,48	
100	6	10	0,00	501,79	9	0,01	507,63	9	0,00	564,33	
100	10	11	0,00	387,19	9	0,01	483,11	10	0,00	405,40	
150	2	10	0,00	496,92	5	0,00	856,06	8	0,00	705,03	
150	6	3	0,01	967,17	3	0,02	978,78	3	0,01	970,37	
150	10	8	0,01	592,96	8	0,05	662,97	8	0,01	613,70	
200	2	7	0,02	773,23	6	0,12	858,03	7	0,09	815,41	
200	6	0	0,08	1200,92	1	0,11	1183,58	1	0,15	1169,62	
200	10	4	0,11	910,71	3	0,20	986,23	4	0,19	974,61	
250	2	1	0,14	1180,53	1	0,16	1158,93	1	0,16	1185,66	
250	6	0	0,15	1201,52	0	0,26	1201,74	0	0,21	1201,93	
250	10	0	0,22	1201,55	0	0,32	1201,71	0	0,28	1201,62	
300	2	1	0,16	1188,59	0	0,25	1202,03	0	0,25	1202,09	
300	6	0	0,28	1201,99	0	0,36	1202,30	0	0,31	1202,41	
300	10	0	0,38	1201,94	0	0,44	1202,45	0	0,40	1202,40	
<i>total/média</i>		114	0,09	732,23	100	0,13	787,90	110	0,11	758,78	

Tabela 4.6: Resultados computacionais por número de tarefas e períodos de tempo (instâncias B)

tegrados de planeamento e escalonamento. As heurísticas podem ser usadas para encontrar soluções iniciais de boa qualidade que podem eventualmente permitir derivar desigualdades válidas para os modelos, ajudando assim a fortalecê-los. Apesar das heurísticas poderem ajudar a encontrar boas soluções para instâncias difíceis, estas não garantem que a solução óptima do problema seja encontrada. Em Bensmaine et al. (2014), os autores propuseram uma abordagem heurística para resolver o problema integrado de planeamento e escalonamento de processos. O algoritmo consiste essencialmente em calcular o tempo de disponibilidade de cada máquina, e em afetar as operações às máquinas, segundo uma determinada ordem, na máquina com o

Restrições aos conjuntos de <i>grid points</i>										
$ N $	τ_0	<i>Sem restrições</i>			<i>Restrição I</i>			<i>Restrição II</i>		
		<i>opt</i>	<i>gap</i>	<i>t</i>	<i>opt</i>	<i>gap</i>	<i>t</i>	<i>opt</i>	<i>gap</i>	<i>t</i>
40	2	15	0,00	0,06	15	0,00	0,05	15	0,00	0,05
	6	15	0,00	0,07	15	0,00	0,06	15	0,00	0,04
	10	15	0,00	0,04	15	0,00	0,05	15	0,00	0,03
60	2	15	0,00	0,15	15	0,00	0,14	15	0,00	0,14
	6	15	0,00	0,10	15	0,00	0,11	15	0,00	0,09
	10	15	0,00	0,16	15	0,00	0,15	15	0,00	0,15
80	2	15	0,00	0,35	15	0,00	0,35	15	0,00	0,31
	6	15	0,00	0,43	15	0,00	0,37	15	0,00	0,34
	10	15	0,00	0,41	15	0,00	0,35	15	0,00	0,32
100	2	15	0,00	1,01	15	0,00	0,84	15	0,00	0,70
	6	15	0,00	0,75	15	0,00	0,63	15	0,00	0,53
	10	15	0,00	0,87	15	0,00	0,70	15	0,00	0,65
<i>total/média</i>		180	0,00	0,36	180	0,00	0,32	180	0,00	0,28

Tabela 4.7: Resultados computacionais por número de tarefas e períodos de tempo (instâncias C)

menor tempo disponível. O processo é repetido até todas as operações serem afetadas às máquinas. Em cada iteração, a ordem das operações é revista. Segundo os autores, o algoritmo permite resolver o problema de forma rápida, acomodando assim eficazmente mudanças de mercado ou falhas da máquina. Uma contribuição anterior de El-Rewini e Lewis (1990) descreve um problema semelhante em que é usada uma heurística para agendar tarefas para processamento em paralelo num multiprocessador. Recentemente, em (Lin et al., 2013), os autores definiram um algoritmo genético para ajudar a resolver um problema de escalonamento multi-objectivo em máquinas paralelas não relacionadas. Com esta heurística, (Lin et al., 2013) conseguiram gerar soluções de elevada qualidade.

Em Pinedo (2012), o autor descreve várias abordagens heurísticas diferentes para o problema de escalonamento em máquinas paralelas, considerando diferentes funções objectivo. Em (Dalfard e Mohammadi, 2012), os autores exploram um problema

Restrições aos conjuntos de <i>grid points</i>											
N	M	Sem restrições			Restrição I			Restrição II			
		<i>opt</i>	<i>gap</i>	<i>t</i>	<i>opt</i>	<i>gap</i>	<i>t</i>	<i>opt</i>	<i>gap</i>	<i>t</i>	
100	2	0	0,03	1200,64	1	0,03	1144,56	0	0,03	1200,50	
100	6	15	0,00	7,50	15	0,00	10,94	15	0,00	10,20	
100	10	15	0,00	3,08	15	0,00	2,87	15	0,00	2,96	
150	2	0	0,22	1200,72	0	0,22	1200,67	0	0,24	1200,63	
150	6	10	0,00	479,04	10	0,00	468,66	9	0,01	542,24	
150	10	15	0,00	14,31	15	0,00	18,82	15	0,00	21,35	
200	2	0	0,66	1201,16	0	0,63	1201,03	0	0,61	1200,98	
200	6	5	0,02	823,52	5	0,01	835,64	7	0,01	714,69	
200	10	12	0,00	355,21	13	0,00	352,76	13	0,00	332,83	
250	2	0	0,75	1201,32	0	0,73	1201,38	0	0,77	1201,29	
250	6	2	0,35	1106,91	2	0,50	1092,39	2	0,42	1099,94	
250	10	10	0,00	451,13	11	0,00	479,09	11	0,05	392,42	
300	2	0	0,78	1202,32	0	0,80	1202,14	0	0,81	1202,15	
300	6	0	0,61	1201,02	0	0,71	1201,14	0	0,58	1200,95	
300	10	5	0,32	821,03	6	0,30	747,28	6	0,34	789,01	
<i>total/média</i>		89	0,25	751,26	93	0,26	743,96	93	0,26	740,81	

Tabela 4.8: Resultados computacionais por número de tarefas e máquinas (instâncias A)

multi-objectivo de escalonamento flexível em máquinas paralelas. As abordagens propostas em Dalfard e Mohammadi (2012) permitiram resolver instâncias de pequena e média dimensão de forma eficiente. Em Rietz et al. (2015), os autores desenvolveram duas heurísticas para o problema integrado de planeamento e escalonamento estudado neste capítulo. Propuseram uma heurística construtiva e uma heurística baseada no problema de transporte com as quais obtiveram resultados de boa qualidade.

O uso de heurísticas em problemas \mathcal{NP} -difíceis pode ajudar a gerar soluções de qualidade razoável para instâncias de média e grande dimensão. No contexto da resolução de modelos de programação inteira, as heurísticas podem revelar-se uma boa forma de obter uma solução inicial ou de melhorar a solução dada pelo *solver* de optimização. As heurísticas usadas para gerar soluções iniciais são procedimentos normalmente simples e rápidos. Iniciar a resolução de um modelo de programação

		Restrições aos conjuntos de <i>grid points</i>								
$ N $	$ M $	<i>Sem restrições</i>			<i>Restrição I</i>			<i>Restrição II</i>		
		<i>opt</i>	<i>gap</i>	<i>t</i>	<i>opt</i>	<i>gap</i>	<i>t</i>	<i>opt</i>	<i>gap</i>	<i>t</i>
50	2	15	0,00	11,54	15	0,00	28,62	15	0,00	11,81
50	6	15	0,00	1,05	15	0,00	1,62	15	0,00	1,57
50	10	15	0,00	0,54	15	0,00	0,65	15	0,00	0,63
100	2	5	0,01	931,19	2	0,01	1077,35	4	0,01	1006,27
100	6	15	0,00	114,12	12	0,00	293,83	14	0,00	183,60
100	10	15	0,00	3,65	14	0,00	85,30	15	0,00	9,34
150	2	0	0,01	1200,80	0	0,05	1200,99	0	0,02	1200,91
150	6	8	0,00	662,10	3	0,01	1059,86	6	0,01	857,90
150	10	13	0,00	194,15	13	0,00	236,97	13	0,00	230,29
200	2	0	0,19	1201,45	0	0,36	1201,93	0	0,39	1201,84
200	6	2	0,01	1091,52	1	0,06	1145,99	2	0,03	1107,98
200	10	9	0,01	591,90	9	0,01	679,92	10	0,01	649,82
250	2	0	0,41	1202,44	0	0,45	1203,26	0	0,51	1203,24
250	6	0	0,09	1201,23	0	0,21	1201,03	0	0,08	1200,88
250	10	1	0,01	1179,93	1	0,08	1158,10	1	0,06	1185,09
300	2	0	0,46	1203,55	0	0,53	1204,29	0	0,53	1204,45
300	6	1	0,29	1188,11	0	0,36	1201,46	0	0,28	1201,34
300	10	0	0,08	1200,86	0	0,17	1201,04	0	0,15	1201,11
<i>total/média</i>		114	0,09	732,23	100	0,13	787,90	110	0,11	758,78

Tabela 4.9: Resultados computacionais por número de tarefas e máquinas (instâncias B)

inteira com uma solução gerada de forma heurística permite reduzir rapidamente o intervalo de optimalidade e eliminar assim muitos nodos da árvore de partição, nos casos em que é usado o método de partição e avaliação. Estratégias heurísticas podem também ser usadas para simplificar os modelos reduzindo o seu tamanho. O inconveniente é o risco de se perder a solução óptima com as simplificações introduzidas. Contudo, na prática, o uso destas abordagens pode conduzir a soluções cujo intervalo de optimalidade, não sendo nulo, é suficientemente pequeno para justificar a sua utilização. Na seguinte secção, serão descritas as abordagens heurísticas inspiradas nesta última ideia.

		Restrições aos conjuntos de <i>grid points</i>								
		<i>Sem restrições</i>			<i>Restrição I</i>			<i>Restrição II</i>		
$ N $	$ M $	<i>opt</i>	<i>gap</i>	<i>t</i>	<i>opt</i>	<i>gap</i>	<i>t</i>	<i>opt</i>	<i>gap</i>	<i>t</i>
40	2	15	0,00	0,09	15	0,00	0,09	15	0,00	0,07
	6	15	0,00	0,06	15	0,00	0,04	15	0,00	0,04
	10	15	0,00	0,02	15	0,00	0,02	15	0,00	0,02
60	2	15	0,00	0,26	15	0,00	0,28	15	0,00	0,26
	6	15	0,00	0,09	15	0,00	0,06	15	0,00	0,08
	10	15	0,00	0,06	15	0,00	0,06	15	0,00	0,05
80	2	15	0,00	0,81	15	0,00	0,71	15	0,00	0,67
	6	15	0,00	0,22	15	0,00	0,23	15	0,00	0,17
	10	15	0,00	0,15	15	0,00	0,13	15	0,00	0,13
100	2	15	0,00	2,01	15	0,00	1,62	15	0,00	1,39
	6	15	0,00	0,38	15	0,00	0,33	15	0,00	0,30
	10	15	0,00	0,24	15	0,00	0,21	15	0,00	0,19
<i>total/média</i>		180	0,00	0,36	180	0,00	0,32	180	0,00	0,28

Tabela 4.10: Resultados computacionais por número de tarefas e máquinas (instâncias C)

4.7.1 Estratégias heurísticas

Nesta secção, são descritas duas estratégias heurísticas para melhorar a resolução do modelo de fluxos proposto neste capítulo. Ambas assentam na resolução iterativa da relaxação linear do modelo (4.4)–(4.8). A ideia é fixar iterativamente o valor das variáveis segundo determinados critérios, e terminar com a resolução do modelo de programação inteira após fixar esse subconjunto de variáveis.

A primeira abordagem é descrita no Algoritmo (2). Nessa abordagem, procurou-se reduzir o número de arcos do grafo associado ao modelo de fluxos, fixando o fluxo em alguns dos arcos, *i.e.* fixando o valor de algumas das variáveis do modelo. Fixar o valor de uma variável corresponde também a definir o período de tempo e a máquina onde vai ser executada uma determinada tarefa. Fixar o valor de algumas variáveis pode ajudar a resolver mais facilmente o modelo resultante. Contudo, pode também excluir

a solução óptima do modelo original, ou até mesmo tornar o problema impossível. Problemas impossíveis ocorrem quando as tarefas são alocadas a máquinas que ficam sem tempo para poder acomodar as restantes tarefas da instância.

O algoritmo começa por resolver a relaxação linear do modelo (4.4)–(4.8). De seguida, é escolhida a variável cujo valor é o mais elevado, e também superior a um determinado valor limite. Em caso de empate, é escolhida a variável que corresponde à tarefa com maior tempo de processamento. O valor da variável escolhida é fixado a 1. Se o valor da variável é fraccionário, isso corresponde a arredondar para cima o valor dessa variável. A relaxação do modelo resultante é então resolvida, e o processo repete-se até a solução obtida ser inteira ou até ser atingido o número máximo de iterações estipulado. Se ainda houver variáveis fraccionárias no final deste processo, o modelo (4.4)–(4.8) com as tarefas ainda por alocar é resolvido como um modelo de programação inteira.

A segunda abordagem que é considerada, difere essencialmente da primeira na forma como são fixadas as variáveis. Enquanto que no Algoritmo 2 é fixado o valor de um conjunto determinado de variáveis, no Algoritmo 3, é seleccionado um conjunto de variáveis que são forçadas a tomar valores inteiros (binários) na resolução seguinte do modelo (4.4)–(4.8). Inicialmente, as tarefas da instância são reordenadas segundo um determinado critério. Posteriormente, as variáveis são forçadas a tomar valores inteiros por essa ordem. Os modelos que são resolvidos nesta segunda abordagem são modelos de programação inteira mista baseados em (4.4)–(4.8). A ideia é deixar o modelo decidir se essas variáveis devem tomar o valor 0 ou 1, ao invés da abordagem anterior, onde o valor das variáveis é fixado exactamente. Após resolver o modelo de programação inteira mista, o valor das variáveis inteiras é fixado, *i.e.* a tarefa é alocada a um período de tempo e a uma máquina, reduzindo assim o tamanho da instância, e o processo é repetido até ser encontrada uma solução inteira, ou até se atingir um número máximo de iterações. Com esta abordagem, evita-se o problema de uma fixação inválida das variáveis, que conduz a um problema impossível. O inconveniente é que o modelo de programação inteira mista é mais difícil de resolver do que a relaxação linear associada ao modelo (4.4)–(4.8).

