



VARIANTES DO MÉTODO DE GERAÇÃO DE COLUNAS DE GILMORE E GOMORY

João Gabriel Oliveira Marques

Faculdade de Ciências Aplicadas, UNICAMP
Limeira - SP

joao.marques@fca.unicamp.br

Washington Alves de Oliveira

Faculdade de Ciências Aplicadas, UNICAMP
Limeira, SP

washington.oliveira@fca.unicamp.br

Antonio Carlos Moretti

Faculdade de Ciências Aplicadas, UNICAMP
Limeira, SP

moretti@ime.unicamp.br

Mateus Pereira Martin

Departamento de Engenharia de Produção, UFSCar
São Carlos - SP

martin@ufscar.br

RESUMO

Um dos mais notáveis avanços na área dos problemas de corte e empacotamento ocorreu na década de 1960, quando Gilmore e Gomory desenvolveram o método de geração de colunas, que se mostrou uma estratégia de resolução eficiente para o problema. Entretanto, estes mesmos autores sugerem que há possíveis melhorias, por exemplo gerando colunas de maior impacto no problema mestre. Assim, este trabalho estuda variações no modelo da mochila do método de geração de colunas para o problema de corte de estoque unidimensional relaxado, sem avaliar técnicas para obtenção da solução inteira. Os estudos computacionais realizados mostram que algumas abordagens obtiveram soluções gerando menor número de colunas em comparação com o método tradicional para algumas instâncias, enquanto para outras abordagens mais colunas foram geradas mas em menor tempo computacional total. O trabalho ainda prevê margem para futuros estudos com problemas integrados ou com outras aplicações do método de geração de colunas.

PALAVRAS CHAVE. Geração de colunas. Corte de estoque unidimensional. Desempenho.

PM - Programação Matemática

ABSTRACT

One of the most remarkable advances for cutting and packing problems occurred in the 1960s, when Gilmore and Gomory developed the column generation method, which proved to be an effective and efficient solution strategy, albeit needing heuristics to get an integer solution. However, since this seminal work little has been studied about modifications in the proposed method seeking improvement in performance for the one-dimensional cutting stock problem. This work studies modifications in the knapsack model of the column generation method for the relaxed one-dimensional cutting stock problems. The computational studies performed show that this approach obtained solutions to the problem generating smaller number of columns compared to the traditional method for some instances, whereas for other instances more columns were generated but in less total computational time. The work still provides scope for future studies with larger problems or with other applications of the method of generating columns.

KEYWORDS. Column Generation. One-dimensional cutting stock problem. Performance.

MP - Mathematical Programming



1. Introdução

Os Problemas de Corte e Empacotamento (PCE) se referem a qualquer problema matemático que consista na repartição de um objeto maior em objetos menores, seja para o fim de corte do objeto maior, seja para alocação dos objetos menores em um espaço maior delimitado. Em vista disso, costuma-se trabalhar com 3 tipos principais de problemas diferenciados pela quantidade de dimensões variáveis (Arenales et al. 2007, Wäscher et al. 2007). Os problemas unidimensionais, por exemplo, são considerados nas operações industriais de corte de rolos de papel, bobinas de metal, barras de aço e afins. Os bidimensionais podem ser aplicados nos cortes de madeira, aço ou até mesmo de moldes de roupa em tecido. E os tridimensionais aparecem frequentemente na forma de problemas de posicionamento de contêineres, organização de pallets, entre outros. A Figura 1 ilustra os padrões de corte para uma instância de um problema unidimensional de objetos de tamanho únicos e três itens menores demandados. Essa figura destaca os padrões homogêneos que são aqueles que apresentam um único tipo de item.

