

Problema de Minimização de Pilhas Abertas: Uma Abordagem Elementar

Marco Antonio Moreira de Carvalho¹, Nei Yoshihiro Soma²

¹Departamento de Ciência da Computação
Grupo de Otimização e Algoritmos (GOAL)
Universidade Federal de Ouro Preto (UFOP)
Campus Universitário Morro do Cruzeiro – 35.400-000
Ouro Preto – MG – Brasil

²Departamento de Ciência da Computação
Instituto Tecnológico de Aeronáutica (ITA)
Praça Marechal Eduardo Gomes, 50, Vila das Acácias – 12.228-900
São José dos Campos – SP – Brasil

mamc@iceb.ufop.br, soma@ita.br

Resumo. *Apresenta-se neste trabalho uma nova heurística para o Problema de Minimização de Pilhas Abertas, um problema de escalonamento relacionado aos problemas de corte e empacotamento. A heurística utiliza representação em grafos e baseia-se no bem conhecido algoritmo de busca em largura. O método é comparado com a heurística de melhor desempenho disponível na literatura e também com os resultados ótimos para um conjunto de aproximadamente seis mil instâncias benchmark. Nos experimentos computacionais realizados, o método proposto obteve um gap de 0,9% em relação ao ótimo e um comportamento médio melhor em relação ao método de melhor desempenho, além de tempo de execução sempre abaixo de um segundo.*

PALAVRAS CHAVE. *Sequenciamento de Padrões. Corte e Empacotamento. Escalonamento.*

Área Principal. *PO na Administração & Gestão da Produção.*

Abstract. *It is presented in this paper a new heuristic for the Minimization of Open Stacks Problem, a scheduling problem related to cutting and packing problems. The proposed heuristic uses graph representation and it is based on the well-known breadth-first search algorithm. The method is compared with the best performance heuristic available in the literature and also with the optimal results for a set of approximately six thousand benchmark instances. In the computational experiments performed, the proposed method achieved a gap of 0.9% from the optimal, a better average performance when compared to the best heuristic of the literature, and running times always below one second.*

KEYWORDS. *Pattern Sequencing. Cutting and Packing. Scheduling.*

Main Area. *OR in Administration & Production Management.*

1 Introdução

O Problema de Minimização de Pilhas Abertas (ou MOSP, de *Minimization of Open Stacks Problem*) remonta à um ambiente de produção em que produtos com demandas específicas são produzidos em lotes por uma única máquina. Quando um pedido de compra é realizado por um cliente, os produtos correspondentes são mantidos em uma pilha ao redor da máquina que os produziu; cada pilha armazena somente produtos relativos a um mesmo pedido de um cliente. Quando o primeiro produto de um pedido tiver de ser produzido, a respectiva pilha é considerada “aberta” e assim permanece até que o último produto do mesmo pedido seja produzido, quando então a pilha é considerada “fechada” e pode ser removida para outro local a fim de dar continuidade ao processo de produção.

Neste problema, não é suficiente apenas gerar uma agenda para o processamento de matéria-prima e conseqüente fabricação de produtos de forma a atender as demandas; também é necessário atingir a melhor utilização do espaço físico disponível, agilizando a linha de produção. Admite-se a existência de uma limitação física para o armazenamento de pilhas ao redor da máquina de produção, de tal forma que não haja espaço suficiente para que uma pilha associada a cada um dos pedidos seja aberta simultaneamente. Sendo assim, pode ser necessária a remoção temporária de pilhas ainda não fechadas para que novas pilhas sejam abertas e os produtos continuem sendo produzidos. Posteriormente, quando os produtos referentes às pilhas removidas forem produzidos, tais pilhas podem ser novamente trazidas de volta para o ambiente de produção e outras pilhas devem ser removidas para a obtenção do espaço necessário ao redor da máquina de processamento.

A remoção constante de pilhas abertas afeta o custo associado à fabricação dos produtos, exigindo a alocação de mão-de-obra, maquinário e tempo adicionais para o transporte e a administração das pilhas removidas temporariamente. Além disso, alguns produtos são frágeis, como na indústria de vidro, em que o manuseio excessivo pode gerar outros prejuízos como aqueles advindos da quebra de produtos. É desejável, então, que as pilhas de produtos sejam removidas uma única vez, depois de fechadas, para diminuir os custos relacionados à produção dos produtos.

A Figura 1 apresenta um exemplo de instância MOSP, em que são fornecidos dados sobre a composição de pedidos por produtos. Note que informações sobre a quantidade de produtos de um mesmo tipo na composição de um dos pedidos não são relevantes para este problema específico, conforme detalhado a seguir. Os produtos são enumerados de p_1 a p_{10} e os pedidos por produtos, enumeradas de c_1 a c_{10} . O símbolo \otimes indica que o pedido c_i contém o produto p_j . Espaços em branco indicam o contrário.

