UNIVERSIDADE DE SÃO PAULO

Instituto de Ciências Matemáticas e de Computação

UMA ESTRATÉGIA DE SOLUÇÃO PARA O PROBLEMA DE MINIMIZAÇÃO DO NÚMERO MÁXIMO DE PILHAS ABERTAS UTILIZANDO O PROBLEMA DO CAIXEIRO VIAJANTE

CLAUDIA FINK
PEDRO MUNARI
ALYSSON M. COSTA
HORACIO H. YANASSE

Nº 95

NOTAS DO ICMC

São Carlos - SP

UNIVERSIDADE DE SÃO PAULO
Instituto de Ciências Matemáticas e de Computação
ISSN 0103-2577

UMA ESTRATÉGIA DE SOLUÇÃO PARA O PROBLEMA DE MINIMIZAÇÃO DO NÚMERO MÁXIMO DE PILHAS ABERTAS UTILIZANDO O PROBLEMA DO CAIXEIRO VIAJANTE

CLAUDIA FINK
PEDRO MUNARI
ALYSSON M. COSTA
HORACIO H. YANASSE

Nº 95

NOTAS DO ICMC

SÉRIE COMPUTAÇÃO

São Carlos – SP
Jan./2010

Uma estratégia de solução para o problema de minimização do número máximo de pilhas abertas utilizando o problema do caixeiro viajante

Claudia Fink[1], Pedro Munari[1], Alysson M. Costa[1], Horacio H. Yanasse[2]

[1]Instituto de Ciências Matemáticas e de Computação
Universidade de São Paulo - Campus de São Carlos
claudiaf@icmc.usp.br; munari@icmc.usp.br; alysson@icmc.usp.br

[2]Instituto Nacional de Pesquisas Espaciais
São José dos Campos - SP
horacio@lac.inpe.br

Resumo

O problema de minimização do número máximo de pilhas abertas (MOSP) surge em diversos contextos, em particular, no sequenciamento do corte de placas de matéria-prima. Este trabalho propõe uma estratégia de solução baseada na resolução de um problema do caixeiro viajante (TSP) para a obtenção de um ciclo Hamiltoniano que é convertido em uma solução para o MOSP. A partir do grafo de peças associado ao problema MOSP, o TSP é definido por meio de uma regra de atribuição de distâncias que visa obter uma solução de boa qualidade. Nos testes computacionais realizados com exemplares da literatura, a estratégia proposta mostrou-se eficiente em relação às heurísticas específicas para o MOSP, obtendo a melhor solução para a maioria dos exemplares testes mesmo quando o TSP foi resolvido de forma aproximada. O desempenho da estratégia foi ainda melhor quando se utilizou a solução exata do TSP, obtendo a solução ótima em 80,42% dos casos.

Palavras chave: Problema de minimização do número máximo de pilhas abertas, problema do caixeiro viajante, sequenciamento, otimização combinatória.

Abstract

The minimization of open stacks problem (MOSP) arises in many contexts, and in particular when sequencing the cutting of boards of raw material. In this paper, we propose a solution method based on solving traveling salesman problem (TSP) in order to obtain a Hamiltonian cycle that is converted in a solution to the MOSP. From the graph of pieces associated with the MOSP, the TSP is defined by using a distance setting rule aiming to find a solution with good quality. In the computational tests carried out with instances from the literature, the proposed method was efficient in relation to other heuristics for the MOSP obtaining the best solution in the majority of instances, even when the TSP was solved approximately. The performance was even better when the exact solution of the TSP was used obtaining the optimal in 80.42% of the cases.

Keywords: Minimization of open stacks problem, traveling salesman problem, sequencing, combinatorial optimization.

1 Introdução

O problema de minimização do número máximo de pilhas abertas, MOSP (do inglês, *Minimization of open stacks problem*), aparece em diversas situações práticas. Por exemplo, considere o corte de placas de matéria-prima em peças menores. Supondo que os padrões de corte já tenham sido determinados, o problema consiste em sequenciar o corte das placas considerando o empilhamento de peças. As peças resultantes do corte de uma placa são empilhadas ao redor da máquina de corte e uma nova pilha é aberta toda vez que um novo tipo de peça é cortado. Uma pilha é formada por um único tipo de peça e deve permanecer aberta até que a última placa que contenha uma peça do tipo correspondente seja cortada. 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 peças do tipo correspondente a serem cortadas, as peças que estão na pilha podem ser removidas permitindo que as demais sejam adicionadas no mesmo local como se a pilha ainda estivesse aberta.

