

ABORDAGENS PARA OTIMIZAÇÃO INTEGRADA DOS PROBLEMAS DE GERAÇÃO E SEQUENCIAMENTO DE PADRÕES DE CORTE: CASO UNIDIMENSIONAL

Gisele C. F. Pileggi

Curso de Eng. Produção / Univ. Presbiteriana Mackenzie
São Paulo – SP
giselepileggi@mackenzie.com.br

Reinaldo Morabito *

Dep. Eng. Produção / Univ. Federal de São Carlos
São Carlos – SP
morabito@power.ufscar.br

Marcos Nereu Arenales

Dep. Matemática Aplicada e Estatística / Univ. de São Paulo
São Carlos – SP
arenales@icmc.usp.br

* *Corresponding author* / autor para quem as correspondências devem ser encaminhadas

Recebido em 06/2004; aceito em 08/2005

Received June 2004; accepted August 2005

Resumo

O problema de geração de padrões de corte (ou problema de corte de estoque) consiste em determinar o conjunto de padrões em que unidades demandadas (itens) são cortadas de unidades maiores (objetos) tal que, por exemplo, o custo ou a perda de material é minimizado. O problema de seqüenciamento de padrões de corte consiste em determinar a seqüência em que os padrões são cortados tal que, por exemplo, o número máximo de pilhas abertas (pilhas de itens com demanda apenas parcialmente produzida, que ainda serão cortados de um ou mais padrões seguintes nessa seqüência) é minimizado. Em geral, uma boa solução para o problema de geração de padrões não corresponde a uma boa solução para o problema de seqüenciamento de padrões e vice-versa. Esses dois problemas são freqüentemente resolvidos, tanto na prática como na literatura, de forma independente e sucessiva. Este trabalho apresenta três abordagens heurísticas para resolver de forma integrada os problemas de geração e seqüenciamento de padrões, considerando o *trade-off* entre os objetivos envolvidos. Embora essas abordagens possam ser aplicadas para problemas de corte e empacotamento de qualquer dimensão, neste trabalho elas são analisadas e comparadas apenas para o caso de corte unidimensional.

Palavras-chave: problemas de corte e empacotamento; geração de padrões de corte; seqüenciamento de padrões de corte.

Abstract

The cutting pattern generating problem (or cutting stock problem) consists in determining the set of patterns in which ordered units (items) are cut from larger units (objects) so that, for example, the cost or waste of material is minimized. The cutting pattern sequencing problem consists in determining the sequence in which the patterns are cut so that, for example, the maximum number of open stacks (stacks of items with demand only partially produced and that will be cut in the next cutting patterns of the sequence) is minimized. In general a good solution for the pattern generating problem does not correspond to a good solution for the pattern sequencing problem and vice-versa. These problems are frequently solved, both in practice and in the literature, in an independent and successive way. This work presents three heuristic approaches to deal with the integrated pattern generating and sequencing problem, considering the trade-off between the objectives involved. Although the approaches can be applied to cutting and packing problems of any dimension, in this work they are analyzed and compared only for the one-dimensional cutting case.

Keywords: cutting and packing problems; cutting pattern generation; cutting pattern sequencing.

1. Introdução

O problema de geração de padrões de corte (ou problema de corte de estoque) consiste em cortar objetos em estoque de tamanhos e quantidades conhecidos, para atender uma demanda de itens de tamanhos e quantidades especificados pelos clientes, de forma a otimizar algum critério, por exemplo, minimizar o custo ou a perda do material cortado. Os objetos são cortados de acordo com padrões de corte que definem maneiras diferentes de arranjar itens dentro de objetos. Os problemas de corte e empacotamento, em geral, pertencem à classe de problemas NP-difíceis (Dowland & Dowland, 1992; Dyckhoff & Finke, 1992).

Em alguns processos industriais de corte, os padrões de corte não podem ser processados em qualquer ordem. Por exemplo, no caso de fábricas que possuem limitação de espaço físico ao redor do equipamento de corte, ou no caso desse equipamento possuir estações de descarregamento automáticas. Nesses casos é preciso definir uma seqüência na qual os padrões são processados a fim de otimizar algum critério, ou seja, é preciso resolver um problema de seqüenciamento de padrões de corte. A seqüência na qual os padrões são processados também interfere nos níveis de estoque intermediário, no manuseio de materiais, nos prazos de entrega, entre outros, uma vez que os padrões podem conter diferentes conjuntos de itens. O problema de seqüenciamento de padrões depende do problema de geração de padrões e vice-versa, o problema de geração depende do problema de seqüenciamento, pois, os padrões a serem gerados precisam ser seqüenciáveis do ponto de vista de algum critério. Assim como os problemas de corte, os problemas de seqüenciamento também pertencem à classe de problemas NP-difíceis (Linhares & Yanasse, 2002).

Em geral, esses dois problemas são tratados, tanto na prática como na literatura, de forma independente e sucessiva. Em uma primeira etapa, resolve-se o problema de geração de padrões e, a partir da solução obtida, resolve-se o problema de seqüenciamento de padrões. Entretanto, com freqüência, uma boa solução para o problema de geração (p.e., um conjunto de padrões com baixa perda de material) não resulta numa boa solução para o problema de seqüenciamento (p.e., uma seqüência desses padrões com pequeno número máximo de pilhas abertas de itens) e vice-versa, o que evidencia um conflito entre os objetivos dos dois problemas.

O presente trabalho apresenta abordagens para resolver de forma integrada os problemas de geração e seqüenciamento de padrões, que consideram a dependência existente entre eles. Convém salientar que poucos são os trabalhos encontrados na literatura que tratam dos problemas de geração e de seqüenciamento de forma integrada, por exemplo, Dyson & Gregory (1974) e Madsen (1979, 1988) resolvem esses problemas de forma seqüencial, enquanto Armbruster (2002) e Pinto (2004) tratam esses problemas de forma integrada. Além disso, devido à complexidade dos problemas envolvidos, existe uma grande motivação para a busca de algoritmos mais eficientes. Abordagens que resolvam de forma integrada os problemas de geração e seqüenciamento podem ser aplicadas, por exemplo, no processo de fabricação de tubos de aço, papel, móveis, vidros planos, chapas de fibra de madeira reconstituída, chapas de fibra de vidro para placas de circuito impresso, entre outros.

Duas estratégias básicas podem ser exploradas para resolver esses dois problemas de forma integrada. Uma consiste em combinar os problemas de geração e seqüenciamento em um único problema e otimizar uma função objetivo que agregue os objetivos desses dois problemas. Na outra, a função objetivo de um dos problemas é otimizada considerando o objetivo do segundo problema como uma restrição. Acredita-se que combinar dois problemas difíceis de serem resolvidos (NP-difíceis) num único problema também resulte

num problema difícil. Para a segunda estratégia, duas alternativas podem ser exploradas. Uma delas consiste em otimizar a função objetivo do problema de seqüenciamento de padrões e considerar o objetivo do problema de geração de padrões como uma restrição (problema P1, seção 4). A outra otimiza a função objetivo do problema de geração e considera o objetivo do problema de seqüenciamento como uma restrição (problema P2, seção 4). Em ambos os casos critérios de tolerância podem ser estabelecidos a fim de que as soluções obtidas estejam dentro de limites especificados pela empresa. Essas alternativas são úteis para gerar curvas de *trade-off* (ou curvas de eficiência) entre, por exemplo, a perda de material e o número máximo de pilhas abertas.

Neste trabalho três abordagens heurísticas inspiradas no problema P2 são apresentadas para resolver os problemas de geração e seqüenciamento de forma integrada, baseadas em procedimentos iterativos heurísticos. O enfoque é em abordagens simples e que possam ser facilmente aplicadas na prática para dar apoio às decisões. Embora essas abordagens possam ser aplicadas para problemas de corte ou empacotamento de qualquer dimensão, neste trabalho as abordagens são apresentadas e comparadas apenas para o caso de corte unidimensional. O caso bidimensional será objeto de um próximo trabalho ainda a ser compilado.

Este artigo está organizado da seguinte maneira: nas seções 2 e 3 os problemas de geração e seqüenciamento de padrões são brevemente revisados. Na seção 4 é discutida a dependência entre esses dois problemas e as três abordagens de solução são apresentadas. Na seção 5, o desempenho das abordagens é comparado, analisando-se os resultados computacionais obtidos para exemplos de corte unidimensional. Finalmente, na seção 6, as conclusões deste estudo e perspectivas para pesquisa futura estão descritas.

2. Geração de Padrões de Corte

Os trabalhos de Gilmore & Gomory (1961,1963) impulsionaram intensa pesquisa nos problemas de corte de estoque (neste texto referido por geração de padrões de corte), quando introduziram a técnica de geração de colunas e resolveram de forma razoável problemas grandes como ocorre na prática. Centenas de artigos que tratam dos problemas de geração de padrões podem ser encontrados na literatura, conforme indicam os artigos de revisão e edições especiais em Hinxman (1980), Dyckhoff & Waescher (1990), Dyckhoff & Finke (1992), Dowsland & Dowsland (1992), Sweeney & Paternoster (1992), Morabito & Arenales (1992), Bischoff & Waescher (1995), Dyckhoff *et al.* (1997), Arenales *et al.* (1999), Wang & Waescher (2002), Hifi (2002), Lodi *et al.* (2002), Oliveira & Waescher (2005) e SICUP (2005). Considere a seguinte notação para a formulação matemática:

- p número de padrões de corte
- m número de tipos de itens
- a_{ij} número de itens do tipo i no padrão de corte j
- b_i número demandado de itens do tipo i
- c_j perda de material associada ao padrão de corte j
- x_j número de vezes que o padrão de corte j é utilizado

Por simplicidade, considere apenas um tipo de objeto em estoque, com comprimento L . Note que para cada padrão de corte j corresponde um vetor $a_j = (a_{1j}, a_{2j}, \dots, a_{mj})$ satisfazendo

$\sum_{i=1}^m l_i a_{ij} \leq L$, em que l_i é o comprimento do item do tipo i . Suponha que todos os p padrões de corte sejam conhecidos *a priori* e que os objetos estejam disponíveis em quantidades ilimitadas. O modelo matemático de geração de padrões de corte unidimensional pode ser formulado da seguinte forma:

$$\text{Minimizar } f(x) = \sum_{j=1}^p c_j x_j \quad (1)$$

$$\text{Sujeito a: } \sum_{j=1}^p a_{ij} x_j \geq b_i \quad i = 1, \dots, m \quad (2)$$

$$x_j \geq 0 \text{ e inteiro } \quad j = 1, \dots, p \quad (3)$$

A minimização da função $f(x)$ em (1) resulta no conjunto de padrões de corte de perda mínima, na qual $c_j = L - \sum_{i=1}^m l_i a_{ij}$. As restrições (2) garantem que toda a demanda seja atendida e as restrições (3) garantem que o número de vezes que cada padrão j é repetido seja um número inteiro e não-negativo. Na prática o número de padrões de corte p pode ser extremamente grande, o que inviabiliza a determinação explícita de todos os padrões. A exigência de integralidade sobre as variáveis x_j e o número extremamente grande de padrões de corte contribuem para tornar o problema de geração de padrões difícil de ser resolvido computacionalmente.

Nos problemas práticos, com demandas razoavelmente grandes em relação aos elementos a_{ij} (i.e., $b_i \gg L/l_i$, $i = 1, 2, \dots, m$), o número de vezes que os padrões devem ser utilizados em geral é grande. Nestes casos, Gilmore & Gomory (1961, 1963) sugeriram relaxar a restrição de integralidade sobre as variáveis x_j , argumentando que o simples arredondamento da solução ótima do problema linear constitui uma boa solução para o problema inteiro, e utilizar o método simplex com um procedimento de geração de colunas, resumidamente descrito como:

PASSO 1: Dada uma base $B=[a_1, \dots, a_m]$ (colunas do problema (1)-(3)), determine $\pi = c_B \cdot B^{-1}$, o vetor multiplicador simplex.

PASSO 2: Determine o menor custo relativo, isto é resolva o problema da mochila:

$$\hat{c} = \text{Min} (L - \sum_{i=1}^m l_i \alpha_i) - (\sum_{i=1}^m \pi_i \alpha_i) \text{ sujeito a: } \sum_{i=1}^m l_i \alpha_i \leq L, 0 \leq \alpha_i \leq b_i \text{ inteiro.}$$

PASSO 3: Se $\hat{c} \geq 0$, então a base B é ótima e pare; senão a coluna $a=(\alpha_1, \dots, \alpha_m)$ entra na base (uma coluna básica é escolhida para sair da base no passo usual do método simplex). Retorne ao passo 1 com a nova base B .

