

---

O problema de minimização de pilhas abertas –  
novas contribuições

*Claudia Fink*

---



SERVIÇO DE PÓS-GRADUAÇÃO DO ICMC-USP

Data de Depósito:

Assinatura: \_\_\_\_\_

# O problema de minimização de pilhas abertas - novas contribuições<sup>1</sup>

**Claudia Fink**

***Orientador: Prof. Dr. Horacio Hideki Yanasse***  
***Co-Orientador: Prof. Dr. Alysson Machado Costa***

Tese apresentada ao Instituto de Ciências Matemáticas e de Computação - ICMC-USP, como parte dos requisitos para obtenção do título de Doutor em Ciências - Ciências de Computação e Matemática Computacional. *VERSÃO REVISADA*

**USP – São Carlos**  
**Dezembro de 2012**

---

<sup>1</sup>Este trabalho contou com apoio financeiro da FAPESP e CNPq.

Ficha catalográfica elaborada pela Biblioteca Prof. Achille Bassi  
e Seção Técnica de Informática, ICMC/USP,  
com os dados fornecidos pelo(a) autor(a)

F499p Fink, Claudia  
O problema de minimização de pilhas abertas -  
novas contribuições / Claudia Fink; orientador  
Horacio Hideki Yanasse; co-orientador Alysson  
Machado Costa. -- São Carlos, 2012.  
89 p.

Tese (Doutorado - Programa de Pós-Graduação em  
Ciências de Computação e Matemática Computacional) --  
Instituto de Ciências Matemáticas e de Computação,  
Universidade de São Paulo, 2012.

1. Otimização Combinatória. 2. Heurística. I.  
Yanasse, Horacio Hideki, orient. II. Costa, Alysson  
Machado, co-orient. III. Título.

*Aos meus pais  
Olávio e Maria.*



# Agradecimentos

---

---

A Deus por sua fiel companhia.

Ao meu orientador professor Horacio Hideki Yanasse pelos seus ensinamentos, paciência e apoio durante esta pesquisa.

Ao professor Alysson Machado Costa pela co-orientação do trabalho e pela motivação sempre presente nas nossas conversas.

Aos professores José Fernando e Maria Antónia pelo companheirismo e perseverança transmitidos nos momentos bons e ruins.

Aos meus pais Olávio e Maria Fink e minha irmã Neusa pelo carinho, amor e todas as oportunidades proporcionadas. Muito obrigada. Amo muito vocês.

A todos os meus amigos. Aos de infância, colégio, graduação e pós-graduação pelo apoio e amizade constantes. Aos de dança pelos muitos momentos de descontração. Aos do laboratório pelos momentos de estudo e pela alegria sempre presente. A dona Hilda e todas as meninas, em especial Ester, Isabela e Scheila pelo companheirismo e inesquecível período em que moramos juntas.

Aos professores e funcionários do ICMC que, direta ou indiretamente, contribuíram para a realização deste trabalho.

À Faculdade de Engenharia do Porto e CNPq pelo apoio durante os seis meses de estudo realizados em Portugal.

À FAPESP pelo apoio financeiro.





# Resumo

---

---

O Problema de Minimização do Número Máximo de Pilhas Abertas (MOSP, do inglês *minimization of open stacks problem*) é um problema de otimização combinatorial da família NP-Difícil que vem recebendo grande atenção na literatura especializada. Este trabalho apresenta novas contribuições em termos de modelos e técnicas de resolução para o problema.

A primeira parte deste trabalho lidou com modelos matemáticos, sendo analisados os modelos existentes que se baseiam em programação inteira mista. Variações de um modelo da literatura foram propostas, com o objetivo de tentar diminuir o tempo de execução necessário para se obter uma solução exata com a utilização de pacotes comerciais. Os resultados mostraram que as propostas são capazes de acelerar a solução de algumas classes de instâncias mas, que de maneira geral, métodos baseados em relaxação linear encontram dificuldade em provar a otimalidade devido à baixa qualidade dos limitantes inferiores. Uma outra contribuição deste trabalho foi o desenvolvimento de um modelo conjunto para o problema MOSP e para o problema de minimização da duração de pedidos (MORP, do inglês *minimization of order spread problem*). Este modelo propõe um *framework* unificado em que os dois problemas podem ser resolvidos ao mesmo tempo, tendo suas funções objetivo individuais ponderadas através de pesos definidos pelo usuário.

A segunda parte do trabalho voltou-se para o desenvolvimento de métodos heurísticos para o MOSP. Duas estratégias de solução foram desenvolvidas. O primeiro método propõe uma transformação heurística entre o problema MOSP e o clássico problema do caixeiro viajante (TSP, do inglês *traveling salesman problem*). A partir de uma representação em grafo do MOSP, o TSP é definido por meio de uma regra de atribuição de distâncias baseadas nos graus dos nós. Nos testes computacionais, a estratégia proposta mostrou-se eficiente em relação às heurísticas específicas para o MOSP, obtendo a solução ótima do MOSP em 80,42% das instâncias testadas e sendo competitiva em termos de tempo computacional com algumas das melhores heurísticas da literatura. O segundo método heurístico proposto utilizou a ideia de decomposição. De fato, neste método, um corte no grafo associado ao problema original divide-o em problemas menores, que são resolvidos. A solução global é obtida através da junção das soluções dos subproblemas e, em alguns casos, é possível demonstrar a otimalidade da solução obtida. Testes computacionais indicam a validade da proposta e apontam caminhos para pesquisas futuras.

**Palavras-chave:** Sequenciamento de padrões, minimização do número máximo de pilhas abertas, modelo integrado, método heurístico.



# Abstract

---

---

The minimization of open stacks problem (MOSP) is a well known NP-hard combinatorial optimization problem that has been extensively discussed in the specialized literature. This study presents some new contributions in terms of models and solution methods for this problem.

The first part of this thesis dealt with mathematical models. The existing mixed-integer models have been analyzed and variants of a well known model have been proposed, with the goal of reducing the time needed by commercial packages to obtain proved-optimal solutions. The results of computational tests on a widely used set of instances have indicated that the modifications proposed are able to reduce the time needed to obtain optimal solutions for some classes of instances. Nevertheless, a conclusion has been the fact that mixed-integer programming models have difficulty in obtaining convergence due to the low quality linear relaxation bounds. Another contribution of this thesis is the proposal of a single model that is able to deal with both the MOSP and with the Minimization of Order Spread Problem (MORP). This unified framework allows both problems to be jointly solved, by using a weighted objective function that included both original objectives.

The second part of this thesis dealt with the development of heuristic strategies. Two solution strategies have been proposed. The first method proposes a heuristic conversion between MOSP and Traveling Salesman Problem (TSP) instances. This conversion relies the assignment distances to the TSP instance based on the degree of the vertices of the associated MOSP graph. Computational tests have shown that the proposed methodology is efficient, both in terms of solution quality (optimal solutions were obtained for 80.42% of the tested instances) and computational effort. The second method uses a decomposition idea. A cut is made in the graph associated with the original MOSP problem, yielding two smaller problems, which are solved. In some cases, the obtained combined solution can be proved optimal. Computational tests have shown the validity of the proposal and indicate new research opportunities.

**Keywords:** Pattern sequencing, minimization of open stack problem, integrated model, heuristic.



# Sumário

---

---

<b>Resumo</b>	<b>v</b>
<b>Abstract</b>	<b>vii</b>
<b>Lista de Figuras</b>	<b>xi</b>
<b>Lista de Tabelas</b>	<b>xiv</b>
<b>1 Introdução</b>	<b>1</b>
<b>2 O MOSP</b>	<b>5</b>
2.1 Considerações sobre o MOSP . . . . .	5
2.1.1 Grafos MOSP . . . . .	7
2.2 O MOSP na literatura . . . . .	8
<b>3 Modelo MOSP e modelo integrado MOSP/MORP</b>	<b>33</b>
3.1 Modelo MOSP de Yanasse e Pinto (2003) . . . . .	33
3.2 Variações do modelo MOSP de Yanasse e Pinto (2003) e novo modelo . . . . .	36
3.2.1 Experimentos computacionais . . . . .	41
3.2.2 Considerações finais . . . . .	46
3.3 Modelo integrado MOSP/MORP . . . . .	47
3.3.1 Modelo integrado MOSP/MORP . . . . .	48
3.3.2 Experimentos computacionais . . . . .	50
3.3.3 Considerações finais . . . . .	55
<b>4 Heurísticas para solução do MOSP</b>	<b>57</b>
4.1 Uma estratégia de solução do MOSP utilizando o TSP . . . . .	57
4.1.1 O problema do caixeiro viajante . . . . .	58
4.1.2 Estratégia de solução MOSP/TSP proposta . . . . .	59
4.1.3 Experimentos computacionais . . . . .	64
4.1.4 Considerações finais . . . . .	68
4.2 Heurística de particionamento do grafo MOSP . . . . .	69
4.2.1 Os limitantes inferior e superior para o MOSP . . . . .	69
4.2.2 Heurística de particionamento estudada . . . . .	73
4.2.3 Experimentos computacionais . . . . .	78
4.2.4 Considerações finais . . . . .	81



---

# Lista de Figuras

---

---

2.1	Grafo MOSP de itens e grafo MOSP de padrões do problema. . . . .	8
2.2	Exemplo do funcionamento do problema GMLP. . . . .	20
3.1	Restrições (3.2)-(3.5) do modelo MOSP para o exemplo dado pela matriz <b>A</b> da Seção 3.1. . . . .	38
3.2	Restrições (3.3) consideradas no modelo MOSP-3 para o exemplo dado pela matriz <b>A</b> da Seção 3.1. . . . .	39
3.3	Gráfico de perfis de desempenho comparando os modelos MOSP, MOSP-1, MOSP-2, MOSP-3 e MOSP-4 em relação ao tempo computacional. . . . .	43
3.4	Gráfico de perfis de desempenho comparando os modelos MOSP, MOSP-4 e MOSP NOVO em relação ao tempo computacional. . . . .	44
4.1	Grafo TSP obtido a partir do problema MOSP apresentado na Tabela 4.1, usando a estratégia proposta. . . . .	63
4.2	Comparação entre a estratégia proposta e as heurísticas específicas para o MOSP, considerando os valores das soluções obtidas na solução das instâncias de teste. . . . .	65
4.3	Grafos reduzidos obtidos durante a heurística HCA. . . . .	70
4.4	Grafos obtidos durante a heurística HNCM. . . . .	72
4.5	Subgrafos obtidos com $cut \in [2, 4]$ . . . . .	74





# Lista de Tabelas

---

---

2.1	Padrões de corte. . . . .	6
2.2	Processo de corte dos padrões. . . . .	6
2.3	Estratégias utilizadas para a solução do MOSP em cada um dos trabalhos considerados. . . . .	10
3.1	Classes utilizadas nos experimentos. . . . .	41
3.2	Porcentagem de instâncias em que é obtida uma solução ótima em cada classe wbo_M_N com os modelos MOSP, MOSP-1, MOSP-2, MOSP-3, MOSP-4 e MOSP NOVO. . . . .	45
3.3	Média do tempo computacional, em segundos, para resolver cada instância das classes wbo_M_N no CPLEX com os modelos MOSP, MOSP-1, MOSP-2, MOSP-3, MOSP-4 e MOSP NOVO. . . . .	45
3.4	Influência do valor de $Q$ em relação a quantidade de instâncias da classe wbo_20_10 para as quais é obtida uma solução ótima. . . . .	46
3.5	Influência do valor de $Q$ em relação ao tempo computacional médio, em segundos, para resolver cada instância da classe wbo_20_10. . . . .	46
3.6	Classes de instâncias utilizadas nos testes computacionais. . . . .	51
3.7	Média do tempo computacional (em segundos) para resolver as instâncias da classe wbo_10_10 pelo CPLEX usando o modelo MOSP/MORP com prioridade nos diferentes grupos de variáveis. . . . .	52
3.8	Média do valor obtido na relaxação linear do modelo MOSP/MORP (teste (i)) e a média do valor do limitante inferior calculado no nó raiz do método <i>branch-and-bound</i> no CPLEX (testes (ii) e (iii)). A média é calculada para cada classe de instâncias. . . . .	53
3.9	Média do tempo computacional (em segundos) para resolver a relaxação linear do modelo MOSP/MORP (teste (i)) e o nó raiz no método <i>branch-and-bound</i> no CPLEX (testes (ii) e (iii)). A média é calculada para cada classe de instâncias. . . . .	53
3.10	Quantidade de problemas em que é obtida a solução ótima dos modelos considerando ou não prioridade nas variáveis $X$ . As instâncias foram agrupadas conforme o valor de $Q$ em cada classe. . . . .	54
3.11	Diferença entre o valor da solução obtido em cada modelo e a solução ótima da instância. . . . .	55
4.1	Padrões de corte. . . . .	63

4.2	Sequências de itens analisadas a partir do ciclo Hamiltoniano obtido no TSP, sequência de padrões correspondentes e número máximo de pilhas abertas. . . . .	63
4.3	Classes de instâncias utilizadas nos testes computacionais. . . . .	65
4.4	Informações sobre a solução das instâncias de teste pertencentes às classes especificadas. Os resultados foram agrupados de acordo com a densidade (D) das instâncias e são apresentados a média aritmética (M) e o desvio padrão (DP) de cada grupo, considerando o valor da solução e o tempo de solução (em milissegundos). . . . .	66
4.5	Valor da solução e tempo de solução para cada instância das classes gp_50_50 e gp_100_100 usando os métodos de solução indicados. . . . .	66
4.6	Quantidade de instâncias para as quais é obtida a diferença de 0, 1, 2, 3 ou mais pilhas em relação à solução ótima do MOSP. . . . .	67
4.7	Quantidade de instâncias para as quais a solução do MOSP obtida pela estratégia proposta, utilizando a solução exata do TSP, possui diferença de 0, 1, 2 ou 3 pilhas em relação à solução ótima do MOSP. . . . .	68
4.8	Arestas formadas com os padrões de corte da Tabela 2.1. . . . .	73
4.9	Classes de instâncias utilizadas nos testes computacionais e faixa de corte considerada para cada classe. . . . .	78
4.10	Número de instâncias que satisfazem cada um dos testes de parada aplicados na Heurística de particionamento considerando a condição $Boundary[cut] \leq LB$ . . . . .	79
4.11	Número de instâncias que satisfazem cada um dos testes de parada aplicados na Heurística de particionamento quando todas as instâncias são particionadas. . . . .	80
4.12	Soma do tempo computacional médio (em segundos) utilizado para resolver cada subproblema gerado em cada classe com o CPLEX quando todas as instâncias são particionadas. . . . .	80
4.13	Soma do tempo computacional médio (em segundos) utilizado para resolver cada subproblema gerado em cada classe com o CPLEX quando considerada a restrição $Boundary[cut] \leq LB$ do particionamento. . . . .	81
4.14	Quantidade de subgrafos de cada classe em que é atingido o tempo limite de 1800 segundos fixado para o CPLEX, utilizando ou não a condição $Boundary[cut] \leq LB$ . . . . .	81

---

# Introdução

---

O planejamento da produção das indústrias tem cada vez mais recebido investimentos no sentido de aumentar a eficiência de produção em relação, por exemplo, ao tempo de entrega dos produtos e diminuição de custos operacionais. A organização dos cenários de produção pode ser bastante complexa uma vez que as decisões tomadas em uma etapa da produção tendem a influenciar as decisões das demais etapas. Neste contexto, podem ser citadas as indústrias de papel, de aço, de móveis e têxtil, em que o planejamento da produção contém uma etapa importante onde peças maiores (objetos) são cortadas em itens menores (itens). O planejamento desta etapa pode ser dividido em duas fases.

A primeira fase consiste em definir quais itens serão obtidos de cada objeto considerando a necessidade de minimizar perdas e ou atender toda a demanda no tempo estabelecido. Este problema é conhecido como problema de corte de estoque e sua solução é dada por um conjunto de padrões de corte e a frequência que cada um desses padrões deve ser cortado.

Na segunda fase de planejamento deve-se decidir a ordem em que os padrões obtidos serão cortados. Cada corte resulta no empilhamento dos itens ao redor da máquina de corte. Uma nova pilha é aberta toda vez que um novo tipo de item é cortado e permanece aberta até que o último padrão que contém o tipo de item correspondente a esta pilha seja cortado. Este é um problema de sequenciamento de corte dos padrões e ganha importância em situações práticas quando, por exemplo, devido a limitações de espaço, o número de pilhas que podem ficar abertas ao redor da máquina de corte é restrito. Este problema de sequenciamento de padrões é conhecido como MOSP (do inglês, *Minimization of open stack problem*).

Outra aplicação para o MOSP surge no sequenciamento da produção de indústrias, em que uma quantidade de pedidos de clientes deve ser satisfeita. Cada pedido é composto de diferentes

produtos e somente um produto é produzido por vez pela empresa. Uma pilha é designada para cada pedido e esta pilha é aberta logo que o primeiro produto deste pedido é produzido. Quando todos os produtos do pedido forem produzidos, a pilha é fechada e encaminhada para o cliente. Um objetivo possível do problema de sequenciamento dos produtos é a minimização do número de pedidos que estão simultaneamente em aberto. Em especial, no trabalho de Linhares e Yanasse (2002) os problemas *modified cutwidth problem* e o *gate matrix layout problem* são resolvidos como sendo equivalentes ao MOSP.

Além do MOSP, outros problemas de sequenciamento de padrões com objetivos específicos têm sido bastante discutidos na área de planejamento de produção. Em Fink e Voss (1999) e Linhares e Yanasse (2002) são citados os seguintes problemas correlatos: o problema de minimização da duração de pedidos (MORP, do inglês *The minimization of order spread problem*), o problema de minimização de descontinuidades (MDP, do inglês *The minimization of discontinuities problem*) e o problema de minimização de troca de ferramentas (MTSP, do inglês *The minimization of tool switches problem*).

Nosso trabalho tem como foco de estudo o MOSP que, como pode ser visto em Linhares e Yanasse (2002), é um problema NP-difícil. A solução exata do MOSP pelos métodos sugeridos na literatura é dependente da quantidade de padrões e itens do problema e pode ser impraticável em algumas situações. Atualmente, um método de enumeração por itens (Yanasse e Lamosa, 2007) e dois métodos de programação dinâmica (de la Banda e Stuckey, 2007; Chu e Stuckey, 2009) foram propostos e parecem ser as melhores opções disponíveis na literatura para obter uma solução exata do MOSP. Com o intuito de obter resultados num tempo de solução menor, nos últimos anos também foram desenvolvidos inúmeros métodos heurísticos.

Este trabalho consiste num estudo do problema MOSP e organiza conceitos teóricos necessários para desenvolver, como trabalho de doutorado, métodos de solução eficientes para o problema. Na primeira parte do trabalho é estudada a formulação matemática do MOSP proposta em Yanasse e Pinto (2003). Algumas variações são computacionalmente analisadas com o intuito de tentar diminuir o tempo computacional necessário para resolver o modelo. Em seguida, um novo modelo para o MOSP é proposto baseado nas variações estudadas. Ainda focando a formulação matemática, é proposto um modelo integrado dos problemas MOSP e MORP. A função objetivo do novo modelo integrado inclui o objetivo dos dois problemas. O modelo também pode ser utilizado para resolver os problemas MOSP e MORP de maneira independente.

A segunda parte do trabalho está voltada para a solução heurística do problema MOSP. É proposta uma estratégia de solução que consiste em converter uma instância do MOSP em uma instância do problema do caixeiro viajante (TSP, do inglês *Traveling salesman problem*) de modo a obter uma boa solução para o problema original através da resolução deste outro problema. A motivação para esta investigação está no fato do TSP possuir uma literatura bastante rica tanto em formulações como em métodos exatos. No segundo método heurístico proposto, a ideia é decompor o problema original em problemas menores. São definidas algumas condições de otimalidade relacionando os valores da solução dos subproblemas obtidos com o problema original.

A continuação deste trabalho está organizada da seguinte forma. Inicialmente, no Capítulo 2, são descritos conceitos teóricos do MOSP e as principais contribuições de alguns trabalhos da literatura. O Capítulo 3 aborda a formulação matemática do MOSP: primeiro um modelo da literatura é analisado e algumas variações são computacionalmente testadas; em seguida, é proposto um modelo integrando o MOSP e o MORP. No Capítulo 4 são discutidos dois métodos heurísticos desenvolvidos: o primeiro consiste em relacionar o MOSP com o TSP; no segundo, uma heurística de decomposição do MOSP é estudada. Para finalizar, no Capítulo 5 é apresentada a conclusão do trabalho de doutorado realizado.



---

## O MOSP

---

Neste Capítulo, inicialmente, são discutidos conceitos teóricos do MOSP e apresentadas duas maneiras de modelar o MOSP como um problema de busca em grafo. Em seguida, são descritas as principais contribuições de alguns trabalhos da literatura.

### 2.1 Considerações sobre o MOSP

Como visto anteriormente, o problema de minimização do número máximo de pilhas abertas (MOSP) aparece em diversas situações práticas como, por exemplo, o corte de placas de matéria-prima em itens menores. Supondo que os padrões de corte já tenham sido determinados, o problema consiste em encontrar uma sequência de corte dos padrões que produza o menor número máximo de pilhas abertas em torno da máquina de corte.

Os itens resultantes do corte de um padrão são empilhados ao redor da máquina de corte e uma nova pilha é aberta toda vez que um novo tipo de item é cortado. Uma pilha é formada por um único tipo de item e permanece aberta até que o último padrão que contém um item do tipo correspondente seja cortado. Após isso, a pilha é considerada fechada e pode ser removida. O sequenciamento dos padrões deve minimizar o número de pilhas que permanecem abertas simultaneamente, motivado pela existência de limitações físicas para o armazenamento das pilhas ao redor da máquina de corte. Em geral, o tamanho de cada pilha é ignorado. Isto porque mesmo que exista uma limitação para o tamanho de uma pilha, entende-se que quando sua capacidade máxima for atingida e ainda existirem itens do tipo correspondente a serem cortados, os itens que estão na pilha podem ser removidos permitindo que os demais sejam adicionados no mesmo local como se a pilha ainda estivesse aberta.

Em seguida é apresentado um exemplo MOSP com seis padrões de corte previamente definidos ( $P_1, P_2, P_3, P_4, P_5, P_6$ ) e sete tipos de itens ( $a, b, c, d, e, f, g$ ). O conjunto de padrões de corte é apresentado na Tabela 2.1, em que, por exemplo, o padrão  $P_1$  contém os tipos de item  $a, d$  e  $e$ .

**Tabela 2.1:** Padrões de corte.

Padrão	Tipos de item
$P_1$	$a, d, e$
$P_2$	$b, e, f$
$P_3$	$e, g$
$P_4$	$d, e, f$
$P_5$	$b, c$
$P_6$	$a, b, e$

Na Tabela 2.2 é mostrado o processo de corte dos padrões considerando a sequência  $P_5, P_3, P_6, P_1, P_2, P_4$ . Logo que o primeiro padrão  $P_5$  é cortado, as pilhas dos itens  $b$  e  $c$  são abertas. Antes de iniciar o corte do próximo padrão, a pilha do item  $c$  é removida, já que, nenhum dos padrões ainda por cortar contém este tipo de item. Ao terminar o processo de corte dos padrões, o número máximo de pilhas abertas para a sequência de padrões considerada é 4.

**Tabela 2.2:** Processo de corte dos padrões.

Estágio	Padrão	Pilhas abertas	Número de pilhas
1	$P_5$	$b, c$	2
2	$P_3$	$b, e, g$	3
3	$P_6$	$a, b, e$	3
4	$P_1$	$a, b, d, e$	4
5	$P_2$	$b, d, e, f$	4
6	$P_4$	$d, e, f$	3

Seja uma sequência de padrões  $S_P$ . No processamento destes padrões é possível estabelecer a ordem em que as pilhas dos itens são fechadas. Para completar a pilha de item  $j$ , todos os padrões que contém o tipo de item  $j$  (e talvez alguns outros padrões) precisam aparecer na sequência ótima de padrões ocupando posições entre 1 até o estágio em que a pilha do item  $j$  é completada. Por exemplo, o corte da sequência de padrões da Tabela 2.2, a pilha do item  $a$  é fechada no estágio 4 e os padrões  $P_1, P_5$  e  $P_6$  são todos processados anteriormente e no estágio 4.

Seja a sequência de itens  $S_\pi$  correspondente a ordem em que as  $M$  pilhas do problema são fechadas e  $S_P^j$  uma sequência dos padrões necessários para completar a pilha do item  $j$ . Pode-se construir uma sequência de padrões do problema adicionando em ordem as subsequências  $S_P^j$  para  $j = S_{\pi_i}, i = 1, \dots, M$ , em que,  $S_{\pi_i}$  é o tipo de item que ocupa a  $i$ -ésima posição na sequência de itens  $S_\pi$ . Os padrões repetidos não são incluídos na sequência de padrões. Se a sequência de itens considerada é ótima, então a sequência de padrões assim construída também é ótima (Yanasse, 1997a). Portanto, para encontrar uma solução ótima para o MOSP é possível buscar uma sequência



ótima de itens que corresponde a ordem em que as pilhas são fechadas e, dessa forma, a sequência de padrões é calculada a partir desta sequência de itens.

Considerando o problema MOSP da Tabela 2.1 e uma sequência de itens  $S_\pi = \{c, g, a, b, d, e, f\}$  com número máximo de pilhas abertas igual a 4. As subsequências de padrões  $S_P^j$ , conforme a ordem dos tipos de itens da sequência, são:  $S_P^c = \{P_5\}$ ,  $S_P^g = \{P_3\}$ ,  $S_P^a = \{P_1, P_6\}$ ,  $S_P^b = \{P_2, P_5, P_6\}$ ,  $S_P^d = \{P_1, P_4\}$ ,  $S_P^e = \{P_1, P_2, P_3, P_4, P_6\}$  e  $S_P^f = \{P_2, P_4\}$ . A sequência de padrões é  $S_P = \{S_P^c, S_P^g, S_P^a, S_P^b, S_P^d, S_P^e, S_P^f\} = \{P_5, P_3, P_1, P_6, P_2, P_4\}$ . Note que a sequência  $S_P$  é concluída quando adicionadas as cinco primeiras subsequências, ou seja, não seria necessário calcular  $S_P^e$  e  $S_P^f$ .

Seja outra sequência de itens,  $S_\pi = \{a, b, c, d, e, f, g\}$ , com número máximo de pilhas abertas igual a 5. Neste caso, uma sequência de padrões gerada é  $S_P = \{P_1, P_6, P_2, P_5, P_4, P_3\}$  concluída antes de adicionar  $S_P^f$  e  $S_P^g$ . Esta mesma sequência de padrões pode ser obtida a partir da sequência de itens  $S_\pi = \{a, b, c, d, f, e, g\}$ .

### 2.1.1 Grafos MOSP

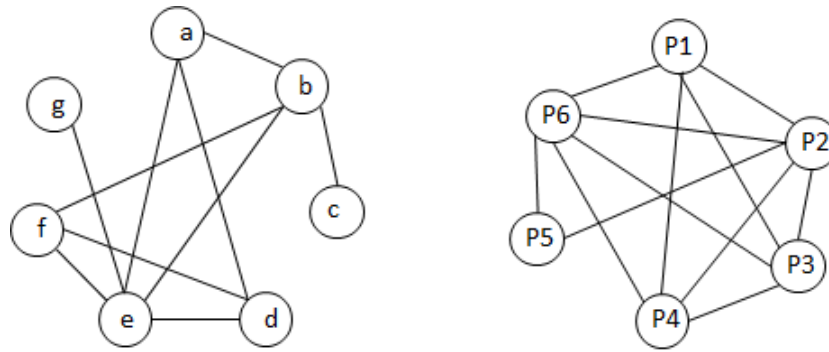
O MOSP pode ser modelado como um problema de busca em grafo. Alguns trabalhos consideram um grafo em que os vértices representam os padrões do problema MOSP (grafo MOSP de padrões) enquanto outros utilizam um grafo em que os vértices indicam os tipos de itens do problema (grafo MOSP de itens). Os dois grafos são definidos a seguir.

O grafo MOSP de padrões é definido como  $G^P(V^P, E^P)$ , em que cada vértice  $v \in V^P$  indica um padrão do problema MOSP e existe uma aresta  $a_{ij} \in E^P$  ligando dois vértices  $v_i, v_j \in V^P$  se, e somente se, os padrões associados a estes vértices possuem algum tipo de item em comum.

No grafo MOSP de itens  $G^I(V^I, E^I)$ , cada vértice  $v \in V^I$  representa um tipo de item do problema MOSP e existe uma aresta  $a_{ij} \in E^I$  ligando dois vértices  $v_i, v_j \in V^I$  se, e somente se, os itens associados a estes vértices aparecem simultaneamente em algum padrão. Ainda que dois itens estejam juntos em mais de um padrão, apenas uma aresta é criada entre os vértices equivalentes. Em  $G^I(V^I, E^I)$ , cada padrão do problema corresponde a um clique.

A Figura 2.1 mostra o grafo MOSP de itens e o grafo MOSP de padrões do problema apresentado na Tabela 2.1. A inspiração para construção do grafo MOSP de itens vem de Lins (1989) que considerou um caso especial do MOSP em que os padrões possuem, no máximo, dois tipos distintos de itens. Após isso, Yanasse (1996) mostrou que qualquer exemplar do MOSP poderia ser modelado como um problema de busca em grafos.

As representações em grafo, de itens e de padrões, do MOSP têm motivado pesquisadores a explorar as características dos grafos associados para tentar obter métodos de solução mais eficientes. Algoritmos polinomiais para o MOSP no caso de se obter grafos MOSP com topologias especiais foram sugeridos por Yanasse (1996). Outros trabalhos que exploram a representação do MOSP como um problema de busca em grafo são (Yuen e Richardson, 1995; Ashikaga, 2001; Ashikaga



**Figura 2.1:** Grafo MOSP de itens e grafo MOSP de padrões do problema.

e Soma, 2009; Becceneri et al., 2004; Yanasse e Lamosa, 2007; Fink et al., 2010; de Carvalho e Soma, 2011), entre outros.

## 2.2 O MOSP na literatura

O MOSP tem recebido considerável atenção da literatura, tendo sido alvo de estudo do *Constraint Modelling Challenge* (Smith e Gent, 2005), o que resultou na proposta e divulgação de diversas metodologias para o problema.

A primeira formulação matemática para este problema foi proposta por Yanasse (1997a) e é baseada em um modelo para o problema de minimização de trocas de ferramentas (MTSP) proposto por Tang e Denardo (1988). Outros modelos para o MOSP foram propostos em Faggioli e Bentivoglio (1998), Yanasse e Pinto (2003), Pinto (2004), Laporte et al. (2004) e Fink et al. (2009). Os experimentos realizados pelos autores, mostram que estes modelos são resolvidos de forma exata em tempo computacional não proibitivo (inferior à 30 minutos) apenas para exemplares pequenos do problema ou exemplares com características específicas (por exemplo, que podem ser reduzidos a exemplares menores através de técnicas de pré-processamento).

Alternativamente ao uso destes modelos para resolver o MOSP, são encontrados na literatura outros métodos exatos. Métodos de enumeração implícita são tratados em Yuen e Richardson (1995), Yanasse (1997a), Faggioli e Bentivoglio (1998), Becceneri et al. (2004), Yanasse (2002) e Yanasse e Lamosa (2007). Estratégias de programação dinâmica foram recentemente propostas em de la Banda e Stuckey (2005, 2007) e Chu e Stuckey (2009). Métodos de ambas estratégias podem ser classificados em dois grupos, aqueles que buscam uma sequência de fechamento das pilhas abertas que é convertida numa sequência de corte dos padrões e, por outro lado, aqueles que buscam diretamente a sequência de padrões. Dos trabalhos citados, Yuen e Richardson (1995), Faggioli e Bentivoglio (1998) e de la Banda e Stuckey (2005, 2007) pertencem ao segundo grupo.

É importante ressaltar que o MOSP é um problema NP-difícil (Linhares e Yanasse, 2002) e, assim, a solução de instâncias de grande porte por métodos exatos não é garantida na prática, exceto para instâncias com características especiais. Como alternativa aos métodos exatos para

resolver o MOSP são propostos métodos heurísticos (Yuen, 1991, 1995; Yanasse, 1997b; Faggioli e Bentivoglio, 1998; Linhares e Yanasse, 2002; Becceneri et al., 2004; Ashikaga, 2001; Ashikaga e Soma, 2009; Fink et al., 2010; de Carvalho e Soma, 2011).

Dado o aumento do esforço computacional necessário para resolver exatamente o MOSP à medida que o tamanho do problema cresce, é interessante verificar se é possível reduzir o tamanho do problema MOSP e facilitar a sua resolução pelos métodos exatos. Algumas propostas de redução, também chamadas de estratégias de pré-processamento, são observadas em Yuen e Richardson (1995), Becceneri et al. (2004), de la Banda e Stuckey (2005, 2007), Chu e Stuckey (2009) e Yanasse e Senne (2010).

Linhares e Yanasse (2002) citam vários problemas que estão relacionados com o MOSP, em particular, alguns problemas no contexto de projetos VLSI (do inglês, *Very-large-scale integration*) são resolvidos como um MOSP. Yanasse (1997a), Yanasse e Pinto (2003); Pinto (2004); Yanasse e Lamosa (2007) apresentam algumas relações entre a solução ótima do MOSP e de outros problemas de sequenciamento como o MORP o MDP e o MTSP, definidos no Capítulo 1.

Os trabalhos são apresentados na Tabela 2.3 na qual são destacadas as estratégias utilizadas pelos autores para a solução do MOSP. Na coluna em que é mostrado o método exato de solução são utilizados os símbolos "BB" para indicar o método *branch and bound* e "PD" para indicar o método de programação dinâmica e ainda, "EP" se a enumeração é feita com padrões ou "EI" se a enumeração é feita com os itens.

Na sequência são apresentados em mais detalhes os principais desenvolvimentos realizados para o MOSP, dando especial atenção aos trabalhos mais relevantes para a proposta desenvolvida nesta tese.