Outra aplicação para o MOSP surge no sequenciamento da produção de uma indústria, 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 do sequenciamento dos produtos poderia ser a minimização do número de pedidos que estão simultaneamente em aberto, o que levaria ao MOSP.

O MOSP é equivalente a vários outros problemas apresentados na literatura. Linhares e Yanasse (2002) citam vários problemas que estão relacionados com o MOSP, em particular, no contexto de projetos VLSI (do inglês, *Very-large-scale integration*). Yanasse (1997a) apresenta algumas relações entre a solução ótima do MOSP e de outros problemas de sequenciamento como o MORP (do inglês, *Minimization of order spread problem*) e o MDP (do inglês, *Minimization of discontinuities problem*).

Existem várias heurísticas sugeridas na literatura para a resolução do MOSP (vide Yuen (1991), Yuen (1995), Faggioli e Bentivoglio (1998), Ashikaga e Soma (2009), Becceneri et al. (2004), entre outros). Alguns trabalhos abordando o MOSP são baseados na teoria de programação inteira mista, como os modelos apresentados por Yanasse (1997a) e Yanasse e Pinto (2003) e os métodos exatos de enumeração implícita propostos por Yuen e Richardson (1995), Yanasse (1997a), Faggioli e Bentivoglio (1998), Becceneri et al. (2004) e Yanasse et al. (2007). Recentemente, uma estratégia utilizando programação dinâmica foi proposta por de la Banda e Stuckey (2007). Vale ressaltar, que o MOSP é um problema NP-difícil (Linhares e Yanasse, 2002) e, assim, a resolução de instâncias de grande porte por métodos exatos não é garantida na prática.

O MOSP pode ser definido como um problema em um grafo cujos vértices correspondem aos tipos de peças a serem sequenciadas e existe uma aresta adjacente a dois vértices se e somente se existe pelo menos um padrão de corte em que ambas as peças aparecem (Yanasse, 1997b). A definição do MOSP como um problema em um grafo foi utilizado por pesquisadores para a criação de heurísticas rápidas capazes de obter soluções com boa qualidade. Da mesma forma, fomos levados a pensar se uma solução para o MOSP não

poderia ser obtida relacionando-o com o problema do caixeiro viajante, ou TSP (do inglês, *Traveling salesman problem*). A vantagem potencial desta abordagem está no fato de que existem métodos heurísticos e exatos bastante eficientes disponíveis na literatura para a resolução 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).

Neste trabalho é proposta uma estratégia de solução para o MOSP que consiste em definir um TSP associado construído a partir do problema original de tal forma a obter uma boa solução para o problema original. A definição do TSP associado é feita por meio da atribuição de distâncias às arestas do grafo associado ao MOSP. Em seguida, o TSP é resolvido por métodos disponíveis na literatura. Como se trata de uma associação e não uma equivalência 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.

Este artigo está estruturado da seguinte maneira. Na Seção 2, é definido o grafo MOSP e são abordados alguns métodos de solução que consideram o problema MOSP definido em um grafo. Uma breve revisão sobre o TSP é dada na Seção 3 e, na Seção 4, é apresentada a estratégia de solução proposta para o MOSP utilizando o TSP. Na Seção 5, são descritos os testes computacionais realizados para a verificação do desempenho da estratégia proposta. A exposição e discussão dos resultados são feitas na Seção 6 e, por fim, na Seção 7 são apresentadas as conclusões e algumas propostas futuras.

2 Definição do grafo MOSP

O MOSP pode ser representado como um grafo $G(V, A)$, no qual cada vértice $v \in V$ representa um tipo de peça. Existe uma aresta $a_{ij} \in A$ ligando dois vértices $v_i, v_j \in V$ se, e somente se, as peças dos tipos correspondentes pertencerem a um mesmo padrão. Não existe uma atribuição de pesos às arestas, elas apenas representam a composição dos padrões de corte. Ainda que duas peças estejam presentes em mais de um padrão simultaneamente, apenas uma aresta é criada entre os vértices equivalentes. Na Figura 1, é ilustrado o grafo MOSP associado ao conjunto de padrões de corte descrito a sua direita. No exemplo, o padrão P_0 é formado por peças do tipo 0, 3 e 4. Assim, estão definidas as arestas a_{03}, a_{04} e a_{34}. Cada padrão de corte especificado corresponde a um clique no grafo, isto é, um sub-grafo completo.