Note que este método permite a geração explícita de apenas uma coluna (padrão) durante cada iteração simplex, mantendo implicitamente as demais. Para problemas unidimensionais, isto em geral corresponde a resolver um problema da mochila. Métodos para resolver este problema podem ser encontrados em Gilmore & Gomory (1963), Martello & Toth (1990), Pisinger (2000) e Soma & Toth (2002). As *Abordagens 1 e 3* (descritas na seção 4) utilizam o método simplex com geração de colunas para resolver o problema (1)-(3). Heurísticas para tratar o arredondamento de soluções em casos em que o simples arredondamento não constitui uma boa solução podem ser encontradas em Haessler (1980), Vance *et al.* (1994), Waescher & Gau (1996), Riehme *et al.* (1996), Pinto (1999) e Poldi & Arenales (2003).

Uma alternativa ao método simplex com geração de colunas para resolver aproximadamente o problema (1)-(3) é utilizar heurísticas construtivas gulosas (Hinixman, 1980; Dyckhoff & Finke, 1992; Dowsland & Dowsland, 1992; Sweeney & Paternoster, 1992; Silveira & Morabito, 2002; Belluzzo & Morabito, 2005; Poldi & Arenales, 2005). A estratégia, que consiste na redução do problema pela geração sucessiva de bons padrões de corte (*repeated exhaustion reduction*; Hinixman, 1980), é bastante utilizada na prática. Este procedimento, utilizado neste trabalho pela *Abordagem 2* (descrita na seção 4), é apresentado a seguir:

PASSO 1: Inicialização: $k=1$ (k é o número do padrão em construção).

Para cada item i , defina um valor v_i (tipicamente $v_i = l_i$) e a demanda residual

$$\hat{b}_i = b_i$$

PASSO 2: Gere o padrão de corte mais valioso, isto é resolva o problema da mochila:

$$\text{Max } \sum_{i=1}^m v_i \alpha_i \text{ sujeito a: } \sum_{i=1}^m l_i \alpha_i \leq L, 0 \leq \alpha_i \leq \hat{b}_i \text{ inteiro.}$$

PASSO 3: Utilize o padrão mais valioso (passo 2) tanto quanto possível:

$$x_k = \min \left\{ \left\lfloor \frac{\hat{b}_i}{\alpha_i} \right\rfloor, \alpha_i > 0, i = 1 \dots m \right\}$$

PASSO 4: Atualize a demanda residual: $\hat{b}_i = \hat{b}_i - x_k \alpha_i, i = 1, \dots, m$.

PASSO 5: Se $\hat{b}_i = 0, i = 1, \dots, m$, então a demanda é satisfeita, pare. Senão faça $k=k+1$ e retorne ao passo 2.

Note que este procedimento não tem garantia de encontrar uma solução ótima global, uma vez que ele determina uma seqüência de ótimos locais: em cada iteração determina o padrão mais valioso para a iteração. Uma vantagem atribuída a essa abordagem é que, na prática, novos pedidos podem ser incorporados à carteira de pedidos dentro do horizonte de planejamento. A incorporação destes pedidos pode resultar em uma solução global com uma menor perda, uma vez que novos itens ficam disponíveis para serem combinados na geração dos padrões. Por outro lado, uma desvantagem deste procedimento é que, apesar de gerar padrões com baixa perda nas primeiras iterações, à medida que avança, padrões com perdas cada vez maiores tendem a ser gerados, o que compromete a solução global do problema. Isso em geral ocorre, pois no início existe uma variedade maior de tipos de itens, incluindo itens com dimensões pequenas, que são fáceis de serem arranjados em pequenos espaços.

3. Seqüenciamento de Padrões de Corte

Conforme mencionado, a ordem na qual os padrões de corte são produzidos pode ser importante em alguns sistemas produtivos. Nesses casos, antes de efetivar o corte dos objetos, é preciso determinar em qual seqüência os padrões de corte serão processados a fim de otimizar algum critério, ou seja, é preciso resolver um problema de seqüenciamento de padrões. Alguns trabalhos tratando o problema de seqüenciamento podem ser encontrados em Dyson & Gregory (1974), Madsen (1979, 1988), Lins (1989), Yuen (1991, 1995), Yanasse (1996a, 1996b, 1997), Faggioli & Bentivoglio (1998), Foerster & Waescher (1998), Fink & Voß (1999), Becceneri (1999), Ashikaga (2001) e Pinto (2004). As abordagens utilizadas por esses autores para resolver o problema de seqüenciamento baseiam-se em

procedimentos heurísticos, metaheurísticas (*simulated annealing* e busca tabu), relaxação Lagrangeana, uso de teoria de grafos e fluxos em rede.

Os itens de um mesmo tipo cortados são arranjados em uma pilha que é considerada aberta enquanto a produção daquele tipo não for terminada. Durante o seqüenciamento dos padrões de corte, diferentes objetivos podem ser de interesse, como, por exemplo, a minimização do número máximo de pilhas abertas de itens, *MOSP* (do inglês *Minimization of Open Stack Problem*), a minimização do espalhamento de ordens ou a minimização do número de descontinuidades. Neste trabalho considera-se o *MOSP*. Suponha que uma pilha só possa ser removida após um padrão ser completamente cortado e, desta forma, o número máximo de pilhas abertas ocorre exatamente após o processamento de um padrão e antes que qualquer pilha completa seja removida. Na maioria dos trabalhos encontrados na literatura, considera-se que pilhas acabadas só possam ser removidas após o término do corte do padrão. Pinto (2004) admite a possibilidade da remoção de pilhas acabadas durante o corte do padrão.

O *MOSP* ocorre, por exemplo, em situações práticas nas quais existe limitação de espaço físico, ou seja, quando o número de pilhas de itens que podem permanecer abertas ao redor da máquina de corte é limitado, como, por exemplo, em certas fábricas de móveis. Como cada tipo de item abre uma nova pilha que só é removida quando a demanda do mesmo é totalmente atendida, dependendo da seqüência escolhida para o processamento dos padrões de corte, pode ser preciso que algumas pilhas sejam temporariamente removidas e posteriormente trazidas de volta para serem completadas, o que poderá acarretar em um maior manuseio de materiais, maior dificuldade para administração dos itens e um comprometimento da produção final. O *MOSP* também aparece em processos industriais com equipamentos de corte com número limitado de estações de descarregamento de itens, como é o caso de certas fábricas de chapas de fibra de madeira reconstituída (Morabito & Garcia, 1998a, 1998b; Belluzzo & Morabito, 2005).

Para resolver o *MOSP*, diferentes abordagens podem ser utilizadas, como por exemplo, aquelas nas referências citadas no início desta seção. Neste trabalho escolheu-se a heurística 3 de Yuen (1995) devido à sua simplicidade e superioridade comparativamente às demais heurísticas propostas por esse autor com relação à qualidade das soluções obtidas e ao tempo computacional, e também devido à facilidade de sua implementação computacional. Essa heurística é baseada na adequação de um padrão às pilhas já abertas. Quanto maior o número de itens em comum, melhor a adequação de um padrão com as pilhas atuais; por outro lado, quanto maior o número de itens novos introduzidos, pior a adequação do padrão com as pilhas atuais. Para medir a adequação, considere C_j o número de tipos de itens em comum entre o padrão j e as pilhas abertas, e N_j o número de tipos de itens no padrão j que não estão incluídos nas pilhas abertas. A adequação de um padrão j , M_j , é definida como o número de itens em comum ao padrão j e às pilhas abertas, menos o número de tipos de itens novos contidos no padrão j , ou seja, $M_j = C_j - N_j$. A heurística 3 está descrita a seguir.

Seja P o conjunto inicial de padrões de corte.

PASSO 1: Escolha $j \in P$ tal que $M_j = \max \{M_i, i \in P\}$. Se houver empate na escolha, selecione aquele com o menor número de tipos de itens, isto é, se $C_j - N_j = C_k - N_k$ e $N_j < N_k$ então selecione j . Caso ainda haja empate, selecione o primeiro.

PASSO 2: Faça $P = P - \{j\}$. Se $P = \emptyset$ então pare; senão repita o passo 1.

Para ilustração desta heurística considere o conjunto de dados da Tabela 1 (Yuen, 1991) em que cada linha corresponde a um padrão de corte e cada coluna, a um item. Cada célula marcada com X indica a presença do item no padrão. Na Tabela 2 a seqüência obtida com a aplicação da heurística 3 sobre os dados da Tabela 1 é apresentada. O padrão 1 foi utilizado como padrão inicial. Nesta tabela a primeira coluna corresponde aos padrões que foram seqüenciados, a segunda coluna mostra o número de pilhas abertas e a terceira coluna quais as pilhas que foram abertas. Aplicando a heurística 3 sobre os dados da Tabela 1, o número máximo de pilhas abertas obtido é 4, conforme a Tabela 2. Na seção seguinte é descrita a integração entre os problemas de geração e seqüenciamento de padrões de corte.

Tabela 1 – Exemplo de um conjunto de padrões de corte (Yuen, 1991).

Padrão	Item							
	1	2	3	4	5	6	7	8
1			X		X			
2	X	X			X	X		
3	X		X					
4				X	X			X
5			X				X	

Tabela 2 – Seqüência obtida com a aplicação da heurística 3 sobre os dados da Tabela 1.

Padrões seqüenciados	Nº de pilhas abertas	Pilhas abertas
1	2	3 e 5
1-3	3	1, 3, e 5
1-3-5	4	1, 3, 5 e 7
1-3-5-2	4	1, 2, 5 e 6
1-3-5-2-4	3	4, 5 e 8

4. Trade-off entre a Perda e o Número Máximo de Pilhas Abertas

Como os problemas de geração e seqüenciamento de padrões são difíceis de serem resolvidos de forma ótima (NP-difíceis), acredita-se que combiná-los num único problema também resulte em um problema difícil. Suponha um limite suficientemente grande para o número máximo de pilhas abertas permitido. Neste caso não existe nenhuma dificuldade para resolver o problema de seqüenciamento de padrões. Desta forma, o problema integrado reduz-se ao problema de geração de padrões, que é um problema difícil (Pinto, 2004). Considere, para efeito de ilustração, que o objetivo do problema de geração seja a minimização da perda (pe) gerada pelo corte dos padrões e o objetivo do problema de seqüenciamento seja a minimização do número máximo de pilhas abertas (pi) (*MOSP*). Como na realidade existem dois problemas interdependentes com dois objetivos de importância equivalente (e, na maioria das vezes, conflitantes) a serem resolvidos, trata-se, portanto, de um problema bi-objetivo, no qual existe um *trade-off* entre os dois critérios a serem otimizados.

Tal problema bi-objetivo pode ser formulado matematicamente da seguinte maneira. Sejam a_1, a_2, \dots, a_p os possíveis p padrões de corte (colunas) para o problema (1)-(3) e X o espaço de

soluções factíveis do problema (1)-(3), ou seja, $x = (x_1, x_2, \dots, x_p) \in X$ se e somente se satisfaz às restrições (2) e (3). Note que se $x_j > 0$ então a solução x utiliza o padrão a_j . Considere que x utilize n padrões e seja $s(x) = (a_{[1]}, a_{[2]}, \dots, a_{[n]})$ uma seqüência destes n padrões ($[k]=j$ indica que o padrão a_j ocupa a k -ésima posição desta seqüência). Seja $S(x)$ o conjunto de todas as possíveis seqüências $s(x)$ e $g(s(x))$ o número máximo de pilhas abertas na seqüência $s(x)$. Logo, chamando $pe = f(x)$ e $pi = g(s(x))$, tem-se o seguinte problema bi-objetivo:

$$(P0) \quad \begin{aligned} \min \quad & z = (pe, pi) = (f(x), g(s(x))) \\ & x = (x_1, x_2, \dots, x_p) \in X, s(x) \in S(x) \end{aligned}$$

Em problemas mono-objetivos, como o problema de corte (1)-(3), os valores da função objetivo são completamente ordenados, isto é, dados quaisquer dois pontos $x, y \in X$, é sempre verdade que: (i) $f(x) \leq f(y)$ ou (ii) $f(x) \geq f(y)$. Em problemas bi-objetivos (como P0) com objetivos conflitantes (i.e., melhorar um objetivo pode causar a piora de outro e vice-versa), o espaço objetivo (i.e., os valores dos objetivos) é parcialmente ordenado, isto é, dados quaisquer dois pontos x e $y \in X$, existem quatro casos: (i) $f(x) \leq f(y)$ e $g(s(x)) \leq g(s(y))$, (ii) $f(x) \geq f(y)$ e $g(s(x)) \geq g(s(y))$, (iii) $f(x) \leq f(y)$ e $g(s(x)) \geq g(s(y))$ ou (iv) $f(x) \geq f(y)$ e $g(s(x)) \leq g(s(y))$. No caso (i), diz-se que x domina y (analogamente, y domina x no caso (ii)).