## Yuen (1991, 1995)

No trabalho de Yuen (1991) são propostos dois métodos heurísticos para solução do MOSP, ambos compostos de duas etapas. Na primeira etapa, igual para ambas as heurísticas, é selecionado um tipo de item dentre os que têm a pilha já aberta. Na segunda etapa é selecionado um padrão ainda não sequenciado que contenha o tipo de item selecionado na primeira etapa. Cada heurística é executada  $N$  vezes (em que  $N$  é o número de padrões), sendo que em cada execução um padrão diferente é escolhido para iniciar a sequência de padrões. As duas heurísticas foram testadas em instâncias geradas pelo autor e apresentaram resultados similares tanto no número de pilhas obtidas como no tempo computacional utilizado.

A primeira etapa consiste em selecionar um tipo de item que tem a pilha aberta e que aparece no menor número de padrões ainda não cortados. Caso haja empate, dentre estes, soma-se a quantidade de um tipo de item em cada padrão não cortado e seleciona o tipo de item com menor valor da soma. Na heurística 1, a segunda etapa seleciona o padrão que contém o tipo de item escolhido antes e que é repetido mais vezes. Caso haja empate, seleciona dentre estes, o primeiro

**Tabela 2.3:** Estratégias utilizadas para a solução do MOSP em cada um dos trabalhos considerados.

Trabalho	Modelagem e representação		Método de solução			
	Formulação matemática	Grafo de padrões	Grafo de itens	Pré-processamento	Exato	Heurístico
Yuen (1991)						construtiva
Yuen (1995)						construtiva
Yuen e Richardson (1995)		X		X	BB(EP)	
Yanasse (1997a)	X				BB(EI)	
Yanasse (1997b)			X			construtiva
Faggioli e Bentivoglio (1998)	X				BB(EP)	busca tabu
Ashikaga (2001)			X			construtiva
Linhares e Yanasse (2002)	X		X			construtiva
Yanasse (2002)				X	BB(EI)	
Yanasse e Pinto (2003)	X					
Pinto (2004)	X				BB(EI)	
Becceneri et al. (2004)			X		BB(EI)	construtiva
de la Banda e Stuckey (2005)				X	BB(EI)	
Yanasse e Lamosa (2007)					PD(EP)	
Yanasse et al. (2007)					BB(EI)	
de la Banda e Stuckey (2007)					BB(EI)	
Chu e Stuckey (2009)					PD(EP)	
Ashikaga e Soma (2009)			X		PD(EI)	construtiva
Fink et al. (2009)	X				BB(EI)	
Yanasse e Senne (2010)				X		
Fink et al. (2010)						construtiva
de Carvalho e Soma (2011)						construtiva

padrão analisado com menor número de tipos de itens. Seja o tipo de item definido na etapa 1, na heurística 2 seleciona-se o padrão que contém este tipo de item e que abre o menor número de pilhas novas. Se mais de um padrão é obtido, seleciona dentre estes, o primeiro padrão que tem o maior número de tipos de itens com pilhas abertas.

No artigo Yuen (1995), 4 novas heurísticas são propostas e chamadas de heurísticas 3, 4, 5 e 6 respectivamente. Estas heurísticas consideram duas medidas:  $C_j$  que indica o número de tipos de itens comuns do padrão  $j$  com as pilhas abertas, e  $N_j$  que indica o número de tipos de itens do padrão  $j$  que não tem sua pilha aberta.

A heurística 3 é composta de apenas uma etapa na qual seleciona-se o padrão com maior valor de  $M_j = C_j - N_j$ , ou seja, o padrão  $j$  que mais se relaciona com as pilhas abertas no momento. Caso haja empate, seleciona dentre estes, o primeiro padrão com menor valor de  $N_j$ .

A heurística 4 seleciona o padrão com menor valor de  $N_j$ . Caso haja empate, seleciona dentre estes, o primeiro padrão encontrado.

A heurística 5 estende a ideia básica da heurística 3 e tenta visualizar não somente os benefícios do corte atual mas também do corte seguinte. Seleciona o padrão com maior valor de  $M_j$  e caso haja empate, simula o processamento de cada um dos padrões empatados e analisa o valor de  $M_j$  para o corte seguinte. Seleciona o padrão correspondente ao maior valor deste  $M_j$  seguinte e se houver empate, seleciona o primeiro encontrado entre eles.

A heurística 6 envolve de forma diferente os critérios já aplicados. Simula o processamento de todos os padrões ainda disponíveis e calcula o  $M_j$  para o corte seguinte. Seleciona o padrão que corresponde ao maior valor desse  $M_j$  seguinte. Caso haja empate, seleciona dentre estes padrões, aquele com menor valor de  $N_j$ . Se houver novo empate, seleciona o primeiro encontrado entre eles.

Em Yuen (1995), o autor desenvolve um procedimento exaustivo para analisar a otimalidade das soluções heurísticas obtidas, que foi aplicado em instâncias com até 30 padrões. As heurísticas 1 e 2 de Yuen (1991) também foram comparadas. As heurísticas 3 e 5 obtiveram as melhores soluções dentre as 6 heurísticas comparadas e encontraram uma solução ótima na maioria das instâncias consideradas. A heurística 5 porém, requer um tempo computacional bem maior para resolver problemas com número grande de padrões.

## Yuen e Richardson (1995)

Este trabalho aborda o mesmo problema de sequenciamento de padrões definido em Yuen (1991, 1995) representando-o por um grafo MOSP de padrões. É proposto um método de enumeração implícita por padrões em que são considerados alguns critérios de corte e de seleção de nós na árvore de busca. Antes de aplicar o método exato analisa-se a possibilidade de decompor o grafo em um conjunto de componentes. Se for possível, cada um dos componentes é resolvido de maneira independente.

O limitante inferior ( $LB$ ) é igual ao número máximo de tipos de itens em um padrão. O limitante superior ( $UB$ ) inicial é igual a melhor solução obtida entre as 6 heurísticas de Yuen (Yuen, 1991, 1995). O  $UB$  é atualizado toda vez que uma sequência completa de padrões é formada e o número máximo de pilhas desta sequência é menor que o valor atual de  $UB$ . Caso  $UB = LB$  então esta é uma solução ótima do problema e a busca termina. Quando a solução de um nó da árvore excede ou é igual ao valor do  $UB$ , ele é cortado e o método volta ao nó factível mais próximo para continuar a busca (*back-tracking*).

Na ramificação da árvore, são selecionados apenas padrões que têm ao menos um tipo de item comum às pilhas que estão abertas. É considerado um corte baseado na simetria do problema. Se uma subsequência inicial de padrões  $a, b, c, d$  tem a mesma solução que a subsequência invertida  $d, c, b, a$  então o nó com  $d, c, b, a$  é cortado.

Como no artigo de Yuen (1995), aqui também é utilizado um rearranjo dos padrões. Seja  $P_j$  o conjunto dos itens do padrão  $j$ . Se  $i \in P_j$  então  $a_{i,j} = 1$ , caso contrário  $a_{i,j} = 0$ . O número de padrões que contém o tipo de item  $i$  é  $O_i = \sum_{\forall j} a_{i,j}$ . Para cada padrão  $j$  é calculada uma medida de envolvimento deste padrão com os demais padrões dada por  $F_j = \sum_{\forall i} O_i a_{i,j}$ . Os padrões são ordenados conforme seu valor de envolvimento, do menor para o maior. Ao abrir a árvore de busca, os padrões são selecionados conforme este rearranjo obtido. Esta estratégia, segundo os autores, reduz significativamente o tempo computacional do método proposto.

## Yanasse (1997a)

O artigo aborda o problema MOSP e sugere algumas relações entre a solução do MOSP com a de outros problemas de sequenciamento de padrões: o MORP, o MDP e o MTSP (definidos no Capítulo 1). É apresentado um modelo matemático para o MOSP baseado no modelo matemático para o problema de minimização de trocas de ferramentas (MTSP) apresentado por Tang e Denardo (1988). Um método de enumeração por itens é proposto. A solução obtida corresponde à uma sequência de itens da qual é obtida uma sequência de padrões ótima para o problema.

O MTSP consiste em encontrar uma sequência para o processamento das tarefas sendo que, a máquina precisa estar carregada com as ferramentas necessárias para cada tarefa no momento em que a tarefa for processada. O objetivo é minimizar o número total de trocas de ferramentas na máquina. Este problema ocorre quando é utilizado um número de ferramentas superior ao limite da máquina, sendo assim necessário interromper o funcionamento da máquina para realizar a troca das ferramentas.

Dados do problema:

- $N$ : número de tarefas que precisam ser processadas.
- $M$ : número de ferramentas necessárias.

- $\mathbf{A}_j$ : vetor  $M \times 1$  binário que indica as ferramentas necessárias na tarefa  $j$ .
- $n$ : instante imediatamente após o término da  $n$ -ésima tarefa.
- $C$ : número máximo de ferramentas na máquina.
- $\mathbf{e}$ : vetor auxiliar  $1 \times M$  em que,  $\mathbf{e}[i] = 1, i = 1, \dots, M$ .

Variáveis do problema:

- $\mathbf{X}_{jn}$ : vale 1 se a tarefa  $j$  está na  $n$ -ésima posição da sequência, e vale 0 caso contrário.
- $\mathbf{W}_n$ : vetor  $M \times 1$  binário que indica as ferramentas que estão na máquina no instante  $n$ .
- $\mathbf{P}_n$ : vetor  $M \times 1$  binário que indica as ferramentas introduzidas na máquina após o término da  $n$ -ésima tarefa.

Formulação do problema MTSP de Tang e Denardo (1988):

$$\text{Minimizar } \sum_{n=1}^{N-1} \mathbf{e} \mathbf{P}_n \quad (2.1)$$

Sujeito a:

$$\mathbf{P}_n \geq \mathbf{W}_{n+1} - \mathbf{W}_n \quad n = 1, \dots, N-1 \quad (2.2)$$

$$\mathbf{P}_n \geq 0 \quad n = 1, \dots, N-1 \quad (2.3)$$

$$\mathbf{e} \mathbf{W}_n \leq C \quad n = 1, \dots, N-1 \quad (2.4)$$

$$\mathbf{X}_{jn} \mathbf{A}_j \leq \mathbf{W}_n \quad j = 1, \dots, N \text{ e } n = 1, \dots, N \quad (2.5)$$

$$\sum_{n=1}^N \mathbf{X}_{jn} = 1 \quad j = 1, \dots, N \quad (2.6)$$

$$\sum_{j=1}^N \mathbf{X}_{jn} = 1 \quad n = 1, \dots, N \quad (2.7)$$

$$\mathbf{X}_{jn} \in \{0, 1\} \quad j = 1, \dots, N \text{ e } n = 1, \dots, N \quad (2.8)$$

$$\mathbf{P}_n \text{ vetor binário} \quad n = 1, \dots, N \quad (2.9)$$

$$\mathbf{W}_n \text{ vetor binário} \quad n = 1, \dots, N. \quad (2.10)$$

As restrições (2.2) e (2.3) permitem identificar as novas ferramentas alocadas de um instante para o outro. As restrições (2.4) são as restrições de capacidade da máquina. As restrições (2.5) impõem que todas as ferramentas necessárias para a tarefa  $j$  estejam na máquina se ela for processada. Observe que o sinal de desigualdade  $\leq$  permite que na máquina haja também ferramentas que não serão utilizadas pela tarefa atual. As restrições (2.6) e (2.7) garantem que cada tarefa só pode ocupar uma posição na sequência e que cada posição na sequência contém uma única tarefa.

No caso em que  $C \geq M$ , todas as ferramentas podem ser postas na máquina e nenhuma troca será necessária. Por outro lado se  $C < M$ , haverá pelo menos  $M - C$  trocas de ferramentas, que é um limitante inferior para o problema MTSP.

Yanasse (1997a) propõe uma formulação para o MOSP relacionada ao MTSP em que, as tarefas correspondem aos padrões, as ferramentas indicam os tipos de itens e o parâmetro  $C$  indica o número de pilhas abertas e passa ser uma variável no modelo para o MOSP. Seja  $MTSP^*$  e  $MOSP^*$  os valores ótimos dos problemas MTSP e MOSP, respectivamente. As proposições apresentadas em Yanasse (1997a) garantem as duas relações apresentadas a seguir e, baseado nestas informações propõe o modelo para o MOSP.

1. Se a solução ótima do problema MTSP for igual a  $M - C$  então o número ótimo de pilhas no MOSP é menor ou igual a  $C$  ( $MTSP^* = M - C \rightarrow MOSP^* \leq C$ ).
2. Se a solução ótima do problema MTSP for maior que  $M - C$  então o ótimo do MOSP será maior que  $C$  ( $MTSP^* > M - C \rightarrow MOSP^* > C$ ).

Formulação do problema MOSP de Yanasse (1997a):

$$MOSP^* = \text{Minimizar } C \quad (2.11)$$

Sujeito a:

$$\sum_{n=1}^{N-1} \mathbf{eP}_n = M - C \quad (2.12)$$

e (2.2) – (2.10).

Os testes realizados por Tang e Denardo (1988) mostram que o modelo MTSP necessita de tempo computacional considerado impraticável o que, devido a semelhança, deve ocorrer com o modelo MOSP proposto. O artigo não apresenta testes computacionais indicando tempo e viabilidade do modelo MOSP proposto.

Yanasse (1997a) propõe um método simples de enumeração por itens para a solução do problema MOSP estudado. Uma característica importante deste método é que a enumeração é feita sobre os itens e não por padrões. A sequência de itens corresponde à ordem em que as pilhas dos itens são completadas. A ideia de encontrar uma sequência ótima de itens e a partir desta obter uma sequência de padrões foi discutido inicialmente na Seção 2.1.

Em cada nó analisado são calculados novos limitantes superior e inferior. O limitante superior em cada nó  $j$ ,  $UB^j$ , é igual ao número de tipos de item que ainda não tiveram suas pilhas completadas. O limitante inferior inicial  $LB_1$  corresponde ao número máximo de tipos de itens por padrão. Seja o nó  $j$ , para cada nó  $k$  ramificado de  $j$  calcula-se o número de pilhas abertas em  $k$ ,  $v_j^k$ . Um limitante inferior calculado em  $j$  é  $LB_2 = \min\{v_j^k, \forall k\}$  e o outro,  $LB_3$ , é igual ao número de pilhas abertas no nó  $j$  conforme sequência parcial de itens deste nó. O limitante inferior do nó



inicial é  $LB^0 = \max\{LB_1, LB_2\}$  e para os nós  $j > 0$  é  $LB^j = \max\{LB_1, LB_2, LB^p\}$ , em que,  $p$  é o nó antecessor de  $j$ .

Seja  $v^{UB}$  o valor da melhor solução encontrada até o momento. Os critérios de corte considerados são: i) Se  $LB^j \geq v^{UB}$  então corta o nó  $j$ , ii) Se  $LB^j \geq UB^j$  então corta o nó  $j$  e atualiza o valor  $v^{UB}$  com  $LB^j$  se  $v^{UB} > LB^j$ .

Para ramificação da árvore de busca é adotado um critério guloso. Primeiro é feita uma busca em profundidade escolhendo sempre o nó mais promissor para continuar a busca. Nó promissor é aquele com o menor número de pilhas abertas. Quando um nó da árvore é cortado o método volta para o nó factível mais próximo e continua a busca.

## Yanasse (1997b)

É proposto um método heurístico baseado num procedimento de contração de arestas do grafo MOSP de itens.

A heurística chamada *arc collapsing algorithm* utiliza critérios simples para selecionar uma aresta a ser contraída e os vértices incidentes a ela são fundidos em um único vértice. Todas as arestas incidentes a estes dois vértices passam a ser incidentes ao novo vértice formado.

Este processo é realizado até que o grafo resultante seja trivialmente resolvido. Uma vez obtida a solução para o grafo modificado, a solução para o grafo original é obtida, inserindo convenientemente nesta sequência as arestas excluídas durante o procedimento.

## Faggioli e Bentivoglio (1998)

O trabalho de Faggioli e Bentivoglio (1998) analisa o MOSP e apresenta uma formulação do problema, métodos heurísticos e um método exato de enumeração por padrões. O modelo proposto considera:

- $N$ : número de padrões.
- $M$ : número de itens.
- $\mathbf{A} = [a_{ij}]$ : matriz  $N \times M$  binária em que,  $a_{ij} = 1$  se o padrão  $i$  contém o item  $j$ .
- $c_j = \sum_{i=1}^N a_{ij}$ : indica a quantidade de padrões  $i$  que contém o item  $j$ .

Seja  $S_P = (s_1, s_2, \dots, s_N)$  uma sequência dos padrões de corte. Uma sequência de vetores  $\mathbf{v}_k$ ,  $k = 1, \dots, N$  pode ser obtida da seguinte maneira:

$$\mathbf{v}_0 = 0 \text{ e } \mathbf{v}_k = \mathbf{v}_{k-1} + \mathbf{A}_{s_k}, k = 1, \dots, N.$$

em que,  $\mathbf{A}_{s_k}$  representa a linha  $s_k$  da matriz  $\mathbf{A}$  e  $\mathbf{v}_N = \mathbf{c}$ . Seja  $o_k = \langle \mathbf{v}_k \rangle + \langle \mathbf{c} - \mathbf{v}_{k-1} \rangle - M$ , em que,  $\langle \mathbf{v}_k \rangle$  é igual ao número de elementos do vetor  $\mathbf{v}_k$  que são maiores que zero. Desta forma, o número máximo de pilhas abertas pela sequência  $S_P$  é calculado por:

$$\text{maximizar}_{k=1,\dots,N} (o_k).$$

O problema de sequenciamento dos padrões de corte consiste em encontrar uma sequência que gera o menor valor máximo de pilhas abertas. Portanto, o seguinte problema precisa ser resolvido:

$$\text{minimizar}_{S_P} (\text{maximizar}_{k=1,\dots,N} (o_k)).$$

Além deste modelo, os autores apresentam dois métodos heurísticos, ambos com duas etapas. A primeira etapa consiste em gerar soluções iniciais com uma heurística gulosa. A segunda etapa consiste em melhorar a solução inicial utilizando um método de busca tabu ou busca local.

Na heurística gulosa são calculadas  $N$  sequências, cada uma iniciando com um padrão diferente. São adotados três critérios para a escolha do próximo padrão da sequência. Primeiro, selecionam-se os padrões que abrem o menor número de pilhas novas. Segundo, dentre os padrões já selecionados, selecionam-se os padrões que fecham o maior número de pilhas. Terceiro, dentre os padrões selecionados na etapa anterior, seleciona o padrão com maior número de tipos de itens iguais às pilhas já abertas. Se depois destes três critérios ainda houver empate, o primeiro dentre estes é escolhido. Não é difícil perceber a semelhança desta heurística com as heurísticas 3 e 5 de Yuen (1995).

O método de busca tabu consiste em efetuar trocas possíveis entre as posições dos padrões na melhor sequência obtida pela heurística gulosa. A busca tabu pára quando a solução atual não é melhorada após nove iterações.

A segunda heurística consiste em aplicar um método de busca local ao invés da busca tabu. Neste caso, a heurística gulosa e a busca local interagem, ou seja, calcula-se uma solução inicial e aplica-se a busca local, uma nova solução inicial é calculada e a busca local é aplicada novamente, assim por diante, até serem analisadas as  $N$  soluções iniciais obtidas pela heurística gulosa.

Os resultados obtidos pelas duas heurísticas foram comparados com a solução ótima obtida por um procedimento exato de enumeração por padrões. Este método exato é um melhoramento do método apresentado por Yuen e Richardson (1995). Foram adicionados novos critérios de corte permitindo assim a solução de problemas maiores.

## Linhares e Yanasse (2002)

Linhares e Yanasse (2002) apresentam uma formulação para o MOSP e, em seguida, o *cutwidth problem* é analisado e resolvido como um problema MOSP.

A formulação proposta para o MOSP considera:

- $N$ : quantidade de padrões.
- $M$ : quantidade de itens.
- $P$ : conjunto  $\{1, \dots, N\}$  de padrões.
- $\mathbf{A} = [a_{i,j}]$ : vale 1 se o item  $i$  está no padrão  $j$ , e vale 0 caso contrário.
- $\Gamma$ : conjunto com todas as possíveis sequências dos padrões  $\{1, \dots, N\}$ .
- $\pi(j)$ : posição que o padrão  $j$  ocupa na sequência  $\pi$ .
- $q_{i,j}^\pi$ : vale 1 se  $\exists u, v \in P$  tal que  $a_{i,u} = a_{i,v} = 1$  e  $\pi(u) \leq j \leq \pi(v)$ , e vale 0 caso contrário.

A matriz  $\mathbf{Q}^\pi$  recebe as colunas da matriz  $\mathbf{A}$  ordenadas conforme a sequência  $\pi$ , depois, qualquer zero entre as colunas correspondentes a  $u$  e  $v$  que satisfazem a definição de  $q_{i,j}^\pi$  recebe valor 1. Esta matriz  $\mathbf{Q}^\pi$  satisfaz a propriedade de 1s-consecutivos.

O objetivo do problema MOSP consiste em encontrar uma sequência  $\pi$  com o menor número possível de pilhas abertas simultaneamente, ou seja,

$$MOSP = \underset{\pi \in \Gamma}{\text{minimizar}} \left( \underset{j=1, \dots, N}{\text{maximizar}} \left( \sum_{i=1}^M q_{i,j}^\pi \right) \right).$$

Seja o problema MOSP representado pela matriz  $\mathbf{A}$  seguinte, em que, as linhas correspondem aos itens e as colunas aos padrões.

$$\mathbf{A} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 \end{bmatrix}.$$

As matrizes seguintes mostram a construção da matriz  $\mathbf{Q}^\pi$  para  $\pi = \{1, 2, 3, 4\}$ . Primeiro, a matriz  $\mathbf{Q}^\pi$  recebe as colunas da matriz  $\mathbf{A}$  conforme a sequência  $\pi$ . Depois, alguns elementos da matriz passam a valer 1 conforme a definição de  $\mathbf{Q}^\pi$ . Neste exemplo apenas  $q_{3,2}^\pi$  é alterado pois  $a_{3,1} = a_{3,3} = 1$  e  $(\pi(1) = 1) \leq (\pi(3) = 3)$ . Para obter a solução do MOSP basta somar os elementos de cada coluna e o maior destes valores, neste exemplo é 3, representa o número máximo de pilhas abertas para a sequência  $\pi$ .

$$\mathbf{Q}^{\{1,2,3,4\}} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 \end{bmatrix} \implies \mathbf{Q}^{\{1,2,3,4\}} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & \boxed{1} & 1 & 1 \\ 1 & 0 & 0 & 0 \end{bmatrix}.$$

Ao analisar a sequência  $\pi = \{2, 1, 3, 4\}$  é obtida a matriz  $\mathbf{Q}^\pi$  seguinte. Neste caso, não é necessário acrescentar nenhum valor 1 à matriz  $\mathbf{Q}^\pi$  inicial. O máximo de pilhas abertas para a sequência  $\pi = \{2, 1, 3, 4\}$  é 2, e portanto, esta sequência retorna uma solução melhor que a solução obtida para a sequência  $\pi = \{1, 2, 3, 4\}$  anterior.

$$\mathbf{Q}^{\{2,1,3,4\}} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 \end{bmatrix}.$$

Esta apresentação do MOSP permite visualizar sua semelhança com o problema *cutwidth* modificado. Para resolver o *cutwidth problem*, os vértices do grafo  $G = (V, E)$  são arranjados numa sequência  $\pi$  dos vértices  $\{1, \dots, N\}$ , de maneira que, para todo  $j \in [1, |V|]$  há no máximo  $K$  arestas com um vértice final entre as posições  $[1, j]$  da sequência  $\pi$  e o outro entre as posições  $(j, |V|]$ . Para todas as sequências  $\pi \in \Gamma$  é calculado um  $K$  máximo e dentre estes o menor  $K$  corresponde ao *cutwidth* do grafo.

No *cutwidth* modificado, os vértices do grafo são arranjados numa sequência  $\pi$ , de maneira que, para todo  $j \in (1, |V|)$  há no máximo  $K$  arestas com um vértice final entre as posições  $[1, j]$  da sequência  $\pi$  e o outro entre as posições  $(j, |V|]$ . Para todas as sequências  $\pi \in \Gamma$  é calculado um  $K$  máximo e dentre estes o menor  $K$  corresponde ao *cutwidth* do grafo. O *cutwidth* modificado pode ser escrito na forma

$$\text{minimizar}_{\pi \in \Gamma} (Z_{MCUT}^\pi(G)),$$

em que,

$$Z_{MCUT}^\pi(G) = \text{maximizar}_{1 < j < |V|} (|\{(u, v) \in E : \pi(u) < j < \pi(v)\}|).$$

É possível entender o *cutwidth* modificado como uma linha ordenada de vértices, em que, um plano ortogonal cruza a reta na posição  $j$ . A ordem dos vértices na reta é dada pela sequência  $\pi$ .

O *cutwidth* modificado do grafo  $G = (V, E)$  pode ser resolvido pelo algoritmo MOSP mostrado no exemplo anterior. Para isso, o problema é representado numa matriz  $\mathbf{R}$  com  $|E|$  linhas e  $|V|$  colunas. As colunas da matriz  $\mathbf{R}$  correspondem aos vértices do grafo, e cada linha  $i$  da matriz representa uma aresta  $(u, v) \in E$ , de forma que,  $r_{i,u} = r_{i,v} = 1$  e  $r_{i,j} = 0$  para  $j \neq u$  e  $j \neq v$ .

A matriz  $\mathbf{Q}^\pi$  recebe as colunas da matriz  $\mathbf{R}$  conforme a sequência  $\pi$  dos padrões. Então,  $\forall j \in (1, |V|)$  verifica-se quais arestas  $(u, v)$  cruzam o plano ortogonal traçado na posição  $j$  da reta, ou seja,  $\pi(u) < j < \pi(v)$ . Se a aresta  $(u, v)$  referente a linha  $i$  da matriz  $\mathbf{Q}^\pi$  cruza o plano ortogonal que está em  $j$ , então  $q_{i,j}^\pi = 1$ .

A matriz  $\mathbf{R}$  seguinte corresponde ao grafo do problema da matriz  $\mathbf{A}$  anterior. O grafo é construído, de forma que, os vértices indicam as colunas da matriz  $\mathbf{A}$ . As linhas de  $\mathbf{R}$  correspondem respectivamente as arestas  $(1, 3)$ ,  $(1, 4)$  e  $(3, 4)$ , e as colunas correspondem respectivamente aos vértices 1, 2, 3 e 4.

$$\mathbf{R} = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 \end{bmatrix}.$$

As matrizes seguintes mostram a construção da matriz  $\mathbf{Q}^\pi$  para a sequência  $\pi = \{1, 2, 3, 4\}$ . Primeiro,  $\mathbf{Q}^\pi$  recebe as colunas da matriz  $\mathbf{R}$  conforme a ordem dos vértices da sequência  $\pi$ . Depois, conforme a definição de  $\mathbf{Q}^\pi$ ,  $q_{1,2}^\pi = q_{2,2}^\pi = q_{2,3}^\pi = 1$ . O *cutwidth* modificado do grafo da matriz  $\mathbf{A}$  para a sequência  $\pi = \{1, 2, 3, 4\}$  é  $K = 3$ .

$$\mathbf{Q}^{\{1,2,3,4\}} = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 \end{bmatrix} \implies \mathbf{Q}^{\{1,2,3,4\}} = \begin{bmatrix} 1 & \boxed{1} & 1 & 0 \\ 1 & \boxed{1} & \boxed{1} & 1 \\ 0 & 0 & 1 & 1 \end{bmatrix}.$$

Assim como o MOSP, muitos problemas de otimização combinatorial satisfazem a propriedade de 1s-consecutivos. Outro problema relacionado ao MOSP surge nos projetos VLSI e é conhecido como GMLP (do inglês, *gate matrix layout problem*). Este problema é representado por um circuito constituído de linhas verticais (*gates*) onde ficam os transistores (indicados por pontos) utilizados para unir os *gates*. Esta união é feita por linhas horizontais conhecidas como *nets*, que conectam os *gates* pelos pontos posicionados num mesmo nível.

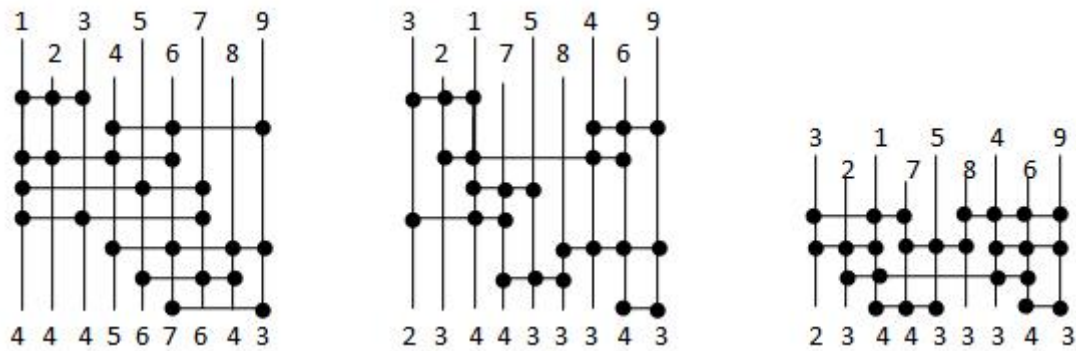
Este problema pode ser representado matricialmente considerando como colunas os *gates* e como linhas os possíveis níveis para posicionar os pontos. Assim, na matriz  $\mathbf{A}$ ,  $a_{i,j} = 1$  se o *gate*  $j$  tem um transistor no nível  $i$ , e  $a_{i,j} = 0$  caso contrário. Ao adicionar os *nets*, esta matriz satisfaz a propriedade de 1s-consecutivos.

No contexto do problema MOSP definido anteriormente, colocar os *nets* resulta na matriz  $\mathbf{Q}^\pi$  para uma sequência  $\pi$  dos *gates*, em que o objetivo é minimizar o número de linhas horizontais do circuito, ou seja,

$$\text{minimizar}_{\pi \in \Gamma} (\text{maximizar}_{j=1, \dots, N} \sum_{i=1}^M q_{i,j}^\pi).$$

A Figura 2.2, retirada do artigo de Linhares e Yanasse (2002), ilustra o funcionamento do GMLP. O problema inicial é representado pelo primeiro esquema da figura, em que, na parte superior esta indicada a ordem dos *gates* e na parte inferior a soma de *nets* que cruzam cada *gate*. Nesta sequência  $\{1, 2, 3, 4, 5, 6, 7, 8, 9\}$  dos *gates*, a soma máxima nas colunas é 7 e portanto, é possível desenhar este esquema em 7 níveis. O segundo esquema mostra uma nova ordenação dos *gates* com soma máxima igual a 4, portanto, este esquema pode ser desenhado com apenas 4 níveis, conforme mostra o terceiro esquema desta figura.

Linhares e Yanasse (2002) listam uma série de problemas relacionados ao problema MOSP que têm sido estudados de maneira independente pelos pesquisadores: *One-dimensional logic*, *PLA folding*, *Interval thickness*, *Node search game*, *Edge search game*, *Narrowness*, *Split bandwidth*,



**Figura 2.2:** Exemplo do funcionamento do problema GMLP.

*Graph path-width, Edge separation, Vertex separation.* Estes problemas não serão analisados neste trabalho. Para maiores detalhes verificar as referências indicadas em Linhares e Yanasse (2002).

## Yanasse e Pinto (2003); Pinto (2004); Yanasse e Lamosa (2007)

Em Yanasse e Pinto (2003) é proposto um novo modelo para o MOSP que é comparado com o modelo de Yanasse (1997a) e com uma extensão do modelo descrito por Laporte et al. (2004). Estes dois últimos modelos da literatura determinam a ordem em que os padrões devem ser processados para que o número máximo de pilhas abertas seja minimizado. Já o novo modelo descrito, procura determinar a ordem em que as pilhas são fechadas, da mesma forma que a enumeração de itens proposta em Yanasse (1997a). O desempenho computacional observado nos experimentos realizados mostrou-se, de uma maneira geral, superior em relação às formulações de Yanasse (1997a) e a extensão de Laporte et al. (2004). O modelo Yanasse e Pinto (2003) será definido e analisado no Capítulo 3 onde serão discutidas algumas melhorias.

O trabalho de Yanasse e Lamosa (2007) apresenta uma formulação matemática para o problema de corte integrado ao problema de sequenciamento de padrões. Este modelo é decomposto em dois problemas por relaxação lagrangiana. O problema dual é resolvido por um método do subgradiente modificado. O problema de corte foi resolvido pelo método de geração de colunas e o problema de sequenciamento por um procedimento de enumeração dos itens.

Pinto (2004) reúne, num só trabalho, as estratégias de solução apresentadas em Yanasse e Pinto (2003) e Yanasse e Lamosa (2007). Além disso, são mostrados outros resultados direcionadas ao problema de sequenciamento de padrões, que foi bastante analisado pela autora. Sabendo que o modelo de Yanasse e Lamosa (2007) é computacionalmente difícil de resolver, foram estudadas algumas variações e novos métodos de solução foram propostos. Além disso, o modelo MTSP foi formulado como um problema de fluxo em rede e obteve resultados satisfatórios mas, não pôde ser estendido para um problema MOSP.

## Becceneri et al. (2004)

No trabalho de Becceneri et al. (2004) o problema MOSP é representado por um grafo MOSP de itens e é resolvido com um método heurístico e um método de enumeração por itens. São apresentados métodos de pré-processamento para tentar simplificar o problema e também são discutidos alguns limitantes inferiores para o MOSP.

O primeiro método de pré-processamento utilizado tem como objetivo eliminar redundâncias. Uma redundância ocorre quando um padrão  $i$  domina um padrão  $j$ , ou seja, todos os tipos de itens contidos no padrão  $j$  também estão contidos no padrão  $i$ . Neste caso, o padrão  $j$  pode ser processado logo após o padrão  $i$  sem aumentar o número de pilhas abertas. O outro método de pré-processamento consiste em procurar por árvores, ciclos ou 1-árvores no grafo resultante. A ideia é explorar a estrutura do grafo.

