UMA NOVA FORMULAÇÃO PARA UM PROBLEMA DE SEQÜENCIAMENTO DE PADRÕES EM AMBIENTES DE CORTE

Horacio Hideki Yanasse
Instituto Nacional de Pesquisas Espaciais (INPE)
Laboratório Associado de Computação e Matemática Aplicada (LAC)
Av. dos Astronautas, 1758, Jardim da Granja,
S.J.Campos, SP, 12227-010
e-mail: horacio@lac.inpe.br

Maria José Pinto
Centro Técnico Aeroespacial (CTA)
Instituto de Estudos Avançados (IEAv)
Departamento de Informática (EIN-A)
Caixa Postal 6044 - São José dos Campos, SP, 12231-970
maju@ieav.cta.br

Resumo

Propõe-se uma nova formulação matemática para um problema de seqüenciamento de padrões em ambientes de corte. Compara-se esta formulação com outras sugeridas anteriormente na literatura.

Palavras-Chave: Problema de Corte de Estoque, Sequenciamento de Padrões, MOSP, MTSP.

Abstract

We propose a new mathematical formulation of a pattern sequencing problem in cutting settings. We compare this formulation with previous ones suggested in the literature.

Keywords: Cutting Stock Problem, Pattern Sequencing , MOSP, MTSP.

1. INTRODUÇÃO

O problema de corte de estoque consiste em cortar peças maiores (*objetos*), para obtenção de peças menores (*itens*) em dimensões e quantidades específicas. O objetivo normalmente é minimizar os custos da produção, desperdícios, maximizar lucros, etc. Na solução deste problema são obtidos os *padrões de corte* (como os objetos serão cortados) e a quantidade de vezes que cada padrão será cortado para atender toda a demanda.

Associado ao problema de corte, podemos ter um problema de seqüenciamento dos padrões, uma vez que a ordem de corte tem influência, por exemplo, no estoque dos itens sendo cortados, nas descontinuidades no corte dos itens, no número máximo de pilhas abertas durante o processo de corte, etc. Neste trabalho focalizamos este último problema ao qual denotamos de MOSP (do inglês *Minimization of Open Stack Problem*), etc. No caso do MOSP, cada tamanho diferente de item a ser cortado, abre uma pilha nova que permanece aberta até que o último item daquele tamanho seja cortado.

O problema surge devido a, por exemplo, limitações físicas, quando o número de pilhas que podem ficar abertas ao redor da máquina de corte é limitado, sendo necessário o remanejamento das pilhas incompletas, comprometendo a eficiência do processo produtivo. Outros fatores que justificam o interesse neste problema de sequenciamento são, por exemplo: custos de estoque, restrições de espaço de armazenagem, etc.

Yanasse [1] apresentou uma formulação matemática para o MOSP, baseada na formulação proposta por Tang e Denardo [2] para o MTSP (do inglês *Minimization of the*

number of Tool Switches Problem). O MTSP consiste em minimizar o número de trocas de ferramentas durante o seqüenciamento das tarefas que precisam ser processadas em uma determinada máquina. As trocas ocorrem devido à limitação na quantidade de ferramentas que podem ser carregadas na máquina num mesmo instante e ao fato do número de ferramentas necessárias para processar todas as tarefas ser superior à esta limitação. A importância deste problema está no fato de que, a cada troca realizada, a produção da máquina fica comprometida devido à interrupção necessária para a realização das trocas. Quando, por exemplo, o tempo de interrupção da máquina é proporcional ao número de ferramentas trocadas, reduzir o número de trocas implicará em um tempo disponível maior de produção de máquina.

Devido às dificuldades computacionais para se resolver o MTSP utilizando a formulação proposta por Tang e Denardo [2], duas novas formulações para o MTSP foram recentemente propostas, uma por Yanasse e Pinto [3] e outra por Laporte, Salazar e Semet [4]. Em Yanasse e Pinto [3], o MTSP é formulado como um problema de fluxo com restrições adicionais e em Laporte, Salazar e Semet [4] como um problema do caixeiro viajante.

Apesar de algumas características interessantes observadas em ambas formulações, constatou-se que algumas dificuldades computacionais para se resolver o MTSP ainda persistem.