Uma solução $x^* \in X$ é uma solução eficiente se não existe qualquer outra solução $x \in X$ que domine x^* . O conjunto de todas as soluções eficientes define uma fronteira ótima ou curva eficiente (curva de *trade-off*), conforme ilustrado na Figura 1. Em problemas com objetivos conflitantes, como é o caso do problema P0, é pouco provável que o conjunto das soluções eficientes seja unitário ou contenha poucos pontos. Existem diversos métodos para resolver problemas multi-objetivos (Cohon, 1978; Steuer, 1986). Duas etapas importantes na resolução destes problemas são: a determinação de soluções eficientes (i.e., a fronteira ótima) e a escolha de uma solução eficiente. No presente trabalho um método *a posteriori* é aplicado baseado na minimização de um dos objetivos, sujeito à limitação do outro objetivo (Arroyo, 2002). A idéia deste método é que, variando-se esta limitação, é possível gerar os pontos da fronteira ótima. Por exemplo, pode-se minimizar o número máximo de pilhas abertas (pi), sujeito à uma limitação (diga-se, $p\hat{e}$) para a perda de material (pe), ou seja:

$$(P1) \quad \begin{aligned} \min \quad & pi = g(s(x)) \\ & pe = f(x) \leq p\hat{e} \\ & x = (x_1, x_2, \dots, x_p) \in X, s(x) \in S(x) \end{aligned}$$

Note que o problema P1 consiste em determinar um conjunto de padrões de corte (x) que, quando seqüenciados, abram o menor número máximo de pilhas (pi) possível, sujeito à restrição de que a perda gerada pelo corte desses padrões (pe) seja menor ou igual à perda tolerada ($p\hat{e}$). Variando-se convenientemente a limitação $p\hat{e}$, é possível gerar os pontos da fronteira eficiente. Alternativamente ao problema P1, pode-se minimizar a perda de material (pe), sujeito à uma limitação (diga-se, $p\hat{i}$) para o número máximo de pilhas abertas (pi), ou seja:

$$(P2) \quad \begin{aligned} \min \quad & pe = f(x) \\ & pi = g(s(x)) \leq p\hat{i} \text{ para algum } s(x) \in S(x) \\ & x = (x_1, x_2, \dots, x_p) \in X \end{aligned}$$

Os pontos desta curva de *trade-off* correspondem às soluções de mínima perda para diferentes valores de número máximo de pilhas abertas ou, inversamente, às soluções de mínimo número máximo de pilhas abertas para diferentes valores de perda de material. Considere as quatro soluções **C**, **D**, **E** e **F**, representadas no gráfico da Figura 1. As soluções **C** e **F** (sobre a curva) dominam a solução **D** (fora da curva), pois a perda e o número máximo de pilhas das soluções **C** e **F** são menores ou iguais à perda e ao número máximo de pilhas da solução **D**. Por outro lado, as soluções **C** e **F** não dominam uma a outra, nem dominam a solução **E** (sobre a curva).

Como o número máximo de pilhas abertas é um número positivo e inteiro, o conjunto de pontos do eixo y (número máximo de pilhas) do gráfico da Figura 1 é discreto e, mais que isso, é finito e definido pelo intervalo $[1, \overline{pi}]$, como é visto mais adiante. Como as soluções precisam satisfazer os critérios de tolerância dos problemas de geração e seqüenciamento de padrões de corte, os valores \hat{pi} e \hat{pe} delimitam a parte da curva que contém as soluções factíveis (do ponto de vista das tolerâncias estabelecidas pelo usuário). Ou seja, os pontos sobre a curva de eficiência contidos nos intervalos $[1, \hat{pi}]$ e $[\underline{pe}, \hat{pe}]$ correspondem às soluções alvo do *trade-off* entre a perda e o número máximo de pilhas abertas. Em particular, o ponto **A** da curva de *trade-off* (Figura 1) pode ser obtido resolvendo o problema:

$$(P3) \quad \begin{aligned} \min \quad & pi = g(s(x)) \\ & pe = f(x) = \underline{pe} \\ & x = (x_1, x_2, \dots, x_p) \in X, s(x) \in S(x). \end{aligned}$$

Para este problema a solução é definida por um conjunto de padrões de corte (x) que, quando seqüenciados, abram o menor número máximo de pilhas abertas (pi) e satisfaçam a restrição de que a perda gerada pelo corte desses padrões (pe) seja igual à perda mínima (\underline{pe}). Este problema é difícil de ser resolvido de forma ótima devido à dificuldade de se examinar todos os conjuntos de padrões de corte do espaço de soluções X com perda igual à perda mínima. Por outro lado, o ponto **B** da curva de *trade-off* pode ser facilmente obtido pela solução homogênea (padrões de corte com apenas um único tipo de item). Na prática, pode ser que não seja possível encontrar uma solução que satisfaça os critérios de tolerância de perda e de número máximo de pilhas (ponto **X** da Figura 1).

Conforme discutido, os demais pontos da fronteira ótima (Figura 1) podem ser obtidos variando-se o valor de \hat{pe} no problema P1, ou variando-se o valor de \hat{pi} no problema P2. De fato, por meio deste procedimento, por exemplo, fazendo-se $\hat{pi} = 1, 2, \dots, \overline{pi}$ no problema P2, obter-se-ia todos os pontos da fronteira eficiente, embora isso seja em geral computacionalmente intratável do ponto de vista prático. Em vista da dificuldade em determinar a fronteira ótima, a procura é pelas soluções que estejam o mais próximo possível desta fronteira (isto é, boas aproximações).

Em resumo, duas alternativas de solução foram descritas acima para tratar os problemas de geração e seqüenciamento de padrões de corte de forma integrada (problema P0). Em uma delas, o problema de seqüenciamento é resolvido considerando a perda tolerada como uma restrição (problema P1) e, na outra, resolve-se o problema de geração considerando como restrição o número máximo de pilhas abertas tolerado (problema P2). Nas seções seguintes

são apresentadas três abordagens, inspiradas no problema P2, para resolver o problema integrado de geração e seqüenciamento de padrões de corte.

4.1 Abordagem 1

A idéia para a *Abordagem 1* surgiu da observação de como os problemas de geração e seqüenciamento de padrões em geral são resolvidos na prática: de forma seqüencial e iterativa. Na primeira etapa o problema de geração de padrões é resolvido e, a partir da solução obtida (conjunto de padrões de corte de perda mínima), resolve-se o problema de seqüenciamento em uma segunda etapa. Se a solução obtida do ponto de vista do problema de seqüenciamento não satisfizer as tolerâncias impostas, interfere-se no problema de geração, a fim de que um novo conjunto de padrões seja gerado na primeira etapa, e assim sucessivamente, até que um certo número de iterações seja atingido ou que uma solução que satisfaça as tolerâncias impostas pelo usuário seja encontrada.

Dyson & Gregory (1974) e Madsen (1988) resolvem o problema de geração e seqüenciamento de padrões de corte utilizando procedimentos em duas etapas. Outros trabalhos abordam o problema de geração de padrões de forma integrada ao processo produtivo. Por exemplo, Hendry *et al.* (1996) propõem um procedimento em duas etapas para resolver problemas de programação da produção e corte. Gramani (2001) e Poltroniere *et al.* (2005) propõem modelos para problemas combinados de dimensionamento de lotes e corte.

A Figura 2 apresenta um esquema ilustrativo da *Abordagem 1*. Após a descrição geral desta abordagem, cada uma das rotinas é descrita detalhadamente. Inicialmente é preciso definir o valor de \hat{p}_i , o número máximo de iterações (*max_iter*), as regras que serão utilizadas para escolha da seqüência (*esc_seq*) e do padrão que será nomeado proibido (*esc_pad*), e o número de iterações que o padrão ficará proibido (*LC*). O problema de geração de padrões (1)-(3) é então resolvido pela *rotina de geração de padrões* sem nenhuma restrição adicional. A solução obtida, $p_e = \underline{pe}$ (padrões de corte com perda mínima) é ótima do ponto de vista de perda. O conjunto de padrões obtido é então seqüenciado na *rotina de seqüenciamento* e uma seqüência é selecionada usando a regra definida por *esc_seq*.

Se $p_i \leq \hat{p}_i$, então o procedimento termina; a solução encontrada é a solução ótima de P2. Caso contrário, isto é, se $p_i > \hat{p}_i$, então a solução precisa ser factibilizada. A estratégia utilizada para factibilizar a solução consiste em alterar o conjunto de padrões gerado na *rotina de geração de padrões*. Para isso, procura-se no conjunto de padrões gerado um padrão para ser eliminado, cuja substituição pelos respectivos padrões homogêneos satisfaça a tolerância do número máximo de pilhas. O primeiro padrão (do conjunto de padrões gerado) é avaliado na *rotina avalia eliminação*. Se o número máximo de pilhas abertas obtido pelo seqüenciamento da solução gerada pela eliminação deste padrão (com posterior factibilização da solução) não satisfizer o número máximo de pilhas tolerado, o segundo padrão é avaliado, e assim sucessivamente, até que $p_i \leq \hat{p}_i$ ou que todos os p padrões tenham sido testados. Se a avaliação de todos os p padrões não satisfizer o número máximo de pilhas tolerado, então um padrão é eliminado utilizando a regra definida por *esc_pad*. Após a factibilização da solução, o novo conjunto de padrões obtido, p' (com $p' \geq p$), é avaliado na *rotina avalia eliminação*. A idéia é factibilizar a solução com a remoção do menor número possível de padrões (Figura 2). Este processo termina quando uma solução com $p_i \leq \hat{p}_i$ for encontrada.

Após a obtenção de uma solução factível (i.e., com $pi \leq p\hat{i}$), se a perda obtida por essa solução for menor ou igual à perda da solução incumbente (i.e., $pe \leq pe_i$), a solução incumbente é atualizada. É verificado se o número máximo de iterações foi atingido. Em caso afirmativo o procedimento termina; caso contrário, a *rotina de inclusão de padrões proibidos* é executada proibindo um ou mais padrões de serem gerados (utilizando a regra definida por *esc_pad*) quando a *rotina de geração de padrões* for novamente executada. O procedimento continua até que o número máximo de iterações seja atingido, ou que uma solução de perda mínima com $pi \leq p\hat{i}$ seja encontrada.

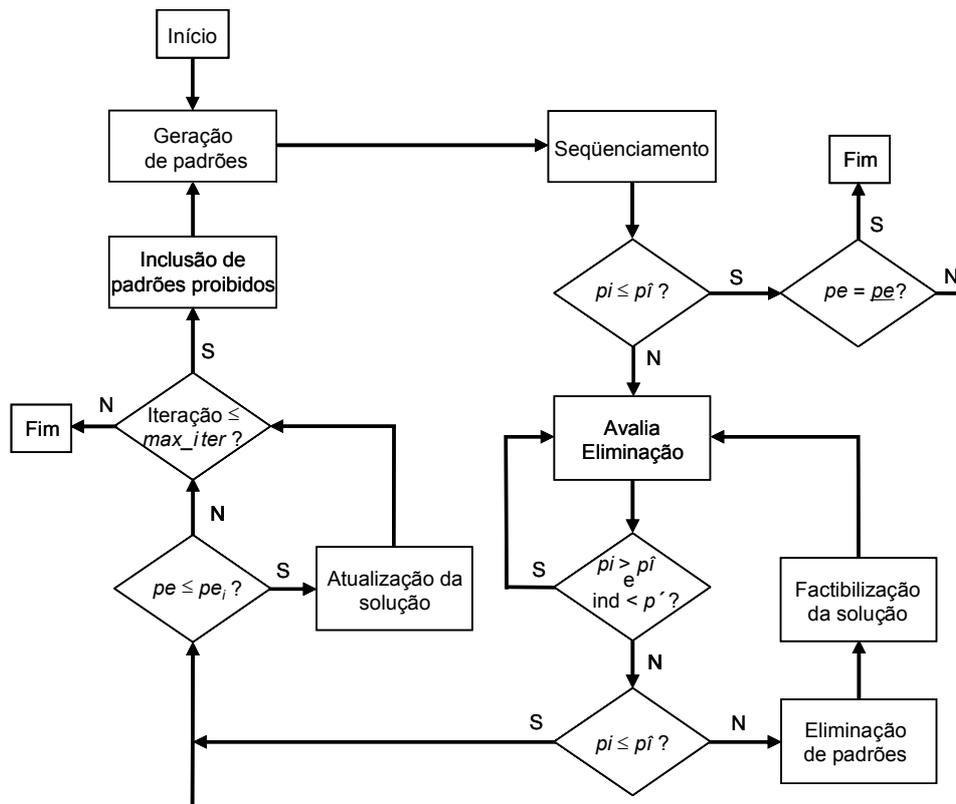


Figura 2 – Esquema ilustrativo da Abordagem 1.