Para calcular um limitante superior do problema é proposta a heurística de nó de custo mínimo (HNCM). Esta heurística tem sido utilizada no trabalho de diversos autores devido à qualidade da solução que é obtida para o MOSP. A HNCM está baseada no percorrimento de arestas do grafo MOSP de itens com o objetivo de minimizar a quantidade de vértices que permanecem simultaneamente abertos. O percorrimento das arestas é feito escolhendo-se as arestas que são adjacentes aos vértices de menor grau. Um vértice é dito aberto se ele já foi visitado, mas nem todas as arestas incidentes a ele foram percorridas. A pilha correspondente a um determinado tipo de item (vértice) é considerada fechada quando todas as arestas incidentes a este vértice forem percorridas. A solução obtida pela heurística corresponde a um conjunto ordenado de arestas que determina a ordem de fechamento dos vértices. Os autores então obtêm a sequência de padrões correspondente através do percorrimento destas arestas. A heurística HNCM é representada no seguinte algoritmo, em que

- $G(V, E)$ : grafo MOSP de itens.
- $degree(v)$ : grau do vértice  $v \in V$ .
- $SetV$ : conjunto ordenado dos vértices  $v$  ainda não sequenciados (isto é,  $degree(v) > 0$ ). Os vértices são ordenados de forma não decrescente do valor  $degree(v)$ .
- $SetV_{min}$ : conjunto com os vértices de menor grau em  $SetV$ .
- $Arc$ : conjunto ordenado das arestas de  $E$  já percorridas.
- $Open$ : conjunto de vértices que estão abertos (isto é, ao menos uma aresta incidente a  $v \in E$ , mas não todas, está em  $Arc$ ).
- $Z$ : número máximo de pilhas abertas.

1. Inicializa  $SetV$ ,  $Open \leftarrow \{\}$ ,  $Z = 0$  e  $Arc \leftarrow \{\}$ .
  2. Enquanto  $|Arc| \neq |E|$  faça:
    - 2.1 Procura dois vértices adjacentes  $v_1, v_2 \in SetV_{min}$  tal que  $(v_1, v_2) \notin Arc$ .
    - 2.2 Se não existe  $v_1, v_2$ , percorra o conjunto ordenado  $SetV - SetV_{min}$  e seleciona o primeiro vértice  $v_2 \in SetV - SetV_{min}$  adjacente a algum vértice  $v_1 \in SetV_{min}$  tal que  $(v_1, v_2) \notin Arc$ .
    - 2.3  $Open \leftarrow Open \cup \{v_1, v_2\}$ .
    - 2.4 Se  $Z < |Open|$  então  $Z = |Open|$ .
    - 2.5  $Arc \leftarrow Arc \cup \{(v_1, v_2)\}$ .
    - 2.6 Para  $v = v_1, v_2$  faça:
      - Atualiza  $degree(v)$ .
      - Se  $degree(v) = 0$  então  $Open \leftarrow Open - \{v\}$ .
    - 2.7 Enquanto  $\exists v_i, v_j \in Open$  tal que  $(v_i, v_j) \in E$  e  $(v_i, v_j) \notin Arc$  faça:
      - $Arc \leftarrow Arc \cup \{(v_i, v_j)\}$ .
      - Para  $v = v_i, v_j$  faça:
        - Atualiza  $degree(v)$ .
        - Se  $degree(v) = 0$  então  $Open \leftarrow Open - \{v\}$ .
    - 2.8 Atualiza  $SetV$  e  $SetV_{min}$ .
  3. Retorna  $Z$  e  $Arc$ .
- 

Para calcular o limitante inferior do problema MOSP é aplicado um procedimento de contração de arcos (HCA). No grafo MOSP de itens  $G(V, E)$  seja  $v_j$  o vértice com menor grau em  $G$ . Para fechar o vértice (completar a pilha)  $v_j$ , todos os vértices  $v_i$  tal que  $\exists (v_i, v_j) \in E$  também serão abertos. Nesse instante, haverá no mínimo as pilhas dos itens  $v_i$  mais a pilha de  $v_j$  abertas. Dessa relação vem o limitante inferior  $lb_d = 1 + \min\{degree(v), \forall v \in V\}$ . Outro limitante para o MOSP surge do fato de que a solução MOSP não é menor que a quantidade máxima de tipos de itens num padrão:  $lb_p = \max\{|P_j|, j = 1, \dots, N\}$ .

A ideia da estratégia de contração é reduzir recursivamente o grafo  $G$ , até que um grafo completo seja obtido. A cada iteração da heurística seleciona-se um arco  $a = (v_i, v_j) \in E$ . Este arco é contraído em  $G$ , ou seja, juntam-se os vértices  $v_i$  e  $v_j$  num único vértice  $v_i|v_j$  de maneira que os vértices antes adjacentes a  $v_i$  e  $v_j$  agora são adjacentes do novo vértice  $v_i|v_j$ . Se é obtido um grafo  $G'(V', E')$  completo com ordem  $|G'|$  maior que o valor  $lb = \max\{lb_p, lb_d\}$ , obtêm-se o valor do limitante inferior  $lb_c = |G'|$ . Caso contrário  $lb_c = lb$ , sendo que o valor de  $lb$  pode ser atualizado no decorrer da estratégia recebendo o valor do menor grau dos vértices do grafo mais 1. O procedimento para obter o limitante inferior pode ser escrito conforme o algoritmo seguinte.

---



**HLI - Cálculo do limitante inferior (Becceneri et al., 2004)****Passo 0:** Inicialização

1. Seja  $G$  o grafo MOSP de itens e  $|G|$  a ordem de  $G$ .
2.  $lb_p = \max\{|P_j|, j = 1, \dots, N\}$ .
3.  $lb_d = 1 + \min\{\text{degree}(v), \forall v \in G\}$ .
4.  $lb = \max\{lb_p, lb_d\}$ .

**Passo 1:** Teste de parada

1. Se ( $G$  é completo ou  $|G| \leq lb$ ), então  $lb_c = lb$ . FIM.
2. Senão, seja  $S$  o conjunto de vértices de  $v \in G$  ordenados de forma não decrescente, conforme o valor de  $\text{degree}(v)$ .

**Passo 2:** Contração do arco

1. Escolha um arco  $a = (v_i, v_j) \in G$ , tal que  $v_i$  e  $v_j$  tenham os menores graus em  $S$ .
2.  $G \leftarrow G - \{a\}$ .
3.  $dmin = \min\{\text{degree}(v), \forall v \in G\}$ .  
Se ( $dmin \geq lb$  e  $dmin + 1 > lb$ ), então  $lb = dmin + 1$ .
4. Volta para Passo 1.

No método de enumeração deste trabalho é considerada uma relação de equivalência entre os itens. Seja o grafo MOSP de itens  $G(V, E)$  e  $N(i)$  conjunto de vértices adjacentes ao vértice  $i \in V$ . A relação de equivalência é dada por: se  $\exists i, j \in V$  tal que  $i \neq j$  e  $N(i) = N(j)$  então, dada uma solução factível do problema em que  $i$  e  $j$  não ocupam posições consecutivas na sequência de itens, é possível construir uma nova sequência de itens em que  $i$  e  $j$  aparecem em posições consecutivas indiferentemente da ordem. Além disso, o valor da solução correspondente à nova sequência de itens é menor ou igual ao valor na sequência original.

Considerando o grafo  $G$  e os vértices  $i$  e  $j$  que satisfazem a relação de equivalência. É possível eliminar o vértice  $i$  (ou o vértice  $j$ ) e todas as arestas incidentes a este vértice. No método exato, os vértices equivalentes são mapeados e eliminados antes de executar o método de enumeração por itens. No final os itens eliminados são inseridos na sequência obtida na enumeração. A relação de equivalência é também classificada, por alguns autores, como um método de pré-processamento.

No método de enumeração, a busca pela solução ótima começa por uma busca em profundidade e considera cortes baseados nos valores dos limitantes inferior e superior em cada nó, conforme descrito em Yanasse (1997a). Os testes computacionais realizados pelos autores mostraram a qualidade das soluções obtidas com a heurística HNCM proposta. E comparando os dois métodos de enumeração, o original e o melhorado, foi constatado ganho no tempo de execução aplicando a nova estratégia.

## Smith e Gent (2005)

Em 2005, o MOSP foi tema do desafio “*Constraint Modelling Challenge 2005*” e os trabalhos que concorreram ao prêmio e as estratégias utilizadas pelos concorrentes são reportados em Smith e Gent (2005).

Nos trabalhos encaminhados ao desafio foram utilizadas estratégias de pré-processamento que, em geral, parecem contribuir bastante na eficácia e rapidez na solução do problema. A relação de dominância entre os padrões, previamente utilizada em Becceneri et al. (2004), foi considerada por vários autores. Outra estratégia de pré-processamento utilizada por de la Banda e Stuckey (2005) e Simonis (2005) consiste em verificar se é possível decompor o problema original em subproblemas em que os padrões de um subproblema não têm nenhum item comum com os padrões do outro subproblema. Esta relação foi proposta em Yuen e Richardson (1995).

Miller (2005) provou algumas relações para calcular os limitantes inferiores. Em de la Banda e Stuckey (2005) foram utilizados alguns limitantes encontrados em Becceneri et al. (2004) e propostos outros. Nos trabalhos de Pesant (2005) e Shaw e Laborie (2005), o limitante inferior é obtido utilizando a teoria do problema de coloração de grafos. Em Simonis (2005) são resolvidos pequenos subproblemas com 3 a 5 padrões para calcular o limitante inferior.

Os métodos propostos para resolver o MOSP são bem diversificados e englobam métodos de programação inteira, de busca local, de programação dinâmica e, como dizem os organizadores, de modelos de programação restrita, que cobrem os métodos propostos de coloração de grafo, de permutação dos itens e de sequenciamento.

Os experimentos computacionais mostram que o sequenciamento dos itens ao invés dos padrões é melhor, confirmando assim ideias que já vinham sendo defendidas na literatura por alguns autores. Além disso, mostram o ganho de tempo com o uso dos métodos de pré-processamento. Alguns autores tentaram tratar as relações de simetria que dificultam bastante a solução do problema MOSP, porém, nenhuma estratégia obteve muito êxito.

O melhor dentre os métodos propostos foi apresentado em de la Banda e Stuckey (2005). Consiste num método exato de programação dinâmica e utiliza os dois métodos de pré-processamento destacados anteriormente. O limitante inferior é calculado conforme Becceneri et al. (2004) e o limitante superior é calculado com um conjunto de heurísticas. Este método será apresentado em mais detalhes na continuação deste capítulo.

## Yanasse (2002); Yanasse et al. (2007)

No trabalho Yanasse et al. (2007) foi implementado um método de enumeração por itens com ordenação parcial proposto em Yanasse (2002). No trabalho original não foram realizados experimentos computacionais para avaliar o desempenho deste método.

Seja  $G(V, E)$  o grafo MOSP de itens e  $N(i) \subseteq V$  o conjunto de vértices adjacentes ao vértice  $i$ . São definidos dois tipos de ordenamento parcial obtidos a partir do grafo MOSP de itens:

1. Sejam os vértices  $i, j \in V$  tal que  $i \in N(j)$  e  $|N(i)| = |N(j)| = 2$ . Sempre existe uma solução ótima para o MOSP em que as pilhas de  $i$  e  $j$  são completadas uma imediatamente após a outra.
2. Sejam os vértices  $i, j \in V$ . Se  $N(j) \subset \{N(i) \cup \{i\}\}$  então existe uma dominância do vértice  $i$  sobre o  $j$ . Sempre existe uma solução ótima para o MOSP em que a pilha  $j$  é completada antes da pilha  $i$ .

É possível ainda escrever uma relação de equivalência entre os vértices que também pode ser analisada na ordenação parcial:

3. Sejam os vértices  $i, j \in V$ . Se  $N(i) = N(j)$  ou  $N(i) \cup \{j\} = N(j) \cup \{i\}$  então existe uma relação de equivalência entre os vértices  $i$  e  $j$ . Sempre existe uma solução ótima para o MOSP em que as pilha  $i$  e  $j$  são completadas uma imediatamente após a outra.

De (1), (2) e (3) são geradas relações entre os itens que sugerem qual a ordem que devem ser analisados durante a enumeração. O método de enumeração de Becceneri et al. (2004) considera apenas a relação (3).

Na estratégia de solução proposta, a solução inicial do problema é obtida com a heurística de nó de custo mínimo de Becceneri et al. (2004). Para a enumeração da árvore de busca é adotada uma estratégia de busca em profundidade seguindo sempre o nó mais promissor (busca gulosa). Os nós são cortados comparando-se o valor do limitante inferior do nó e o valor do limitante superior que corresponde à melhor solução obtida na busca.

Os experimentos computacionais mostraram que o método de enumeração parcial é mais eficiente que o método exato de Becceneri et al. (2004), com o qual foi comparado.

## de la Banda e Stuckey (2005, 2007)

de la Banda e Stuckey (2005, 2007) apresentam um método de programação dinâmica para resolver o problema de minimização do número máximo de pilhas abertas (MOSP). Os cortes aplicados durante a busca consideram relações entre os valores dos limitantes inferior e superior e relações de dominância entre padrões.

Seja  $P$  o conjunto dos padrões do problema,  $S$  o conjunto dos padrões que ainda não foram processados e  $c(p)$  o conjunto dos tipos de itens do padrão  $p$ . Considerando que o padrão  $p$  é sequenciado imediatamente antes de qualquer padrão do conjunto  $A \subset P$ , o número de pilhas abertas no momento em que  $p$  é processado pode ser definido como:

$$a(p, A) = c(p) \cup (c(A) \cap c(P - A - \{p\})), \quad (2.13)$$

ou seja, as pilhas necessárias ao processar  $p$  mais as pilhas abertas antes de  $p$  e que ainda serão necessárias no corte dos padrões seguintes à  $p$ . A programação dinâmica segue a relação  $stacks(S)$  seguinte, que indica o número mínimo de pilhas necessários para processar o conjunto de padrões  $P - S$  já processados.

$$stacks(S) = \underset{p \in S}{\text{minimizar}} (\max\{a(p, S - \{p\}), stacks(S - \{p\})\}). \quad (2.14)$$

Para cada conjunto  $S$  são analisadas duas relações, a primeira consiste numa relação de dominância entre padrões e a segunda fornece o valor do limitante inferior para o conjunto  $S$ .

1. Se  $\exists j \in S$  tal que as pilhas de todos os itens em  $c(j)$  estão abertas, isto é  $c(j) \subseteq (c(P - S) \cap c(S))$  então existe uma sequência ótima em  $S$  que começa com o padrão  $j$ .
2. Seja  $G_P$  o grafo MOSP de itens correspondente ao conjunto de padrões  $P$  e  $degree_{G_P}(c)$  o grau do vértice  $c$  no grafo  $G_P$ . O número mínimo de pilhas necessárias para o conjunto de padrões  $S$  é no mínimo  $b(S) = |(c(P - S) \cap c(S))| + \min \{degree_{G_S}(c) + 1, \forall c \in c(S)\}$ .

Com o resultado de (2) é possível melhorar o limitante inferior do número de pilhas necessárias para processar o próximo padrão  $p$ , definido em 2.13:

$$a'(p, A) = \max\{a(p, A), b(A)\}. \quad (2.15)$$

Os autores também utilizam técnicas de pré-processamento no problema inicial, já utilizadas em Becceneri et al. (2004). Uma delas é o de dominância entre padrões para a qual também é apresentada uma demonstração. O outro pré-processamento aplicado consiste em particionar o conjunto  $P$  de padrões subconjuntos  $P_1, P_2, \dots, P_n$ , tal que  $P = P_1 \cup P_2 \cup \dots \cup P_n$  e  $c(P_1) \cap c(P_2) \cap \dots \cap c(P_n) = \emptyset$  (Yuen e Richardson, 1995).

O limitante inferior do problema MOSP foi obtido aplicando o método de Becceneri et al. (2004) que é demonstrada por de la Banda e Stuckey (2007). Para calcular um limitante superior foram analisados 11 métodos heurísticos, alguns destes já propostos na literatura como a heurística de nó de custo mínimo (Becceneri et al., 2004) e a heurística 3 de Yuen (Yuen, 1995). Comparando as 11 heurísticas em relação ao número máximo de pilhas abertas, a heurística de nó de custo mínimo obteve os melhores resultados, seguida pela heurística 5.

A heurística 5 é proposta por de la Banda e Stuckey (2005, 2007) e consiste em selecionar o padrão  $p$  com o menor número máximo de pilhas abertas calculado com 2.15, ou seja, seleciona  $p$  tal que:

$$\underset{p \in S}{\text{min}} a'(p, S - \{p\}). \quad (2.16)$$

A ordem em que os padrões  $p \in S$  são analisados durante a enumeração influencia bastante no tempo de solução do método. Ao aplicar o método exato para resolver as classes de instâncias propostas em Smith e Gent (2005), apenas em três instâncias não foi obtida uma solução ótima. Nestas instâncias foi atingido o limite de chamadas do procedimento.

## Chu e Stuckey (2009)

Seja o problema de decisão  $LOSP(k)$  (do inglês, *Limited Open Stacks Problem*), que é factível para um valor fixo de  $k$  se existe uma sequência de padrões que gera no máximo  $k$  pilhas abertas durante o processo de corte. O método de solução proposto em de la Banda e Stuckey (2005, 2007) consiste em obter uma solução do MOSP resolvendo uma sequência de problemas  $LOSP(k)$ . O valor de  $k$  é reduzido toda vez que é encontrada uma solução para o problema. O método aplica uma enumeração por padrões.

No artigo de Chu e Stuckey (2009) é proposto um método de solução do MOSP que combina as estratégias de programação dinâmica e enumeração, de maneira semelhante ao método proposto em de la Banda e Stuckey (2005, 2007) porém, considera a ideia de sequenciar o fechamento das pilhas ao invés da enumeração por padrões anteriormente estudada.

Seja

- $P$ : conjunto dos padrões do problema.
- $c(P)$ : conjunto dos itens contidos nos padrões  $P$ .
- $N(\alpha)$ : conjunto de itens que estão nos padrões que contém o item  $\alpha$ .
- $S$ : conjunto de itens que foram fechados.
- $O(S) = \cup_{\alpha \in S} N(\alpha)$ : conjunto de itens que foram abertos ao completar os itens em  $S$ .
- $o(\alpha, S) = N(\alpha) - O(S)$ ,  $\forall \alpha \notin S$ : conjunto de itens que serão abertos se  $\alpha$  for o próximo item a ser fechado.

A quantidade de pilhas que permanecem simultaneamente abertas ao completar a pilha do item  $\alpha$  é dada por  $|O(S) - S \cup o(\alpha, S)|$ . O problema  $LOSP(k)$  é infactível se  $|O(S) - S \cup o(\alpha, S)| > k$ , ou seja,  $\alpha$  não pode ser a próxima pilha a ser completada e diz-se que a sequência de itens  $S \cup \{\alpha\}$  não é viável. Outro item é analisado e se ocorrer que  $|O(S) - S \cup o(\alpha, S)| > k, \forall \alpha \in P - S$  a sequência  $S \cup \{P - S\}$  é infactível. A estratégia volta para o nó factível (solução mosp menor ou igual a  $k$ ) mais próximo. O  $LOSP(k)$  é resolvido para  $k$  variando de  $|C| - 1, \dots, 1$ .

As subsequências  $S$ , conforme são analisadas, ficam armazenadas e rotuladas como viáveis ou inviáveis. Uma subsequência que no início da busca é rotulada como viável pode num momento futuro ser atualizada como inviável.

O algoritmo pode ser melhorado ao considerar a seguinte relação de dominância entre os itens. Se  $o(\alpha_i, S) \subseteq o(\alpha_j, S)$  tal que  $\alpha_i, \alpha_j \notin S$  e  $i < j$ , então é possível sequenciar  $i$  antes de  $j$  ao invés de sequenciar  $j$  imediatamente. Baseando-se na relação de dominância entre itens e considerando  $f(\alpha_j, S)$  o conjunto dos itens  $\alpha_i$  que satisfazem esta relação, os autores apresentam o seguinte teorema.

**Teorema 1.** *Seja  $|f(\alpha_j, S)| \geq |o(\alpha_j, S)|$  e  $S \cup \{\alpha_j\}$  uma sequência viável, então se existe uma solução  $U' = S \cup \{P - S\}$  para o problema  $LOSP(k)$  existe também uma solução  $U = S \cup \{\alpha_j\} \cup \{P - S - \{\alpha_j\}\}$ .*

Aplicando o Teorema 1 no método de enumeração ocorre que o item da sequência  $S$  será o  $\alpha_j$  e os demais nós, em que  $\alpha_i \in \{P - S\}$  e  $\alpha_j \neq \alpha_i$ , são cortados da árvore de busca.

Outra ideia aplicada no método de enumeração consiste em analisar um conjunto de padrões promissores para a enumeração, em vez de considerar apenas um padrão  $p$ . Cada um dos nós promissores é analisado, por exemplo, para  $m$  nós sucessores. Os autores sugerem considerar como padrão promissor aquele que, se sequenciado, fecha ao menos uma das pilhas anteriormente abertas. Os resultados obtidos mostram que aplicar esta estratégia reduz o tempo de busca de uma solução ótima do problema.

Os autores apresentam uma heurística para relaxar o problema MOSP que é baseada na ideia de contração das arestas no grafo MOSP de itens utilizada em Becceneri et al. (2004). Seja  $G'$  o grafo que é atualizado toda vez que dois vértices são contraídos, o vértice  $c \in G'$  e

$$F(c) = \sum_{c' \in N(c)} \left( \frac{|N(c) - N(c')|}{|N(c)|} \right) \quad (2.17)$$

uma média ponderada da quantidade de vértices em  $N(c)$  que não são adjacentes aos vértices em  $N(c')$ . Com  $F(c)$  tenta-se medir o quão semelhante é a vizinhança dos vértices  $c$  e  $c'$  no grafo  $G'$ .

Na heurística de relaxação são aplicadas etapas de contração e de expansão do grafo  $G'$ . A técnica de contração consiste em contrair dois vértices  $i, j \in G'$  tal que  $F(i) = \max\{F(c), \forall c \in G'\}$  e  $|N(i) - N(j)| = \max\{|N(i) - N(c')|, \forall c' \in G', c' \in N(i)\}$ .

Em relação aos experimentos computacionais realizados foram geradas instâncias com até 100 padrões e 100 itens. O método de enumeração proposto por Chu e Stuckey (2009) se destaca por resolver as instâncias SP2, SP3, SP4 propostas em Smith e Gent (2005) que ainda não tinham sido resolvidas por outros métodos exatos.

## Ashikaga (2001); Ashikaga e Soma (2009)

O grafo MOSP de itens é utilizado e foi observado que, na maioria das instâncias testadas no trabalho, o grafo complementar é esparso e tem um conjunto independente maximal grande se comparado ao número de vértices do grafo. O foco do trabalho foi analisar o problema utilizando seu conjunto independente maximal e clique maximal.

Os problemas de detecção do clique maximal e do conjunto independente maximal são NP-difíceis (Garey e Johnson, 1979). Este fato, encaminhou o trabalho para o desenvolvimento de dois métodos heurísticos que partem do cálculo do clique maximal, encontrado através da busca gulosa de vértice de grau mínimo no grafo complementar do grafo MOSP de itens.

Estudos anteriores mostraram que a busca de um clique maximal tende a funcionar muito bem em grafos em que o tamanho do clique maximal é próximo ao número de vértices do grafo, que é o caso dos grafos MOSP analisados. E nos experimentos realizados pelos autores, aproximar o clique maximal através da busca gulosa pelo vértice de grau mínimo mostrou ser uma estratégia viável.

Em Ashikaga (2001) são propostos dois métodos para calcular a sequência dos itens do problema MOSP e a partir dela, obter a sequência de processamento dos padrões. Para obter uma sequência de padrões a partir da sequência de itens, os autores utilizam um procedimento simples, que consiste em percorrer a sequência de tipos de itens do final para o início. Os padrões que contém cada item percorrido são inseridos na sequência de padrões, também do final para o início, considerando cada padrão uma única vez.

Na primeira heurística, chamada HER (Heurística extensão-rotação) são encontrados ciclos hamiltonianos através de um algoritmo extensão-rotação no qual, o circuito hamiltoniano é composto pelos vértices do clique maximal. O número permitido de rotações é fixo e estas rotações só podem ser feitas até um certo número de extensões ser calculado. A segunda heurística chamada HCC (Heurística de contração recursiva dos cliques), consiste em encontrar um clique maximal do grafo MOSP de itens e contrair todos os vértices deste clique em um único vértice, denominado hipernó. Em seguida, um novo clique maximal que inclua o hipernó é obtido a partir do grafo resultante. Este procedimento continua até o grafo ser todo contraído em um único hipernó. A cada contração, os vértices que formam o clique são inseridos em uma sequência de tipos de itens, a qual determina a solução obtida pela heurística.

A heurística HCC é descrita a seguir. Seja  $K_i$  o conjunto dos vértices que formam o clique maximal do grafo  $G_i$ .

---

### HCC - Heurística de contração recursiva de cliques (Ashikaga, 2001)

---

1. Seja  $G_0(V, E)$  o grafo MOSP de itens.
  2. Inicializa a sequência de itens  $S_\pi = \{\}$  e  $i = 0$ .
  3. Contração dos cliques.
    - 3.1 Encontra o clique maximal  $K_i$  tal que, se  $i > 0$  então  $h_{i-1} \in K_i$ .
    - 3.2  $S_\pi \leftarrow S_\pi \cup K_i$ .
    - 3.2 Contraí o clique maximal num hipernó  $h_i$  e retorna o grafo  $G_{i+1}$ .
    - 3.3 Se  $|G_{i+1}| = 1$  então vá para o passo 4.
    - 3.4 Senão,  $i = i + 1$  e vá para o passo 4.1.
  4. Retorna uma sequência de itens  $S_\pi$ .
- 

Os experimentos computacionais realizados pelo autor compararam estas duas heurísticas propostas com a heurística 3 (YUEN3) de Yuen (1995) e com a heurística de Yanasse et al. (1999).

A heurística HER foi a mais rápida mas, obteve solução para o MOSP pior que a heurística de Yanasse et al. (1999). Num segundo conjunto de testes com instâncias maiores, foram comparadas apenas as heurísticas HER e de Yuen. A HER foi muito mais rápida. Quanto ao número de pilhas obtido, a HER foi melhor em 87 das 90 classes testadas. Estas 3 classes em que a HER não foi a melhor, foram as classes com instâncias mais esparsas.

Em Ashikaga e Soma (2009) é proposta uma nova estratégia de solução para o MOSP que faz uso da heurística HER (Ashikaga, 2001). A heurística proposta (AS) é comparada com a heurística de nó de custo mínimo (HNCM) de Becceneri et al. (2004) e a YUEN3. HNCM apresenta a solução com número máximo de pilhas abertas melhor que as demais heurísticas, seguida pela heurística AS. Porém, HNCM utiliza mais tempo computacional para resolver as instâncias consideradas. Comparando as heurísticas YUEN3 e AS em relação ao tempo computacional, ambas são muito rápidas entretanto, AS apresentou desempenho significativamente melhor que YUEN3 conforme aumenta o número de itens na instância.

## Yanasse e Senne (2010)

Em Yanasse e Senne (2010) são discutidos métodos de pré-processamento que têm como objetivo reduzir o tamanho (número de itens ou padrões) das instâncias MOSP tratadas ou identificar condições particulares da instância. A aplicação destas técnicas tende a reduzir o esforço necessário para a solução das instâncias.

Instâncias em que os grafos são árvores, 1-árvores ou grafos completos são facilmente resolvidos (como visto em Yanasse (1996) e Yanasse (1997a)).

Outro método de pré-processamento consiste em verificar a existência de subgrafos disjuntos no grafo MOSP de padrões. Cada um destes subgrafos pode ser resolvido de maneira independente. O valor ótimo da solução do grafo original é o maior valor MOSP obtido entre os subgrafos e a sequência de corte dos padrões obedece a sequência de padrões obtida em cada subgrafo. Esta técnica foi sugerida em Yuen e Richardson (1995).

É possível reduzir o número de padrões de uma instância analisando relações de dominância entre eles. Um padrão é dito dominado quando todos os tipos de itens deste estão contidos também em outro padrão da instância. Os padrões dominados são retirados da instância e serão inseridos na sequência de padrões obtida como solução da instância resultante. O padrão dominado é adicionado logo após o padrão que o dominou, sem aumentar o número máximo de pilhas abertas da sequência de padrões. A dominância entre padrões foi utilizada, por exemplo, em Yuen e Richardson (1995), Becceneri (1999), Becceneri et al. (2004) e de la Banda e Stuckey (2005, 2007).

Com o intuito de reduzir o número de itens da instância, são analisadas relações de adjacência dos vértices do grafo MOSP de itens. Numa destas técnicas de pré-processamento, uma aresta  $(i, j)$  é contraída se os vértices  $i$  e  $j$  tem grau dois e o vértice resultante  $i/j$  tem também grau dois.



Na solução final do grafo, o vértice  $i$  é fechado logo antes que o vértice  $j$  é aberto ou então,  $j$  é fechado logo antes que  $i$  é aberto (ver Yanasse (2002)).

Outra relação analisada considera  $ADJ(i)$  o conjunto de vértices do grafo MOSP de itens que são adjacentes ao vértice  $i$ . Se existem dois vértices distintos  $i$  e  $j$  tal que  $\{ADJ(i) \cup \{i\}\} = \{ADJ(j) \cup \{j\}\}$  então existe sempre uma solução ótima para o MOSP em que os vértices  $i$  e  $j$  são fechados um logo após o outro. Assim, é possível considerar apenas um dos vértices  $i$  ou  $j$  ao resolver o MOSP.

Relações de adjacência no grafo MOSP de itens também são utilizadas para reduzir o número de padrões da instância. Se existem dois vértices distintos  $i$  e  $j$  tal que  $ADJ(i) = ADJ(j)$  então existe sempre uma solução ótima para o MOSP em que  $i$  e  $j$  são fechados um imediatamente após o outro. Ou seja, os padrões que contêm o item  $j$  serão sequenciados logo após os padrões que contêm o item  $i$  ou vice-versa.

Redução do número de padrões pela determinação de uma instância equivalente. Para esse pré-processamento, os autores propõem um novo algoritmo que gera uma instância com menos padrões e é equivalente ao grafo de itens da instância original. No início do algoritmo todas as arestas do grafo original são classificadas como tipo 1 e enquanto houver arestas do tipo 1 é feito: i) Seleciona um padrão, entre os padrões originais restantes, correspondente ao clique com maior número de arestas do tipo 1; ii) determina um clique maximal que contém o padrão selecionado e armazena o clique (este clique corresponde a um padrão da instância equivalente); iii) classifica todas as arestas do clique como tipo 2. A nova instância equivalente é formada por padrões que correspondem aos cliques obtidos na etapa (ii) do algoritmo.

## de Carvalho e Soma (2011)

Neste trabalho são propostas simplificações em dois métodos da literatura: a Heurística de Extensão e Rotação (HER) de Ashikaga e Soma (2009) e o método de programação dinâmica de de la Banda e Stuckey (2007). Na verdade, o artigo não trabalha com o método de programação dinâmica, mas sim, com uma heurística utilizada no método exato.

No método de programação dinâmica (de la Banda e Stuckey, 2007) são utilizadas 11 heurísticas para calcular o limitante superior do método, sendo escolhido o melhor valor dentre os 11 obtidos. A simplificação proposta em de Carvalho e Soma (2011) consiste em utilizar uma destas heurísticas (a quinta na lista de de la Banda e Stuckey (2007), denominada DLB5, para resolver o MOSP, excluindo todo o restante do algoritmo exato proposto no método original. Nos experimentos apresentados em 2007 a heurística DLB5 obteve o segundo melhor desempenho dentre as 11 heurísticas utilizadas.

A simplificação aplicada na heurística HER não utiliza a heurística de extensão e rotação de Pósa (1976) considerada em HER. Os vértices remanescentes, que não são conjugados, são orde-

nados de forma não decrescente conforme o grau do vértice no grafo original, ao invés de analisar o grau no grafo atual.

Nos experimentos computacionais foram utilizadas as classes de instâncias apresentadas em Smith e Gent (2005) e comparados os resultados obtidos com a HER, sua simplificação HERS e a heurística DLB5. O valor da solução para o MOSP por cada um destes métodos foi comparado com o valor obtido com o método de programação dinâmica reportado em de la Banda e Stuckey (2007) e Chu e Stuckey (2009). A heurística DLB5 obteve o melhor resultado médio em relação ao número máximo de pilhas abertas, seguida por HERS. Em relação ao tempo computacional, HERS e DLB5 obtiveram tempos de execução médios abaixo de um milissegundo, inclusive nas classes maiores (com instâncias de 100 itens e 100 padrões).

---

# Modelo MOSP e modelo integrado MOSP/MORP

---

---

Neste capítulo tem-se por objetivo discutir aspectos das formulações matemáticas existentes para o MOSP e propor modificações e novos modelos. Inicialmente, o modelo de Yanasse e Pinto (2003) é analisado e novas desigualdades válidas são formuladas com o intuito de melhorar a convergência de métodos baseados na resolução da relaxação linear. Em seguida, um novo modelo é proposto. A principal característica deste novo modelo é o fato dele poder modelar simultaneamente os problemas MOSP e MORP.

## 3.1 Modelo MOSP de Yanasse e Pinto (2003)

O modelo MOSP apresentado por Yanasse e Pinto (2003) determina a ordem em que as pilhas dos itens são fechadas. Em cada estágio a pilha de algum tipo de item é completada, e para isso, todos os padrões que contêm deste item são processados e outros itens que estão nestes padrões também têm as suas pilhas abertas. Antes de fechar-se a pilha em cada estágio, somam-se todas as pilhas abertas até este estágio e desta soma subtrai-se o número de pilhas que já foram fechadas. O resultado deste cálculo é o número de pilhas abertas no estágio.