Para avaliar a sensibilidade destes algoritmos aos parâmetros que os definem, foram

Algoritmo 2: Abordagem heurística 1

Dados: Instância do problema integrado de planejamento e escalonamento e respectivo modelo (4.4)–(4.8);

Dados: $maxit$: número máximo de iterações;

Dados: ϵ : tolerância usada para fixar o valor das variáveis;

Resultado: Solução para a instância do problema;

Resolver a relaxação linear do modelo (4.4)–(4.8);

$contadorIteracoes \leftarrow 0$;

enquanto $contadorIteracoes < maxit$ e a solução corrente não é inteira **faça**

$maiorFluxo \leftarrow 0$;

$i \leftarrow 0$;

para todo $j \in N_t, t \in T, (a, a + p_j) \in E_t$ **faça**

se $x_{aj}^t \geq 1 - \epsilon$ **então**

se $x_{aj}^t \geq maiorFluxo$ **então**

se $x_{aj}^t == maiorFluxo$ **então**

se $p_j > p_i$ **então**

$maiorFluxo \leftarrow x_{aj}^t$;

$i \leftarrow j$;

fim

senão

$maiorFluxo \leftarrow x_{aj}^t$;

$i \leftarrow j$;

fim

fim

fim

fim

se $maiorFluxo > 0$ e é válido fixar $x_{aj}^t \leftarrow 1$ **então**

 % atendendo ao tempo máximo de processamento

$x_{aj}^t \leftarrow 1$;

 Resolver a relaxação linear do modelo (4.4)–(4.8) resultante;

$contadorIteracoes \leftarrow contadorIteracoes + 1$;

senão

$contadorIteracoes \leftarrow maxit$;

fim

fim

se A solução da última relaxação linear de (4.4)–(4.8) não for inteira **então**

 Resolver o último modelo de programação inteira (4.4)–(4.8);

fim

Algoritmo 3: Abordagem heurística 2

Dados: Instância do problema integrado de planeamento e escalonamento e respetivo modelo (4.4)–(4.8);

Dados: *maxit*: número máximo de iterações;

Dados: *crit*: critério de ordenação das tarefas da instância;

Resultado: Solução para a instância do problema;

Ordenar as tarefas da instância segundo o critério *crit*;

contadorIteracoes \leftarrow 0 ;

contadorTarefaAFixar \leftarrow 1 ;

enquanto *contadorIteracoes* < *maxit* e a solução corrente não é inteira **faça**

Definir as variáveis x_{aj}^t com $j == \text{contadorTarefaAFixar}$ como binária no próximo modelo a resolver ;

Resolver o modelo de programação inteira mista (4.4)–(4.8) resultante;

para todo $j == \text{contadorTarefaAFixar}, t \in T, (a, a + p_j) \in E_t$ **faça**

se $x_{aj}^t == 1$ **então**

$x_{aj}^t \leftarrow 1$;

fim

fim

contadorTarefaAFixar \leftarrow *contadorTarefaAFixar* + 1 ;

contadorIteracoes \leftarrow *contadorIteracoes* + 1 ;

fim

se A solução da última relaxação linear de (4.4)–(4.8) não for inteira **então**

Resolver o último modelo de programação inteira (4.4)–(4.8);

fim

conduzidos testes preliminares em algumas das instâncias descritas na Secção 4.6. Os testes realizados com estes algoritmos mostraram que a fixação de mais do que uma variável de cada vez conduz muitas vezes a problemas que são impossíveis. Na próxima secção, são descritos os resultados dos testes computacionais mais aprofundados que permitem ilustrar o desempenho destas abordagens.

4.7.2 Resultados computacionais

Para avaliar o desempenho das abordagens descritas na secção anterior, foram conduzidas várias experiências computacionais usando as 20 instâncias mais difíceis do conjunto A de Kis e Kovács (2012), *i.e.* aquelas para as quais se obtiveram os piores resultados através do modelo (4.4)–(4.8). Estas instâncias foram ordenadas pelo valor do intervalo de optimalidade verificado após o limite de 1200 segundos. Todas as instâncias têm um intervalo de optimalidade superior a 85%. Com o objectivo de realizar comparação com os testes efectuados na Secção 4.6.2, estes serão repetidos num computador com processador i7 de 3,5 GHz e 32 GB de RAM. O *solver* de optimização usado foi o CPLEX 12.6.1. O tempo máximo de resolução foi fixado também em 1200 segundos.

O significado das colunas nas tabelas apresentadas nesta secção é o seguinte:

- *inst*: número da instância na classe A;
- *z*: solução após 1200s de resolução;
- *b*: melhor limite inferior;
- *t*: tempo usado para resolver a instância;
- $|N|$: número de tarefas da instância;
- *gap*: intervalo de optimalidade, *i.e.* $(1 - b/z)$;
- *fv*: variáveis fixadas com a heurística.

Os parâmetros usados em cada uma das abordagens heurísticas descritas na secção anterior foram os seguintes. Para o Algoritmo 2:

- tolerância ϵ : 0.4, *maxit*: 200;

- tolerância ϵ : 0.2, *maxit*: 100;
- tolerância ϵ : 0.4, *maxit*: 100.

Para o Algoritmo 3, foram considerados adicionalmente os seguintes critérios de ordenação das tarefas e número máximo de iterações:

- *maxit* $\in \{10, 15\}$, *b1*: ordem decrescente do tempo de processamento das tarefas (mais demorada primeiro);
- *maxit* $\in \{10, 15\}$, *b2*: ordem crescente do tempo de processamento das tarefas (mais rápida primeiro);
- *maxit* $\in \{10, 15\}$, *b3*: ordem crescente da soma das penalidades $e_j + \ell_j$ (mais elevada primeiro).

As Tabelas 4.11 e 4.12 mostram os resultados dos testes computacionais obtidos com o Algoritmo 2. Nestas tabelas, as instâncias que não foram resolvidas são representadas através de uma linha em branco. A não resolução de uma instância pode acontecer por duas razões. O *solver* não conseguiu encontrar uma solução inteira válida no final do processo iterativo associado à heurística, ou a fixação das variáveis conduziu a um problema impossível.

Os resultados obtidos com o Algoritmo 2, com tolerância $\epsilon = 0,4$ e número máximo de iterações *maxit* = 200, mostram que foi possível resolver 17 das 20 instâncias seleccionadas com intervalos optimalidade que superam claramente os obtidos ao resolver-se o modelo (4.4)–(4.8) com um *solver* de optimização. Com estes parâmetros, o intervalo médio de optimalidade para as instâncias resolvidas foi de 34% em vez dos 89% com o modelo de fluxos (4.4)–(4.8). A soma do valor das funções objectivo (z) obtidos com o modelo de programação inteira (4.4)–(4.8) foi de 359054, baixando para 105176 com este conjunto de parâmetros. Os resultados obtidos com outros parâmetros mostram a sensibilidade do algoritmo aos mesmos. Com uma tolerância $\epsilon = 0,2$, o número de instâncias resolvidas diminui sensivelmente. Contudo, nos casos em que foi encontrada uma solução inteira, essa solução é normalmente bem melhor do que aquela que é obtida com o modelo (4.4)–(4.8).

Por outro lado, reduzir o número máximo de iterações do Algoritmo 2 parece piorar a qualidade dos resultados, conforme pode ser observado na Tabela 4.12. Esse efeito

pode ser explicado em parte pelo facto do último modelo de programação inteira que é resolvido após terem sido fixadas todas as variáveis ser ainda um problema de média dimensão. Além disso, o tempo remanescente para a sua resolução é reduzido, uma vez que uma boa parte do tempo total de computação é consumido nas iterações do algoritmo associadas à fixação das variáveis.

Nas Tabelas 4.13–4.14 são apresentados os resultados computacionais obtidos com o Algoritmo 3. Tal como para o Algoritmo 2, linhas em branco representam instâncias não resolvidas pela heurística para a respectiva combinação de parâmetros. As instâncias do grupo de 20 instâncias difíceis que não estão listadas na tabela não foram resolvidas pelo algoritmo em nenhuma das combinações de parâmetros consideradas. De forma geral, os resultados obtidos com o Algoritmo 3 são piores que os obtidos com o Algoritmo 2. Estes resultados podem ser explicados pelo facto do Algoritmo 3 obrigar à resolução de um modelo de programação inteira mista em cada iteração do processo, enquanto que o Algoritmo 2 implica apenas a resolução de uma relaxação linear (modelo de programação linear) que é muito mais simples de ser resolvida. Observou-se também que no Algoritmo 2 são fixadas mais variáveis do que no Algoritmo 3, o que pode também ser explicado pelo maior tempo que é necessário para resolver os modelos de programação inteira mista no Algoritmo 3.

Para o Algoritmo 3, foram testados três critérios diferentes de ordenação e dois valores diferentes para o número máximo de iterações. As Tabelas 4.13–4.14 ilustram os resultados computacionais obtidos. À medida que o número máximo *maxit* de iterações aumenta, os resultados tendem a piorar. Esse comportamento pode ser explicado pelo número crescente de problemas de programação inteira mista que têm de ser resolvidos. Entre os critérios de ordenação das tarefas, o melhor parece ser o *b1*, quando *maxit* = 10, *i.e.* a ordenação das tarefas por ordem decrescente do seu tempo de processamento. Aplicando esse critério, o algoritmo conseguiu obter soluções com um intervalo médio de optimalidade igual a 52%, que, apesar de ser pior do que o do Algoritmo 2, é significativamente melhor que aquele obtido com o modelo (4.4)–(4.8). Em alguns casos, as instâncias foram resolvidas com intervalos de optimalidade de apenas 10%. Com o critério *b2* que corresponde a ordenar as tarefas por ordem crescente do seu tempo de processamento, os resultados obtidos foram

muito semelhantes ao critério $b1$. Com este critério, o intervalo médio foi de 59%. O critério $b3$, que assenta na soma das penalidades $e_j + \ell_j$, levou aos piores resultados entre os três critérios considerados. Nesse caso, o intervalo médio de optimalidade subiu para 70%. Embora os resultados mostrem um intervalo médio de optimalidade alto (de 60%), é de notar que este é de 90% quando é resolvido o modelo de fluxos (4.4)–(4.8). Para o mesmo algoritmo, utilizando os mesmos critérios de ordenação e utilizando agora o parâmetro $maxit = 15$ (Tabela 4.14), com $b1$, o intervalo médio de optimalidade é de 44%, o mais baixo de toda a combinação de parâmetros. No caso de $b2$, o intervalo médio situa-se nos 69%, enquanto que com $b3$ é de 83%, sendo a média dos três de 65%. Em alguns casos, o algoritmo falhou mesmo em encontrar uma solução inteira quando tinha encontrada essa solução com os outros critérios.

<i>inst</i>	Modelo (4.4)–(4.8)					Algoritmo 2									
	<i>z</i>	<i>b</i>	<i>gap</i>	<i>t</i>	$ N $	$\epsilon = 0.4; maxit = 200$					$\epsilon = 0.2; maxit = 200$				
						<i>z</i>	<i>gap</i>	<i>b</i>	<i>fv</i>	<i>t</i>	<i>z</i>	<i>gap</i>	<i>b</i>	<i>fv</i>	<i>t</i>
57	2719	351,63	0,87	1200,7	150	417	0,15	352,97	41	1201,33					
101	9067	1316,57	0,85	1200,94	200	1485	0,08	1370,51	154	1201,05					
102	7053	1056,44	0,85	1200,95	200	3846	0,72	1062,37	112	1200,50					
146	16838	2485,00	0,85	1201,33	250	6926	0,64	2502,11	175	1201,25					
147	21814	2201,24	0,90	1201,3	250	13502	0,84	2207,62	112	1186,34					
148	21168	2473,28	0,88	1201,27	250	9608	0,74	2484,01	136	1200,84					
160	1804	251,20	0,86	1200,72	250	258	0,02	253,14	169	1187,72	267	0,06	251,31	47	1186,18
186	55509	8115,57	0,85	1202,2	300										
191	37001	4924,43	0,87	1202,29	300										
192	42723	6008,79	0,86	1201,75	300										
193	40312	5558,59	0,86	1203,04	300	12198	0,54	5593,14	200	1201,61					
194	45633	4994,76	0,89	1201,86	300	26025	0,81	5003,53	130	1201,40					
195	44228	4934,75	0,89	1201,87	300	29973	0,84	4945,53	117	1186,87					
206	1591	131,07	0,92	1201,12	300	139	0,05	132,62	200	1186,12	2188	0,94	132,31	57	1200,89
207	2582	146,52	0,94	1201,12	300	155	0,02	151,97	192	1188,30	153	0,04	147,27	49	1185,57
209	2628	119,00	0,95	1201,19	300	144	0,11	127,87	200	1165,60	145	0,18	119,00	46	1201,19
210	2715	127,85	0,95	1201,17	300	174	0,24	132,99	200	1201,08					
217	936	83,33	0,91	1200,79	300	87	0,00	87,00	200	417,86	87	0,04	83,33	100	1187,68
218	1395	126,78	0,91	1200,79	300	130	0,00	130,00	200	360,67	129	0,02	126,78	81	1192,42
219	1338	106,89	0,92	1200,63	300	109	0,01	107,38	200	1187,15	117	0,08	107,16	100	1190,43
<i>médias</i>	17953	2275,68	0,89	1201,35		6187	0,34	1567,34	161	1098,57	441	0,19	138,17	69	1192,05

Tabela 4.11: Resultados computacionais - Algoritmo 2 (parte I)

Modelo (4.4)–(4.8)						Algoritmo 2				
inst	z	b	gap	t	N	$\epsilon = 0.4; maxit = 100$				
						z	gap	b	fv	t
57	2719	351,63	0,87	1200,7	150	418	0,16	1190,75	41	352,971
101	9067	1316,57	0,85	1200,94	200	5950	0,78	1195,80	100	1336,07
102	7053	1056,44	0,85	1200,95	200	5688	0,81	1195,55	100	1060,07
206	1591	131,07	0,92	1201,12	300	142	0,06	1188,80	100	133,23
207	2582	146,52	0,94	1201,12	300	1799	0,92	1186,21	100	148,41
209	2628	119,00	0,95	1201,19	300	137	0,13	1185,54	100	119,12
210	2715	127,85	0,95	1201,17	300	1812	0,93	1200,66	100	129,76
médias	17953	2275,68	0,89	1201,35		2278	0,54	1191,90	92	468,52

Tabela 4.12: Resultados computacionais - Algoritmo 2 (parte II)

Modelo (4.4)–(4.8)			Algoritmo 3											
inst	z	gap	$b1; maxit = 10$				$b2; maxit = 10$				$b3; maxit = 10$			
			z	gap	b	t	z	gap	b	t	z	gap	b	t
57	2719	0,87	481	0,27	353,40	1200,92	386	0,08	355,42	1200,64	385	0,08	355,90	1201,09
101	9067	0,85	6010	0,78	1318,85	1201,56	8136	0,84	1316,65	1200,31				
102	7053	0,85	4858	0,78	1056,54	1201,20	7017	0,85	1056,47	1186,24				
160	1804	0,86	1198	0,79	251,20	1185,63	1855	0,86	251,20	1185,85	334	0,25	251,20	1200,94
206	1591	0,92	144	0,08	132,28	1185,76	2366	0,94	131,07	1200,44	2278	0,94	131,07	1200,92
207	2582	0,94	1920	0,92	147,00	1200,55	2536	0,94	146,55	1186,37	2573	0,94	146,52	1200,64
209	2628	0,95	1990	0,94	119,15	1185,93	2658	0,96	119,23	1200,64	2874	0,96	119,23	1186,90
210	2715	0,95	2351	0,95	127,85	1201,26	3032	0,96	127,92	1185,56	2707	0,95	128,00	1200,80
217	936	0,91	90	0,07	83,48	1201,31	86	0,01	84,86	1200,47	1231	0,93	83,33	1201,30
218	1395	0,91	130	0,02	126,78	1202,45	128	0,00	128,00	938,20	1085	0,88	126,78	1185,67
219	1338	0,92	115	0,07	107,16	1186,68	108	0,00	108,00	1068,59	181	0,41	106,89	1185,26
médias	3075	0,90	1753	0,52	347,61	1195,75	2573	0,59	347,76	1159,39	1516	0,70	160,99	1195,95

Tabela 4.13: Resultados computacionais com $maxit = 10$ do Algoritmo 3

Modelo (4.4)–(4.8)			Algoritmo 3											
<i>inst</i>	<i>z</i>	<i>gap</i>	<i>b1; maxit = 15</i>				<i>b2; maxit = 15</i>				<i>b3; maxit = 15</i>			
			<i>z</i>	<i>gap</i>	<i>b</i>	<i>t</i>	<i>z</i>	<i>gap</i>	<i>b</i>	<i>t</i>	<i>z</i>	<i>gap</i>	<i>b</i>	<i>t</i>
57	2719	0,87	385	0,08	355,90	1201,09	374	0,05	355,50	1200,84	422	0,17	352,12	1201,11
160	1804	0,86	1302	0,81	251,30	1185,67	1462	0,83	251,20	1185,88	1543	0,84	251,20	1185,92
206	1591	0,92	1834	0,93	131,07	1200,77	150	0,13	131,07	1185,45	2119	0,94	131,07	1185,63
207	2582	0,94	273	0,46	146,66	1185,91	2282	0,94	146,52	1185,59	2339	0,94	146,52	1185,63
209	2628	0,95	2239	0,95	119,03	1166,13	2559	0,95	119,01	1186,01	2658	0,96	119,00	1180,73
210	2715	0,95	383	0,67	127,85	1200,39	3078	0,96	128,18	1201,28	2714	0,95	127,85	1185,54
217	936	0,91	86	0,00	86,00	815,96	507	0,84	83,35	1185,45	1076	0,92	83,33	1186,16
218	1395	0,91	129	0,02	126,81	1201,37	283	0,55	126,78	1186,04	1423	0,91	126,78	1201,37
219	1338	0,92	113	0,05	107,16	1200,58	1340	0,92	106,89	1201,01				
<u>médias</u>	1968	0,91	749	0,44	161,31	1150,87	1337	0,69	160,94	1190,84	1787	0,83	167,23	1189,01

Tabela 4.14: Resultados computacionais com $maxit = 15$ do Algoritmo 3

4.8 Conclusões

Neste capítulo, foi apresentado um novo modelo de programação inteira exacto e abordagens heurísticas baseadas nesse modelo para o problema integrado de planeamento e escalonamento. O modelo é pseudo-polinomial e assenta numa representação do problema através de um grafo que representa a afectação das tarefas aos períodos de tempo e às máquinas. O modelo foi testado num vasto conjunto de instâncias da literatura. Os resultados mostram que, para muitas instâncias de média e grande dimensão, a resolução directa do modelo supera outras abordagens descritas recentemente por Kis e Kovács (2012).