A Figura 2 apresenta o conjunto de produtos do exemplo anterior e dois possíveis sequenciamentos para a fabricação dos mesmos. O símbolo \otimes indica que um produto do cliente c_i foi produzido, enquanto o símbolo ‘—’ indica que a pilha permanece aberta, embora não esteja sendo utilizada. No primeiro sequenciamento, a lista de produtos \mathcal{L} é definida como $\mathcal{L} = \{p_5, p_3, p_1, p_7, p_9, p_4, p_2, p_6, p_8, p_{10}\}$. Neste sequenciamento (à esquerda na referida figura), o número máximo de pilhas abertas é 3; após o processamento do produto p_6 , o cliente c_3 terá seu pedido completo (linha 3 e colunas 5-8), mas o produto p_6 também abre a pilha associada ao cliente c_{10} (linha 10 e coluna 8). O segundo sequenciamento apresenta uma solução com estrutura diferente, mas com mesmo número máximo de pilhas abertas.

A seguir, os trabalhos mais relevantes no tratamento do problema de minimização

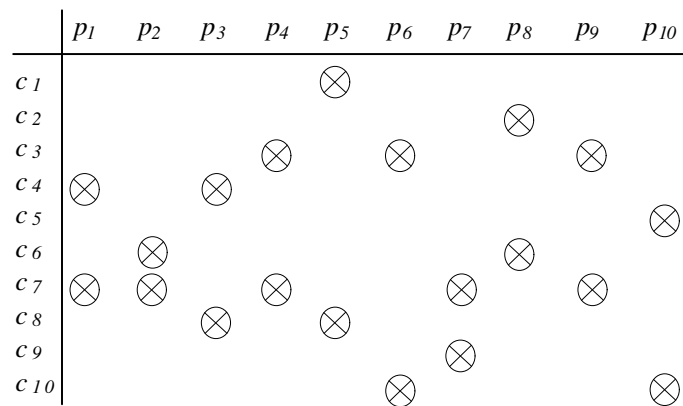


Figura 1. Um exemplo de instância do problema de minimização de pilhas abertas.

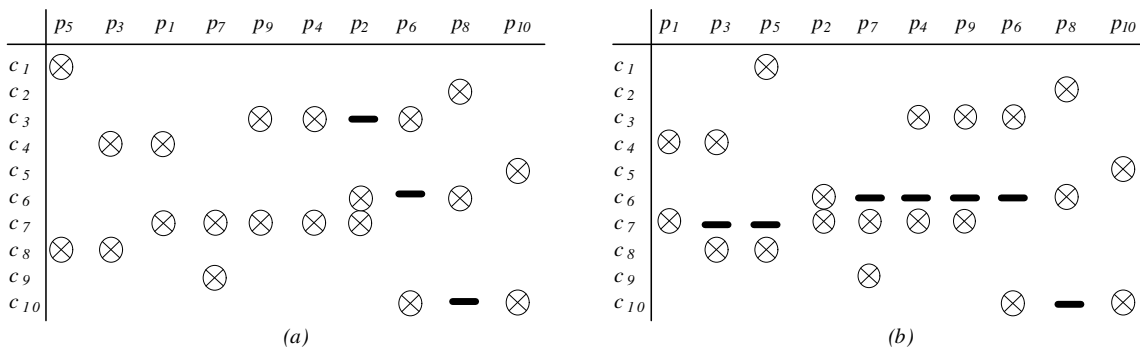


Figura 2. Duas possíveis seqüências para produção de produtos.

de pilhas abertas são sintetizados de acordo com o tipo de abordagem.

1.1 Trabalhos Teóricos

O trabalho de Linhares (2002) apresenta diversos resultados teóricos sobre o MOSP. A complexidade do MOSP é provada NP-Difícil e é estabelecida a equivalência entre o MOSP e problemas relacionados ao projeto de circuitos VLSI, bem como é analisado o relacionamento entre o MOSP e outros problemas de sequenciamento.

Em Yanasse (2010), sete métodos de pré-processamento para reduzir o tamanho das instâncias MOSP tratadas e o esforço necessário para sua solução são discutidos.

1.2 Métodos Exatos

Yanasse (1997) adaptou para o MOSP um modelo originalmente desenvolvido para o Problema de Minimização de Troca de Ferramentas (*Minimization of Tool Switches Problem – MTSP*). Uma estratégia *branch-and-bound* é apresentada no mesmo trabalho.

Em Yanasse (2002), Becceneri (2004) e Yanasse (2007), são aplicados métodos de ordenamento parcial com diferentes critérios de equivalência. Nestes trabalhos também foram derivados novos limitantes e boas heurísticas para obtenção destes.

Em Yanasse (2003) é apresentado um novo modelo que utiliza grafos especiais e cujo objetivo é determinar a ordem em que as pilhas serão fechadas minimizando o número máximo de pilhas abertas, ao contrário dos outros trabalhos cujo foco é a ordem em que

produtos são processados. Fink (2009) apresenta quatro variações para este modelo, para tentar acelerar sua resolução.

Posteriormente, outra adaptação de um modelo para o MTSP é proposta por Pinto (2004). Desta vez foi utilizado como base um modelo que usa restrições do Problema do Caixeiro Viajante. Este modelo foi comparado ao de Yanasse (1997), sendo que cada modelo obteve o melhor desempenho para cada metade do conjunto de instâncias.