Um parâmetro utilizado no modelo é a matriz binária  $\mathbf{S}$  que indica quais pilhas são formadas ao completar cada item, ou seja,  $S_{ij} = 1$  se ao completar a pilha do item  $i$  são processados padrões que contêm também o item  $j$ . Desta forma, a pilha do item  $j$  é aberta no estágio em que a pilha  $i$  é completada ou em algum estágio anterior. Em seguida é definido o modelo MOSP de Yanasse e Pinto (2003).

Dados do problema:

- $M$ : número de tipos de itens do problema.
- $K$ : constante igual a  $M$ .
- $\mathbf{S}_{ij}$ : vale 1 se ao completar a pilha do item  $i$  a pilha do item  $j$  é aberta, e vale 0 caso contrário.

Variáveis do problema:

- $\mathbf{X}_{ij}$ : vale 1 se a pilha do item  $i$  é completada no estágio  $j$ , e vale 0 caso contrário.
- $\mathbf{W}_{ij}$ : vale 1 se a pilha referente ao item  $i$  foi aberta antes ou no estágio  $j$ , e vale 0 caso contrário.
- $\mathbf{C}$ : máximo de pilhas abertas nos estágios  $j, j = 1, \dots, M$ .

Formulação do problema MOSP:

$$\text{Minimizar } \mathbf{C} \tag{3.1}$$

Sujeito a:

$$\sum_{i=1}^M \mathbf{W}_{ij} \leq \mathbf{C} + j - 1 \quad j = 1, \dots, M \tag{3.2}$$

$$\sum_{i=1}^M \sum_{j=1}^t \mathbf{X}_{ij} \mathbf{S}_{ip} \leq K \mathbf{W}_{pt} \quad t = 1, \dots, M \text{ e } p = 1, \dots, M \tag{3.3}$$

$$\sum_{j=1}^M \mathbf{X}_{ij} = 1 \quad i = 1, \dots, M \tag{3.4}$$

$$\sum_{i=1}^M \mathbf{X}_{ij} = 1 \quad j = 1, \dots, M \tag{3.5}$$

$$\mathbf{X}_{ij} \in \{0, 1\} \quad i = 1, \dots, M \text{ e } j = 1, \dots, M \tag{3.6}$$

$$\mathbf{W}_{ij} \in \{0, 1\} \quad i = 1, \dots, M \text{ e } j = 1, \dots, M \tag{3.7}$$

$$1 \leq \mathbf{C} \leq M \quad \mathbf{C} \in \mathbb{R}. \tag{3.8}$$

O objetivo do modelo é minimizar o número máximo de pilhas abertas  $\mathbf{C}$ . As restrições (3.2) garantem que o número total de pilhas abertas até o estágio  $j$  menos o número de pilhas já completadas ( $j - 1$ ) limita o valor do número máximo de pilhas abertas  $\mathbf{C}$ . As restrições (3.3) indicam quais pilhas  $p$  são abertas para completar o item  $i$  de cada estágio  $t$ . As restrições (3.4) e (3.5) garantem que as pilhas de todos os itens são completadas, uma em cada estágio. Enfim, as restrições (3.6)-(3.8) definem as variáveis do problema.

Considerando a matriz  $\mathbf{A}$  seguinte, em que as linhas indicam os padrões e as colunas os itens. O padrão 1 contém itens do tipo 2, 4 e 5, o padrão 2 contém itens do tipo 1, 2 e 4, e assim por diante.

A matriz  $\mathbf{S}$ , que indica quais pilhas precisam ser abertas (colunas) ao completar a pilha de cada item (linha), é obtida a partir da matriz  $\mathbf{A}$ . Conforme a linha 1 desta matriz, para completar a pilha de itens do tipo 1 são abertas as pilhas de todos os demais itens e para completar a pilha do item 2 são abertas as pilhas referentes aos itens 1, 2, 4 e 5, e assim por diante.

$$\mathbf{A} = \begin{bmatrix} 0 & 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 0 & 0 \end{bmatrix}, \quad \mathbf{S} = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 1 & 1 \\ 1 & 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \end{bmatrix}.$$

Ao resolver o problema associado à matriz  $\mathbf{A}$  aplicando o modelo MOSP é obtida uma solução ótima dada pelas matrizes  $\mathbf{X}$  e  $\mathbf{W}$  seguintes. A matriz  $\mathbf{X}$  indica o estágio (coluna) em que a pilha de cada item (linha) é completada, por exemplo, no estágio 1 é completada a pilha referente ao item 2, no estágio 2 a pilha do item 4, e assim por diante.

$$\mathbf{X} = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{bmatrix}, \quad \mathbf{W} = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \end{bmatrix}.$$

A matriz  $\mathbf{W}$  indica a partir de qual estágio (coluna) a pilha de cada item (linha) é aberta. Como no estágio 1 é completada a pilha referente ao item 2, sabe-se pela matriz  $\mathbf{S}$  que as pilhas referentes aos itens 1, 2, 4 e 5 são então abertas. Portanto  $\mathbf{W}_{1,1} = \mathbf{W}_{2,1} = \mathbf{W}_{4,1} = \mathbf{W}_{5,1} = 1$  e, além disso, permanecerão como pilhas já abertas (mesmo depois que são fechadas) até o último período ( $\mathbf{W}_{1,j} = \mathbf{W}_{2,j} = \mathbf{W}_{4,j} = \mathbf{W}_{5,j} = 1, j = 1, \dots, M$ , pelas restrições (3.3)).

Uma solução ótima do modelo é a sequência de itens  $\{2, 4, 5, 1, 3\}$  da matriz  $\mathbf{X}$  e a partir desta, é obtida a sequência de padrões  $\{1, 2, 3, 4\}$  que gera no máximo 4 pilhas abertas em cada estágio. O cálculo da sequência de padrões e do máximo número de pilhas abertas é feita da seguinte maneira.

Seja a sequência ótima de itens obtida no modelo matemático:  $\{2, 4, 5, 1, 3\}$ . Para completar a pilha do item 2 precisam ser processados os padrões 1 e 2. Em seguida, para completar a pilha do item 4 é necessário processar os padrões 1, 2 e 3. Para completar a pilha do item 5 precisam ser processados os padrões 1 e 3. Para a pilha do item 1, os padrões 2, 3 e 4 e para o item 3 os padrões 3 e 4. É obtida assim, uma sequência auxiliar com padrões repetidos  $\{1, 2, 1, 2, 3, 1, 3, 2, 3, 4, 3, 4\}$ . Percorre-se esta sequência da esquerda para a direita e, logo que o padrão aparece na sequência, ele é eliminado das posições seguintes. Assim, é obtida a sequência de padrões  $\{1, 2, 3, 4\}$ . O número máximo de pilhas abertas da sequência de padrões é calculado conforme mostrado na Seção 2.1.

## 3.2 Variações do modelo MOSP de Yanasse e Pinto (2003) e novo modelo

Nesta Seção são analisadas algumas variações para o modelo MOSP discutido na Seção 3.1. Inicialmente, são discutidas cinco variações para o modelo, sendo a quinta delas também chamada MOSP Novo. Depois, são apresentados os testes computacionais realizados.

### Variação 1: Dominância entre itens

A primeira variação do modelo (3.1)-(3.8) consiste em acrescentar uma restrição na qual, se existem dois itens distintos  $i, p$  tal que  $S_{i,t} = S_{p,t}$  para todo  $t = 1, \dots, M$  então o item  $i$  é sequenciado imediatamente antes ou depois do item  $p$ . Em outras palavras, para completar a pilha do item  $i$  ou do item  $p$  são abertas as mesmas pilhas, incluindo eles próprios. Escolhendo, por exemplo, que o item  $i$  é sequenciado imediatamente antes do item  $p$ , pode-se escrever a seguinte restrição:

$$\begin{aligned} \text{Se } \exists i, p (i \neq p, i, p \in \{1, \dots, M\}) \text{ tal que } S_{i,t} = S_{p,t}, \text{ para } t = 1, \dots, M \\ \text{então } X_{i,j} = X_{p,j+1}, j = 1, \dots, M - 1. \end{aligned} \quad (3.9)$$

Considere que  $S_{i,t} = S_{p,t} = S_{k,t}$  para  $t = 1, \dots, M$  e  $i \neq p \neq k$ . Neste caso, é impossível completar a pilha do item  $i$  imediatamente antes da pilha do item  $p$  e, também, completar a pilha do item  $i$  imediatamente antes da pilha do item  $k$ . Uma possível sequência correta seria completar a pilha do item  $i$  imediatamente antes de completar a pilha do item de  $p$  e completar a pilha do item  $p$  imediatamente antes de completar a pilha do item  $k$ . Pela matriz  $S$  do exemplo apresentado na Seção 3.1 é possível identificar por exemplo que  $S_{1,t} = S_{4,t} = S_{5,t}$ ,  $t = 1, \dots, M$ . Assim, os itens 1, 4 e 5 podem ser sequenciados um imediatamente depois do outro mantendo o mesmo número de pilhas abertas.

Para cada par  $(i,p)$  satisfazendo  $S_{i,j} = S_{p,j}$ ,  $j = 1, \dots, M$ , acrescentam-se  $(M - 1)$  restrições no modelo (3.1)-(3.8).

A restrição 3.9 foi inicialmente analisada em Pinto (2004) em que foram resovidos problemas com no máximo 10 padrões e 10 itens. Ao comparar o tempo computacional dos modelos original e original com a nova restrição, Pinto (2004) verificou que não houve diferença significativa nos tempos, exceto para duas das vinte classes.

O modelo com estas restrições (3.9) é referenciada no texto como MOSP-1.

### Variação 2: Dominância entre elementos da matriz W

Esta variação consiste em acrescentar a restrição (3.10) ao modelo MOSP-1, o modelo resultante será denotado por MOSP-2.

A restrição

$$\mathbf{W}_{i,j-1} \leq \mathbf{W}_{i,j}, \quad j = 2, \dots, M \text{ e } i = 1, \dots, M \quad (3.10)$$

indica que a partir do momento que uma pilha  $i$  é aberta ( $\mathbf{W}_{i,j} = 1$ ) esta pilha permanece como “já aberta” até que a última pilha seja completada. Esta restrição já é implicitamente imposta por (3.3) nos modelos MOSP e MOSP-1, mas com a inclusão de (3.10) espera-se melhorar a relaxação linear do modelo e conseqüentemente, a convergência do *branch-and-cut* utilizado. São acrescentadas  $M(M - 1)$  restrições ao modelo MOSP-1.

A restrição (3.10) também foi inicialmente analisada em Pinto (2004). Conforme seus testes computacionais em problemas com no máximo 10 padrões e 10 itens, o acréscimo das restrições (3.10) no modelo MOSP-1 não alterou significativamente o tempo computacional. Porém, em classes com 10 itens e 20 padrões, o tempo computacional reduziu na maioria das classes.

### Varição 3: Valor do parâmetro $K$

Seja o problema MOSP dado pela matriz  $\mathbf{A}$  apresentado na Seção 3.1. Ao aplicar o modelo MOSP, é obtida a matriz das restrições (3.2)-(3.5) conforme mostrado na Figura 3.1. As linhas 1-5 representam as restrições (3.2), as linhas 6-30 as restrições (3.3), as linhas 31-35 as restrições (3.4) e as linhas 36-40 as restrições (3.5). No topo da figura, estão indicadas as variáveis  $\mathbf{W}_{i,j}$ ,  $\mathbf{X}_{i,j}$  e  $\mathbf{C}$ , lembrando que  $j$  indica o período em que a pilha do item  $i$  é completada. Considera-se o parâmetro  $K$  igual ao número de tipos de itens,  $K = 5$ .

As restrições das linhas 6-10 da figura são referentes as pilhas que são formadas no primeiro estágio do processamento. Seja a sequência ótima de itens  $\{2, 4, 5, 1, 3\}$  calculada anteriormente. A pilha referente ao item 2 é completada no estágio 1 e assim,  $\mathbf{S}_{2,j}$  para  $j = 1, \dots, M$  indica as pilhas formadas neste estágio, e os demais  $\mathbf{S}_{i,j}, i \neq 2, j = 1, \dots, M$  não são consideradas pois são multiplicadas pelas variáveis  $\mathbf{X}_{i,1} = 0, i = 1, \dots, M$ . Desta maneira, é possível considerar  $K = 1$  nas restrições 6-10.

Nas restrições das linhas 11-15 são indicadas as pilhas necessárias para completar as pilhas do primeiro e do segundo estágios. No primeiro estágio foi completada a pilha do item 2. No segundo estágio completa-se a pilha do item 4 e as pilhas que são formadas neste estágio são indicadas por  $\mathbf{S}_{4,j}, j = 1, \dots, M$ . Os demais elementos  $\mathbf{S}_{i,j}, i \neq 1, j = 1, \dots, M$  não são utilizados. As restrições 11-15 são, portanto, válidas para  $K = 2$ .

Esta análise é feita para a sequência  $\{2, 4, 5, 1, 3\}$  e são obtidas as restrições conforme a Figura 3.2. Isto mostra que é válido considerar  $K = t$ , em que  $t$  é o período de processamento e varia de  $t = 1, \dots, M$ .

W	11	12	13	14	15	21	22	23	24	25	31	32	33	34	35	41	42	43	44	45	51	52	53	54	55	C	
1	1																										-1
2		1																									-1
3			1																								-1
4				1																							-1
5					1																						-1
6	-5																										0
7																											0
8																											0
9																											0
10																											0
11																											0
12																											0
13																											0
14																											0
15																											0
16																											0
17																											0
18																											0
19																											0
20																											0
21																											0
22																											0
23																											0
24																											0
25																											0
26																											0
27																											0
28																											0
29																											0
30																											0
31																											0
32																											0
33																											0
34																											0
35																											0
36																											0
37																											0
38																											0
39																											0
40																											0

Figura 3.1: Restrições (3.2)-(3.5) do modelo MOSP para o exemplo dado pela matriz A da Seção 3.1.



	W	11	12	13	14	15	21	22	23	24	25	31	32	33	34	35	41	42	43	44	45	51	52	53	54	55	C		
6	-1						s21																				<=	0	
7							s22																					<=	0
8							s23																					<=	0
9							s24																					<=	0
10							s25																					<=	0
11	-2						s21																					<=	0
12							s22																					<=	0
13							s23																					<=	0
14							s24																					<=	0
15							s25																					<=	0
16	-3						s21																					<=	0
17							s22																					<=	0
18							s23																					<=	0
19							s24																					<=	0
20							s25																					<=	0
21	-4						s11																					<=	0
22							s12																					<=	0
23							s13																					<=	0
24							s14																					<=	0
25							s15																					<=	0
26	-5						s11																					<=	0
27							s12																					<=	0
28							s13																					<=	0
29							s14																					<=	0
30							s15																					<=	0

Figura 3.2: Restrições (3.3) consideradas no modelo MOSP-3 para o exemplo dado pela matriz A da Seção 3.1.

Este modelo MOSP-3 difere do modelo MOSP-2 apenas por considerar  $K = t$  ao invés de  $K = M$ .

#### Varição 4: Redução do número de restrições

A quarta variação analisada consiste em modificar a restrição (3.2) do modelo MOSP-3 para

$$\sum_{i=1}^M \mathbf{W}_{ij} \leq C + j - 1, \quad j = 1, \dots, M - LI \quad (3.11)$$

em que, LI é um limitante inferior do problema MOSP.

Esta modificação é motivada por um fato que ocorre quando o problema MOSP é resolvido com um método de enumeração dos itens. Chega-se em um estágio na árvore de enumeração em que, qualquer caminho subsequente não modifica mais o valor já obtido de número máximo de pilhas abertas, ou seja, a solução do problema MOSP é encontrada antes de obter uma sequência completa dos itens. Desta forma, é possível desconsiderar no modelo MOSP algumas restrições que não influenciam na solução ótima do problema.

Neste modelo MOSP-4 estão incluídas todas as variações propostas.

#### Varição 5: Nova formulação para o MOSP

Na sequência é apresentada uma nova formulação para o problema MOSP, baseada nas quatro variações apresentadas no decorrer desta Seção. Este modelo é referenciado no texto como MOSP NOVO.

Os dados e variáveis utilizados na formulação já foram definidos na Seção 3.1.

Formulação MOSP NOVO:

$$\text{Minimizar } \mathbf{C} \quad (3.12)$$

Sujeito a:

$$\sum_{i=1}^M \mathbf{W}_{ij} \leq \mathbf{C} + j - 1 \quad j = 1, \dots, M - LI \quad (3.13)$$

$$\sum_{j=1}^M \mathbf{X}_{jt} \mathbf{S}_{jp} \leq \mathbf{W}_{pt} \quad t = 1, \dots, M - LI \text{ e } p = 1, \dots, M \quad (3.14)$$

$$\mathbf{W}_{i,j-1} \leq \mathbf{W}_{i,j} \quad j = 2, \dots, M - LI \text{ e } i = 1, \dots, M \quad (3.15)$$

$$\sum_{j=1}^M \mathbf{X}_{ij} = 1 \quad i = 1, \dots, M \quad (3.16)$$

$$\sum_{i=1}^M \mathbf{X}_{ij} = 1 \quad j = 1, \dots, M \quad (3.17)$$

$$\mathbf{X}_{ij} \in \{0, 1\} \quad i = 1, \dots, M \text{ e } j = 1, \dots, M \quad (3.18)$$

$$\mathbf{W}_{ij} \in \{0, 1\} \quad i = 1, \dots, M \text{ e } j = 1, \dots, M \quad (3.19)$$

$$1 \leq \mathbf{C} \leq M \quad \mathbf{C} \in \mathbb{R}. \quad (3.20)$$

O objetivo do modelo é minimizar o número máximo de pilhas abertas  $C$ . Neste modelo são consideradas as restrições (3.14) e (3.15) ao invés de (3.3) do modelo MOSP de Yanasse e Pinto (2003). O limitante inferior LI foi utilizado para limitar o número das restrições (3.13)-(3.15) e corresponde ao número máximo de itens num padrão.

### 3.2.1 Experimentos computacionais

Nesta seção são apresentados alguns experimentos computacionais que tem como objetivo analisar o desempenho do modelo quando aplicadas as variações propostas.

A implementação foi desenvolvida em linguagem C e compilada usando o mingw32 3.4.2, uma versão do compilador gcc para sistemas operacionais Microsoft Windows 32 bits, disponível na interface Dev-C++. Os problemas foram resolvidos usando a biblioteca *Callable Library* disponível no CPLEX 10.0. Os experimentos foram realizados em um computador com processador Pentium 4, 3.0 GHz, 2 GB de RAM e sistema operacional Microsoft Windows XP.

#### Instâncias de teste

As instâncias de teste foram apresentadas em Smith e Gent (2005) e são mostradas na Tabela 3.1. O número de instâncias em cada classe é mostrado na coluna “Problemas”.  $M$  é o número de tipos de itens,  $N$  é a quantidade de padrões e  $Q$  é o número máximo de itens em cada padrão. Na quinta coluna da Tabela são mostrados os valores considerados para  $Q$ , de forma que, existem 4 tipos de instâncias “wbo\_10\_10” e cada uma delas com um valor de  $Q$  compreendido no intervalo [2;5]: 2, 3, 4 ou 5. Tem-se, portanto, 40 instâncias “wbo\_10\_10” sendo 10 com  $Q = 2$ , 10 instâncias com  $Q = 3$  e assim por diante. Todas as instâncias foram resolvidas aplicando os modelos MOSP, MOSP-1, MOSP-2, MOSP-3, MOSP-4 e MOSP-NOVO. Estes modelos são resolvidos aplicando o CPLEX com tempo máximo de solução para cada instância limitado em 30 minutos.

**Tabela 3.1:** Classes utilizadas nos experimentos.

Classes	Problemas	$M$	$N$	Valores de $Q$
wbo_10_10	40	10	10	[2;5]
wbo_10_20	40	10	20	[2;5]
wbo_10_30	40	10	30	[2;5]
wbo_15_15	60	15	15	[2;7]
wbo_15_30	60	15	30	[2;7]
wbo_20_10	70	20	10	[4;10]
wbo_20_20	90	20	20	[2;10]
wbo_30_10	100	30	10	[6;15]
wbo_30_15	120	30	15	[4;15]
wbo_30_30	140	30	30	[2;15]

## Perfis de desempenho

A análise dos métodos de solução é feita utilizando-se perfis de desempenho, uma técnica proposta por Dolan e Moré (2002) e brevemente descrita na sequência. Seja  $\mathcal{P}$  o conjunto de  $n_p$  instâncias de teste e  $\mathcal{S}$  o conjunto de  $n_s$  *softwares* correspondentes aos métodos implementados. O valor da solução obtida para a instância  $p \in \mathcal{P}$  utilizando o software  $s \in \mathcal{S}$  é denotado por  $t_{p,s}$ . Para um *software*  $s \in \mathcal{S}$ , a comparação de seu desempenho na solução de uma instância  $p \in \mathcal{P}$ , em relação ao desempenho do melhor *software*, é feita utilizando-se a *razão de desempenho*:

$$r_{p,s} = \frac{t_{p,s}}{\min\{t_{p,s} : \forall s \in \mathcal{S}\}}.$$

Para obter uma avaliação do desempenho do *software* em relação aos demais, é definido o *perfil de desempenho* do *software*  $s$ , dado por:

$$\rho_s(\tau) = \frac{1}{n_p} |\{p \in \mathcal{P} : r_{p,s} \leq \tau\}|,$$

com  $|\cdot|$  representando o número de elementos no conjunto. O perfil de desempenho  $\rho_s(\tau)$  é uma função que associa a um dado valor  $\tau \in \mathbb{R}$ , a fração de instâncias resolvidas pelo *software* com um desempenho dentro de um fator  $\tau$  do melhor desempenho obtido. Também pode ser visto como a probabilidade de que a razão de desempenho  $r_{p,s}$  associada ao *software*  $s$  não seja maior que  $\tau$ :

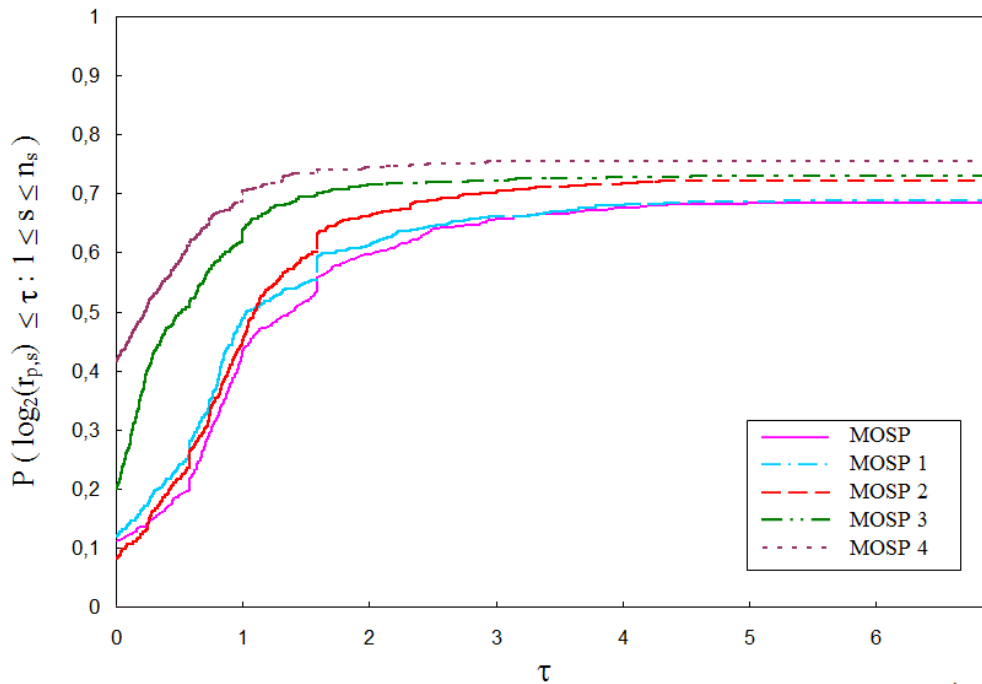
$$\rho_s(\tau) = P(r_{p,s} \leq \tau : 1 \leq s \leq n_s).$$

Os perfis de desempenho calculados devem ser apresentados em um único gráfico, cuja abscissa represente a variação do parâmetro  $\tau$ . Neste experimento, o cálculo dos perfis de desempenho e a criação dos gráficos são feitos com o auxílio da planilha `perfis.xls`, versão 1.2, descrita por Munari (2009) e disponível em <http://www.otm.icmc.usp.br/aplicativos>.

## Resultados e discussão

Para o cálculo dos perfis de desempenho e a criação dos gráficos, foi utilizada a planilha `perfis.xls`, versão 1.2, descrita por Munari (2009). Nos gráficos é comparado o tempo computacional, em segundos, utilizado no CPLEX para resolver as instâncias do experimento com cada um dos 5 modelos definidos anteriormente.

Para facilitar a visualização das curvas no gráfico de perfis de desempenho são considerados dois grupos de modelos e, para cada grupo, é obtido o gráfico de comparação. Inicialmente são comparados os modelos MOSP, MOSP-1, MOSP-2, MOSP-3 e MOSP-4 e é obtido o gráfico da Figura 3.3. Em seguida são comparados os modelos MOSP, MOSP-4 e MOSP NOVO conforme mostrado na Figura 3.4.



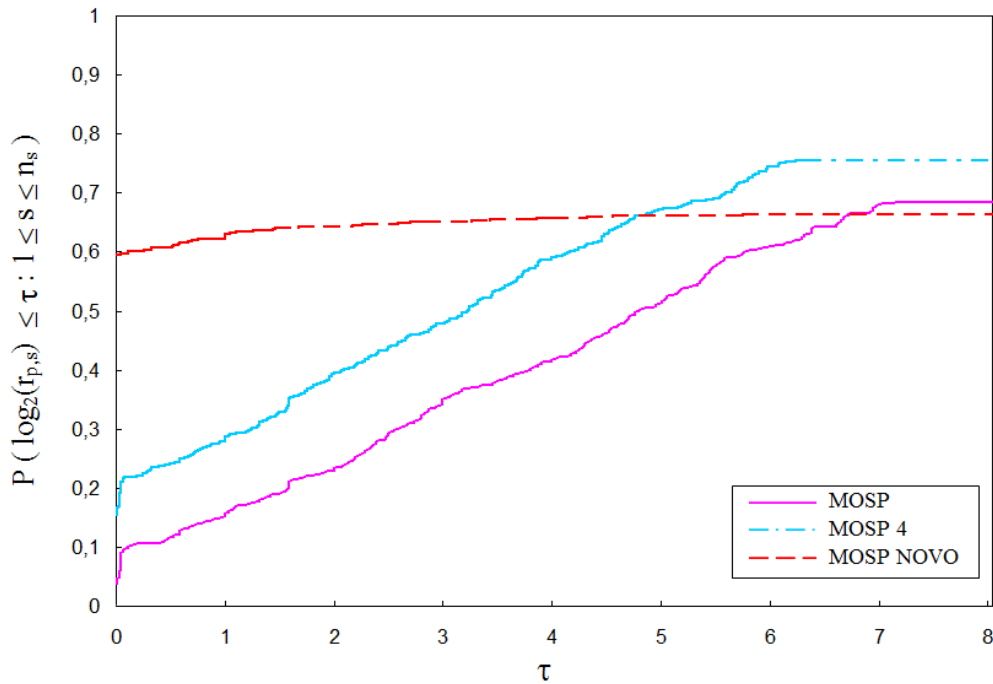
**Figura 3.3:** Gráfico de perfis de desempenho comparando os modelos MOSP, MOSP-1, MOSP-2, MOSP-3 e MOSP-4 em relação ao tempo computacional.

Na Figura 3.3, os valores à esquerda do gráfico mostram que o modelo MOSP-4 alcança a solução ótima em mais de 40% das instâncias utilizando o menor tempo computacional se comparado com os demais modelos, seguido do modelo MOSP-3 com aproximadamente 20% de instâncias com solução ótima.

Nenhum dos modelos alcança a margem superior do gráfico, isso indica que os modelos não encontram uma solução ótima para todas as instâncias. Analisando a margem direita, é verificado que, em relação à quantidade de soluções ótimas obtidas por cada um dos modelos, o modelo MOSP-4 alcança a otimalidade em quase 80% das instâncias seguido, respectivamente, pelos modelos MOSP-3, MOSP-2, MOSP-1 e MOSP.

Esta abordagem mostra que o perfil do modelo MOSP-4 domina os demais modelos comparados e, portanto, o desempenho geral deste modelo é melhor.

Para analisar os resultados obtidos com o modelo MOSP NOVO, é utilizado novamente o gráfico de perfis de desempenho definido no início desta Seção. Neste gráfico são comparados os modelos MOSP, MOSP NOVO e MOSP-4, este último é incluído na comparação devido ao melhor desempenho apresentado na análise feita anteriormente. Na Figura 3.4, os valores da margem esquerda do gráfico mostram que o MOSP NOVO encontra a solução ótima em 60% das instâncias testadas utilizando o menor tempo computacional. Este valor é significativamente grande se comparado com o modelo MOSP-4 com menos de 20% das instâncias e o MOSP com menos de 10%. Porém, em relação ao número de instâncias para as quais foi encontrada uma solução ótima (ver margem direita do gráfico), o modelo MOSP NOVO teve desempenho inferior aos modelos MOSP-4 e MOSP.



**Figura 3.4:** Gráfico de perfis de desempenho comparando os modelos MOSP, MOSP-4 e MOSP NOVO em relação ao tempo computacional.

Este perfil de desempenho do modelo MOSP NOVO pode ser explicado com base na comparação mais específica dos resultados obtidos. Em relação aos outros modelos, o MOSP NOVO apresenta maior dificuldade em resolver as instâncias “mais difíceis”, isto é, aqueles com matriz de padrões mais esparsa. Porém nas demais instâncias, que representam a maioria das instâncias utilizadas, o tempo de solução é melhor em relação aos demais modelos. Em outras palavras, com relação a quantidade de instâncias resolvidas na otimalidade e ao tempo computacional, o MOSP NOVO teve desempenho muito bom para as instâncias menores e pior desempenho para as instâncias maiores.

No sentido de complementar a análise do tempo computacional e da quantidade de instâncias resolvidas feita através dos gráficos das Figuras 3.3 e 3.4, são apresentadas as Tabelas 3.2 e 3.3. Na Tabela 3.2 é mostrada a porcentagem de instâncias em que foi obtida uma solução ótima para cada um dos modelos analisados, respeitando o tempo limite de 30 minutos para a solução de cada instância. Nas classes wbo\_10\_10, wbo\_10\_20 e wbo\_10\_30 foi obtida uma solução ótima para todas as instâncias, nos cinco modelos testados. A porcentagem diminui conforme aumenta o número de itens ( $M$ ) e padrões ( $N$ ) das instâncias. A dificuldade na solução destas instâncias também é verificada na Tabela 3.3, que mostra o aumento de tempo computacional médio em relação ao tamanho da instância.

A classe wbo\_30\_30 é a única que apresentou instâncias (com  $Q = 7$ ) para as quais não foi obtida solução factível no tempo limite de 30 minutos. O modelo MOSP não encontrou solução factível para uma destas instâncias, enquanto que, o modelo MOSP-1 não encontrou solução

**Tabela 3.2:** Porcentagem de instâncias em que é obtida uma solução ótima em cada classe wbo\_M\_N com os modelos MOSP, MOSP-1, MOSP-2, MOSP-3, MOSP-4 e MOSP NOVO.

Classes	MOSP	MOSP-1	MOSP-2	MOSP-3	MOSP-4	MOSP NOVO
wbo_10_10	100	100	100	100	100	100
wbo_10_20	100	100	100	100	100	100
wbo_10_30	100	100	100	100	100	100
wbo_15_15	90	97	93	98	98	95
wbo_15_30	90	92	98	98	100	87
wbo_20_10	69	69	89	89	90	60
wbo_20_20	74	74	78	79	79	70
wbo_30_10	38	40	37	38	42	35
wbo_30_15	48	48	54	52	81	44
wbo_30_30	59	56	59	61	66	59

factível para duas destas instâncias. Os modelos MOSP-2, MOSP-3, MOSP-4 e MOSP NOVO encontraram ao menos uma solução factível para todas as instâncias.

**Tabela 3.3:** Média do tempo computacional, em segundos, para resolver cada instância das classes wbo\_M\_N no CPLEX com os modelos MOSP, MOSP-1, MOSP-2, MOSP-3, MOSP-4 e MOSP NOVO.

Classes	MOSP	MOSP-1	MOSP-2	MOSP-3	MOSP-4	MOSP NOVO
wbo_10_10	0,26	0,30	0,26	0,19	0,20	0,18
wbo_10_20	0,15	0,15	0,17	0,13	0,14	0,08
wbo_10_30	0,11	0,12	0,12	0,09	0,09	0,03
wbo_15_15	327,28	193,34	199,53	95,30	92,08	221,48
wbo_15_30	241,38	237,32	148,42	109,27	107,94	244,34
wbo_20_10	676,83	680,56	439,94	384,19	318,85	770,93
wbo_20_20	576,28	564,60	491,46	462,78	452,45	557,04
wbo_30_10	1368,15	1309,03	1353,20	1316,08	1223,62	1241,44
wbo_30_15	1121,83	1103,26	1074,92	1057,43	1007,22	1014,24
wbo_30_30	867,30	890,66	886,91	861,03	763,78	756,70

Ao analisar todas as classes de instâncias, é constatada a influência do valor de  $Q$  no tempo computacional necessário para resolver os modelos. As Tabelas 3.4 e 3.5 exemplificam este comportamento para a classe com  $M = 20$  e  $N = 10$ . A Tabela 3.4 mostra o aumento da quantidade de instâncias em que é obtida uma solução ótima conforme aumenta o valor de  $Q$  (lembrando que  $Q$  indica o número de itens contidos em cada padrão). A Tabela 3.5 mostra como diminui o tempo computacional para solução das instâncias conforme aumenta o número de itens por padrão. Este comportamento vem do fato de que, quando  $Q$  tem valor alto, a solução MOSP (número de pilhas) do problema tende ser maior e o espaço de busca consiste de sequências de itens com número de pilhas semelhante, assim, utilizando limitantes no procedimento de solução, é eliminado um grande número de sequências do espaço de busca e, portanto, acelera-se a solução do problema.