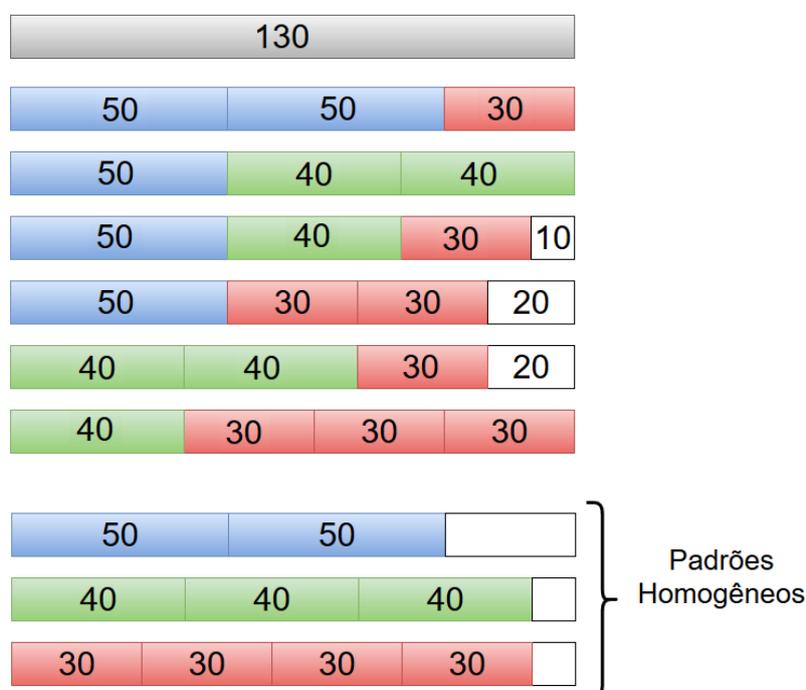


Figura 1: Árvore de busca dos possíveis padrões de corte

Um grande marco para os PCE foram os trabalhos de Gilmore e Gomory da década de 1960, que desenvolveram o chamado "método de geração de colunas", uma abordagem capaz de obter soluções satisfatórias para problemas de corte de estoque em tempo computacional factível (Gilmore & Gomory 1961, 1963, 1965). É importante destacar que Ford Jr & Fulkerson (1958) e Dantzig & Wolfe (1960) propuseram essa técnica no contexto de outros problemas. A formulação proposta por Gilmore & Gomory (1961) para o problema de corte de estoque unidimensional pode ser compreendida como uma decomposição de Dantzig-Wolfe na formulação de Kantorovich (1960). O método ficou amplamente conhecido na comunidade científica, que o adota até hoje. A partir das muitas pesquisas que procuravam aperfeiçoar o modelo de corte para aumentar sua proximidade com a realidade da indústria foram desenvolvidas, entretanto muito pouco se explorou sobre o método de geração de colunas em si. Destacamos a sugestão encontrada em Gilmore & Gomory (1963):



The whole purpose of the knapsack calculation is to find a column representing a cutting pattern that will yield an improvement; in fact, by maximizing the knapsack calculation we find the column that will yield the most improvement per unit increase in the associated variable x_j (call this amount s_j). However, this criterion pays no attention at all to the question of how big x_j will be when the pivot step is completed, and the improvement in the objective function depends on this as the improvement is the product $x_j s_j$. We have found that much too often in our calculations a large s_j led only to a small improvement, while a much smaller s_j , but associated with a large change in x_j , led to a larger improvement. [Gilmore, P.C., Gomory R.E.; A linear programming approach to the cutting stock problem - part II, Operations Research, 1963, vol. 11, No. 6. pp. 863-888].

Este trabalho estuda o método de geração de colunas de Gilmore e Gomory para resolução de problemas de corte unidimensional, com foco em como o subproblema da mochila pode ser modificado de forma a se obter “melhores” resultados. É importante notar que aqui não serão avaliadas técnicas de aproximação para obtenção da solução inteira. Diferentes abordagens para o modelo matemático do subproblema da mochila são propostas visando obter colunas mais eficientes para o problema principal. Estudos computacionais são realizados para avaliar a robustez da abordagem proposta, analisando-se seus respectivos impactos no número total de iterações e na obtenção de soluções. A estrutura do artigo é como segue. A Sessão 2 apresenta o método tradicional de geração de colunas para o problema estudado. A Sessão 3 desenvolve as propostas de aceleração do método. A Sessão 4 exhibe os estudos computacionais realizados. Por fim, a Sessão 5 discute os comentários finais.