Por sua complexidade, equivalência a outros problemas e aplicação em manufatura, no ano de 2005 o MOSP foi escolhido como tema do *First Constraint Modelling Challenge* (Smith, 2005), cujo conjunto de instâncias é considerado como o único *benchmark* para o problema. O modelo vencedor foi proposto por de la Banda (2007), que consiste de dois métodos de pré-processamento dos dados de entrada e uma implementação do geral e bem conhecido algoritmo A*.

Mais recentemente, Lopes (2011) desenvolveu um modelo de programação inteira para o MOSP também aplicado a outros problemas, entre eles o MTSP. O MOSP é modelado como um grafo de intervalos, em que cada pilha aberta é associada a um intervalo. O objetivo é ordenar tais intervalos de forma que a sobreposição entre eles seja mínima, e desta forma, fazendo com que menos pilhas sejam abertas simultaneamente.

1.3 Métodos Heurísticos

Os trabalhos seminais de Yuen (1991, 1995) contêm as heurísticas mais difundidas para tratamento do MOSP – dentre os trabalhos da restrita literatura sobre o problema, a citação a estas duas referências é unânime. Nos dois trabalhos destacados são apresentadas, seis heurísticas rápidas e de simples implementação. Entre estas se encontra a denominada *Yuen3*, que foi considerada como a de melhor desempenho por muitos anos na literatura.

A *Heurística de Nó de Custo Mínimo*, apresentada em Becceneri (2004) é considerada como a de melhor desempenho atualmente (conforme Yanasse (2010)). A heurística se baseia em grafos MOSP (em que os vértices representam pedidos de compra e são conectados caso compartilhem algum tipo de produto, vide Seção 2) e consiste em sequenciar as suas arestas, priorizando os vértices de menor grau. Esta heurística possui complexidade $O(c^3)$, em que c denota o número de pedidos de compra de produtos.

Novamente a abordagem por teoria de grafos é utilizada na heurística *Ashikaga Soma* (ou *Heurística de Extensão-Rotação*) (Ashikaga, 2009). Os vértices do grafo MOSP são sequenciados como um caminho Hamiltoniano, cuja parte inicial é formada por um clique – eventualmente maximal. A *Heurística Ashikaga Soma* possui complexidade $O(c^2)$ no pior caso, em que c denota o número de pedidos de compra de produtos.

Simplificações para os métodos introduzidos por de la Banda (2007) e Ashikaga (2009) são propostas em Carvalho (2011). Nos experimentos reportados, a simplificação proposta para a heurística *Ashikaga Soma* obteve ligeiro melhor desempenho que o método original, e a heurística gerada a partir do método exato proposto em de la Banda (2007) superou as duas primeiras heurísticas.

2 Uma Abordagem Elementar

Para a representação do problema, foram utilizados grafos MOSP (conforme descrito em Yanasse (1997)), em que os vértices representam os pedidos de compra, ligados por arestas

se e somente se os respectivos pedidos compartilham um mesmo tipo de produto entre si. Arestas múltiplas e laços não são considerados. Desta forma, todos os pedidos que envolvem um mesmo tipo de produto são ligados entre si, induzindo cliques em tais grafos, de maneira que os grafos MOSP são formados pela união dos cliques de uma instância. Novos e maiores cliques podem ainda ser induzidos por tal união, indicando regiões de concentração no grafo e fornecendo informações importantes para solução do problema.

A Figura 3 apresenta em seu lado esquerdo o grafo MOSP referente ao exemplo dado na Figura 1. No lado direito da referida figura são ressaltados alguns dos cliques induzidos pelo compartilhamento de produtos. Por exemplo, o produto p_8 é compartilhado pelos pedidos 2 e 6.

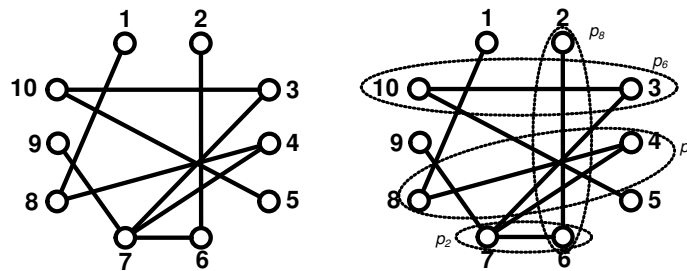


Figura 3. Grafo MOSP correspondente ao exemplo da Figura 1.2.

Transferindo o problema para o domínio da teoria dos grafos, o problema consiste em explorar as características subjacentes como os possíveis percursos em grafos, os já citados cliques maximais e os diferentes ordenamentos de vértices. Uma vez realizadas as operações escolhidas, é necessário obter a sequência de processamento dos produtos, transferindo o problema novamente para o domínio das matrizes.

O ponto de partida para o método desenvolvido é derivado do uso da estrutura de banda em matrizes esparsas. Tal estrutura de banda é obtida como resultado do Problema de Minimização de Banda em Matrizes (*Matrix Bandwidth Minimization – MBM*), que consiste em minimizar a distância entre elementos não nulos e a diagonal principal da matriz. Em outras palavras, busca-se diagonalizar a matriz ao máximo, através da permutação de linhas e colunas da mesma.