**Tabela 3.4:** Influência do valor de  $Q$  em relação a quantidade de instâncias da classe wbo\_20\_10 para as quais é obtida uma solução ótima.

Classe	Q	MOSP	MOSP-1	MOSP-2	MOSP-3	MOSP-4	MOSP NOVO
wbo_20_10	4	0	0	40	40	40	10
wbo_20_10	5	10	10	80	80	90	40
wbo_20_10	6	70	70	100	100	100	40
wbo_20_10	7	100	100	100	100	100	40
wbo_20_10	8	100	100	100	100	100	90
wbo_20_10	9	100	100	100	100	100	100
wbo_20_10	10	100	100	100	100	100	100

**Tabela 3.5:** Influência do valor de  $Q$  em relação ao tempo computacional médio, em segundos, para resolver cada instância da classe wbo\_20\_10.

Classe	Q	MOSP	MOSP-1	MOSP-2	MOSP-3	MOSP-4	MOSP NOVO
wbo_20_10	4	1800,06	1800,03	1626,99	1459,72	1422,755	1741,51
wbo_20_10	5	1778,65	1763,53	927,70	858,73	537,494	1116,40
wbo_20_10	6	904,46	908,15	399,51	284,69	202,221	1143,37
wbo_20_10	7	131,20	111,04	49,17	43,15	41,044	1173,21
wbo_20_10	8	108,76	170,80	66,30	35,03	19,731	219,79
wbo_20_10	9	11,74	8,24	7,71	6,29	6,578	1,82
wbo_20_10	10	2,92	2,10	2,19	1,73	2,13	0,43

### 3.2.2 Considerações finais

Conforme os resultados apresentados na Seção 3.2.1, nota-se que a quantidade de instâncias em que é obtida uma solução ótima diminui bastante conforme aumenta o tamanho da instância. Além disso, quanto mais esparsa a matriz  $A$  (itens  $X$  padrões) do problema MOSP, maior a dificuldade em resolver a instância através dos modelos analisados.

Os resultados obtidos pelo modelo MOSP original de Yanasse e Pinto (2003) foram sempre melhorados por algum dos modelos MOSP-1, MOSP-2, MOSP-3 e MOSP-4. O modelo MOSP-4, que consiste na união de todas as variações propostas, obteve os melhores resultados, tanto em relação a quantidade de instâncias com solução ótima como em relação ao tempo computacional utilizado. Exceto nas instâncias menores, em que o MOSP NOVO se destacou.

Em geral, foi constatado que estes modelos, quando resolvidos pelo CPLEX, fornecem um limitante inferior muito ruim. Com isso, o método encontra dificuldades em provar a otimalidade da solução, ainda que esta tenha sido já encontrada. Neste sentido, uma análise da relaxação lagrangiana do modelo poderá direcionar a pesquisa em busca de resultados melhores.

Este trabalho também mostra que modificações no modelo podem, de fato, trazer melhorias nos tempos de convergência. Esta é outra linha de pesquisa que pode ser investigada em trabalhos futuros.



### 3.3 Modelo integrado MOSP/MORP

Os problemas de sequenciamento de padrões são problemas de otimização combinatorial e têm sido bastante discutidos na literatura. Na área de planejamento de produção várias situações práticas com objetivos específicos tem sido estudadas e alguns destes problemas são conhecidos como MOSP, MORP, MDP e o MTSP (estes problemas foram definidos na Seção 1). Nesta seção são considerados os problemas MOSP e MORP.

Com o intuito de minimizar o número de pilhas abertas pode-se pensar em minimizar o tempo em que a pilha permanece aberta. O problema de minimização da duração de pedidos (MORP) que poderia ser usado para minimizar o tempo que as pilhas permanecem abertas já foi tratado na literatura. Em relação às similaridades, o MOSP e o MORP não são equivalentes, isto é, uma solução ótima do MORP não é necessariamente uma solução ótima do MOSP e vice-versa. A relação entre estes dois problemas foi previamente estudada por Yanasse (1997a). O autor apresenta uma conjectura de que há sempre uma solução (uma sequência de padrões) que é simultaneamente ótima para o MOSP e para o MORP. O autor também comenta que mesmo que a conjectura seja verdadeira, isso não ajudaria a resolver o MOSP, pois, assim como o MOSP, o MORP também é um problema NP-difícil.

Admite-se que o MORP foi introduzido na literatura por Madsen (1988). Neste trabalho, é aplicado um procedimento 3-opt para resolver um TSP gerado a partir do MORP. A solução obtida para o problema gerado é considerada uma solução para o MORP. No trabalho de Foerster e Washer (1998) é proposto um procedimento 3-opt e um método de *simulated annealing* para resolver o MORP e, diferentemente do trabalho anterior, não é gerado um TSP e os métodos propostos são aplicados diretamente no MORP. Ainda nesta linha de metaheurísticas para solução do problema, Fink e Voss (1999) propõe um método de *simulated annealing* e um método de busca tabu para resolver dois problemas de sequenciamento de padrões, o MORP e o MOSP. Alguns autores discutem relações entre o MOSP e MORP (por exemplo Linhares e Yanasse (2002), Yanasse (1997a), Yanasse e Pinto (2003), Pinto (2004), Yanasse e Lamosa (2007) e da Silva Lopes (2011)), mesmo não propondo um método de solução para ambos os problemas.

A maioria dos estudos trata o MOSP e o MORP de forma independente, mas os dois problemas aparecem juntos em algumas situações. Nos problemas de corte de estoque pode ser interessante minimizar o número de pilhas abertas (MOSP) devido ao espaço físico limitado em torno da máquina de corte ao mesmo tempo em que se minimiza o tempo que as pilhas permanecem abertas (MORP) com o intuito de remanejá-las para outra área da empresa ou para entrega. Outra situação em que o MOSP e MORP aparecem juntos ocorre nas indústrias de funcionamento *make-to-order* em que vários tipos de produtos precisam ser feitos para completar o pedido de um cliente. Neste tipo de indústria é importante calcular uma sequência de produção que minimize o número de pedidos simultaneamente abertos (MOSP), porém, cada pedido deve permanecer aberto o menor tempo possível para ser encaminhado ao cliente (MORP).

Em seguida é apresentado um exemplo para melhor entender as definições do MOSP e MORP apresentadas anteriormente. Seja o exemplo do problema MOSP definido na Tabela 2.1 com os padrões  $P_1, P_2, P_3, P_4, P_5$  e  $P_6$  e os tipos de itens  $a, b, c, d, e, f$  e  $g$ . A Tabela 2.2 mostra o processo de corte da sequência de padrões  $P_5, P_3, P_6, P_1, P_2, P_4$  que resulta num número máximo de pilhas abertas igual a 4. Neste processo de corte dos padrões, em relação ao tempo em que as pilhas permanecem abertas, observa-se que as pilhas dos itens  $a, b, c, d, e, f$  e  $g$  tem duração de 2, 5, 1, 3, 5, 2 e 1 estágios, respectivamente. Portanto, a duração máxima das pilhas para a sequência de padrões considerada é 5 e a soma da duração das pilhas é 19.

Nesta seção, é apresentado um modelo de programação inteira mista que pode ser usado para resolver tanto o MOSP como o MORP. A função objetivo inclui o objetivo dos dois problemas. O modelo pode ser utilizado para resolver o MOSP sem considerar o MORP e vice-versa.

Na sequência, o novo modelo integrado MOSP/MORP é apresentado e os resultados dos experimentos computacionais são discutidos. Em seguida, são feitas as considerações finais e apresentadas possíveis extensões desta frente de investigação.

### 3.3.1 Modelo integrado MOSP/MORP

O novo modelo integra dois problemas, o MOSP e o MORP. Com este modelo MOSP/MORP é possível resolver cada um dos problemas de maneira independente. Na sequência é apresentada a formulação matemática desenvolvida para o MOSP/MORP.

Índices:

- $i$ : padrões.
- $k$ : itens.
- $j$ : posição dos padrões na sequência.

Dados do problema:

- $a_{ik}$ : vale 1 se o tipo de item  $k$  está no padrão  $i$ , e vale 0 caso contrário.
- $N$ : número de padrões.
- $M$ : número de tipos de itens.

Variáveis do problema:

- $X_{ij}$ : vale 1 se o padrão  $i$  está na posição  $j$  da sequência, e vale 0 caso contrário.
- $Y_{jk}^L$ : vale 1 se a pilha do item  $k$  é aberta ao cortar o  $j$ -ésimo padrão da sequência, e vale 0 caso contrário.

- $\mathbf{Y}_{jk}^U$ : vale 1 se a pilha do item  $k$  é fechada quando cortado o  $j$ -ésimo padrão da sequência, e vale 0 caso contrário.
- $\mathbf{C}$ : número máximo de pilhas abertas durante o processo de corte.
- $\mathbf{S}$ : duração máxima das pilhas abertas durante o processo de corte.

Formulação do problema integrado MOSP/MORP:

$$\text{Minimizar } Q_1\mathbf{C} + Q_2\mathbf{S} \quad (3.21)$$

Sujeito a:

$$\sum_{j=1}^N \mathbf{X}_{ij} = 1 \quad i = 1, \dots, N \quad (3.22)$$

$$\sum_{i=1}^N \mathbf{X}_{ij} = 1 \quad j = 1, \dots, N \quad (3.23)$$

$$\text{Se } (a_{ik} = 1) \text{ então } \sum_{j'=1}^j \mathbf{Y}_{j'k}^L \geq \mathbf{X}_{ij} \quad i = 1, \dots, N, k = 1, \dots, M, j = 1, \dots, N \quad (3.24)$$

$$\text{Se } (a_{ik} = 1) \text{ então } \sum_{j''=j}^N \mathbf{Y}_{j''k}^U \geq \mathbf{X}_{ij} \quad i = 1, \dots, N, k = 1, \dots, M, j = 1, \dots, N \quad (3.25)$$

$$\sum_{j=1}^N \mathbf{Y}_{jk}^L = 1 \quad k = 1, \dots, M \quad (3.26)$$

$$\sum_{j=1}^N \mathbf{Y}_{jk}^U = 1 \quad k = 1, \dots, M \quad (3.27)$$

$$\sum_{k=1}^M \left( \sum_{j'=1}^j \mathbf{Y}_{j'k}^L + \sum_{j''=j}^N \mathbf{Y}_{j''k}^U - 1 \right) \leq \mathbf{C} \quad j = 1, \dots, N \quad (3.28)$$

$$\sum_{j=1}^N j (\mathbf{Y}_{jk}^U - \mathbf{Y}_{jk}^L) \leq \mathbf{S} - 1 \quad k = 1, \dots, M \quad (3.29)$$

$$1 \leq \mathbf{S} \leq N \quad \mathbf{S} \in \mathbb{R} \quad (3.30)$$

$$1 \leq \mathbf{C} \leq M \quad \mathbf{C} \in \mathbb{R} \quad (3.31)$$

$$\mathbf{Y}_{jk}^L, \mathbf{Y}_{jk}^U \in \{0, 1\} \quad j = 1, \dots, N, k = 1, \dots, M \quad (3.32)$$

$$\mathbf{X}_{ij} \in \{0, 1\} \quad i = 1, \dots, N, j = 1, \dots, N. \quad (3.33)$$

O objetivo do modelo é minimizar uma soma ponderada entre o número máximo de pilhas abertas e a duração máxima de abertura de uma pilha. As restrições (3.22) e (3.23) garantem que todos os padrões são sequenciados, um em cada posição da sequência. As restrições (3.24) garantem que as pilhas correspondentes aos itens do padrão cortado no estágio  $j$  são abertas neste estágio ou antes e, de maneira similar, as restrições (3.25) garantem que estas pilhas são fechadas no estágio  $j$  ou depois. As restrições (3.26) e (3.27) impõem que as pilhas são abertas e fechadas uma única vez. Nas restrições (3.28) é calculado o número de pilhas que estão simultaneamente abertas em cada estágio de corte e, assim, o número máximo de pilhas abertas é obtido. Nas restrições (3.29) é calculada a duração de cada pilha e, então, a duração máxima das pilhas abertas é obtida. Os limitantes das variáveis reais  $\mathbf{C}$  e  $\mathbf{S}$  são impostos nas restrições (3.30) e (3.31). As variáveis binárias  $\mathbf{Y}_{jk}^L$ ,  $\mathbf{Y}_{jk}^U$  e  $\mathbf{X}_{ij}$  são definidas nas restrições (3.32) e (3.33).

O número de variáveis no MOSP/MORP é  $(N^2 + 2.M.N + 2)$  e o número de restrições é  $(2.N + 2.N \cdot \sum_{j=1}^N \sum_{k=1}^M a_{ik} + 2.M + N + M)$ .

Alguns autores da literatura consideram como objetivo do MORP minimizar a soma de estágios em que cada pilha permanece aberta. O modelo MOSP/MORP anterior pode ter a definição para esta função objetivo modificada para

$$\text{Minimizar } Q_1 \mathbf{C} + Q_2 \sum_{k=1}^M \sum_{j=1}^N j (\mathbf{Y}_{jk}^U - \mathbf{Y}_{jk}^L) \quad (3.34)$$

e, conseqüentemente, as restrições (3.29) e (3.30) são descartadas. Assim, o modelo passa a ter  $(N^2 + 2.M.N + 1)$  variáveis e  $(2.N + 2.N. \sum_{j=1}^N \sum_{k=1}^M a_{ik} + 2.M + N)$  restrições. Seja a função objetivo 3.34. Para resolver o problema com prioridade no MOSP é preciso adotar  $Q_1 > M(N - 1)$ , sendo este o valor máximo de  $\sum_{k=1}^M \sum_{j=1}^N j. (\mathbf{Y}_{jk}^U - \mathbf{Y}_{jk}^L)$  e ocorre quando todas as  $M$  pilhas são abertas no primeiro estágio ( $j = 1$ ) e fechadas no último estágio ( $j = N$ ).

### Inequações válidas analisadas

Com o intuito de melhorar o desempenho do novo modelo integrado MOSP/MORP foram formuladas algumas inequações válidas:

- Para cada posição  $j$  da sequência de corte, o número de pilhas que são abertas no estágio  $j$  é menor ou igual ao limitante superior ( $UB$ ) do número de pilhas abertas.

$$\sum_{k=1}^M \mathbf{Y}_{jk}^L \leq UB, \quad j = 1, \dots, N. \quad (3.35)$$

- Para cada posição  $j$  da sequência de corte, a pilha de um item  $k$  só pode ser fechada em  $j$  se esta pilha foi aberta antes ou na posição  $j$ .

$$\sum_{j'=1}^j \mathbf{Y}_{j'k}^L \geq \sum_{j'=1}^j \mathbf{Y}_{j'k}^U, \quad j = 1, \dots, N, k = 1, \dots, M. \quad (3.36)$$

- Para cada posição  $j$  da sequência de corte, a pilha de um item  $k$  só pode ser aberta em  $j$  se esta pilha será fechada nesta mesma posição ou em alguma das posições seguintes.

$$\sum_{j''=j}^N \mathbf{Y}_{j''k}^U \geq \sum_{j''=j}^N \mathbf{Y}_{j''k}^L, \quad j = 1, \dots, N, k = 1, \dots, M. \quad (3.37)$$

### 3.3.2 Experimentos computacionais

Nesta seção são apresentados alguns experimentos computacionais que tem como objetivo analisar o desempenho do novo modelo integrado MOSP/MORP. O peso considerado para o objetivo MOSP é  $Q_1 = N + 1$  e o peso para o objetivo MORP  $Q_2 = 1$ , de maneira que, o objetivo

MORP de minimizar a duração máxima das pilhas fica em segundo lugar. As inequações válidas, apresentadas na Seção 3.3.1, são adicionadas ao modelo MOSP/MORP para verificar se há algum ganho em relação a qualidade das soluções obtidas.

As instâncias de teste foram resolvidas usando a biblioteca *Callable Library* disponível no CPLEX 12.0. Foi definido um tempo máximo para solução de cada instância igual a 30 minutos (1800 segundos). As implementações computacionais foram feitas em linguagem C e compiladas usando o mingw32 3.4.2, uma versão do compilador gcc para sistemas operacionais Microsoft Windows 32 bits, disponível na interface Dev-C++. Todos os testes apresentados na sequência foram executados em um computador com processador Pentium 4, 3.0 GHz, 2 GB de RAM e sistema operacional Microsoft Windows XP.

### Instâncias de teste

As instâncias de teste são as utilizadas em Smith e Gent (2005). Na Tabela 4.9, são descritas as classes de instâncias aqui consideradas. Nas classes com nome “wbo”, as instâncias tem número fixo de itens por padrão. Para as classes “wbp”, tem-se instâncias nos quais cada item está contido em um número fixo de padrões. A classe “wbop” é constituída por instâncias nas quais o número de itens por padrão é igual ao número de padrões que contém este tipo de item. No nome das classes também estão indicados o número de tipos de itens ( $M$ ) e o número de padrões ( $N$ ) das instâncias que as compõem. A coluna “Exemplares” mostra a quantidade de instâncias em cada classe. Na quinta coluna da Tabela são mostrados os valores considerados para  $Q$ , de forma que, existem 4 tipos de instâncias “wbo\_10\_10” e cada uma delas com um valor de  $Q$  compreendido no intervalo [2;5]: 2, 3, 4 ou 5. Tem-se, portanto, 40 instâncias wbo\_10\_10 sendo 10 com  $Q = 2$ , 10 instâncias com  $Q = 3$  e assim por diante.

**Tabela 3.6:** Classes de instâncias utilizadas nos testes computacionais.

Classes	Exemplares	$M$	$N$	Valores de $Q$
wbo_10_10	40	10	10	[2;5]
wbo_15_30	60	15	30	[2;7]
wbo_30_30	140	30	30	[2;15]
wbp_30_10	40	30	10	[2;5]
wbop_20_20	90	20	20	[2;10]

### Resultados e discussão

No CPLEX é possível definir uma ordem de prioridade das variáveis do problema para ser seguida na ramificação da árvore do método *branch-and-bound*. As variáveis com maior prioridade são ramificadas primeiro. Para verificar o efeito desta escolha, as instâncias são resolvidas várias vezes, e cada vez um dos grupos de variáveis é priorizado. A Tabela 3.7 mostra a média do tempo computacional (em segundos) para resolver as instâncias da classe wbo\_10\_10 pelo CPLEX

usando o modelo MOSP/MORP. Na coluna “Nenhum” não é definida uma ordem de prioridade, nas demais colunas, o grupo de variáveis indicado recebe prioridade 1 enquanto as outras variáveis recebem prioridade 0. Os resultados da Tabela mostram um ganho significativo no tempo quando considerada prioridade nas variáveis  $X$ . Devido aos bons resultados obtidos, nos demais testes também será utilizada prioridade nas variáveis  $X$ .

**Tabela 3.7:** Média do tempo computacional (em segundos) para resolver as instâncias da classe wbo\_10\_10 pelo CPLEX usando o modelo MOSP/MORP com prioridade nos diferentes grupos de variáveis.

Classes wbo_M_N_Q	Grupo de variáveis com prioridade					
	Nenhum	$Y^L$	$Y^U$	$X$	$C$	$S$
wbo_10_10_2	20,36	49,98	43,65	18,45	20,35	20,44
wbo_10_10_3	138,39	234,91	201,13	53,23	139,22	139,19
wbo_10_10_4	195,29	446,99	353,61	112,47	194,42	194,71
wbo_10_10_5	530,06	666,24	566,22	348,45	529,53	529,88
<b>Média_10_10</b>	<b>221,03</b>	<b>349,53</b>	<b>291,15</b>	<b>133,15</b>	<b>220,88</b>	<b>221,06</b>

Nos próximos testes são analisados resultados do modelo MOSP/MORP e das inequações válidas (3.35)-(3.37) descritas na Seção 3.3.1. No modelo integrado é incluída a inequação (3.35) e o modelo resultante será indicado como “+(3.35)”. As inequações (3.36) e (3.37) serão incluídas juntas no modelo MOSP/MORP e este modelo será referido como “+(3.36)(3.37)”.

Inicialmente, é analisado o valor do limitante inferior do nó raiz do método *branch-and-bound* no CPLEX. O nó raiz é resolvido de três maneiras: (i) os métodos de pré-processamento e a adição de inequações de corte aplicados no CPLEX são desativados, (ii) os métodos de pré-processamento são habilitados e a adição dos cortes continua desativada e, por último, (iii) os métodos de pré-processamento e a aplicação de inequações de corte são permitidos no CPLEX. Ao comparar o valor da solução objetivo do modelo MOSP/MORP relaxado (teste (i)) e o valor do limitante inferior obtido nos outros dois testes verifica-se o quanto o limitante inicial é melhorado com as técnicas implementadas no CPLEX. Os três testes foram aplicados para o modelo MOSP/MORP proposto e para os modelos “+(3.35)” e “+(3.36)(3.37)”.

Na Tabela 3.8 são mostrados, para todas as classes de instâncias, o valor médio da relaxação linear do modelo MOSP/MORP (teste (i)) e a média do valor do limitante inferior calculado no nó raiz do método *branch-and-bound* quando as opções de pré-processamento e adição de cortes são habilitadas ou não no CPLEX (testes (ii) e (iii)). O valor médio obtido no modelo relaxado foi igual nos três modelos: modelo MOSP/MORP, modelo “+(3.35)” e “+(3.36)(3.37)”. Quando habilitado o pré-processamento no CPLEX, o valor médio do limitante inferior obtido nos três modelos também foi igual. Dos resultados apresentados nesta tabela, observa-se que ao utilizar o pré-processamento os resultados melhoram significativamente em relação ao modelo relaxado. Por outro lado, a adição dos cortes implementados no CPLEX apresentou melhores resultados apenas na classe wbp\_30\_10. Em relação a adição das inequações (3.35), (3.36) e (3.37) no modelo MOSP/MORP, esta estratégia não contribuiu em nada para a melhoria do limitante inicial do pro-

**Tabela 3.8:** Média do valor obtido na relaxação linear do modelo MOSP/MORP (teste (i)) e a média do valor do limitante inferior calculado no nó raiz do método *branch-and-bound* no CPLEX (testes (ii) e (iii)). A média é calculada para cada classe de instâncias.

Classes	(i) Modelo relaxado	(ii) Pré-processamento	(iii) Pré-processamento e cortes
	Modelo,+(3.35) e +(3.36)(3.37)	Modelo,+(3.35) e +(3.36)(3.37)	Modelo +(3.35) +(3.36)(3.37)
wbo_10_10	11,00	36,40	37,20 37,23 37,60
wbo_15_30	31,00	136,00	136,00 136,00 136,05
wbo_30_30	31,00	256,19	256,43 256,43 256,31
wbop_20_20	21,00	121,19	121,14 121,14 121,08
wbp_30_10	30,29	78,82	97,84 98,28 98,08

**Tabela 3.9:** Média do tempo computacional (em segundos) para resolver a relaxação linear do modelo MOSP/MORP (teste (i)) e o nó raiz no método *branch-and-bound* no CPLEX (testes (ii) e (iii)). A média é calculada para cada classe de instâncias.

Classes	(i) Modelo relaxado	(ii) Pré-processamento	(iii) Pré-processamento e cortes
	Modelo +(3.35) +(3.36)(3.37)	Modelo +(3.35) +(3.36)(3.37)	Modelo +(3.35) +(3.36)(3.37)
wbo_10_10	0,10 0,09 0,21	0,16 0,23 0,21	1,00 1,02 1,41
wbo_15_30	6,51 6,42 18,31	21,46 29,94 22,60	115,12 126,43 118,49
wbo_30_30	12,61 13,31 57,53	23,85 23,34 35,81	202,03 203,79 270,72
wbop_20_20	3,53 3,68 17,08	9,23 10,61 16,40	62,93 63,74 77,83
wbp_30_10	0,78 0,80 3,23	7,38 7,54 9,36	28,37 28,42 37,54

blema, pois, o valor da média é semelhante nos três modelos.

O tempo computacional gasto para resolver apenas o nó raiz no método *branch-and-bound* no CPLEX, para as classes de instâncias estudadas, é mostrado na Tabela 3.9. Verifica-se que ao habilitar o pré-processamento e a aplicação de inequações de corte no CPLEX, o tempo para resolver o nó raiz aumenta consideravelmente, ultrapassando os 200 segundos para instâncias com 30 padrões e 30 itens.

Uma grande dificuldade na solução de alguns modelos matemáticos é a baixa qualidade do limitante inferior calculado no CPLEX, o que inviabiliza o uso destas técnicas de solução devido ao alto tempo computacional necessário. Nos resultados apresentados, observou-se que ao utilizar as técnicas implementadas no CPLEX para resolver o modelo proposto consegue-se melhorar o limitante inferior inicial do problema. Porém, o valor do limitante ainda é muito baixo gerando um grande trabalho de busca no método *branch-and-bound*.

A Tabela 3.10 mostra a quantidade de instâncias para as quais foi obtida uma solução ótima com o CPLEX. As instâncias de cada classe foram agrupados de acordo com a densidade da matriz da instância (valor de  $Q$ ) e cada um destes grupos possui 10 instâncias. As classes wbo\_15\_30 e wbo\_30\_30 não tiveram nenhuma instância resolvida e, por isso, não foram incluídas na Tabela.

**Tabela 3.10:** Quantidade de problemas em que é obtida a solução ótima dos modelos considerando ou não prioridade nas variáveis  $X$ . As instâncias foram agrupadas conforme o valor de  $Q$  em cada classe.

Classes	Nenhuma prioridade			Prioridade nas variáveis $X$		
	Modelo	+(3.35)	+(3.36)(3.37)	Modelo	+(3.35)	+(3.36)(3.37)
wbo_10_10_2	10	10	10	10	10	10
wbo_10_10_3	10	10	10	10	10	10
wbo_10_10_4	10	10	10	10	10	10
wbo_10_10_5	9	9	9	9	9	9
<b>Soma wbo_10_10</b>	<b>39</b>	<b>39</b>	<b>39</b>	<b>39</b>	<b>39</b>	<b>39</b>
wbop_20_20_2	2	3	1	4	5	2
wbop_20_20_3	0	0	0	0	0	0
wbop_20_20_4	0	0	0	0	0	0
wbop_20_20_5	0	0	0	0	0	0
wbop_20_20_6	0	0	0	0	0	0
wbop_20_20_7	0	0	0	0	0	0
wbop_20_20_8	0	0	0	0	0	0
wbop_20_20_9	0	0	0	0	0	0
wbop_20_20_10	0	0	0	0	0	0
<b>Soma wbop_20_20</b>	<b>2</b>	<b>3</b>	<b>1</b>	<b>4</b>	<b>5</b>	<b>2</b>
wbp_30_10_2	2	1	1	8	9	7
wbp_30_10_3	0	0	0	7	4	4
wbp_30_10_4	0	0	0	3	3	0
wbp_30_10_5	0	0	0	0	0	0
<b>Soma wbp_30_10</b>	<b>2</b>	<b>1</b>	<b>1</b>	<b>18</b>	<b>16</b>	<b>11</b>



Em cada classe, o número de instâncias resolvidas é maior nos grupos que têm as instâncias mais esparsas. Na classe das instâncias menores, wbo\_10\_10, apenas 1 das 40 instâncias não foi resolvida. Por outro lado, a classe wbop\_20\_20 que tem 90 instâncias, teve no máximo 5 resolvidas. Quando considerada a prioridade nas variáveis  $X$ , o número de instâncias resolvidas pelo CPLEX aumentou de forma significativa na classe wbo\_30\_10, o número de instâncias resolvidas passou de 2 para 18.

As Tabelas anteriores mostram como é difícil encontrar uma solução ótima para algumas classes de instâncias. É interessante saber, qual a diferença entre o valor obtido ao completar os 30 minutos e o valor da solução ótima da instância. A Tabela 3.11 mostra a quantidade de instâncias, em cada classe, que foi obtida solução igual à solução ótima (Diferença = 0) ou que a diferença é de 1, 2, 3, 4, 5 ou mais pilhas. Os resultados na classe wbo\_15\_30 mostram que, mesmo não resolvendo nenhum problema dentro dos 30 minutos estabelecidos, a maioria das soluções obtidas difere em até duas pilhas da solução ótima. Na classe wbo\_30\_10, que teve 18 instâncias resolvidas no CPLEX (ver Tabela 3.10), todos os valores obtidos para o MOSP são iguais ao valor ótimo.

**Tabela 3.11:** Diferença entre o valor da solução obtido em cada modelo e a solução ótima da instância.

Classes	Diferença	Modelo	+(3.35)	+(3.36)(3.37)
wbo_10_10	0	40	40	40
wbo_15_30	0	19	19	24
	1	26	18	17
	2	11	21	14
	3	4	2	4
	4	0	0	1
wbo_30_30	1	1	0	0
	3	3	5	0
	4	9	7	9
	5	16	13	9
	>5	111	115	122
wbop_20_20	0	51	54	41
	1	33	25	24
	2	6	11	25
wbp_30_10	0	40	40	40

### 3.3.3 Considerações finais

O modelo integrado MOSP/MORP proposto neste trabalho foi analisado apenas em relação à solução obtida para o MOSP. Foram feitos testes com algumas ferramentas disponíveis no CPLEX 12.0 e também formuladas e implementadas algumas inequações de corte para o modelo. O objetivo foi acelerar o tempo de solução do modelo integrado.

O tempo de solução das instâncias no CPLEX mostrou-se bastante sensível à prioridade dada as variáveis de ramificação. De fato, o tempo para solução dos problemas menores diminuiu e mesmo nos problemas em que o tempo chega aos 30 minutos fixados, o valor da solução obtida é mais próxima da solução ótima desejada quando as variáveis adequadas são ramificadas primeiro. Mesmo com a correta escolha de prioridade, o modelo MOSP/MORP gera uma solução ótima apenas para problemas menores, com 10 padrões, e dos demais problemas poucos são resolvidos.

As inequações válidas que foram sugeridas com o intuito de diminuir o tempo para solução do modelo não apresentaram uma contribuição significativa, porém, é visível a necessidade de encontrar novas restrições para “fortalecer” o modelo. Algumas inequações já foram formuladas e precisam ainda ser analisadas:

- O número máximo de pilhas abertas é no mínimo o número máximo de itens por padrão.

$$\sum_{k=1}^M \mathbf{a}_{ik} \leq \mathbf{C} \quad i = 1, \dots, N.$$

- Se um padrão  $i$  é cortado antes do  $j$ -ésimo padrão da sequência, então as pilhas dos itens  $k$  contidos neste padrão já foram abertas, ou seja,  $\mathbf{Y}_{jk}^L = 0$ .

$$\text{Se } (a_{ik} = 1) \text{ então } \mathbf{Y}_{jk}^L \leq 1 - \sum_{j'=1}^{j-1} \mathbf{X}_{ij'} \quad i = 1, \dots, N, j = 1, \dots, N, k = 1, \dots, M.$$

- Se um padrão  $i$  é cortado depois do  $j$ -ésimo padrão da sequência, então as pilhas dos itens  $k$  contidos neste padrão não podem ser fechadas na posição  $j$ , ou seja,  $\mathbf{Y}_{jk}^U = 0$ .

$$\text{Se } (a_{ik} = 1) \text{ então } \mathbf{Y}_{jk}^U \leq 1 - \sum_{j''=j+1}^N \mathbf{X}_{ij''} \quad i = 1, \dots, N, j = 1, \dots, N, k = 1, \dots, M.$$

Assim como foram feitos experimentos para observar o comportamento do modelo para solução do MOSP, também precisa ser analisado seu comportamento quando é dada prioridade para a solução do MORP. Em seguida, modificar os parâmetros da função objetivo de forma representar situações em que o MOSP e o MORP devem ser integrados, já que, este é o grande atrativo do modelo integrado MOSP/MORP aqui proposto.

---

# Heurísticas para solução do MOSP

---

Nesta seção são apresentadas duas heurísticas para solução do problema MOSP. A primeira delas relaciona o grafo MOSP com o grafo que representa o problema do caixeiro viajante (TSP do inglês, *Traveling salesman problem*). Na segunda heurística, a ideia é decompor o grafo MOSP em subgrafos e, a partir da solução destes subgrafos obter a solução do grafo original.

## 4.1 Uma estratégia de solução do MOSP utilizando o TSP

Como mostrado na Seção 2.1, o MOSP pode ser representado como um grafo MOSP de itens. Fomos levados a pensar se uma solução para o MOSP não poderia ser obtida relacionando-o com o TSP a partir do grafo MOSP de itens. A vantagem potencial desta abordagem está no fato de que existem métodos heurísticos e exatos bastante eficientes disponíveis na literatura para a solução do TSP (Laporte, 1992; Applegate et al., 2006). De fato, a estratégia de se converter outros problemas combinatórios no TSP e fazer o uso das técnicas existentes já foram utilizadas em outros trabalhos na literatura (Madsen, 1988; Hertz et al., 1998; Ríos-Mercado e Bard, 1999; Agarwala et al., 2000).

Nesta seção é proposta uma estratégia de solução para o MOSP que consiste em definir um TSP a partir do problema MOSP. A associação do TSP com o MOSP é feita por meio da atribuição de distâncias às arestas do grafo TSP que será resolvido por métodos disponíveis na literatura. Como se trata de uma associação e não uma redução entre os problemas, a solução ótima obtida para o TSP não necessariamente é uma solução ótima para o MOSP original. Entretanto, os resultados obtidos sugerem que a atribuição de distâncias proposta leva à obtenção de boas soluções para o MOSP.

Uma abordagem semelhante à proposta neste trabalho foi utilizada por Madsen (1988), porém na resolução do MORP. Para a definição do TSP, a regra de atribuição de distâncias utilizada foi bastante simples, dada pelo número de itens em comum entre os padrões do corte. Alguns testes computacionais foram realizados pelo autor mas os resultados não foram comparados com outros métodos de solução da literatura.

Antes de detalhar a nova estratégia de solução e os experimentos computacionais realizados para a verificação do desempenho da estratégia proposta, é dada uma breve revisão sobre o TSP.