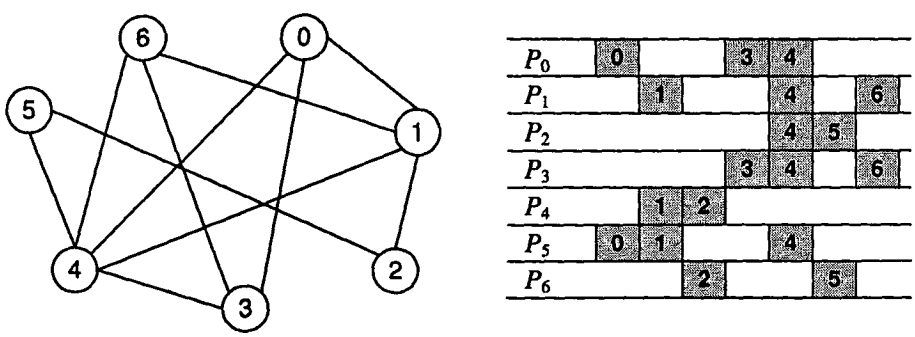

Figura 1: Grafo MOSP correspondente ao conjunto de padrões de corte à direita.

Métodos de solução do problema baseados no grafo MOSP têm sido propostos recentemente na literatura. Em particular, duas heurísticas específicas para a resolução do MOSP consideradas dentre as melhores existentes utilizando esta representação são a heurística de contração recursiva de cliques (HCC) e a heurística de nó de custo mínimo (HNCM). Elas são descritas a seguir.

A HCC, proposta por Ashikaga (2001), consiste em encontrar um clique maximal do grafo MOSP 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 peças, a qual determina a solução obtida pela heurística. Para obter uma sequência de padrões, o autor utiliza um procedimento simples, que consiste em percorrer a sequência de tipos de peças do final para o início. Os padrões que contém cada peça percorrida são inseridos na sequência de padrões, também do final para o início, considerando cada padrão uma única vez.

Proposta por Becceneri et al. (2004), a HNCM é uma das melhores heurísticas conhecidas para a resolução do MOSP, quanto à qualidade da solução (Ashikaga e Soma, 2009). Nesta heurística, é realizado o percorrimento de arestas do grafo MOSP, 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 a nós 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 peça (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 arcos que determina a ordem de fechamento dos vértices. Os autores então obtêm a sequência de padrões correspondente através do percorrimento destes arcos.

3 O problema do caixeiro viajante

O problema do caixeiro viajante, ou TSP (do inglês, *Traveling Salesman Problem*), é um dos problemas mais estudados de otimização combinatória (Laporte, 1992). O objetivo do TSP é encontrar a 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 resolução de outros problemas práticos, como o roteamento de veículos e o sequenciamento de tarefas. Apesar de sua descrição bastante simples, o TSP pertence à classe NP-difícil (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 $i, j = 1, ..., n$, 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 aresta a_{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 resolução 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 resolução 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 heurísticas de inserção está no critério de seleção do vértice a ser 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 resolução do TSP. O *software* livre Concorde possui uma implementação computacional destes métodos exatos considerada bastante eficiente (Applegate et al., 2009), sendo capazes de resolver instâncias de grande porte.

4 Estratégia de solução

No TSP definido a partir do grafo MOSP $G(V, A)$, o conjunto de vértices V passa a representar as cidades que devem ser visitadas e o conjunto de arestas A se torna o conjunto de estradas ligando estas cidades. A atribuição de distâncias às arestas é feita de modo que a resolução do TSP leve a uma boa solução para o MOSP. Diversas regras podem ser escolhidas para realizar essa atribuição e a qualidade da solução para o MOSP é bastante influenciada por essa escolha. Após a obtenção de uma solução para o TSP, é derivada uma solução para o MOSP. Nesta seção, são discutidas algumas regras e estratégias para a realização destas etapas. Ao final, é apresentada a estratégia de solução proposta neste trabalho.