Contudo, para algumas instâncias mais difíceis, o intervalo de optimalidade continua a ser elevado. Com o objectivo de melhorar a resolução dessas instâncias, foram propostas duas abordagens heurísticas, que consistem na resolução iterativa do modelo de fluxos em rede e na fixação das variáveis desse modelo. Os respetivos algoritmos heurísticos foram testados com essas instâncias nas mesmas condições que o modelo (4.4)–(4.8). Os testes mostram que é possível em muitos casos melhorar substancialmente os resultados usando esse tipo de abordagens heurísticas.

Capítulo 5

O problema de encaminhamento de veículos com múltiplas rotas e gestão integrada de inventários

Conteúdo

5.1	Introdução	111
5.2	O problema de encaminhamento de veículos com múltiplas rotas e gestão integrada de inventários	115
5.2.1	Definição	115
5.2.2	Dados e parâmetros	116
5.2.3	Exemplo de uma instância do problema	118
5.3	Um modelo pseudo-polinomial de fluxos em arcos	122
5.4	Resultados computacionais	124
5.5	Conclusões	130

5.1 Introdução

O problema de encaminhamento de veículos pode ser aplicado em problemas reais de distribuição logística, com o objectivo de reduzir os custos de transporte, que englo-

bam, entre outros, custos com condutores e veículos. A integração deste problema com o problema de gestão de inventários pode traduzir-se em poupanças consideráveis, uma vez que isto traz uma gestão dos recursos mais eficiente do que aquela conseguida através da optimização local dos dois problemas.

O encaminhamento de veículos e gestão integrada de inventário (*IRP - Inventory Routing Problem*) tem como objectivo minimizar o custo total das operações de transporte de mercadorias do fornecedor para o cliente, de forma a que este mantenha um inventário que lhe permita satisfazer as suas procuras a cada período de um dado horizonte de planeamento.

Este problema pode incorporar a informação do horizonte temporal, políticas de gestão de inventários, rotas ou tipo de frota assim como a dimensão da mesma (Coelho et al., 2014). As rotas consideram-se directas ou não, se se visitar um único cliente por rota ou vários, respectivamente (Coelho et al., 2014). O horizonte diz-se finito, se definido para um curto período, ou infinito, quando o escalonamento das rotas é efectuado para um longo período de tempo (Coelho et al., 2014, Bertazzi e Speranza, 2012).

Tipicamente, o objectivo é minimizar os custos gerais inerentes ao transporte, reduzindo as penalidades associadas ao inventário, que normalmente representam custos de armazenagem (Bertazzi e Speranza, 2012).

Várias aplicações práticas têm sido implementadas na indústria, o que permite às empresas reduzirem os custos de inventário, transporte e melhorarem a qualidade de serviço. Um recente estudo de Hanczar (2012) descreve a implementação deste problema numa empresa de distribuição de combustível. Um trabalho anterior de Miller (1987) descreve uma aplicação do problema numa empresa com uma frota de navios que entrega componentes químicos a armazéns espalhados por todo o mundo. Os autores apresentam um modelo de programação inteira que implementaram com sucesso na empresa. Uma grande mudança ocorreu numa empresa que fornece carbonato de cálcio para toda a Europa depois da consideração do problema de encaminhamento de veículos e gestão integrada de inventários na sua estratégia diária, tendo assim conseguido uma redução de custos de milhões de dólares por ano (Dauzère-Pérès et al., 2007). Diferentes abordagens deste problema têm vindo a ser aplicadas na indústria

marítima, podendo alguns exemplos ser encontrados em Al-Khayyal e Hwang (2007), Song e Furman (2013), Persson e Göthe-Lundgren (2005), Grønhaug et al. (2010).

O problema explorado neste capítulo é o problema de encaminhamento de veículos e gestão integrada de inventário em que é permitido aos veículos efectuarem mais de uma rota em cada período do horizonte de planeamento. O problema é designado por problema de encaminhamento de veículos com múltiplas rotas e gestão integrada de inventários. Na terminologia anglo-saxónica, este problema é designado por multi-trip inventory routing problem. A consideração das múltiplas rotas pode trazer vantagens no sentido em que o custo de utilização dos veículos é normalmente fixo por período de planeamento, podendo assim haver poupanças a este nível. Por outro lado, a consideração desta variante com múltiplas rotas torna o problema, que é já de si de difícil resolução, ainda mais complexo.

Uma vez que existe um crescente interesse na aplicação deste problema em casos práticos na indústria, vários algoritmos de programação inteira e heurísticas têm vindo a ser propostos por diversos autores da literatura. A dificuldade de resolução do problema de encaminhamento de veículos e gestão integrada de inventários tem vindo a motivar maioritariamente o desenvolvimento de heurísticas, que em muitos casos apresentam bons resultados (Herer e Levy, 1997, Archetti et al., 2011, Cordeau et al., 2015, Hemmati et al., 2015). Em Coelho e Laporte (2013) é resolvido de forma exacta o problema de encaminhamento de veículos e gestão integrada de inventários com múltiplos produtos e veículos. Os autores apresentam um modelo de programação inteira usando para a sua resolução exacta, ao qual são também adicionadas desigualdades válidas com o objectivo de reforçar o mesmo. Na resolução do modelo é utilizado um algoritmo de partição e avaliação sucessivas com cortes, capaz de resolver instâncias com um máximo de 5 veículos, 5 produtos, 7 períodos de planeamento e 50 clientes. O problema com uma procura constante por parte dos clientes foi abordada por Zhong e Aghezzaf (2012). Para a resolução desta variante, os autores recorrem a um método de relaxação Lagrangeana, obtêm limites inferiores e superiores para o seu modelo com o objectivo de obter soluções de boa qualidade em tempos aceitáveis. Yu et al. (2008) propõem também um método de relaxação Lagrangeana combinado com um método de subgradiente, capaz de resolver instâncias com até 200 clientes.

Em Archetti e Speranza (2016), os autores propõem dois modelos de programação inteira para resolver o problema de encaminhamento de veículos e gestão integrada de inventários para o problema de inventário, quando gerido pelo fornecedor e quando gerido pelo cliente, comparando as duas abordagens.

Os problemas de encaminhamento de veículos com múltiplas rotas são tipicamente de difícil resolução quando comparados com o problema usual de encaminhamento de veículos. Uma revisão da literatura desta variante foi proposta por Şen e Bülbül (Universidade Bilgi, Istambul, Turquia, 2008).

Uma vez que a resolução deste problema não é trivial, vários métodos heurísticos têm sido propostos. Taillard et al. (1996) descreve um algoritmo de pesquisa tabu para a resolução deste problema. O mesmo problema foi abordado por Brandão e Mercer (1998) onde é descrito um algoritmo heurístico também utilizando pesquisa tabu. Uma resolução utilizando uma heurística construtiva em três fases foi proposta por Petch e Salhi (2003). Um procedimento de memória adaptativa foi proposto em Olivera e Viera (2007), tendo os seus resultados sido comparados com os obtidos com outros métodos da literatura (Taillard et al., 1996, Brandão e Mercer, 1998, Petch e Salhi, 2003). Salhi e Petch (2007) propõem pela primeira vez um algoritmo genético para este problema. Um problema de encaminhamento de veículos com múltiplas rotas e restrições adicionais de acessibilidade foi estudado por Alonso et al. (2008), tendo estes proposto um algoritmo de pesquisa tabu com o qual foram resolvidas instâncias geradas com até 1000 clientes.

Azi et al. (2007) propõem um método de programação inteira exacto para o problema de encaminhamento de um único veículo com janelas temporais e múltiplas rotas. O algoritmo é dividido em duas fases: na primeira, todas as rotas válidas são geradas e na segunda as rotas são afectadas aos diferentes momentos do horizonte de planeamento. Os autores generalizaram posteriormente o algoritmo para o caso de múltiplos veículos (Azi et al., 2010). Os autores utilizam um algoritmo de geração de colunas, capaz de resolver instâncias com um número de clientes entre 25 e 50.

Em Macedo et al. (2011) é abordado o problema de encaminhamento de veículos com janelas temporais e múltiplas rotas. Os autores propõem um modelo de fluxos em rede pseudo-polinomial, onde os vértices do grafo subjacente correspondem a instantes

de tempo do horizonte de planeamento e em que os arcos definem rotas válidas. É proposto um algoritmo exacto que considera uma desagregação iterativa dos vértices do grafo, que são inicialmente agregados de forma a obter-se um modelo, mais pequeno e consequentemente de mais fácil resolução.

Neste capítulo é abordado o problema de encaminhamento de veículos com múltiplas rotas e gestão integrada de inventários. Na Secção 5.2 é introduzido o problema através da sua definição, considerações e parâmetros que o definem. O Exemplo 5.2.3 é representativo dos dados fornecidos ao problema através de uma instância. Para a resolução do problema foi proposto um novo modelo de fluxos em rede pseudo-polinomial, descrito na Secção 5.3, onde é realizada uma apresentação pormenorizada do mesmo. Resultados computacionais são apresentados na Secção 5.4 e por fim uma breve conclusão na Secção 5.5.

5.2 O problema de encaminhamento de veículos com múltiplas rotas e gestão integrada de inventários

5.2.1 Definição

A classe de problemas de encaminhamento de veículos e gestão integrada de inventários considera um contexto em que um ou mais tipos de produtos são enviados de um fornecedor a um conjunto de clientes através de uma frota de veículos. Os clientes têm de ser servidos durante diversos períodos de um horizonte de tempo, tendo para cada um deles uma determinada procura. O que diferencia esta classe de problemas do problema de encaminhamento de veículos é o facto de ser o fornecedor a gerir o inventário dos clientes, i.e., as quantidades de produto entregues a cada cliente em cada período não são necessariamente iguais às respectivas procuras. As entregas dos produtos têm apenas de ser feitas de modo a que os clientes tenham disponível, a cada período, a quantidade necessária de produtos.

Neste capítulo, é abordada uma variante deste problema, que considera o problema de encaminhamento de veículos com múltiplas rotas, ou seja, em que cada veículo pode ser afectado a mais do que uma rota em cada período do horizonte de planeamento.

Trata-se então do problema de encaminhamento de veículos com múltiplas rotas e gestão integrada de inventários.

Considera-se que existe armazém onde está localizada uma frota de veículos, que abastece um conjunto de clientes com um único tipo de produto.

O objectivo deste problema consiste em determinar o conjunto óptimo de rotas que minimizem os custos totais de transporte e os eventuais custos de armazenagem nos cliente relativos à gestão dos inventários. Ou seja, sempre que uma encomenda for entregue antes do período definido, incorre-se numa penalidade proporcional aos custos de armazenagem dos produtos nos clientes. Por outro lado, considera-se que os clientes têm uma capacidade ilimitada de armazenagem. No que diz respeito às entregas antecipadas, estas não podem ser parciais. Isto significa que toda a procura de um período é entregue numa única visita ao cliente correspondente, seja esta feita no próprio período ou em períodos anteriores. Neste problema, o número de veículos disponíveis é limitado, assim como a capacidade de cada veículo, sendo que a carga a transportar em cada rota não pode ultrapassar a sua capacidade. Assume-se que cada unidade de produto ocupa uma unidade de volume no veículo e que o tempo despendido no transporte é equivalente à distância percorrida. Cada veículo poderá realizar diversas rotas por período, na medida em que a soma das suas durações não ultrapasse a duração do mesmo, ou seja não ultrapasse a duração de um dia de trabalho.

5.2.2 Dados e parâmetros

Para clarificar a apresentação formal do problema, é apresentada em seguida uma lista exaustiva dos parâmetros que o caracterizam:

- $D = \{0\}$: armazém;
- $S = \{1, \dots, N\}$: clientes;
- $T = \{1, \dots, \tau\}$: períodos de tempo do horizonte de planeamento.

O armazém é associada ao índice 0. Os clientes estão localizados a uma determinada distância do armazém, distribuídos segundo as suas coordenadas cartesianas.

O horizonte de planeamento define o período de tempo para o qual as entregas aos clientes terão que ser efectuadas. Estes períodos serão posteriormente divididos por unidades de tempo designados por dias de trabalho.

Considera-se que um cliente não pode ser visitado mais do que uma vez em cada período de tempo e que existe um único tipo de produto. Além disso, assumimos que a visita a um cliente num período t obriga à entrega da procura desse período, e eventualmente a de períodos posteriores. Não são admitidas rupturas de *stock* (stock out), ou seja, todos os clientes têm que impreterivelmente ter à sua disposição, em cada período, as quantidades de produto definidas. Finalmente, considera-se que não existe *stock* inicial nos clientes, ou seja, no instante 0 do horizonte temporal nenhum dos clientes tem à sua disposição qualquer unidade de produto.

- C : capacidade dos veículos (frota homogénea);
- F : número de veículos disponíveis;
- W : duração de um dia de trabalho;
- d_i^t : procura do cliente i no período t ;
- N_{max} : número máximo de clientes visitados por rota.

Os custos considerados neste problema são os seguintes:

- C_v : custo fixo de utilização de um veículo por dia de trabalho;
- C_r : custo de transporte associado à rota r ;
- C_{h_i} : custo de armazenagem de uma unidade de produto no cliente i durante um período de tempo;
- $C_{H_r^t}$: custo total de armazenagem associado à rota r ($C_{H_r^t} = \sum_{i \in N_r} C_{h_i} t_r^i$, sendo t_r^i o tempo total de espera no cliente i recolhidos através da rota r).

Seguem algumas definições adicionais:

- Ψ_t : conjunto de rotas válidas no período t ;

- N_r : conjunto dos clientes visitados pela rota r ;
- α_{irt}^t : igual a 1 se na rota r é entregue a procura do fornecedor i no período t , ou igual a 0 caso contrário;
- uma rota r é caracterizada por um conjunto de clientes (visitados pela rota) e pelos períodos de tempo das procuras que são entregues no âmbito dessa mesma rota.

5.2.3 Exemplo de uma instância do problema

Exemplo 4. *Considere-se o exemplo de uma instância para o problema de encaminhamento de veículos com múltiplas rotas e gestão integrada de inventários.*

A Tabela 5.1 indica todos os parâmetros que a definem. Na Tabela 5.1a define-se a localização do armazém, na Tabela 5.1b define-se a capacidade dos veículos (C), o tamanho da frota (F), a duração de um dia de trabalho (W), o número de períodos do horizonte de planeamento (τ), e o número de clientes (N). Na Tabela 5.1c estão representadas as coordenadas cartesianas (x, y) dos clientes, assim como os custos unitários de armazenamento em cada um dos clientes C_{h_i} . Por último, a Tabela 5.1d define as procuras d_i^t nos períodos t para o cliente i .

A representação gráfica desta instância poderá ser observada na Figura 5.1, onde os vértices representam o armazém e os clientes distribuídos segundo as suas coordenadas cartesianas. Todas as ligações entre clientes e armazém estão também representadas, assim como as correspondentes distâncias a percorrer nesse trajecto.