Rotina de geração de padrões: esta rotina resolve o problema de geração de padrões de corte unidimensional (problema (1)-(3), com a relaxação da restrição de integralidade das variáveis), utilizando o método simplex com geração de colunas. O algoritmo lexicográfico de Gilmore & Gomory (1963), que se trata de um método exato de busca em profundidade primeiro, foi utilizado para resolver o problema da mochila envolvido no procedimento de geração de padrões unidimensionais. Uma restrição adicional é que o número máximo de tipos de itens permitidos está limitado em $p\hat{i}$, visto que padrões com mais de $p\hat{i}$ itens são infactíveis, independentemente da seqüência em que os mesmos sejam cortados. Outra

restrição que precisa ser verificada na geração dos padrões é em relação aos padrões proibidos obtidos na *rotina de inclusão de padrões proibidos* (nesta rotina o padrão – ou padrões – é nomeado proibido utilizando a regra *esc_pad*). Com isso, para que ocorra a atualização de um padrão, o mesmo precisa ser mais valioso que o padrão incumbente, não conter mais que $\hat{p}i$ itens e não ser igual a nenhum padrão proibido.

Para satisfazer essas restrições, modificações foram feitas nos passos 3 e 5 do algoritmo de Gilmore & Gomory (1963). O passo 3 deste algoritmo, que faz a atualização da solução incumbente, verifica se o padrão atual é mais valioso que a solução incumbente. Além dessa verificação, o passo 3 aqui modificado verifica se o padrão atual contém um número de tipos de itens menor ou igual a $\hat{p}i$ e se o mesmo não é igual a nenhum padrão proibido. Se essas três condições forem satisfeitas, então a solução é atualizada. O passo 5 do algoritmo avalia a alocação de itens do tipo $s+1$ com a remoção de um item do tipo s . Além dessa avaliação, o passo 5 aqui modificado verifica se o padrão atual é mais valioso que a solução incumbente, se a restrição quanto a $\hat{p}i$ é satisfeita e se o mesmo não é igual a um padrão proibido.

Rotina de seqüenciamento: na rotina de seqüenciamento de padrões o problema de minimizar o número máximo de pilhas abertas (*MOSP*) é resolvido. Esta rotina utiliza como método de seqüenciamento a heurística 3 de Yuen (1995). Dado um padrão inicial, esta heurística determina a seqüência com o menor número máximo de pilhas abertas. Para que a seqüência obtida não seja dependente do padrão inicial, são geradas p seqüências (onde p é o número de padrões de corte) e a seqüência com o menor número máximo de pilhas abertas é selecionada. Se ocorrer empate, isto é, se diferentes seqüências resultarem no mesmo número máximo de pilhas abertas, diferentes regras podem ser utilizadas. Considere o conjunto de padrões da Tabela 1 (seção 3). As seqüências geradas, iniciando com cada um dos 5 padrões (considerando que os padrões iniciais das seqüências são os mesmos da permutação identidade: 1, 2, ..., n), estão descritas na Tabela 3.

Tabela 3 – Seqüências obtidas para os padrões de corte da Tabela 1.

Seqüências	Padrões	Número de pilhas abertas
1	1, 3, 5, 2, 4	2, 3, 4 , 4, 3
2	2, 1, 3, 5, 4	4 , 3, 3, 3, 3
3	3, 1, 5, 2, 4,	2, 3, 4 , 4, 3
4	4, 1, 3, 5, 2	3, 2, 3, 4 , 4
5	5, 1, 3, 2, 4	2, 2, 3, 4 , 3

O número máximo de pilhas abertas é 4 (veja em negrito na Tabela 3), obtido por todas as seqüências; desta forma, o menor número máximo de pilhas abertas é 4. Observe que o menor número máximo de pilhas abertas não poderia ser menor que 4 visto que o padrão 2 possui quatro tipos de itens (Tabela 1). Três regras (*esc_seq*) são definidas para escolher uma seqüência.

A regra denominada *primeira* escolhe a primeira seqüência com o menor número máximo de pilhas abertas. Para o exemplo da Tabela 1 a seqüência escolhida é a seqüência que inicia com o padrão 1 (1, 3, 5, 2, 4). A regra denominada *aleatória* escolhe aleatoriamente uma seqüência, dentre as que empataram (isto é, dentre as que resultaram no menor número

máximo de pilhas abertas). Neste exemplo qualquer uma das cinco seqüências da Tabela 3 pode ser escolhida, visto que o número máximo de pilhas abertas obtido por todas as cinco seqüências é 4. E a regra denominada *mais_demora* escolhe a seqüência que mais demora em atingir o menor número máximo de pilhas abertas. Observe na Tabela 3 que as seqüências 4 e 5 abrem 4 pilhas com o corte do quarto padrão (padrões 5 e 2, respectivamente), ou seja, essas duas seqüências são as que mais demoram em abrir 4 pilhas. Como aconteceu um empate, esta regra escolhe arbitrariamente a primeira delas.

Rotina de atualização da solução: esta rotina armazena a melhor solução encontrada do ponto de vista de perda, dado que o número máximo de pilhas abertas é menor ou igual a \hat{p}_i , isto é, a solução dominante do ponto de vista de perda é armazenada. Por exemplo, considere $\hat{p}_i = 5$ e duas soluções **A** e **B**, com perdas iguais a 2% e 4%, respectivamente, e com números máximos de pilhas abertas iguais a 5 e 3, respectivamente. A solução armazenada neste caso seria a solução **A**. A solução de menor número máximo de pilhas abertas para $pe = \underline{pe}$ também é armazenada.

Rotina de eliminação de padrões: esta rotina determina um ou mais padrões para serem eliminados com o propósito de diminuir o número máximo de pilhas abertas. Diferentes regras (*esc_pad*) podem ser utilizadas na escolha do padrão (ou padrões) a ser eliminado. Considere novamente o conjunto de padrões da Tabela 1 e suponha que a regra utilizada para escolha da seqüência seja a regra *primeira*. Desta forma, a seqüência 1, com os padrões 1, 3, 5, 2, 4 e números de pilhas abertas iguais a 2, 3, 4, 4, 3, seria escolhida, conforme Tabela 3. Esta seqüência é utilizada a seguir para exemplificar as diferentes regras de escolha dos padrões a serem eliminados. Quatro regras são definidas para eliminar um padrão.

A regra *primeiro* escolhe o primeiro padrão dentre todos os padrões que, quando seqüenciados, abrem um número de pilhas igual ao número máximo de pilhas abertas, obtido pela seqüência escolhida, neste caso 4. Utilizando esta regra, o padrão 5 seria o escolhido, dado que é o primeiro padrão que, quando seqüenciado, abre 4 pilhas. A regra *aleatório* escolhe aleatoriamente um padrão. Com isso qualquer um dos cinco padrões poderia ser aleatoriamente escolhido. A regra *aleatório_mp* escolhe aleatoriamente um padrão dentre aqueles que, quando seqüenciados, abrem um número de pilhas igual ao número máximo de pilhas abertas. Neste exemplo, um sorteio aleatório seria feito entre os padrões 2 e 5, visto que ambos, quando seqüenciados, abrem 4 pilhas. A regra *menor_perda* escolhe, dentre todos os padrões que, quando seqüenciados, abrem um número de pilhas igual ao número máximo de pilhas abertas, aquele padrão que, quando substituído pelos respectivos padrões homogêneos, fornece a menor perda. Utilizando esta regra, os padrões 2 e 5 seriam avaliados. O padrão 2, que contém itens 1, 2, 5 e 6, é substituído por quatro padrões homogêneos. O padrão 5, com itens 3 e 7, é substituído por dois padrões homogêneos. Suponha que a perda do padrão 2 seja de 90 unidades e que o número de repetições seja igual a 2, o que equivale a 180 unidades. E suponha que a perda do padrão 5 seja de 120 unidades e que ele seja repetido apenas uma vez. Considere que a perda dos padrões homogêneos do padrão 2, ponderada pelos respectivos números de repetições de cada padrão homogêneo, seja igual a 240 unidades, e que para o padrão 5 esta perda seja igual a 250 unidades. Neste caso o padrão 2 seria escolhido para ser eliminado.

Rotina de factibilização da solução: após a eliminação de um padrão, a solução deixa de ser factível do ponto de vista do problema de geração de padrões, ou seja, as restrições de

demanda (2) deixam de ser atendidas. A rotina de factibilização determina uma solução factível do ponto de vista do problema de geração, incluindo um padrão homogêneo para cada tipo de item, que havia no padrão que foi eliminado. Estes padrões são inseridos a partir da posição do padrão que foi eliminado. Com a inclusão dos padrões homogêneos, um novo conjunto de padrões de corte é obtido. Considere o conjunto de padrões de corte da Tabela 1 e suponha que o padrão 2 tenha sido eliminado. O novo conjunto de padrões de corte obtido após a factibilização da solução está listado na Tabela 4, onde X indica que o item está presente no padrão.

Tabela 4 – Conjunto de padrões de corte obtido após a eliminação do padrão 2.

Padrão	Item							
	1	2	3	4	5	6	7	8
1			X		X			
2	X							
3		X						
4					X			
5						X		
6	X		X					
7				X	X			X
8			X				X	

Rotina de inclusão de padrões proibidos: esta rotina escolhe um ou mais padrões de corte nomeados *proibidos*, que não podem ser gerados nas próximas iterações pela *rotina de geração de padrões*. O propósito desta proibição é obter um novo conjunto de padrões de corte com uma solução que satisfaça p^i . Todo padrão proibido é obtido pela *rotina de eliminação de padrões*, quando a *rotina de seqüenciamento* é executada após a *rotina de geração de padrões*, utilizando a regra *esc_seq*. A proibição pode ser feita para um ou mais padrões e os mesmos podem ficar proibidos por uma ou mais iterações. Três opções podem ser utilizadas: escolher um padrão para ficar proibido por uma iteração ($LC=1$); escolher um padrão para ficar proibido por x iterações ($LC=x$); ou escolher um padrão para ficar proibido para sempre ($LC=\infty$). Considere novamente o conjunto de padrões da Tabela 1 e suponha que o padrão 3 foi escolhido como proibido na primeira iteração. Se $LC=1$, então ele não poderá ser gerado na segunda iteração, mas, a partir da terceira iteração, já estará liberado. No caso de $LC=x$, considere, por exemplo, $x=5$. O padrão 3 neste caso ficará proibido pelas iterações 2 a 6 e na iteração 7 estará liberado. No caso de $LC=\infty$, o padrão 3 ficará proibido para sempre, isto é, nunca mais poderá ser gerado.

4.2 Abordagem 2

A idéia para a *Abordagem 2* é inspirada em alguns processos industriais de corte encontrados na prática, em que os problemas de geração e seqüenciamento de padrões de corte são resolvidos de forma simultânea. Por meio de um procedimento iterativo, o objetivo do problema de geração de padrões de corte é otimizado, considerando o número máximo de pilhas abertas como uma restrição. Em cada iteração deste procedimento, uma heurística construtiva gulosa é utilizada para gerar os padrões de corte, considerando a restrição do

número máximo de pilhas que podem ser abertas. Este procedimento é repetido por um determinado número de vezes, variando-se os valores relativos dos itens. Na Figura 3 um esquema ilustrativo da *Abordagem 2* é apresentado. Após a descrição geral desta abordagem, cada uma das rotinas é descrita detalhadamente.

Inicialmente é preciso definir o número máximo de iterações, max_iter . Os valores dos itens são atualizados na *rotina de atualização dos valores dos itens* (na primeira iteração eles são iguais ao comprimento de cada item). Na *rotina de geração de padrões* o melhor padrão do ponto de vista de perda é gerado, respeitando a limitação do número máximo de pilhas que pode ser aberto \hat{p}_i . Para tanto é preciso considerar as pilhas de itens que já estão abertas, para saber quantos e quais tipos de itens podem fazer parte do padrão que está sendo gerado. Após a geração do padrão, a *rotina de determinação de repetição do padrão* avalia a demanda atual e determina quantas vezes o mesmo pode ser repetido. A *rotina de atualização da demanda* atualiza a demanda dos itens que fazem parte do padrão previamente gerado. Neste passo, pelo menos um item tem sua demanda exaurida. Todos os itens que tiverem sua demanda atendida são eliminados, a fim de que não sejam incluídos em nenhum outro padrão. Se a demanda de todos os itens ainda não tiver sido totalmente atendida, um novo padrão é gerado na *rotina de geração de padrão*. Esta sub-iteração (Figura 3) continua até que a demanda de todos os itens tenha sido atendida.

