# Algoritmo *Branch-and-Price* para o Problema de Corte de Estoque Não-Guilhotinado

**Vinícius Loti de Lima e Thiago Alves de Queiroz**
Unidade de Matemática e Tecnologia - UFG/Regional Catalão,
Av. Dr. Lamartine P. Avelar, 1120, Setor Universitário, 75704-020, Catalão-GO.
vini.loti@gmail.com
taq@ufg.br

**RESUMO**

O problema de corte de estoque bidimensional (2PCE) considera o corte de itens retangulares em recipientes também retangulares com o objetivo de minimizar o número de recipientes utilizados. A literatura normalmente aborda o 2PCE com a restrição de cortes guilhotinados. O caso mais geral do 2PCE considera cortes não guilhotinados. A partir de uma análise feita na literatura, observou-se a não existência de métodos exatos para a resolução do 2PCE não guilhotinado. Sendo assim, este trabalho apresenta um algoritmo *branch-and-price* para a resolução do 2PCE não guilhotinado, bem como é o primeiro a apresentar resultados computacionais em instâncias *benchmark* da literatura. Os resultados computacionais são bem promissores, pois mostram que 81% das instâncias foram resolvidas na otimalidade.

**PALAVRAS CHAVE. Problema de Corte de Estoque. Cortes não-guilhotinados.** ***Branch-and-price*.**

**Tópicos: PM - Programação Matemática. OC - Otimização Combinatória.**

**ABSTRACT**

The two-dimensional cutting stock (2CSP) problem considers the cutting of rectangular items from rectangular bins, aiming to minimize the number of bins used. The literature usually considers the 2CSP with guillotine cuts. The most general case of the 2CSP considers non-guillotine cuts. Analyzing the literature, it has been observed that there is no exact methods for solving the non-guillotine 2CSP. In this way, this paper presents a branch-and-price algorithm for solving the non-guillotine 2CSP, as well as it is the first to present computational results on benchmark instances from the literature. The computational results are very promising, since they have shown that 81% of the instances were solved to optimality.

**KEYWORDS. Cutting Stock Problem. Non-guillotine Cuts. Branch-and-price.**

**Paper Topics: PM - Mathematical Programming. OC - Combinatorial Optimization.**

## 1. Introdução

O problema de corte de estoque bidimensional (2PCE) é um problema em que são dadas demandas de itens retangulares, tal que esse itens devem ser cortados de placas, chamadas de recipientes, também retangulares e de tamanho maior do que os itens, em que o objetivo é minimizar o número de recipientes utilizados para a obtenção da demanda dos itens. O 2PCE é equivalente a um problema de empacotamento em recipientes, em que são dadas também demandas de itens retangulares que devem ser empacotados em recipientes inicialmente vazios, com o objetivo de minimizar o número de recipientes utilizados para empacotar toda a demanda de itens.

O 2PCE é um problema que aparece frequentemente na indústria, sendo NP-difícil [Garey e Johnson, 1979], com aplicações no corte de chapas de madeira, de vidros, de chapas metálicas, entre outras [Cintra et al., 2008]. A literatura frequentemente considera o 2PCE com a restrição de cortes guilhotinados, que são cortes efetuados de forma paralela a um dos lados do recipiente indo de uma extremidade a outra. Exemplos práticos de cortes guilhotinados podem ser observados em uma marcenaria, em que os cortes nas placas de madeira são realizados por uma serra circular, ou em uma gráfica, em que os cortes nos papéis são efetuados por uma guilhotina.

Denota-se um problema de corte por *não guilhotinado* se não existe a restrição de corte guilhotinado, ou seja, qualquer configuração de corte é permitida (incluindo cortes guilhotinados). A solução ótima do 2PCE não guilhotinado utiliza sempre um número menor ou igual de recipientes do que a solução ótima da versão com cortes guilhotinados. Assim, ao considerar aplicações práticas, como o corte por laser, em que os itens podem ser cortados em qualquer posição da placa, e problemas de empacotamento, em que os itens não podem ser empilhados, mas podem ser arranjados em qualquer posição da placa, torna-se importante um método de resolução eficiente para o 2PCE não guilhotinado.

