UNIVERSIDADE ESTADUAL DE CAMPINAS
INSTITUTO DE MATEMÁTICA, ESTATÍSTICA E COMPUTAÇÃO CIENTÍFICA
DEPARTAMENTO DE MATEMÁTICA APLICADA

FERNANDO RIBEIRO DE SENNA

Problema de corte de estoque unidimensional: método de Gilmore e Gomory e heurísticas

Monografia apresentada ao Instituto de Matemática, Estatística e Computação Científica da Universidade Estadual de Campinas como parte dos requisitos para obtenção de créditos na disciplina Projeto Supervisionado, sob a orientação do(a) Prof. Antônio Carlos Moretti.

Campinas
2019

Sumário

1 Introdução

O presente estudo versa sobre o Problema de Corte de Estoque Unidimensional, que consiste em obter itens menores a partir de um item maior (estoque ou rolo-mestre), aplicando cortes em apenas uma dimensão, os quais podem ser feitos paralela ou perpendicularmente ao comprimento de tais itens. O objetivo maior é encontrar um plano de corte, isto é, um conjunto de formas de cortar o rolo-mestre, denominadas padrões de corte, e o número de vezes que cada um desses padrões é utilizado com a finalidade de satisfazer as demandas dos itens menores.

A abordagem matemática, nesse tipo de problema, é utilizada em diversos estudos existentes para otimizar a produção, visando à minimização de algum custo. Nesse texto, será discutida a minimização do desperdício e do número *setup*, isto é, a quantidade de vezes em que o maquinário é rearranjado para a utilização de um novo padrão de corte, ou ainda, o número de padrões distintos utilizados. A minimização do *setup* é importante devido ao tempo em que a produção fica parada para reorganizar o maquinário.

Outro fator importante a ser minimizado é o excesso de produção, pois este precisa ser estocado, gerando um novo custo. Alguns artigos, como (HAESSLER, 1975), não consideram essa possibilidade, ou seja, apenas aquilo que for demandado poderá ser produzido. Todavia, neste trabalho, haverá permissão de que a produção exceda a demanda e o excedente também será minimizado.

Uma das possíveis avaliações do problema, possuindo aplicações reais, é o corte de fios ou barras, que são cortados perpendicularmente ao comprimento do elemento de estoque. Com isso, não são necessárias adaptações do maquinário ao utilizar padrões distintos, reduzindo a importância da minimização do *setup*.

Outra abordagem para a questão com ampla aplicação prática, tal como discutido em (HAESSLER, 1975), (HAESSLER, 1980) e (MARTIN; MORETTI; GOMES-RUGGIERO; SALLES NETO, 2018), versa sobre o problema de corte de bobinas de papel, que é similar ao corte de bobinas de aço. Nela, os rolos-mestre são cortados paralelamente ao seu comprimento, que é, então, completamente "desenrolado" para ser processado e dividido em itens menores. Dadas tais características, devem ser feitas modificações no maquinário ao variar entre os padrões de corte, tornando relevante que o número de *setup* seja o menor possível. Esse será o problema discutido majoritariamente no presente texto.

Ele consiste em satisfazer um pedido de itens (bobinas) de m larguras diferentes (denotadas l_i), com correspondentes demandas d_i. Para isso, existem em estoque rolos-mestre que podem ter larguras variadas ou única. Será considerado que todos os itens de estoque são idênticos, com largura única L, conforme feito nos artigos citados acima, e que a disponibilidade destes é infinita.

Esse problema é importante devido à sua ampla aplicação na área industrial e à dificuldade computacional de solucioná-lo. Neste trabalho, serão apresentados e discutidos alguns modelos e algoritmos propostos na literatura que atuam nessa otimização, similarmente ao que é feito em (HAESSLER, 1988).

2 Modelo Matemático

Uma forma de se otimizar esse problema é fazer uma abordagem através de programção linear. Um possível modelo, em que a função-objetivo minimiza apenas o desperdício, seria:

$$
\mathbf{P1}\begin{cases}
\textbf{Minimizar} \quad z = & \sum_{j=1}^{n} x_j \\
\textbf{Sujeito a} & \sum_{j=1}^{n} a_{i,j}x_j \geq d_i \ \forall\ i\ \in \{1,2,\ldots,m\} \\
& x_j \geq 0 \text{ inteiro } \forall\ j\ \in\ \{1,2,\ldots,n\}.
\end{cases} \tag{1}
$$

$\mathbf{x_j}$**:** Indica o número de vezes que o padrão j é utilizado.