4.1 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 $a_{ij} \in A$ uma aresta determinada pelos vértices $v_i, v_j \in V$. A distância d_{ij} associada a esta aresta é dada pela expressão:

$$d_{ij} = \max\{\Omega(v_i), \Omega(v_j)\} - |\mathrm{ADJ}(v_i) \cap \mathrm{ADJ}(v_j)|\,, \tag{1}$$

em que $|\,.\,|$ denota o número de elementos no conjunto, $\Omega(v)$ é o grau do vértice v e $\mathrm{ADJ}(v)$ representa o conjunto de vértices adjacentes a v.

A regra dada na expressão (1) é baseada em uma estimativa de pior caso para o número máximo de pilhas que são abertas quando uma aresta $a_{ij} \in A$ é percorrida e o vértice v_j é visitado. A visita a um vértice do grafo MOSP representa o fechamento da pilha do tipo de peça correspondente e a abertura de todas aquelas representadas pelos vértices adjacentes,

caso já não estejam abertas. Com isso, o maior grau dentre os dois vértices de uma aresta fornece um limitante superior para o número de pilhas que podem ser abertas ao percorrê-la. Este limitante é mais realista se dele for subtraído o número de vértices adjacentes a ambos os vértices que determinam a aresta. De fato, quando uma aresta é percorrida, os vértices adjacentes a ambos os vértices que a determinam já foram abertos devido à visita ao vértice de saída.

Em testes preliminares, outras regras de atribuição semelhantes 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.

4.2 Conversão da solução do TSP em uma sequência de tipos de peça

Toda solução factível de um TSP corresponde a um ciclo Hamiltoniano, cuja distância total correspondente independe do vértice inicial. Entretanto, para o grafo MOSP, deve-se obter um circuito Hamiltoniano, isto é, o vértice inicial não deve ser visitado novamente, e o número máximo de pilhas abertas pode ser diferente dependendo do vértice tomado como inicial. Desta forma, é necessário definir uma estratégia para a conversão de uma solução do TSP na melhor solução possível para o MOSP.

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 n vezes, 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.

4.3 Sequência de padrões a partir de uma sequência de peças

Em métodos baseados no grafo MOSP, a solução obtida é geralmente dada por uma sequência de tipos de peças. Entretanto, o objetivo do MOSP é obter uma sequência de padrões de corte, havendo, assim, a necessidade de converter a solução obtida. Conforme discutido na Seção 2, Ashikaga (2001) percorre de forma inversa a sequência de tipos de peças para obter uma sequência de padrões correspondente. Por outro lado, Becceneri et al. (2004) utilizam um conjunto de arcos obtido como solução da heurística, para a construção da sequência de padrões.

Na estratégia de solução aqui considerada, é proposto um procedimento para a obtenção de uma sequência de padrões a partir da sequência de peças, baseando-se no procedimento utilizado por Becceneri et al. (2004) que foi sugerido em Yanasse (1997b). A ideia consiste em percorrer a sequência de peças e inserir um padrão na sequência de padrões quando, pela primeira vez, todos os tipos de peças do padrão tiverem sido percorridos. A sequência de tipos de peça é percorrida do início para o fim, porém sempre que a pilha de um tipo

de peça é fechada, as pilhas dos tipos de peças correspondentes a vértices adjacentes são abertas. Este procedimento é descrito no Algoritmo 1, que por questão de exposição, faz uso do Algoritmo 2 nas linhas 7 e 13, com diferentes valores para o parâmetro j. A notação $|P_j|$ corresponde ao número de tipos de peça no padrão de corte P_j. O arranjo π_flag é utilizado para sinalizar se uma pilha ainda não foi aberta ($= 0$), se encontra aberta ($= 1$) ou já foi fechada ($= -1$). Cada componente de P_flag contabiliza a quantidade de tipos de peça que restam ser analisados em um padrão, sendo igual a -1 caso o padrão já tenha sido adicionado à sequência de padrões.

Algoritmo 1. Obtenção de uma sequência de padrões a partir de uma sequência de tipos de peças.