Gilmore e Gomory [1961, 1963] propuseram um método de solução para o problema de corte de estoque unidimensional (1PCE), em que foi elaborado um modelo de programação inteira com um número exponencial de variáveis. Como o modelo pode possuir um número muito grande de variáveis, os autores elaboraram o método de geração de colunas para a resolução da relaxação linear do modelo, obtendo assim um limitante inferior para o modelo original. O método de geração de colunas pode ser combinado com uma árvore *branch-and-bound*, resultando em um método *branch-and-price*. Vance [1998] fez uma comparação entre formulações do *branch-and-price* para o problema de corte de estoque unidimensional (1PCE), sendo uma das formulações, a formulação convexa, obtida a partir da decomposição de Dantzig-Wolfe, enquanto a outra formulação é o modelo proposto por Gilmore e Gomory [1961]. Vance [1998] ainda comenta sobre estratégias diferentes de ramificações da árvore, em que o tipo de modelo considerando e, consequentemente, o tipo de ramificação, pode levar a diferentes estratégias para a resolução do subproblema da geração de colunas (problema de *pricing*).

Um algoritmo *branch-and-cut-and-price* consiste de uma árvore *branch-and-bound*, em que, em cada nó, o modelo é resolvido por meio da geração de colunas, obtendo um limitante inferior, assim como a adição de planos de corte no modelo em cada nó para tentar eliminar soluções não interessantes do modelo, podendo levar a uma convergência mais rápida. Belov e Scheithauer [2006] propuseram um algoritmo *branch-and-cut-and-price* que resolve o 1PCE e o 2PCE guilhotinado com 2 estágios de corte. Os autores consideraram o modelo proposto por Gilmore e Gomory [1961] e planos de corte *Chvátal-Gomory*. Outro algoritmo *branch-and-price-and-cut* foi proposto para o 1PCE por Alves e de Carvalho [2008]. Os autores ainda propuseram um método para acelerar a convergência da geração de colunas usando *inequações duais*.

Silva et al. [2010] desenvolveram um modelo de programação inteira para a resolução do 2PCE guilhotinado para um limite de 2 e 3 estágios de corte. A ideia do modelo é considerar cortes ortogonais para a obtenção de novos recipientes/itens. Considerando o 2PCE guilhotinado limitado a 2 estágios de corte, Macedo et al. [2010] propuseram um modelo de fluxo em arcos. Os autores ainda apresentaram métodos para calcular limitantes inferiores, originando uma nova família de planos de corte. Ainda considerando o 2PCE guilhotinado em 2 estágios, Furini et al. [2012] propuseram um método de geração de colunas para a versão do problema que considera recipientes de tamanhos variados. Estes autores propuseram ainda um modelo de programação linear e uma heurística baseada em programação dinâmica. Em outro trabalho, Furini e Malaguti [2013] apresentaram três modelos para a resolução exata do 2PCE guilhotinado com 2 estágios, ainda considerando recipientes de tamanhos variados.

Cintra et al. [2008] apresentaram algoritmos baseados em programação dinâmica para a resolução exata do problema da mochila bidimensional guilhotinado, que considera um limite de $k$ estágios de corte, para qualquer $k$, ou a versão do problema sem limite de estágios de corte. O problema da mochila pode aparecer como subproblema na resolução do problema de corte de estoque. A partir dos algoritmos de programação dinâmica, os autores desenvolveram um algoritmo de geração de colunas para a obtenção de um limitante inferior para o 2PCE guilhotinado, bem como desenvolveram uma heurística para a obtenção de uma solução factível a partir do problema residual resultante da geração de colunas. Recentemente, Furini et al. [2016] apresentaram um método para a resolução do problema da mochila bidimensional guilhotinada sem limite de estágios por meio de um modelo de programação inteira. Os autores ainda estenderam o modelo para a resolução do 2PCE guilhotinado sem limite de estágios, mas não apresentaram resultados computacionais.

A partir dos dados coletados da literatura, percebeu-se a não existência de métodos exatos para a resolução do 2PCE não guilhotinado. Sendo assim, propõe-se neste trabalho uma abordagem exata, em particular, um algoritmo *branch-and-price*, para o 2PCE não guilhotinado com o intuito de preencher essa lacuna na literatura. Como a versão não guilhotinada é a mais geral para o 2PCE, denota-se o problema apenas por 2PCE.