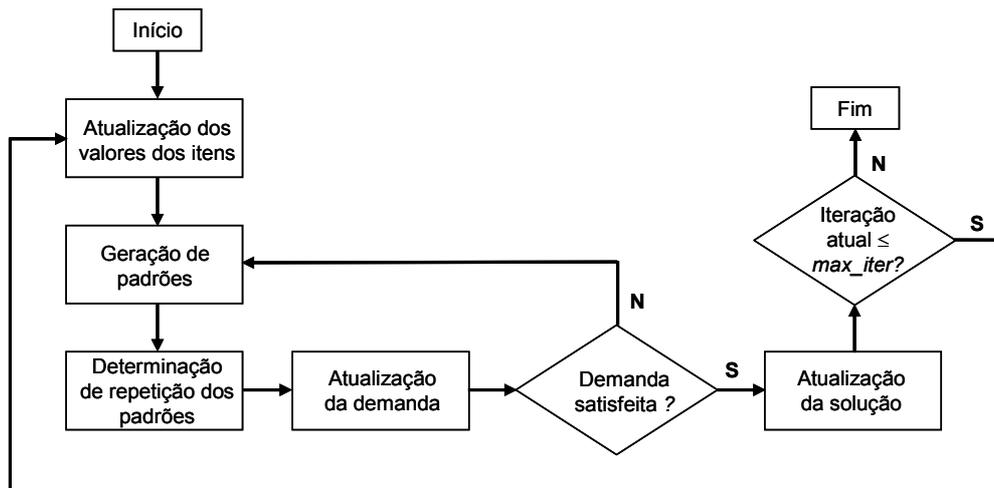


Figura 3 – Esquema ilustrativo da *Abordagem 2*.

Na *rotina de atualização dos valores dos itens*, os mesmos são atualizados e a sub-iteração, que consiste na geração dos padrões de corte por meio da heurística construtiva gulosa, é novamente executada. A cada novo conjunto de padrões gerado é avaliado se a perda obtida é menor que a perda da solução incumbente para atualização da solução. O procedimento termina quando o número máximo de iterações (max_iter) estabelecido tiver sido atingido. No final do algoritmo tem-se o conjunto de padrões de corte com a menor perda obtida e, conseqüentemente, uma seqüência para o corte destes padrões (determinada pela geração dos mesmos).

Rotina de atualização dos valores dos itens: conforme mencionado na seção 2, uma das dificuldades quando se utiliza heurísticas construtivas gulosas com objetivo de minimizar a perda é a concentração de itens menos favoráveis nos padrões finais, o que pode resultar em padrões com perdas elevadas. Logo, uma estratégia é alterar os valores dos itens (em relação aos seus comprimentos) com o objetivo de tentar produzir soluções gerais melhores.

Seja $B = (a_1, a_2, \dots, a_q)$, $q \leq m$, o resultado de uma iteração da heurística gulosa da Figura 3, onde cada $a_j = (a_{1j}, a_{2j}, \dots, a_{mj})$ corresponde a um padrão de corte j com custo c_j . A idéia aqui é encontrar valores duais π_i para os itens $i=1,2,\dots,m$, que minimizem o desvio entre $\sum_{i=1}^m \pi_i a_{ij}$ e c_j , similarmente ao método simplex em que B é uma base e o sistema $\pi^T B = c$ tem solução única, produzindo o vetor multiplicador simplex π .

O posto da matriz B (conforme definida acima) é q , uma vez que em cada iteração da heurística gulosa, a demanda de pelo menos um tipo de item é completamente satisfeita. Portanto, os padrões de corte gerados nas próximas iterações não incluirão este tipo de item, e assim, suas colunas associadas conterão zeros na posição correspondente deste tipo de item, formando um conjunto de colunas linearmente independente. Se $q = m$ então o sistema $B^T \pi = c$ tem solução única; caso contrário ($q < m$), ele tem infinitas soluções. Note que B^T tem q colunas linearmente independentes, para as quais q coordenadas de π estão associadas, e fixamos as restantes em zero para determinar uma solução particular.

Logo, em cada iteração do procedimento da Figura 3, resolve-se o sistema linear acima para atualizar os valores dos itens $i=1, 2, \dots, m$. Na primeira iteração todos os itens têm valor igual ao comprimento, $v_i=l_i$. Nas iterações seguintes esses valores são atualizados para $v_i \leftarrow (1 + \pi_i)v_i$, ou seja, o valor anterior acrescido (ou decrescido) por um multiplicador dado pelo valor de utilidade π_i .

Rotina de geração de padrões: esta rotina resolve um problema de corte unidimensional por meio do algoritmo lexicográfico de Gilmore & Gomory (1963). Cada vez que esta rotina é executada o melhor padrão de corte, do ponto de vista de perda, é gerado. O primeiro padrão gerado é o primeiro a ser seqüenciado; o segundo padrão gerado é o segundo a ser seqüenciado, e assim sucessivamente. Uma restrição adicional é imposta ao problema da mochila, que limita o número máximo de tipos de itens permitidos no padrão, igual a \hat{p}_i , dado que padrões com mais que \hat{p}_i itens são infactíveis. Além desta restrição, o número máximo de pilhas abertas com o seqüenciamento dos padrões também deve ser menor ou igual a \hat{p}_i . Para satisfazer estas duas restrições, assim como na *Abordagem 1*, foram feitas adaptações no algoritmo de Gilmore & Gomory (1963).

O passo 3 aqui modificado verifica se o número de tipos de itens do padrão que está sendo gerado, mais o número de tipos de itens associados às pilhas que estão abertas, é menor ou igual a \hat{p}_i , além de verificar se esse padrão é mais valioso que a solução incumbente. Se essas condições forem satisfeitas, então a solução é atualizada. O passo 5 aqui modificado verifica, a cada eliminação de um item no padrão atual, se o mesmo é mais valioso que a solução incumbente e se o número de tipos de itens no padrão que está sendo gerado, mais o número de tipos de itens associados às pilhas que estão abertas, é menor ou igual a \hat{p}_i para atualização da solução incumbente. A verificação de alocação de itens do tipo $s+1$ também é feita da mesma forma que no passo 5 original do algoritmo de Gilmore & Gomory (1963).

Considere um vetor $piAb = (p_i, p_i, \dots, p_i)$ que indica quais pilhas de itens estão abertas quando um novo padrão vai ser gerado, onde:

$$p_i = \begin{cases} 1, & \text{se a pilha associada ao item } i \text{ está aberta e o padrão avaliado não contém este item} \\ 0, & \text{caso contrário} \end{cases}$$

Por exemplo, considere $m = 10$ tipos de itens e $\hat{p} = 3$. Inicialmente o vetor $PiAb = (0, 0, 0, 0, 0, 0, 0, 0, 0, 0)$, visto que nenhuma pilha foi aberta, pois nenhum padrão foi gerado. Suponha que o primeiro padrão gerado contenha 3 tipos de itens diferentes, diga-se os itens 1, 7 e 9. Com o corte deste padrão, suponha que a demanda do item 7 tenha sido satisfeita, ou seja, a pilha associada a este item foi fechada. O vetor $PiAb$ neste caso passa a ser dado por $(1, 0, 0, 0, 0, 0, 0, 1, 0)$, o que indica que as pilhas associadas aos itens 1 e 9 estão abertas. Na geração do segundo padrão apenas um tipo de item, além dos itens 1 e 9, poderá ser incluído, pois os itens 1 e 9 estão com suas pilhas abertas.

Rotina de determinação de repetição do padrão: nesta rotina o número de vezes (x_j) que cada padrão j (gerado na *rotina de geração de padrões*) deve ser repetido é determinado. Este número é o mínimo entre as demandas b_i dos diferentes tipos de itens i , que fazem parte do padrão j , dividido pelo número de vezes a_{ij} que os itens i aparecem no padrão j , ou seja,

$$\min_{i \in a_j} \left\lceil \frac{b_i}{a_{ij}} \right\rceil, \text{ onde } \lceil z \rceil \text{ é o menor inteiro maior ou igual a } z. \text{ Considere o exemplo com } L=100$$

e $m=3$, com respectivos comprimentos e demandas dos itens: $l_1 = 15$, $l_2 = 20$ e $l_3 = 50$ e $b_1 = 1200$, $b_2 = 545$ e $b_3 = 300$.

Suponha que o padrão (0,2,1) tenha sido gerado. Este padrão é repetido 273 vezes, pois

$$\min \left\lceil \frac{545}{2}, \frac{300}{1} \right\rceil = \min \lceil 272,5 ; 300 \rceil = 273.$$

Rotina de atualização da demanda: esta rotina atualiza a demanda dos itens gerados com o corte do padrão previamente obtido, ou seja, $b_i \leftarrow \max \{0, b_i - a_{ij}x_j\}$, onde x_j é o número de vezes que o padrão j é repetido, definido na *rotina de determinação de repetição do padrão*.

Rotina de atualização da solução: nesta rotina a melhor solução encontrada do ponto de vista de perda é armazenada.

4.3 Abordagem 3

A terceira abordagem, *Abordagem 3*, consiste em resolver os problemas de geração e seqüenciamento de padrões, utilizando o método simplex com um procedimento de geração de colunas que considera a restrição do número máximo de pilhas abertas, \hat{p} . Isto é feito durante a geração de cada coluna (padrão) por meio da heurística 3 de Yuen (1995). Diferentemente da *Abordagem 1*, na *Abordagem 3* todas as soluções obtidas pelo método *simplex* também são factíveis do ponto de vista do seqüenciamento, isto é, cada conjunto de padrões de corte gerado em cada iteração do método simplex sempre satisfaz o número máximo de pilhas tolerado, \hat{p} , quando seqüenciado. No procedimento de geração de

colunas, só são avaliadas as colunas que possuem custo relativo negativo, ou seja, que quando inseridas na base melhoram o valor da função objetivo. Para cada coluna com custo relativo negativo é avaliado se a entrada da mesma na base (e conseqüentemente a saída de uma outra coluna) resultará em $pi \leq \hat{pi}$, ou seja, toda coluna gerada pelo procedimento de geração de colunas resultará na melhora do valor da função objetivo e em $pi \leq \hat{pi}$.

A *Abordagem 3* consiste em alterar os passos 2 e 3 do método simplex com geração de colunas apresentado na seção 2, da seguinte forma:

PASSO 2': Sejam $a^{(1)}, a^{(2)}, \dots$, a melhor solução, a segunda melhor solução, etc., no sub-problema $\min c - \pi a$ do passo 2 (note que o algoritmo anterior considera apenas $a^{(1)}$). Denote por $c^{(1)}, c^{(2)}, \dots$, as perdas das colunas $a^{(1)}, a^{(2)}, \dots$, respectivamente.

PASSO 3': Se $c^{(1)} - \pi a^{(1)} \geq 0$, então pare com uma solução factível. Senão seja $a^{r(s)}$ a coluna a sair da base caso a coluna $a^{(s)}$ entre na base. Encontre o primeiro padrão s tal que:

- i) $c^{(s)} - \pi a^{(s)} < 0$ (custo relativo negativo, portanto a perda diminui);
- ii) o conjunto $\{a^{(1)}, \dots, a^{(s)}, \dots, a^{(m)}\}$ (a coluna $a^{(s)}$ substitui a coluna $a^{r(s)}$) abre um número máximo de pilhas menor ou igual a \hat{pi} (estabelecido *a priori*).

Se não existe tal padrão, pare com uma solução factível; caso contrário, volte para o passo 1.

Para resolver o problema de geração de padrões de corte unidimensional, o algoritmo lexicográfico de Gilmore & Gomory (1963) é utilizado. Assim como nas Abordagens 1 e 2, modificações foram feitas nos passos 3 e 5 desse algoritmo a fim de garantir que a restrição do problema de seqüenciamento seja verificada.

O passo 3 aqui modificado verifica se o padrão gerado (coluna $a^{(s)}$) é mais valioso que a solução incumbente, se a restrição quanto ao número máximo de tipos de itens no padrão está sendo respeitada, se o custo relativo dessa coluna é negativo (i.e., a entrada da coluna $a^{(s)}$ na base fará com que a perda diminua) e se o novo conjunto de padrões (obtido com a entrada da coluna $a^{(s)}$ na base) é seqüenciável, ou seja, se o número máximo de tipos de itens obtido com o seqüenciamento desses padrões é menor ou igual a \hat{pi} . Caso todas essas condições sejam satisfeitas, então a solução é atualizada.