Um dos métodos heurísticos mais antigos e mais bem sucedidos para o tratamento do problema é a heurística (*Reverse*) *Cuthill-Mckee* (Cuthill, 1969). Desenvolvido para matrizes simétricas, o método utiliza a representação por grafos em sua abordagem. Uma matriz simétrica pode ser interpretada como uma matriz de incidências de um grafo não direcionado, e com base nela o grafo correspondente é construído. Cada vértice representa a linha e coluna de mesmo índice, e as adjacências obedecem ao estabelecido pela matriz.

A heurística *Cuthill-Mckee* consiste na realização de uma Busca em Largura (*Breadth-First Search – BFS*) utilizando-se como critério o vértice de menor grau, ou seja, a busca sempre é iniciada pelo vértice de menor grau, e os vizinhos dos vértices são analisados em ordem não decrescente de grau. Empates são resolvidos a favor do vértice de menor índice. A sequência de vértices obtida determina a sequência das linhas e colunas da matriz na solução.

A busca em largura é um dos algoritmos mais simples para se percorrer um grafo e é o arquétipo de muitos algoritmos de grafos importantes (Cormen, 2001). Dado um

grafo, a partir de um vértice inicial, a busca em largura explora toda a sua vizinhança imediata. Uma vez terminado esse processo, um dos vértices desta vizinhança explorada é selecionado como novo ponto de partida para uma nova execução deste processo, que é repetido até que todo o grafo tenha sido explorado, sem que vértices sejam visitados mais que uma vez. Em casos de diferentes componentes, uma vez que um componente seja totalmente explorado, um dos vértices de outro componente é utilizado como ponto de partida para uma nova busca. A busca em largura retorna uma lista de vértices indicando a ordem de exploração do grafo, o que depende fortemente da escolha dos vértices utilizados como pontos de partida a cada instante.

Notavelmente, a BFS nunca fora aplicada ao MOSP, muito embora tenha sido aplicado ao Problema de Minimização de Espalhamento de Ordens (Madsen , 1979) – um problema correlato ao MOSP e também um caso especial do MBM (Yanasse , 1997). De fato, as matrizes que representam instâncias MOSP ou mesmo grafos MOSP não são necessariamente simétricas ou fortemente esparsas. Ainda, a estrutura de banda não é um requisito para geração de soluções ótimas para o MOSP (Linhares , 2002).

No entanto, como será mostrado, a estrutura resultante para uma grande quantidade de instâncias da literatura, quando utilizados grafos MOSP como entrada produz boas soluções, em geral, subótimas. Além disto, a aplicação direta da BFS em grafos MOSP tem baixa complexidade computacional, sendo possível obter a detecção de componentes no grafo, circunstância sob a qual o problema pode ser decomposto em subproblemas distintos e assim tratado implicitamente.

O método aqui sugerido está baseado na aplicação do método *Cuthill-Mckee* a grafos MOSP, em outras palavras, aplicar a busca em largura em grafos MOSP analisando os vértices em ordem não-crescente de grau, resolvendo empates a favor do vértice de menor índice. A Figura 4 apresenta uma busca em largura aplicada ao grafo MOSP da Figura 3 utilizando a notação proposta por Cormen (2001): três cores são utilizadas para identificar os vértices – branco para vértices ainda não visitados; cinza para vértices já visitados, porém, com vértices adjacentes ainda não visitados e preto para aqueles vértices cuja vizinhança tenha sido completamente visitada. A busca se inicia com todos vértices brancos e termina quando todos os vértices forem pretos. A fila que indica a ordem de exploração dos vértices é denominada Q .

Um dos atrativos da busca em largura é que por sequenciar toda a vizinhança de um vértice a cada vez, os vértices que compõem cliques e regiões densas do grafo MOSP tendem a aparecer consecutivamente na solução, o que possui importância reconhecida na solução do MOSP. Efetivamente, isto equivale a sequenciar contiguamente os pedidos com produtos em comum.

A BFS retorna somente uma ordenação dos pedidos de compra de produtos (representados pelos vértices do grafo MOSP). Para obter a ordem de fabricação dos produtos a partir da ordenação dos pedidos, utiliza-se o princípio de pilha (último a entrar, primeiro a sair) associado aos pedidos. Cada pedido é analisado, e todos os produtos que o compõem são adicionados à solução final, sem repetição.

