



UM MÉTODO HEURÍSTICO BASEADO EM PROGRAMAÇÃO DINÂMICA PARA O PROBLEMA DE CORTE BIDIMENSIONAL GUILHOTINADO RESTRITO

Rejane Joas Silveira
Reinaldo Morabito

Departamento de Engenharia de Produção
Universidade Federal de São Carlos
São Carlos – SP
E-mails: rejane_silveira@bol.com.br
morabito@power.ufscar.br

Resumo

Neste artigo estudamos um caso particular dos problemas de corte, denominado problema bidimensional guilhotinado restrito (PGR). O PGR é um problema NP-difícil que aparece em diversos processos industriais de corte de chapas retangulares, em particular, na indústria de vidro e placas de circuito impresso. Para resolvê-lo, exploramos uma variação do método exato de CHRISTOFIDES & HADJICONSTANTINO (1995), baseada numa relaxação do espaço de estados de uma formulação de programação dinâmica do PGR, num procedimento do tipo otimização do subgradiente, e numa heurística de factibilização. O resultado é um método sem garantia de otimalidade, porém bem mais rápido e capaz de resolver problemas maiores do que o método exato de Christofides e Hadjiconstantinou. O desempenho computacional do método é avaliado resolvendo-se diversos exemplos da literatura e exemplos aleatórios, e comparando-se as soluções obtidas com as de CHRISTOFIDES & HADJICONSTANTINO (1995) e da conhecida heurística de WANG (1983).

Palavras-chave: *problema de corte, padrões de corte bidimensionais guilhotinados restritos, programação dinâmica, heurísticas.*

1. Introdução

O problema de corte consiste em determinar a “melhor” forma de cortar unidades de

material (aqui denominadas *objetos*), de maneira a produzir um conjunto de unidades menores (*itens*). Esse problema aparece em diversos processos industriais de corte onde os objetos,

em geral disponíveis em estoque, correspondem a barras de aço, bobinas de papel e alumínio, placas metálicas e de madeira, chapas de vidro e fibra de vidro, peças de couro, etc., e os itens, com dimensões especificadas, são em geral encomendados através de uma carteira de pedidos de clientes. Centenas de artigos foram publicados nas literaturas de pesquisa operacional e gestão da produção tratando problemas de corte, conforme mostram os exames e edições especiais em DYCKHOFF (1992), DOWSLAND & DOWSLAND (1992), DYCKHOFF & FINKE (1992), SWEENEY & PATERNOSTER (1992), MORABITO & ARENALES (1992), BISCHOFF & WAESCHER (1995), MUKHACHEVA (1997), DYCKHOFF *et al.* (1997), e ARENALES *et al.* (1999).

No presente artigo, tratamos um caso particular dos problemas de corte, denominado *problema bidimensional guilhotinado restrito* (PGR). Nossa motivação para estudá-lo é que, além de ser um problema de difícil solução exata, ele é importante em diversos processos industriais de corte de chapas retangulares, por exemplo, no corte de chapas de fibra de vidro (objetos) para produção de placas de circuito impresso (itens) encomendadas por clientes (SILVEIRA, 1999). O PGR é *bidimensional* porque envolve duas dimensões relevantes para a solução (comprimento e largura dos objetos e itens), é *guilhotinado* porque, devido às restrições impostas pelos equipamentos, os cortes devem ser do tipo guilhotinado (um corte é guilhotinado se, ao ser realizado sobre um retângulo, produzir dois retângulos), e é *restrito* porque há limitações para o número máximo de vezes que um item poderá ser cortado a partir de um objeto, além das limitações impostas pelas dimensões físicas do objeto.

Ao tratar o PGR, a maioria dos autores propôs métodos aproximados. Exemplos disso podem ser encontrados em WANG (1983), que propôs uma heurística que combina itens em arranjos horizontais e verticais para formar padrões de corte; em VASKO (1989) e OLIVEIRA & FERREIRA (1990), que apresentaram refinamentos do algoritmo de Wang; em MORABITO

& ARENALES (1996), que propuseram uma abordagem baseada numa busca em um grafo-e/ou, em MORNAR & KHOSHNEVIS (1997), que adaptaram o algoritmo de Wang para uma aplicação particular de cortes de placas de circuito impresso, e em FAYARD *et al.* (1998), que apresentaram um algoritmo baseado em problemas da mochila unidimensionais para combinar faixas horizontais e verticais dentro de sub-retângulos.

Poucos autores apresentaram métodos exatos para o PGR, entre eles, VISWANATHAN & BAGCHI (1993), que propuseram um procedimento de busca em árvore baseado na estratégia de busca o melhor primeiro; HIFI (1997), que sugeriu um melhoramento no algoritmo de Viswanathan e Bagchi; CHRISTOFIDES & WHITLOCK (1977), que apresentaram um algoritmo de busca em árvore que utiliza uma rotina baseada no problema de transporte para obter limitantes durante a busca; e CHRISTOFIDES & HADJICONSTANTINO (1995), que também propuseram um algoritmo de busca em árvore, porém utilizando limitantes derivados de uma relaxação do espaço de estados de uma formulação de programação dinâmica, e um procedimento do tipo otimização do subgradiente (ascensão no espaço de estados). Tal método é mais eficaz do que o de CHRISTOFIDES & WHITLOCK (1977). CHRISTOFIDES & HADJICONSTANTINO (1995) não chegaram a aplicar este método para resolver exemplos reais de processos de corte, mas apenas para resolver alguns exemplos da literatura e exemplos gerados aleatoriamente, todos de tamanho moderado.

Neste artigo exploramos uma variação do método de CHRISTOFIDES & HADJICONSTANTINO (1995), onde, ao invés de aplicarmos o algoritmo de busca em árvore, optamos por desenvolver uma simples heurística de factibilização que, em cada iteração do procedimento de otimização do subgradiente, utiliza a solução relaxada de programação dinâmica para tentar acentuar a melhor solução factível obtida até então. O resultado é um método sem garantia de otimalidade, porém,

bem mais rápido e capaz de resolver problemas maiores do que o método exato de CHRISTOFIDES & HADJICONSTANTINO (1995). O desempenho computacional do método é ilustrado resolvendo-se alguns exemplos da literatura e exemplos aleatórios, e comparando-se as soluções obtidas com as soluções ótimas do método de CHRISTOFIDES & HADJICONSTANTINO (1995) e com as soluções produzidas por métodos heurísticos, entre eles, o algoritmo de Wang (esta é a heurística mais referenciada na literatura para resolver o PGR).

Os resultados destes experimentos mostram que uma das vantagens do presente método sobre outros métodos heurísticos é que, nos casos em que ele encontra uma solução ótima, em geral também fornece um certificado de otimalidade. Além disso, nos casos em que a solução obtida é subótima, o método é capaz de produzir um excelente limitante superior para o valor da solução ótima do PGR, o que permite uma boa estimativa do *gap* de otimalidade. Convém salientar que, na maioria dos exemplos aqui analisados, este limitante igualou-se ou ficou muito perto do valor da solução ótima.