Para se estender a formulação proposta de Yanasse e Pinto [3] do MTSP para o MOSP seria necessário conhecer de antemão o número máximo de pilhas abertas. Este valor, normalmente, não está disponível o que, portanto, inviabiliza esta extensão. Quanto ao modelo de Laporte, Salazar e Semet [4], ele pode ser estendido da mesma forma que foi feita com o modelo de Tang e Denardo [2].

Neste trabalho, estamos propondo uma nova formulação para a resolução do MOSP, com um embasamento completamente distinto dos anteriores.

Espera-se que, com esta formulação, o desempenho computacional para se obter uma solução ótima para o problema seja melhorado em relação aos modelos propostos anteriormente.

2. NOVO MODELO

Nos modelos de Yanasse [1] e Laporte, Salazar e Semet [4], procura-se determinar a ordem em que os padrões devem ser processados para que o número máximo de pilhas abertas seja minimizado.

Neste novo modelo proposto, procura-se determinar a ordem em que as pilhas são fechadas, da mesma forma que a enumeração proposta no algoritmo branch-and-bound em Yanasse [1]. Cada pilha corresponde a um item diferente do problema. Para que uma pilha correspondente a um certo item i seja fechada é preciso que todos os padrões que contenham o item i sejam processados. Outros padrões, não necessariamente contendo o item i, podem também ser processados. Em Yanasse [5] foi apresentada a transformação do MOSP para um problema de percorrimento de arcos em um grafo e, com isso, fica trivial mostrar que existe sempre uma seqüência ótima onde apenas os padrões que completam cada um dos itens na seqüência são processados. Portanto, é suficiente procurar por uma seqüência dos padrões deste tipo.

Ao final do processamento de todos os padrões que completam um item, necessariamente pelo menos uma pilha é completada, portanto, o número de pilhas completas sempre cresce de uma unidade. O número total de pilhas resultante é sempre igual ao número de pilhas abertas mais o número de pilhas completas.

Pilhas uma vez abertas sempre permanecem como pilhas (abertas ou fechadas) até o processamento final de todos os padrões. Assim, todas as pilhas resultantes após a finalização de cada item são sempre as mesmas pilhas que estavam presentes na finalização anterior de algum item mais as novas pilhas abertas para a finalização do item corrente.

Considere um exemplar simples de um problema de seqüenciamento de padrões, dado pela Tabela 1.

Tabela 1. Exemplo ilustrativo, $M = 5$, $N = 8$

Padrões	**Itens**				
	1	**2**	**3**	**4**	**5**
1	1	0	0	1	0
2	1	0	1	0	1
3	0	1	0	1	0
4	0	0	1	1	0
5	1	0	0	0	1
6	0	0	1	0	1
7	0	0	0	1	0
8	1	1	0	1	0

Na Tabela 1, os itens presentes em cada padrão estão indicados com um 1. Caso o item não esteja presente no padrão, a indicação é feita com um 0. Assim, o item 2, por exemplo, está presente nos padrões 3 e 8 somente; o item 5 está presente nos padrões 2, 5 e 6.

Para se completar o item 1 por exemplo, teremos que processar os padrões 1, 2, 5 e 8. Se processarmos estes padrões, abriremos pilhas correspondentes aos itens 1, 2, 3, 4 e 5, ou seja, uma pilha para cada um dos itens que vão ser processados juntos com o item 1. Ao final do processamento dos padrões que completam o item 1, teremos um total de 5 pilhas, 4 pilhas abertas correspondentes aos itens 2, 3, 4 e 5 (admitindo-se que nenhuma delas seja também fechada), e uma pilha fechada correspondente ao item 1. Para completar o item 2, teremos que processar os padrões 3 e 8. Assim, para se completar o item 2, abriremos pilhas correspondentes aos itens 1, 2 e 4. Mas, se decidimos completar o item 2 após completar o item 1, todas estas pilhas já estão abertas e ou completas. Assim, as pilhas abertas e as fechadas permanecem as mesmas, mas ao final do processamento do padrão 3, o único padrão que ainda necessitaria ser processado pois o padrão 8 já foi processado para se completar o padrão 1, teremos mais uma pilha fechada, a pilha 2, além da pilha 1. Observe que podemos ignorar o fato de que o padrão 8 já foi processado e processá-lo novamente. Isto não aumenta o número de pilhas além das já existentes. Para se saber quais pilhas realmente estão abertas basta apenas saber quais as pilhas que foram fechadas anteriormente.