$\mathbf{a_j}$**:** Vetor com m entradas em que a i-ésima corresponde ao número de itens de largura l_i gerados por cada utilização de um dado padrão de corte j.

n: Número de padrões de corte possíveis, dadas as larguras dos itens demandados e do rolo-mestre.

Esse modelo busca, apenas, a minimização do desperdício de material. É importante observar que $\sum_{j=1}^{n} x_j$ corresponde ao número de rolos-mestre gastos na produção e não ao desperdício total. Contudo, considerando-se que o gasto destes cresce com o aumento do desperdício, a minimização do gasto total de itens do estoque causa a minimização do desperdício. Além disso, essa abordagem evita que padrões de corte que não geram desperdício e que teriam, assim, custo zero, sejam usados demasiadamente, gerando grande excesso de produção. Assim, além de minimizar o desperdício, reduz-se o excedente de produção.

Um outro modelo de programação linear, que visa, também, à minimização do *setup*, seria:

$$
\mathbf{P2}\begin{cases}
\textbf{Minimizar} \quad z = & c_1\sum_{j=1}^{n} x_j + c_2\sum_{j=1}^{n}\delta_j \\
\textbf{Sujeito a} & \sum_{j=1}^{n} a_{i,j}x_j \geq d_i \ \forall\ i\ \in \{1,2,\ldots,m\} \\
& x_j \leq M\delta_j \ \forall\ j\ \in\ \{1,2,\ldots,n\} \\
& x_j \geq 0 \text{ inteiro } \forall\ j\ \in\ \{1,2,\ldots,n\} \\
& \delta_j \text{ binária } \forall\ j\ \in\ \{1,2,\ldots,n\}.
\end{cases} \tag{2}
$$

$\mathbf{c_1}$**:** Custo unitário de um rolo-mestre.

$\mathbf{c_2}$**:** Custo por *setup*.

$\delta_\mathbf{j}$**:** Variável que assume valor 1 se o padrão j for utilizado e 0 caso contrário.

M: Parâmetro de valor muito grande em relação aos demais dados do problema.

Caso fosse desejado que não houvesse excesso de produção, conforme discutido na Seção 1, bastaria alterar o modelo. A substituição dos sinais de desigualdade das restrições relativas à demanda por igualdades atingiria esse objetivo. Para tanto, essas restrições seriam dadas por:

$$\sum_{j=1}^{n} a_{i,j} x_j = d_i \; \forall \; i \in \{1, 2, \ldots, m\}. \tag{3}$$

Todavia, esses problemas são combinatoriais e não podem ser resolvidos convencionalmente pelo Método Simplex, mesmo que implementados em *softwares* próprios para esse algoritmo (denominados *solvers*), devido ao seu grande custo computacional.

Uma das causas de tal custo é a restrição de integralidade das variáveis x_j, além da presença de variáveis δ_j binárias no segundo modelo. Ademais, o desempenho computacional é dificultado pela enorme quantidade de colunas apresentadas pelo problema, isto é, o número (n) de variáveis x_j e correspondentes vetores $\mathbf{a_j}$, os quais representam padrões de corte distintos.

Como solução, diversos autores propuseram formas alternativas de abordar esses aspectos. Nas próximas seções, serão apresentadas algumas dessas técnicas e resultados obtidos em diversos testes para cada uma delas.

3 Método de Gilmore e Gomory

Em (GILMORE; GOMORY, 1961) e (GILMORE; GOMORY, 1963), é apresentado o chamado método de geração de colunas. Nele, o problema de corte de estoque unidimensional é abordado de acordo com o Modelo P1, ou seja, o *setup* é desconsiderado. Também é analisada a perspectiva de vários comprimentos diferentes para os itens de estoque, um ponto que não será abordado nesse estudo.

Ademais, utilizam-se variáveis x_j contínuas. Como as demandas geralmente são grandes, não há grande prejuízo em utilizar heurísticas de arredondamento no resultado final. Logo, esses artigos apresentam uma solução para a dificuldade computacional gerada pelo número excessivo de colunas.

Em vez de gerar todos os padrões de corte possíveis, o problema é analisado através do Método Simplex Revisado, exposto em (BAZARAA; JARVIS; SHERALI, 2010). Uma base inicial factível é proposta e, então, deve-se escolher a melhor variável para entrar nesta base. Diferentemente do método simplex revisado exposto no livro citado, não são conhecidas as demais colunas do problema, necessárias para encontrar os valores dos custos reduzidos e definir a melhor variável para entrar na base.