O passo 5 aqui modificado, a cada eliminação de um item no padrão atual, verifica se o mesmo é mais valioso que a solução incumbente, se a restrição quanto a \hat{pi} é satisfeita, se o custo relativo do padrão é negativo e se o novo conjunto de padrões obtido com a entrada da coluna $a^{(s)}$ na base (padrão avaliado com o elemento a_s igual a $a_s - 1$) é seqüenciável. Caso todas essas condições sejam satisfeitas, então a solução é atualizada. Estas verificações são feitas, antes de verificar se compensa retirar um item tipo s para inserir itens tipo $s+1, s+2, \dots, m$.

Convém ressaltar que as *Abordagens 1, 2 e 3* podem ser utilizadas para gerar pontos aproximados da curva de eficiência, variando-se \hat{pi} em P2. Assim como as *Abordagens 1 e 2*, a *Abordagem 3*, mesmo com um método exato de seqüenciamento, também não é exata. Para mais detalhes das *Abordagens 1, 2 e 3*, o leitor pode consultar Pileggi (2002).

5. Resultados Computacionais

Certa dificuldade foi encontrada para analisar o desempenho das *Abordagens 1, 2 e 3* devido à falta de métodos exatos e/ou exemplos resolvidos otimamente na literatura para comparação. Também não foram encontrados na literatura limitantes inferiores eficazes para a perda mínima e o número máximo de pilhas abertas do problema integrado de geração e seqüenciamento de padrões de corte. Dispõe-se apenas do limitante inferior para a perda mínima, obtido resolvendo-se o problema de geração de padrões de corte de forma independente, que não necessariamente é um bom limitante para a perda mínima do problema integrado. Em função disso, a estratégia utilizada neste trabalho foi comparar as *Abordagens 1, 2 e 3* entre si e com esse limitante inferior para a perda mínima. As implementações computacionais foram codificadas na linguagem Pascal e foram compiladas utilizando o compilador Delphi, versão 5.0. O microcomputador utilizado foi um Pentium IV com 1600Mhz e 512Mb de memória.

Para resolver o problema de geração de padrões de corte (1)-(3), uma versão do método simplex com um procedimento de geração de colunas em Morabito & Arenales (2000) foi utilizada. Nesta versão, os padrões homogêneos são utilizados como solução inicial para a fase I do método simplex, e os sistemas lineares são resolvidos pelo método de eliminação de Gauss com pivotamento parcial. De acordo com Gilmore & Gomory (1961, 1963) um critério de parada é interromper o algoritmo simplex se após algumas iterações a melhoria na função objetivo for pouco significativa. Nos testes realizados, o método foi interrompido se após 10 iterações a redução do valor da função objetivo não foi superior a 0,1%. Os problemas da mochila envolvidos no problema (1)-(3) (*Abordagens 1 e 3*) e na *Abordagem 2*, para geração de padrões de corte unidimensionais, foram resolvidos utilizando o algoritmo lexicográfico de Gilmore & Gomory (1963), com as modificações descritas na seção 4. Outros métodos poderiam ter sido utilizados, conforme referências na seção 2.

Para resolver o problema de seqüenciamento de padrões de corte, foi implementada a heurística 3 de Yuen (1995). Outras heurísticas de seqüenciamento poderiam ter sido utilizadas, como, por exemplo, a heurística com busca tabu de Faggioli & Bentivoglio (1998) e o método frugal de Ashikaga (2001), que envolve teoria dos grafos. As *Abordagens 1 e 3* utilizam a heurística de 3 de Yuen para resolver o problema de seqüenciamento, conforme descrito nas seções 4.1 e 4.3.

5.1 Experimentos com exemplos aleatórios

Nos experimentos analisados, considerou-se $L=100$ com a demanda dos itens gerada aleatoriamente no intervalo $[1000, 10000]$. O comprimento dos itens (l_i) foi gerado aleatoriamente nos intervalos $[0,10L; 0,75L]$ (tamanhos diversos) e $[0,05L; 0,50L]$ (tamanhos pequenos). Desta forma, os experimentos foram divididos nas seguintes classes:

Classe 1: $m = 10$ e $l_i \in [0,10L; 0,75L]$ **Classe 3:** $m = 10$ e $l_i \in [0,05L; 0,50L]$
Classe 2: $m = 20$ e $l_i \in [0,10L; 0,75L]$ **Classe 4:** $m = 20$ e $l_i \in [0,05L; 0,50L]$

O número máximo de pilhas abertas tolerado, \hat{p}_i , foi fixado arbitrariamente em 3. Note que este número torna os problemas das classes 2 e 4 (com $m = 20$) bem mais restritivos do que os das classes 1 e 3 (com $m = 10$), respectivamente, conforme é visto nas tabelas a seguir. Em cada classe, foram gerados exemplos até se obterem 50 exemplos ineficazes, ou seja,

com número máximo de pilhas abertas (obtido com o seqüenciamento do conjunto de padrões de corte gerado pelo método simplex) maior que o número máximo de pilhas abertas tolerado, $p\hat{i}$. Para a *Abordagem 1* os parâmetros utilizados foram: esc_pad =menor_perda, esc_seq =aleatória, $LC=m/2$ e $max_iter=100$. Para a *Abordagem 2*, o parâmetro max_iter foi fixado em m . E para a *Abordagem 3*, a regra utilizada para esc_seq foi aleatória (para mais detalhes de como esses parâmetros foram aferidos, veja Pileggi, 2002).

Nas Tabelas 5 a 8 a coluna “Método de Solução” corresponde ao método utilizado para resolver os problemas de geração e seqüenciamento de padrões de corte. A linha “Simplex com Heurística 3” corresponde à aplicação do método simplex com o procedimento de geração de colunas (sem restrição para o número máximo de tipos de itens nos padrões) para solução do problema de geração de padrões, seguido do procedimento de seqüenciamento utilizando a heurística 3 de Yuen (1995). A linha “Simplex ($p\hat{i}$) com Heurística 3” corresponde à utilização do método simplex com o procedimento de geração de colunas (com no máximo $p\hat{i}$ tipos de itens nos padrões) para solução do problema de geração de padrões, seguido do procedimento de seqüenciamento utilizando a heurística 3 de Yuen (1995). Note que estes métodos não garantem uma solução com número máximo de pilhas menor ou igual a 3 (veja discussão anterior sobre geração dos exemplos aleatórios), mas produzem limitantes inferiores para a perda mínima. As colunas “Perda (%)”, “Pilha” e “Tempo” correspondem, respectivamente, às médias das perdas (em porcentagem), dos números máximos de pilhas abertas e dos tempos de execução do algoritmo (em segundos). A perda do conjunto de padrões gerado pelo método simplex foi calculada da seguinte forma: área total de objetos cortados menos área total de itens produzidos, sobre área total de objetos cortados.

Tabela 5 – Resultados médios obtidos para os 50 exemplos da classe 1.

$m = 10, l_i \in [0,10L; 0,75L]$			
Método de Solução	Perda	Pilha	Tempo
<i>Simplex com Heurística 3</i>	4,88	4,14	< 0,01
<i>Simplex ($p\hat{i}$) com Heurística 3</i>	4,88	4,10	< 0,01
<i>Abordagem 1</i>	4,96	3,00	0,18
<i>Abordagem 2</i>	5,65	3,00	< 0,01
<i>Abordagem 3</i>	5,06	3,00	0,01

Tabela 6 – Resultados médios obtidos para os 50 exemplos da classe 2.

$m = 20, l_i \in [0,10L; 0,75L]$			
Método de Solução	Perda	Pilha	Tempo
<i>Simplex com Heurística 3</i>	3,77	5,36	0,01
<i>Simplex ($p\hat{i}$) com Heurística 3</i>	3,77	5,34	0,01
<i>Abordagem 1</i>	4,40	3,00	29,11
<i>Abordagem 2</i>	4,56	3,00	0,02
<i>Abordagem 3</i>	4,57	3,00	0,87

Tabela 7 – Resultados médios obtidos para os 50 exemplos da classe 3.

$m = 10, l_i \in [0,05L; 0,50L]$			
Método de Solução	Perda	Pilha	Tempo
<i>Simplex com Heurística 3</i>	0,36	5,04	< 0,01
<i>Simplex (\hat{p}_i) com Heurística 3</i>	0,42	4,56	< 0,01
<i>Abordagem 1</i>	0,61	3,00	0,85
<i>Abordagem 2</i>	1,44	3,00	< 0,01
<i>Abordagem 3</i>	0,75	3,00	0,03

Tabela 8 – Resultados médios obtidos para os 50 exemplos da classe 4.

$m = 20, l_i \in [0,05L; 0,50L]$			
Método de Solução	Perda	Pilha	Tempo
<i>Simplex com Heurística 3</i>	0,00	6,92	0,01
<i>Simplex (\hat{p}_i) com Heurística 3</i>	0,00	6,58	0,01
<i>Abordagem 1</i>	1,28	3,00	65,49
<i>Abordagem 2</i>	0,95	3,00	0,03
<i>Abordagem 3</i>	0,47	3,00	9,20

Conforme se pode observar nas Tabelas 5 a 7, a *Abordagem 1* apresentou as menores perdas para as classes 1, 2 e 3, mas os maiores tempos computacionais, enquanto a *Abordagem 3* apresentou a menor perda para a classe 4 (Tabela 8) em um tempo computacional bem menor do que o da *Abordagem 1*. Observe que, para as classes 1 a 3, os resultados obtidos pela *Abordagem 3* são bastante próximos dos obtidos pela *Abordagem 1*, apesar de esta última se comportar sempre melhor. Já para a classe 4, com 20 tipos de itens de tamanhos pequenos, a *Abordagem 3* encontra soluções melhores que as da *Abordagem 1*.

Com relação à superioridade da *Abordagem 3* para os exemplos da classe 4, note que a *Abordagem 1* foi a que se comportou pior dentre as três abordagens (observe o gráfico da Figura 4). Conforme o número de tipos de itens m aumenta e o tamanho dos mesmos diminui (compare o número de tipos de itens e o tamanho dos mesmos para as classes 1 e 4), para um mesmo valor de \hat{p}_i , a *Abordagem 3* se comporta melhor que a *Abordagem 1*. Note no gráfico que, para a classe 1, mais de 90% das soluções são obtidas a partir da *rotina de geração de padrões* e, para a classe 4, nenhuma solução para os problemas de geração e seqüenciamento de padrões é obtida a partir desta rotina. Isto é, nos exemplos da classe 1, com a proibição de alguns padrões de corte, o método simplex consegue encontrar soluções de baixa perda (não necessariamente de perda mínima) factíveis do ponto de vista do problema de seqüenciamento de padrões. Já para a classe 4, nenhuma das soluções obtidas pelo método simplex foi factível do ponto de vista do seqüenciamento de padrões, sendo necessário, para todos os exemplos desta classe, a factibilização das soluções para que o número máximo de pilhas abertas fosse menor ou igual a \hat{p}_i . Se por um lado isso resolve o problema de seqüenciamento, por outro pode gerar um aumento da perda (veja na Tabela 8 a perda obtida pela *Abordagem 1*).

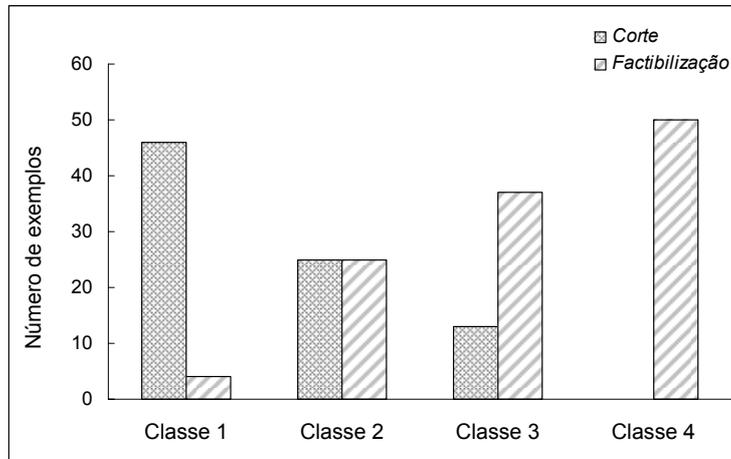


Figura 4 – Exemplos resolvidos pela *rotina de corte* e pela *rotina de factibilização da solução* para as classes 1-4.