Entrada: Conjunto de padrões $\{P_0, \ldots, P_{m-1}\}$; sequência de tipos de peças $S_\pi = \{\pi_0, \ldots, \pi_{n-1}\}$.
Saída: Sequência de padrões S_P.

```
1  S_P = ∅;
2  Defina os arranjos π_flag[i] = 0 e P_flag[j] = |P_j|, para i = 0,...,n−1 e j = 0,...,m−1;
3  Para i = 0,...,n − 1 faça
4  {
5     k = π_i;
6     /* Analisar os padrões que contenham o tipo de peça k */
7     Se (π_flag[k] = 0) então execute o Algoritmo 2 com j = k;
8
9     /* Percorrer os vértices adjacentes a v_k */
10    Para cada l tal que v_l ∈ ADJ(v_k) faça
11    {
12       /* Analisar os padrões que contenham o tipo de peça l */
13       Se (π_flag[l] = 0) então execute o Algoritmo 2 com j = l;
14    }
15    /* Sinalizar que a pilha do tipo de peça k foi fechada */
16    π_flag[k] = −1;
17 }
```

Algoritmo 2. Define a ordem de processamento dos padrões para o fechamento de uma pilha.

Entrada: Conjunto de padrões $\{P_0, \ldots, P_{m-1}\}$; arranjos π_flag e P_flag; tipo de peça j.
Saída: Atualização da sequência S_P e dos arranjos π_flag e P_flag.

```
1  Para cada padrão P_q contendo o tipo de peça j faça
2  {
3     Se (P_flag[q] > 0) P_flag[q] = P_flag[q] − 1;
4
5     /* Verificar se todos os tipos de peças do padrão já foram processados */
6     Se (P_flag[q] = 0) então
7     {
8        /* Adicionar o padrão P_q à sequência S_P e sinalizá-lo como processado */
9        S_P = S_P + {P_q};
10       P_flag[q] = −1;
11    }
12 }
13 /* Sinalizar que a pilha do tipo de peça j foi aberta */
14 π_flag[j] = 1;
```

O procedimento utilizado por Ashikaga (2001) 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. Assim, o procedimento aqui proposto foi adotado como padrão na estratégia de solução.

4.4 Descrição da proposta

Baseando-se na discussão apresentada, a seguinte estratégia de solução para o MOSP é proposta:

- *Passo 1.* Defina o grafo MOSP $G(V, A)$.
- *Passo 2.* Para cada aresta $a_{ij} \in A$, atribua a distância d_{ij} de acordo com (1).
- *Passo 3.* Acrescente arestas com distância infinita ao conjunto A, de modo a tornar $G(V, A)$ um grafo completo.
- *Passo 4.* Resolva o TSP descrito por $G(V, A)$, obtendo um ciclo Hamiltoniano $H = \{v_{\pi_1}, v_{\pi_2}, \ldots, v_{\pi_n}, v_{\pi_1}\}$.
- *Passo 5.* Para cada $i = 1, \ldots, n$, determine a sequência de tipos de peças H_i, obtida a partir de H com início em v_{π_i} e término em $v_{\pi_{i-1}}$, tal que $v_{\pi_0} = v_{\pi_n}$.
- *Passo 6.* Para cada $i = 1, \ldots, n$, obtenha a sequência de padrões correspondente à H_i e calcule o número máximo de pilhas abertas z_i.
- *Passo 7.* A solução para o MOSP é dada pela sequência de padrões correspondente ao menor z_i obtido.

A resolução do TSP no *Passo 4* 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 tendência de se obter uma boa solução para o MOSP.

Para ilustrar a estratégia proposta, um exemplo é apresentado na sequência. Considere o grafo MOSP dado na Figura 1. A cada aresta $a_{ij} \in A$ é atribuída uma distância d_{ij} de acordo com a expressão (1). Por exemplo, a aresta a_{03} recebe a distância $d_{03} = max\{3, 3\} - |\{1, 3, 4\} \cap \{0, 4, 6\}| = 2$. Arestas com distância infinita são inseridas no grafo para torná-lo completo. O resultado é dado na Figura 2, denotando arestas com distância infinita por linhas tracejadas.