Com estas observações, seja

N Número de padrões a serem processados;

M Número total de itens;

K Constante ($K \geq M$);

$\boldsymbol{S}_j$ Vetor $M \times 1$ de 1's e 0's que informa quais itens estão presentes juntos com o item j nos padrões do problema; ou seja S_{ij} é igual a 1 se o item i e o item j estão presentes em algum padrão do problema, e 0, caso contrário.

n Instante imediatamente após completar o n-ésimo item;

C Número máximo de pilhas abertas;

$\boldsymbol{e}$ Vetor auxiliar $1 \times M$ de 1's;

$\boldsymbol{x}_{jn}$ é 1 se o item j é o n-ésimo item a ser completado e, é 0, caso contrário;

$\boldsymbol{W}_n$ Vetor $M \times 1$ de 1's e 0's que fornece os itens cujas pilhas já estão abertas ou fechadas no instante n, $n = 1, ..., M$.

O MOSP pode então ser formulado como se segue.

Modelo 1:

Função objetivo

$$\min C \tag{1}$$

Sujeito à

$$\boldsymbol{eW}_n \leq C+n-1 \qquad n = 1,\ldots, M \tag{2}$$

$$\sum_{j=1}^{M} \sum_{t=1}^{n} \boldsymbol{x}_{jt} \boldsymbol{S}_j \leq K\boldsymbol{W}_n \qquad n = 1,\ldots, M \tag{3}$$

$$\sum_{n=1}^{M} \mathbf{x}_{jn} = 1 \qquad j = 1,\ldots, M \tag{4}$$

$$\sum_{j=1}^{M} \mathbf{x}_{jn} = 1 \qquad n = 1,\ldots, M \tag{5}$$

$$\boldsymbol{x}_{jn} \in \{0,1\} \qquad j = 1,\ldots, M\,;\ n = 1,\ldots, M \tag{6}$$

$\boldsymbol{W}_n$ vetor de inteiros *0* ou *1*. (7)

Após a finalização da *n*-ésima pilha, temos *n* pilhas fechadas. Durante a finalização da *n*-ésima pilha, teremos um total de *C* pilhas abertas, mais o número de pilhas já completas no instante *n-1*. A restrição (2) apenas relaciona o número total de pilhas (abertas e fechadas) durante cada finalização de uma pilha deve ser menor ou igual ao número máximo de pilhas abertas mais as pilhas fechadas até a finalização anterior de uma pilha. As restrições (3) indicam que se uma pilha *i* é finalizada, todo o item *j* que aparece conjuntamente com o item *i* em algum padrão também formará uma pilha. As restrições (4) e (5) indicam que cada uma das pilhas será finalizada em alguma ordem e, as restrições (6) e (7) são as restrições de integralidade das variáveis de decisão.

3. COMPARAÇÃO COM MODELOS ANTERIORES

Para fins de comparação, apresentamos os modelos formulados anteriormente por Yanasse [1] e, o estendido a partir de Laporte, Salazar e Semet [4].

Seja

$\boldsymbol{A}_j$ Vetor $M \times 1$ de 1's e 0's que informa quais itens estão presentes no padrão j; ou seja A_{ij} é igual a 1 se o item *i* está no padrão *j*, e 0, caso contrário.

n Instante imediatamente após o processamento do *n*-ésimo padrão;

$\boldsymbol{x}_{jn}$ $\begin{cases} 1, \text{se o padrão } j \text{ é o } n\text{-ésimo padrão na seqüência;} \\ 0, \text{caso contrário.} \end{cases}$

$\boldsymbol{W}_n$ Vetor $M \times 1$ de 1's e 0's que fornece os itens cujas pilhas estão abertas na máquina em n, $n = 1,\ldots,N$;