Este trabalho está organizado da seguinte maneira: na próxima seção discutimos a modelagem do PGR e o método de solução aqui proposto baseado em CHRISTOFIDES & HADJICONSTANTINO (1995), e na seção 3 analisamos os resultados computacionais do método, comparando-os com os dos outros métodos. Finalmente, na seção 5, apresentamos as conclusões e perspectivas para pesquisa futura.

2. Modelagem do PGR e Método de Solução

Problemas de corte têm sido abordados na literatura principalmente por: (i) programação linear combinada com uma técnica de geração de colunas (padrões de corte) (GILMORE & GOMORY, 1965), e (ii) heurística gulosa de aspiração da demanda, combinada com um procedimento de geração de padrões de corte (HINXMAN, 1980). A abordagem (i) pode ser

apropriada quando o problema de corte é *irrestrito*, isto é, não há limitações para o número máximo de vezes em que um tipo de peça poderá ser cortado a partir de uma placa, além das limitações impostas pelas dimensões físicas da placa. Uma heurística gulosa de aspiração da demanda bastante simples é a chamada *redução exhaustiva repetitiva* (HINXMAN, 1980):

Heurística de aspiração:

Enquanto a demanda de peças não for cumprida, faça-se:

Passo 1: Gere o melhor padrão de corte para o PGR.

Passo 2: Repita este padrão, tanto quanto for possível, em função da demanda das peças. Ou seja, corte no maior número de vezes possível este padrão, levando em conta a demanda atual das peças deste padrão.

Passo 3: Atualize a demanda das peças, ou seja, retire da demanda atual a quantidade de peças que foi cortada no passo 2. Eventualmente, inclua na demanda as peças de novos pedidos.

Podemos verificar que a maior dificuldade da heurística de aspiração se encontra no passo 1, que é o foco deste trabalho. A seguir, descrevemos uma formulação de programação dinâmica para resolver o PGR do passo 1.

2.1 Formulação de programação dinâmica

Seja R_0 o retângulo (chapa) de comprimento e largura $L \times W$, e R um conjunto de n retângulos menores (peças) R_1, R_2, \dots, R_n , com comprimentos e larguras $l_1 \times w_1, l_2 \times w_2, \dots, l_n \times w_n$, demandas b_1, b_2, \dots, b_n , e valores v_1, v_2, \dots, v_n . O PGR consiste em encontrar o padrão de corte guilhotinado para R_0 com o maior valor total possível, usando-se não mais do que b_i réplicas de cada retângulo $R_i, i = 1, 2, \dots, n$ (uma vez que temos a restrição de não gerar estoque, ou seja, podemos produzir no máximo b_i réplicas de cada retângulo R_i). Se todos os valores v_i forem iguais

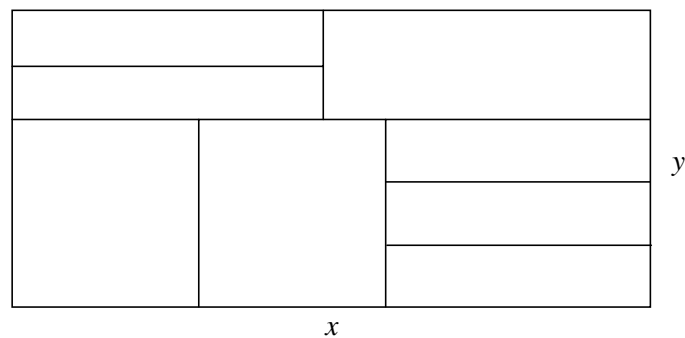


Figura 1 – Exemplo de padrão de corte para (x, y) em 3-estágios

às áreas $l_i w_i$, então o objetivo corresponde a minimizar a perda de material. Admitimos que os comprimentos e larguras da chapa e das peças são números inteiros e, por simplicidade, que peças $l_i \times w_i$ são diferentes de peças $w_i \times l_i$ (casos em que as peças podem ser rotacionadas podem ser tratados por meio de simples modificações nas definições de F_0 em (3) e (7), e X e Y em (11) e (12) adiante).

Para desenvolver uma fórmula recursiva de programação dinâmica, admitimos que os cortes são produzidos em *estágios*. A Figura 1 ilustra um padrão de corte em 3 estágios para um retângulo (x, y) , onde, no primeiro estágio, um corte horizontal divide o retângulo (x, y) em 2 retângulos (superior e inferior); no segundo estágio, um corte vertical divide o retângulo superior em dois retângulos (superior esquerdo e superior direito), e dois cortes verticais dividem o retângulo inferior em três retângulos (inferior esquerdo, inferior intermediário e inferior direito); e, no terceiro estágio, um corte horizontal divide o retângulo superior esquerdo em dois retângulos, e dois cortes horizontais dividem o retângulo inferior direito em três retângulos. Note que os cortes de cada estágio são sempre paralelos aos eixos x ou y , e que os cortes de dois estágios consecutivos são sempre perpendiculares entre si.

Sejam s_1, s_2, \dots, s_n as quantidades de peças dos tipos $1, 2, \dots, n$ que podem ser usadas para produzir um padrão de corte factível para o retângulo (x, y) . Representamos essas peças pela seqüência ordenada:

$$S_{xy} = \{s_1, s_2, \dots, s_n\}$$

Note que, para a chapa (L, W) , podemos usar quaisquer peças do conjunto R , e assim:

$$S_{LW} = \{b_1, b_2, \dots, b_n\}$$

Definimos $F_k(x, y, S_{xy})$ como o valor do padrão de corte ótimo em até k -estágios, para um retângulo de tamanho (x, y) , usando uma combinação factível de retângulos do conjunto S_{xy} , com cortes no primeiro estágio paralelos ao eixo y . Similarmente, definimos $G_k(x, y, S_{xy})$ como o valor do padrão de corte ótimo em até k -estágios, para um retângulo de tamanho (x, y) , usando uma combinação factível de retângulos do conjunto S_{xy} , com cortes no primeiro estágio paralelos ao eixo x . Sejam $X = \{1, 2, \dots, L\}$ e $Y = \{1, 2, \dots, W\}$. As funções recursivas $F_k(x, y, S_{xy})$ e $G_k(x, y, S_{xy})$, para todo $k \geq 1$, $x \in X$, $y \in Y$ e $S_{xy} \subseteq S_{LW}$, são definidas como (CHRISTOFIDES & HADJICONSTANTINOU, 1995):

$$F_k(x, y, S_{xy}) = \max \left[\begin{array}{l} G_{k-1}(x, y, S_{xy}), \\ \max_{\substack{x' < x, \\ x' \in X, \\ S' \subset S_{xy}}} \left[F_k(x', y, S') + G_{k-1}(x - x', y, S_{xy} - S') \right] \end{array} \right] \quad (1)$$

$$G_k(x, y, S_{xy}) = \max \left[\begin{array}{l} F_{k-1}(x, y, S_{xy}), \\ \max_{\substack{y' < y, \\ y' \in Y, \\ S' \subset S_{xy}}} \left[G_k(x, y', S') + F_{k-1}(x, y - y', S_{xy} - S') \right] \end{array} \right] \quad (2)$$