2. O problema de corte de estoque unidimensional e geração de colunas

O problema de corte de estoque unidimensional pode ser enunciado da seguinte maneira: deseja-se cortar a menor quantidade de objetos mestre R em m tipos de itens menores de tamanho l_i , $i \in \{1, \dots, m\}$, para atender uma demanda d_i de cada tipo de item. A formulação de Gilmore & Gomory (1961) associa uma variável de decisão para cada padrão de corte possível, isto é, diante de n padrões de corte diferentes, onde cada padrão j gera uma quantidade a_{ij} de itens de tamanho l_i . Observando-se o Modelo (1) do problema enunciado, nota-se que a matriz A dos padrões de corte pode ser muito grande dependendo dos valores que compõe o problema sob análise. O método de geração de colunas busca contornar essa explosão combinatorial.

$$P_o \left\{ \begin{array}{l} \text{Min} \quad z = \sum_{j=1}^n x_j \\ \text{sujeito a} \\ \sum_{j=1}^n a_{ij} x_j \geq d_i, \text{ para } i = 1, 2, \dots, m, \\ x \geq 0 \text{ e inteiros.} \end{array} \right. \quad (1)$$

Sabendo-se que a^j , uma coluna de A , representa um padrão de corte e decompondo-se a matriz A e o vetor c nas colunas que compõe e que não compõe a base ($A = [B|N]$ e $c = [c_B|c_N]$), em qualquer ponto extremo de um problema de programação linear de minimização o valor da função objetivo na base B é dado por $z = c_B^t B^{-1} b - (c_B^t B^{-1} a^j - c_j)$, tem-se no ponto ótimo que o segundo termo é menor ou igual a 0, *i.e.* não existe nenhuma outra coluna que, se entrar na base, reduza o valor de z . Assim, utilizando-se apenas k colunas para o problema (1), este resultado será o ótimo do problema se, e somente se, não existir um a^{k+1} tal que $c_B^t B^{-1} a^{k+1} - 1 > 0$ (Bazaraa et al. 2004).



Daí, simplifica-se dizer que o método de geração de colunas consiste em um algoritmo iterativo que ataca dois modelos matemáticos, o P_o em (1) e o P_M em (2), onde $w = c_B^t B^{-1}$, i.e. o custo reduzido das variáveis de folga do problema P_o (Gilmore & Gomory 1963). Note que P_M consiste em um problema da mochila, no qual l_i representa o comprimento do item i e L representa o comprimento do rolo a ser cortado.

$$P_M \left\{ \begin{array}{l} \text{Max} \quad z_M = w^t a - 1 = \sum_{i=1}^m w_i a_i - 1 \\ \text{sujeito a} \quad \sum_{i=1}^m l_i a_i \leq L \\ \quad \quad \quad a_i \geq 0 \text{ e inteiros.} \end{array} \right. \quad (2)$$

O critério de parada do algoritmo é alcançado quando $z_M \leq 0$ em (2), o que significa que não existe um padrão a^j que, se adicionado no modelo, entraria na base do problema P_o .

3. Modificações no método de Gilmore e Gomory

A fim de melhorar o método tradicional foram feitas modificações no sub-problema da mochila em (2), como sugerido pelos próprios autores do método. A proposta de Gilmore e Gomory para o sub-problema original é a busca por qualquer coluna cujo custo reduzido seja o maior possível; entretanto isso não garante que a melhora na função objetivo será a melhor possível, uma vez que essa melhora é dada pelo produto do custo reduzido pelo teste da razão do algoritmo simplex. Dessa forma, utilizando o sub-problema como o Modelo (3), o número de iterações para conclusão do método deve ser menor ou igual à utilização do problema da mochila.