## 2. Método de Resolução para o 2PCE

Para o 2PCE são dados $W$ e $H$, que são a largura e a altura, respectivamente, do recipiente, $m$, que é o número de itens diferentes, e $w$, $h$ e $d$, que são os vetores de largura, altura e demanda, respectivamente, dos $m$ itens. O objetivo do problema é obter $d_i$ cópias do item $i$, para todo $i = 1, \ldots, m$, a partir de cortes nos recipientes de dimensões $(W, H)$, visando minimizar o número de recipientes cortados.

Pode-se utilizar o modelo proposto por Gilmore e Gomory [1961, 1963, 1965] para a resolução do 2PCE. Para tanto, faz-se necessária a definição de padrões de corte. Dada uma instância $I$ do 2PCE com $m$ itens diferentes, um padrão de corte é um vetor $p = (p_1, \ldots, p_m)$, que representa um empacotamento válido em um único recipiente, em que são cortadas $p_i$ cópias de cada item $i$. O modelo (1) pode ser utilizado para a resolução do 2PCE para a instância $I$, em que todos os $n$ padrões de corte são considerados nas colunas da matriz $P$:

$$\begin{aligned} \text{Minimizar} \quad & \sum_{i=1}^{n} x_i \\ \text{Sujeito a:} \quad & Px \geq d \\ & x \in \mathbb{Z}_+^n. \end{aligned} \tag{1}$$

As variáveis $x_i \in Z_+$, para $i = 1, \ldots, n$, do modelo (1) indicam o número de cópias do padrão de corte $i$ que serão usadas na solução visando satisfazer a demanda $d$ dos itens. Como cada padrão de corte representa um recipiente, então a função objetivo do modelo visa minimizar o número de padrões de corte utilizados.

O número de padrões de corte possíveis para uma instância é da ordem exponencial, portanto o modelo (1) possui um número exponencial de variáveis. Descobrir todos os padrões de corte possíveis pode ser uma tarefa de alto custo computacional. Além do mais, a restrição de integralidade pode tornar o modelo mais difícil de ser resolvido. A partir disso, pode-se aplicar um método de resolução da relaxação linear do modelo (1), chamado de *método de geração de colunas*, para obter uma solução geralmente fracionária do problema.

A relaxação linear do modelo (1), chamada de problema mestre, é dada pela mudança do domínio das variáveis de $\mathbb{Z}_+^n$ para $\mathbb{R}_+^n$. A relaxação linear do modelo ainda possui um número muito grande de variáveis para ser resolvida diretamente. Para tanto, o método de geração de colunas considera a resolução de um modelo, chamado de problema mestre restrito, dado pela modificação do problema mestre considerando apenas um subconjunto $P'$ das colunas de $P$.

Como o problema mestre restrito pode considerar apenas um subconjunto pequeno dos padrões de corte possíveis, ele pode ser resolvido diretamente pelo método Simplex em um tempo aceitável. Porém, uma solução ótima do problema mestre restrito, não é necessariamente uma solução ótima do problema. Assim, o método de geração de colunas vai buscar por padrões de corte (colunas), que, adicionadas no subconjunto $P'$ do problema mestre restrito, melhora a solução, aproximando-se da solução do problema mestre [Vance, 1998].

Para a obtenção de padrões de corte que podem melhorar a solução do problema mestre restrito, resolve-se o problema do *pricing*, que é dado pela fórmula (2). A resolução do *pricing* envolve $\pi$, a solução ótima do dual do problema mestre restrito.

$$p(\pi) := \min_i \{1 - \pi p_i \mid \forall p_i \in P\} = \max_i \{\pi p_i \mid \forall p_i \in P\} \tag{2}$$

Após a resolução do *pricing*, obtém-se $i$, que é o índice da coluna candidata a melhorar a solução do problema mestre restrito. Caso $\pi p_i > 1$, então a coluna $p_i$ possui custo reduzido negativo e, portanto, quando adicionada ao problema mestre restrito, vai melhorar a sua solução. Insere-se a coluna $p_i$ em $P'$ e resolve-se o problema mestre restrito novamente, obtendo uma nova solução dual que vai ser utilizada no *pricing* novamente. Por outro lado, se $\pi p_i \leq 1$, então a coluna $p_i$ possui custo reduzido não negativo e, assim, não vai melhorar a solução do problema mestre restrito, concluindo que as colunas de $P'$ são as colunas da solução ótima do problema mestre restrito. Com isso, a geração de colunas é finalizada.