Para encontrar tais colunas, recorre-se a um subproblema, denominado "Problema da Mochila", mais simples computacionalmente que P1 e que consiste em um problema de programação linear com apenas uma restrição e cuja solução é a próxima coluna a ser utilizada pelo método simplex revisado na solução do problema original. A solução desse subproblema fornece, portanto, a próxima variável x_j a entrar na base e o correspondente vetor $\mathbf{a_j}$, permitindo que se prossiga com a resolução de P1.

Além do método de geração de colunas, são propostas ferramentas para acelerar o processamento do Problema da Mochila, especialmente em (GILMORE; GOMORY, 1963). Algumas formas sugeridas são programação dinâmica e análises das características do problema que podem acelerar a resolução desse subproblema. Todavia, esse aprofundamento não será discutido nesse trabalho.

3.1 Abordagem Matemática

Como o Método Simplex Revisado será utilizado para solucionar o problema de corte de estoque dimensional, é necessário colocar P1 em sua forma padrão. Para isso, acrescenta-se a variável de folga x_{n+i} à i-ésima restrição com sinal negativo, para todo $i \in \{1,2,\dots,m\}$. Além disso, o custo dessas variáveis na função-objetivo é nulo. O problema fica:

$$\textbf{P1 na forma padrão} \begin{cases} \textbf{Minimizar} \quad z = & \displaystyle\sum_{j=1}^{n} x_j + 0\sum_{i=1}^{m} x_{n+i} \\ \quad \textbf{Sujeito a} & \displaystyle\sum_{j=1}^{n} a_{i,j}x_j - x_{n+i} = d_i \ \forall\ i\ \in \{1,2,\dots,m\} \\ & x_j \geq 0 \text{ inteiro } \forall\ j\ \in\ \{1,2,\dots,n\} \\ & x_{n+i} \geq 0 \text{ inteiro } \forall\ i\ \in\ \{1,2,\dots,m\}. \end{cases} \tag{4}$$

Define-se, então, a matriz de produção $\mathbf{A} \in \mathbb{R}^{m\times(m+n)}$ como

$$\mathbf{A} = [a_1, a_2, \dots, a_n, -e_1, -e_2, \dots, -e_m], \tag{5}$$

sendo e_i um vetor canônico, ou seja, um vetor com todas as entradas nulas, exceto a i-ésima, correspondendo à i-ésima variável de folga.

Em uma dada iteração do Método Simplex Revisado, tem-se uma base factível para o problema $\mathbf{B} \in \mathbb{R}^{m\times m}$ com correspondente matriz não-básica $\mathbf{N} \in \mathbb{R}^{m\times(n-m)}$ de forma que $\mathbf{A} = [\mathbf{B}, \mathbf{N}]$. Sejam $I(\mathbf{B})$ e $I(\mathbf{N})$ os conjuntos dos índices das variáveis básicas e não-básicas, respectivamente. Particiona-se, também, o vetor de custo $\mathbf{c} \in \mathbb{R}^{(n+m)\times 1}$ em $\mathbf{c^t} = [\mathbf{c^t_B}, \mathbf{c^t_N}]$, sendo que $\mathbf{c}$ tem as n primeiras entradas iguais a 1, correspondentes às variáveis estruturais x_j, e as últimas m iguais a 0, correspondendo às folgas $x_{n+1}, x_{n+2}, \dots, x_{n+m}$.

Ao avaliar P1, observa-se que é interessante introduzir variáveis de folga na base, pois elas possuem custo zero, enquanto as estruturais apresentam custo 1. Como qualquer base é formada apenas por m variáveis e a função-objetivo tem sentido de minimização, a presença de variáveis de folga na base diminui o número de variáveis estruturais e, por conseguinte, o número de variáveis não nulas com custo positivo, o que melhora o valor da função-objetivo. Todavia, nem sempre a introdução de uma variável básica é interessante devido às restrições do problema. Logo, deve-se verificar se é vantajosa tal mudança de base.

Seja x_{n+i} uma variável de folga. Para que seja interessante introduzi-la na base, deve-se ter:

$$\begin{aligned} z_{n+i} - c_{n+i} &> 0 \qquad (6)\\ z_{n+i} &> 0 \qquad (7)\\ \mathbf{wa_{n+i}} &> 0 \qquad (8)\\ \mathbf{w}(-\mathbf{e_i}) &> 0 \qquad (9)\\ w_i &< 0. \qquad (10) \end{aligned}$$

Assim, caso haja alguma variável dual w_i negativa, a folga correspondente x_{n+i} deve ser escolhida para entrar na base.