A diferença entre a perda obtida pela *Abordagem 2* e o limitante inferior para as classes 1 e 2 (itens de tamanhos diversos) é praticamente constante, menor que 0,80%. O mesmo ocorre para as classes 3 e 4 (itens de tamanhos pequenos), onde esta diferença fica em torno de 1%. Para problemas nos quais a diferença entre o número máximo de pilhas abertas, obtido pelo seqüenciamento dos padrões de corte gerados pelo método simplex, e \hat{p}_i não é muito grande, a *Abordagem 1* apresenta melhor desempenho que as *Abordagens 2* e 3. Porém, quando esta diferença começa a aumentar, a *Abordagem 3* apresenta melhores resultados, seguida da *Abordagem 2*. À medida que o número de tipos de itens aumenta, o tempo computacional das abordagens também aumenta, mas o tempo da *Abordagem 1* é bem maior que os das demais.

Uma análise foi feita para comparar as três abordagens de uma forma mais agregada, considerando o desempenho médio das mesmas em todas as classes. Na Tabela 9 o desempenho das abordagens está avaliado pelo número de colocações das mesmas. As colunas “Primeira”, “Segunda” e “Terceira” indicam respectivamente o número de classes nas quais as *Abordagens 1, 2 e 3* obtiveram a primeira colocação (isto é, a menor perda), a segunda e a terceira colocação, respectivamente. Convém observar que não houve casos de empate entre as abordagens. Pode-se observar que a *Abordagem 1* obteve a menor perda média em 3 dos 4 dos casos, a *Abordagem 2* em nenhuma classe e a *Abordagem 3* em apenas uma classe. Note que a *Abordagem 2* obteve o pior desempenho em 2 dos 4 casos e a *Abordagem 3* foi a que mais variou seu desempenho médio.

Tabela 9 – Comparação das abordagens para as classes 1-4.

Método de Solução	Primeira	Segunda	Terceira
<i>Abordagem 1</i>	3	0	1
<i>Abordagem 2</i>	0	2	2
<i>Abordagem 3</i>	1	2	1

Na Tabela 10, o desempenho médio das abordagens com relação à perda e ao tempo computacional está ilustrado. Nesta tabela a coluna “Perda” indica a perda média (em porcentagem) obtida por cada abordagem e “Tempo” indica o tempo médio (em segundos) gasto por cada abordagem, considerando as quatro classes. Do ponto de vista da perda média, a *Abordagem 3* apresenta os menores valores, uma diferença de 0,46% da média obtida para o limitante inferior (2,25%). Note que a diferença entre o limitante inferior e a *Abordagem 2* (0,90%) é praticamente o dobro da diferença obtida para a *Abordagem 3* (0,46%) e mais que 1,5 vezes para a *Abordagem 1* (0,56%). Observe que o tempo computacional da *Abordagem 1* é maior que das *Abordagens 2 e 3*.

Tabela 10 – Resultados médios obtidos para as classes 1-4.

Método de Solução	Perda	Tempo
<i>Simplex com Heurística 3</i>	2,25	0,01
<i>Abordagem 1</i>	2,81	23,91
<i>Abordagem 2</i>	3,15	0,02
<i>Abordagem 3</i>	2,71	2,53

5.2 Geração de curvas de *trade-off*

Todos os experimentos acima foram realizados para $p\hat{i} = 3$. Novos experimentos foram realizados para avaliar o desempenho das três abordagens para diferentes valores de $p\hat{i}$, ou seja, para avaliar o desempenho das mesmas para gerar curvas de *trade-off* (variando-se os valores de $p\hat{i}$). Como não se dispõe de bons limitantes inferiores para a perda e o máximo número de pilhas abertas, mas apenas para a perda, a análise da qualidade das *Abordagens* fica um pouco limitada à comparação das mesmas entre si. As três abordagens foram utilizadas para resolver um exemplo (escolhido arbitrariamente) das classes 1 e 2. Os dados utilizados para a geração das curvas de *trade-off* encontram-se listados nas Tabelas 11 e 12.

Tabela 11 – Dados do exemplo da classe 1.

Item	1	2	3	4	5	6	7	8	9	10
l_i	11	36	22	53	31	44	16	40	13	41
d_i	3042	1683	9637	7911	2971	7746	4701	4580	5174	5825

Tabela 12 – Dados do exemplo da classe 2.

Item	1	2	3	4	5	6	7	8	9	10
l_i	22	42	33	64	24	38	41	10	30	45
d_i	4913	8383	4454	5563	5674	5228	5009	4736	9063	6160
Item	11	12	13	14	15	16	17	18	19	20
l_i	70	11	26	57	39	71	37	72	20	62
d_i	7412	6245	8954	9553	9339	6440	4941	1976	7372	4200

Nas Figuras 5 e 6 o eixo x corresponde à menor perda (em porcentagem) e o eixo y , ao menor número máximo de pilhas abertas obtido com o seqüenciamento dos padrões gerados pelo método simplex. Para o exemplo da classe 1 ($m=10$ e $l_i \in [0,10L; 0,75L]$), variou-se $p\hat{i} = 1,2,\dots,5$ (note que 5 é o limitante superior para este exemplo, pois é o número máximo de pilhas abertas obtidas pelo método simplex, sem a restrição para o número máximo de tipos de itens nos padrões, e a heurística 3). Nas Figuras 5 e 6 ilustram-se as curvas de *trade-off* para o exemplo classes 1 e 2. Observe na Figura 5 que as curvas geradas pelas três abordagens são bastante próximas, porém, a *Abordagem 1* apresenta soluções um pouco melhores que as demais. Para este exemplo da classe 1, a *Abordagem 1* dominou as *Abordagens 2* e 3. Para o exemplo da classe 2 ($m=20$ e $l_i \in [0,10L; 0,75L]$), variou-se $p\hat{i} = 1,2,\dots,7$ (note que 7 é o limitante superior para este exemplo).

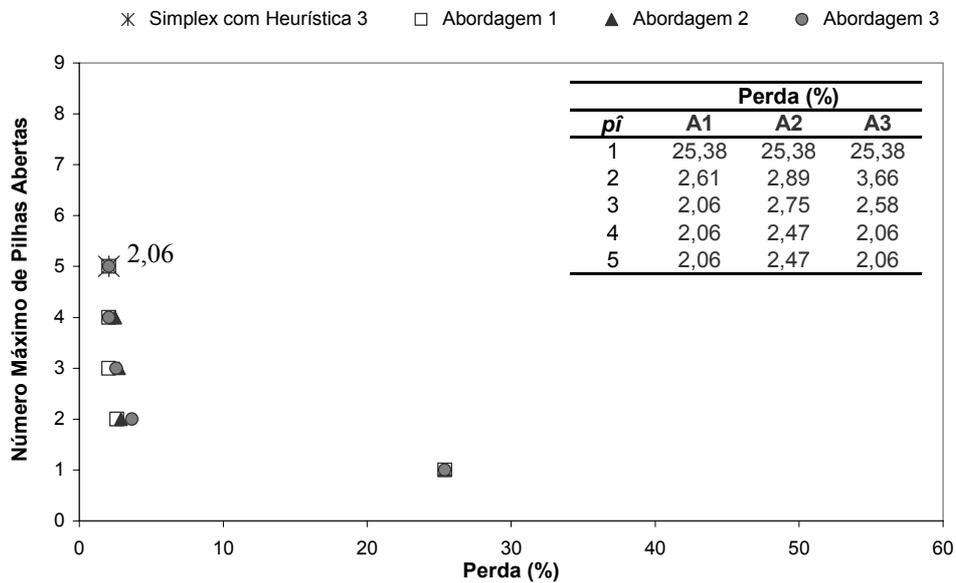


Figura 5 – Curvas de *trade-off* entre a perda e o número máximo de pilhas abertas no exemplo da classe 1.

Observe na Figura 6 que as *Abordagens 1* e 3 apresentam um desempenho um pouco melhor que a *Abordagem 2*. Observe que a *Abordagem 1* não domina a *Abordagem 2* e a *Abordagem 2* não domina a *Abordagem 1* (veja nas colunas A1 e A2 da tabela do gráfico da Figura 6, as linhas com $p\hat{i} = 3$ e 4). A *Abordagem 1* também não domina a *Abordagem 3* e a *Abordagem 3* não domina a *Abordagem 1* (veja nas colunas A1 e A3 da tabela do gráfico da Figura 6, as linhas com $p\hat{i} = 3$ e 4). Os resultados obtidos pelas curvas de *trade-off* mostram que as *Abordagens 1* e 2, 1 e 3 e 2 e 3 não dominam uma a outra, e as perdas médias obtidas pela *Abordagem 2* são, em geral, piores.

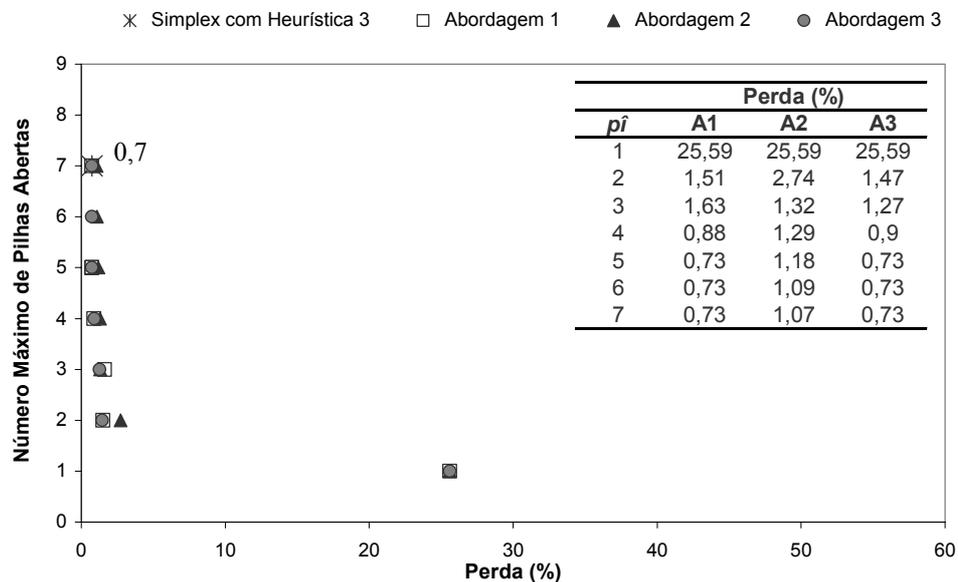


Figura 6 – Curvas de *trade-off* entre a perda e o número máximo de pilhas abertas para o exemplo da classe 2.

6. Conclusões e Perspectivas

Neste trabalho três abordagens foram propostas para resolver o problema integrado de geração e seqüenciamento de padrões de corte. As três abordagens baseiam-se na minimização do objetivo do problema de corte, sujeito a uma limitação do objetivo do problema de seqüenciamento (problema P2). Conforme mencionado, não se dispõe de limitantes inferiores eficazes para a perda e o número máximo de pilhas abertas do problema integrado. Em função disso, as abordagens foram comparadas entre si, e com o limitante inferior para a perda obtido do problema de geração de padrões.

De modo geral as três abordagens tiveram um bom desempenho, ou seja, o *gap* entre as soluções obtidas e o limitante inferior é razoavelmente pequeno (Tabelas 5 a 8). A *Abordagem 1* encontrou os melhores resultados com relação ao número de primeiras colocações, isto é, encontrou as menores perdas em três das quatro classes analisadas. A *Abordagem 2* foi a que, em geral, obteve as piores colocações (apresentou as piores perdas em três das quatro classes analisadas). A *Abordagem 3* foi a que apresentou um comportamento mais irregular com relação ao número de colocações. Comparando o desempenho médio das três abordagens com relação à perda, a *Abordagem 3* obteve os menores valores, seguida da *Abordagem 1*. Isso deve ao resultado obtido pela *Abordagem 1* na classe 4 ($m = 20$ e os itens são de tamanhos pequenos). A *Abordagem 2* obteve as piores perdas médias pois, em geral, para as quatro classes, foi a que apresentou os piores resultados.

Com relação aos tempos computacionais, a *Abordagem 2* foi a mais rápida, com um tempo médio de milésimos de segundo, seguida da *Abordagem 3* com tempos médios de 2 segundos. A *Abordagem 1* foi a mais lenta, com um tempo médio na ordem de 30 segundos

e, no pior caso, aproximadamente 1 minuto (classe 4). Isto ocorre em função do grande número de vezes que a heurística de seqüenciamento é executada na tentativa de factibilizar a solução quando o número de tipos de itens aumenta e o tamanho dos mesmos diminui. Esses tempos são bastante razoáveis considerando o tipo de decisão envolvida. Para as três abordagens, os tempos computacionais aumentam de $m=10$ para $m=20$, como era esperado.