É importante observar que o *pricing* para o 2PCE pode ser resolvido como um problema da mochila bidimensional (2PM). O 2PM considera o empacotamento de itens em um único recipiente, em que cada item possui um valor e o objetivo é encontrar um empacotamento de valor máximo. Para o *pricing*, considera-se os itens da instância do 2PCE sendo resolvida, tal que o valor do item $i$ é dado por $\pi_i$ [Cintra et al., 2008].

Conforme já mencionado, ao final da geração de colunas, tem-se a solução ótima do problema mestre. No entanto, a solução do problema mestre fornece um limitante inferior para o modelo (1), em que as variáveis podem ter valores fracionários, o que não é interessante para o 2PCE. Porém, pode-se combinar o método de geração de colunas com uma árvore *branch-and-bound*, resultando em um algoritmo *branch-and-price* para a resolução do modelo (1) [Vance, 1998].

O *branch-and-price* considera a resolução de subproblemas para buscar uma solução inteira. Considerando-se o modelo dado pelo problema mestre, subproblemas podem ser obtidos a partir da adição de restrições com o intuito de eliminar soluções fracionárias. Portanto, ao solucionar o problema mestre pelo método de geração de colunas, caso a solução seja fracionária, em que a variável $x_i$ apresenta valor fracionário $y_i$, ramifica-se esse problema em dois subproblemas, um com a adição da restrição (3) e outro com a adição da restrição (4).

$$x_i \leq \lfloor y_i \rfloor \tag{3}$$

$$x_i \geq \lceil y_i \rceil \tag{4}$$

Uma solução inteira ótima é satisfeita por uma das duas restrições, (3) ou (4), tal que nenhum dos subproblemas poderá considerar novamente alguma solução em que $x_i = y_i$. Esta ramificação é utilizada para eliminar soluções fracionárias e influenciar na busca por soluções inteiras. Ao resolver um dos subproblemas, caso a sua solução seja fracionária, ramifica-se esse subproblema e o processo anterior é repetido até que não existam mais subproblemas, ou até que a solução ótima (inteira) seja encontrada.

Ao encontrar uma solução inteira de um subproblema, compara-se essa solução com a melhor conhecida visando atualizar a melhor solução inteira conhecida. Caso essa solução seja a ótima para (1), finaliza-se o algoritmo *branch-and-price*. A garantia de otimalidade de uma solução inteira é dada a partir da análise de limitantes. Quando o valor da solução inteira for igual ao melhor limitante inferior conhecido, então essa é a solução ótima. Um limitante inferior pode ser obtido ao arredondar para cima o valor da solução do problema mestre do modelo (1).

Ao resolver a geração de colunas para um subproblema, pode-se obter uma solução inteira heuristicamente para tentar atualizar a melhor solução inteira conhecida. Essa solução heurística é dada pela resolução do modelo com as colunas da solução do problema mestre usando a restrição de integralidade das variáveis.

### 2.1. A resolução do *pricing*

Um aspecto importante do *pricing* é que para melhorar a solução do problema mestre restrito, uma coluna não precisa ter valor máximo, mas sim ter custo reduzido negativo. Portanto não é necessário a resolução do *pricing* na otimalidade, ou seja, pode-se utilizar heurísticas em sua resolução. No entanto, a geração de colunas só é finalizada quando se tem a certeza da não existência de colunas com custo reduzido negativo.

Quando se adiciona restrições do tipo (3) e (4) no modelo, variáveis adicionais são consideradas no dual e devem ser consideradas no *pricing*. Para lidar com as restrições do tipo (4), não se adiciona a restrição diretamente ao modelo, mas sim subtrai-se $x_i * \lceil y_i \rceil$ do valor $d_i$ do lado direito das restrições. Para lidar com as restrições do tipo (3), utiliza-se uma estratégia similar a de Vance [1998]. Para um subproblema, todas as colunas relacionadas a uma restrição do tipo (3) são marcadas como proibidas. Então, na etapa do *pricing*, tenta-se gerar apenas colunas que não são proibidas. Considerando isso, uma coluna proibida pode ser retornada do *pricing* apenas quando não existirem colunas não proibidas com custo reduzido negativo. Essa estratégia é suficiente para garantir a otimalidade da geração de colunas com as restrições adicionais de ramificação.