$\boldsymbol{P}_n$ Vetor $M \times 1$ que contabiliza as trocas dos itens que estão sendo cortados na máquina de corte, ocorridas do instante *n* para o *n+1*;

O modelo 2 de Yanasse [1] é o seguinte:

Função objetivo

$$\min C \tag{8}$$

Restrições

$$\boldsymbol{P}_n \geq \boldsymbol{W}_{n+1} - \boldsymbol{W}_n \qquad N = 1,\ldots, N\text{-}1 \tag{9}$$

$$\boldsymbol{P}_n \geq \boldsymbol{0} \qquad N = 1,\ldots, N\text{-}1 \tag{10}$$

$$\boldsymbol{e}\boldsymbol{W}_n \leq C \qquad N = 1,\ldots, N \tag{11}$$

$$\boldsymbol{x}_{jn}\boldsymbol{A}_j \leq \boldsymbol{W}_n \qquad J = 1,\ldots, N;\ n = 1,\ldots, N \tag{12}$$

$$\sum_{n=1}^{N-1} \boldsymbol{e}\boldsymbol{P}_n = M - C \tag{13}$$

$$\sum_{n=1}^{N} \boldsymbol{x}_{jn} = 1 \qquad J = 1,\ldots, N \tag{14}$$

$$\sum_{j=1}^{N} \boldsymbol{x}_{jn} = 1 \qquad n = 1,\ldots, N \tag{15}$$

$$\boldsymbol{x}_{jn} \in \{0,1\} \qquad J = 1,\ldots, N;\ n = 1,\ldots, N \tag{16}$$

$$\left.\begin{array}{l} \boldsymbol{P}_n,\ n = 1,\ldots,N-1 \\ \boldsymbol{W}_n,\ n = 1,\ldots,N \end{array}\right\} \text{ vetor de inteiros } 0 \text{ ou } 1. \tag{17}$$

Neste modelo (2), a função objetivo (8) minimiza o número máximo de pilhas abertas, enquanto que as restrições (9), juntamente com as restrições (10), indicam as pilhas novas abertas com o corte do *n*-ésimo padrão. As restrições (11) garantem que o sequenciamento dos padrões não abrirá mais do que *C* pilhas. As restrições (12) indicam quais são as pilhas abertas caso o *j*-ésimo padrão for processado no instante *n*. garantem que As restrições (13) garantem que o número de pilhas remanejadas é exatamente *M-C*, ou seja, apenas pilhas completas são remanejadas. As restrições (14), (15) e (16) impõem que todos os padrões sejam processados uma única vez em alguma ordem. Finalmente, as restrições (17) estão relacionadas à condição de integralidade de variáveis do problema. Observe que a variável de decisão $\boldsymbol{x}_{jn}$ neste modelo (2) é completamente diferente do modelo (1).

Seja:

x_{ij} variáveis binárias iguais a *1*, se e somente se, a tarefa *i* é imediatamente seguida pela tarefa *j*.

y_{ij} variáveis binárias iguais a *1*, se e somente se, a ferramenta *i* está na máquina enquanto a tarefa *j* está sendo processada.

z_{ij} variáveis binárias iguais a *1*, se e somente se, a ferramenta *i* terá que ser inserida na máquina no início do processamento da tarefa *j*. Em outras palavras, $z_{ij} = 1$ corresponde à troca de ferramentas.

Duas tarefas adicionais (fictícias), denominadas de ***O*** e ***D***, são introduzidas para representar o início e o fim das operações. No caso, não existe ligação entre estes nós fictícios, ou seja: $\boldsymbol{x}_{\boldsymbol{OD}} = 0$. Feitas estas considerações e definindo que: $J_1 = 1,2,\ldots,N,\boldsymbol{D}$ e $J_2 = \boldsymbol{O},1,2,\ldots,N,$ o modelo (3), extensão do modelo proposto por Laporte, Salazar e Semet [4] para o MOSP fica:

Min C (18)