$$P_{M.1} \left\{ \begin{array}{l} \text{Max} \quad z_M = \left(\sum_{i=1}^m w_i a_i - 1 \right) \left(\min_{p=1,2,\dots,m} \left\{ \frac{\bar{b}_p}{\bar{a}_p} : \bar{a}_p > 0 \right\} \right) \\ \text{sujeito a} \quad \sum_{i=1}^m l_i a_i \leq L \\ \quad \quad \quad a_i \geq 0 \text{ e inteiros} \end{array} \right. \quad (3)$$

Note que $(\min_{p=1,2,\dots,m} \{ \frac{\bar{b}_p}{\bar{a}_p} : \bar{a}_p > 0 \})$ representa o teste da razão do algoritmo simplex, onde $\bar{b}_p = \sum_{j=1}^m B_{pj}^{-1} b_j$ e $\bar{a}_p = B_p^{-1} a = \sum_{j=1}^m B_{pj}^{-1} a_j$. Evidentemente a característica não-linear do modelo $P_{M.1}$ em (3) torna seu uso inviável na prática, entretanto deste modelo pode-se derivar melhores ideias para o desenvolvimento de outras variações. Daí criou-se o primeiro modelo $P_{M.2a}$ em (4), abaixo.

$$P_{M.2a} \left\{ \begin{array}{l} \text{Max} \quad z_M = \sum_{i=1}^m w_i a_i - \max_{p=1,2,\dots,m} \left\{ \frac{\bar{a}_p}{\bar{b}_p} \right\} \\ \text{sujeito a} \quad \sum_{i=1}^m l_i a_i \leq L \\ \quad \quad \quad \sum_{i=1}^m w_i a_i \geq 1 + \varepsilon \\ \quad \quad \quad a_i \geq 0 \text{ e inteiros.} \end{array} \right. \quad (4)$$

Note que, para garantir que a coluna gerada irá melhorar o modelo de corte, é adicionada a restrição $w^t a \geq 1 + \varepsilon$ no modelo, fazendo com que o critério de parada seja a infactibilidade do modelo.



Esse modelo apresenta uma não-linearidade na função objetivo, que pode ser facilmente sanada. Sabendo que $\bar{t} = \max_{p=1,2,\dots,m} \left\{ \frac{\bar{a}_p}{\bar{b}_p} \right\}$ deve ser respeitado, é possível chegar no modelo $P_{M.2b}$ em (5), onde \bar{t} é variável, mas certamente assumirá valor igual a $\max_{p=1,2,\dots,m} \left\{ \frac{\bar{a}_p}{\bar{b}_p} \right\}$.

$$P_{M.2b} \left\{ \begin{array}{l} \text{Max} \quad z_M = \left(\sum_{i=1}^m w_i a_i \right) - \bar{t} \\ \text{sujeito a} \quad \sum_{i=1}^m l_i a_i \leq L \\ \sum_{i=1}^m w_i a_i \geq 1 + \varepsilon \\ \bar{t} \sum_{j=1}^m B_{pj}^{-1} b_j \geq \sum_{j=1}^m B_{pj}^{-1} a_j \text{ para } p = 1, 2, \dots, m \\ a_i \geq 0 \text{ e inteiros} \\ \bar{t} \geq 0. \end{array} \right. \quad (5)$$

Este modelo faz uma grotesca aproximação da multiplicação por uma soma para contornar a não linearidade. Mas é possível diminuir o erro dessa aproximação fazendo-se a combinação linear convexa desta: $\lambda(w^T a) + (1 - \lambda)(-\bar{t})$ com $\lambda \in [0, 1]$. Assim chega-se ao modelo $P_{M.3}$ em (6), para o qual escolhe-se um λ a cada iteração. É importante notar que este modelo, se corretamente parametrizado, certamente realizará um número menor ou igual de iterações que o modelo original de Gilmore e Gomory, uma vez que tomando $\lambda = 1$, ambos os modelos tornam-se exatamente iguais.