Não é trivial a elaboração de um algoritmo para o *pricing* que pode evitar gerar colunas não proibidas. Para a resolução do *pricing* para o 2PCE na sua versão não guilhotinada, pode-se utilizar inicialmente o algoritmo *DP* de programação dinâmica proposto por Cintra et al. [2008] para o problema da mochila guilhotinada sem limite de estágios. O algoritmo *DP* pode ser considerado uma heurística para o problema da mochila não guilhotinada. Portanto, sempre que esse algoritmo gerar uma coluna proibida ou com custo reduzido não negativo, utiliza-se um algoritmo enumerativo para buscar colunas não proibidas com custo reduzido negativo.

O algoritmo enumerativo consiste da enumeração de todos os vetores de padrões de corte possíveis. O primeiro desses vetores que fornecer um custo reduzido negativo é checado quanto a viabilidade de ser válido para o 2PCE. O checador de factibilidade resolve o problema de empacotamento ortogonal bidimensional, que checa se um conjunto de itens podem ser empacotado sem sobreposição dentro de um dado recipiente. É importante destacar que o algoritmo enumerativo permite a checagem exata de colunas não proibidas de custo reduzido negativo sem ter que gerar as colunas pela resolução direta do problema da mochila, o que é uma tarefa de alto custo computacional.

## 3. Resultados Computacionais

Os algoritmos para a resolução do 2PCE foram implementados usando a linguagem de programação C++ e os modelos de programação linear foram resolvidos usando o Gurobi Optimizer 6.5 [Gurobi Optimization, 2016]. O computador utilizado para rodar os experimentos possui processador Core i7-4700MQ em 2,4 Ghz, com 12 GB de RAM, sob o sistema operacional Linux

Ubuntu 14.04 (64-bit). Os testes foram feitos sob 42 instâncias, em que as 12 instâncias "gcutd" foram propostas por Cintra et al. [2008] e as demais foram propostas por Hifi e Roucairol [2001]. Um tempo limite de 3600 segundos foi imposto ao *branch-and-price* para resolver cada instância.

Algumas informações dos resultados são apresentados na tabela 1. As colunas "W" e "H" apresentam, respectivamente, a largura e o comprimento do recipiente. A coluna "$w_{min}$" apresenta a razão entre a menor largura entre os itens com a largura do recipiente, enquanto a coluna "$w_{max}$" apresenta a razão entre a maior largura entre os itens com a largura do recipiente. Similarmente, a coluna "$h_{min}$" apresenta a razão entre a menor altura entre os itens com a altura do recipiente, enquanto a coluna "$h_{max}$" apresenta a razão entre a maior altura entre os itens com a altura do recipiente. A coluna "Nós" apresenta o número de nós explorados na árvore do *branch-and-price*. A coluna "GAP" apresenta o GAP da solução encontrada, que é calculado como $GAP = (z_{ub} - z_{lb})/z_{ub}$, em que $z_{ub}$ é o melhor limitante superior encontrado e $z_{lb}$ é o melhor limitante inferior encontrado. A coluna "Tempo" indica o tempo de execução do algoritmo para cada instância, em que os campos preenchidos com "tl" indicam que o tempo de execução chegou a 3600 segundos e, então, o algoritmo retornou a melhor solução encontrada sem garantia de otimalidade. A coluna "Solução" apresenta a solução encontrada para cada instância.