A figura 5.2 representa uma solução válida para esta instância, que tem no primeiro período um custo de 214, no segundo 198 e no último 131, com um custo total de 543. Neste caso, não é possível utilizar apenas um veículo no primeiro período, uma vez que a distância que o mesmo teria que percorrer para recolher todas as procuras dos clientes é superior a 120 (duração de um dia de trabalho). Assim, dois veículos realizam as rotas para o primeiro período, sendo em cada uma das rotas entregue a procura para períodos posteriores, nomeadamente dos clientes 4 e 3. Estas recolhas para períodos posteriores incorrem numa penalidade por cada unidade em armazém. No segundo período, pode recorrer-se a um único veículo que realiza duas rotas. No

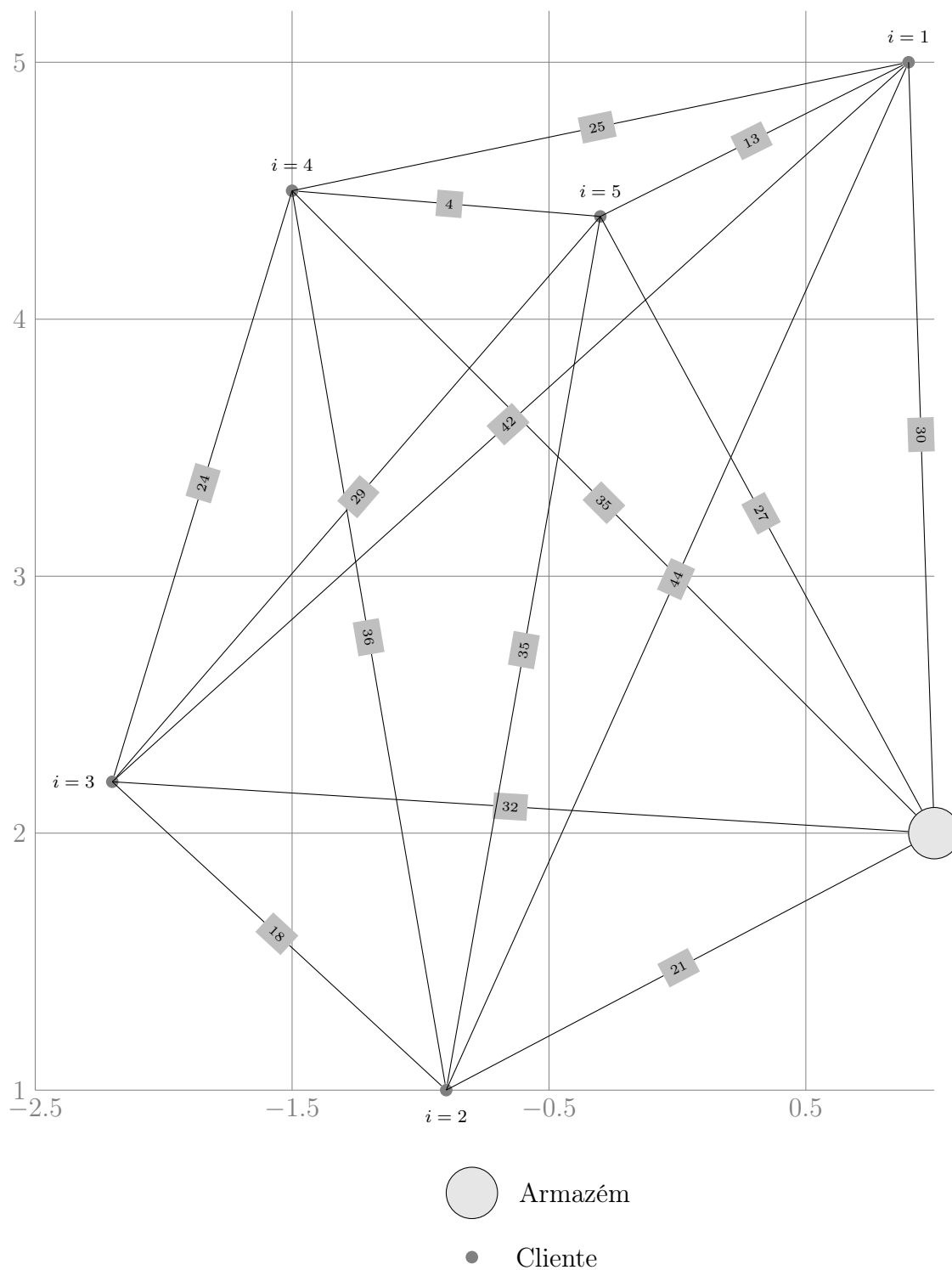


Figura 5.1: Representação da instância graficamente com o armazém, clientes e respectivas distâncias situadas a meia distância dos arcos.

						t	i	d_i^t		
						1	1	4		
						1	2	2		
						1	3	2		
						1	4	2		
						1	5	2		
(a)										
A	10	20	C	10	x	y	C_{h_i}	1	5	2
						2	1	2		
						2	2	2		
						2	3	2		
						2	4	2		
						2	5	2		
(b)						(c)				
						3	1	4		
						3	2	2		
						3	3	2		
						3	4	2		
						3	5	3		
						(d)				

Tabela 5.1: Exemplo de uma instância do problema de encaminhamento de veículos com múltiplas rotas e gestão integrada de inventários. (a) Dados da localização do Armazém, (b) Dados gerais do problema, (c) Dados dos clientes e (c) Dados das procuras dos clientes.

terceiro período, um só veículo realiza as entregas restantes aos clientes em que as procuras não foram ainda satisfeitas, recorrendo-se a uma só rota. Neste último período não há custos de armazenagem uma vez que todas as procuras são satisfeitas e consumidas nesse período. A Tabela 5.2 discrimina os valores da função objectivo para os três períodos representados na Figura 5.1.

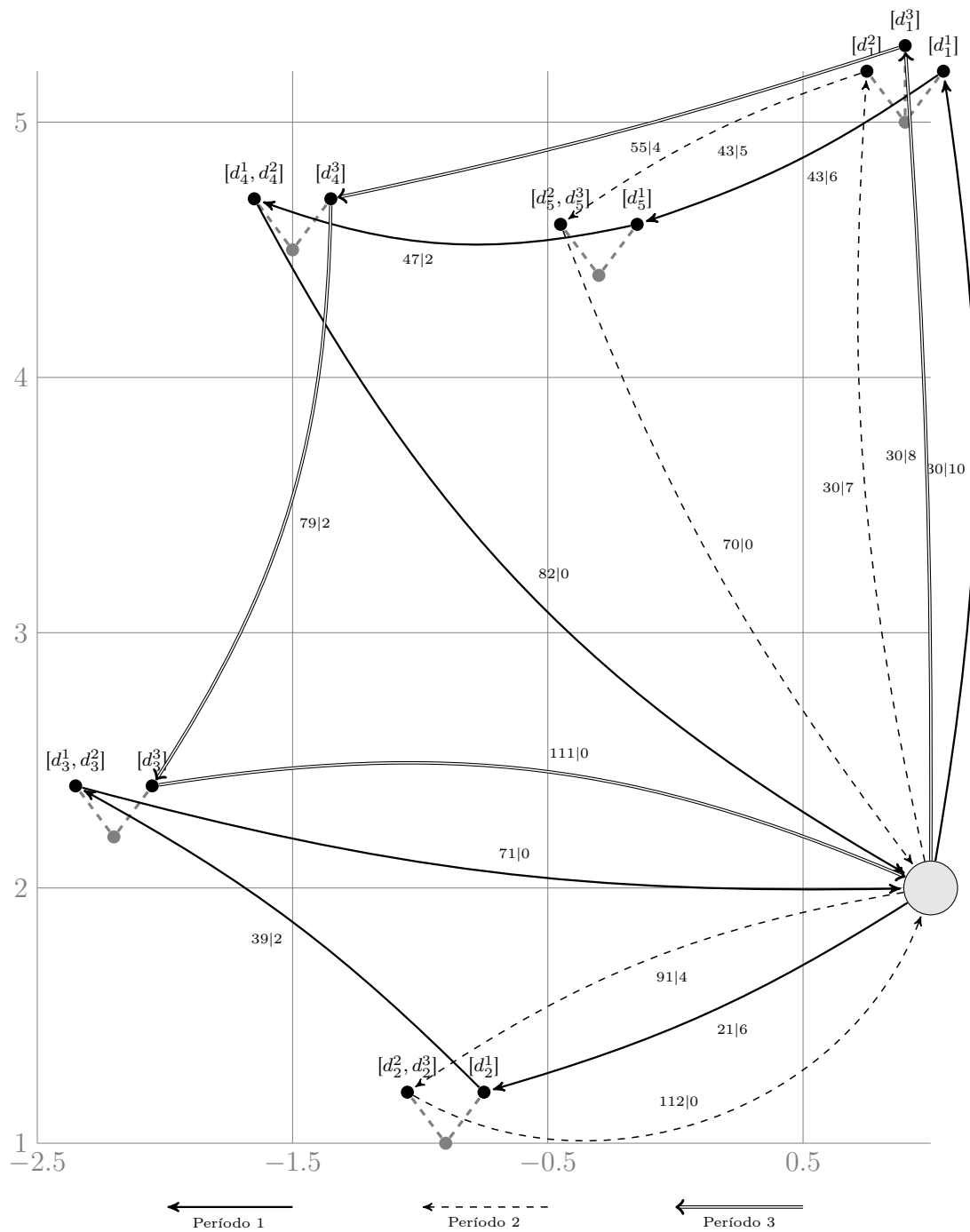


Figura 5.2: Solução válida para a instância IRP. A meio de cada ligação entre cliente é representada a distância percorrida, assim como o volume da mercadoria do veículo no trajeto.

T	C_r	C_v	C_{h_i}	total
1	$82 + 71$	$20 + 20$	$9 \times 2 + 3 \times 2$	214
2	112	20	$12 \times 3 + 15 \times 2$	198
3	111	20	0	131
				543

Tabela 5.2: Valores da função objectivo para os três períodos.

5.3 Um modelo pseudo-polinomial de fluxos em arcos

Nesta secção, é descrito um novo modelo de programação inteira para o problema de encaminhamento de veículos com múltiplas rotas e gestão integrada de inventários.

Trata-se de um modelo de fluxos em rede definido num conjunto de grafos acíclicos e directos, um por cada período do horizonte de planeamento, denotados por $G_t = (V, A_t)$, $t \in T$, sendo V o seu conjunto de vértices, $V = \{0, \dots, W + 1\}$, e A_t o seu conjunto de arcos, que representam o conjunto de todas as rotas válidas do período $t \in T$, assim como os tempos de espera no armazém. Um fluxo que percorre o grafo representa um dia de trabalho de um veículo, ou seja, a sequência de rotas e tempos de espera que este efectua desde o instante 0 até ao instante W de um dado período do horizonte de planeamento. Uma rota é definida por uma sequência de clientes a visitar, assim como as respectivas quantidades de produto a entregar a cada um destes. Para que uma rota seja válida, a soma das quantidades de produtos a entregar a cada fornecedor não pode ultrapassar a capacidade do veículo, nem o tempo de deslocação necessário pode ultrapassar um dia de trabalho (um período do horizonte de planeamento). Note-se que uma mesma rota pode começar em instantes diferentes, mantendo-se válida. O conjunto de todas as rotas válidas é gerado antecipadamente, sendo que no modelo a variável x_{uvr}^t representa a rota r que começa no instante u e termina no instante v do período $t \in T$. As rotas são geradas através de um processo recursivo que vai excluindo rotas que violem a capacidade dos veículos C e/ou a duração máxima da rota W .

$$\min \sum_{t \in T} \sum_{(u,v)^r \in \Psi_t} C_r x_{uv}^t + C_v \sum_{t \in T} \sum_{(0,v)^r \in \Psi_t} x_{0vr}^t + \sum_{t \in T} \sum_{(u,v)^r \in \Psi_t} C_{H_r^t} x_{uv}^t \quad (5.1)$$

$$\text{s.a.} \quad \sum_{t \in T, t \leq t'} \sum_{(u,v)^r \in \Psi_t | i \in N_r} \alpha_{irt}^{t'} x_{uv}^t = 1, \quad \forall i \in S, t' \in T, \quad (5.2)$$

$$\sum_{(0,v)^r \in \Psi_t} x_{0vr}^t \leq F, \quad \forall t \in T, \quad (5.3)$$

$$- \sum_{(u,v)^r \in \Psi_t} x_{uv}^t + \sum_{(v,y)^s \in \Psi_t} x_{vys}^t = \begin{cases} 0, & \text{if } v = 1, \dots, W-1, \\ - \sum_{(0,v)^r \in \Psi_t} x_{0vr}^t, & \text{if } v = W, \end{cases} \quad \forall t \in T, \quad (5.4)$$

$$x_{uv}^t \in \{0, 1\}, \forall (u,v)^r \in \Psi_t, \forall t \in T. \quad (5.5)$$

A função objectivo (5.1) representa a soma dos custos de transporte das rotas efectuadas, C_r , do custo de utilização dos veículos utilizados, C_v , e dos custo de armazenagem por dia e por item, C_{H_i} .

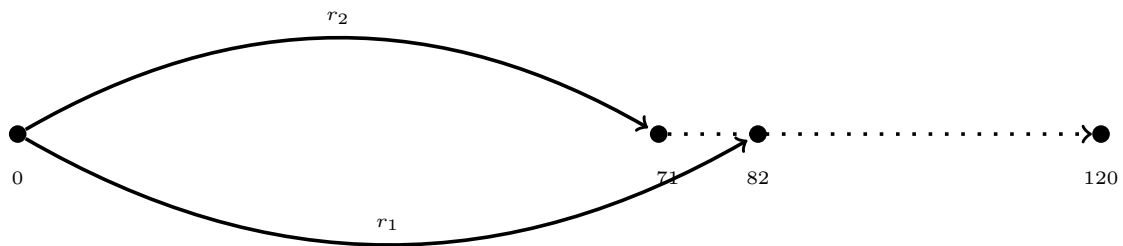
As restrições (5.2) garantem que as entregas das procuras de todos os períodos, para cada um dos clientes, são satisfeitas por uma e uma só das rotas efectuadas. Esta entrega pode ser efectuada no próprio período ou em períodos anteriores ao mesmo.

As restrições (5.3) impõem que não são utilizados mais de F veículos em cada período t . A conservação de fluxo é assegurada através das restrições (5.4).

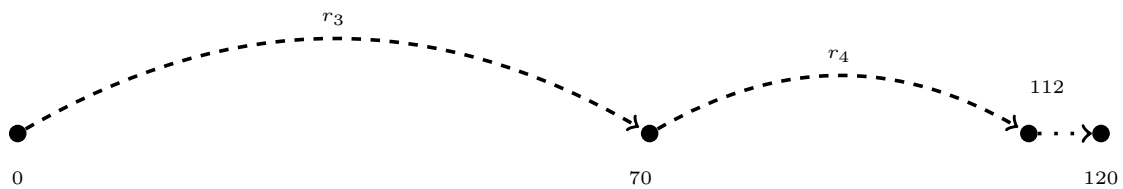
Exemplo 5. *Os grafos da Figura 5.3 representam a solução válida representada na Figura 5.2 do Exemplo 4.*

Estes grafos possuem uma dimensão $W = 120$ que representa a duração de um dia de trabalho, sendo 0 o início do mesmo e W o seu fim. O conjunto de arcos representados correspondem às rotas efectuadas. Cada vértice define um instante de tempo e cada arco representa uma rota percorrida para visitar um conjunto de clientes. Nestes fluxos estão definidos os arcos para os três períodos, tendo o primeiro (5.3a) dois veículos a realizar uma rota cada, no segundo período (5.3b), um único veículo

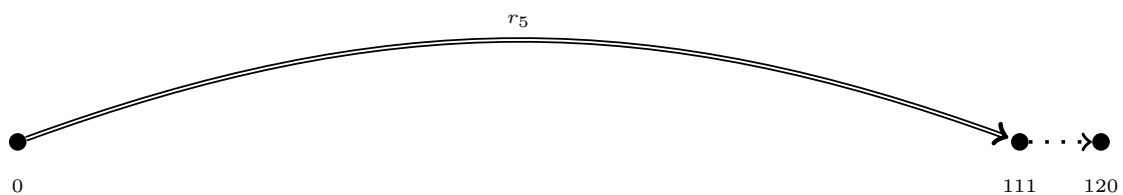
realiza duas rotas e no terceiro e último período (5.3c), um veículo realiza uma rota. Esta é uma solução válida para instância do Exemplo 4.