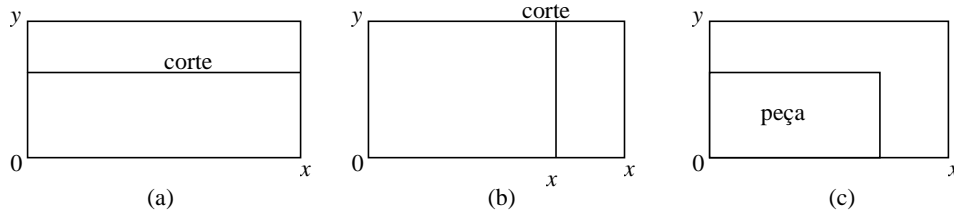


Figura 2 – (a) Nenhum corte neste estágio, mas um corte no estágio anterior paralelo ao eixo x , (b) corte neste estágio paralelo ao eixo y , (c) alocação de uma peça no retângulo (x, y) em $k = 0$

O primeiro termo do colchete da fórmula (1) corresponde ao caso em que, no *primeiro estágio* do padrão ótimo em até k estágios de (x, y) , não há cortes em (x, y) paralelos ao eixo y , o que implica que este padrão pode ser considerado como um padrão em até $k-1$ estágios, com os cortes do primeiro estágio paralelos ao eixo x , como na Figura 2(a). O segundo termo do colchete da fórmula (1) corresponde ao caso em que, no *primeiro estágio* do padrão ótimo em até k estágios de (x, y) , existe pelo menos um corte em (x, y) paralelo ao eixo y , digamos um corte vertical $x' \in X$, $x' < x$, produzindo dois retângulos (x', y) e $(x-x', y)$, como na Figura 2(b). Neste caso, temos dois padrões de corte associados com estes retângulos. No primeiro, as peças pertencem ao conjunto S' , os cortes no *primeiro estágio* do padrão ótimo de (x', y) são paralelos ao eixo y , e ainda há k estágios no total (dado que pode haver outros cortes verticais $x'' < x'$ em (x', y) , que não implicam em outro estágio em (x, y)). No segundo, as peças pertencem ao conjunto complementar $S_{xy} - S'$, os cortes no *primeiro estágio* do padrão ótimo de $(x-x', y)$ são paralelos ao eixo x , e há somente $k-1$ estágios no total (dado que cortes horizontais em $(x-x', y)$ implicam em outro estágio em (x, y)). Similarmente para a fórmula (2).

Um valor para $k=0$ em $F_k(x, y, S_{xy})$ ou $G_k(x, y, S_{xy})$ (fórmulas (1) e (2)) corresponde a uma alocação da maior peça (i.e., mais valiosa) do conjunto S_{xy} para o retângulo (x, y) (ver Figura 2(c)). Os dois cortes para produzir esta peça a partir de (x, y) não são considerados como estágios adicionais de corte. Assim, condições iniciais são fornecidas, para todo x, y e S_{xy} , por:

$$F_0(x, y, S_{xy}) = \max \left[0, \max_{i \in S_{xy}} [v_i | l_i \leq x, w_i \leq y] \right], \quad (3)$$

$$G_0(x, y, S_{xy}) = F_0(x, y, S_{xy}). \quad (4)$$

Note que o máximo entre $F_k(x, y, S_{xy})$ e $G_k(x, y, S_{xy})$ em (1)-(4) corresponde à solução ótima do PGR k -estágios para o retângulo (x, y) , gerado pelo uso das peças em S_{xy} , quando a direção dos cortes do primeiro estágio não é especificada. Em particular, $\max[F_k(L, W, S_{LW}), G_k(L, W, S_{LW})]$ corresponde à solução ótima do PGR k -estágios para a chapa $R_0 = L \times W$. Se não for especificado um número máximo de estágios (i.e., PGR multiestágios), devemos considerar o valor de k como sendo aquele em que:

$$F_k(L, W, S_{LW}) = F_{k+1}(L, W, S_{LW})$$

e

$$G_k(L, W, S_{LW}) = G_{k+1}(L, W, S_{LW})$$

A principal dificuldade desta formulação é a dimensão do espaço de estados associada principalmente à variável de estado S_{LW} .

A seguir, discutimos uma relaxação do espaço de estados da formulação (1)-(4), combinada com um procedimento do tipo otimização do subgradiente (ascensão no espaço de estados), conforme explorado em CHRISTOFIDES & HADJICONSTANTINO (1995), que produz bons limitantes para o problema original. Em seguida, desenvolvemos uma heurística de factibilização que, em cada iteração do procedimento de otimização do subgradiente, utiliza a solução relaxada para produzir boas soluções factíveis para o problema original. Convém salientar que

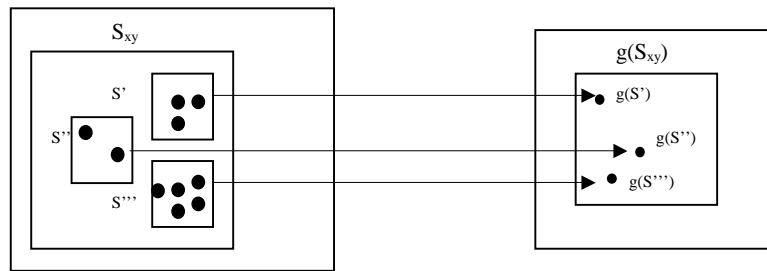


Figura 3 – Representação da relaxação do espaço de estados

CHRISTOFIDES & HADJICONSTANTINO (1995) não exploraram heurísticas de factibilização; ao contrário, aplicaram os limitantes obtidos conforme discussão acima num procedimento de busca em árvore, para garantir encontrar uma solução ótima para o problema.

2.2 Relaxação do espaço de estados

Considere-se a formulação (1)-(4). Seja $g(.)$ uma função de mapeamento escalar inteira, do domínio de estados (x, y, S_{xy}) para algum outro domínio $(x, y, g(S_{xy}))$, com menor cardinalidade. Por exemplo, seja $g(S_{xy}) = \sum_{s_i \in S_{xy}} s_i$; logo, todos os subconjuntos $S' \subset S_{xy}$ com mesmo valor $g(S')$ (i.e., com mesma quantidade *total* de peças disponíveis) são representados por apenas um ponto no novo domínio (Figura 3). Portanto, trata-se de uma relaxação do espaço de estados original.