Contudo, é possível que não seja vantajoso modificar a base com uma variável de folga. Nesse caso, deve-se escolher uma variável estrutural para entrar na base.

Seja x_j uma variável estrutural não-básica com $j \in I(\mathbf{N})$ e correspondente padrão de corte dado por $\mathbf{a_j}$. A entrada dessa variável na base é vantajosa se seu custo reduzido for positivo, isto é, $z_j - c_j > 0$, sendo $z_j = \mathbf{c_B}^t\mathbf{B}^{-1}\mathbf{a_j} = \mathbf{wa_j}$, com $\mathbf{w} \in \mathbb{R}^{1\times m}$ como vetor das variáveis duais correspondentes à presente base $\mathbf{B}$. Portanto:

$$\begin{aligned} z_j - c_j &> 0 \qquad (11)\\ z_j &> c_j \qquad (12)\\ \mathbf{wa_j} &> 1. \qquad (13) \end{aligned}$$

Como não foram gerados os n padrões de corte, será necessário, na j-ésima iteração do algoritmo do método de geração de colunas, encontrar o vetor $\mathbf{a_j}$ que satisfaça a Desigualdade 13. Além disso, o padrão de corte deve ser factível, isto é, $a_{i,j}$ deve ser inteiro para todo $i \in \{1, 2, \ldots, m\}$ e não pode precisar de um rolo-mestre maior do que o disponível para ser utilizado, ou seja, deve satisfazer:

$$\sum_{i=1}^{m} l_i a_i \leq L. \qquad (14)$$

Uma forma de encontrar padrão de corte que satisfaça as condições dadas pelas Desigualdades 13 e 14 é resolver o seguinte problema de programação linear:

$$\textbf{(Problema da Mochila)} \begin{cases} \textbf{Maximizar} & z_m = & w_1a_{1,j} + w_2a_{2,j} + \ldots + w_ma_{m,j} \\ \textbf{Sujeito a} & & l_1a_{1,j} + l_2a_{2,j} + \ldots + l_ma_{m,j} \leq L \\ & & a_{i,j} \geq 0 \text{ inteiro } \forall\, i \in \{1,2,\ldots,m\}. \end{cases} \tag{15}$$

As variáveis de decisão desse problema são os coeficientes $a_{i,j}$ utilizados na matriz de produção $\mathbf{A}$ de P1 referentes ao j-ésimo padrão de corte. Por esse motivo, utiliza-se o nome "método de geração de colunas", pois elas são geradas ao longo da resolução do problema original.

É importante observar que se a solução do Problema da Mochila for tal que $z_m^* \leq 1$, não haverá nenhum padrão de corte factível que satisfaça $z_j > 1$ e, por conseguinte, a base atual $\mathbf{B}$ é ótima e o problema original está resolvido. Caso $z_m^* > 1$, pode-se utilizar o padrão de corte encontrado como solução do Problema da Mochila para ser uma coluna do problema original e introduzi-lo na base.

Outro ponto importante relativo ao uso do Problema da Mochila é a inserção, dentro da otimização relacionada aos valores das variáveis duais, do máximo possível de itens num mesmo padrão, gerando uma minização do desperdício de cada padrão, que se alia à minimização do gasto total de rolos-mestre pelo plano de corte ótimo.

Se houver variável candidata a ingressar na base, seja ela estrutural ou de folga, realiza-se o pivoteamento do Método Simplex Revisado e procede-se a uma nova busca. Caso contrário, a atual base $\mathbf{B}$ de P1 e sua correspondente solução já podem ser consideradas ótimas.

É relevante o fato de que, apesar de ser permitido o excedente produtivo, ele só ocorre caso haja variáveis de folga positivas, cujos valores correspondem ao excesso dos itens relacionados a elas. Caso nenhuma delas seja básica, ou seja, o número de *setup* seja igual ao de itens distintos no "pedido", não haverá excedente de produção.

3.2 Algoritmo

Passo 1: Determinar base inicial factível.

Serão utilizados os m padrões homogêneos, isto é, padrões de corte cujos itens produzidos sejam todos de mesmo comprimento, que possuem o máximo possível de itens. Assim, definem-se as m primeiras colunas conhecidas da matriz $\mathbf{A}$ $(\mathbf{a_1}, \mathbf{a_2}, \ldots, \mathbf{a_m})$ da seguinte forma:

$$a_{i,j} = \begin{cases} \left\lfloor \frac{L}{l_i} \right\rfloor, & \text{se } i = j \\ 0, & \text{senão} \end{cases} \qquad \forall\, i \in \{1,2,\ldots,m\}, \tag{16}$$

em que $\lfloor x \rfloor$ representa o maior inteiro menor ou igual a x.