Considerando a aplicação da busca em largura mostrada na Figura 4, a fila de vértices retornada é $Q = \{1, 8, 4, 7, 9, 3, 6, 10, 2, 5\}$. O último vértice inserido em Q é 5, composto pelo produto p_{10} que é inserido na solução final. O próximo pedido analisado

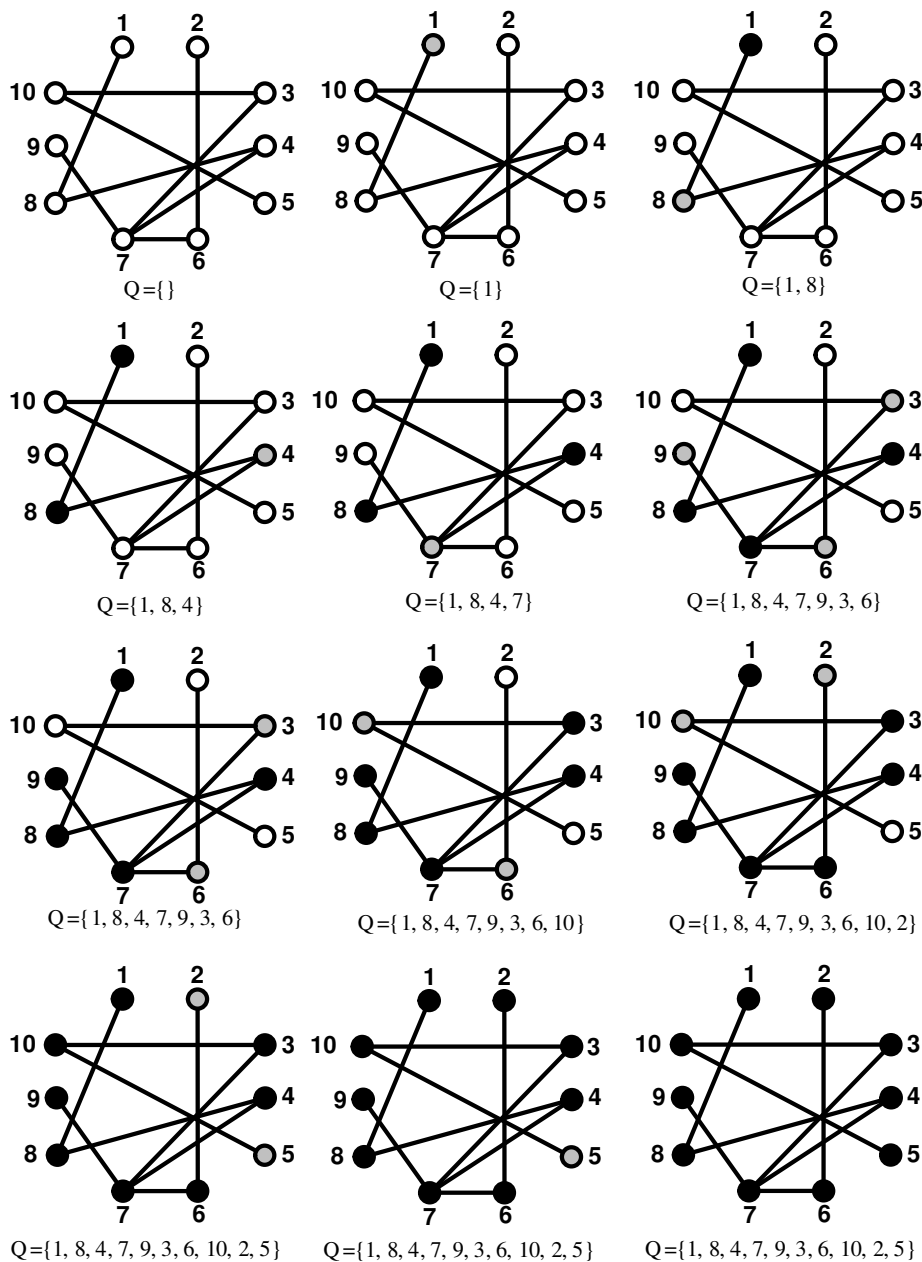


Figura 4. Busca em largura aplicada ao grafo MOSP de exemplo.

é o número 2, composto pelo produto p_8 que também é inserido na solução. Este processo é repetido para todos os pedidos, e a lista final de produtos é dada por $\mathcal{L}=\{p_5, p_3, p_1, p_7, p_9, p_4, p_2, p_6, p_8, p_{10}\}$, cujo número máximo de pilhas abertas é 3 (vide Figura 2(a)).

Um aspecto importante desta estratégia é que ela gerará pequenos erros se o grafo MOSP for composto por um clique dominante com alguns poucos vértices em sua vizinhança, ou ainda, por um conjunto de cliques fracamente conectados. Nestes casos, a busca em largura explorará vértices de diferentes cliques alternadamente, o que pode resultar em um maior número de pilhas abertas na solução final devido ao sequenciamento de pedidos não relacionados por produtos em comum. Para contornar estas situações, duas regras são utilizadas em conjunto com a busca em largura:

1. Se uma pilha se mantém aberta durante um longo período sem receber nenhum produto, então antecipa-se a ordem de processamento dos produtos correspondentes de forma que a pilha possa ser fechada antecipadamente; e
2. Se um produto está relacionado à abertura de uma nova pilha, então posterga-se a ordem de processamento deste produto para que a nova pilha seja aberta mais tardiamente, possibilitando que outras pilhas sejam fechadas antes.

Estas duas regras são aplicada à lista \mathcal{L} e caso haja melhoria na solução, as alterações necessárias são realizadas.