A fórmula recursiva (1) para o problema de corte pode agora ser reescrita para o novo espaço relaxado, para todo $k \geq 1$, $x \in X$, $y \in Y$ e $g(S_{xy})$, como segue:

$$F_k(x, y, g(S_{xy})) = \max \left[\begin{array}{l} G_{k-1}(x, y, g(S_{xy})), \\ \max_{\substack{x' < x, \\ x' \in X, \\ g(S') < g(S_{xy})}} \left[F_k(x', y, g(S')) + G_{k-1}(x - x', y, g(S_{xy} - S')) \right] \end{array} \right]$$

Similarmente para a fórmula recursiva (2). Como se trata de uma relaxação da formulação (1)-(2), temos que:

$$F_k(x, y, g(S_{xy})) \geq F_k(x, y, S_{xy})$$

e

$$G_k(x, y, g(S_{xy})) \geq G_k(x, y, S_{xy})$$

e assim, $\max [F_k(x, y, g(S_{xy})), G_k(x, y, g(S_{xy}))]$ produz limitantes superiores para a solução ótima do PGR com até k -estágios (uma prova formal disto pode ser encontrada em CHRISTOFIDES *et al.*, 1981). Obviamente, é desejável que a função g seja fácil de ser computada, e que tenha sua contrapartida muito menor do que seu domínio. CHRISTOFIDES & HADJICONSTANTINO (1995) propuseram associar um peso inteiro não negativo q_i a cada peça i no conjunto R , e definiram g como:

$$g(S_{xy}) = q \equiv \sum_{s_i \in S_{xy}} s_i q_i \quad \text{e} \quad g(S') = q' \equiv \sum_{s_i \in S'} s_i q_i$$

Logo, $g(S_{xy} - S')$ pode ser facilmente computada por $q - q'$, e as fórmulas (1)-(4) são reescritas por:

$$F_k(x, y, q) = \max \left[\begin{array}{l} G_{k-1}(x, y, q), \\ \max_{\substack{x' < x, \\ x' \in X, \\ q' = 0, \dots, q}} \left[F_k(x', y, q') + G_{k-1}(x - x', y, q - q') \right] \end{array} \right] \quad (5)$$

$$G_k(x, y, q) = \max \left[\begin{array}{l} F_{k-1}(x, y, q), \\ \max_{\substack{y' < y, \\ y' \in Y, \\ q' = 0, \dots, q}} \left[G_k(x, y', q') + F_{k-1}(x, y - y', q - q') \right] \end{array} \right] \quad (6)$$

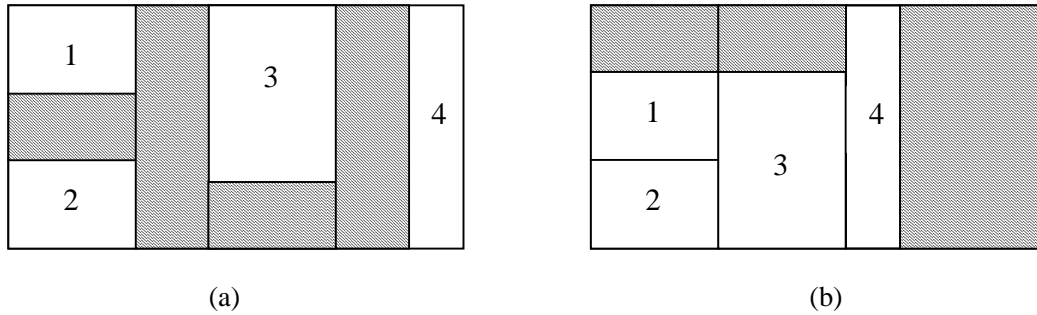


Figura 4 – (a) Padrão de corte não normalizado, (b) padrão de corte normalizado

e inicializadas por

$$F_0(x, y, q) = \max \left[0, \max_{i=1, \dots, n} [v_i \mid l_i \leq x, w_i \leq y, q_i \leq q] \right], \quad (7)$$

$$G_0(x, y, q) = F_0(x, y, q). \quad (8)$$

As recursões (5)-(8), aplicadas para todo $k \geq 1, x \in X, y \in Y$ e $q = 0, \dots, Q$, onde

$$Q = \sum_{b_i \in S_{LW}} b_i q_i, \quad (9)$$

podem ser usadas para obtermos um limitante superior Z_{UB} para o valor da solução ótima do PGR k -estágios, para um dado conjunto de pesos q_1, q_2, \dots, q_n . Este limitante é calculado por:

$$Z_{UB} = \max [F_k(L, W, Q), G_k(L, W, Q)] \quad (10)$$

Note que para $q_1 = 0, q_2 = 0, \dots, q_n = 0$ e k suficientemente grande, o limitante Z_{UB} em (10) corresponde à solução ótima do problema guilhotinado *irrestrito*. A computação de Z_{UB} pode ser reduzida pela restrição aos padrões de cortes *normais* (HERZ, 1972; CHRISTOFIDES & WHITLOCK, 1977). Qualquer padrão de corte é dito normalizado se qualquer peça cortada tem sua margem do lado esquerdo e sua margem inferior adjacentes a outra peça cortada ou às bordas de R_0 , como mostra a Figura 4. Uma consequência disso é que os conjuntos $X = \{1, 2, \dots, L\}$ e $Y = \{1, 2, \dots, W\}$ em (5)-(6) podem ser significativamente reduzidos, sem perda de generalidade, ao serem redefinidos por:

$$X = \left\{ x \mid x = \sum_{i=1}^n \theta_i l_i, 1 \leq x \leq L, 0 \leq \theta_i \leq b_i \right. \\ \left. e \theta_i \text{ inteiro}, i = 1, \dots, n \right\} \quad (11)$$

$$Y = \left\{ y \mid y = \sum_{i=1}^n \tau_i w_i, 1 \leq y \leq W, 0 \leq \tau_i \leq b_i \right. \\ \left. e \tau_i \text{ inteiro}, i = 1, \dots, n \right\} \quad (12)$$

Neste caso, as fórmulas recursivas (5) e (6), aplicadas para todo $k \geq 1, x \in X, y \in Y$ e $q = 0, \dots, Q$ (conforme (9), (11) e (12)), são reescritas por:

$$F_k(x, y, q) = \max \left[\begin{array}{c} G_{k-1}(x, y, q), \\ \max_{\substack{x' < x, \\ x' \in X, \\ q' = 0, \dots, q}} \left[\begin{array}{c} F_k(x', y, q') + \\ + G_{k-1}(\lfloor x - x' \rfloor, y, q - q') \end{array} \right] \end{array} \right] \quad (13)$$

$$G_k(x, y, q) = \max \left[\begin{array}{c} F_{k-1}(x, y, q), \\ \max_{\substack{y' < y, \\ y' \in Y, \\ q' = 0, \dots, q}} \left[\begin{array}{c} G_k(x, y', q') + \\ + F_{k-1}(x, \lfloor y - y' \rfloor, q - q') \end{array} \right] \end{array} \right] \quad (14)$$

onde $\lfloor x \rfloor = \max \{ x_1 \mid x_1 \leq x, x_1 \in X \cup \{0\} \}$, $\lfloor y \rfloor = \max \{ y_1 \mid y_1 \leq y, y_1 \in Y \cup \{0\} \}$ (estas definições são necessárias para garantirmos que o resultado das diferenças $x - x'$ e $y - y'$ em (13) e (14) pertençam aos conjuntos X e Y em (11) e (12)), e $F_k(x, y, q) = G_k(x, y, q) = 0$ se $x = 0$ ou $y = 0$ (note agora que, por causa das

definições de $\lfloor x \rfloor$ e $\lfloor y \rfloor$, podemos ter $x = 0$ ou $y = 0$). As fórmulas (13)-(14) são inicializadas conforme antes, por (7)-(8).