Como esses padrões de corte definem uma base, as m primeiras variáveis serão básicas e assumirão valor

$$x_i = \frac{d_i}{\left\lfloor \frac{L}{l_i} \right\rfloor} \quad \forall\, i \in \{1,2,\ldots,m\}. \tag{17}$$

Passo 2: Verificar se é interessante tornar básica alguma variável de folga.

Para tanto, deve-se verificar a existência de variáveis duais negativas. Caso uma delas o seja, a correspondente folga é candidata a entrar na base e passa-se ao Passo 4. Caso contrário, nenhuma variável de folga é candidata e deve-se prosseguir com o Passo 3.

Passo 3: Verificar se é interessante tornar básica variável estrutural.

Devem ser encontrados os valores da variáveis duais correpondentes à atual base. Com isso, resolve-se o Problema da Mochila proposto na Seção 3.1.

Se, na condição de otimalidade dele, obtém-se $z_m^* > 1$, a variável estrutural x_j correspondente ao padrão de corte encontrado como solução desse subproblema é candidata a entrar na base e o vetor $\mathbf{a_j}$ é acoplado a P1, passando a figurar dentre as colunas conhecidas da matriz $\mathbf{A}$. Nesse caso, prossegue-se para o Passo 4. Caso contrário, a base atual $\mathbf{B}$ de P1 já é ótima e o problema está resolvido.

Passo 4: Introduzir variável na base.

Se alguma variável for considerada candidata a entrar na base nos Passos 2 ou 3, efetua-se o teste da razão para definir qual dentre as m variáveis básicas sairá da base. Realiza-se normalmente o pivoteamento característico do Método Simplex Revisado, retornando ao Passo 2.

3.3 Implementação

Para os fins desse estudo, o algoritmo foi implementado em linguagem C utilizando-se o pacote *glpkAPI* como *solver* para resolução dos problemas de programação linear através da IDE *CodeBlocks* em um computador *Dell Inspiron 15 Série 7000* com processador *Intel Core i7* 8^a geração e memória RAM de 16GB.

Devido à possibilidade de se utilizar um *solver* para resolver tais problemas, algumas mudanças foram realizadas na implementação.

Inicialmente, foi implementado um problema de programação linear com m colunas referentes aos padrões de corte homogêneos apresentados pela Equação 16, incluindo também a demanda de cada item. Em seguida, foi encontrada uma solução para esse modelo matemático, para que o próprio *glpkAPI* calculasse o valor das variáveis.

Dadas as características do modelo, as variáveis básicas necessariamente são estruturais e assumem o valor previsto pela Equação 17, pois todos os padrões devem ser utilizados a fim de satisfazer a demanda, uma vez que se o i-ésimo padrão não for utilizado, o i-ésimo item não será produzido. Ademais, se houvesse excesso de algum item (x_j superior ao valor dado pela Equação 17), haveria alguma variável de folga básica, o que não pode ser o caso uma vez que todas as m variáveis básicas são estruturais.

É importante observar que, ao resolver o atual problema (e o das próximas iterações), o *solver* já verifica se é interessante introduzir uma variável de folga na base e fornece a correspondente solução "ótima" com as folgas que a melhoram, se esse for o caso.

Em seguida, obtêm-se as variáveis duais necessárias para implementação do Problema da Mochila, que também é solucionado pelo *glpkAPI* e fornece o melhor padrão de corte para aquela base. Com essa informação, é verificado se esse padrão de corte melhora a atual solução. Se esse for o caso, acopla-se essa coluna ao problema original e ele continua sua busca pela otimalidade do problema original. Caso contrário, o programa é encerrado, fornecendo a melhor solução encontrada.

Assim, as principais modificações são o fato de o pivoteamento não ser feito por um algoritmo próprio, mas pelo *solver* e não ser necessário verificar a utilidade de se acrescentar uma variável de folga à base.

Além dessas alterações, o critério de análise para verificar a utilidade da solução do Problema da Mochila foi alterado para

$$z_m^* > 1 + \varepsilon, \tag{18}$$

sendo ε um número muito pequeno que não modifica a solução ótima, mas é utilizado com a finalidade de evitar erros provenientes de instabilidade numérica.

Por fim, ao resultado foi implementada uma heurística de arredondamento que tomava, no lugar do valor de x_j obtido pela aplicação do método, o menor inteiro maior ou igual a x_j, ou seja, o padrão de corte j deve ser utilizado $\lceil x_j \rceil$ vezes.