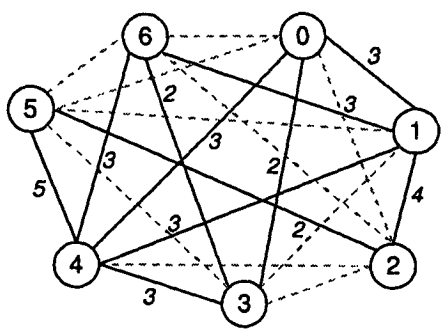

Figura 2: Grafo TSP obtido a partir do grafo MOSP apresentado na Figura 1, usando a estratégia proposta. As linhas tracejadas representam arestas com distância infinita.

Uma possível solução para o TSP representado na Figura 2 é dada pelo ciclo Hamiltoniano $H = \{0, 4, 5, 2, 1, 6, 3, 0\}$, com distância total igual a 21 (neste caso, tem-se o valor ótimo). A partir deste ciclo, são analisadas as possíveis sequências de tipos de peças H_i e as respectivas sequências de padrões, para $i = 1, ..., n$, apresentados na Tabela 1. 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.

i	H_i	Sequência de padrões	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

Tabela 1: Sequências de peças analisadas a partir do ciclo Hamiltoniano obtido no TSP, sequência de padrões correspondentes e número máximo de pilhas abertas.

5 Testes 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 3, foram implementadas para a resolução do TSP. Inicialmente, foi realizada a comparação da estratégia proposta com outros métodos da literatura. Para isso, foram implementados dois procedimentos específicos para a resolução do MOSP, as heurísticas de contração recursiva de cliques (HCC) e de nó de custo mínimo (HNCM), apresentadas na Seção 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 dos problemas de teste.

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.

5.1 Problemas de teste

Os problemas de teste foram retirados da biblioteca utilizada no Desafio de Modelagem de Restrições 2005, disponível em `http://www.cs.st-andrews.ac.uk/~ipg/challenge/`. Na Tabela 2, são descritas as classes de problemas aqui consideradas. Um problema que pertence à classe cujo nome é iniciado por `wbo` possui um número fixo de tipos de peças por padrão. Este número varia dentre os problemas da classe como, por exemplo, de 2 a 5 na classe `wbo_10_10`. Para as classes que iniciam com `wbp`, tem-se problemas nos quais cada tipo de peça 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 problemas variados, considerados de grande porte. No nome das classes também estão indicados o número de tipos de peças ($|S_\pi|$) e o número de padrões ($|S_P|$) dos problemas que as compõem.

Classe	Problemas	$\lvert S_\pi \rvert$	$\lvert S_P \rvert$
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
gp50by50	4	50	50
gp100by100	4	100	100

Tabela 2: Classes de problemas utilizadas nos testes computacionais.

5.2 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 problemas de teste e $\mathcal{S}$ o conjunto de n_s *softwares* correspondentes aos métodos implementados. O valor da solução obtida para o problema $p \in \mathcal{P}$ 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 resolução de um problema $p \in \mathcal{P}$, 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:

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

com $|.|$ 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 problemas resolvidos pelo *software* com um desempenho dentro de um fator τ 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 τ:

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

Os perfis de desempenho calculados devem ser apresentados em um único gráfico, cuja abscissa represente a variação do parâmetro τ. Neste trabalho, 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`.

6 Resultados e discussão

As informações sobre a resolução dos problemas de teste usando-se os métodos implementados são dadas nas Tabelas 3 e 4. No total, são 378 problemas de teste agrupados 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 dos problemas. A resolução utilizando a HNCM foi a que exigiu maior tempo em grande parte dos problemas. Por outro lado, a heurística HCC resolveu os problemas

em um tempo computacional muito pequeno e sem relação aparente com a densidade do problema, não ultrapassando 3 milissegundos nem mesmo para os problemas de teste com maior dimensão. Entretanto, as soluções obtidas por esta heurística foram relativamente inferiores.

		HID				HIA				HCC				HNCM			
		Valor		t (ms)		Valor		t (ms)		Valor		t (ms)		Valor		t (ms)	
Classe	D	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	0,48
	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 3: Informações sobre a resolução dos problemas de teste pertencentes às classes especificadas. Os resultados foram agrupados de acordo com a densidade (D) dos problemas e são apresentados a média aritmética (M) e o desvio padrão (DP) de cada grupo, considerando o valor da solução e o tempo de resolução (em milissegundos).