Uma vez computado o valor do limitante Z_{UB} em (10) para um dado conjunto de pesos q_1, q_2, \dots, q_n , e k suficientemente grande, é necessário fazer um *backtracking* para determinar as peças em R que compõem o padrão correspondente (não necessariamente factível) para a chapa R_0 . Isso requer o uso de ponteiros que indicam, para cada conjunto de valores (k, x, y, q) , os termos na recursão (13)-(14) que levaram aos valores de $F_k(x, y, q)$ e $G_k(x, y, q)$. Uma simples rotina *backtracking* pode ser definida para determinar a seqüência de cortes realizados sobre a chapa R_0 , em conjunto com uma estrutura de dados apropriada para representar a lista de peças em R produzidas durante o processo de corte. Seja γ_i o número de vezes que a peça i foi usada neste padrão. Se $\gamma_i \leq b_i$ para todo $i=1, \dots, n$, então uma solução *factível* para o PGR multiestágios foi encontrada, e logo é *ótima*. Caso contrário, um procedimento do tipo otimização do subgradiente pode ser utilizado para otimizar o limitante Z_{UB} em (10), variando-se os pesos q_1, q_2, \dots, q_n .

2.3 Ascensão no espaço de estados

Um método de modificação do espaço de estados, associado com a formulação relaxada (13)-(14), pode ser usado numa tentativa de minimizar o limitante Z_{UB} em (10). O objetivo desta técnica é penalizar as infactibilidades de modo a forçar que a solução do problema relaxado se aproxime cada vez mais da factibilidade. Sejam q o vetor de pesos (q_1, q_2, \dots, q_n) , $f(q)$ o limitante superior Z_{UB} produzido por (10) usando o vetor q , e D a máxima dimensão do espaço de estados relaxado que desejamos modificar (isto é, os maiores valores de Q em (9)). Temos que resolver o seguinte problema:

$$\begin{aligned} f(q^*) &= \min_{q \geq 0} [f(q)] \\ \text{s.a.:} \quad &\sum_{b_i \in S_{LW}} b_i q_i \leq D \end{aligned}$$

A minimização de $f(q)$ no problema acima é, em geral, difícil, dado que $f(q)$ é uma função descontínua em q . Entretanto, um procedimento do tipo “otimização do subgradiente” para modificar os pesos q_i e ascender no espaço de estados pode ser usado (CHRISTOFIDES & HADJICONSTANTINOU, 1995). O procedimento é baseado no seguinte princípio: considere-se γ_i conforme definido anteriormente. Se $\gamma_i \leq b_i$ para todo $i = 1, \dots, n$, então o problema está resolvido; caso contrário, então é razoável tentar reduzir a diferença $\gamma_i - b_i$ para se mover em direção à factibilidade. Um simples modo de se conseguir isso é incrementar o peso q_i das peças para as quais $\gamma_i > b_i$ e, ao mesmo tempo, decrementar o peso q_i das peças para as quais $\gamma_i \leq b_i$.

Neste trabalho utilizamos a seguinte fórmula para atualizar os valores dos pesos q_i :

$$q_i = \begin{cases} q_i + \lfloor t\sqrt{\gamma_i - b_i} \rfloor & \text{se } \gamma_i > b_i, \\ \max[0, q_i - \lfloor t\sqrt{b_i - \gamma_i} \rfloor] & \text{se } \gamma_i \leq b_i, \end{cases} \quad (15)$$

onde $\lfloor z \rfloor$ denota o maior inteiro menor ou igual a z , e t é o tamanho do passo escalar positivo, dado por:

$$t = \max \left[1, \sqrt{\pi(Z_{UB} - Z_{LB}) / \sum_{i=1}^m (b_i - \gamma_i)^2} \right]$$

Note que esta fórmula, similar ao tamanho dos passos na otimização do subgradiente (HELD *et al.*, 1974), é uma pequena modificação da utilizada em CHRISTOFIDES & HADJICONSTANTINOU (1995). Nossa motivação para o limite inferior $t = 1$ é permitir que q_i em (15) seja alterado sempre que $\gamma_i \neq b_i$. O parâmetro π é inicialmente fixado em 2,0, sendo então reduzido pela metade a cada 3 iterações, até que π seja inferior a $\varepsilon = 0,05$ (i.e., após 18 iterações). Z_{LB} é o valor de um limitante inferior (correspondendo a uma solução factível) para a solução ótima do PGR, inicializado por alguma heurística. Z_{UB} é o valor do limitante superior

corrente em (10). Note que, se $Z_{UB} = Z_{LB}$, então o procedimento de otimização do subgradiente pode ser interrompido, com Z_{LB} sendo a solução ótima para o problema original.

Algoritmo de otimização do subgradiente:

Passo 1: Determine X e Y em (11)-(12), e escolha valores iniciais para Z_{LB} e para o vetor q , por exemplo, $q_i = 0$, $i = 1, \dots, n$.

Passo 2: Resolva a formulação relaxada (13)-(14), com k suficientemente grande, para obter o valor ótimo Z_{UB} em (10) relativo ao vetor q atual.

Passo 3: Cheque se o padrão de corte associado com Z_{UB} gera uma solução factível para o PGR. Se sim, então esta solução é ótima e o procedimento é terminado. Caso contrário, se $Z_{UB} < Z_{min}$ (o mínimo limitante superior obtido até então), atualize Z_{min} com $Z_{min} = Z_{UB}$.

Passo 4: Execute uma heurística de factibilização (descrita a seguir) para encontrar uma solução factível para o problema. Se o valor desta solução for maior que Z_{LB} (o valor da melhor solução factível obtida até então), atualize Z_{LB} . Se $Z_{LB} = Z_{min}$, então esta solução é ótima e o procedimento é terminado.

Passo 5: Modifique o vetor q usando a fórmula dada em (15), e atualize Q em (9) e π .

Passo 6: Volte para o passo 2 com o novo vetor q , a menos que um número suficiente de iterações já tenha sido realizado, ou $\pi \leq \epsilon$.

No final do algoritmo, a solução ótima para o PGR pode ter sido encontrada (passo 3 ou 4). Caso contrário, a solução apresentada é a melhor obtida no passo 4 com a heurística de factibilização, que não tem garantia de otimalidade. Em CHRISTOFIDES & HADJICONSTANTINO (1995), não há o passo 4, e o melhor limitante superior Z_{min} obtido pelo algoritmo é usado num procedimento de busca em árvore para resolver otimamente o problema.

2.4 Heurística de factibilização

Uma heurística de factibilização bastante simples é resumida nos seguintes passos:

Passo 1: Retire aleatoriamente do padrão as peças que excederam a demanda atual. Por exemplo, se o padrão contém três peças do tipo 1, e apenas duas peças do tipo 1 são demandadas, então retire aleatoriamente do padrão uma peça do tipo 1.

Passo 2: Para cada “buraco” produzido no passo 1 (i.e., espaço formado pelas peças que foram retiradas do padrão), determine a solução homogênea (arranjo com peças do mesmo tipo) mais valiosa, dentre os tipos de peças que ainda não excederam a demanda. Guarde o valor da melhor solução homogênea obtida para cada buraco.