As instâncias que apresentam "0" no campo "Nós" não conseguiram resolver a geração de colunas do primeiro nó, na otimalidade, antes do tempo limite. Assim, para essas instâncias, a solução factível apresentada foi encontrada ao resolver o melhor problema mestre restrito considerando a restrição de integralidade e o limitante inferior utilizado para o cálculo do GAP é o limitante inferior contínuo dado por $\lceil (\sum_i w_i h_i d_i)/(WH) \rceil$. Para as demais instâncias, o limitante inferior é dado $\lceil z \rceil$, em que $z$ é o valor da solução da geração de colunas no primeiro nó. É importante ressaltar que todas as instâncias com $GAP = 0,0$ foram resolvidas na otimalidade. Portanto, tem-se que 34 das 42 instâncias foram resolvidas na otimalidade. Pode-se observar ainda que 27 dessas 34 instâncias tiveram a solução ótima obtida analisando apenas um nó, o que foi devido a heurística usada para a obtenção do limitante superior.

## 4. Considerações Finais

Este trabalho apresentou uma abordagem exata do tipo *branch-and-price* para o problema de corte de estoque bidimensional não guilhotinado. Mesmo sendo um problema com muitas aplicações, ao melhor de nosso conhecimento, não existem trabalhos na literatura que propuseram métodos de resolução exata para este problema.

Foram realizados teste em instâncias *benchmark* da literatura e o algoritmo proposto conseguiu resolver $81,0\%$ das instâncias na otimalidade. Entre as contribuições do trabalho, destaca-se a proposta de resolução do *pricing* a partir de um algoritmo enumerativo em conjunto com o uso do conceito de colunas proibidas. Como não se conhece na literatura resultados em instâncias, tem-se que os resultados computacionais apresentados também são uma grande contribuição, pois fornece um meio de comparação para futuras pesquisas em problemas de corte de estoque.

A partir desta pesquisa foi submetido o artigo intitulado *A branch-and-price algorithm for the two-dimensional guillotine cutting stock problem*, que está em avaliação no periódico *Computers & Operations Research - ISSN 0305-0548* e aborda o 2PCE na versão guilhotinada. Atualmente, tem-se trabalhado em uma extensão do algoritmo para contemplar uma estrutura genérica capaz de resolver várias versões de problemas de empacotamento. Um artigo com esses resultados, intitulado *A general framework for bin packing and cutting stock problems*, está em preparação e deverá ser submetido para a revista *Mathematical Programming - ISSN 0025-5610*.

Por fim, é importante destacar que o aluno Vinícius forneceu a ideia original do algoritmo, implementou todo o algoritmo e fez a parte de experimentação e comparação com a literatura. Ele sempre teve a iniciativa de propor e testar modificações no algoritmo, bem mais do que aquelas discutidas nas reuniões semanais. Foi assim que ele conseguiu chegar em um algoritmo eficiente para resolver a versão geral (não guilhotinada) do problema de corte de estoque, até então sem resultados em métodos, modelos e experimentos por parte da literatura.

Tabela 1: Resultados do *branch-and-price* para o 2PCE na sua versão não guilhotinada.