Uma consequência desse arredondamento é a possível geração de excessos de produção inexistentes na solução ótima do problema considerando-se as variáveis x_j contínuas. Conforme discutido na Seção 3.1, só há excedente no ótimo caso o número de *setup* seja inferior ao número de itens distintos demandados. Todavia, apesar dessa inferioridade da solução após tal tratamento numérico, ele é essencial para que a solução tenha algum valor prático, dado ser impossível a utilização de padrões de corte um número não inteiro de vezes.

4 Modificação do Método de Gilmore e Gomory

Em (HAESSLER, 1980), é proposta uma modificação do método de geração de colunas proposto em (GILMORE; GOMORY, 1961) e (GILMORE; GOMORY, 1963) e discutido na Seção 3 a fim de buscar uma redução do *setup* e de problemas de arredondamento. Por outro lado, a modificação implica em maior custo computacional.

Os problemas de arredondamento são aqueles discutidos em seções anteriores, causando o excesso de produção de alguns itens que não apresentariam esse excedente se fosse considerado o ótimo contínuo. O gasto de rolos-mestre e o desperdício também crescem com a heurística de arredondamento.

No artigo referido, o valor obtido de x_j na solução ótima é arredondado para o maior inteiro menor ou igual a x_j, ou seja, o padrão de corte j é utilizado $\lfloor x_j \rfloor$ vezes, realizando um truncamento, diferentemente do que foi feito no presente estudo. Esse truncamento pode levar à não satisfação da demanda desejada, conforme acontece em diversos dos exemplos apresentados em (HAESSLER, 1980), pois é preferido não satisfazer a demanda em alguns casos a gerar um excesso de produção.

A modificação do método é feita na geração dos padrões de corte, isto é, no Problema da Mochila. Ele é alterado para buscar padrões em que um mesmo item apareça menos vezes. Essa abordagem se justifica pelo fato de que, se um padrão gera um número menor de itens iguais, ele deverá ser utilizado mais vezes em busca da satisfação da demanda, reduzindo o *setup*.

4.1 Modificações do Algoritmo

Define-se um vetor $\mathbf{UP} \in \mathbb{R}^{m\times 1}$ que limita o número máximo de itens do tipo i que pode ser produzido pelos padrões de corte.

A primeira modificação é na base inicial. Continuam-se tomando padrões de corte homogêneos. Todavia, esses padrões devem obedecer ao limite máximo de itens gerados imposto por $\mathbf{UP}$. Assim, os primeiros m padrões de corte $\mathbf{a_1}, \mathbf{a_2}, \ldots, \mathbf{a_m}$ serão dados por:

$$a_{i,j} = \begin{cases} \min\left\{ \left\lfloor \frac{L}{l_i} \right\rfloor, UP_i \right\}, & \text{se } i = j \\ 0, & \text{senão} \end{cases} \quad \forall\, i \in \{1,2,\ldots,m\}. \tag{19}$$

Consequentemente, as variáveis básicas também terão seus valores alterados. Os valores assumidos pelas primeiras m variáveis estruturais, que serão básicas, serão:

$$x_i = \max\left\{ \frac{d_i}{\left\lfloor \frac{L}{l_i} \right\rfloor}, \frac{d_i}{UP_i} \right\} \quad \forall\, i \in \{1,2,\ldots,m\}. \tag{20}$$

Já com essa primeira modificação é possível notar que os padrões de corte são mais utilizados que anteriormente ou continuam com o mesmo uso, satisfazendo o objetivo desejado.

A segunda modificação é feita no Problema da Mochila, a fim de que seja satisfeito o limite de itens por padrão de corte. Com a modificação, tem-se:

$$\textbf{(Nova Mochila)} \begin{cases} \textbf{Maximizar} & z_m = & w_1 a_{1,j} + w_2 a_{2,j} + \ldots + w_m a_{m,j} \\ \textbf{Sujeito a} & & l_1 a_{1,j} + l_2 a_{2,j} + \ldots + l_m a_{m,j} \leq L \\ & & 0 \leq a_{i,j} \leq UP_i \text{ inteiro } \forall\, i \in \{1,2,\ldots,m\}. \end{cases} \tag{21}$$

Uma vez atingido o ótimo com as novas condições, é possível parar o processamento do problema e adotar a solução encontrada, ou remover os limites impostos e proceder com o método de geração de colunas original. É esperado que, mesmo que se continue com o método sem a restrição do limite de itens por padrão de corte, a solução final apresente melhorias relacionadas ao número de *setup* e ao resultado obtido com o arredondamento.