Passo 3: Preencha o buraco que resultar no maior valor para o padrão. Atualize a demanda das peças utilizadas para preencher este buraco.

Passo 4: Volte para o passo 2 enquanto existirem buracos que ainda possam ser preenchidos por peças demandadas.

Esta heurística é executada no passo 4 do algoritmo de otimização do subgradiente (seção 2.3). Ao término do algoritmo, se tivermos encontrado uma solução factível com valor Z_{min} , então esta solução é ótima. Caso contrário, o algoritmo retorna a melhor solução produzida pela heurística de factibilização ao longo de todas as iterações, junto com o limitante inferior Z_{min} para o valor da solução ótima do PGR.

3. Resultados Computacionais

Nesta seção apresentamos alguns resultados computacionais para ilustrar o desempenho computacional do método proposto na seção 2. Tanto as implementações deste método quanto as do algoritmo proposto em WANG (1983) foram codificadas em linguagem Pascal (compilador

Tabela 1 – Resultados computacionais para 6 exemplos da literatura

Prob	Dados				Wang		Christ			Christ/Fact		Ótima	Melh
	(L, W)	n	X	Y	Sol.	Tempo	Lim.	Sol.	Tempo	Sol.	Tempo	Sol.	%
CW1	(15,10)	7	12	10	244*	< 1	244	244*	< 1	244*	< 1	244	0,0
CW2	(40,70)	10	29	56	2732	< 1	2892	2732	214	2732	219	2892	0,0
CW3	(40,70)	20	25	55	1400	< 1	1860	1400	199	1860*	218	1860	32,9
OF1	(70,40)	10	48	32	2172	< 1	2737	2737*	341	2737*	164	2737	26,0
OF2	(70,40)	10	41	32	2083	< 1	2717	2083	249	2506	206	2690	20,3
WA	(70,40)	20	55	25	2277	< 1	2721	2721*	5	2721*	3	2721	19,5
<i>Média</i>						< 1			168		135		16,5

* solução ótima

Delphi 4 da Borland), e foram rodadas num microcomputador Pentium (350 Mhz, 256 Mbytes de memória RAM).

Dividimos os experimentos desta seção em duas partes: exemplos retirados da literatura (seção 3.1) e exemplos gerados aleatoriamente (seção 3.2). Em todos os exemplos resolvidos pelo algoritmo proposto em WANG (1983), usamos como parâmetro $\beta = 0,03$, ou seja, permitimos uma perda de no máximo 3% dentro dos arranjos horizontais e verticais gerados pelo algoritmo. Convém salientar que este foi o valor que apresentou os melhores resultados em WANG (1983). Como o algoritmo de Wang é suficientemente rápido (conforme é visto a seguir, ele consumiu menos de 1 segundo para resolver cada exemplo) e produz soluções relativamente boas, estas foram utilizadas para inicializar o valor de Z_{LB} no passo 1 do algoritmo de otimização do subgradiente (seção 2.3). Desta maneira, os resultados a seguir indicam quanto o segundo algoritmo é capaz de melhorar as soluções do primeiro.

3.1 Exemplos da literatura

Tomamos 6 exemplos amplamente utilizados na literatura para comparações de métodos de solução para o PGR: os exemplos CW1, CW2 e

CW3 em CHRISTOFIDES & WHITLOCK (1977), OF1 e OF2 em OLIVEIRA & FERREIRA (1990), e WA em WANG (1983). As soluções ótimas destes exemplos são conhecidas na literatura. A Tabela 1 resume os dados dos exemplos e os resultados computacionais obtidos; os demais dados dos exemplos podem ser encontrados em SILVEIRA (1999). As colunas Wang, Christ e Christ/Fact apresentam, respectivamente, as soluções obtidas pela heurística de WANG (1983), pelo algoritmo de otimização do subgradiente de CHRISTOFIDES & HADJICONSTANTINOU (1995) (algoritmo da seção 2.3 sem o passo 4), e pelo algoritmo de otimização do subgradiente junto com a heurística de factibilização (algoritmo da seção 2.3 com o passo 4, e algoritmo da seção 2.4). Os tempos computacionais (coluna Tempo) referem-se a segundos, e o símbolo “< 1” indica um tempo menor que 1 segundo. A coluna Lim apresenta o melhor limitante superior gerado por Chris e Christ/Fact, e as colunas Ótima e Melh apresentam, respectivamente, a solução ótima e a melhoria (em %) da solução de Christ/Fact em relação à solução da heurística de Wang.

Observe na Tabela 1 que: (i) enquanto Wang foi capaz de encontrar a solução ótima apenas de CW1, Christ encontrou a solução ótima de CW1, OF1 e WA, e Christ/Fact a solução ótima de

CW1, *CW3*, *OF1* e *WA*; (ii) *Christ* e *Christ/Fact* produziram limitantes superiores iguais ou muito próximos do valor da solução ótima; (iii) *Christ/Fact* produziu soluções em média 16,5% melhores que *Wang*, porém que demandaram um tempo computacional substancialmente superior (135 segundos em média). Além disso, *Christ/Fact* produziu soluções melhores do que *Christ* nos exemplos *CW3* e *OF2*.

Convém salientar que o método exato de CHRISTOFIDES & HADJICONSTANTINO (1995), baseado no algoritmo de otimização do subgradiente e num procedimento de busca em árvore, foi capaz de encontrar as soluções ótimas de todos os exemplos acima, no entanto os tempos computacionais foram bem elevados (em alguns casos chegaram à ordem de horas num microcomputador IBM 486). Ao compararmos *Christ/Fact* com outros métodos heurísticos da literatura, por exemplo, o método de FAYARD *et al.* (1998), este encontrou soluções inferiores às de *Christ/Fact* para os exemplos *CW2* (2731), *CW3* (1740) e *OF1* (2713), mas superior para o exemplo *OF2* (2586), com tempos computacionais da ordem de poucos segundos numa estação de trabalho Sparc-Server20.

3.2 Exemplos gerados aleatoriamente

Geramos, aleatoriamente, 20 exemplos com chapas de dimensões $(L, W) = (100, 100)$ e $n = 10$ peças, divididos em duas classes: exemplos com peças “grandes” e “pequenas”. Essa classificação tem sido utilizada por alguns autores na literatura, como em BEASLEY (1985), MORABITO & ARENALES (1995, 1996), e HIFI (1997).

- Exemplos com peças *grandes*: as dimensões $l_i, w_i, i=1, 2, \dots, n$, foram sorteadas (e em seguida arredondadas) de uma distribuição uniforme nos intervalos $[0,25L, 0,75L]$, $[0,25W, 0,75W]$, respectivamente.
- Exemplos com peças *pequenas*: as dimensões $l_i, w_i, i=1, 2, \dots, n$, foram sorteadas (e em seguida arredondadas) de uma distribuição uniforme nos intervalos $[0,20L, 0,50L]$, $[0,20W, 0,50W]$, respectivamente.