$$P_{M.3} \left\{ \begin{array}{l} \text{Max} \quad z_M = \lambda \sum_{i=1}^m w_i a_i - (1 - \lambda) \bar{t} \\ \text{sujeito a} \quad \sum_{i=1}^m l_i a_i \leq L \\ \sum_{i=1}^m w_i a_i \geq 1 + \varepsilon \\ \bar{t} \sum_{j=1}^m B_{pj}^{-1} b_j \geq \sum_{j=1}^m B_{pj}^{-1} a_j \text{ para } p = 1, 2, \dots, m \\ a_i \geq 0 \text{ e inteiros} \\ \bar{t} \geq 0. \end{array} \right. \quad (6)$$

Outras duas variações foram propostas a partir de outras ideias. A variação $P_{M.4}$ em (7) considera a hipótese de que, se a real intenção do algoritmo for diminuir o tempo de obtenção de uma solução para o problema relaxado (não necessariamente diminuir o número de colunas geradas), gerar colunas que respeitem apenas as condições de factibilidade sem tentar otimizar uma função pode ser uma boa alternativa para o modelo original.

$$P_{M.4} \left\{ \begin{array}{l} \text{Max} \quad z_M \\ \text{subject to:} \quad \sum_{i=1}^m l_i a_i \leq L \\ \sum_{i=1}^m w_i a_i > 1 + \varepsilon \\ a_i \geq 0 \text{ e inteiros, } i = 1, \dots, m \end{array} \right. \quad (7)$$



Por fim a variação $P_{M.5}$ em (8) busca gerar padrões que sejam similares as quantidades demandadas de cada tipo de item, o que em teoria poderia gerar padrões mais adequados a cada problema em específico, como se o modelo fosse auto-parametrizável para cada problema.

$$P_{M.5} \left\{ \begin{array}{l} \text{Max} \quad \sum_{i=1}^m d_i a_i \\ \text{subject to:} \quad \sum_{i=1}^m l_i a_i \leq L \\ \quad \quad \quad \sum_{i=1}^m w_i a_i > 1 + \varepsilon \\ \quad \quad \quad a_i \geq 0 \text{ e inteiros, } i = 1, \dots, m \end{array} \right. \quad (8)$$

Esses modelos são avaliados na próxima sessão por meio de experimentos computacionais.

4. Estudos computacionais

Os estudos computacionais para avaliar a robustez dos modelos propostos consideram as 18 classes de instâncias propostas por (Foerster & Wascher 2000), e utilizados por (Yanasse & Limeira 2006), (Moretti & Salles Neto 2008), entre outros autores. O gerador de instâncias desenvolvido por Gau e Wäscher (Gau & Wäscher 1995) disponível em <https://paginas.fe.up.pt/~esicup/problem_generators?category_id=1> foi utilizado. Essas classes de instâncias variam segundo: o tamanho dos itens demandados (l_i), o tamanho do objeto cortado (L) e a demanda dos itens (d_i). Para cada classe foram utilizadas 100 amostras de problemas diferentes e medidos o número de iterações e tempo computacional para completude do problema. As variações propostas na Seção 3, bem como o modelo original da Seção 2, foram implementados em ambiente MATLAB com utilização da biblioteca do CPLEX como resolvidor, e executados em ambiente de processador AMD FX6300 (3.50 GHz), de 16 GB de memória RAM.

A Tabela 1 apresenta os resultados obtidos, os quais são comparados pela média de colunas geradas e tempo computacional para cada classe de problema. Os melhores resultados das variações são destacados em azul, em contraposição aos melhores resultados do método original, destacados em vermelho.

Os resultados deixam claro que duas variações se destacam, as $P_{M.3}$ e $P_{M.4}$, a primeira gerou menos colunas em 11 das 18 classes, já a segunda possuiu menor tempo computacional em 12 das 18 classes. Ademais infere-se que o $P_{M.5}$ não possui um único teste em que tenha se saído melhor e percebe-se que o $P_{M.2b}$ foi levemente pior que o $P_{M.3}$, o que indica que a modificação de um para o outro (o uso do multiplicador λ) foi uma melhoria.