### 4.1.1 O problema do caixeiro viajante

O TSP é um dos problemas mais estudados de otimização combinatória (Laporte, 1992). O objetivo do TSP é encontrar uma rota de menor distância que passe exatamente uma vez por todas as cidades que devem ser visitadas e retorne para a cidade inicial. O TSP é utilizado na modelagem e na solução de outros problemas práticos, como o roteamento de veículos e o sequenciamento de tarefas. Apesar de sua descrição bastante simples, o TSP pertence à categoria de problemas NP-Completo (Garey e Johnson, 1979).

A definição de um TSP pode ser feita por meio de uma matriz  $D$  de distâncias (ou custos), cujo elemento  $d_{ij}$  representa a distância entre as cidades  $i$  e  $j$ , para  $i, j = 1, \dots, n$ , sendo  $d_{ii} = 0$ . Se  $d_{ij} = d_{ji}$ , para todo par  $i, j$ , o problema é denominado simétrico (TSP), caso contrário, tem-se um problema do caixeiro viajante assimétrico (ATSP).

O TSP pode ser formulado como um problema em um grafo, no qual os vértices representam as cidades que devem ser visitadas e a cada arco  $e_{ij}$  está associada a distância  $d_{ij}$  entre os vértices que a determina. O objetivo é encontrar um ciclo Hamiltoniano de distância mínima, isto é, um ciclo que passa em cada um dos vértices do grafo exatamente uma vez e retorna ao vértice inicial, minimizando  $\sum_{i=1}^n d_{ip(i)}$ , sendo  $p(i)$  o sucessor de  $i$  no ciclo obtido.

Laporte (1992) apresenta uma revisão sobre os principais métodos de solução para o TSP, além de descrever algumas aplicações. No que diz respeito ao ATSP, um trabalho de revisão recente é o de Oncan et al. (2009), no qual são classificadas e estabelecidas relações entre diversas propostas de formulações do ATSP.

Os métodos heurísticos clássicos para a solução do TSP podem ser classificados em dois grupos, as heurísticas construtivas, que geram ciclos a partir da inserção de vértices, e as heurísticas de melhoria que usam busca em vizinhança para melhorar uma solução já conhecida. Para mais detalhes sobre estas heurísticas, recomenda-se os trabalhos de Lin e Kernighan (1973), Rosenkrantz et al. (1977) e Hahsler e Hornik (2007).

Duas heurísticas construtivas clássicas para a solução do TSP são a de inserção arbitrária e a de inserção mais distante (Rosenkrantz et al., 1977). Estes procedimentos de inserção iniciam com um ciclo formado por um único vértice. De forma iterativa, uma cidade  $k$  ainda não percorrida é selecionada e inserida no ciclo entre dois vértices consecutivos  $i$  e  $j$  de modo a minimizar a

expressão  $d_{ik} + d_{kj} - d_{ij}$ . O procedimento termina quando todos os vértices forem inseridos no ciclo. A diferença entre as duas heurísticas de inserção está no critério de seleção do vértice que será inserido no ciclo. Na inserção mais distante, o vértice  $k$  escolhido é aquele com a maior distância para qualquer um dos vértices do ciclo corrente. Já na inserção arbitrária,  $k$  é selecionado aleatoriamente.

Applegate et al. (2006) descrevem os principais desenvolvimentos teóricos e práticos que são a base de métodos exatos utilizados para a solução do TSP. O *software* livre Concorde possui uma implementação computacional destes métodos exatos considerada bastante eficiente (Applegate et al., 2009).

### 4.1.2 Estratégia de solução MOSP/TSP proposta

No TSP definido a partir do grafo MOSP de itens  $G(V, E)$ , o conjunto de vértices  $V$  passa a representar as cidades que devem ser visitadas e o conjunto de arcos  $E$  se torna o conjunto de estradas ligando estas cidades. Após ser obtida uma solução para o TSP, é derivada uma solução para o MOSP. A estratégia de solução MOSP/TSP é dada pelas seguintes etapas.

---

#### Heurística MOSP/TSP

---

**Passo 1:** Defina o grafo MOSP de itens  $G(V, E)$ .

**Passo 2:** Para cada aresta  $e_{ij} \in E$ , atribua a distância  $d_{ij}$  de acordo com alguma regra.

**Passo 3:** Resolva o TSP descrito por  $G(V, E)$  obtendo um ciclo Hamiltoniano  $H = \{v_{\pi_1}, v_{\pi_2}, \dots, v_{\pi_n}, v_{\pi_1}\}$ .

**Passo 4:** Converta o ciclo Hamiltoniano  $H$  numa solução para o MOSP.

---

O sucesso do método é dependente da regra de atribuição das distâncias usada no Passo 2. Diversas regras podem ser usadas para essa atribuição conforme será discutido nesta seção. Em particular, uma regra proposta mostrou-se melhor nos testes realizados e foi considerada na estratégia.

A solução do TSP no Passo 3 pode ser feita por meio de um método exato ou heurístico. Uma solução ótima para o TSP não implica em uma solução ótima para o MOSP. Entretanto, se a conversão do MOSP para o TSP for eficaz, quanto melhor a solução obtida para o TSP, maior será a chance de se obter uma boa solução para o MOSP.

Em relação ao Passo 4, um procedimento que consiste em converter o ciclo Hamiltoniano em uma solução do MOSP será apresentada nessa seção.

Uma aresta  $e_{ij}$  é percorrida quando se sai do vértice  $i$  até o vértice  $j$ , ou vice-versa. Para que uma pilha do item  $j$  seja fechada, todos os padrões contendo este tipo de item precisam ser processados e, com isso, a pilha de outros tipos de item são abertas. Olhando para o grafo MOSP de itens, um vértice  $j$  é fechado quando todas as arestas incidentes a  $j$  são percorridas. Um vértice  $i$  é dito aberto logo que uma aresta incidente é percorrida.

### Atribuição de distâncias

Para que a solução do TSP seja uma boa solução para o MOSP é essencial que haja uma atribuição adequada de distâncias. Uma atribuição de distâncias de acordo com os graus dos vértices é intuitiva, uma vez que está relacionada com o número de pilhas que podem ser abertas caso a aresta seja percorrida. Seja  $e_{ij} \in E$  uma aresta determinada pelos vértices  $v_i, v_j \in V$ . A distância  $d_{ij}$  associada a esta aresta é dada pela expressão:

$$d_{ij} = \max\{\text{degree}(v_i), \text{degree}(v_j)\} - |\text{ADJ}(v_i) \cap \text{ADJ}(v_j)| - 1, \quad (4.1)$$

em que  $|\cdot|$  denota o número de elementos no conjunto,  $\text{degree}(v)$  é o grau do vértice  $v$  e  $\text{ADJ}(v)$  representa o conjunto de vértices adjacentes a  $v$ . A seguinte proposição mostra uma importante propriedade da regra de atribuição de distâncias definida acima.

**Proposição 1.** *Seja uma aresta  $a_{ij}$  do grafo MOSP  $G(V, E)$ . O critério de atribuição de distâncias dada pela expressão 4.1 é uma estimativa de pior caso para o número máximo de pilhas que são abertas quando esta aresta é percorrida.*

**Demonstração:** Supondo que a aresta  $e_{ij}$  é percorrida saindo de  $v_i$  para  $v_j$ . Quando  $v_j$  é visitado, a pilha correspondente  $j$  já está aberta (pois é adjacente ao vértice  $v_i$ ) e novas pilhas serão abertas somente se existem vértices adjacentes à  $v_j$  que não são adjacentes à  $v_i$ . Assim, o número máximo de novas pilhas quando percorrida a aresta  $e_{ij}$  de  $v_i$  para  $v_j$  é  $\text{degree}(v_j) - |\text{ADJ}(v_i) \cap \text{ADJ}(v_j)| - 1$ . Um resultado semelhante é obtido ao supor que a aresta  $e_{ij}$  é percorrida saindo de  $v_j$  para  $v_i$ . Como não se sabe o sentido em que a aresta será percorrida, é usado o máximo entre  $(\text{degree}(v_i) - |\text{ADJ}(v_i) \cap \text{ADJ}(v_j)| - 1)$  e  $(\text{degree}(v_j) - |\text{ADJ}(v_i) \cap \text{ADJ}(v_j)| - 1)$  para calcular um limitante superior para o número de novas pilhas abertas. ■

Em testes preliminares, outras regras de atribuição foram utilizadas, porém apresentaram desempenho inferior quanto à qualidade da solução obtida para o MOSP. Nestas regras, foram utilizados a média aritmética e o menor valor entre os graus dos vértices, além de uma regra que atribuía distância 1 a todas as arestas.

### Conversão da solução do TSP em uma solução para o MOSP

Toda solução factível de um TSP é indicada por um ciclo Hamiltoniano, cuja distância total correspondente independe do vértice inicial. Entretanto, na solução do grafo MOSP, o vértice inicial não deve ser visitado mais de uma vez e o número máximo de pilhas abertas pode ser diferente dependendo do vértice tomado como inicial.

Para obter uma solução para o MOSP (uma sequência de padrões) a partir da solução do TSP (ciclo Hamiltoniano) é necessário realizar duas conversões: obter uma sequência de itens a partir da solução do TSP e, em seguida, calcular uma sequência de padrões correspondente.

Dada uma solução para o TSP, uma ideia simples para decidir qual vértice deve ser escolhido como inicial na solução do MOSP, consiste de um procedimento de múltiplos inícios, no qual o número máximo de pilhas abertas é calculado iniciando-se de cada um dos  $n$  vértices. A sequência que corresponder ao menor número máximo de pilhas abertas é escolhida. Apesar do cálculo do valor associado a uma sequência ser realizado  $M$  vezes ( $M$  é o número de itens/cidades do problema), este procedimento é computacionalmente rápido.

Outra estratégia bastante utilizada na literatura em casos semelhantes, consiste em inserir um vértice artificial no TSP. A distância deste vértice para qualquer outro é definida como nula. Assim, dada uma solução factível do TSP, uma sequência é obtida adotando-se o vértice consecutivo ao artificial como sendo a origem. Com isso, o antecessor do vértice artificial passa a ser o último vértice. De acordo com os testes realizados, o procedimento de múltiplos inícios resultou em uma solução com melhor qualidade para o MOSP em relação à utilização de um nó artificial. Por esta razão, este procedimento é adotado na estratégia de solução aqui proposta.

Para a obtenção de uma sequência de padrões a partir da sequência de itens, é proposto um procedimento baseando em Becceneri et al. (2004) e Yanasse (1997b), porém considerando a adjacência entre os vértices do grafo. A ideia consiste em percorrer a sequência de itens e inserir um padrão na sequência de padrões quando, pela primeira vez, todos os tipos de itens do padrão tiverem sido percorridos. A sequência de tipos de itens é percorrida do início para o fim, e sempre que a pilha de um tipo de item é fechada, as pilhas dos tipos de itens correspondentes a vértices adjacentes são abertas. Se existe algum padrão em que todos os tipos de itens estão com suas pilhas abertas, este padrão é colocado na sequência de padrões antes de qualquer outro padrão necessário para fechar a pilha corrente.

Este procedimento é descrito no Algoritmo 1, que faz uso do Algoritmo 2 com diferentes valores para  $j$ . Nos Algoritmos 1 e 2,  $|P_j|$  corresponde ao número de tipos de item no padrão de corte  $P_j$ . O vetor  $\pi\_flag$  é utilizado para indicar se uma pilha ainda não foi aberta ( $= 0$ ), se está aberta ( $= 1$ ) ou já foi fechada ( $= -1$ ). Cada elemento de  $P\_flag$  indica a quantidade de tipos de item em cada padrão que ainda não tiverem suas pilhas abertas durante o procedimento. Quando  $P\_flag[j] = 0$  o padrão  $j$  é adicionado à sequência de padrões e atualiza-se  $P\_flag[j] = -1$ .

O Algoritmo 1 é um procedimento que percorre a sequência de tipos de itens  $S_\pi$  a partir do primeiro elemento. Na  $i$ -ésima iteração, todos os padrões que contém o tipo de item  $k = \pi_i$  são analisados, se a pilha correspondente ainda não foi fechada. Depois disso, são analisados todos os padrões que contém algum item  $l$  tal que  $v_l \in ADJ(v_k)$  no grafo MOSP. Note que, como consequência, a pilha de um outro tipo de item pode ser completada, mesmo que este outro item apareça depois de  $k$  em  $S_\pi$ .

---

**Algoritmo 1: Cálculo de uma sequência de padrões a partir de uma sequência de tipos de itens.**

---

1. Seja o conjunto de padrões  $\{P_0, \dots, P_{N-1}\}$  e a sequência de tipos de itens  $S_\pi = \{\pi_0, \dots, \pi_{M-1}\}$ .
2.  $S_P = \emptyset$ .

3. Defina  $\pi\_flag[i] = 0$  e  $P\_flag[j] = |P_j|$ , para  $i = 0, \dots, M - 1$  e  $j = 0, \dots, N - 1$ .
4. Para  $i = 0, \dots, M - 1$  faça:
  - 4.1  $k = \pi_i$ .
  - 4.2 Se  $(\pi\_flag[k] = 0)$  então execute o Algoritmo 2 com  $j = k$ .
  - 4.3 Para cada  $l$  tal que  $v_l \in \text{ADJ}(v_k)$  faça:
    - Se  $(\pi\_flag[l] = 0)$  então execute o Algoritmo 2 com  $j = l$ .
  - 4.4  $\pi\_flag[k] = -1$ .
5. Retorna uma sequência de padrões  $S_P$ .

**Algoritmo 2: Cálculo da ordem de processamento dos padrões para o fechamento de uma pilha.**

1. Seja o conjunto de padrões  $\{P_0, \dots, P_{N-1}\}$ , os vetores  $\pi\_flag$  e  $P\_flag$  e o tipo de item  $j$ .
2. Para cada padrão  $P_q$  contendo o tipo de item  $j$  faça:
  - 2.1 Se  $(P\_flag[q] > 0)$  então  $P\_flag[q] = P\_flag[q] - 1$ .
  - 2.2 Se  $(P\_flag[q] = 0)$  então:
    - $S_P = S_P + \{P_q\}$ .
    - $P\_flag[q] = -1$ .
3.  $\pi\_flag[j] = 1$ .
4. Retorna a sequência  $S_P$  e vetores  $\pi\_flag$  e  $P\_flag$  atualizados.

Em Ashikaga (2001); Ashikaga e Soma (2009) os autores sugerem uma maneira muito simples para converter uma sequência de itens em uma sequência de padrões. A sequência de itens é percorrida na ordem inversa e, os padrões que contém cada tipo de item são inseridos na sequência de padrões também na ordem inversa. Este procedimento também foi considerado em testes preliminares, porém a sequência de padrões obtida apresentava um número máximo de pilhas abertas maior, quando comparado ao resultado obtido utilizando-se o Algoritmo 1. Portanto, o procedimento aqui proposto foi adotado como padrão na estratégia de solução.

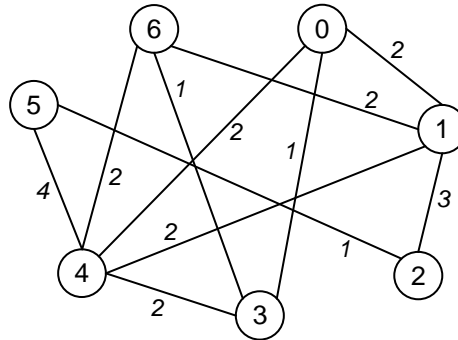
### Exemplo

O problema MOSP é dado pelos padrões de corte mostrados na Tabela 4.1. No exemplo, o padrão  $P_0$  é formado por itens do tipo 0, 3 e 4 e gera as arestas  $e_{03}$ ,  $e_{04}$  e  $e_{34}$  no grafo MOSP de itens  $G(V, E)$ . A cada aresta  $e_{ij} \in E$  é atribuída uma distância  $d_{ij}$  de acordo com a expressão (4.1). Por exemplo, a aresta  $e_{03}$  recebe a distância  $d_{03} = \max\{3, 3\} - |\{1, 3, 4\} \cap \{0, 4, 6\}| - 1 = 1$ . O grafo resultante é mostrado na Figura 4.1.



**Tabela 4.1:** Padrões de corte.

Padrão	Tipos de item
$P_0$	0, 3, 4
$P_1$	1, 4, 6
$P_2$	4, 5
$P_3$	3, 4, 6
$P_4$	0, 1
$P_5$	0, 1, 4
$P_6$	2, 5



**Figura 4.1:** Grafo TSP obtido a partir do problema MOSP apresentado na Tabela 4.1, usando a estratégia proposta.

Uma possível solução para o TSP representado na Figura é dada pelo ciclo Hamiltoniano  $H = \{0, 4, 5, 2, 1, 6, 3, 0\}$ , com distância total igual a 14 (neste caso, tem-se o valor ótimo). A partir deste ciclo, são analisadas as possíveis sequências de tipos de itens  $S_\pi^i$  determinadas em  $H$  e as respectivas sequências de padrões  $S_P^i$ , conforme apresentado na Tabela 4.2. A sequência de padrões com o menor valor  $Z_i$  deve ser selecionada como uma solução para o MOSP. Neste exemplo, uma sequência de padrões com valor  $Z_i = 4$  pode ser escolhida arbitrariamente.

**Tabela 4.2:** Sequências de itens analisadas a partir do ciclo Hamiltoniano obtido no TSP, sequência de padrões correspondentes e número máximo de pilhas abertas.

$i$	$S_\pi^i$	$S_P^i$	$Z_i$
1	{0, 4, 5, 2, 1, 6, 3}	{ $P_0, P_5, P_1, P_3, P_2, P_4, P_6$ }	4
2	{4, 5, 2, 1, 6, 3, 0}	{ $P_0, P_5, P_1, P_3, P_2, P_4, P_6$ }	4
3	{5, 2, 1, 6, 3, 0, 4}	{ $P_2, P_6, P_4, P_1, P_5, P_0, P_3$ }	4
4	{2, 1, 6, 3, 0, 4, 5}	{ $P_4, P_6, P_2, P_1, P_5, P_0, P_3$ }	4
5	{1, 6, 3, 0, 4, 5, 2}	{ $P_1, P_4, P_5, P_0, P_3, P_2, P_6$ }	5
6	{6, 3, 0, 4, 5, 2, 1}	{ $P_1, P_3, P_0, P_5, P_2, P_4, P_6$ }	4
7	{3, 0, 4, 5, 2, 1, 6}	{ $P_0, P_3, P_1, P_5, P_2, P_4, P_6$ }	4

### 4.1.3 Experimentos computacionais

Com o intuito de analisar o desempenho da estratégia de solução proposta, as heurísticas de inserção mais distante (HID) e inserção arbitrária (HIA), descritas na Seção 4.1.1, foram implementadas para a solução do TSP. Inicialmente, foi realizada a comparação da estratégia proposta com outros métodos da literatura. Para isso, foram implementados dois procedimentos específicos para a solução do MOSP, as heurísticas de contração recursiva de cliques (HCC)(Ashikaga, 2001) e de nó de custo mínimo (HNCM) (Becceneri et al., 2004), apresentadas no Capítulo 2. Em seguida, para verificar a qualidade das soluções obtidas pela estratégia proposta, o método exato de enumeração implícita descrito por Yanasse et al. (2007) foi utilizado para a obtenção da solução ótima das instâncias de teste.

A análise dos métodos de solução também é feita utilizando-se perfis de desempenho, uma técnica proposta por Dolan e Moré (2002) e brevemente descrita na Seção 3.2.1. Para o cálculo dos perfis de desempenho e a criação dos gráficos, foi utilizada a planilha perfis.xls, versão 1.2, descrita por Munari (2009).

As implementações computacionais foram feitas em linguagem C e compiladas usando o mingw32 3.4.2, uma versão do compilador gcc para sistemas operacionais Microsoft Windows 32 bits, disponível na interface Dev-C++. Todos os testes apresentados na sequência foram executados em um único computador, com processador Intel Core 2 Duo 2.39 MHz, 2 GB de RAM e sistema operacional Microsoft Windows Vista.

#### Instâncias de teste

As instâncias de teste foram retiradas de Smith e Gent (2005). Na Tabela 4.3, são descritas as classes de instâncias aqui consideradas. Uma instância que pertence à classe cujo nome é iniciado por “wbo” possui um número fixo de tipos de itens por padrão. Este número varia dentre as instâncias da classe como, por exemplo, de 2 a 5 na classe wbo\_10\_10. Para as classes que iniciam com “wbp”, tem-se instâncias nas quais cada tipo de item é contido em um número fixo de padrões. As iniciais “wbop”, indicam que a classe combina as características das classes “wbo” e “wbp”. As classes com nome iniciando por “gp” possuem instâncias variadas, consideradas de grande porte. No nome das classes também estão indicados o número de tipos de itens ( $M$ ) e o número de padrões ( $N$ ) das instâncias que as compõem.

#### Resultados e discussão

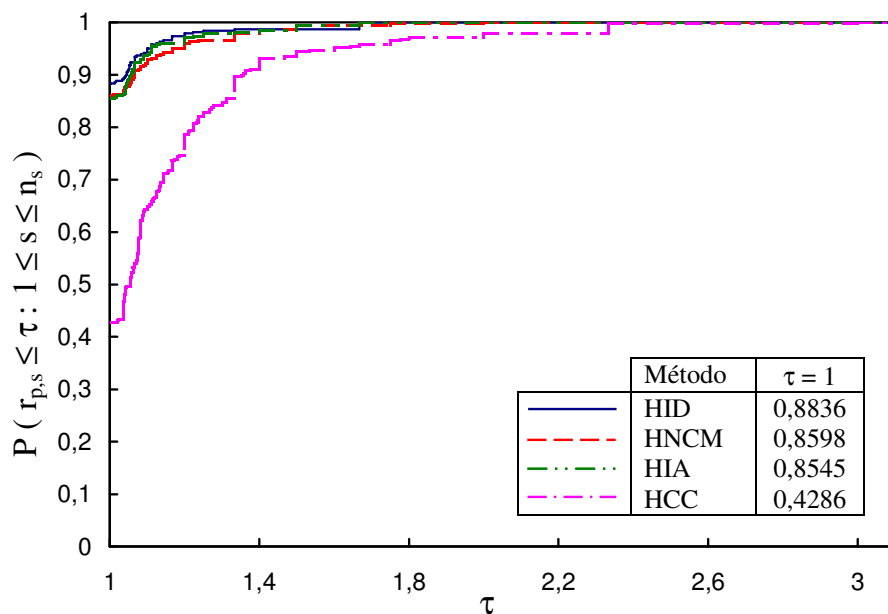
As informações sobre a solução das instâncias de teste utilizando os métodos implementados são dadas nas Tabelas 4.4 e 4.5. No total, são 378 instâncias de teste agrupadas de acordo com a densidade dos dados de entrada. Pode-se verificar que os valores das soluções obtidas pelos procedimentos HID e HIA foram bastante próximos aos obtidos pela heurística HNCM. O tempo destes três métodos também foi semelhante e, em geral, proporcional à densidade das instâncias. A

**Tabela 4.3:** Classes de instâncias utilizadas nos testes computacionais.

Classes	Problemas	$M$	$N$
wbo_10_10	40	10	10
wbo_15_30	60	15	30
wbo_30_30	140	30	30
wbop_20_20	90	20	20
wbp_30_10	40	30	10
gp_50_50	4	50	50
gp_100_100	4	100	100

solução utilizando a HNCM foi a que exigiu maior tempo em grande parte das instâncias. Por outro lado, a heurística HCC resolveu as instâncias em um tempo computacional muito pequeno e sem relação aparente com a densidade da instância, não ultrapassando 3 milissegundos nem mesmo para as instâncias de teste com maior dimensão. Entretanto, as soluções obtidas por esta heurística foram relativamente inferiores.

Na Figura 4.2, é apresentado o gráfico de perfis de desempenho contendo a comparação dos valores das soluções. O perfil da heurística HCC mostra que as soluções obtidas por este método são bastante inferiores em relação às obtidas pelos demais. Já os três outros procedimentos obtiveram desempenhos bastante próximos, sendo que o HID foi o que obteve o melhor desempenho geral, além de ter encontrado a melhor solução para 88,36% das instâncias. Neste contexto, conclui-se que a estratégia proposta obteve um desempenho geral semelhante ao da heurística HNCM, a qual é conhecida na literatura como uma das melhores heurísticas para a solução do MOSP, de acordo com as comparações realizadas por Yanasse e Lamosa (2007) e Becceneri et al. (2004).

**Figura 4.2:** Comparação entre a estratégia proposta e as heurísticas específicas para o MOSP, considerando os valores das soluções obtidas na solução das instâncias de teste.

**Tabela 4.4:** Informações sobre a solução das instâncias de teste pertencentes às classes especificadas. Os resultados foram agrupados de acordo com a densidade (D) das instâncias e são apresentados a média aritmética (M) e o desvio padrão (DP) de cada grupo, considerando o valor da solução e o tempo de solução (em milissegundos).

Classe	D	HID				HIA				HCC				HNCM			
		Valor		t (ms)		Valor		t (ms)		Valor		t (ms)		Valor		t (ms)	
		M	DP	M	DP	M	DP	M	DP	M	DP	M	DP	M	DP	M	DP
wbo_10_10	0,200	2,8	0,42	0,0	0,00	3,0	0,00	0,0	0,00	3,7	0,48	0,0	0,00	3,4	0,52	0,1	0,32
	0,300	5,2	0,63	0,3	0,48	5,2	0,63	0,0	0,00	6,4	0,84	0,0	0,00	5,3	0,82	0,0	0,00
	0,400	7,0	0,67	0,0	0,00	7,0	0,67	0,1	0,32	7,7	0,48	0,0	0,00	7,0	0,67	0,0	0,00
	0,500	8,7	0,67	0,0	0,00	8,7	0,67	0,1	0,32	8,8	0,63	0,0	0,00	8,7	0,67	0,2	0,42
wbo_15_30	0,133	6,2	0,63	0,0	0,00	5,8	0,42	0,2	0,42	7,2	1,03	0,0	0,00	6,1	0,74	0,1	0,32
	0,200	9,7	0,82	0,2	0,42	9,9	0,74	0,3	0,48	10,6	0,52	0,1	0,32	9,6	0,70	0,4	0,52
	0,267	11,8	0,42	0,2	0,42	11,8	0,42	0,2	0,42	12,7	0,48	0,0	0,00	11,8	0,42	0,2	0,42
	0,333	13,6	0,52	0,5	0,71	13,6	0,52	0,3	0,48	13,9	0,32	0,0	0,00	13,6	0,52	0,2	0,42
	0,400	14,4	0,52	0,2	0,42	14,4	0,52	0,2	0,42	14,4	0,52	0,0	0,00	14,4	0,52	0,2	0,42
	0,467	14,8	0,42	0,2	0,42	14,8	0,42	0,2	0,42	14,8	0,42	0,0	0,00	14,8	0,42	0,2	0,42
wbo_30_30	0,067	4,8	0,63	0,3	0,48	5,8	0,63	0,4	0,52	6,4	0,84	0,3	0,48	6,2	1,32	0,4	0,52
	0,100	10,8	0,63	0,9	0,88	10,7	0,82	0,4	0,52	14,7	1,57	0,2	0,42	10,8	0,92	0,5	0,53
	0,133	15,8	0,79	0,6	0,52	15,6	0,84	0,8	0,63	19,0	1,63	0,1	0,32	15,5	0,71	0,0	0,00
	0,167	19,1	0,99	1,0	0,47	19,2	1,03	0,8	0,63	21,8	1,14	0,1	0,32	18,8	0,79	1,0	0,00
	0,200	22,3	0,82	0,9	0,32	22,4	0,97	0,9	0,57	23,9	1,37	0,4	0,52	22,2	0,92	1,5	0,53
	0,233	24,1	0,74	1,1	0,32	24,1	0,32	1,3	0,48	25,9	0,74	0,0	0,00	23,9	0,57	1,7	0,48
	0,267	25,4	0,84	1,2	0,42	25,5	0,71	1,1	0,32	26,8	0,79	0,0	0,00	25,4	0,84	2,0	0,00
	0,300	26,8	0,63	1,5	0,71	26,9	0,57	1,5	0,71	27,7	0,48	0,0	0,00	27,1	0,74	2,0	0,00
	0,333	27,2	0,42	1,2	0,42	27,1	0,32	1,1	0,32	27,7	0,67	0,2	0,42	27,1	0,32	2,0	0,00
	0,367	28,3	0,67	1,2	0,42	28,3	0,67	1,5	0,71	28,5	0,71	0,0	0,00	28,3	0,67	2,1	0,32
	0,400	28,9	0,57	1,5	0,71	28,9	0,57	1,3	0,48	28,9	0,57	0,1	0,32	28,9	0,57	2,2	0,42
	0,433	29,3	0,48	1,3	0,48	29,3	0,48	1,5	0,53	29,3	0,48	0,0	0,00	29,3	0,48	2,3	0,48
	0,467	29,7	0,48	2,0	1,15	29,7	0,48	1,4	0,52	29,7	0,48	0,1	0,32	29,7	0,48	2,5	0,53
	0,500	29,9	0,32	1,5	0,53	29,9	0,32	1,7	0,67	29,9	0,32	0,2	0,42	29,9	0,32	2,7	0,48
wbop_20_20	0,100	4,1	0,99	0,2	0,42	3,5	0,85	0,1	0,32	6,9	0,99	0,1	0,32	3,0	0,00	0,0	0,00
	0,150	9,3	0,67	0,2	0,42	9,3	0,67	0,2	0,42	11,7	0,82	0,0	0,00	9,1	0,88	0,4	0,52
	0,200	12,7	0,95	0,4	0,52	12,9	0,74	0,2	0,42	14,9	0,88	0,0	0,00	12,8	0,92	0,3	0,48
	0,250	15,5	0,53	0,6	0,52	15,5	0,53	0,6	0,52	16,8	0,79	0,0	0,00	15,6	0,70	0,3	0,48
	0,300	17,3	0,48	0,2	0,42	17,4	0,52	0,5	0,53	17,9	0,57	0,1	0,32	17,4	0,52	0,4	0,52
	0,350	18,2	0,42	0,5	0,53	18,2	0,42	0,5	0,53	18,8	0,42	0,2	0,42	18,4	0,52	0,5	0,53
	0,400	19,0	0,00	0,5	0,53	19,0	0,00	0,3	0,48	19,0	0,00	0,1	0,32	19,0	0,00	0,6	0,52
	0,450	19,9	0,32	0,5	0,53	19,9	0,32	0,3	0,48	19,9	0,32	0,1	0,32	19,9	0,32	0,6	0,52
	0,500	20,0	0,00	0,5	0,53	20,0	0,00	0,5	0,53	20,0	0,00	0,0	0,00	20,0	0,00	0,7	0,48
	wbp_30_10	0,200	15,4	0,52	0,5	0,53	15,6	1,07	0,8	0,42	20,5	1,72	0,2	0,42	17,0	2,21	0,7
0,300		22,5	1,08	0,9	0,57	22,7	1,06	0,9	0,57	25,4	0,84	0,1	0,32	23,6	1,35	1,2	0,42
0,400		26,4	0,70	1,1	0,32	26,4	0,70	1,0	0,00	28,3	0,67	0,0	0,00	26,8	0,92	1,8	0,42
0,500		29,2	0,42	1,2	0,42	29,2	0,42	1,2	0,42	29,2	0,42	0,0	0,00	29,2	0,42	2,1	0,57

**Tabela 4.5:** Valor da solução e tempo de solução para cada instância das classes gp\_50\_50 e gp\_100\_100 usando os métodos de solução indicados.

Classe	D	HID		HIA		HCC		HNCM	
		Valor	t (ms)	Valor	t (ms)	Valor	t (ms)	Valor	t (ms)
gp_50_50	0,539	31	8	31	7	44	0	30	9
	0,632	40	12	40	12	43	0	40	12
	0,654	40	10	40	12	43	0	40	12
	0,808	45	15	45	15	46	1	45	14
gp_100_100	0,543	61	91	61	92	90	2	60	120
	0,672	76	105	75	105	82	2	75	159
	0,677	75	102	76	98	91	2	75	160
	0,857	95	156	95	147	97	3	96	192

Dado o desempenho geral semelhante da estratégia proposta em relação à HNCM, é interessante verificar a qualidade da solução obtida por estes métodos em relação à solução ótima das instâncias. Para isso, foi utilizado o método exato de enumeração proposto por Becceneri et al. (2004) e aprimorado por Yanasse et al. (2007). As instâncias de teste foram resolvidas usando o arquivo executável fornecido por estes autores, no mesmo ambiente computacional considerado anteriormente. O tempo de execução foi limitado em 500 segundos, porém, em 4 instâncias excedeu-se a memória computacional disponível sem que o método pudesse identificar a solução ótima, sendo considerada a melhor solução obtida nesses casos.

Para a comparação dos resultados, foi calculada a diferença entre a solução ótima e a solução obtida por cada método, em cada uma das instâncias de teste. Na Tabela 4.6, é apresentada a quantidade de instâncias para as quais cada método obteve a solução ótima ou igual à ótima (+0) e a quantidade de instâncias com diferença de 1, 2, 3 ou mais pilhas em relação à solução ótima (colunas +1, +2, +3 e >3, respectivamente). A quantidade de soluções em (+0) foi próxima para os métodos considerados. Entretanto, a diferença obtida pela estratégia proposta não ultrapassou 3 pilhas em nenhuma das instâncias (independente da forma de solução do TSP), enquanto a HNCM obteve uma diferença maior que três pilhas para 7 instâncias.

**Tabela 4.6:** Quantidade de instâncias para as quais é obtida a diferença de 0, 1, 2, 3 ou mais pilhas em relação à solução ótima do MOSP.

Método	+0	+1	+2	+3	>3
HID	295	60	21	2	0
HNCM	294	53	18	6	7
HIA	290	63	19	6	0

Um último teste foi realizado para verificar o quanto a qualidade da solução do MOSP é prejudicada devido à solução aproximada do TSP. Para isso, as instâncias de teste foram resolvidas pela estratégia proposta, utilizando a solução exata do TSP obtida pelo *software* Concorde (Applegate et al., 2009). O tempo de solução de cada instância não ultrapassou 8 segundos, no mesmo ambiente computacional considerado anteriormente.