Em todos os exemplos, os valores de utilidade $v_i, i=1, 2, \dots, n$, foram considerados iguais às áreas $l_i w_i$ das peças, e as demandas d_i foram sorteadas de uma distribuição uniforme no intervalo $[1, 4]$. Os demais dados destes 20 exemplos podem ser encontrados em SILVEIRA (1999). As Tabelas 2 e 3 apresentam os resultados obtidos para os exemplos com peças grandes e pequenas, respectivamente. Como estes exemplos foram gerados aleatoriamente, a solução ótima não é conhecida *a priori*. Portanto, só temos garantia de otimalidade quando a solução gerada em *Christ* ou *Christ/Fact* é factível e tem valor igual ao limitante superior (passos 3 ou 4 do algoritmo da seção 2.3). Nas tabelas, o símbolo “(1)” na coluna *Ótima* indica que o exemplo tem solução ótima desconhecida.

Como podemos observar, para os exemplos com peças grandes (25% a 75% do tamanho da placa original), as soluções apresentadas em *Christ* e *Christ/Fact* são quase sempre ótimas e acompanhadas de um certificado de otimalidade; além disso, são superiores às soluções encontradas em *Wang* (em média mais de 15%, conforme Tabela 2). Note que *Christ/Fact* supera *Christ* apenas no problema 5 (neste caso, não sabemos se a solução é ótima, mas sabemos que, caso não seja, o *gap* de otimalidade é, no máximo, 3,3%), o que sugere que a heurística de factibilização é menos efetiva em problemas com peças grandes.

Para os exemplos com peças pequenas (20% a 50% do tamanho da placa original), as soluções obtidas em *Christ* e *Christ/Fact* nem sempre são acompanhadas de um certificado de otimalidade (note que não sabemos se as soluções dos problemas 1, 2, 3, 6, 8 e 10 na Tabela 3 são ótimas). Isso se deve ao fato de que agora cabem mais peças dentro da placa original (em comparação aos problemas com peças grandes), aumentando, assim, as chances de que o algoritmo de otimização de subgradiente gere soluções infactíveis (i.e., colocando mais peças no padrão de corte do que a demanda requerida). Contudo, as soluções obtidas em *Christ/Fact* ainda são melhores que as encontrados em *Wang*

Tabela 2 – Resultados computacionais para 10 exemplos aleatórios com peças grandes, com $(L,W) = (100,100)$, e $n = 10$.

Prob	Dados		Wang		Christ			Christ/Fact		Ótima	Melh
	X	Y	Sol.	Tempo	Lim.	Sol.	Tempo	Sol.	Tempo	Sol.	%
1	24	30	8438	< 1	8828	8828	3	8828	3	8828	4,6
2	26	13	6768	< 1	8244	8244	< 1	8244	< 1	8244	21,8
3	35	24	6570	< 1	8180	8180	2	8180	2	8180	24,5
4	33	22	8634	< 1	8799	8799	5	8799	5	8799	1,9
5	32	37	8758	< 1	9507	8758	20	9194	16	(1)	5,0
6	40	25	7642	< 1	8464	8464	2	8464	2	8464	10,8
7	27	27	6791	< 1	8751	8751	< 1	8751	< 1	8751	28,9
8	31	35	9536	< 1	9715	9715	< 1	9715	< 1	9715	1,9
9	40	26	5828	< 1	8919	8919	< 1	8919	< 1	8919	53,0
10	35	31	8768	< 1	9611	8768	92	8768	92	(1)	0,0
Média				< 1			12		12		15,2

(1) A solução ótima não é conhecida

Tabela 3 – Resultados computacionais para 10 exemplos aleatórios com peças pequenas, com $(L,W) = (100,100)$ e $n = 10$.

Prob	Dados		Wang		Christ			Christ/Fact		Ótima	Melh
	X	Y	Sol.	Tempo	Lim.	Sol.	Tempo	Sol.	Tempo	Sol.	%
1	51	43	9026	< 1	9446	9026	631	9238	621	(1)	2,3
2	42	43	9022	< 1	9308	9022	68	9216	66	(1)	2,2
3	45	41	8640	< 1	9356	8640	221	8932	137	(1)	3,4
4	58	39	9518	< 1	9682	9682	1	9682	1	9682	1,7
5	55	46	8649	< 1	9381	9381	3	9381	3	9381	8,5
6	53	42	6568	< 1	9148	6568	367	8914	331	(1)	35,7
7	34	50	8648	< 1	9656	9656	< 1	9656	< 1	9656	11,7
8	50	49	9043	< 1	9891	9043	429	9533	427	(1)	5,4
9	41	31	8238	< 1	8964	8964	< 1	8964	< 1	8964	8,8
10	35	53	7958	< 1	9477	7958	188	7958	188	(1)	0,0
Média				< 1			196		178		8,0

(1) A solução ótima não é conhecida

(em média 8%, conforme Tabela 3), apesar de os tempos computacionais serem bem maiores (178 segundos em média). Além disso, nos problemas onde a solução ótima não foi encontrada, *Christ/Fact* produziu soluções em geral melhores do que *Christ* (problemas 1, 2, 3, 6 e 8), indicando que a heurística de factibilização é mais efetiva em problemas com peças pequenas.

4. Conclusões e Perspectivas para Pesquisa Futura

Neste artigo estudamos o problema de corte bidimensional guilhotinado restrito. Para resolvê-lo, exploramos uma variação do método de CHRISTOFIDES & HADJICONSTANTINO (1995), baseada numa relaxação do espaço de

estados de uma formulação de programação dinâmica do PGR (seção 2.2), num procedimento do tipo otimização do subgradiente (seção 2.3), e numa heurística de factibilização (seção 2.4). O desempenho computacional deste método heurístico foi avaliado resolvendo-se 6 exemplos da literatura (seção 3.1) e 20 exemplos aleatórios (seção 3.2), e comparando-se as soluções obtidas com as soluções da conhecida heurística de WANG (1983).

Os resultados mostraram que a abordagem foi capaz de produzir soluções melhores do que a heurística de Wang (com parâmetro $\beta = 0,03$), com melhorias médias de 16,5% nos exemplos da literatura, e 15% e 8% nos exemplos aleatórios com peças grandes e pequenas, respectivamente. Além disso, muitas destas soluções são ótimas e, nos casos em que não são, a abordagem, partindo da solução ótima do problema *irrestrito* (relaxado), foi capaz de produzir um limitante superior relativamente próximo do valor da solução ótima do PGR, o que permite uma boa estimativa do *gap* de otimalidade (nos experimentos este limitante superior coincidiu com o valor da solução ótima em 5 dos 6 exemplos da literatura, em pelo menos 8 dos 10 exemplos aleatórios com peças grandes, e em pelo menos 4 dos 10 exemplos aleatórios com peças pequenas). Uma vantagem deste método é que, nos casos em que ele encontrou uma solução ótima, em geral também forneceu um certificado de otimalidade. Por outro lado, uma desvantagem dele foram os tempos

computacionais, substancialmente superiores aos da heurística de Wang.