A aplicação destas regras requer um exame da composição de cada um dos pedidos por produtos. Com a incorporação destas regras, o método proposto, batizado de HBF_{2r} , possui complexidade limitada por $O(p^2c)$, em que c denota o número de pedidos por produtos e p denota o número de produtos. De acordo com resultados computacionais reportados na próxima Seção, o pior caso não é frequente na execução do método HBF_{2r} , que tem como um dos pontos fortes o seu tempo de execução.

3 Experimentos Computacionais

Todos os experimentos foram realizados em um computador Pentium IV duo core 3.2 GHz de frequência com 1 GB de memória RAM sob o sistema operacional Fedora Linux 11.

HBF_{2r} foi codificada em ANSI C, compilada com gcc 4.4.1 sem opções de otimização. Um total de 5806 instâncias propostas para o *First Constraint Modelling Challenge* (Smith, 2005) foram utilizadas, sendo 5785 instâncias agrupadas por características comuns e 21 instâncias individuais. Cada conjunto de instâncias segue o formato *nome_c_p*, em que c denota o número de pedidos de compra e p denota o número de produtos.

As comparações são realizadas entre o métodos proposto, HBF_{2r} , os valores ótimos para o conjunto de instâncias, determinados em Chu (2009), e também a *Heurística de Nó de Custo Mínimo*, proposta em Becceneri (2004), referida como *HNCM*. Este último método foi codificado e compilado usando o mesmo ambiente computacional utilizado por HBF_{2r} , seguindo fielmente a descrição da implementação original.

A Tabela 1 apresenta a comparação das soluções considerando as coleções de instâncias. Na segunda coluna são apresentadas as soluções ótimas, e para cada método há uma coluna associada à quantidade média de pilhas abertas e outra com o tempo médio de execução, expresso em milissegundos. A última coluna da tabela apresenta o número de instâncias de cada coleção.

Em 82% do total dos casos, houve empate entre as soluções obtidas por HBF_{2r} e *HNCM*. Em 11% dos casos, HBF_{2r} obteve melhores soluções, e piores soluções nos 7% restantes. O *gap* entre as soluções é de apenas 0,4%, a favor de HBF_{2r} .

Os tempos de execução médios e máximos são 0,36 ms e 155,14 ms para HBF_{2r} e 0,22 ms e 20,00 ms para *HNCM*. A despeito da complexidade dos métodos, observa-se que os tempos de execução são extremamente atraentes.

HBF_{2r} encontrou os valores ótimos para 88% do total de instâncias. A maior diferença entre as soluções foi de 4 pilhas, que ocorreu em apenas um caso. Em 89% dos casos em que HBF_{2r} não atingiu os valores ótimos, a diferença foi de apenas uma pilha. O *gap* em relação à solução ótima é de 0,9%.

Tabela 1. Comparação entre soluções ótimas, *HNCM* e HBF_{2r} considerando as coleções de instâncias do *First Constraint Modelling Challenge* (Smith , 2005).

Conjunto	Solução Ótima	<i>HNCM</i> Solução	Tempo de Execução (ms)	HBF_{2r} Solução	Tempo de Execução (ms)	Instâncias
problem_10_10	8,03	8,05	0,02	8,06	0,05	550
problem_10_20	8,92	8,95	0,04	8,95	0,06	550
problem_15_15	12,88	12,95	0,04	12,95	0,13	550
problem_15_30	14,02	14,05	0,18	14,06	0,18	220
problem_20_10	15,88	16,11	0,13	15,96	0,18	550
problem_20_20	17,97	18,03	0,18	18,08	0,27	220
problem_30_10	23,95	24,38	0,24	24,05	0,38	550
problem_30_15	25,97	26,21	0,18	26,11	0,44	220
problem_30_30	28,32	28,40	0,36	28,57	0,68	110
problem_40_20	36,38	36,61	1,00	36,68	1,00	110
Shaw_20_20	13,68	14,04	0,00	14,00	0,30	25
wbo_10_10	5,93	6,08	0,25	5,95	0,06	40
wbo_10_20	7,35	7,50	0,00	7,45	0,09	40
wbo_10_30	8,20	8,25	0,00	8,33	0,12	40
wbo_15_15	9,35	9,62	0,17	9,47	0,16	60
wbo_15_30	11,58	11,75	0,00	11,88	0,19	60
wbo_20_10	12,90	13,07	0,14	13,09	0,19	70
wbo_20_20	13,69	14,07	0,33	14,01	0,30	90
wbo_30_10	20,05	20,50	0,40	20,24	0,43	100
wbo_30_15	20,96	21,40	0,25	21,37	0,57	120
wbo_30_30	22,56	23,04	0,29	23,04	0,90	140
wbop_10_10	6,75	6,83	0,00	6,75	0,06	40
wbop_10_20	8,08	8,15	0,00	8,10	0,09	40
wbop_10_30	8,62	8,69	0,00	8,74	0,12	40
wbop_15_15	10,37	10,43	0,17	10,45	0,16	60
wbop_15_30	12,15	12,30	0,17	12,57	0,23	60
wbop_20_10	14,28	14,65	0,25	14,35	0,21	40
wbop_20_20	14,87	15,08	0,22	15,03	0,29	90
wbop_30_10	22,48	22,73	0,00	22,58	0,42	40
wbop_30_15	22,38	22,73	0,50	22,65	0,61	60
wbop_30_30	23,84	24,13	0,36	24,31	0,93	140
wbp_10_10	7,28	7,43	0,00	7,30	0,05	40
wbp_10_20	8,71	8,77	0,00	8,76	0,07	70
wbp_10_30	9,31	9,34	0,10	9,34	0,09	100
wbp_15_15	11,05	11,28	0,00	11,17	0,15	60
wbp_15_30	13,09	13,18	0,17	13,18	0,19	120
wbp_20_10	15,13	15,45	0,25	15,23	0,19	40
wbp_20_20	15,41	15,66	0,00	15,64	0,30	90
wbp_30_10	23,18	24,33	0,25	23,40	0,43	40
wbp_30_15	22,98	23,68	0,33	23,40	0,57	60
wbp_30_30	24,46	24,99	0,29	24,90	0,91	140