Na Figura 3, 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% dos problemas. 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 resolução do MOSP, de acordo com as comparações realizadas por Yanasse et al. (2007) e Becceneri et al. (2004).

Dado o desempenho geral semelhante da estratégia proposta em relação à HNCM, é

		HID		HIA		HCC		HNCM	
Classe	D	Valor	t (ms)	Valor	t (ms)	Valor	t (ms)	Valor	t (ms)
gp50by50	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
gp100by100	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

Tabela 4: Valor da solução e tempo de resolução para cada problema das classes gp50by50 e gp100by100 usando os métodos de solução indicados.

interessante verificar a qualidade da solução obtida por estes métodos em relação à solução ótima dos problemas. Para isso, foi utilizado o método exato de enumeração proposto por Becceneri et al. (2004) e aprimorado por Yanasse et al. (2007). Os problemas de teste foram resolvidos 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 problemas excedeu-se a memória computacional disponível sem que o método pudesse identificar a solução ótima, sendo considerada a melhor solução obtida nesses casos.

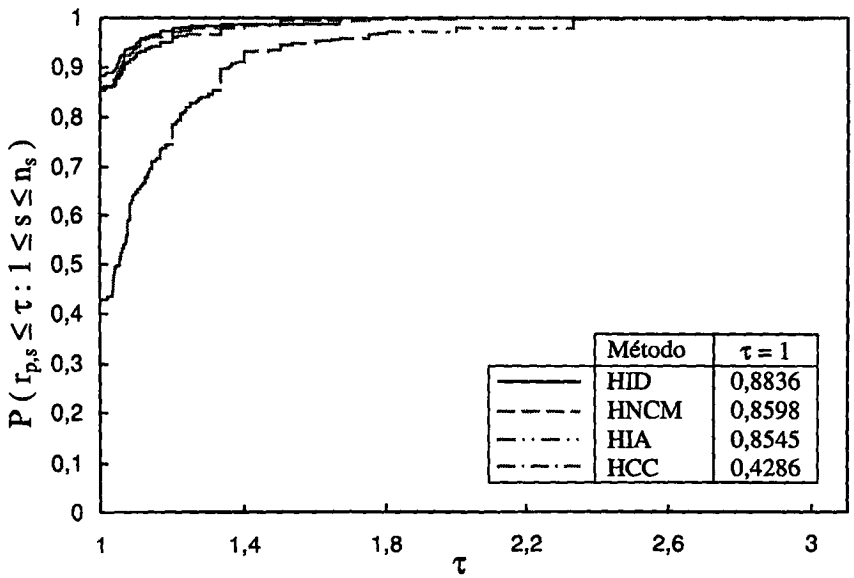

Figura 3: Comparação entre a estratégia proposta e as heurísticas específicas para o MOSP, considerando os valores das soluções obtidas na resolução dos problemas de teste.

Para a comparação dos resultados, foi calculada a diferença entre a solução ótima e a solução obtida por cada método, em cada um dos problemas de teste. Na Tabela 5, é apresentada a quantidade de problemas para os quais cada método obteve a solução ótima (+0) e a quantidade de problemas 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 ótimas foi próxima para os métodos considerados. Entretanto, a diferença obtida pela estratégia proposta não ultrapassou 3 pilhas em nenhum dos problemas (independente da forma de resolução do TSP), enquanto a HNCM obteve uma diferença maior que três pilhas para 7

problemas.

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

Tabela 5: Quantidade de problemas para os quais é obtida a diferença de 0, 1, 2, 3 ou mais pilhas em relação à solução ótima.

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

Na Tabela 6, é dada a diferença entre a solução ótima e a solução obtida pela estratégia proposta usando a solução exata do TSP, conforme a densidade dos problemas. Em nenhum dos problemas, 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%) problemas e para apenas 1 problema foi obtida a diferença de 3 pilhas em relação à solução ótima. Se comparados estes valores com os apresentados na Tabela 5, 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.