A Figura 2 apresenta os resultados de dois modelos: $P_{M.3}$ e $P_{M.4}$. As coordenadas de cada ponto são dadas pelo valor relativo do resultado em relação ao modelo original P_M , ou seja, quanto menor o valor no gráfico, melhor foi o resultado do modelo especificado para aquela classe de problemas. Assim, a Figura 2 deixa nítida a contraposição das duas variações, dado que a maioria dos pontos de cada uma se encontra em quadrantes opostos.

Note que uma variação ótima seria aquela se encontrasse no quadrante inferior direito, tanto sendo mais rápida quanto gerando menos colunas. Entretanto permanece ainda a dúvida de qual critério é mais interessante para o problema, conforme será discutido na próxima sessão.



Tabela 1: Resultados dos testes computacionais
Tempo Computacional (s) Colunas Geradas

	P_M	$P_{M.2b}$	$P_{M.3}$	$P_{M.4}$	$P_{M.5}$	P_M	$P_{M.2b}$	$P_{M.3}$	$P_{M.4}$	$P_{M.5}$
Classe 01	0.50	1.18	0.81	0.72	1.25	17.96	33.32	25.48	51.24	48.80
Classe 02	0.48	1.17	0.69	0.70	1.31	17.94	30.76	20.12	49.80	49.08
Classe 03	0.93	4.38	1.99	1.26	3.67	29.54	85.82	44.94	95.78	95.16
Classe 04	0.92	4.82	2.08	1.26	4.03	29.70	83.94	38.00	96.14	96.02
Classe 05	1.92	12.44	6.42	2.09	8.41	51.12	173.20	79.34	173.20	173.20
Classe 06	1.89	24.68	30.73	2.10	9.97	50.54	172.04	70.50	172.04	172.04
Classe 07	0.15	0.20	0.20	0.13	0.38	8.68	8.78	8.62	12.26	22.26
Classe 08	0.15	0.16	0.19	0.12	0.40	8.78	7.46	7.56	12.16	22.44
Classe 09	0.38	0.55	0.51	0.33	1.28	17.78	19.08	18.08	32.58	58.88
Classe 10	0.49	0.57	0.52	0.37	1.83	20.84	18.44	17.66	37.28	69.48
Classe 11	1.77	3.49	2.54	1.23	6.50	53.06	56.46	48.52	107.44	182.64
Classe 12	1.74	2.54	2.04	1.22	6.79	51.88	46.00	43.50	107.52	180.24
Classe 13	0.10	0.11	0.13	0.08	0.14	7.22	5.84	5.88	7.12	8.68
Classe 14	0.10	0.11	0.12	0.07	0.14	7.24	5.68	5.66	6.98	8.46
Classe 15	0.31	0.37	0.39	0.19	0.55	19.30	14.08	13.96	19.66	26.28
Classe 16	0.30	0.35	0.36	0.19	0.58	19.36	13.70	13.64	18.96	27.02
Classe 17	1.09	1.65	1.42	0.50	3.61	56.38	37.44	34.68	51.02	101.94
Classe 18	1.11	1.39	1.24	0.52	3.86	56.14	35.18	34.56	52.08	104.84