| Instâncias | Itens | W | H | $w_{min}$ | $w_{max}$ | $h_{min}$ | $h_{max}$ | Nós | GAP | Tempo | Solução |
|---|---|---|---|---|---|---|---|---|---|---|---|
| gcut1d | 10 | 250 | 250 | 0,34 | 0,74 | 0,26 | 0,67 | 1 | 0,0 | 0,1 | 294 |
| gcut2d | 20 | 250 | 250 | 0,27 | 0,74 | 0,07 | 0,67 | 1 | 0,0 | 0,1 | 345 |
| gcut3d | 30 | 250 | 250 | 0,28 | 0,74 | 0,25 | 0,70 | 1 | 0,0 | 0,1 | 332 |
| gcut4d | 50 | 250 | 250 | 0,25 | 0,72 | 0,25 | 0,74 | 1 | 0,0 | 0,2 | 836 |
| gcut5d | 10 | 500 | 500 | 0,29 | 0,71 | 0,26 | 0,73 | 1 | 0,0 | 0,1 | 197 |
| gcut6d | 20 | 500 | 500 | 0,27 | 0,74 | 0,26 | 0,71 | 1 | 0,0 | 0,1 | 338 |
| gcut7d | 30 | 500 | 500 | 0,29 | 0,75 | 0,26 | 0,73 | 35 | 0,0 | 0,9 | 591 |
| gcut8d | 50 | 500 | 500 | 0,25 | 0,75 | 0,26 | 0,72 | 1 | 0,0 | 0,7 | 690 |
| gcut9d | 10 | 1000 | 1000 | 0,34 | 0,69 | 0,29 | 0,67 | 1 | 0,0 | 0,1 | 131 |
| gcut10d | 20 | 1000 | 1000 | 0,26 | 0,74 | 0,27 | 0,73 | 1 | 0,0 | 0,1 | 293 |
| gcut11d | 30 | 1000 | 1000 | 0,27 | 0,75 | 0,27 | 0,67 | 1 | 0,0 | 0,2 | 330 |
| gcut12d | 50 | 1000 | 1000 | 0,27 | 0,72 | 0,25 | 0,75 | 1 | 0,0 | 0,3 | 672 |
| HH | 5 | 98 | 127 | 0,13 | 0,66 | 0,14 | 0,43 | 1 | 0,0 | 0,9 | 2 |
| CW1 | 25 | 105 | 125 | 0,20 | 0,63 | 0,20 | 0,62 | 3 | 0,0 | 415,0 | 9 |
| CW2 | 35 | 165 | 145 | 0,21 | 0,63 | 0,23 | 0,64 | 1 | 0,0 | 608,3 | 12 |
| CW3 | 40 | 207 | 267 | 0,22 | 0,63 | 0,22 | 0,64 | 1 | 0,0 | 3463,4 | 15 |
| Hchl2 | 35 | 130 | 130 | 0,14 | 0,45 | 0,15 | 0,44 | 0 | 0,0 | tl | 6 |
| Hchl9 | 35 | 76 | 65 | 0,13 | 0,57 | 0,15 | 0,48 | 0 | 0,0 | tl | 10 |
| 2s | 10 | 70 | 40 | 0,10 | 0,50 | 0,23 | 0,78 | 4 | 0,0 | 3220,4 | 2 |
| 3s | 20 | 70 | 40 | 0,16 | 0,61 | 0,23 | 0,83 | 1 | 0,0 | 0,1 | 23 |
| A1s | 20 | 60 | 50 | 0,18 | 0,72 | 0,18 | 0,66 | 1 | 0,0 | 0,1 | 23 |
| A2s | 20 | 60 | 60 | 0,23 | 0,70 | 0,20 | 0,55 | 1 | 0,0 | 2,2 | 11 |
| STS2S | 30 | 85 | 55 | 0,12 | 0,51 | 0,18 | 0,56 | 0 | 8,33 | tl | 12 |
| STS4S | 20 | 99 | 99 | 0,16 | 0,49 | 0,14 | 0,44 | 0 | 0,0 | tl | 5 |
| OF1 | 10 | 40 | 70 | 0,10 | 0,98 | 0,13 | 0,79 | 7 | 0,0 | 33,0 | 3 |
| OF2 | 10 | 40 | 70 | 0,10 | 0,68 | 0,19 | 0,67 | 11 | 0,0 | 2,7 | 4 |
| W | 20 | 40 | 70 | 0,23 | 0,83 | 0,16 | 0,61 | 1 | 0,0 | 0,1 | 23 |
| CHL1S | 30 | 100 | 132 | 0,12 | 0,63 | 0,10 | 0,52 | 0 | 16,7 | tl | 6 |
| CHL2S | 10 | 55 | 62 | 0,16 | 0,56 | 0,18 | 0,50 | 1 | 0,0 | 4,7 | 3 |
| A3 | 20 | 80 | 70 | 0,18 | 0,54 | 0,21 | 0,50 | 4 | 0,0 | 236,5 | 7 |
| A4 | 20 | 70 | 90 | 0,16 | 0,61 | 0,10 | 0,37 | 4 | 20,0 | tl | 5 |
| A5 | 20 | 100 | 132 | 0,12 | 0,63 | 0,10 | 0,52 | 0 | 20,0 | tl | 5 |
| CHL5 | 10 | 20 | 20 | 0,10 | 0,70 | 0,05 | 1,00 | 1 | 0,0 | 0,3 | 3 |
| CHL6 | 30 | 130 | 130 | 0,09 | 0,48 | 0,14 | 0,53 | 0 | 16,7 | tl | 6 |
| CHL7 | 35 | 130 | 130 | 0,14 | 0,42 | 0,15 | 0,44 | 0 | 0,0 | tl | 6 |
| CU1 | 35 | 175 | 150 | 0,20 | 0,64 | 0,21 | 0,64 | 0 | 8,3 | tl | 12 |
| CU2 | 25 | 125 | 100 | 0,22 | 0,64 | 0,20 | 0,58 | 1 | 0,0 | 122,3 | 14 |
| Hchl3s | 10 | 98 | 127 | 0,13 | 0,66 | 0,12 | 0,43 | 0 | 0,0 | tl | 3 |
| Hchl4s | 10 | 98 | 127 | 0,13 | 0,66 | 0,12 | 0,43 | 0 | 33,3 | tl | 3 |
| Hchl6s | 22 | 244 | 253 | 0,16 | 0,41 | 0,14 | 0,43 | 0 | 0,0 | tl | 5 |
| Hchl7s | 40 | 241 | 263 | 0,16 | 0,56 | 0,13 | 0,41 | 0 | 12,5 | tl | 8 |
| Hchl8s | 10 | 20 | 49 | 0,10 | 0,70 | 0,02 | 0,41 | 3 | 0,0 | 1,2 | 2 |