Sujeito a:

$$\sum_{\substack{j \in J_1 \\ j \neq i}} x_{ij} = 1 \qquad i \in J_2 \qquad (19)$$

$$\sum_{\substack{i \in J_2 \\ i \neq j}} x_{ij} = 1 \qquad j \in J_1 \qquad (20)$$

$$\sum_{i=1}^{M} y_{ij} \leq C \qquad j = 1, \ldots, N \qquad (21)$$

$$x_{ij} + y_{kj} - y_{ki} \leq z_{kj} + 1 \qquad i, j = 1,2,\ldots,N \text{ e } k = 1,2,\ldots,M \qquad (22)$$

$$\sum_{i \in Q} \sum_{j \in Q} x_{ij} \leq |Q| - 1 \qquad \text{para todos os possíveis subconjuntos } Q \qquad (23)$$

$$\sum_{j=1}^{N} \sum_{i=1}^{M} z_{ij} \leq M\text{-}C \qquad (24)$$

$y_{kj} = 1$ para toda ferramenta k necessária para o processamento da tarefa j (25)

$z_{kj} = 0$ para toda ferramenta k que não é necessária no processamento da tarefa j (26)

$x_{ij} = 0$ ou 1 para todos i e j (27)

$y_{ij} = 0$ ou 1 para todos i e j (28)

$z_{ij} = 0$ ou 1 para todos i e j (29)

Neste modelo (3), a função objetivo minimiza o número máximo de pilhas abertas, enquanto que as restrições (19) e (20) são restrições de atribuição e as restrições (21), de capacidade. As restrições (22) impõem que se a ferramenta k não foi utilizada pela tarefa i e não foi colocada na máquina para o processamento da tarefa j, então a tarefa j não poderá seguir a tarefa i e requerer a ferramenta k ao mesmo tempo. As restrições (23) previnem a formação de ciclos. A restrição (24) impõe que o número de trocas é igual à diferença entre o número total de ferramentas e a capacidade da máquina. As restrições (25) que o impõem que todas as ferramentas necessárias ao processamento da tareja j estejam presentes na máquina quando esta tarefa for processada. As restrições (26) impõem que não há necessidade de se inserir uma nova ferramentas se a ferramenta não é necessária para o processamento. As restrições (27), (28) e (29) são as restrições de integralidade das variáveis.

Na Tabela 2 comparamos os modelos 1, 2 e 3, em termos de variáveis e restrições.

Tabela 2 – Comparação do número de variáveis e restrições dos modelos 1, 2 e 3

	Número de variáveis	Número de restrições
Modelo 1	$2M^2+1$	$3M + M^2$
Modelo 2	$2MN + N^2-M+1$	$3N - M + MN + MN^2 + 1$
Modelo 3	$2MN + (N+2)^2+1$	$3N+MN^2+2^N+3$

Da Tabela 2 podemos observar que existem ganhos em termos de número de variáveis e restrições no caso onde $M \leq N$. Mesmo para vários casos onde $M > N$ espera-se um bom desempenho do modelo 1 devido ao reduzido número de restrições em comparação aos demais

modelos. Além disso, a resolução do problema de atribuição para definir uma ordem de finalização de um item deve ser facilitada pelo fato de que quando $M > N$, deverão ocorrer pelo menos $M - N$ oportunidades onde nenhum novo padrão precisará ser seqüenciado para se finalizar algum item, ou seja, deveremos ter pelo menos $M - N$ oportunidades onde mais de um item será finalizado conjuntamente com algum outro item. Isto pode ser observado pelo fato de que teremos uma ordenação de finalização de M itens, mas apenas N padrões. Assim, a atribuição de ordem que realmente precisa ser descoberta seria, na verdade, uma equivalente a N itens e não uma de M itens. A facilidade de resolução pelo modelo (1) para este caso precisa ainda ser verificada experimentalmente.

4. REDUÇÕES E REFINAMENTOS AO MODELO 1

No modelo (1) proposto, algumas restrições do conjunto (2) não precisam ser incorporadas pois não serão ativas. Caso se tenha um limitante inferior para o valor ótimo C^* do problema, podemos escrever as restrições (2) apenas até o momento quando o lado direito tomar o valor M pois o lado esquerdo nunca poderá superar este valor.