Os resultados para instâncias individuais são apresentados na Tabela 2.

É importante ressaltar que, apesar de se tratar de uma heurística, HBF_{2r} apresenta um desempenho muito bom quando comparada com resultados ótimos. Ainda, quando comparado com *HNCM*, considerado como método heurístico de melhor desempenho (conforme Yanasse (2010)), HBF_{2r} apresenta um comportamento mais regular e mais próximo do ótimo, embora a *gap* entre as duas heurísticas seja pequeno.

Entre as 21 instâncias individuais, HBF_{2r} obteve as soluções ótimas para 12, ao

Tabela 2. Comparação entre soluções ótimas, $HNCM$ e HBF_{2r} , considerando as instâncias individuais do *First Constraint Modelling Challenge* (Smith, 2005).

Instância	Solução		Tempo de Execução (ms)	Tempo de Execução	
	Ótima	$HNCM$ Solução		HBF_{2r} Solução	(ms)
GP1_50_50	45	45	0,00	45	0,98
GP2_50_50	40	40	0,00	42	1,26
GP3_50_50	40	40	0,00	41	1,13
GP4_50_50	30	30	0,00	31	0,67
GP5_100_100	95	96	20,00	96	5,87
GP6_100_100	75	75	10,00	75	4,51
GP7_100_100	75	75	10,00	76	4,35
GP8_100_100	60	60	20,00	62	7,61
Miller_20_40	13	13	0,00	13	0,39
NWRS1_10_20	3	3	0,00	3	0,04
NWRS2_10_10	4	4	0,00	4	0,07
NWRS3_15_25	7	7	0,00	7	0,07
NWRS4_15_25	7	7	0,00	7	0,14
NWRS5_20_30	12	12	0,00	12	0,19
NWRS6_20_30	12	12	0,00	12	0,19
NWRS7_25_60	10	10	0,00	10	0,43
NWRS8_25_60	16	16	0,00	16	0,98
SP1_25_25	9	9	0,00	9	0,59
SP2_50_50	19	23	0,00	21	8,91
SP3_75_75	34	37	0,00	36	42,65
SP4_100_100	53	57	0,00	57	155,13

passo que $HNCM$ obteve as soluções ótimas para 17. A diferença entre os métodos se deram nas instâncias GP e SP, com $HNCM$ se sobressaindo nas primeiras e HBF_{2r} se sobressaindo nas últimas. No entanto, a diferença entre as soluções em ambos os casos não excedeu 2 pilhas abertas.

Os tempos de execução continuam baixos para as instâncias individuais, no entanto, nota-se o rápido aumento do tempo de execução de HBF_{2r} para as instâncias SP3_75_75 e SP4_100_100, embora o mesmo não ocorra para instâncias de dimensões semelhantes, como as instâncias GP.

4 Conclusão

Neste trabalho foi apresentada uma nova heurística para o Problema de Minimização de Pilhas Abertas, um problema NP-Difícil de escalonamento de produção correlato aos problemas de corte e empacotamento.

A heurística utiliza representação em grafos e a ela aplica o clássico algoritmo de busca em largura, inspirado por um método para solução do Problema de Minimização de Banda em Matrizes. Salvo melhor juízo, esta metodologia de solução não havia sido aplicada anteriormente ao Problema de Minimização de Pilhas Abertas. Associado a esta metodologia, duas regras de correção são aplicadas à solução obtida no intuito de superar casos em que hajam falhas e assim aprimorar a solução.

Nos experimentos computacionais realizados, o desempenho da heurística proposta foi aferido em comparação com os resultados ótimos de um conjunto de aproximadamente seis mil instâncias da literatura e também em comparação com a heurística de melhor desempenho encontrado na literatura. De acordo com os dados reportados, a heurística

proposta obteve um *gap* geral de 0,9% em relação aos resultados ótimos e impôs um *gap* de 0,4% à heurística de melhor desempenho da literatura, apresentando melhor comportamento médio. Em 88% dos casos os valores ótimos foram atingidos. Os tempos de execução são extremamente satisfatórios, com máximo de 155 milissegundos para a maior instância.