Quando a redução necessária entre o número máximo de pilhas abertas, obtido pelo seqüenciamento dos padrões gerados pelo método simplex, e o limitante imposto sobre o número máximo de pilhas abertas \hat{p}_i é pequena, a *Abordagem 1* apresenta os melhores desempenhos. Conforme essa diferença aumenta, os melhores resultados são obtidos pela *Abordagem 3*. Acredita-se que isto ocorre pois, à medida que essa diferença aumenta, as soluções factíveis obtidas pela *Abordagem 1* são geradas na *rotina de factibilização da solução* (Figura 4), que tem como objetivo obter uma solução com $p_i \leq \hat{p}_i$, sem uma grande preocupação com a perda. Por outro lado, quando a diferença entre o número máximo de pilhas abertas, obtido pelo seqüenciamento dos padrões gerados pelo método simplex, e \hat{p}_i é pequena, em geral as melhores soluções do ponto de vista de perda e factíveis (i.e., com $p_i \leq \hat{p}_i$) são obtidas pela *rotina de geração de padrões*, que tem como objetivo a minimização da perda.

As três abordagens propostas podem ser utilizadas para gerar curvas de *trade-off* entre a perda e o número máximo de pilhas abertas, variando-se \hat{p}_i . Conforme ilustrado pelas curvas das Figuras 5 e 6, as *Abordagens 1, 2 e 3* não dominam uma a outra, e as perdas médias obtidas pela *Abordagem 2* são, em geral, piores. As três abordagens também podem ser utilizadas para tratar problemas de corte e empacotamento com mais de uma dimensão e com outras restrições adicionais, como, por exemplo, o caso bidimensional guilhotinado em 2-estágios e o caso bidimensional guilhotinado não-estagiado (Pileggi, 2002). Esses casos serão objeto de um próximo trabalho ainda a ser compilado.

Uma perspectiva importante para pesquisa futura é desenvolver efetivamente estudos de casos aplicando as abordagens propostas em situações reais e comparar os resultados com os obtidos pelas empresas. Pesquisas podem ainda ser realizadas para melhorar a *Abordagem 1*, em particular a *rotina de eliminação de padrões* e a *rotina de factibilização da solução*, com o objetivo de encontrar soluções factíveis (do ponto de vista do número máximo de pilhas abertas) com perdas menores. Conforme visto, a *Abordagem 3* pode deixar de encontrar soluções melhores (do ponto de vista de perda), pelo fato de não poder visitar vértices que correspondam a padrões de corte que, quando seqüenciados, abram mais pilhas do que o número máximo tolerado (\hat{p}_i). Uma pesquisa interessante seria desenvolver alternativas que permitam que vértices infactíveis (do ponto de vista do número máximo de pilhas abertas) sejam visitados, com o objetivo de obter soluções melhores, isto é, com perdas menores. A rotina de seqüenciamento das *Abordagens 1 e 3* utiliza a heurística 3 de Yuen (1995). Uma questão a ser investigada é como variam os comportamentos destas abordagens ao se utilizar outras heurísticas de seqüenciamento mais poderosas.

Agradecimentos

Os autores agradecem aos três árbitros anônimos pelos úteis comentários e sugestões. Esta pesquisa foi apoiada pela FAPESP e CNPq.

Referências Bibliográficas

- (1) Arenales, M.; Morabito, R. & Yanasse, H. (1999). Cutting and packing problems. *Pesquisa Operacional*, **19**(2), 107-299.
- (2) Armbruster, M. (2002). A solution procedure for a pattern sequencing problem as part of a one-dimensional cutting stock problem in the steel industry. *European Journal of Operational Research*, **141**, 328-340.
- (3) Arroyo, J.E.C. (2002). Heurísticas e metaheurísticas para otimização combinatória multiobjetivo. 256f. Tese (Doutorado em Engenharia Elétrica) – Faculdade de Engenharia Elétrica e de Computação, Universidade Estadual de Campinas.
- (4) Ashikaga, F.M. (2001). Um método frugal para o problema de minimização de pilhas abertas. 91f. Dissertação (Mestrado em Ciências no Curso de Engenharia Eletrônica e Computação na Área de Informática) – Instituto Tecnológico de Aeronáutica, São José dos Campos.
- (5) Becceneri, J.C. (1999). O problema de seqüenciamento de padrões para a minimização do número máximo de pilhas abertas em ambientes de cortes industriais. 145f. Tese (Doutorado em Ciências no Curso de Engenharia Eletrônica e Computação na Área de Informática) – Instituto Tecnológico de Aeronáutica, São José dos Campos.
- (6) Belluzzo, L. & Morabito, R. (2005). Otimização nos padrões de corte de chapas de fibra de madeira reconstituída: um estudo de caso. *Pesquisa Operacional*, **25**(3), 391-415.
- (7) Bischoff, E. & Waescher, G. (Eds.) (1995). Cutting and packing. *European Journal of Operational Research*, **84**(3).
- (8) Cohon, J.L. (1978). *Multiobjective programming & planning*. Academic Press, New York.
- (9) Dowsland, K. & Dowsland, W. (1992). Packing problems. *European Journal of Operational Research*, **56**, 2-14.
- (10) Dyckhoff, H. & Finke, U. (1992). *Cutting and packing in production and distribution: Typology and bibliography*. Springer-Verlag, Heidelberg.
- (11) Dyckhoff, H.; Scheithauer, G. & Terno, J. (1997). *Cutting and packing*. In: *Annotated bibliographies in combinatorial optimization* [edited by M. Amico; F. Maffioli and S. Martello], John Wiley & Sons, New York, 393-414.
- (12) Dyckhoff, H. & Waescher, G. (Eds.) (1990). Cutting and packing. *European Journal of Operational Research*, **44**(2).
- (13) Dyson, R.G. & Gregory, A.S. (1974). The cutting stock problem in the flat glass industry. *Operational Research Quarterly*, **25**, 41-53.
- (14) Faggioli, E. & Bentivoglio, C.A. (1998). Heuristic and exact methods for the cutting sequencing problem. *European Journal of Operational Research*, **110**(3), 564-575.
- (15) Fink, A. & Voß, S. (1999). Applications of modern heuristic search methods to pattern sequencing problems. *Computer and Operations Research*, **26**(1), 17-34.
- (16) Foerster, H.E. & Waescher, G. (1998). Simulated annealing for order spread minimization in sequencing cutting patterns. *European Journal of Operational Research*, **110**, 272-281.

- (17) Gilmore, P. & Gomory, R. (1961). A linear programming approach to the cutting stock problem. *Operations Research*, **9**, 849-859.
- (18) Gilmore, P. & Gomory, R. (1963). A linear programming approach to the cutting-stock problem II. *Operations Research*, **11**, 863-888.
- (19) Gramani, M.C.N. (2001). Otimização do processo de cortagem acoplado ao planejamento da produção. 118f. Tese (Doutorado em Engenharia Elétrica), Faculdade de Engenharia Elétrica e de Computação, Universidade Estadual de Campinas.
- (20) Haessler, R. (1980). A note on computational modifications to the Gilmore-Gomory cutting stock algorithm. *Operations Research*, **28**, 1001-1005.
- (21) Hendry, L.C.; Fok, K.K. & Shek, K.W. (1996). A cutting stock and scheduling problem in the copper industry. *Journal of the Operational Research Society*, **47**, 38-47.
- (22) Hinxman, A. (1980). The trim-loss and assortment problems: a survey. *European Journal of Operational Research*, **5**, 8-18.
- (23) Hifi, M. (2002). Special issue: Cutting and packing problems. *Studia Informatica Universalis*, **2**(1), 1-161.
- (24) Linhares, A. & Yanasse, H.H. (2002). Connections between cutting-pattern sequencing, VLSI Design, and flexible machines. *Computers & Operations Research*, **29**(12), 1759-1772.
- (25) Lins, S. (1989). Traversing trees and scheduling tasks for duplex corrugator machines. *Pesquisa Operacional*, **9**, 40-54.
- (26) Lodi, A.; Martello, S. & Monaci, M. (2002). Two-dimensional packing problems: a survey. *European Journal of Operational Research*, **141**, 241-252.
- (27) Madsen, O. (1979). Glass cutting in a small firm. *Mathematical Programming*, **17**, 85-90.
- (28) Madsen, O. (1988). An application of travelling-salesman routines to solve pattern-allocation problems in the glass industry. *Journal of the Operational Research Society*, **39**, 249-256.
- (29) Martello, S. & Toth, P. (1990). *Knapsack problems: algorithms and computer implementations*. John Wiley & Sons, Chichester.
- (30) Morabito, R. & Arenales, M. (1992). Um exame dos problemas de corte e empacotamento. *Pesquisa Operacional*, **12**(1), 1-20.
- (31) Morabito, R. & Arenales, M. (2000). Optimizing the cutting of stock plates in a furniture company. *International Journal of Production Research*, **38**(12), 2725-2742.
- (32) Morabito, R. & Garcia, V. (1998a). Uma abordagem para o problema de cortes de chapas de fibra de madeira reconstituída. *Pesquisa Operacional*, **18**(1), 37-57.
- (33) Morabito, R. & Garcia, V. (1998b). The cutting stock problem in a hardboard industry: A case study. *Computers & Operations Research*, **25**(6), 469-485.
- (34) Oliveira, J.F. & Waescher, G. (2005). Special Issue on Cutting and Packing. *European Journal of Operational Research* (a aparecer).

- (35) Pileggi, G.C.F. (2002). Abordagens para otimização integrada dos problemas de geração e seqüenciamento de padrões de corte. 154f. Tese (Doutorado em Ciências da Computação e Matemática Computacional) – Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, São Carlos.
- (36) Pinto, M.J. (1999). O problema de corte de estoque inteiro. 98f. Dissertação (Mestrado em Ciências da Computação e Matemática Computacional) – Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, São Carlos.
- (37) Pinto, M.J. (2004). Algumas contribuições à resolução do problema de corte integrado ao problema de seqüenciamento dos padrões. Tese (Doutorado em Computação e Matemática Aplicada) – Instituto Nacional de Pesquisas Espaciais, São José dos Campos.
- (38) Pisinger, D. (2000). A minimal algorithm for the bounded knapsack problem. *INFORMS Journal on Computing*, **12**, 75-84.
- (39) Poldi, K.C. & Arenales, M. (2005). Dealing with small demand in the integer cutting stock problem with limited different stock lengths. Notas do ICMC-USP, n. 85, São Carlos.
- (40) Poltroniere, S.C.; Arenales, M.; Toledo, F.M.G. & Poldi, K.C. (2005). Coupling cutting stock and lot sizing problems in the paper industry. Notas do ICMC-USP, n. 83, São Carlos.
- (41) Riehme, J.; Scheithauer, G. & Terno, J. (1996). The solution of two-stage guilhotine cutting stock problems having extremely varying order demands. *European Journal of Operational Research*, **91**, 543-552.
- (42) Silveira, R. & Morabito, R. (2002). Um método heurístico baseado em programação dinâmica para o problema de corte bidimensional guilhotinado restrito. *Gestão & Produção*, **9**(1), 78-92.
- (43) SICUP (2005). *Special Interest Group on Cutting and Packing*. Disponível em: <<http://www.apdio.pt/sicup/>>.
- (44) Soma, N.Y. & Toth, P. (2002). An exact algorithm for the subset sum problem. *European Journal of Operational Research*, **136**, 57-66.
- (45) Steuer, R.E. (1986). *Multiple criteria optimization: theory, computation & application*. John Wiley, New York.
- (46) Sweeney, P. & Paternoster, E. (1992). Cutting and packing problems: a categorized, application-oriented research bibliography. *Journal of the Operational Research Society*, **43**, 691-706.
- (47) Vance, P.; Barnhart, C.; Johnson, E. & Nemhauser, G. (1994). Solving binary cutting stock problems by column generation and branch-and-bound. *Computational optimization and applications*, **3**, 111-130.
- (48) Yanasse, H.H. (1996a). Minimization of open orders – polynomial algorithms for some special cases. *Pesquisa Operacional*, **16**, 1-26.
- (49) Yanasse, H.H. (1996b). A transformation for solving a pattern sequencing problem in the wood cut industry. INPE/LAC Technical Report 6, 15p, São José dos Campos.

- (50) Yanasse, H. (1997). On a pattern sequencing problem to minimize the maximum number of open stacks. *European Journal of Operational Research*, **100**(3), 454-463.
- (51) Yuen, B. (1991). Heuristics for sequencing cutting patterns. *European Journal of Operational Research*, **55**, 183-190.
- (52) Yuen, B. (1995). Improved heuristics for sequencing cutting patterns. *European Journal of Operational Research*, **87**, 57-64.
- (53) Waescher, G. & Gau, T. (1996). Heuristics for the integer one-dimensional cutting stock problem: a computational study. *Operations Research Spektrum*, **18**, 131-144.
- (54) Wang, P.Y. & Waescher, G. (2002). Cutting and packing. *European Journal of Operational Research*, **41**(2), 239-469.