4.2 Implementação

A implementação desse método foi realizada da mesma forma que a do método de Gilmore e Gomory, conforme explanado na Seção 3.2. As únicas modificações realizadas foram concernentes ao limite de itens por padrão de corte conforme proposto na Seção 4.1.

Como sugerido no artigo, o limite superior das variáveis foi tomado como

$$UP_i = \max\left\{1, \left\lfloor \frac{d_i}{10} \right\rfloor\right\} \quad \forall\, i \in \{1, 2, \ldots, m\}. \tag{22}$$

Ademais, o programa interrompe seu processamento uma vez obtida uma solução ótima com os padrões de corte sendo limitados por UP_i, não continuando com o Método de Geração de Colunas mesmo que isso reduza o uso de rolos-mestre.

Por fim, a heurística de arredondamento utilizada nesse estudo foi igual à feita para o método de Gilmore e Gomory, ou seja, substituir x_j pelo menor inteiro maior ou igual a x_j para facilitar a comparação com as implementações dos outros métodos, em que a demanda é sempre satisfeita. Logo, o padrão de corte j será utilizado $\lceil x_j \rceil$ vezes, e não $\lfloor x_j \rfloor$ como proposto pelo artigo citado.

5 *Sequential Heuristic Procedure* - SHP

Em (HAESSLER, 1975) é proposta uma heurística para solucionar o problema de corte considerando-se a minimização do desperdício do uso de rolos-mestre e do número de *setup*. Não é utilizado nenhum dos dois problemas de programação linear propostos na Seção 2, apenas um desenvolvimento com algoritmos iterativos que buscam obter bons padrões de corte capazes de solucionar o problema.

A heurística busca encontrar padrões de corte com baixo desperdício e que sejam usados várias vezes, diminuindo a necessidade de adaptar o maquinário para o processamento do próximo padrão. Para tanto, deve-se limitar o número de itens iguais gerados pelo mesmo padrão de corte com a finalidade de que o mesmo padrão seja utilizado muitas vezes, como é feito na modificação do método de Gilmore e Gomory apresentado na Seção 4.

Também são apresentadas pelo autor do citado artigo características dos pedidos que tornem útil a heurística. Essas considerações serão discutidas junto da apresentação dos resultados obtidos com as implementações dos algoritmos descritos no presente estudo.

5.1 Parâmetros

Para a implementação da heurística, são utilizados os seguintes parâmetros:

r: Vetor de demandas restantes, ou seja, em dada iteração, r_i representa o número de itens de comprimento l_i que ainda devem ser produzidos, considerando-se quanto já foi produzido pelos padrões de corte já gerados.

$\mathbf{I_1}$: Estimativa do número de rolos-mestre que ainda serão necessários para satisfazer as demandas restantes.

$$I_1 = \frac{\sum_{i=1}^{m} r_i l_i}{L}. \tag{23}$$

$\mathbf{I_2}$: Número médio de rolos a ser produzido no próximo padrão de corte a ser gerado.

$$I_2 = \frac{\sum_{i=1}^{m} r_i}{I_1}. \tag{24}$$

MAXTL: Desperdício máximo permitido por padrão de corte. O artigo sugere que seja utilizado $0,006L \leq MAXTL \leq 0,03L$.

MINR: Número mínimo de rolos permitido num dado padrão de corte. O artigo sugere definir $\lfloor I_2 - 1 \rfloor$.

MAXR: Número máximo de rolos permitido num padrão de corte. Depende do maquinário disponível, ou seja, é igual ao número de facas disponível (se for um problema de corte de papel, como o discutido no artigo).

MINU: Número mínimo de rolos-mestre a serem processados pelo padrão. O artigo (HAESSLER, 1975) propõe iniciar com $0,5I_1 \leq MINU \leq 0,9I_1$.

5.2 Restrições

Assim como nos métodos propostos anteriormente, o j-ésimo padrão de corte é representado pelo vetor $\mathbf{a_j}$, ou seja, suas entradas $a_{i,j}$ representam o número de itens de largura l_i a serem produzidos por ele. Para ser considerado válido (factível) pela heurística, ele deve satisfazer as seguintes condições:

1) O desperdício não deve exceder o máximo permitido.

$$L - \sum_{i=1}^{m} a_{i,j} l_i \leq MAXTL. \tag{25}$$

2) O número total de rolos no padrão de corte deve satisfazer o intervalo permitido.

$$MINR \leq \sum_{i=1}^{m} a_{i,j} \leq MAXR. \tag{26}$$

3) O número de vezes que cada tamanho aparece num padrão de corte deve ser limitado para que o padrão seja usado o número de vezes desejado.

$$\frac{r_i}{a_{i,j} * MINU} \geq 1 \;\; \forall \; a_{i,j} > 0. \tag{27}$$

5.3 Algoritmo

Em uma dada iteração j, busca-se um novo padrão de corte $\mathbf{a_j}$, encontra-se o número x_j de vezes que ele será utilizado e se atualizam as demandas restantes (r_i). Na primeira iteração, considera-se $r_i = d_i \; \forall \; i \in \{1,2,\ldots,m\}$.