## Agradecimentos

Os autores gostariam de agradecer ao Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq - processos 121472/2016-7 e 308312/2016-3) e a Fundação de Amparo à Pesquisa do Estado de Goiás (FAPEG).

## Referências

Alves, C. e de Carvalho, J. V. (2008). A stabilized branch-and-price-and-cut algorithm for the multiple length cutting stock problem. *Computers & Operations Research*, 35(4):1315 – 1328.

Belov, G. e Scheithauer, G. (2006). A branch-and-cut-and-price algorithm for one-dimensional stock cutting and two-dimensional two-stage cutting. *European Journal of Operational Research*, 171(1):85 – 106.

Cintra, G., Miyazawa, F., Wakabayashi, Y., e Xavier, E. (2008). Algorithms for two-dimensional cutting stock and strip packing problems using dynamic programming and column generation. *European Journal of Operational Research*, 191(1):61 – 85.

Furini, F. e Malaguti, E. (2013). Models for the two-dimensional two-stage cutting stock problem with multiple stock size. *Computers & Operations Research*, 40(8):1953 – 1962.

Furini, F., Malaguti, E., Durán, R. M., Persiani, A., e Toth, P. (2012). A column generation heuristic for the two-dimensional two-staged guillotine cutting stock problem with multiple stock size. *European Journal of Operational Research*, 218(1):251 – 260.

Furini, F., Malaguti, E., e Thomopulos, D. (2016). Modeling two-dimensional guillotine cutting problems via integer programming. *INFORMS Journal on Computing*, 28(4):736–751.

Garey, M. R. e Johnson, D. S. (1979). *Computers and intractability: a guide to the theory of NP-completeness*. Freeman, San Francisco.

Gilmore, P. C. e Gomory, R. E. (1961). A linear programming approach to the cutting-stock problem. *Operations Research*, 9(6):849–859.

Gilmore, P. C. e Gomory, R. E. (1963). A linear programming approach to the cutting stock problem—part ii. *Operations Research*, 11(6):863–888.

Gilmore, P. C. e Gomory, R. E. (1965). Multistage cutting stock problems of two and more dimensions. *Operations Research*, 13(1):94–120.

Gurobi Optimization, I. (2016). Gurobi optimizer reference manual. URL `http://www.gurobi.com`.

Hifi, M. e Roucairol, C. (2001). Approximate and exact algorithms for constrained (un) weighted two-dimensional two-staged cutting stock problems. *Journal of Combinatorial Optimization*, 5 (4):465–494.

Macedo, R., Alves, C., e de Carvalho, J. V. (2010). Arc-flow model for the two-dimensional guillotine cutting stock problem. *Computers & Operations Research*, 37(6):991 – 1001.

Silva, E., Alvelos, F., e de Carvalho, J. M. V. (2010). An integer programming model for two- and three-stage two-dimensional cutting stock problems. *European Journal of Operational Research*, 205(3):699 – 708.

Vance, P. H. (1998). Branch-and-price algorithms for the one-dimensional cutting stock problem. *Computational Optimization and Applications*, 9(3):211–228.