Na Tabela 4.7, é dada a diferença entre a solução ótima e a solução obtida pela estratégia proposta usando a solução exata do TSP, conforme a densidade das instâncias. Em nenhuma das instâncias, a solução obtida apresentou diferença superior a 3 pilhas. Observe que a solução exata do TSP resultou na obtenção da solução ótima para o MOSP em 304 (80,42%) instâncias e para apenas 1 instância foi obtida a diferença de 3 pilhas em relação à solução ótima. Se comparados estes valores com os apresentados na Tabela 4.6, observa-se o melhor comportamento da estratégia proposta quando usada a solução exata do TSP. Portanto, uma melhor solução para o TSP resulta, em geral, em uma melhor solução para o MOSP, indicando que a conversão proposta é eficaz.

Note que, de modo geral, a estratégia proposta apresentou maior diferença em relação à solução ótima para as instâncias com densidade  $D \leq 0,300$ . É também nestas instâncias que o

**Tabela 4.7:** Quantidade de instâncias para as quais a solução do MOSP obtida pela estratégia proposta, utilizando a solução exata do TSP, possui diferença de 0, 1, 2 ou 3 pilhas em relação à solução ótima do MOSP.

Classes	D	+0	+1	+2	+3
wbo_10_10	0,200	8	2	0	0
	0,300	8	2	0	0
	0,400	10	0	0	0
	0,500	10	0	0	0
wbo_15_30	0,133	9	1	0	0
	0,200	8	2	0	0
	0,267	9	1	0	0
	0,333	10	0	0	0
	0,400	10	0	0	0
	0,467	10	0	0	0
wbo_30_30	0,067	1	6	3	0
	0,100	0	3	6	1
	0,133	1	6	3	0
	0,167	1	7	2	0
	0,200	5	5	0	0
	0,233	8	2	0	0
	0,267	9	1	0	0
	0,300	9	1	0	0
	0,333	10	0	0	0
	0,367	10	0	0	0
	0,400	10	0	0	0
	0,433	10	0	0	0
	0,467	10	0	0	0
	0,500	10	0	0	0

Classes	D	+0	+1	+2	+3
wbop_20_20	0,100	10	0	0	0
	0,150	4	6	0	0
	0,200	7	3	0	0
	0,250	9	1	0	0
	0,300	10	0	0	0
	0,350	10	0	0	0
	0,400	10	0	0	0
	0,450	10	0	0	0
wbp_30_10	0,200	6	3	1	0
	0,300	9	1	0	0
	0,400	10	0	0	0
	0,500	10	0	0	0
gp_50_50	0,539	0	1	0	0
	0,654	1	0	0	0
	0,632	0	1	0	0
	0,808	1	0	0	0
gp_100_100	0,543	0	1	0	0
	0,672	0	1	0	0
	0,677	0	1	0	0
	0,857	1	0	0	0
<b>Soma</b>		<b>304</b>	<b>58</b>	<b>15</b>	<b>1</b>

método exato de enumeração apresentou o maior tempo computacional. Para a classe wbo\_30\_30, o método exato de enumeração não obteve a solução ótima de 4 instâncias dentro do tempo especificado, cuja densidade é 0,067. Também teve dificuldade em resolver duas instâncias de densidade 0,200 da classe wbp\_30\_10, nos quais o tempo computacional foi de 281 e 317 segundos. Isto mostra a dificuldade em resolver as instâncias MOSP mais esparsas. Entretanto, vale ressaltar que não somente a densidade influencia na dificuldade de solução das instâncias, como pode ser observado para as classes gp\_50\_50 e gp\_100\_100, cujas instâncias são mais densas e possuem maior número de padrões e itens.

#### 4.1.4 Considerações finais

A utilização do TSP para a obtenção de uma solução para o MOSP se mostrou bastante eficiente na prática. A estratégia de solução proposta apresentou um bom desempenho geral em relação às heurísticas específicas para o MOSP, entre elas a HNCM. Considerando a solução aproximada do TSP, a HID obteve melhor desempenho quanto à solução obtida para o MOSP. Resultados ainda

melhores foram obtidos com a utilização da solução exata do TSP, que levou à solução ótima de 80,42% das instâncias de teste. Para as demais instâncias, a diferença da solução obtida pela estratégia em relação à solução ótima nunca ultrapassou 3 pilhas.

Como sugestão de trabalhos futuros, pode-se estudar a aplicação de heurísticas de melhoria sobre a solução obtida pela estratégia proposta utilizando, por exemplo, os métodos propostos por Fink e Voss (1999). Novas regras de atribuição de distâncias e a utilização do ATSP também podem ser verificadas, de modo a obter uma melhor aproximação para o MOSP.

## 4.2 Heurística de particionamento do grafo MOSP

Nesta Seção é apresentada a heurística de particionamento que está sendo desenvolvida. A ideia básica é decompor o problema MOSP em vários subproblemas e, a partir da solução destes subproblemas, obter a solução do problema MOSP original. No decorrer desta Seção serão discutidos os métodos utilizados para calcular os limitantes inferior e superior do problema MOSP, uma estratégia para particionar o problema e as relações de otimalidade consideradas no método heurístico.

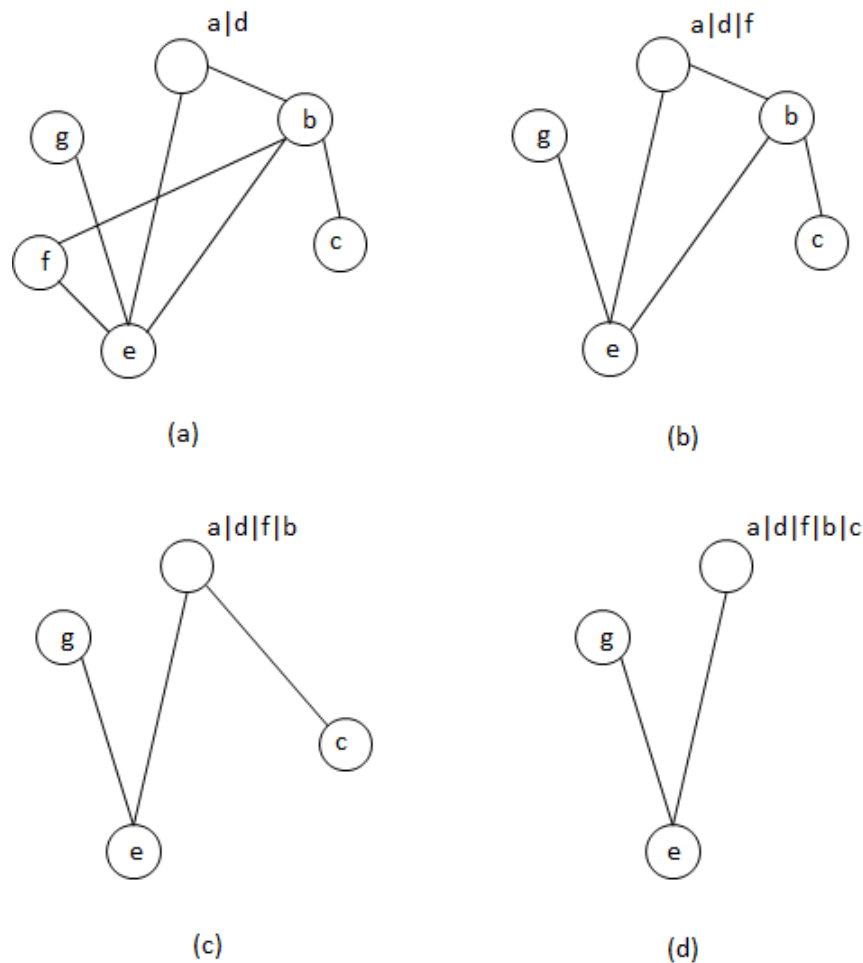
### 4.2.1 Os limitantes inferior e superior para o MOSP

Para calcular o limitante inferior do problema MOSP foi implementada um procedimento de contração de arcos HCA (Becceneri et al., 2004) descrita no Capítulo 2. Aplicando a heurística HCA no grafo MOSP de itens da Figura 2.1 são obtidos os grafos reduzidos mostrados na Figura 4.3 conforme explicação seguinte.

Inicialmente, calcula-se  $lb_p = 3$ ,  $lb_d = 2$  e  $lb = 3$ . No Passo 1 da heurística,  $G$  não é um grafo completo e não satisfaz  $|G| \leq lb$ , então calcula-se  $SetV = \{c, g, a, d, f, b, e\}$ . A aresta  $e = (a, d)$  é selecionada no início do Passo 2 e ao contraí-la é obtido o grafo  $G$  mostrado em (a) com valor  $dmin = 1$ . Volta ao Passo 1. É obtido um novo conjunto  $SetV = \{c, g, a|d, f, b, e\}$ . Ao contrair a aresta  $e = (a|d, f)$  escolhida, obtem-se o grafo  $G$  mostrado em (b) em que,  $dmin = 1$ . Volta ao Passo 1 e calcula um novo conjunto  $SetV = \{c, g, a|d|f, b, e\}$  e seleciona-se  $e = (a|d|f, b)$  que quando contraída retorna o grafo (c) com  $dmin = 1$ . Calcula o novo  $SetV = \{c, g, a|d|f|b, e\}$ , contrai  $e = (c, a|d|f|b)$  e retorna o grafo (d). No Passo 1, é satisfeita a restrição  $|G| = 3 \leq lb = 3$  e, assim,  $lb_c = 3$  e FIM da heurística.

Quando o grafo  $G$  do problema MOSP é disjunto (ou seja, é possível dividir os vértices em subgrafos que não estão ligados por nenhuma aresta), a heurística HCA é aplicada em cada um dos subgrafos disjuntos de  $G$  e o valor do limitante inferior é dado pelo maior  $lb_c$  obtido nos subgrafos. Para identificar os subgrafos disjuntos do grafo MOSP é utilizado um método de decomposição do grafo que também pode ser classificado como um método de pré-processamento.

Seja  $P$  o conjunto de padrões do problema. O método de decomposição é aplicado no grafo MOSP de padrões, ou seja, são obtidos  $K$  conjuntos disjuntos de padrões. Para cada um desses



**Figura 4.3:** Grafos reduzidos obtidos durante a heurística HCA.

subconjuntos  $sub(K)$  é construído o grafo MOSP de itens e aplicada a heurística HCA. Note que, se no grafo de padrões do problema MOSP existe algum padrão ou conjunto de padrões que formam um subgrafo disjunto, necessariamente os itens contidos nestes padrões estarão isolados dos demais itens. Portanto, encontrar subgrafos disjuntos no grafo de padrões corresponde a encontrar subgrafos disjuntos no grafo de itens.

No algoritmo de decomposição do grafo de padrões apresentado a seguir, considera-se um vetor  $aux$  para orientar a busca de padrões adjacentes (que tem pelo menos um item em comum) que pode assumir os valores: -1 se o padrão não é adjacente a nenhum padrão analisado, 0 se o padrão é adjacente a algum padrão analisado, ou 1 se o padrão já foi analisado.

---

### Decomposição do grafo de padrões

---

1.  $P \leftarrow \{\text{padrões}\}$  e  $K = 0$ .
2. Enquanto  $P \neq \{\}$  faça:
  - 2.1  $K = K + 1$  e  $sub(K) = \{\}$ .
  - 2.2  $aux[j] = -1, \forall j$  padrão do problema.



**2.3** Escolha  $j \in P$  e faça  $aux[j] = 0$ .

**2.4** Enquanto  $\exists j : aux[j] = 0$  faça:

- $aux[j] = 1$ .
- $P \leftarrow P - \{j\}$ .
- $sub(K) \leftarrow sub(K) \cup \{j\}$ .
- Faça  $aux[i] = 0, \forall i$  ligado a  $j$  por algum item.

**3.** Retorna  $K$  subgrafos encontrados.

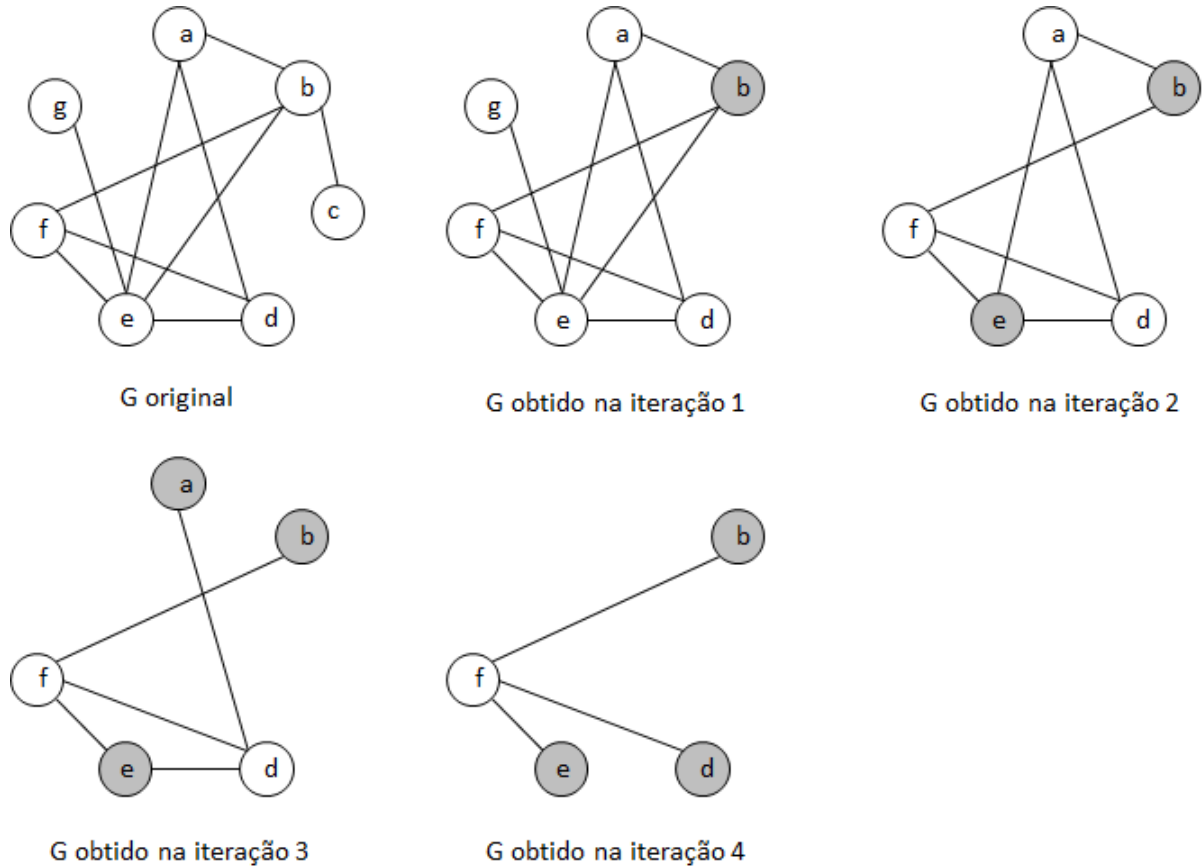
Para calcular um limitante superior para o MOSP foi utilizada a heurística de nó de custo mínimo HNCM (Becceneri et al., 2004) descrita no Capítulo ???. Um vértice é considerado aberto a partir do momento que uma aresta incidente a ele é percorrida e é dito fechado quando todas as arestas incidentes ao vértice foram percorridas. Na heurística é aplicada a ideia de percorrimento das arestas do grafo MOSP de itens segundo um critério guloso: tentar fechar primeiro os vértices de menor grau. Neste sentido, o método procura percorrer primeiro as arestas que ligam vértices de menor grau ou as arestas que ligam vértices abertos. As arestas são ordenadas conforme são percorridas e, a partir desta sequência de arestas é calculada a sequência de processamento dos padrões.

Aplicando a heurística HNCM (conforme descrição feita no Capítulo ???) no exemplo da Tabela 2.1 com grafo MOSP de itens mostrado na Figura 2.1, calcula-se inicialmente  $Open = \{\}$ ,  $Z = 0$ ,  $Arc = \{\}$ ,  $degree = [3, 4, 1, 3, 5, 3, 1]$  e  $SetV = \{c, g, a, d, f, b, e\}$ . A heurística pára quando  $|Arc| = |E|$ . A Figura 4.4 mostra o grafo obtido depois de cada uma das iterações da heurística, descritas a seguir. Os vértices hachurados no grafo são os vértices que permanecem abertos depois da iteração calculada.

**Iteração 1:** Não existem dois vértices adjacentes em  $SetV_{min} = \{c, g\}$ , então busca o vértice com menor grau  $v_j \in SetV - SetV_{min}$  que é adjacente a algum vértice  $v_i \in SetV_{min}$ . Seleciona  $v_i = c$  e  $v_j = b$ , estes vértices são considerados abertos  $open = \{c, b\}$ ,  $Z = 2$  e atualiza o conjunto de arestas percorridas  $Arc = \{(c, b)\}$ . Ao atualizar o vetor com o grau de cada vértice é obtido  $degree = [3, 3, 0, 3, 5, 3, 1]$  e pode-se fechar o vértice  $c$ ,  $Open = \{b\}$ , pois todas as arestas incidentes ao vértice foram percorridas. Atualiza  $SetV = \{g, a, b, d, f, e\}$  e  $SetV_{min} = \{g\}$ .

**Iteração 2:** Seleciona  $e \in SetV - SetV_{min}$  adjacente ao vértice  $g \in SetV_{min}$  que são adicionados ao conjunto de vértices abertos,  $Open = \{b, e, g\}$  e  $Z = 3$ . Existem duas arestas ligando qualquer dois vértices contidos em  $Open$ ,  $(g, e)$  e  $(b, e)$ , que são incluídas em  $Arc = \{(c, b), (g, e), (b, e)\}$ . Atualiza  $degree = [3, 2, 0, 3, 3, 3, 0]$  e  $Open = \{b, e\}$ .  $SetV = \{b, a, d, e, f\}$  e  $SetV_{min} = \{b\}$ .

**Iteração 3:** Seleciona os vértices adjacentes  $a \in SetV - SetV_{min}$  e  $b \in SetV_{min}$ . Com isso são obtidos os conjuntos  $Open = \{a, b, e\}$  e  $Arc = \{(c, b), (g, e), (b, e), (b, a), (a, e)\}$ . Atualiza  $degree = [1, 1, 0, 3, 2, 3, 0]$ ,  $SetV = \{a, b, e, d, f\}$  e  $SetV_{min} = \{a, b\}$ .



**Figura 4.4:** Grafos obtidos durante a heurística HNCM.

**Iteração 4:** Seleciona  $d \in SetV - SetV_{min}$  e  $a \in SetV_{min}$ . Agora são quatro vértices abertos  $Open = \{a, b, d, e\}$ , ou seja,  $Z = 4$ .  $Arc = \{(c, b), (g, e), (b, e), (b, a), (a, e), (a, d), (d, e)\}$ ,  $degree = [0, 1, 0, 1, 1, 3, 0]$  e  $Open = \{b, d, e\}$ .  $SetV = \{b, d, e, f\}$  e  $SetV_{min} = \{b, d, e\}$ .

**Iteração 5:** Seleciona  $f \in SetV - SetV_{min}$  e  $b \in SetV_{min}$  o que gera quatro vértices abertos  $Open = \{b, d, e, f\}$ . As três arestas  $(b, f)$ ,  $(d, f)$  e  $(e, f)$  são adicionadas ao conjunto  $Arc = \{(c, b), (g, e), (b, e), (b, a), (a, e), (a, d), (d, e), (b, f), (d, f), (e, f)\}$  e verifica-se  $|ARC| = |E|$ .

A heurística retorna a solução dada pela sequência de arestas  $Arc$  e  $Z = 4$ . A sequência de padrões é calculada baseando-se no percorrimento das arestas ordenadas em  $Arc$ . Logo que todas as arestas formadas com o padrão são percorridas, o padrão é incluído na sequência de corte. A Tabela 4.8 mostra as arestas formadas em cada um dos 6 padrões do exemplo trabalhado. A sequência de padrões obtida é  $S_P = P_5, P_3, P_6, P_1, P_2, P_4$  e corresponde ao número máximo de pilhas abertas  $Z = 4$ .

**Tabela 4.8:** Arestas formadas com os padrões de corte da Tabela 2.1.

Padrão	Arestas
$P_1$	$(a, d), (a, e), (d, e)$
$P_2$	$(b, e), (b, f), (e, f)$
$P_3$	$(e, g)$
$P_4$	$(d, e), (d, f), (e, f)$
$P_5$	$(b, c)$
$P_6$	$(a, b), (a, e), (b, e)$

## 4.2.2 Heurística de particionamento estudada

Trabalha-se aqui com o grafo MOSP de itens indicado como  $G(V, E)$  em que  $V$  é o conjunto de vértices e  $E$  o conjunto de arestas do grafo conforme definido na Seção 2.1. De maneira bastante resumida é dada a seguinte estrutura da heurística de particionamento:

---

### Heurística de particionamento

---

**Passo 0:** Inicialização

1.  $Z^G = UB$  e  $S_P^G = S_P^{UB}$ .

Se  $Z^G = LB$ , então é ótimo. FIM.

**Passo 1:** Particionamento: o corte da sequência inicial de padrões.

**Passo 2:** Solução exata ou heurística dos subproblemas e análise de otimalidade.

---

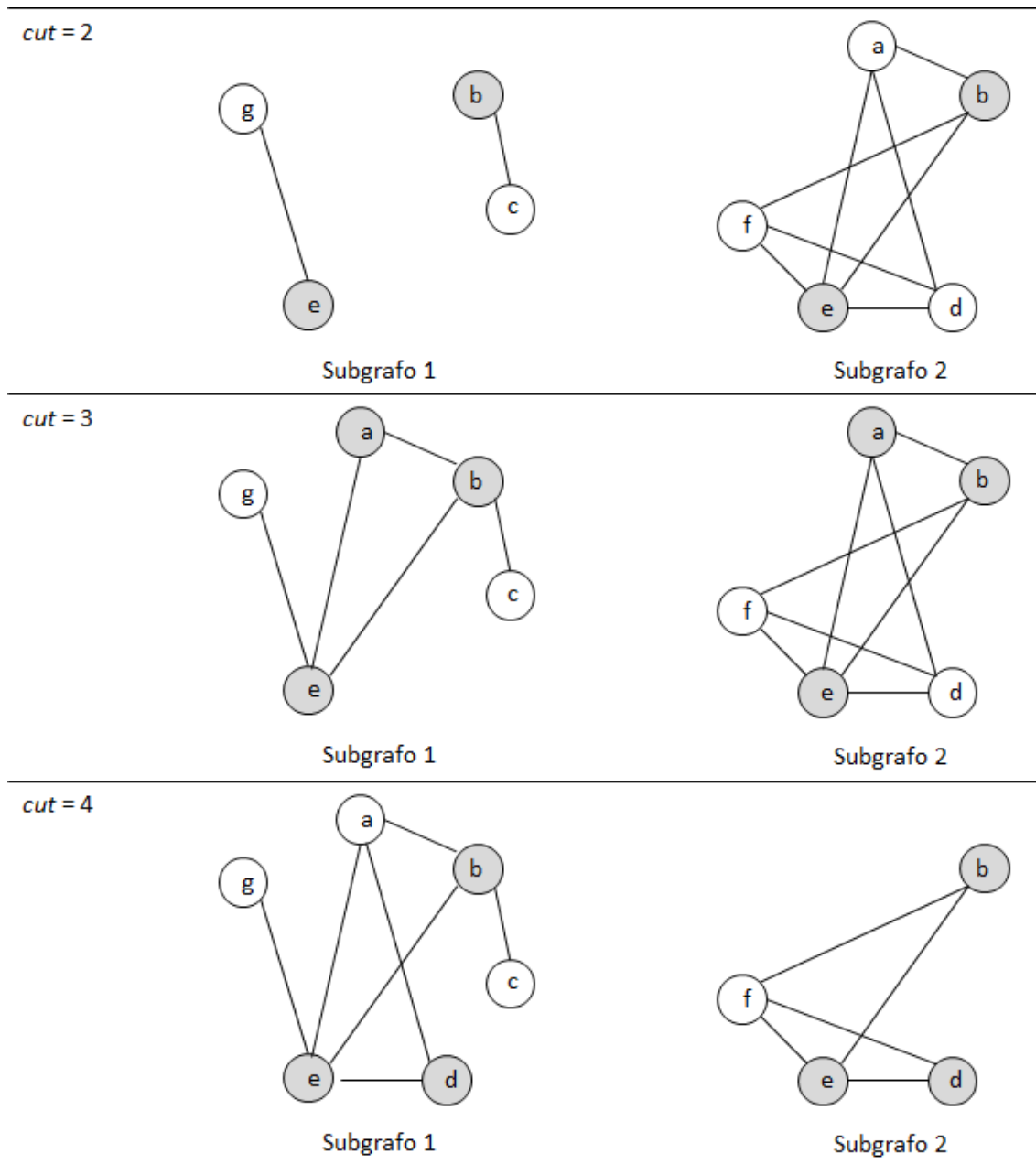
em que,  $LB$  é o valor do limitante inferior do problema MOSP,  $UB$  é o valor do limitante superior e  $S_P^{UB}$  a sequência de padrões correspondente ao  $UB$ . A heurística retorna o valor da solução MOSP  $Z^G$  e a sequência de padrões  $S_P^G$  correspondente.

### O particionamento do problema MOSP

Seja o conjunto de padrões do problema. A ideia é particionar este conjunto de padrões em dois subconjuntos. Cada um dos subconjuntos vai gerar um novo grafo MOSP de itens, que será chamado subgrafo de  $G$ .

A estratégia utilizada para particionar o conjunto de padrões consiste em escolher um ponto de corte numa sequência de padrões inicial (corte da sequência de padrões). O corte (*cut*) pode ser feito em todas as posições da sequência de padrões inicial, ou seja, o valor de *cut* pode assumir valores inteiros no intervalo  $[1, N]$ . O subgrafo 1 é composto pelos padrões das posições  $1, \dots, cut$  da sequência de padrões inicial e o subgrafo 2 é formado pelos padrões das posições  $(cut + 1), \dots, N$ . Ao considerar  $cut = N$ , não há divisão do conjunto de padrões, já que, todos os padrões serão atribuídos ao subgrafo 1 e o subgrafo 2 será vazio. Para tentar evitar a geração de subgrafos com tamanho (número de vértices) igual ou próximo ao do grafo  $G$  é definida uma faixa de possíveis

posições para o corte, ou seja,  $cut \in [cut_{min}, cut_{max}]$ . Quando os subgrafos 1 e 2 são gerados, pode ocorrer de alguns itens pertencerem tanto ao subgrafo 1 como ao subgrafo 2. Estes itens comuns serão chamados de itens ou vértices de fronteira.



**Figura 4.5:** Subgrafos obtidos com  $cut \in [2, 4]$ .

Voltando ao problema MOSP do exemplo dado na Tabela 2.1. Seja  $S_P = (P_5, P_3, P_6, P_1, P_2, P_4)$  a sequência de padrões inicial e  $cut \in [2, 4]$  a faixa de possíveis cortes. A Figura 4.5 mostra os subgrafos obtidos para os valores de  $cut$ , nos quais, os vértices comuns nos dois subgrafos de cada corte aparecem hachurados. Para  $cut = 2$ , o subgrafo 1 é formado pelos padrões  $(P_5, P_3)$ , o subgrafo 2 pelos padrões  $(P_6, P_1, P_2, P_4)$  e a quantidade de vértices de fronteira é dois (vértices b e e). Para  $cut = 3$ , o subgrafo 1 é formado pelos padrões  $(P_5, P_3, P_6)$ , o subgrafo 2 pelos padrões  $(P_1, P_2, P_4)$  e os vértices de fronteira são a, b e e. Quando  $cut = 3$  há três vértices de fronteira (b, d

e  $e$ ), em que o subgrafo 1 é formado pelos padrões  $(P_5, P_3, P_6, P_1)$  e o subgrafo 2 pelos padrões  $(P_2, P_4)$ . Apenas um dos três possíveis cortes é selecionado: o corte que gera o menor número de vértices de fronteira, que neste exemplo, é o corte  $cut = 2$ .

Após obtidos os dois subgrafos, estes serão resolvidos separadamente apenas se o número de vértices de fronteira for menor ou igual ao limitante inferior ( $LB$ ) do problema ( $Boundary[cut] \leq LB$ ), caso contrário, resolve-se o problema sem particionar o grafo  $G$ . Esta condição foi adotada após a realização de alguns testes computacionais que serão discutidos na Seção 4.2.3.

---

**Passo 1:** Particionamento: corte da sequência inicial de padrões

---

1. Fixa  $[cut_{min}, cut_{max}]$ .
  2. Calcula  $Boundary[i], i = 1, \dots, N$ .
  3. Seleciona  $cut : Boundary[cut] = \min\{Boundary[i], i = cut_{min}, \dots, cut_{max}\}$ .
  4. Se  $Boundary[cut] > LB$  então:
 

Resolve  $G$  sem particionar e retorna o ótimo  $Z^G$  e  $S_P^G$ . FIM.
  5. Gera os subgrafos 1 e 2:
 

Subgrafo 1  $\leftarrow \{S_P^{UB}[i], i = 1, \dots, cut\}$ ;

Subgrafo 2  $\leftarrow \{S_P^{UB}[i], i = (cut + 1), \dots, N\}$ .
- 

### Solução exata ou heurística dos subproblemas e análise de otimalidade

Considere a seguinte notação utilizada no decorrer do texto:

- $O_i$  e  $H_i$  o valor da solução ótima e heurística do grafo  $i$ ;
- $S_P^{O_i}$  e  $S_P^{H_i}$  a sequência de padrões correspondente a  $O_i$  e  $H_i$ ;
- $O_i^*$  e  $H_i^*$  o valor da solução ótima e heurística do grafo  $i$  considerando os vértices de fronteira abertos no primeiro estágio de corte dos padrões;
- $S_P^{O_i^*}$  e  $S_P^{H_i^*}$  a sequência de padrões correspondente a  $O_i^*$  e  $H_i^*$ .

Inicialmente, calcula-se a solução heurística de cada subgrafo considerando os vértices de fronteira abertos no início do sequenciamento. Esta solução heurística  $H^*$  pode ser obtida a partir da sequência de padrões inicial. Seja  $stacks[i]$  o número de pilhas abertas em cada estágio de corte  $i, i = 1, \dots, N$ . Trabalhando novamente com o exemplo MOSP dado anteriormente, da sequência de corte  $(P_5, P_3, P_6, P_1, P_2, P_4)$  é obtido o vetor  $stacks = [2, 3, 3, 4, 4, 3]$ , conforme mostrado na Tabela 2.2 da Seção 2.1. O valor da solução heurística  $H^*$  dos subgrafos é dado por  $Z^{H^*} = 4$ .

Considera-se, sem perda de generalidade,  $S1$  o subgrafo com o maior valor da solução heurística e  $S2$  o outro subgrafo. Sendo assim, no exemplo apresentado ocorre que  $S1$  recebe os padrões do subgrafo 2,  $S2$  recebe os padrões do subgrafo 1,  $H_{S1}^* = 4$  e  $H_{S2}^* = 3$ . Em outras palavras,

$H_{S_1}^* \geq H_{S_2}^*$  indica que a solução do subgrafo  $S_2$  não piora o valor da solução do subgrafo  $S_1$  quando são juntadas as sequências de  $S_P^{H_{S_1}^*}$  e  $S_P^{H_{S_2}^*}$  para formar a solução do grafo  $G$ .

No Passo 2 da heurística busca-se satisfazer algumas relações entre as soluções ótima e heurística dos subgrafos obtidos do particionamento. O objetivo é mostrar que uma solução é ótima para o grafo  $G$  original. Intuitivamente, é analisado primeiro o subgrafo  $S_1$  que, como definido anteriormente, tem o maior número máximo de pilhas abertas  $H_{S_1}^*$ .

Calcula-se a solução ótima do subgrafo  $S_1$  sem estabelecer pilhas (vértices) abertas no início do sequenciamento, o valor MOSP obtido é dado por  $O_{S_1}$ . Se  $H_{S_1}^* = O_{S_1}$ , ou seja, se a solução heurística considerando pilhas abertas no início do processo de corte tem o mesmo valor que a solução ótima de  $S_1$  sem restrições, então a sequência de padrões  $S_P^{H_{S_1}^*}$  é uma sequência ótima do subgrafo  $S_1$ .

Se satisfeitas as relações  $H_{S_1}^* \geq H_{S_2}^*$  e  $H_{S_1}^* = O_{S_1}$ , a sequência de padrões  $S_P^G = [S_P^{H_{S_1}^*}, S_P^{H_{S_2}^*}] = S_P^{UB}$  com valor MOSP  $Z^G = O_{S_1}$  é uma solução ótima para  $G$ . Caso esta relação de otimalidade não seja satisfeita tenta-se melhorar o limitante inferior  $LB$  de  $G$  fazendo  $LB = \max\{LB, O_{S_1}\}$ .

Outra relação de otimalidade implementada utiliza a solução ótima do subgrafo  $S_1$  considerando os vértices de fronteira abertos no primeiro estágio de corte dos padrões. O valor desta solução é dado por  $O_{S_1}^*$  correspondente à sequência de padrões  $S_P^{O_{S_1}^*}$ .

Se ocorrer  $O_{S_1}^* = O_{S_1}$ , sabe-se que ao estabelecer que as pilhas dos vértices de fronteira estejam abertas no início do sequenciamento não se piora a solução obtida sem esta restrição. E se, além disso, a solução  $H_{S_2}^*$  é menor ou igual a  $O_{S_1}^*$  então a solução MOSP com sequência de padrões  $[S_P^{O_{S_1}^*}, S_P^{H_{S_2}^*}]$  e número máximo de pilhas abertas igual a  $O_{S_1}^*$  é ótima. Caso a relação de otimalidade não for satisfeita mas, valem as relações  $O_{S_1}^* = O_{S_1} < H_{S_2}^*$  e  $UB > H_{S_2}^*$ , então melhora-se o valor do limitante superior considerando  $UB = H_{S_2}^*$  e  $S_P^{UB} = [S_P^{O_{S_1}^*}, S_P^{H_{S_2}^*}]$ .