Apesar de a heurística de factibilização utilizada ser bastante simples, o presente método produziu resultados razoavelmente bons, se comparados aos de WANG (1983) e FAYARD *et al.* (1998), e parece ser mais efetivo nos exemplos com peças pequenas (Tabela 3) do que grandes (Tabela 2). Entretanto, o método, na sua versão atual, é superado pela abordagem em grafo-e/ou de MORABITO & ARENALES (1996), capaz de produzir, para todos os exemplos aqui analisados, soluções no mínimo tão boas e com tempos computacionais bem inferiores. Outras heurísticas de factibilização mais elaboradas que a atual estão na nossa agenda de pesquisa para tentar melhorar o desempenho atual do método.

Outra perspectiva interessante para pesquisa futura seria um estudo aprofundado em uma fábrica de vidro ou placas de circuito impresso, aplicando a heurística de aspiração (seção 2) combinada com o método aqui proposto, para avaliar na prática o benefício desta abordagem com relação aos métodos utilizados pela empresa.

Agradecimentos

Os autores agradecem aos revisores anônimos pelos úteis comentários e sugestões. Esta pesquisa contou com apoio da FAPESP (processos 95/9522-0, 97/02508-7, 97/13930-1) e CNPq (processo 522973/95-7).

Referências Bibliográficas

- ARENALES, M.; MORABITO, R.; YANASSE, H. Special issue: Cutting and packing problems. **Pesquisa Operacional**, v.19, n.2, p.107-299, 1999.
- BEASLEY, J.E. Algorithms for unconstrained two-dimensional guillotine cutting. **Journal of the Operational Research Society**, v.36, n.4, p.297-306, 1985.
- BISCHOFF, E.; WAESCHER, G. Special Issue on Cutting and Packing Problems. **European Journal of Operational Research**, v.84, n.3, p.503-712, 1995.
- CHRISTOFIDES, N.; HADJICONSTANTINO, E. An Exact Algorithm for Orthogonal 2-D Cutting Problems Using Guillotine Cuts. **European Journal of Operational Research**, v.83, p.21-38, 1995.

- CHRISTOFIDES, N.; MINGOZZI, A.; TOTH, P. State-space relaxation procedure for the computation of bounds to routing problems. **Networks**, v.11, p.145-164, 1981.
- CHRISTOFIDES, N.; WHITLOCK, C. An algorithm for two-dimensional cutting problems. **Operations Research**, v.25, n.1, p.30-44, 1977.
- DOWSLAND, K.; DOWSLAND, W.B. Packing Problems. **European Journal of Operational Research**, v.56, p.2-14, 1992.
- DYCKHOFF, H. Classification of Real World Trim Loss Problems. In: FANDEL, G. *et al.* **Essays on Production Theory and Planning**. Berlin: Springer-Verlag, 1992.
- DYCKHOFF, H.; FINKE, U. **Cutting and Packing in Production and Distribution: Typology and Bibliography**. Heidelberg: Springer-Verlag, 1992.
- DYCKHOFF, H.; SCHEITHAUER, G.; TERNO, J. Cutting and packing. In: AMICO, M.; MAFFIOLI, F.; MARTELLO, S. (eds.). **Annotated bibliographies in combinatorial optimisation**. New York: John Wiley & Sons, 1997. p.393-414.
- FAYARD, D.; HIFI, M.; ZISSIMOPOULOS, V. An efficient approach for large-scale two-dimensional guillotine cutting stock problems. **Journal of the Operational Research Society**, v.49, p.1270-1277, 1998.
- HERZ, J.C. Recursive computational procedure for two-dimensional cutting. **IBM Journal of Research and Development**, p.462-469, 1972.
- HIFI, M. An improvement of Viswanathan and Bagchi's exact algorithm for constrained two-dimensional cutting stock. **Computers and Operations Research**, v.24, n.8, p.727-736, 1997.
- HINXMAN, A.I. The Trim Loss and Assortment Problems: A Survey. **European Journal of Operational Research**, v.5, p.8-18, 1980.
- MORABITO, R.; ARENALES, M. Um Exame dos Problemas de Corte e Empacotamento. **Pesquisa Operacional**, v.12, n.1, p.1-20, 1992.
- MORABITO, R.; ARENALES, M. Performance of Two Heuristics for Solving Large Scale Two-Dimensional Guillotine Cutting Problems. **INFOR**, v.33, n.2, p.145-155, 1995.
- MORABITO, R.; ARENALES, M. Staged and constrained two-dimensional guillotine cutting problems: An AND/OR-graph approach. **European Journal of Operational Research**, v.94, p.548-560, 1996.
- MORNAR, V.; KHOSHNEVIS, B. A cutting stock procedure for printed circuit board production. **Computers and Industrial Engineering**, v.32, n.1, p.57-66, 1997.
- MUKHACHEVA, E.A. **Decision making under conditions of uncertainty: cutting-packing problems**. The International Scientific Collection, Ufa, Russia, 1997.
- OLIVEIRA, J.F.; FERREIRA, J.S. An improved version of Wang's algorithm for two-dimensional cutting problems. **European Journal of Operational Research**, v.44, p.256-266, 1990.
- SILVEIRA, R.J. **O Problema de Corte Bidimensional Guilhotinado e Restrito: Aplicação na Indústria de Circuitos Impressos**. Dissertação (Mestrado) – DEP, Universidade Federal de São Carlos, São Carlos, 1999.
- SWEENEY, P.; PATERNOSTER, E. Cutting and Packing Problems: A Categorized, Application-oriented Research Bibliography. **Journal of the Operational Research Society**, v.43, p.691-706, 1992.
- VASKO, F.J. A computational improvement to Wang's two-dimensional cutting stock algorithm. **Computers and Industrial Engineering**, v.16, n.1, p.109-115, 1989.
- VISWANATHAN, K.V.; BAGCHI, A. Best-first search methods for constrained two-dimensional cutting stock problems. **Operations Research**, v.41, n.4, p.768-776, 1993.
- WANG, P.Y. Two Algorithms for Constrained Two-Dimensional Cutting Stock Problems. **Operations Research**, v.31, p.573-586, 1983.

A HEURISTIC METHOD BASED ON DYNAMIC PROGRAMMING FOR THE CONSTRAINED TWO-DIMENSIONAL GUILLOTINE CUTTING PROBLEM

Abstract

In this paper we study a particular case of two-dimensional cutting problems named constrained guillotine cutting (CGC). The CGC is an NP-hard problem that appears in different industrial processes of cutting rectangular plates, such as in the glass and circuit board industries. To solve the problem we present a variation of the exact method of CHRISTOFIDES & HADJICONSTANTINO (1995), based on a state space relaxation of a dynamic programming formulation of the CGC, a procedure of subgradient optimization type, and a feasibility heuristic. The result is a method without guarantee of optimality, however, faster and able to solve larger problems than the exact method of Christofides and Hadjiconstantinou. The computational performance of the approach is evaluated solving several examples of the literature as well as randomly generated examples, and comparing the solutions obtained with the ones of Christofides and Hadjiconstantinou's method and the well-known heuristic of WANG (1983).

Key words: cutting problem, constrained two-dimensional cutting patterns, dynamic programming, heuristics.