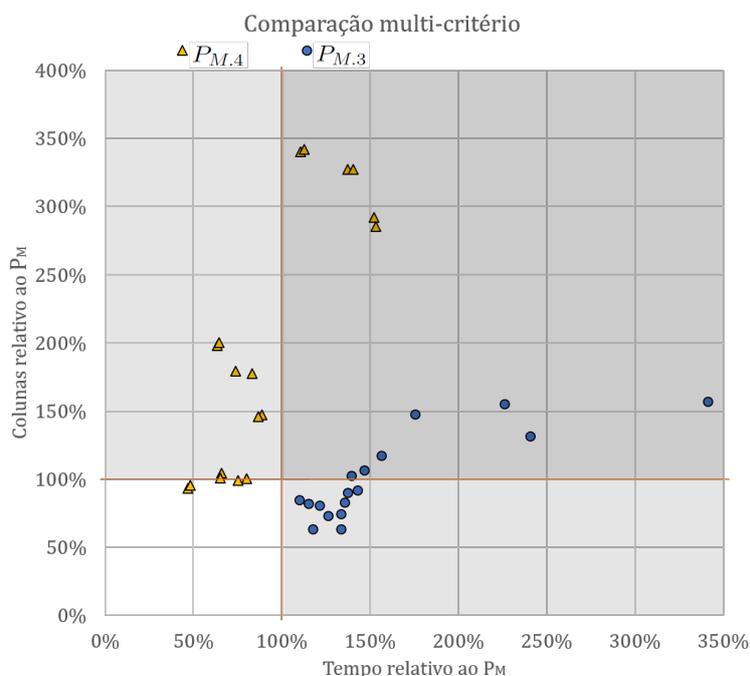


Figura 2: Comparação entre os métodos propostos



5. Conclusões e perspectivas futuras

Os resultados mostram que é bastante viável a aplicação de um sub-problema diferente para a geração de colunas de problemas de corte de estoque unidimensional. Entretanto há um *trade-off* nessa escolha entre tempo computacional e quantidade de colunas geradas. A um trabalho que se interessar por produzir um número menor de colunas, a variação $P_{M,3}$ em (6) pode ser mais interessante, enquanto para aqueles que buscam ganho computacional, a variação $P_{M,4}$ em (7) é mais atrativa.

É importante lembrar que para a resolução do problema de corte de estoque não relaxado, *i.e.* com variáveis inteiras, uma maior quantidade de colunas acarreta em um tempo computacional maior, mas uma quantidade menor de colunas pode acarretar num resultado mais afastado do ótimo real. Sendo assim, ainda permanece a dúvida de qual método traria mais benefícios no final. Esta questão em aberto pode ser explorada mais a fundo com a resolução do modelo de corte inteiro após a geração de colunas por cada variação.

Agradecimentos

Os autores agradecem o apoio financeiro da CAPES, FAPESP e FAEPEX-UNICAMP.

Referências

- Arenales, M., Armentano, V., Morabito, R. & Yanasse, H. (2007), *Pesquisa Operacional: para cursos de engenharia*, Rio de Janeiro: Elsevier-Campus.
- Bazaraa, M. S., Jarvis, J. J. & Sherali, H. D. (2004), *Linear Programming and Network Flows*, Wiley-Interscience.
- Dantzig, G. B. & Wolfe, P. (1960), 'Decomposition principle for linear programs', *Operations research* **8**(1), 101–111.
- Foerster, H. & Wascher, G. (2000), 'Pattern reduction in one-dimensional cutting stock problems', *International Journal of Production Research* **38**(7), 1657–1676.
- Ford Jr, L. R. & Fulkerson, D. R. (1958), 'Constructing maximal dynamic flows from static flows', *Operations research* **6**(3), 419–433.
- Gau, T. & Wäscher, G. (1995), 'CUTGEN1: A problem generator for the standard one-dimensional cutting stock problem', *European Journal of Operational Research* **84**(3), 572–579.
- Gilmore, P. C. & Gomory, R. E. (1961), 'A linear programming approach to the cutting-stock problem', *Operations Research* **9**(6), 849–859.
- Gilmore, P. C. & Gomory, R. E. (1963), 'A linear programming approach to the cutting stock problem—part ii', *Operations research* **11**(6), 863–888.
- Gilmore, P. & Gomory, R. (1965), 'Multistage cutting stock problems of two and more dimensions', *Operations Research* **13**, 94–120.
- Kantorovich, L. (1960), 'Mathematical methods of organizing and planning production', *Management Science* **6**, 366–422.
- Moretti, A. C. & Salles Neto, L. L. (2008), 'Nonlinear cutting stock problem model to minimize the number of different patterns and objects', *Computational & Applied Mathematics* **27**(1), 61–78.
- Wäscher, G., Haußner, H. & Schumann, H. (2007), 'An improved typology of cutting and packing problems', *European Journal of Operational Research* **183**(3), 1109–1130.
- Yanasse, H. H. & Limeira, M. S. (2006), 'A hybrid heuristic to reduce the number of different patterns in cutting stock problems', *Computers & Operations Research* **33**(9), 2744–2756.