Ainda, o método proposto possui complexidade computacional menor em relação ao método heurístico de melhor desempenho da literatura: $O(p^2c)$ contra $O(c^3)$, em que c representa a quantidade de pedidos de compra e p representa a quantidade de produtos.

Considerando-se que se trata de um método heurístico e de acordo com os dados apresentados, é possível concluir que o método proposto é bastante satisfatório, uma vez que obteve bons resultados em um curto tempo de execução e também por sua facilidade de representação e implementação. O método proposto se apresenta como uma boa alternativa para solução do Problema de Minimização de Pilhas abertas, bem como pode ser utilizado para obtenção de bons limitantes quando incorporado em outros métodos mais elaborados.

Na continuação deste trabalho de pesquisa, pretende-se aferir o desempenho do método proposto frente a outros conjuntos de instâncias é até mesmo gerar novos conjuntos de instâncias. Também é considerado um trabalho futuro a investigação das condições que causam o pior desempenho da heurística, como no caso das instâncias individuais GP.

Referências

- Ashikaga, F. M. and Soma, N. Y. (2009). A heuristic for the minimization of open stacks problem. *Pesquisa Operacional*, 29(2):439–450.
- Becceneri, J. C., Yanasse, H. H., and Soma, N. Y. (2004). A method for solving the minimization of the maximum number of open stacks problem within a cutting process. *Comput. Oper. Res.*, 31(14):2315–2332.
- Carvalho, M. A. M. and Soma, N. Y. (2011). Métodos simplificados para o problema de minimização de pilhas abertas. *Gestão & Produção*, 18(2):299–319.
- Chu, G. and Stuckey, P. J. (2009). Minimizing the maximum number of open stacks by customer search. In Gent, I. P., editor, *15th International Conference on Principles and Practice of Constraint Programming*, volume 5732 of *Lecture Notes in Computer Science*, pages 242–257, Lisbon, Portugal. Springer.
- Cormen, T. H., Leiserson, C. E., Rivest, R. L., and Stein, C. (2001). *Introduction to Algorithms*. The MIT Press, 2nd revised edition edition.
- Cuthill, E. and McKee, J. (1969). Reducing the bandwidth of sparse symmetric matrices. In *Proceedings of the 1969 24th national conference*, ACM '69, pages 157–172, New York, NY, USA. ACM.
- de la Banda, M. G. and Stuckey, P. J. (2007). Dynamic programming to minimize the maximum number of open stacks. *INFORMS J. on Computing*, 19(4):607–617.
- Fink, C., Yanasse, H. H., and Costa, A. M. (2009). 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 – SBPO*, Porto Seguro, Brasil.
- Linhares, A. and Yanasse, H. H. (2002). Connections between cutting-pattern sequencing, vlsi design, and flexible machines. *Comput. Oper. Res.*, 29(12):1759–1772.
- Lopes, I. C. d. S. (2011). *Pattern sequencing models in cutting stock problems*. PhD thesis, Universidade do Minho, Portugal.

- Madsen, O. B. G. (1979). Glass cutting in a small firm. *Mathematical programming*, 17:85–90.
- Pinto, M. J. (2004). *Algumas Contribuições à Resolução do Problema de Corte Integrado ao Problema de Sequenciamento dos Padrões*. PhD thesis, Instituto Nacional de Pesquisas Espaciais, São José dos Campos.
- Smith, B.; Gent, I., editor (2005). *First Constraint Modelling Challenge, The Fifth Workshop on Modelling and Solving Problems with Constraints*, Edinburgh. IJCAI.
- Yanasse, H. H. (1997a). On a pattern sequencing problem to minimize the maximum number of open stacks. *European Journal of Operational Research*, 100(3):454–463.
- Yanasse, H. H. (1997b). A transformation for solving a pattern sequencing in the wood cut industry. *Pesquisa Operacional*, 17(1):57–70.
- Yanasse, H. H., Becceneri, J. C., and Soma, N. Y. (2002). 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 – SBPO*, Rio de Janeiro, Brasil.
- Yanasse, H. H., Becceneri, J. C., and Soma, N. Y. (2007). Um algoritmo exato com ordenamento parcial para solução de um problema de programação da produção – experimentos computacionais. *Gestão e Produção*, 14(2):353–361.
- Yanasse, H. H. and Pinto, M. J. (2003). Uma nova formulação para um problema de sequenciamento de padrões em ambientes de corte. In *Anais do XXXV Simpósio Brasileiro de Pesquisa Operacional – SBPO*, Natal, Brasil.
- Yanasse, H. H. and Senne, E. L. F. (2010). The minimization of open stacks problem: A review of some properties and their use in pre-processing operations. *European Journal of Operational Research*, 203(3):559–567.
- Yuen, B. J. (1991). Heuristics for sequencing cutting patterns. *European Journal of Operational Research*, 55(2):183–190.
- Yuen, B. J. (1995). Improved heuristics for sequencing cutting patterns. *European Journal of Operational Research*, 87(1):57–64.