(a) Período 1.



(b) Período 2.



(c) Período 3.

Figura 5.3: Solução IRP para três períodos.

5.4 Resultados computacionais

Para avaliar o desempenho do modelo, foi utilizado um conjunto de instâncias adaptadas a partir das descritas em Moin et al. (2010).

Foram geradas 32 instâncias, sendo que em todas a duração de cada período do horizonte de planeamento (dia de trabalho) é de $W = 140$. Existem duas instâncias para cada combinação dos parâmetros: número de clientes $N \in \{10, 20, 40, 50\}$ e nú-

mero de períodos do horizonte de planeamento $\tau \in \{3, 5, 10, 15\}$. Com este conjunto de instâncias, foram efectuados diferentes testes, variando-se a capacidade dos veículos. Considera-se $C \in \{10, 13, 20\}$, de modo a que para todas as instâncias todas as procuras sejam superiores a respectivamente 20%, 15% e 10% da capacidade do veículo. Foi também considerada uma restrição adicional relativa ao número máximo de clientes a considerar numa rota. As Tabelas 5.3 e 5.4, este parâmetro assume valores que não restringem, ou restringem pouco, o número de clientes a considerar por rota. Nas tabelas que se seguem reportam os resultados dos testes computacionais para os quais o número máximo de clientes assume valores mais pequenos, $N_{max} = 3, 4, 5$.

Os testes computacionais foram efectuados utilizando um PC com processador i7 com 3.5 GHz e memória de 32 GB de RAM. As rotinas de optimização recorreram à versão 12.6.1 do CPLEX. O tempo limite para a resolução do modelo de programação inteira é de 900 segundos. Para o tempo total de geração do modelo (incluindo a geração das rotas) e resolução do mesmo, é fixado um tempo limite de 9500 segundos.

As Tabelas 5.3 – 5.12 reportam os resultados obtidos, tendo as respectivas colunas o seguinte significado:

inst: número da instância;

τ : número de períodos;

N : número de clientes;

W : duração de um dia de trabalho;

N_{max} : número máximo de clientes a visitar por rota;

t_r : tempo de geração das rotas;

t_m : tempo de execução do modelo de programação inteira (5.1) – (5.4);

t_{total} : tempo de execução total ($t_m + t_r$);

n_r : número de rotas geradas;

var: número de variáveis do modelo de programação inteira (5.1) – (5.4);

Lim_{inf} : melhor limite inferior;

Lim_{sup} : melhor limite superior;

gap %: intervalo de integralidade (em percentagem);

opt: “*” se a instância foi resolvida até ao valor óptimo;

As linhas \bar{x} e #* representam, respectivamente, as médias por coluna e o número de instâncias resolvidas até à optimalidade.

As instâncias para as quais o processo de geração das rotas ultrapassa o tempo limite não estão representadas nas tabelas. O símbolo “–” indica que não foi encontrada uma solução para a instância dentro do tempo limite.

Nos testes efectuados sem limite de número de clientes a visitar N_{max} e onde a capacidade do veículo é de 10 e 13 (Tabelas 5.3 e 5.4) foi possível resolver até à optimalidade respectivamente 12 e 10 instâncias das 18 e 12 em que o modelo foi gerado. Neste conjunto de testes, em que não existe uma restrição do número máximo de clientes a visitar por rota, só foi possível encontrar uma solução para instâncias com $N \leq 40$.

Como seria de esperar, o aumento da capacidade dos veículos dificulta a geração e resolução do modelo, uma vez que esta variação aumenta o número de rotas válidas e, conseqüentemente, o número de variáveis do mesmo. As instâncias para as quais não foi encontrada uma solução óptima apresentam contudo intervalos de optimalidade reduzidos, sendo o máximo destes valores, para todos os parâmetros considerados, igual a 8,85%.

A Tabela 5.13 sintetiza o número de instâncias resolvidas, de instâncias em que o modelo é gerado e do número de clientes máximo para o qual foi encontrada uma solução, para os diferentes valores de N_{max} e C (Tabelas 5.3 – 5.12). Pode observar-se que o aumento da capacidade dos veículos diminui consideravelmente o número de instâncias em que o modelo é gerado dentro do limite de tempo, enquanto que o número de soluções óptimas encontradas diminui menos significativamente. Na Tabela 5.14 estão descritos os valores médios das Tabelas 5.3–5.12 para as instâncias 1 a 12, resolvidas para todas as diferentes combinações de parâmetros. É possível verificar que em quase todos os casos as instâncias foram todas resolvidas até à optimalidade. Para as outras, o intervalo de optimalidade não ultrapassa os 1,5%.

Note-se que uma empresa que resolva este problema integrado terá apenas que gerar as rotas válidas uma única vez para cada conjunto de clientes considerado, sendo

que o que varia na prática são as procuras.

<i>inst</i>	τ	N	W	N_{max}	t_r	t_m	t_{total}	n_r	var	Lim_{inf}	Lim_{sup}	$gap\%$	opt
1	3	10	140	10	0,48	0,49	0,98	2864	4313	1021,00	1021	0,00	*
2	3	10	140	10	0,18	0,44	0,62	2204	3632	1062,00	1062	0,00	*
3	5	10	140	10	0,59	1,34	1,93	4845	8287	1710,00	1710	0,00	*
4	5	10	140	10	0,34	0,96	1,29	4133	7455	1734,00	1734	0,00	*
5	10	10	140	10	1,26	14,05	15,31	9389	17773	3419,00	3419	0,00	*
6	10	10	140	10	1,06	30,49	31,55	9500	17778	3410,00	3410	0,00	*
7	15	10	140	10	2,99	41,07	44,07	15153	28712	5068,00	5068	0,00	*
8	15	10	140	10	2,32	421,17	423,50	14644	27962	5081,00	5081	0,00	*
9	3	20	140	20	9,92	10,07	19,99	16783	24387	1839,00	1839	0,00	*
10	3	20	140	20	10,61	5,43	16,03	17493	25207	1828,00	1828	0,00	*
11	5	20	140	20	23,27	119,75	143,02	32974	49023	3003,00	3003	0,00	*
12	5	20	140	20	22,28	70,46	92,74	32751	49111	2991,00	2991	0,00	*
13	10	20	140	20	74,00	901,04	975,05	75056	114620	5825,90	6086	4,27	
14	10	20	140	20	68,31	905,68	974,00	70115	107867	5830,91	6000	2,82	
15	15	20	140	20	157,55	901,84	1059,40	118533	181898	8626,01	9029	4,46	
16	15	20	140	20	150,72	902,96	1053,69	113420	174695	8621,41	8867	2,77	
17	3	40	140	40	830,84	900,34	1731,37	329900	530391	2760,60	2835	2,62	
18	3	40	140	40	618,28	900,11	1518,62	289669	443338	2745,80	2819	2,60	
\bar{x}					109,72	340,43	450,18	64413	100914	3698,70	3767	1,09	
#*													12

Tabela 5.3: Testes computacionais com veículos de capacidade $C = 10$.

<i>inst</i>	τ	N	W	N_{max}	t_r	t_m	t_{total}	n_r	var	Lim_{inf}	Lim_{sup}	$gap\%$	opt
1	3	10	140	10	3,79	0,88	4,80	6076	7624	981,00	981	0,00	*
2	3	10	140	10	3,09	0,74	3,84	5566	7095	1013,00	1013	0,00	*
3	5	10	140	10	5,89	5,40	11,29	12003	16056	1642,00	1642	0,00	*
4	5	10	140	10	4,91	5,02	9,93	11353	15331	1671,00	1671	0,00	*
5	10	10	140	10	12,18	79,47	91,65	25749	36788	3274,00	3274	0,00	*
6	10	10	140	10	12,73	410,99	423,73	26428	37352	3268,00	3268	0,00	*
7	15	10	140	10	25,66	900,70	926,36	41917	60140	4819,01	4853	0,70	
8	15	10	140	10	25,08	900,86	925,93	42230	60247	4783,66	4851	1,39	
9	3	20	140	20	298,95	4,27	303,22	73539	84001	1618,00	1618	0,00	*
10	3	20	140	20	315,80	4,14	319,95	76834	87374	1618,00	1618	0,00	*
11	5	20	140	20	2707,79	23,78	2731,84	144110	168789	2651,00	2651	0,00	*
12	5	20	140	20	3047,98	21,30	3069,45	146436	171466	2652,00	2652	0,00	*
\bar{x}					538,65	196,46	735,17	51020	62689	2499,22	2508	0,17	
#*													10

Tabela 5.4: Testes computacionais com veículos de capacidade $C = 13$.

<i>inst</i>	τ	N	W	N_{max}	t_r	t_m	t_{total}	n_r	var	Lim_{inf}	Lim_{sup}	gap	$\%$	opt
1	3	10	140	3	0,07	0,40	0,48	1829	3272	1055,00	1055	0,00		*
2	3	10	140	3	0,05	0,26	0,31	1569	2991	1081,00	1081	0,00		*
3	5	10	140	3	0,12	1,20	1,32	3472	6899	1731,00	1731	0,00		*
4	5	10	140	3	0,10	0,79	0,89	3131	6438	1759,00	1759	0,00		*
5	10	10	140	3	0,34	6,62	6,96	7181	15525	3444,00	3444	0,00		*
6	10	10	140	3	0,35	3,69	4,04	7304	15542	3430,00	3430	0,00		*
7	15	10	140	3	0,75	6,54	7,29	11512	25006	5090,00	5090	0,00		*
8	15	10	140	3	0,72	36,71	37,43	11348	24601	5122,00	5122	0,00		*
9	3	20	140	3	0,45	3,55	4,00	8733	16287	1925,00	1925	0,00		*
10	3	20	140	3	0,47	4,78	5,25	8955	16614	1921,00	1921	0,00		*
11	5	20	140	3	1,29	164,52	165,81	18465	34437	3146,00	3146	0,00		*
12	5	20	140	3	1,36	151,41	152,78	18253	34519	3158,00	3158	0,00		*
13	10	20	140	3	5,23	900,02	905,26	43722	83094	6138,44	6242	1,66		
14	10	20	140	3	4,93	900,71	905,65	40836	78408	6176,98	6200	0,37		
15	15	20	140	3	10,84	901,09	911,93	68638	131705	9123,02	9482	3,79		
16	15	20	140	3	10,34	901,25	911,59	65580	126569	9146,65	9584	4,56		
17	3	40	140	3	7,14	902,59	909,75	65904	202736	2880,59	2988	3,59		
18	3	40	140	3	6,61	901,17	907,80	61015	169939	2864,28	2944	2,71		
19	5	40	140	3	19,34	901,92	921,28	128966	387819	4800,26	4925	2,53		
20	5	40	140	3	19,06	901,47	920,55	125927	351087	4813,56	4918	2,12		
21	10	40	140	3	71,27	911,83	983,15	287494	890385	9393,70	-			
22	10	40	140	3	70,94	900,27	971,25	285385	837611	9656,08	-			
23	15	40	140	3	153,82	900,36	1054,25	444794	1391948	14085,11	-			
24	15	40	140	3	152,26	900,34	1052,66	439931	1293258	14362,64	-			
25	3	50	140	3	15,58	900,45	916,04	123134	310295	3634,61	3736	2,71		
26	3	50	140	3	14,75	901,88	916,64	115636	272605	3628,34	3655	0,73		
27	5	50	140	3	42,14	900,93	943,10	241310	597347	6041,26	6628	8,85		
28	5	50	140	3	41,29	901,23	942,54	236511	555812	6062,91	6343	4,42		
29	10	50	140	3	161,41	900,47	1061,94	536211	1360972	11870,41	-			
30	10	50	140	3	160,51	900,35	1060,92	536154	1299136	12143,72	-			
31	15	50	140	3	6767,95	901,29	7669,83	830228	2120498	17787,12	-			
32	15	50	140	3	4007,77	901,29	4909,83	825945	2006080	18071,27	-			
\bar{x}					367,16	575,36	942,58	175159	458420	6423,22	4188	1,59		
#*														12

Tabela 5.5: Testes computacionais com veículos de capacidade $C = 10$ e número máximo de clientes a visitar $N_{max} = 3$.

<i>inst</i>	τ	N	W	N_{max}	t_r	t_m	t_{total}	n_r	var	Lim_{inf}	Lim_{sup}	$gap\%$	opt
1	3	10	140	4	0,21	0,67	0,89	2740	4189	1021,00	1021	0,00	*
2	3	10	140	4	0,13	0,36	0,50	2146	3574	1062,00	1062	0,00	*
3	5	10	140	4	0,33	1,15	1,48	4696	8138	1710,00	1710	0,00	*
4	5	10	140	4	0,29	0,92	1,21	4063	7385	1734,00	1734	0,00	*
5	10	10	140	4	0,83	12,59	13,43	9210	17594	3419,00	3419	0,00	*
6	10	10	140	4	0,86	18,34	19,21	9340	17618	3410,00	3410	0,00	*
7	15	10	140	4	1,81	41,04	42,85	14825	28384	5068,00	5068	0,00	*
8	15	10	140	4	1,67	198,01	199,68	14414	27732	5081,00	5081	0,00	*
9	3	20	140	4	4,09	14,95	19,05	15678	23282	1846,00	1846	0,00	*
10	3	20	140	4	4,29	7,67	11,97	16286	24000	1844,00	1844	0,00	*
11	5	20	140	4	10,20	688,71	698,91	31184	47233	3025,00	3025	0,00	*
12	5	20	140	4	10,07	98,31	108,38	30956	47316	3000,00	3000	0,00	*
13	10	20	140	4	34,13	903,93	938,06	71346	110910	5837,72	6041	3,37	
14	10	20	140	4	30,99	901,25	932,25	66469	104221	5826,48	5988	2,70	
15	15	20	140	4	69,78	903,63	973,43	112041	175406	8642,56	8997	3,94	
16	15	20	140	4	66,54	902,39	968,94	106992	168267	8632,41	9211	6,28	
17	3	40	140	4	82,26	900,21	982,49	228982	421909	2779,23	2884	3,63	
18	3	40	140	4	77,62	900,25	977,88	204953	353332	2766,16	2836	2,46	
19	5	40	140	4	206,96	900,58	1107,58	412644	759091	4677,01	4949	5,50	
20	5	40	140	4	197,77	903,43	1101,24	384463	671978	4664,40	4890	4,61	
21	3	50	140	4	223,27	903,04	1126,35	492534	753554	3492,68	3647	4,23	
22	3	50	140	4	212,76	902,44	1115,23	449415	662102	3484,47	3629	3,98	
\bar{x}					56,22	459,27	515,50	122063	201692	3773,78	3877	1,85	
#*													12

Tabela 5.6: Testes computacionais com veículos de capacidade $C = 10$ e número máximo de clientes a visitar $N_{max} = 4$.

<i>inst</i>	τ	N	W	N_{max}	t_r	t_m	t_{total}	n_r	var	Lim_{inf}	Lim_{sup}	gap	%	<i>opt</i>
1	3	10	140	5	0,50	0,48	0,98	2864	4313	1021,00	1021	0,00		*
2	3	10	140	5	0,18	0,47	0,65	2204	3632	1062,00	1062	0,00		*
3	5	10	140	5	0,58	1,34	1,92	4845	8287	1710,00	1710	0,00		*
4	5	10	140	5	0,32	0,99	1,32	4133	7455	1734,00	1734	0,00		*
5	10	10	140	5	1,21	13,72	14,93	9389	17773	3419,00	3419	0,00		*
6	10	10	140	5	1,02	28,85	29,87	9500	17778	3410,00	3410	0,00		*
7	15	10	140	5	2,82	38,38	41,20	15153	28712	5068,00	5068	0,00		*
8	15	10	140	5	2,06	406,25	408,31	14644	27962	5081,00	5081	0,00		*
9	3	20	140	5	9,70	9,87	19,58	16783	24387	1839,00	1839	0,00		*
10	3	20	140	5	10,49	5,23	15,72	17493	25207	1828,00	1828	0,00		*
11	5	20	140	5	22,80	118,66	141,47	32974	49023	3003,00	3003	0,00		*
12	5	20	140	5	22,72	70,31	93,04	32751	49111	2991,00	2991	0,00		*
13	10	20	140	5	73,28	901,06	974,34	75056	114620	5825,90	6088	4,31		
14	10	20	140	5	68,03	904,51	972,54	70115	107867	5830,91	6000	2,82		
15	15	20	140	5	157,00	901,77	1058,78	118533	181898	8626,01	9029	4,46		
16	15	20	140	5	150,01	902,96	1053,01	113420	174695	8621,41	8867	2,77		
17	3	40	140	5	1053,58	900,31	1954,20	329900	530391	2760,60	2835	2,62		
18	3	40	140	5	850,67	900,12	1751,16	289669	443338	2745,80	2819	2,60		
\bar{x}					134,83	339,18	474,06	64413	100914	3698,70	3767	1,09		
#*														12

Tabela 5.7: Testes computacionais com veículos de capacidade $C = 10$ e número máximo de clientes a visitar $N_{max} = 5$.