Classe	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

Classe	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
	0,500	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
gp50by50	0,539	0	1	0	0
	0,654	1	0	0	0
	0,632	0	1	0	0
	0,808	1	0	0	0
gp100by100	0,543	0	1	0	0
	0,672	0	1	0	0
	0,677	0	1	0	0
	0,857	1	0	0	0
Total		**304**	**58**	**15**	**1**

Tabela 6: Quantidade de problemas para os quais a solução obtida pela estratégia proposta, utilizando a solução exata do TSP, possui diferença de 0, 1, 2 ou 3 pilhas em relação à solução ótima.

Note que, de modo geral, a estratégia proposta apresentou maior diferença em relação à solução ótima para os problemas com densidade $D \leq 0,300$. É também nestes problemas que o 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 problemas dentro do tempo especificado, cuja densidade é $0,067$. Também teve dificuldade em resolver dois problemas 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 os problemas MOSP mais esparsos. Entretanto, vale ressaltar que não somente a densidade influencia na dificuldade de resolução dos problemas, como pode ser observado para as classes `gp50by50` e `gp100by100`, cujos problemas são mais densos e possuem maior número de padrões e peças.

7 Conclusões

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 principais 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% dos problemas de teste. Para os demais problemas, 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.

Agradecimentos

Os autores agradecem os comentários e as sugestões dos revisores anônimos e ao Prof. Dr. José Carlos Becceneri por fornecer o arquivo executável correspondente à implementação computacional do método exato de enumeração utilizado. Este trabalho foi financiado pela Fundação de Amparo à Pesquisa do Estado de São Paulo (FAPESP).

Referências

AGARWALA, R.; APPLEGATE, 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.

APPLEGATE, D. L.; BIXBY, R. E.; CHVÁTAL, V.; COOK, W. J. *The traveling salesman problem: a computational study.* Princeton University Press, 2006, 593 p., 2006.

APPLEGATE, D. L.; BIXBY, R. E.; CHVÁTAL, V.; COOK, W. J. Concorde TSP solver. URL: 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. *Informs Journal on Computing*, v. 19, n. 4, p. 607–617, 2007.

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.

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.

GAREY, M. R.; JOHNSON, D. *Computers and intractability: A guide to the theory of NP-completeness.* 1 ed. Freeman, 1979.

HAHSLER, M.; HORNIK, K. TSP - infrastruture 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.

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.

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.

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, disponível em: http://www.icmc.usp.br/~biblio/relatorios_tecnicos.php, 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.

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.

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.

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.

NOTAS DO ICMC

SÉRIE COMPUTAÇÃO

94/07 MARTINS, F.P. – Aspectos teóricos e computacionais da geração de malhas quadtree.

93/07 SANTOS, M.O.; TOLEDO, F.M.B.; ARAÚJO, S.A. – Uma abordagem utilizando relaxação lagrangeana/surrogate para o problema de dimensionamento de lotes e distribuição.

92/07 LOPES, M.A.L.P.; ARENALES, M. N.; SANTOS, M. O. - Some extensions of dual cuts to one-dimensional cutting stock problems.

91/07 ALEM, D. J. ; MUNARI JUNIOR, P. A.; ARENALES, M. N.; FERREIRA, P. A. V. - On the cutting stock problem under stochastic demand.

90/07 CHERRI, A.C.; ARENALES, M.N.; YANASSE, H.H. - The unidimensional cutting stock problem with usable leftover – a heuristic approach.

89/07 TOLEDO, F.M.B. ; ARMENTANO, V.A. - Branch-*and*-bound algorithms for capacitated lot-sizing in parallel machines.

88/07 TOLEDO, F.M.B.; SANTOS, M.O.; ARENALES, M.N.; SELEGHIM JÚNIOR, P. – Logística de distribuição de água em redes urbanas – racionalização energética.

87/05 GOIS, J.P.; ESTÁCIO, K.C.; OISHI, C.M.; BERTTONI, V.; BOTTA, V.A AVANSINI; NAGAMINE, A.; KUROKAWA, F.A.; FEDERSON, F. - Aplicação de volumes finitos na simulação numérica de contaminação em lençóis freáticos.

86/05 MARQUES, F.P.; ARENALES, M.N. - The constrained compartmentalised knapsack problem.

85/05 POLDI, K.C.; ARENALES, M.N. – Dealing with small demand in integer cutting stock problems with limited different stock lengths.