De início  $H_{S_1}^* \geq H_{S_2}^*$  e, intuitivamente, foi explorado o subgrafo  $S_1$  pensando que o sequenciamento desse subconjunto de padrões é que gerava o máximo número de pilhas abertas do problema dado por  $G$ . Porém, depois de calcular os valores  $O_{S_1}^* = O_{S_1}$  e constatar que  $H_{S_2}^* > O_{S_1}^*$  pode ser que o número máximo de pilhas abertas ocorra no sequenciamento do padrões do subgrafo  $S_2$ . Nestes casos, o subgrafo  $S_2$  também precisa ser analisado.

Calcula-se a solução ótima do subgrafo  $S_2$  considerando os vértices de fronteira abertos no primeiro estágio de corte dos padrões. Seja  $O_{S_2}^*$  e  $S_P^{O_{S_2}^*}$  a solução obtida. Se for satisfeita a relação  $O_{S_1}^* = O_{S_1} \geq O_{S_2}^*$ , então a sequência de padrões  $[S_P^{O_{S_1}^*}, S_P^{O_{S_2}^*}]$  com valor MOSP  $O_{S_1}^*$  é uma solução ótima do grafo  $G$ . Ou seja, é mostrado que o subconjunto de padrões em  $S_2$  não gera um número máximo de pilhas abertas maior que o valor ótimo  $O_{S_1}^* = O_{S_1}$  obtido em  $S_1$ .

Não satisfeita a relação de otimalidade, a solução  $O_{S_2}^*$  pode melhorar o valor do limitante superior. Se  $O_{S_1}^* = O_{S_1} < O_{S_2}^*$  e  $UB > O_{S_2}^*$ , então  $UB = O_{S_2}^*$  e  $S_P^{UB} = [S_P^{O_{S_1}^*}, S_P^{O_{S_2}^*}]$ .

Foram mostradas até aqui três relações de otimalidade:

- i)** Se  $H_{S_1}^* \geq H_{S_2}^*$  e  $H_{S_1}^* = O_{S_1}$ , então  $Z^G = O_{S_1}$  e  $S_P^G = [S_P^{H_{S_1}^*}, S_P^{H_{S_2}^*}] = S_P^{UB}$  é ótimo de  $G$ ;

ii)  $O_{S_1}^* = O_{S_1} \geq H_{S_2}^*$ , então  $Z^G = O_{S_1}^*$  e  $S_P^G = [S_P^{O_{S_1}^*}, S_P^{H_{S_2}^*}]$  é ótimo de  $G$ ;

iii)  $O_{S_1}^* = O_{S_1} \geq O_{S_2}^*$ , então  $Z^G = O_{S_1}^*$  e  $S_P^G = [S_P^{O_{S_1}^*}, S_P^{O_{S_2}^*}]$  é ótimo de  $G$ .

uma atualização para o valor  $LB$  e duas atualizações para a solução do  $UB$ :

iv) Se  $O_{S_1} < H_{S_1}^*$  e  $LB < O_{S_1}$ , então  $LB = O_{S_1}$ .

v) Se  $O_{S_1}^* = O_{S_1} < H_{S_2}^*$  e  $UB > H_{S_2}^*$ , então  $UB = H_{S_2}^*$  e  $S_P^{UB} = [S_P^{O_{S_1}^*}, S_P^{H_{S_2}^*}]$ ;

vi)  $O_{S_1}^* = O_{S_1} < O_{S_2}^*$  e  $UB > O_{S_2}^*$ , então  $UB = O_{S_2}^*$  e  $S_P^{UB} = [S_P^{O_{S_1}^*}, S_P^{O_{S_2}^*}]$ .

Alguns resultados são analisados para verificar a eficácia desses limitantes, conforme mostrado na Seção 4.2.3. O algoritmo implementado no Passo 2 consiste da solução dos subgrafos MOSP e do teste das relações de otimalidade definidas anteriormente.

---

### Passo 2: Solução exata ou heurística dos subproblemas e análise de otimalidade

---

1. Faça  $a = \max\{stacks[i], i = 1, \dots, cut\}$  e  $b = \max\{stacks[i], i = (cut + 1), \dots, N\}$ .

2. Seja  $S_1$  o subgrafo com  $H_{S_1}^* = \max\{a, b\}$  e  $S_2$  o outro subgrafo,  $H_{S_2}^* = \min\{a, b\}$ .

3. Calcula  $O_{S_1}$ .

Se  $O_{S_1} = H_{S_1}^*$ , então  $Z^G = O_{S_1}$  e  $S_P^G = S_P^{UB}$  é ótimo. Vá para 5.

Senão  $LB = \max\{LB, O_{S_1}\}$ .

4. Calcula  $O_{S_1}^*$ .

4.1. Se  $O_{S_1}^* = O_{S_1}$ , então

4.1.1. Se  $O_{S_1}^* \geq H_{S_2}^*$ , então  $Z^G = O_{S_1}^*$  e  $S_P^G = [S_P^{O_{S_1}^*}, S_P^{H_{S_2}^*}]$  é ótimo. Vá para 5.

4.1.2.  $UB = \min\{UB, H_{S_2}^*\}$ .

Se  $UB = LB$ , então  $Z^G = H_{S_2}^*$  e  $S_P^G = [S_P^{O_{S_1}^*}, S_P^{H_{S_2}^*}]$  é ótimo. Vá para 5.

4.1.3. Calcula  $O_{S_2}^*$ .

- Se  $O_{S_1}^* \geq O_{S_2}^*$ , então  $Z^G = O_{S_1}^*$  e  $S_P^G = [S_P^{O_{S_1}^*}, S_P^{O_{S_2}^*}]$  é ótimo. Vá para 5.

- $UB = \min\{UB, O_{S_2}^*\}$ .

- Se  $UB = LB$ , então  $Z^G = O_{S_2}^*$  e  $S_P^G = [S_P^{O_{S_1}^*}, S_P^{O_{S_2}^*}]$  é ótimo. Vá para 5.

4.2. Retorna solução  $Z^G$  e  $S_P^G$  sem garantia de otimalidade.

5. Retorna solução ótima  $Z^G$  e  $S_P^G$ .

Para obter a solução ótima dos grafos na heurística é aplicado o modelo MOSP apresentado em Yanasse e Pinto (2003) e descrito na Seção 3.1. O modelo é utilizado para resolver o grafo e os subgrafos MOSP. O modelo determina a ordem em que as pilhas dos itens são fechadas. É importante observar que é usada uma sequência de padrões inicial para particionar o grafo  $G$ , mas

ao resolver cada subproblema com o modelo de Yanasse e Pinto (2003) obtém-se uma sequência de itens. A sequência de padrões é calculada a partir desta sequência de itens obtida com o modelo.

Quando há a necessidade de considerar algumas pilhas abertas no início do sequenciamento, fixa-se o valor de algumas variáveis  $\mathbf{W}_{ij}$  do modelo. Seja  $b$  e  $e$  as pilhas que precisam estar abertas já no primeiro estágio de corte (pois correspondem aos vértices de fronteira gerados pelo corte aplicado no Passo 1 da heurística de particionamento), então  $\mathbf{W}_{bj} = 1, j = 1, \dots, M$  e  $\mathbf{W}_{ej} = 1, j = 1, \dots, M$ .

### 4.2.3 Experimentos computacionais

Nesta seção são mostrados alguns resultados obtidos resolvendo problemas apresentados na literatura. As implementações computacionais foram feitas em linguagem C e compiladas usando o mingw32 3.4.2, uma versão do compilador gcc para sistemas operacionais Microsoft Windows 32 bits, disponível na interface Dev-C++. Todos os testes foram executados em um computador com processador Pentium 4, 3.0 GHz, 2 GB de RAM e sistema operacional Microsoft Windows XP.

#### Problemas de teste

As instâncias de teste foram apresentadas em Smith e Gent (2005). Na Tabela 4.9, são descritas as classes de instâncias aqui consideradas. Nas classes wbo, os problemas tem número fixo de itens por padrão. Para as classes wbp, tem-se instâncias nas quais cada item está contido em um número fixo de padrões. A classe wbop é constituída por instâncias nas quais o número de itens por padrão é igual ao número de padrões que contém este tipo de item. No nome das classes também estão indicados o número de tipos de itens ( $M$ ) e o número de padrões ( $N$ ) das instâncias que as compõem. Na coluna  $[cut_{min}, cut_{max}]$  está indicada a faixa de corte  $[cut_{min}, cut_{max}]$  considerada para cada classe.

**Tabela 4.9:** Classes de instâncias utilizadas nos testes computacionais e faixa de corte considerada para cada classe.

Classe	Problemas	$M$	$N$	$[cut_{min}, cut_{max}]$
wbo_10_10	40	10	10	[2,8]
wbo_15_30	120	15	30	[10,20]
wbo_30_30	140	30	30	[10,20]
wbp_30_10	40	30	10	[2,8]
wbop_20_20	90	20	20	[5,15]



## Solução do modelo MOSP

O modelo inteiro misto para o MOSP proposto por Yanasse e Pinto (2003) foi resolvido usando a biblioteca *Callable Library* disponível no CPLEX 12.1. Foi definido um tempo máximo para solução de cada instância igual a 30 minutos (1800 segundos).

Quando calculado  $O_{S1}$ , se o tempo limite é atingido sem que uma solução ótima seja obtida, considera-se  $O_{S1}$  igual ao valor do limitante inferior obtido com o CPLEX. Esta consideração não afeta as relações de otimalidade definidas na Seção 4.2.2.

Se não é obtida uma solução ótima da instância ao se calcular o valor  $O_{S1}^*$ , considera-se  $O_{S1}^*$  como a melhor solução obtida com o CPLEX. (Isto não afeta as relações de otimalidade definidas na Seção 4.2.2). Note que, nesse caso será utilizada também a sequência de padrões correspondente a  $O_{S1}^*$ .

## Discussão dos resultados

Nas Tabelas 4.10 e 4.11 é mostrada a quantidade de instâncias que satisfazem cada um dos testes de parada aplicados na Heurística de particionamento. Além disso, a coluna “Não part.” mostra a quantidade de instâncias em que  $G$  não é particionado, pois, não satisfaz  $Boundary[cut] \leq LB$ . A Tabela 4.11 não possui esta coluna pois todas as instâncias são particionados. Na coluna “Não é ótima” são contabilizados as instâncias em que não foi possível mostrar a otimalidade, ou seja, o número de instâncias em que o grafo  $G$  foi particionado mas nenhuma das relações de otimalidade ( $UB = LB$ ,  $H_{S1}^* = O_{S1}$ ,  $O_{S1}^* = O_{S1}$  e  $O_{S1}^* \geq H_{S2}^*$ ,  $O_{S1} = O_{S1}^*$  e  $O_{S1}^* \geq O_{S2}^*$ ) foi satisfeita.

**Tabela 4.10:** Número de instâncias que satisfazem cada um dos testes de parada aplicados na Heurística de particionamento considerando a condição  $Boundary[cut] \leq LB$ .

Classe	$UB = LB$	Não part.	$H_{S1}^* = O_{S1}$	$O_{S1}^* = O_{S1}$ e $O_{S1}^* \geq H_{S2}^*$	Não é ótima	$O_{S1} = O_{S1}^*$ e $O_{S1}^* \geq O_{S2}^*$
wbo_10_10	4	14	10	2	9	1
wbo_15_30	18	32	0	0	10	0
wbo_30_15	4	86	5	0	25	0
wbo_30_30	30	95	0	0	15	0
wbop_20_20	44	16	0	0	30	0
wbp_30_10	9	0	0	0	31	0

Ao particionar somente os grafos que satisfazem a relação  $Boundary[cut] \leq LB$ , tenta-se evitar gastar tempo computacional resolvendo um particionamento que não mostrará uma solução ótima para o problema. Realmente, das Tabelas 4.10 e 4.11, percebe-se que a condição imposta funciona bem. Isto por que, a maioria das instâncias que aparecem na coluna “Não part.” da Tabela 4.10 são remanejados para a coluna “Não é ótima” da Tabela 4.11. Poucas são as instâncias, antes não particionadas, em que é mostrada uma solução ótima quando aplicado o particionamento. Na classe wbo\_10\_10, por exemplo, das 14 instâncias não particionadas apenas 4 satisfazem  $H_{S1}^* =$

**Tabela 4.11:** Número de instâncias que satisfazem cada um dos testes de parada aplicados na Heurística de particionamento quando todas as instâncias são particionadas.

Classe	$UB = LB$	$H_{S1}^* = O_{S1}$	$O_{S1}^* = O_{S1}$ e $O_{S1}^* \geq H_{S2}^*$	Não é ótima	$O_{S1} = O_{S1}^*$ e $O_{S1}^* \geq O_{S2}^*$
wbo_10_10	4	14	2	19	1
wbo_15_30	18	0	1	41	0
wbo_30_15	4	9	1	106	0
wbo_30_30	30	0	0	110	0
wbop_20_20	44	0	0	46	0
wbp_30_10	9	0	0	31	0

$O_{S1}$  e as demais são classificadas como “Não é ótima”. Na classe wbo\_15\_30, em apenas 1 das 32 instâncias antes não particionadas é obtida uma solução ótima e, na classe wbo\_30\_15, isso acontece em 5 das 86 instâncias.

Muitas das instâncias param com  $UB = LB$  antes mesmo de tentar o particionamento. Isso mostra que as heurísticas aplicadas para calcular os limitantes superior e inferior do problema são boas. Além disso, a atualização do valor de  $LB$  definida na Seção 4.2.2 é ativada em várias instâncias. A quantidade de instâncias de cada classe nas quais foi atualizado o valor do  $LB$  é: 3 em wbo\_10\_10, 0 em wbo\_15\_30, 16 em wbo\_30\_15, 1 em wbo\_30\_30, 8 em wbop\_20\_20 e 22 em wbp\_30\_10.

Mas, quando é aplicado o particionamento, mesmo sem mostrar uma solução ótima para a instância, será que a solução obtida com o particionamento é melhor que aquela obtida inicialmente com a heurística de nó de custo mínimo? No algoritmo do Passo 2 da heurística de particionamento apresentado na Seção 4.2.2, nota-se que as instâncias que não satisfazem  $O_{S1}^* = O_{S1}$  não atualizam o valor de  $UB$ , logo, retornam uma solução igual ao  $UB$  calculado aplicando a heurística de nó de custo mínimo. Nas demais instâncias, o  $UB$  também não foi atualizado em nenhum momento pelas relações v) e vi) definidas na Seção 4.2.2.

**Tabela 4.12:** Soma do tempo computacional médio (em segundos) utilizado para resolver cada subproblema gerado em cada classe com o CPLEX quando todas as instâncias são particionadas.

Classe	$T_{S1}$	$T_{S1}^*$	$T_{S2}^*$	Total
wbo_10_10	2,114	0,496	0,016	2,626
wbo_15_30	41,722	1,985	0,000	43,707
wbo_30_15	81176,859	13154,451	0,000	94331,310
wbo_30_30	74426,796	509,609	0,000	74936,405
wbop_20_20	3286,595	37,001	0,000	3323,596
wbp_30_10	31524,881	14305,342	0,000	45830,223

Nas Tabelas 4.12 e 4.13 é mostrado o tempo computacional (em segundos) para resolver os subproblemas  $S1$  e  $S2$  gerados em cada instância das classes do experimento. Os subproblemas são resolvidos no CPLEX aplicando o modelo matemático de Yanasse (1997a). As colunas  $T_{S1}$ ,

**Tabela 4.13:** Soma do tempo computacional médio (em segundos) utilizado para resolver cada subproblema gerado em cada classe com o CPLEX quando considerada a restrição  $Boundary[cut] \leq LB$  do particionamento.

Classe	$T_{S1}$	$T_{S1}^*$	$T_{S2}^*$	Total
wbo_10_10	1,313	0,282	0,015	1,610
wbo_15_30	3,217	0,452	0,000	3,669
wbo_30_15	821,283	36,485	0,000	857,768
wbo_30_30	311,079	17,454	0,000	328,533
wbop_20_20	1112,095	16,721	0,000	1128,816
wbp_30_10	31459,030	14314,800	0,000	45773,830

$T_{S1}^*$ ,  $T_{S2}^*$  das duas tabelas indicam, o tempo computacional médio para resolver cada um dos subproblemas  $O_{S1}$ ,  $O_{S1}^*$ ,  $O_{S2}^*$  da instância de cada classe. “Total” indica o tempo computacional médio para cada instância das classes consideradas. Os resultados da Tabela 4.13 são obtidos considerando a restrição  $Boundary[cut] \leq LB$  no particionamento, enquanto que, na Tabela 4.12 todas as instâncias são particionadas.

Das Tabelas anteriores, foi verificado que não se perdem muitas soluções ótimas ao utilizar a condição  $Boundary[cut] \leq LB$  para o particionamento. Nas Tabelas 4.12 e 4.13 verifica-se a significativa redução do tempo computacional utilizado para resolver os subproblemas gerados em cada classe. Num próximo experimento será interessante comparar o tempo computacional para resolver o grafo  $G$  original aplicando o mesmo modelo matemático usado aqui com o tempo total utilizado aplicando a heurística de particionamento.

Quando todos os grafos são particionados, em 87 dos subgrafos resolvidos foi atingido o tempo limite de 1800 segundos imposto ao usar o CPLEX. Quando adotada a condição  $Boundary[cut] \leq LB$  muitas das instâncias que geram esses subgrafos não são particionadas e, com isso, o número de subproblemas não resolvidos em 1800 segundos no CPLEX diminui para 19. Veja os dados da Tabela 4.14.

**Tabela 4.14:** Quantidade de subgrafos de cada classe em que é atingido o tempo limite de 1800 segundos fixado para o CPLEX, utilizando ou não a condição  $Boundary[cut] \leq LB$ .

$Boundary[cut] \leq LB$	Classe	$T_{S1}$	$T_{S1}^*$	$T_{S2}^*$
Não	wbo_30_15	35	5	0
	wbo_30_30	27	0	0
	wbp_30_10	13	7	0
Sim	wbp_30_10	12	7	0

#### 4.2.4 Considerações finais

Nos resultados apresentados, verificou-se que a restrição de particionar apenas os grafos com número de vértices de borda menor ou igual ao valor do limitante inferior ( $Boundary[cut] \leq$

$LB$ ) reflete bem a capacidade de se obter uma solução ótima do problema pela heurística de particionamento, já que, nas instâncias com  $Boundary[cut] > LB$  dificilmente é obtida uma solução ótima se aplicada a heurística. Estas instâncias em que o particionamento gera  $Boundary[cut] > LB$  correspondem à grafos mais densos. Um dos objetivos da nova metodologia de solução do problema MOSP é focar em problemas que geram grafos menos densos, pois, é nesse caso que os demais métodos da literatura apresentam dificuldade.

Alguns pontos ainda precisam ser investigados como, por exemplo, em relação à quantidade de subgrafos nos quais, atinge-se o tempo de 1800 segundos quando resolvidos no CPLEX aplicando o modelo matemático. O uso do modelo proposto por Yanasse e Pinto (2003) para obter uma solução ótima para o MOSP limita o desempenho da heurística quando os subproblemas gerados têm mais de 15 tipos de itens. Além disso, o modelo não apresenta bom desempenho em problemas esparsos (isto é, com matriz de tipos itens por padrões esparsa) contrariando a proposta da heurística de particionamento de resolver os problemas mais esparsos.

A heurística está limitada ao particionamento do problema em apenas dois subproblemas. Ideias futuras consistem em aplicar a heurística de forma iterativa nos subproblemas obtidos em cada iteração. Esta estratégia permitiria resolver problemas grandes (em relação ao número de tipos de itens e padrões), já que os problemas poderiam ser reduzidos até um tamanho aceitável para o método de solução aplicado.

---

## Conclusão

---

Esta tese propôs novas contribuições para o problema de minimização de pilhas abertas (MOSP), divididas em duas partes principais. Primeiramente, foram analisados os modelos lineares inteiro-mistos existentes e novas propostas foram efetuadas. Em seguida, dois métodos heurísticos foram propostos.

Em relação aos modelos matemáticos, as desigualdades válidas propostas com o intuito de fortalecer o modelo original considerado permitiram a resolução de mais instâncias do problema, de forma mais rápida. Por outro, a aplicabilidade deste modelo está limitado a instâncias de pequeno porte. Um caminho de pesquisa suscitado visa a obtenção de melhores relaxações lineares.

Um outro desenvolvimento importante no tocante ao desenvolvimento de modelos foi a proposta de um modelo integrado MOSP/MORP. Em termos computacionais, o tempo necessário para obter a solução do modelo mostrou-se bastante sensível ao ajuste de parâmetros disponíveis nos métodos utilizados. De forma geral, pode-se dizer que esta proposta permite a integração de dois problemas altamente relacionados e gera perspectiva de uma nova linha de pesquisa, que pode ser explorada no futuro.

Em uma outra linha de pesquisa, investigaram-se métodos heurísticos de resolução do MOSP. O primeiro método relaciona heurísticamente o grafo MOSP com o grafo associado que representa um problema do caixeiro viajante (TSP), através da escolha apropriada das ‘distâncias’ dos arcos do problema TSP baseada nos graus dos nós do grafo MOSP. A vantagem de se utilizar uma estratégia de conversão do problema MOSP para o problema TSP é a possibilidade de se usar o vasto arsenal de métodos disponíveis para a resolução deste último problema. De fato, a estratégia se mostrou eficiente: por um lado, a conversão MOSP-TSP se mostrou adequada, dado que havia uma alta correlação entre boas soluções do TSP e boas soluções do MOSP associado. Por outro

lado, a estratégia se mostrou competitiva com outros métodos heurísticos desenvolvidos puramente para o MOSP, obtendo a solução ótima para o MOSP em 80,42% das instâncias testadas e obtendo soluções para as instâncias restantes nunca distantes mais que três pilhas da solução ótima. Tais resultados sugerem a continuação desta linha, por exemplo, através do estudo de novas regras de atribuição de distâncias ou da utilização de variações do TSP que podem, eventualmente, ser mais apropriadas, como o problema do caixeiro viajante assimétrico.

O segundo método heurístico proposto se baseou na decomposição do problema original em subproblemas menores através da obtenção de um corte no grafo MOSP associado ao problema. A ideia foi obter a resolução dos problemas menores e tentar garantir a otimalidade da solução conjunta obtida através da análise dos itens presentes na fronteira (associados a vértices incididos pelos arcos do corte). Um dos objetivos desta nova metodologia de solução do problema MOSP foi focar em problemas que geram grafos menos densos, pois, é nesse caso que os demais métodos da literatura apresentam dificuldade. A heurística foi limitada ao particionamento do problema em apenas dois subproblemas. Ideias futuras consistem em aplicar a heurística de forma iterativa nos subproblemas obtidos em cada iteração. Esta estratégia permitiria resolver problemas grandes, já que estes poderiam ser reduzidos até um tamanho aceitável para o método de solução aplicado.

# Referências Bibliográficas

---

---

- AGARWALA, R.; APPLGATE, D. L.; MAGLOTT, D.; SCHULER, G. D.; SCHÄFFER, A. A. A fast and scalable radiation hybrid map construction and integration strategy. *Genome Research*, v. 10, p. 350–364, 2000.
- APPLGATE, D. L.; BIXBY, R. E.; CHVÁTAL, V.; COOK, W. J. *The traveling salesman problem: a computational study*. Princeton University Press, 593 p., 2006.
- APPLGATE, D. L.; BIXBY, R. E.; CHVÁTAL, V.; COOK, W. J. Concorde TSP solver. <http://www.tsp.gatech.edu/concorde.html>, 2009.
- ASHIKAGA, F. M. *Um método frugal para o problema de minimização de pilhas abertas*. Dissertação de Mestrado, Instituto Tecnológico da Aeronáutica, São José dos Campos/SP, Brasil, 2001.
- ASHIKAGA, F. M.; SOMA, N. Y. A heuristic for the minimization open stacks problem. *Pesquisa Operacional*, v. 29, n. 2, p. 439–450, 2009.
- DE LA BANDA, M. G.; STUCKEY, P. J. Dynamic programming to minimize the maximum number of open stacks. In: *Constraint modelling challenge, In The Fifth Workshop on Modelling and Solving Problems with Constraints, IJCAI 2005*, Edinburg: <http://www.cs.st-andrews.ac.uk/~ipg/challenge/ModelChallenge05.pdf>, 2005, p. 24–31.
- DE LA BANDA, M. G.; STUCKEY, P. J. Dynamic programming to minimize the maximum number of open stacks. *Inform Journal on Computing*, v. 19, n. 4, p. 607–617, 2007.
- BECCENERI, J. C. *O problema de seqüenciamento de padrões para a minimização do número máximo de pilhas abertas em ambientes de cortes industriais*. Tese de Doutorado, Instituto Tecnológico de Aeronáutica, São José dos Campos/SP, Brasil, 1999.
- BECCENERI, J. C.; YANASSE, H. H.; SOMA, N. Y. A method for solving the minimization of the maximum number of open stacks problem within a cutting process. *Computers and Operations Research*, v. 31, p. 2315–2332, 2004.

- DE CARVALHO, M. A. M.; SOMA, N. Y. Métodos simplificados para o problema de minimização de pilhas abertas. *Gestão e Produção*, v. 18, n. 2, p. 299–310, 2011.
- CHU, G.; STUCKEY, P. J. Minimizing the maximum number of open stacks by customer search. In: *International Conference on Principles and Practice of Constraint Programming*, Berlin, Heidelberg: Springer-Verlag, 2009, p. 242–257.
- DOLAN, E. D.; MORÉ, J. J. Benchmarking optimization software with performance profiles. *Mathematical programming*, v. 91, p. 201–213, 2002.
- FAGGIOLI, E.; BENTIVOGLIO, C. A. Heuristic and exact methods for the cutting sequencing problem. *European Journal of Operational Research*, v. 110, p. 564–575, 1998.
- FINK, A.; VOSS, S. Applications of modern heuristic search methods to pattern sequencing problems. *Computers and Operations Research*, v. 26, p. 17–34, 1999.
- FINK, C.; COSTA, A. M.; YANASSE, H. H. Análise do desempenho de variações de uma formulação linear para o problema de minimização do número máximo de pilhas abertas. In: *Anais do XLI Simpósio Brasileiro de Pesquisa Operacional*, 2009.
- FINK, C.; MUNARI, P. A.; COSTA, A. M.; YANASSE, H. H. Uma estratégia de solução para o problema de minimização do número máximo de pilhas abertas utilizando o problema do caixeiro viajante. *Notas do ICMC, Série Computação*, , n. 95, 2010.
- FOERSTER, H.; WASCHER, G. Simulated annealing for order spread minimization in sequencing cutting problem. *European Journal of Operational Research*, v. 110, n. 2, p. 272–281, 1998.
- GAREY, M. R.; JOHNSON, D. S. *Computers and intractability; a guide to the theory of np-completeness*. 1 ed. New York, USA: W. H. Freeman & Co., 1979.
- HAHSLER, M.; HORNIK, K. TSP - infrastructure for the traveling salesperson problem. *Journal of Statistical Software*, v. 23, n. 2, p. 1–21, 2007.
- HERTZ, A.; LAPORTE, G.; MITTAZ, M.; STECKE, K. E. Heuristics for minimizing tool switches when scheduling part types on a flexible machine. *IIE Transactions*, v. 30, n. 8, p. 689–694, 1998.
- LAPORTE, G. The traveling salesman problem: an overview of exact and approximate algorithms. *European Journal of Operational Research*, v. 59, p. 231–247, 1992.
- LAPORTE, G.; SALAZAR-GONZÁLEZ, J. J.; SEMET, F. Exact algorithms for the job sequencing and tool switching problem. *IIE Transactions*, v. 36, p. 37–45, 2004.
- LIN, S.; KERNIGHAN, B. W. An effective heuristic for the traveling salesman problem. *Operations Research*, v. 21, p. 498–516, 1973.



- LINHARES, A.; YANASSE, H. H. Connections between cutting-pattern sequencing, VLSI design, and flexible machines. *Computers and Operations Research*, v. 29, n. 12, p. 1759–1772, 2002.
- LINS, S. Traversing trees and scheduling tasks for duplex corrugator machines. *Pesquisa Operacional*, v. 9, p. 40–54, 1989.
- MADSEN, O. B. G. An application of travelling-salesman routines to solve pattern-allocation problems in the glass industry. *Journal of the Operational Research Society*, v. 39, n. 3, p. 249–256, 1988.
- MILLER, A. Improved lower bounds for solving the minimal open stacks problem. In: *Constraint modelling challenge, In The Fifth Workshop on Modelling and Solving Problems with Constraints, IJCAI 2005*, Edinburg: <http://www.cs.st-andrews.ac.uk/~ipg/challenge/ModelChallenge05.pdf>, 2005, p. 36–43.
- MUNARI, P. A. *Comparação de softwares científicos utilizando perfis de desempenho: automatização dos cálculos pela planilha perfis.xls*. Relatório Técnico 344, Instituto de Ciências Matemáticas e de Computação - Universidade de São Paulo, [http://www.icmc.usp.br/~biblio/BIBLIOTECA/rel\\_tec/RT\\_344.PDF](http://www.icmc.usp.br/~biblio/BIBLIOTECA/rel_tec/RT_344.PDF), 2009.
- ONCAN, T.; ALTINEL, K.; LAPORTE, G. A comparative analysis of several asymmetric traveling salesman problem formulations. *Computers and Operations Research*, v. 36, p. 637–645, 2009.
- PESANT, G. Trying hard to solve the simultaneously open stacks problem with cp. In: *Constraint modelling challenge, In The Fifth Workshop on Modelling and Solving Problems with Constraints, IJCAI 2005*, Edinburg: <http://www.cs.st-andrews.ac.uk/~ipg/challenge/ModelChallenge05.pdf>, 2005, p. 51–56.
- PINTO, M. J. *Algumas contribuições à resolução do problema de corte integrado ao problema de sequenciamento dos padrões*. Tese de Doutorado, Instituto Nacional de Pesquisas Espaciais, São José dos Campos/SP, Brasil, 2004.
- PÓSA, L. Hamiltonian circuits in random graphs. *Discrete Mathematics*, v. 14, p. 359–364, 1976.
- RÍOS-MERCADO, R. Z.; BARD, J. F. An enhanced tsp-based heuristic for makespan minimization in a flow shop with setup times. *Journal of Heuristics*, v. 5, n. 1, p. 53–70, 1999.
- ROSENKRANTZ, D. J.; STEARNS, R. E.; LEWIS, P. M. An analysis of several heuristics for the traveling salesman problem. *SIAM Journal on Computing*, v. 6, n. 3, p. 563–581, 1977.
- SHAW, P.; LABORIE, P. A constraint programming approach to the min-stack problem. In: *Constraint modelling challenge, In The Fifth Workshop on Modelling and Solving Problems with Constraints, IJCAI 2005*, Edinburg: <http://www.cs.st-andrews.ac.uk/~ipg/challenge/ModelChallenge05.pdf>, 2005, p. 62–67.

- DA SILVA LOPES, I. C. *Modelos para sequenciação de padrões em problemas de corte de stock*. Tese de Doutorado, Universidade do Minho, Portugal, 2011.
- SIMONIS, H. Constraint modelling challenge. In: *In The Fifth Workshop on Modelling and Solving Problems with Constraints, IJCAI 2005*, Edinburg: <http://www.cs.st-andrews.ac.uk/~ipg/challenge/ModelChallenge05.pdf>, 2005, p. 68–77.
- SMITH, B. M.; GENT, I. P. Constraint modelling challenge. In: *In The Fifth Workshop on Modelling and Solving Problems with Constraints, IJCAI 2005*, Edinburg: <http://www.cs.st-andrews.ac.uk/~ipg/challenge/ModelChallenge05.pdf>, 2005, p. 1–13.
- TANG, C.; DENARDO, E. Models arising from a flexible manufacturing machine, part i: minimization of the number of tool switches. *Operatios Research*, v. 36, p. 767–777, 1988.
- YANASSE, H.H., B. J. S. N. Ordenamento parcial para reduzir o espaço de busca de uma solução ótima para um problema de sequenciamento de padrões. In: *Anais do XXXIV Simpósio Brasileiro de Pesquisa Operacional*, 2002.
- YANASSE, H.; BECCENERI, J.; SOMA, N. Bounds for a problem of sequencing patterns. *Pesquisa Operacional*, v. 19, n. 2, p. 249–277, 1999.
- YANASSE, H.; LAMOSA, M. An integrated cutting stock and sequencing problem. *European Journal of Operational Research*, v. 183, p. 1352–1370, 2007.
- YANASSE, H. H. Minimization of open orders - polynomial algorithms for some special cases. *Pesquisa Operacional*, v. 16, p. 1–26, 1996.
- YANASSE, H. H. On a pattern sequencing problem to minimize the maximum number of open stacks. *European Journal of Operational Research*, v. 100, p. 454–463, 1997a.
- YANASSE, H. H. A transformation for solving a pattern sequencing problem in the wood cut industry. *Pesquisa Operacional*, v. 17, n. 1, p. 57–70, 1997b.
- YANASSE, H. H.; BECCENERI, J. C.; SOMA, N. Y. Um algoritmo exato com ordenamento parcial para solução de um problema de programação da produção: experimentos computacionais. *Gestão & Produção*, v. 14, n. 2, p. 353–361, 2007.
- YANASSE, H. H.; PINTO, M. J. Uma nova formulação para um problema de sequenciamento de padrões em ambientes de corte. In: *Anais do XXXV Simpósio Brasileiro de Pesquisa Operacional*, 2003, p. 1516–1524.
- YANASSE, H. H.; SENNE, E. L. F. The minimization of open stacks problem: A review of some properties and their use in pre-processing operations. *European Journal of Operational Research*, v. 203, p. 559567, 2010.

- YUEN, B. J. Heuristics for sequencing cutting patterns. *European Journal of Operational Research*, v. 55, p. 183–190, 1991.
- YUEN, B. J. Improved heuristics for sequencing cutting patterns. *European Journal of Operational Research*, v. 87, p. 57–64, 1995.
- YUEN, B. J.; RICHARDSON, K. V. Establishing the optimality of sequencing heuristics for cutting stock problems. *European Journal of Operational Research*, v. 84, p. 590–598, 1995.