5.5 Conclusões

O problema de encaminhamento de veículos com múltiplas rotas e gestão integrada de inventários tem um grande interesse prático no domínio industrial mas, por outro lado, é bastante desafiante a nível da sua resolução. Assim, várias contribuições têm sido propostas na literatura, sendo estas maioritariamente abordagens heurísticas.

Neste capítulo, é proposto um modelo de fluxos em rede para o problema de encaminhamento de veículos com múltiplas rotas e gestão integrada de inventários, que é resolvido de forma exacta para um conjunto de instâncias adaptadas da literatura. O modelo foi capaz de resolver instâncias com até 50 clientes e 15 períodos de tempo, em tempos computacionais razoáveis.

<i>inst</i>	τ	N	W	N_{max}	t_r	t_m	t_{total}	n_r	var	Lim_{inf}	Lim_{sup}	$gap\%$	opt
1	3	10	140	3	0,09	0,49	0,58	2770	4312	1036,00	1036	0,00	*
2	3	10	140	3	0,10	0,42	0,52	2608	4131	1042,00	1042	0,00	*
3	5	10	140	3	0,24	1,20	1,45	6259	10297	1699,00	1699	0,00	*
4	5	10	140	3	0,21	1,91	2,12	6024	9987	1705,00	1705	0,00	*
5	10	10	140	3	0,75	14,02	14,78	14383	25377	3362,00	3362	0,00	*
6	10	10	140	3	0,75	9,19	9,94	14696	25575	3334,00	3334	0,00	*
7	15	10	140	3	1,61	28,90	30,52	23526	41674	4980,00	4980	0,00	*
8	15	10	140	3	1,63	32,31	33,94	23806	41748	4960,00	4960	0,00	*
9	3	20	140	3	1,15	3,47	4,63	17140	27282	1879,00	1879	0,00	*
10	3	20	140	3	1,19	3,34	4,54	17484	27699	1879,00	1879	0,00	*
11	5	20	140	3	3,67	176,91	180,58	39944	64055	3097,00	3097	0,00	*
12	5	20	140	3	3,66	274,12	277,78	40147	64587	3093,00	3093	0,00	*
13	10	20	140	3	13,56	901,27	914,84	95574	157000	6032,15	6108	1,24	
14	10	20	140	3	13,41	901,92	915,33	94838	156127	6010,90	6153	2,31	
15	15	20	140	3	29,33	903,20	932,54	152855	252043	8954,93	9148	2,11	
16	15	20	140	3	28,88	901,89	930,78	152036	251113	8947,47	9130	2,00	
17	3	40	140	3	15,70	900,26	915,98	136919	370124	2517,40	2604	3,33	
18	3	40	140	3	14,96	901,16	916,13	128022	320535	2522,09	2591	2,66	
19	5	40	140	3	46,51	901,86	948,42	299511	794577	4180,97	4514	7,38	
20	5	40	140	3	46,10	901,35	947,48	294856	736939	4174,26	4363	4,33	
21	10	40	140	3	190,60	901,92	1092,62	730541	2000995	8085,85	-		
22	10	40	140	3	186,13	900,50	1086,73	700912	1861751	8274,42	-		
23	15	40	140	3	6246,40	901,65	7148,89	1150988	3173057	12087,15	-		
24	15	40	140	3	3901,78	939,71	4842,60	1106997	2938779	12280,62	-		
25	3	50	140	3	34,12	901,86	936,15	258923	588951	3169,42	3211	1,29	
26	3	50	140	3	32,59	900,10	932,72	245772	530923	3175,76	3237	1,89	
27	5	50	140	3	103,59	900,58	1004,23	570528	1280137	5247,91	-		
28	5	50	140	3	103,24	900,43	1003,73	563319	1213859	5243,98	-		
29	10	50	140	3	8543,64	901,51	9447,06	1385107	3186548	10179,28	-		
30	10	50	140	3	6606,83	902,12	7510,59	1345921	3006965	10371,38	-		
\bar{x}					872,41	560,32	1432,94	320747	772238	5117,40	3778	1,30	
#*													12

Tabela 5.8: Testes computacionais com veículos de capacidade $C = 13$ e número máximo de clientes a visitar $N_{max} = 3$.

<i>inst</i>	τ	N	W	N_{max}	t_r	t_m	t_{total}	n_r	var	Lim_{inf}	Lim_{sup}	$gap\%$	opt
1	3	10	140	4	0,76	0,67	1,57	5254	6802	1002,00	1002	0,00	*
2	3	10	140	4	0,61	0,68	1,29	4849	6378	1029,00	1029	0,00	*
3	5	10	140	4	1,23	3,10	4,33	10741	14794	1659,00	1659	0,00	*
4	5	10	140	4	1,08	4,04	5,12	10178	14156	1691,00	1691	0,00	*
5	10	10	140	4	3,29	67,89	71,18	23404	34443	3293,00	3293	0,00	*
6	10	10	140	4	3,28	36,47	39,76	23938	34862	3288,00	3288	0,00	*
7	15	10	140	4	7,11	900,72	907,84	38155	56378	4860,10	4881	0,43	
8	15	10	140	4	6,94	375,05	382,00	38382	56399	4871,00	4871	0,00	*
9	3	20	140	4	17,02	590,79	607,81	48322	58784	1768,00	1768	0,00	*
10	3	20	140	4	17,71	900,47	918,18	50164	60704	1746,94	1770	1,30	
11	5	20	140	4	43,31	900,04	943,36	102692	127371	2755,41	2830	2,64	
12	5	20	140	4	44,09	900,11	944,21	104069	129099	2757,76	2840	2,90	
13	10	20	140	4	144,46	902,11	1046,58	232015	294549	5351,26	5502	2,74	
14	10	20	140	4	143,79	907,79	1051,60	229735	292118	5362,14	5474	2,04	
15	15	20	140	4	1215,88	902,39	2118,78	367256	468095	7913,83	8254	4,12	
16	15	20	140	4	1335,41	900,69	2236,56	364695	465409	7918,82	8128	2,57	
17	3	40	140	4	573,81	901,38	1475,55	830010	1267666	2350,84	2454	4,20	
18	3	40	140	4	554,88	1155,80	1711,07	743163	1082028	2369,34	2447	3,17	
19	10	20	140	4	143,71	902,02	1045,75	232015	294549	5351,26	5502	2,74	
20	10	20	140	4	142,75	906,85	1049,61	229735	292118	5362,17	5474	2,04	
\bar{x}					220,06	607,95	828,11	184439	252835	3635,04	3708	1,55	
#*													8

Tabela 5.9: Testes computacionais com veículos de capacidade $C = 13$ e número máximo de clientes a visitar $N_{max} = 4$.

<i>inst</i>	τ	N	W	N_{max}	t_r	t_m	t_{total}	n_r	var	Lim_{inf}	Lim_{sup}	$gap\%$	opt
1	3	10	140	5	2,41	0,78	3,26	6033	7581	981,00	981	0,00	*
2	3	10	140	5	2,03	0,79	2,82	5522	7051	1013,00	1013	0,00	*
3	5	10	140	5	4,02	4,42	8,44	11941	15994	1642,00	1642	0,00	*
4	5	10	140	5	3,54	3,92	7,46	11295	15273	1671,00	1671	0,00	*
5	10	10	140	5	9,45	35,75	45,20	25652	36691	3274,00	3274	0,00	*
6	10	10	140	5	9,64	65,19	74,83	26310	37234	3268,00	3268	0,00	*
7	15	10	140	5	19,32	744,39	763,72	41758	59981	4847,00	4847	0,00	*
8	15	10	140	5	19,12	900,85	919,97	42054	60071	4800,57	4850	1,02	
9	3	20	140	5	106,73	3,69	110,41	70003	80465	1619,00	1619	0,00	*
10	3	20	140	5	112,35	4,30	116,65	73105	83645	1619,00	1619	0,00	*
11	5	20	140	5	250,57	10,59	261,16	139173	163852	2652,00	2652	0,00	*
12	5	20	140	5	258,04	41,56	299,62	141486	166516	2652,00	2652	0,00	*
\bar{x}					66,44	151,35	217,80	49528	61196	2503,21	2507	0,08	
#*													11

Tabela 5.10: Testes computacionais com veículos de capacidade $C = 13$ e número máximo de clientes a visitar $N_{max} = 5$.

<i>inst</i>	τ	N	W	N_{max}	t_r	t_m	t_{total}	n_r	var	Lim_{inf}	Lim_{sup}	$gap\%$	opt
1	3	10	140	3	0,16	0,72	0,98	4093	5679	1022,00	1022	0,00	*
2	3	10	140	3	0,16	0,57	0,73	4052	5638	1039,00	1039	0,00	*
3	5	10	140	3	0,72	4,05	4,77	16811	22046	1672,00	1672	0,00	*
4	5	10	140	3	0,64	3,20	3,85	15730	20924	1679,00	1679	0,00	*
5	10	10	140	3	3,48	55,49	58,98	56198	76928	3292,00	3292	0,00	*
6	10	10	140	3	3,36	28,48	31,85	56279	76727	3263,00	3263	0,00	*
7	15	10	140	3	8,07	89,40	97,49	98304	136206	4871,00	4871	0,00	*
8	15	10	140	3	7,80	130,77	138,58	97998	135488	4858,00	4858	0,00	*
9	3	20	140	3	2,53	5,51	8,04	30154	43510	1879,00	1879	0,00	*
10	3	20	140	3	2,55	5,34	7,90	30314	43729	1879,00	1879	0,00	*
11	5	20	140	3	12,22	54,38	66,61	113715	158024	3084,00	3084	0,00	*
12	5	20	140	3	12,42	54,14	66,57	114391	159243	3075,00	3075	0,00	*
13	10	20	140	3	65,30	902,66	967,99	397802	551918	6017,73	6121	1,69	
14	10	20	140	3	61,81	900,62	962,46	378445	527341	6001,78	6130	2,09	
15	15	20	140	3	151,56	900,49	1052,12	692790	967516	8926,57	9111	2,02	
16	15	20	140	3	144,78	907,93	1052,79	664424	927996	8910,75	9193	3,07	
17	3	40	140	3	32,69	904,26	936,98	276003	676815	2265,10	2337	3,08	
18	3	40	140	3	31,86	903,71	935,62	262472	624801	2242,53	2306	2,75	
19	5	40	140	3	153,54	900,66	1054,36	929557	2262473	3707,87	-	-	
20	5	40	140	3	151,47	901,54	1053,13	907747	2102472	3718,66	-	-	
\bar{x}					42,36	382,70	425,09	257364	476274	3670,20	3712	0,82	
#*													12

Tabela 5.11: Testes computacionais com veículos de capacidade $C = 20$ e número máximo de clientes a visitar $N_{max} = 3$.

<i>inst</i>	τ	N	W	N_{max}	t_r	t_m	t_{total}	n_r	var	Lim_{inf}	Lim_{sup}	$gap\%$	opt
1	3	10	140	4	3,08	1,71	4,90	13705	15297	994,00	994	0,00	*
2	3	10	140	4	2,80	2,41	5,21	12989	14581	1026,00	1026	0,00	*
3	5	10	140	4	12,35	19,66	32,02	52329	57579	1620,00	1620	0,00	*
4	5	10	140	4	10,58	13,25	23,83	46617	51826	1644,00	1644	0,00	*
5	10	10	140	4	41,13	84,32	125,46	150651	171436	3155,00	3155	0,00	*
6	10	10	140	4	41,24	82,18	123,43	150374	170877	3151,00	3151	0,00	*
7	15	10	140	4	88,15	219,85	308,03	258498	296505	4673,00	4673	0,00	*
8	15	10	140	4	85,63	771,04	856,68	255786	293381	4677,00	4677	0,00	*
9	3	20	140	4	76,67	900,21	976,90	177872	192875	1700,65	1770	3,92	
10	3	20	140	4	78,02	900,53	978,54	181994	197101	1699,46	1770	3,99	
11	5	20	140	4	349,67	901,81	1251,73	601819	650927	2705,01	2865	5,58	
12	5	20	140	4	362,53	901,48	1264,31	606822	656670	2721,19	2828	3,78	
\bar{x}					95,99	399,87	495,92	209121	230755	2480,53	2514	1,44	
#*													8

Tabela 5.12: Testes computacionais com veículos de capacidade $C = 20$ e número máximo de clientes a visitar $N_{max} = 4$

N_{max}	$C = 10$	$C = 13$	$C = 20$
10	12/18/40	10/12/20	-
3	12/32/50	12/30/50	12/20/40
4	12/22/50	8/20/20	8/12/20
5	12/18/40	11/12/20	-

Tabela 5.13: Síntese das instâncias resolvidas até à optimalidade / instâncias em que o modelo é gerado / máximo de clientes.

C	N_{max}	\bar{t}_r	\bar{t}_m	\bar{t}_{total}	\bar{n}_r	$\bar{v}_{\bar{a}r}$	$\bar{L}im_{inf}$	$\bar{L}im_{sup}$	$gap\%$	$\#opt$
10	20	6,28	59,64	65,92	13561,08	21970,00	2680,50	2680,50	0	12
13	20	538,65	196,46	735,17	51020,08	62688,58	2499,22	2507,67	0,17	10
10	3	0,51	31,71	32,21	8479,33	16844,25	2738,50	2738,50	0	12
10	4	2,90	90,23	93,13	12961,50	21370,42	2685,00	2685,00	0	12
10	5	6,20	57,88	64,08	13561,08	21970,00	2680,50	2680,50	0	12
13	3	1,25	45,52	46,78	17398,92	28893,67	2672,17	2672,17	0	12
13	4	12,20	390,00	402,22	38345,67	50014,17	2560,10	2576,83	0,61	8
13	5	66,44	151,35	217,80	49527,67	61196,17	2503,21	2507,33	0,09	11
20	3	4,51	36,00	40,53	53169,92	73678,50	2634,42	2634,42	0	12
20	4	95,99	399,87	495,92	209121,33	230754,58	2480,53	2514,42	1,44	8

Tabela 5.14: Valores médios das Tabelas 5.3–5.12 para as instâncias 1-12.

Capítulo 6

Conclusão geral e trabalho futuro

Conteúdo

6.1	Conclusão geral	135
6.2	Trabalho futuro	138

6.1 Conclusão geral

Com o objectivo de contribuir para a resolução de problemas integrados em cadeias de abastecimento, esta tese propõe vários modelos e métodos para a resolução de alguns problemas específicos. Os métodos de programação inteira, tanto exactos como heurísticos, têm vindo a ser cada vez mais usados por parte da indústria para melhorar a sua eficiência produtiva. Problemas como o de transporte, gestão de inventário, corte ou escalonamento da produção, que anteriormente eram resolvidos separadamente, são cada vez mais resolvidos de forma integrada. Esta junção dos problemas tem como objectivo uma optimização global. Com efeito, soluções de melhor qualidade do ponto de vista prático são encontradas quando os problemas são resolvidos simultaneamente. Com esta integração surge a necessidade de abordar os problemas de uma forma diferente, uma vez que a complexidade de resolução aumenta consideravelmente do ponto de vista computacional.

Tendo como objectivo contribuir para a resolução destes problemas integrados,

foi considerado neste estudo o problema combinado de corte e escalonamento. Este problema caracteriza-se por um conjunto de objectos que devem ser cortados num determinado número de rolos. Estes objectos têm um prazo de entrega que, se ultrapassado, conduz a uma penalidade. O objectivo é reduzir, por um lado, o número de rolos a utilizar e, por outro, as penalidades associadas aos atrasos. Nesta tese foram propostos vários modelos de programação inteira originais para este problema, assim como métodos de resolução exacta e heurísticas. Nos modelos de programação inteira, destaca-se o modelo pseudo-polinomial de fluxos em rede que com o auxílio de heurísticas permitiu obter resultados satisfatórios para instâncias de referência da literatura. Foi também proposto um modelo de afectação reforçado com planos de corte obtidos através de funções duais válidas. Os vários modelos apresentados nesta tese contribuem efectivamente para resolução eficiente deste problema integrado de corte e empacotamento.