Considere o exemplar da Tabela 1. Já vimos que para finalizar o item 1 abriremos 5 pilhas e para finalizar o item 2 abriremos 3 pilhas. De maneira similar, podemos concluir que para finalizar o item 3, abriremos 4 pilhas (itens 1, 3, 4 e 5); para finalizar o item 4, abriremos 4 pilhas (itens 1, 2, 3 e 4) e para finalizar o item 5 abriremos 3 pilhas (itens 1, 3 e 5). Conseqüentemente, um limitante inferior para C^* é 3 e, as restrições (2) correspondentes a este caso, precisam apenas considerar n variando de 1 até 3 (ao invés de 1 até 5).

É possível que se considere algum procedimento resolução para o modelo (1) onde se relaxe algumas das variáveis inteiras. Neste caso, talvez seja conveniente escolher um valor de K mais "apertado" nas restrições (3). Ao invés de se considerar um único valor de K, pode-se considerar um para cada item. Neste caso, podemos substituir K por uma matriz diagonal $\mathbf{K}$ onde o elemento K_{ii} da diagonal desta matriz é dada por $\sum_{j=1}^{M} \boldsymbol{S}_{ij}$, ou seja, o maior valor possível para a somatória dos elementos i dos vetores $\boldsymbol{S}_j, j = 1,..., M$.

Na parte de experimentos computacionais com o modelo (1) alguns testes devem ser conduzidos para se verificar se a introdução de restrições adicionais ao modelo pode acelerar a sua resolução. Por exemplo, no caso de termos algum $\boldsymbol{S}_i = \boldsymbol{S}_j$, $i \neq j$, podemos fixar a finalização de i imediatamente após a finalização de j (ou vice-versa), uma vez que todas as pilhas para finalizar o outro já estariam abertas de qualquer forma. Portanto, pode-se incluir esta restrição ao modelo. Também, se tivermos algum $\boldsymbol{S}_i < \boldsymbol{S}_j$, podemos impor que a finalização de i deve ser anterior à finalização de j, uma vez que para se tentar minimizar o número de pilhas abertas não estaríamos incorrendo em nenhuma penalidade uma vez que todas as pilhas que seriam abertas com i também terão que ser abertas de qualquer forma quando finalizarmos j. A introdução destas restrições aumenta o número de restrições mas diminui o espaço de busca de soluções. Assim, se o método de solução é baseado em algum processo enumerativo, o esforço computacional em cada iteração aumenta mas o número de iterações deve diminuir.

5. TESTES COMPUTACIONAIS

Os modelos 1 e 2 foram comparados utilizando-se 20 exemplos retirados do artigo de Yanasse e Pinto [3]. Os modelos foram resolvidos utilizando-se a versão 6.5 do CPLEX em uma estação de trabalho Sun UltraSPARC II modelo Ultra 30, 296 MHz, 383MB, sistema operacional versão 5.6. Os resultados obtidos estão apresentados na Tabela 3.

Como pode ser observado, o desempenho do modelo 1 foi superior ao do modelo 2 em 18 dos 20 exemplares testados. As reduções observadas nos tempos computacionais foram

significativas na grande maioria dos casos. Apenas nos exemplares 15 e 18 o desempenho observado do modelo 2 foi bem superior aos do observado com o modelo 1.

Para o caso dos exemplares 15 e 18, testamos a introdução de restrições adicionais no modelo 1 onde um pré-ordenamento da finalização de alguns itens é imposto em virtude de dominâncias. Com isso, os tempos computacionais sofreram redução significativa, passando para 1,05 e 0,07 minutos, respectivamente, para os exemplares 15 e 18.

Tabela 3 – Resolução do MOSP usando os modelos 1, 2 e 3 (tempos em minutos).