Nesta tese, foi também abordado o problema integrado de planeamento e escalonamento, muito utilizado na indústria para o escalonamento de tarefas em máquinas. Este problema caracteriza-se por um conjunto de máquinas idênticas e limitadas pela quantidade, capacidade e por um conjunto de tarefas a realizar nas mesmas. Estas tarefas possuem data de entrega e podem ser realizadas antes ou depois da mesma, mas nunca antes da sua data de lançamento. A realização das tarefas fora do prazo conduz também aqui a uma penalidade. O objectivo é, por um lado, usar as máquinas da forma mais eficiente, e por outro reduzir as penalidades associadas aos atrasos. Para este problema, foi proposto um novo modelo pseudo-polinomial de fluxos em rede. Os testes realizados permitiram concluir que o modelo tem um melhor desempenho do que os modelos descritos na literatura. Os testes efectuados foram realizados nas mesmas condições dos descritos na literatura, tanto do ponto de vista das características do computador usado como do *software* de optimização. De notar que não foram usados cortes para fortalecer o modelo. No modelo de fluxos apresentado foram também utilizadas estratégias de redução de vértices e arcos, recorrendo-se ao conceito de *grid points*. Embora este modelo tenha obtido excelentes resultados, algumas instâncias, devido a características como número de itens ou dimensão dos mesmos, mantiveram-se com um intervalo de optimalidade elevado. Para essas instâncias, fo-

ram propostas várias estratégias heurísticas com o objectivo de reduzir este intervalo. Estas heurísticas assentaram na fixação de variáveis do problema, tendo em conta diferentes metodologias e ordenações das tarefas. Estas ordenações, segundo os critérios, influenciam os resultados obtidos. Através destas heurísticas foi possível reduzir consideravelmente o intervalo de optimalidade para estas instâncias difíceis.

Por último, nesta tese, foi abordado um problema pouco estudado na literatura, o problema de encaminhamento de veículos com múltiplas rotas e gestão integrada de inventários. Sendo este também um problema integrado, caracteriza-se por um conjunto de veículos situados num armazém que têm de entregar determinados artigos a um conjunto de clientes, em diferentes períodos, de modo a satisfazer as suas necessidades. Estes veículos podem realizar várias rotas no mesmo dia de trabalho, podendo nesse período entregar os itens aos clientes para os períodos seguintes. Nesse caso, aplicam-se custos de armazenamento que são proporcionais ao tempo que os itens são mantidos em armazém. O objectivo deste problema contempla o custo de transporte e o custo de armazenamento, que deverão ser minimizados em conjunto. Para resolver este problema, foi proposto um novo modelo pseudo-polinomial de fluxos em rede, com o qual se obtiveram resultados promissores.

O modelo assenta na enumeração completa das rotas antes da geração do mesmo. As variáveis do modelo estão associadas a estas rotas, com início e fim no armazém. Os testes computacionais demonstram também que não só o número de clientes pode influenciar na obtenção de uma solução num curto espaço de tempo mas também o número de veículos, as suas capacidades de carga, assim como o número de períodos do horizonte de planeamento e, conseqüentemente, as procuras para esses períodos.

Uma vez que na indústria existe cada vez mais a necessidade de integrar problemas de forma a obter uma melhor solução global, várias abordagens têm surgido na literatura com o objectivo de resolver problemas integrados, de forma eficiente. Estes algoritmos devem não só fornecer uma solução de boa qualidade, como também disponibilizá-la num curto espaço de tempo. Nesse sentido, esta tese contribui, através de diversas abordagens, para a resolução de três problemas integrados nas cadeias de abastecimento. Foram propostos métodos de programação inteira exactos e métodos heurísticos que têm a capacidade de fornecer uma solução de boa qualidade num curto

espaço de tempo.

6.2 Trabalho futuro

O trabalho realizado nesta tese permitiu resolver instâncias de dimensão média para três problemas integrados que ocorrem em cadeias de abastecimento. Os modelos e métodos propostos permitem fornecer soluções válidas e de boa qualidade em tempos razoáveis. Os resultados foram globalmente promissores, o que permite validar a qualidade e o potencial destas abordagens. Contudo, ainda existe espaço para desenvolvimentos futuros nesta área. Abaixo, são identificadas algumas pistas possíveis de investigação relacionadas em particular com cada um dos problemas abordados.

Para o problema combinado de corte e escalonamento:

- Com o aumento do número de itens aumenta a dimensão dos modelos associados e a dificuldade em resolvê-los. Assim, seria conveniente estudar métodos de redução de arcos e vértices do modelo de fluxos. Alguns dos métodos poderão ser a redução parcial ou total dos arcos de desperdício, utilização dos *grid points* ou a aplicação dos métodos de redução descritos em Brandão e Pedroso (2016).
- A agregação de períodos de tempo reduz a dimensão do problema, possibilitando a obtenção de soluções mais rapidamente. Desta forma, seria útil estudar a agregação de períodos de tempo, iniciando o problema com um conjunto reduzido dos mesmos, e adicionando posteriormente períodos adicionais ao problema de forma iterativa.
- Este problema possui a particularidade da penalidade associada ao atraso de um item ser calculada a partir do último item a ser cortado. Assim, o número máximo de fluxos (ou períodos agregados) será do mesmo número que os pedidos $i \in N$. Neste caso, a dimensão dos períodos agregados será variável. Desta forma, explorando esta particularidade, talvez seja possível desenvolver um modelo monolítico que procure na sua solução a dimensão dos períodos agregados.

Problema integrado de planeamento e escalonamento:

- O modelo de programação inteira aqui proposto supera os descritos na literatura. Contudo, para algumas instâncias a sua resolução continua a ser um desafio. A adição de cortes ao modelo pode contribuir para o seu reforço, pelo que deveria ser explorada no futuro;
- Uma solução para este modelo poderá ser obtida mais rapidamente se, através de uma heurística construtiva, for possível encontrar uma solução inicial, e posteriormente esta ser considerada numa fase inicial da resolução do modelo. Este processo pode conduzir à eliminação de parte significativa dos nodos da árvore de pesquisa aquando da resolução do modelo através do método de partição e avaliação sucessivas;
- No modelo proposto, a ordem em que as tarefas são seleccionadas na formação do grafo parece influenciar o seu número de vértices. A redução do número de vértices e, conseqüentemente, o número de variáveis, reduz a sua complexidade e influencia positivamente a procura de soluções. Nesse sentido, era desejável estudar a forma como a ordenação influencia o número de vértices do grafo.

Problema de encaminhamento de veículos com múltiplas rotas e gestão integrada de inventários:

- O modelo proposto nesta tese não foi reforçado usando planos de corte. A adição de cortes poderá conduzir a um modelo com intervalos de integralidade mais pequenos, levando assim a que mais instâncias de média e grande dimensão possam ser resolvidos de forma exacta;
- Neste trabalho, foi estudada a influência da capacidade dos veículos e do número de clientes a visitar na solução obtida. O estudo de outros parâmetros poderá conduzir a uma melhor compreensão do problema, como é o caso por exemplo da distância máxima a percorrer em cada rota;
- Neste problema, foi assumido que a entrega dos itens aos clientes só poderia ocorrer para o período t em que o veículo faz a entrega, ou para períodos seguintes até t' de forma que a mercadoria tenha que ser entregue na totalidade

para os períodos t a t' . Uma vez que a entrega tem que ser efectuada toda do período t a t' , obrigando a haver custos de armazenagem do período $t + 1$ até t' , seria conveniente estudar a influência da possibilidade dos veículos realizarem entregas sem restrição de quantidade a entregar ao cliente. Isso possibilitaria a redução do custo de armazenagem, uma vez que seria possível a entrega parcial da procura para períodos posteriores.

Referências

- M. K. I. Abdul Rahim, Y. Zhong, E.-H. Aghezzaf, e T. Aouam. Modelling and solving the multiperiod inventory-routing problem with stochastic stationary demand rates. International Journal of Production Research, 52:4351–4363, 2014.
- E. H. Aghezzaf, B. Raa, e H. Van Landeghem. Modeling inventory routing problems in supply chains of high consumption products. European Journal of Operational Research, 169:1048–1063, 2006.
- F. Al-Khayyal e S.-J. Hwang. Inventory constrained maritime routing and scheduling for multi-commodity liquid bulk, part i: Applications and model. European Journal of Operational Research, 176:106–130, 2007.
- F. Alonso, J. M. Alvarez, e E. J. Beasley. A tabu search algorithm for the periodic vehicle routing problem with multiple vehicle trips and accessibility restrictions. Journal of the Operational Research Society, 59:963–76, 2008.
- C. Alves e J. Valério de Carvalho. A stabilized branch-and-price-and-cut algorithm for the multiple length cutting stock problem. Computers & Operations Research, 35:1315–1328, 2008.
- A. C. S. Amaro e A. P. F. D. Barbosa-Póvoa. Planning and scheduling of industrial supply chains with reverse flows: A real pharmaceutical case study. Computers & Chemical Engineering, 32:2606–25, 2008.
- H. Andersson, A. Hoff, M. Christiansen, G. Hasle, e A. Løkketangen. Industrial aspects and literature survey: Combined inventory management and routing. Computers and Operations Research, 37:1515–1536, 2010.

- C. Arbib e F. Marinelli. Integrating process optimization and inventory planning in cutting-stock with skiving option: An optimization model and its application. European Journal of Operational Research, 163:617–630, 2005.
- C. Arbib e F. Marinelli. On cutting stock with due dates. Omega-International Journal of Management Science, 46:11–20, 2014.
- C. Arbib, F. Marinelli, e F. Pezzella. An lp-based tabu search for batch scheduling in a cutting process with finite buffers. International Journal of Production Economics, 136:287–96, 2012.
- C. Archetti e M. G. Speranza. A survey on matheuristics for routing problems. EURO Journal on Computational Optimization, 2:223–46, 2014.
- C. Archetti e M. G. Speranza. The inventory routing problem: the value of integration. International Transactions in Operational Research, 23:393–407, 2016.
- C. Archetti, L. Bertazzi, G. Laporte, e M. G. Speranza. A branch-and-cut algorithm for a vendor-managed inventory-routing problem. Transportation Science, 41:382–391, 2007.
- C. Archetti, L. Bertazzi, A. Hertz, e M. G. Speranza. A hybrid heuristic for an inventory routing problem. INFORMS Journal on Computing, 24:101–16, 2011.
- C. Archetti, N. Bianchessi, S. Irnich, e M. G. Speranza. Formulations for an inventory routing problem. International Transactions in Operational Research, 21:353–74, 2014.
- R. Armstrong, S. Gao, e L. Lei. A zero-inventory production and distribution problem with a fixed customer sequence. Annals of Operations Research, 159:395–414, 2008.
- N. Azi, M. Gendreau, e J.-Y. Potvin. An exact algorithm for a single-vehicle routing problem with time windows and multiple routes. European Journal of Operational Research, 178:755–66, 2007.

- N. Azi, M. Gendreau, e J.-Y. Potvin. An exact algorithm for a vehicle routing problem with time windows and multiple use of vehicles. European Journal of Operational Research, 202:756–63, 2010.
- B. S. Baker e E. G. Coffman. Mutual exclusion scheduling. Theoretical Computer Science, 162:225–243, 1996.
- R. Baldacci, A. Mingozzi, e R. Roberti. Recent exact algorithms for solving the vehicle routing problem under capacity and time window constraints. European Journal of Operational Research, 218:1–6, 2012.
- M. Baldea e I. Harjunoski. Integrated production scheduling and process control: A systematic review. Computers & Chemical Engineering, 71:377–390, 2014.
- M. H. Bassett, J. F. Pekny, e G. V. Reklaitis. Decomposition techniques for the solution of large-scale scheduling problems. AIChE Journal, 42:3373–87, 1996.
- G. Belov. Problems, models and algorithms in one-and two-dimensional cutting. Thesis, 2003.
- G. Belov e G. Scheithauer. A branch-and-cut-and-price algorithm for one-dimensional stock cutting and two-dimensional two-stage cutting. European Journal of Operational Research, 171(1):85–106, 2006a.
- G. Belov e G. Scheithauer. A branch-and-cut-and-price algorithm for one-dimensional stock cutting and two-dimensional two-stage cutting. European Journal of Operational Research, 171:85–106, 2006b.
- M. Bendraouche e M. Boudhar. Scheduling jobs on identical machines with agreement graph. Computers & Operations Research, 39:382–390, 2012.
- J. A. Bennell, L. S. Lee, e C. N. Potts. A genetic algorithm for two-dimensional bin packing with due dates. International Journal of Production Economics, 145: 547–560, 2013.

- A. Bensmaine, M. Dahane, e L. Benyoucef. A new heuristic for integrated process planning and scheduling in reconfigurable manufacturing systems. International Journal of Production Research, 52:3583–3594, 2014.
- L. Bertazzi e M. G. Speranza. Inventory routing problems: an introduction. EURO Journal on Transportation and Logistics, 1:307–326, 2012.
- L. Bertazzi, G. Paletta, e M. G. Speranza. Minimizing the total cost in an integrated vendor - managed inventory system. Journal of Heuristics, 11:393–419, 2005.
- A. Bixby, B. Downs, e M. Self. A scheduling and capable-to-promise application for swift & company. Interfaces, 36:69–86, 2006.
- N. Braga, C. Alves, R. Macedo, e J. Valério de Carvalho. A Model-Based Heuristic for the Combined Cutting Stock and Scheduling Problem, volume 9156 of Lecture Notes in Computer Science, páginas 490–505. Springer International Publishing, 2015.
- N. Braga, C. Alves, R. Macedo, e J. M. Valério de Carvalho. Combined cutting stock and scheduling: a matheuristic approach. International Journal of Innovative Computing and Applications, 7(3):135 – 146, 2016a.
- N. Braga, C. Alves, e J. Valério de Carvalho. Exact solution of combined cutting stock and scheduling problems, volume 682 of Lecture Notes in Economics and Mathematical Systems, capítulo 17, páginas 131–39. Springer International Publishing, 2016b.
- F. Brandão e J. P. Pedroso. Bin packing and related problems: General arc-flow formulation with graph compression. Computers & Operations Research, 69:56–67, 2016.
- J. C. S. Brandão e A. Mercer. The multi-trip vehicle routing problem. Journal of the Operational Research Society, 49:799–805, 1998.
- J. Caceres-Cruz, P. Arias, D. Guimarans, D. Riera, e A. A. Juan. Rich vehicle routing problem. ACM Computing Surveys, 47:1–28, 2014.

- H. Cambazard e B. OSullivan. Propagating the Bin Packing Constraint Using Linear Programming, páginas 129–136. Springer, Berlin, Heidelberg, 2010.
- J. Carlier e E. Néron. A new lp-based lower bound for the cumulative scheduling problem. European Journal of Operational Research, 127:363–82, 2000.
- J. Carlier, F. Clautiaux, e A. Moukrim. New reduction procedures and lower bounds for the two-dimensional bin packing problem with fixed orientation. Computers & Operations Research, 34:2223–2250, 2007.
- Y.-C. Chang e C.-Y. Lee. Machine scheduling with job delivery coordination. European Journal of Operational Research, 158:470–487, 2004.
- B. Chen, C. N. Potts, e G. J. Woeginger. A review of machine scheduling: complexity, algorithms and approximability, páginas 1493–1641. Springer US, Boston, MA, 1999.
- C. S. Chen, S. M. Lee, e Q. S. Shen. An analytical model for the container loading problem. European Journal of Operational Research, 80:68–76, 1995.
- Z. L. Chen e G. L. Vairaktarakis. Integrated scheduling of production and distribution operations. Management Science, 51:614–628, 2005.
- F. Clautiaux, C. Alves, e J. Valério de Carvalho. A survey of dual-feasible and superadditive functions. Annals of Operations Research, 179:317–342, 2010.
- L. C. Coelho e G. Laporte. A branch-and-cut algorithm for the multi-product multi-vehicle inventory-routing problem. International Journal of Production Research, 51:7156–69, 2013.
- L. C. Coelho, J.-F. Cordeau, e G. Laporte. Thirty years of inventory routing. Transportation Science, 48:1–19, 2014.
- J.-F. Cordeau, D. Laganà, R. Musmanno, e F. Vocaturro. A decomposition-based heuristic for the multiple-product inventory-routing problem. Computers & Operations Research, 55:153–66, 2015.

- A. Şen e K. Bülbül. A survey on multi trip vehicle routing problem. VI. International Logistics and Supply Chain Congress, Universidade Bilgi, Istambul, Turquia, 2008.
- V. M. Dalfard e G. Mohammadi. Two meta-heuristic algorithms for solving multi-objective flexible job-shop scheduling with parallel machine and maintenance constraints. Computers & Mathematics with Applications, 64:2111–2117, 2012.
- S. Dauzère-Pérès, A. Nordli, A. Olstad, K. Haugen, U. Koester, M. Per Olav, G. Teistklub, e A. Reistad. Omya hustadmarmor optimizes its supply chain for delivering calcium carbonate slurry to european paper manufacturers. Interfaces, 37:39–51, 2007.
- Z. Degraeve e M. Peeters. Optimal integer solutions to indums: Part 2, benchmark results. INFORMS Journal on Computing, 15:58–81, 2003.
- M. Dell’Amico, J. C. D. Diaz, e M. Iori. The bin packing problem with precedence constraints. Operations Research, 60:1491–1504, 2012.
- M. Delorme, M. Iori, e S. Martello. Bin packing and cutting stock problems: Mathematical models and exact algorithms. European Journal of Operational Research, 255:1–20, 2016.
- B. Detienne. A mixed integer linear programming approach to minimize the number of late jobs with and without machine availability constraints. European Journal of Operational Research, 235:540–552, 2014.
- M. Durbin e K. Hoffman. The dance of the thirty-ton trucks: Dispatching and scheduling in a dynamic environment. Operations Research, 56:3–19, 2008.
- H. Dyckhoff. A new linear programming approach to the cutting stock problem. Operations Research, 29:1092–1104, 1981.
- H. Dyckhoff. A typology of cutting and packing problems. European Journal of Operational Research, 44:145–159, 1990.
- H. El-Rewini e T. G. Lewis. Scheduling parallel program tasks onto arbitrary target machines. Journal of Parallel and Distributed Computing, 9:138–153, 1990.

- M. Erdirik-Dogan e I. E. Grossmann. Simultaneous planning and scheduling of single-stage multi-product continuous plants with parallel lines. Computers & Chemical Engineering, 32:2664–2683, 2008.
- S. P. Fekete e J. Schepers. New classes of fast lower bounds for bin packing problems. Mathematical Programming, 91:11–31, 2001.
- M. R. Garey e D. S. Johnson. Strong np-completeness results - motivation, examples, and implications. Journal of the Acm, 25:499–508, 1978.
- H. N. Geismar, G. Laporte, L. Lei, e C. Sriskandarajah. The integrated production and transportation scheduling problem for a product with a short lifespan. Inform Journal on Computing, 20:21–33, 2008.
- N. F. Giannelos e M. C. Georgiadis. Scheduling of cutting-stock processes on multiple parallel machines. Chemical Engineering Research & Design, 79:747–53, 2001.
- P. C. Gilmore e R. E. Gomory. A linear-programming approach to the cutting-stock problem. Operations Research, 9:849–859, 1961.
- P. C. Gilmore e R. E. Gomory. A linear-programming approach to the cutting stock problem - part ii. Operations Research, 11:863–888, 1963.
- I. E. Grossmann e K. C. Furman. Challenges in Enterprise Wide Optimization for the Process Industries, páginas 3–59. Springer US, Boston, MA, 2009.
- R. Grønhaug, M. Christiansen, G. Desaulniers, e J. Desrosiers. A branch-and-price method for a liquefied natural gas inventory routing problem. Transportation Science, 44:400–415, 2010.
- M. M. Halldórsson, G. Kortsarz, A. Proskurowski, R. Salman, H. Shachnai, e J. A. Telle. Multicoloring trees. Information and Computation, 180:113–129, 2003.
- P. Hanczar. A fuel distribution problem application of new multi-item inventory routing formulation. Procedia - Social and Behavioral Sciences, 54:726–735, 2012.

- A. Haned, A. Soukhal, M. Boudhar, e N. H. Tuong. Scheduling on parallel machines with preemption and transportation delays. Computers & Operations Research, 39:374–381, 2012.
- A. Hemmati, M. Stålhane, L. M. Hvattum, e H. Andersson. An effective heuristic for solving a combined cargo and inventory routing problem in tramp shipping. Computers & Operations Research, 64:274–82, 2015.
- Y. T. Herer e R. Levy. Raising the competitive edge in manufacturing the metered inventory routing problem, an integrative heuristic algorithm. International Journal of Production Economics, 51:69–81, 1997.
- D. S. Hochbaum. Approximation Algorithms for NP-hard Problems, capítulo 2, páginas 46–93. PWS Publishing Company, 1997.
- M. Joly, L. F. L. Moro, e J. M. Pinto. Planning and scheduling for petroleum refineries using mathematical programming. Brazilian Journal of Chemical Engineering, 19:207–28, 2002.
- T. Kis e A. Kovács. A cutting plane approach for integrated planning and scheduling. Computers & Operations Research, 39:320–327, 2012.
- N. Labadie, C. Prins, e C. Prodhon. Metaheuristics for Vehicle Routing Problems. John Wiley & Sons, 2016.
- G. Laporte e Y. Nobert. Exact algorithms for the vehicle routing problem, volume 132, capítulo 5, páginas 147–84. North-Holland, 1987.
- J. B. Lasserre. An integrated model for job-shop planning and scheduling. Management Science, 38:1201–1211, 1992.
- J. Lenstra, D. Shmoys, e É. Tardos. Approximation algorithms for scheduling unrelated parallel machines. Mathematical Programming, 46:259–271, 1990.
- C. L. Li, G. Vairaktarakis, e C. Y. Lee. Machine scheduling with deliveries to multiple customer locations. European Journal of Operational Research, 164:39–51, 2005.

- K. Li, C. Zhou, J. Y. T. Leung, e Y. Ma. Integrated production and delivery with single machine and multiple vehicles. Expert Systems with Applications, 57:12–20, 2016.
- S. L. Li. Multi-job cutting stock problem with due dates and release dates. Journal of the Operational Research Society, 47:490–510, 1996.
- Z. Li e M. G. Ierapetritou. Integrated production planning and scheduling using a decomposition framework. Chemical Engineering Science, 64:3585–3597, 2009.
- Y.-K. Lin, J. W. Fowler, e M. E. Pfund. Multiple-objective heuristics for scheduling unrelated parallel machines. European Journal of Operational Research, 227:239–253, 2013.
- J. Lysgaard, N. A. Letchford, e W. R. Eglese. A new branch-and-cut algorithm for the capacitated vehicle routing problem. Mathematical Programming, 100:423–445, 2004.
- R. Macedo, C. Alves, e J. M. Valério de Carvalho. Arc-flow model for the two-dimensional guillotine cutting stock problem. Computers & Operations Research, 37:991–1001, 2010.
- R. Macedo, C. Alves, J. Valério de Carvalho, F. Clautiaux, e S. Hanafi. Solving the vehicle routing problem with time windows and multiple routes exactly using a pseudo-polynomial model. European Journal of Operational Research, 214:536–45, 2011.
- E. Malaguti, R. M. Duran, e P. Toth. Approaches to real world two-dimensional cutting problems. Omega-International Journal of Management Science, 47:99–115, 2014.
- C. T. Maravelias. A decomposition framework for the scheduling of single- and multi-stage processes. Computers & Chemical Engineering, 30:407–20, 2006.
- C. T. Maravelias e C. Sung. Integration of production planning and scheduling: Overview, challenges and opportunities. Computers & Chemical Engineering, 33:1919–1930, 2009.

- S. Martello e P. Toth. Knapsack problems: algorithms and computer implementations. John Wiley & Sons, Inc., 1990.
- S. Martello, D. Pisinger, e D. Vigo. The three-dimensional bin packing problem. Operations Research, 48:256–67, 2000.
- D. M. Miller. An interactive, computer-aided ship scheduling system. European Journal of Operational Research, 32:363–379, 1987.
- S. M. J. Mirzapour Al-e hashem e Y. Rekik. Multi-product multi-period inventory routing problem with a transshipment option: A green approach. International Journal of Production Economics, 2013.
- A. Mjirda, B. Jarboui, R. Macedo, S. Hanafi, e N. Mladenovi. A two phase variable neighborhood search for the multi-product inventory routing problem. Computers & Operations Research, 52, Part B:291–299, 2014.
- N. Moin, S. Salhi, e N. Aziz. An efficient hybrid genetic algorithm for the multi-product multi-period inventory routing problem. International Journal of Production Economics, 133:334–343, 2010.
- A. Olivera e O. Viera. Adaptive memory programming for the vehicle routing problem with multiple trips. Computers & Operations Research, 34:28–47, 2007.
- J. Pereira. Procedures for the bin packing problem with precedence constraints. European Journal of Operational Research, 250:794–806, 2016.
- N. Perrot. Integer programming column generation strategies for the cutting stock problem and its variants. Tese de Doutoramento, Université Sciences et Technologies - Bordeaux I, 2005.
- J. A. Persson e M. Göthe-Lundgren. Shipment planning at oil refineries using column generation and valid inequalities. European Journal of Operational Research, 163:631–652, 2005.
- R. J. Petch e S. Salhi. A multi-phase constructive heuristic for the vehicle routing problem with multiple trips. Discrete Applied Mathematics, 133:69–92, 2003.

- M. L. Pinedo. Planning and scheduling in manufacturing and services, volume 24. Springer, New York, USA, 2005.
- M. L. Pinedo. Scheduling: theory, algorithms, and systems. Springer Science & Business Media, New York, USA, 2012.
- M. Rao. On the cutting stock problem. Journal of the Computer Society of India, 7: 35–39, 1976.
- H. Reinertsen e T. W. M. Vossen. The one-dimensional cutting stock problem with due dates. European Journal of Operational Research, 201:701–711, 2010.
- J. Rietz, C. Alves, e J. de Valério de Carvalho. Fast heuristics for integrated planning and scheduling, volume 9156 of Lecture Notes in Computer Science, capítulo 30, páginas 413–428. Springer International Publishing, 2015.
- J. Rietz, C. Alves, N. Braga, e J. Valério de Carvalho. An exact approach based on a new pseudo-polynomial network flow model for integrated planning and scheduling. Computers & Operations Research, 76:183–94, 2016.
- B. Roe, L. G. Papageorgiou, e N. Shah. A hybrid milp/clp algorithm for multipurpose batch process scheduling. Computers & Chemical Engineering, 29:1277–91, 2005.
- D. M. Ryan e B. A. Foster. An integer programming approach to scheduling. Computer scheduling of public transport urban passenger vehicle and crew scheduling, páginas 269–280, 1981.
- R. Sadykov e L. A. Wolsey. Integer programming and constraint programming in solving a multimachine assignment scheduling problem with deadlines and release dates. INFORMS Journal on Computing, 18:209–217, 2006.
- S. Salhi e R. J. Petch. A ga based heuristic for the vehicle routing problem with multiple trips. Journal of Mathematical Modelling and Algorithms, 6:591–613, 2007.
- G. Scheithauer. Zuschnitt-und Packungsoptimierung. Springer, Wiesbaden, Germany, 2008.

- V. Schmid, K. F. Doerner, e G. Laporte. Rich routing problems arising in supply chain management. European Journal of Operational Research, 224:435–448, 2013.
- A. Scholl, R. Klein, e C. Jürgens. Bison: A fast hybrid procedure for exactly solving the one-dimensional bin packing problem. Computers & Operations Research, 24:627–645, 1997.
- P. Senthilkumar. Literature review of single machine scheduling problem with uniform parallel machines. Intelligent Information Management, 02:457–74, 2010.
- D. Shabtay e S. Karhi. Online scheduling of two job types on a set of multipurpose machines with unit processing times. Computers & Operations Research, 39:405–412, 2012.
- P. Shaw. A Constraint for Bin Packing, páginas 648–662. Springer, Berlin, Heidelberg, 2004.
- D. E. Shobrys e D. C. White. Planning, scheduling and control systems: why can they not work together. Computers & Chemical Engineering, 24:163–173, 2000.
- E. Silva, F. Alvelos, e J. Valério de Carvalho. An integer programming model for two- and three-stage two-dimensional cutting stock problems. European Journal of Operational Research, 205:699–708, 2010.
- J. H. Song e K. C. Furman. A maritime inventory routing problem: Practical approach. Computers & Operations Research, 40:657–665, 2013.
- H. Stadtler. A comparison of two optimization procedures for 1- and 1 1/2-dimensional cutting stock problems. Operations-Research-Spektrum, 10:97–111, 1988.
- L.-H. Su, T. C. E. Cheng, e F.-D. Chou. A minimum-cost network flow approach to preemptive parallel-machine scheduling. Computers & Industrial Engineering, 64:453–458, 2013.
- S. M. A. Suliman. Pattern generating procedure for the cutting stock problem. International Journal of Production Economics, 74:293–301, 2001.

- C. Sung e C. T. Maravelias. An attainable region approach for production planning of multiproduct processes. AIChE Journal, 53:1298–315, 2007.
- Taillard, xc, D. ric, G. Laporte, e M. Gendreau. Vehicle routeing with multiple use of vehicles. The Journal of the Operational Research Society, 47:1065–70, 1996.
- P. Toth e D. Vigo. An overview of vehicle routing problems, páginas 1–26. Society for Industrial and Applied Mathematics, 2001.
- P. Toth e D. Vigo. Vehicle Routing. MOS-SIAM Series on Optimization. Society for Industrial and Applied Mathematics, 2014.
- J. Valério de Carvalho. Exact solution of binpacking problems using column generation and branchandbound. Annals of Operations Research, 86:629–659, 1999.
- J. Valério de Carvalho. Lp models for bin packing and cutting stock problems. European Journal of Operational Research, 141:253–273, 2002.
- P. H. Vance, C. Barnhart, E. L. Johnson, e G. L. Nemhauser. Solving binary cutting stock problems by column generation and branch-and-bound. Computational Optimization and Applications, 3:111–130, 1994.
- J. Váncza, T. Kis, e A. Kovács. Aggregation - the key to integrating production planning and scheduling. CIRP Annals - Manufacturing Technology, 53:377–80, 2004.
- H. Y. Wang e C. Y. Lee. Production and transport logistics scheduling with two transport mode choices. Naval Research Logistics, 52:796–809, 2005.
- Q. Wang, R. Batta, e R. J. Szczerba. Sequencing the processing of incoming mail to match an outbound truck delivery schedule. Computers & Operations Research, 32:1777–1791, 2005.
- X. Wang e T. C. E. Cheng. Machine scheduling with an availability constraint and job delivery coordination. Naval Research Logistics, 54:11–20, 2007.

- T. Westerlund e J. Isaksson. Some efficient formulations for the simultaneous solution of trim-loss and scheduling problems in the paper-converting industry. Chemical Engineering Research & Design, 76:677–84, 1998.
- G. Wäscher, H. Haußner, e H. Schumann. An improved typology of cutting and packing problems. European Journal of Operational Research, 183:1109–1130, 2007.
- H.-S. Yan e X.-D. Zhang. A case study on integrated production planning and scheduling in a three-stage manufacturing system. IEEE Transactions on Automation Science and Engineering, 4:86–92, 2007.
- H.-a. Yang, Q.-f. Sun, C. Saygin, e S.-d. Sun. Job shop scheduling based on earliness and tardiness penalties with due dates and deadlines: an enhanced genetic algorithm. The International Journal of Advanced Manufacturing Technology, 61:657–666, 2012.
- Y. G. Yu, H. X. Chen, e F. Chu. A new model and hybrid approach for large scale inventory routing problems. European Journal of Operational Research, 189:1022–40, 2008.
- Y. Zhong e E.-H. Aghezzaf. Modeling and solving the multi-period inventory routing problem with constant demand rates. Ghent University, Department of Industrial management, 2012.