Exemplo	M	N	Ótimo	Modelo 1	Modelo 2	Modelo 3
1	9	10	3	**0,03**	4,86	> 60
2	9	10	7	**0,01**	18,52	> 60
3	9	10	8	**0,01**	12,14	2,87
4	10	9	7	**0,03**	1,49	0,60
5	9	10	4	0,05	0,19	**0,02**
6	10	8	8	**0,02**	1,11	0,94
7	10	10	4	**0,07**	0,21	1,41
8	9	10	4	**0,04**	2,69	4,94
9	10	9	8	**0,02**	15,58	> 60
10	9	10	4	**0,03**	0,19	0,04
11	10	10	6	**0,05**	> 60	> 60
12	8	10	5	**0,02**	3,59	0,18
13	10	10	8	**0,02**	2,15	0,12
14	10	9	3	0,25	1,14	**0,06**
15	10	8	4	51,38	**1,84**	> 60
16	9	10	6	**0,02**	1,01	1,15
17	8	10	7	**0,01**	0,62	1,04
18	10	8	5	0,66	0,06	**0,01**
19	10	9	9	**0,02**	0,46	**0,02**
20	9	10	3	**0,04**	0,26	0,93

Portanto, apenas no exemplar 18 o modelo 1 teve um desempenho computacional ligeiramente inferior ao modelo 2. Comparando-se o modelo 1 com o modelo 3, apenas em 3 exemplares o desempenho do modelo 3 foi superior. Cabe observar, entretanto, a grande variância de tempo na resolução dos exemplares observada com o modelo 3.

Cabe observar que nos testes computacionais realizados com o modelo 3 não se consideram todas as restrições que evitam ciclos. Restrições que evitam ciclos são introduzidas na medida que necessárias, ou seja, quando um ciclo é observado, introduz-se uma restrição que evita este ciclo.

5. CONSIDERAÇÕES FINAIS

Neste trabalho, apresentamos uma nova proposta para modelar o MOSP. Esta formulação é fundamentalmente diferente de outras formulações propostas anteriormente na literatura baseadas no MTSP. Em testes computacionais realizados com os modelos propostos anteriormente para o MTSP (vide Pinto e Yanasse [6]), pelo menos para problemas considerados pequenos, o desempenho computacional observado foi inferior comparado com o de métodos de enumeração desenvolvidos para resolução deste problema (vide Laporte, Salazar e Semet [4]).

Com a nova formulação proposta, o desempenho computacional observado nos testes computacionais realizados mostrou-se, de uma maneira geral, superior em relação a outras formulações propostas anteriormente na literatura.

A introdução de restrições adicionais ao modelo para limitar o espaço de busca mostrou-se eficiente para acelerar a sua resolução. Apesar do aumento do número de restrições, a limitação do espaço de busca decorrente parece mais do que compensar o esforço adicional computacional de se manipular matrizes de maior porte.

RECONHECIMENTO

Este trabalho tem apoio financeiro da FAPESP (Fundação de Amparo à Pesquisa do Estado de São Paulo) e do CNPq (Conselho Nacional de Desenvolvimento Científico e Tecnológico).

6. REFERÊNCIAS

1] H. H. Yanasse "*On a pattern sequencing problem to minimize the maximum number of open stack*". European Journal Operations Research **100** (1997), 454-563.

[2] C. S. Tang. e E. V. Denardo, Models Arising from a Flexible Manufacturing Machine, Part I: Minimization of the Number of Tool Switches, *Operations Research* **36** (1988), 767-777.

[3] H.H. Yanasse e M.J. Pinto, The minimization of tool switches problem as a network flow problem with side constraints. Em: XXXIV SBPO – Simpósio Brasileiro de Pesquisa Operacional, realizado no Rio de Janeiro, RJ, 8 a 11 de outubro de 2002. Livro de Resumos do XXXIV SBPO, SPOLM'2002, p. 86. Anais do XXXIV Simpósio Brasileiro de Pesquisa Operacional, em CD-ROM, ref. arq0223.

[4] G. Laporte, J.J. Salazar e F. Semet, Exact Algorithms for the Job Sequencing and Tool Switching Problem (2002). Disponível em <www.gerad.ca/es/publicaciones/ cahiers_ chrono.php>.

[5] H.H. Yanasse, A transformation for solving a pattern sequencing problem in the wood cut industry. Pesquisa Operacional 17(1):57-70, 1997.

[6] M. J. Pinto e H. H. Yanasse, O Problema de Minimização de Troca de Ferramentas: propostas para sua resolução. Em: XXXIII SBPO – Simpósio Brasileiro de Pesquisa Operacional, Publicado em CD-ROM, 1410-1419, 2001.