Os passos do algoritmo são:

Passo 1: Ordenar os itens.

Ordenar os itens restantes, e os respectivos comprimentos, de acordo com as demandas r_i em ordem decrescente. Denote por k o item de maior demanda restante:

$$k = \underset{i \in \{1,2,\ldots,m\}}{\arg\max} \; \{r_i\}. \tag{28}$$

Passo 2: Gerar padrões.

Gerar padrões em ordem lexicográfica decrescente tais que

$$0 < a_{k,j} \leq \left\lfloor \frac{r_k}{MINU} \right\rfloor. \tag{29}$$

A geração de padrões deve ser feita de forma que o próximo item a ser inserido no padrão de corte seja aquele que tem a maior demanda restante (dentre aqueles que ainda não estão no padrão de corte), considerando sempre

$$0 \leq a_{i,j} \leq \left\lfloor \frac{r_i}{MINU} \right\rfloor \;\; \forall \; i \in \{1,2,\ldots,m\}. \tag{30}$$

Essa condição obriga que a terceira restrição imposta na Seção 5.2 seja satisfeita.

Passo 3: Verificar validade do padrão de corte.

Verificar se o padrão de corte encontrado satisfaz as restrições 1 e 2 da Seção 5.2. Se não satisfizer, reduza MINU em uma unidade e retorne ao Passo 2. Caso contrário, prossiga para o Passo 4.

Passo 4: Utilizar o padrão de corte.

O padrão de corte encontrado deve ser utilizado no valor máximo possível, ou seja, o maior valor de forma que nenhum item exceda a demanda restante (r_i).

$$x_j = \min_{i \in \{1,2,\dots,m\}} \left\{ \frac{r_i}{a_{i,j}} \right\}. \tag{31}$$

Passo 5: Atualizar as demandas.

Atualizar o valor das demandas restantes no vetor **r** subtraindo a produção de cada item ao utilizar o padrão de corte gerado pela iteração atual e prosseguir para a busca do próximo padrão, retornando ao Passo 1.

O algoritmo encerra seu processamento quando todas as demandas forem satisfeitas.

5.4 Implementação

A implementação foi feita em linguagem C, utilizando a IDE *CodeBlocks*, em um computador *Dell Inspiron 15 Série 7000* com processador *Intel Core i7* 8^a geração e 16GB de memória RAM.

Para os valores de parâmetros, foram adotados, em todos os testes: $MAXR = 11$, conforme sugerido no artigo que apresenta a heurística *SHP*, $MAXTL = 0,015L$ e $MINU = \lfloor 0,9I_1 \rfloor$ inicialmente.

O algoritmo de ordenação utilizado foi o *selection sort* e, com a finalidade de evitar instabilidade numérica ao ordenar os vetores de demanda e largura repetidas vezes, criou-se um vetor de índices $ind = \{0, 1, \dots, (m-1)\}$ e os demais vetores são acessados por meio desse vetor, que é reordenado a cada iteração.

Para a busca de padrões em ordem lexicográfica decrescente, foi utilizado o Algoritmo 1, que é utilizado após ter sido feita a ordenação dos itens em ordem decrescente das demandas restantes. Dessa forma, o item que mais precisa ser produzido é o de índice 0 e aquele com menor demanda restante é o de índice $(m-1)$. Na descrição do Algoritmo 1, foi dispensado o uso do vetor de índices, para simplificar a notação, embora na implementação computacional ele tenha sido utilizado, conforme mencionado acima.

Para essa busca, foi necessário realizar uma adaptação da heurística, criando um "mecanismo de segurança" para os casos em que não existe padrão de corte que satisfaça as três restrições dadas na Seção 5.2 mesmo quando $MINU = 1$, devido ao fato de as características dos "pedidos" simulados nesse estudo serem muito variadas e ser possível não encontrar um padrão de corte factível mesmo que $MINU = 1$. Assim, se $MINU$ atinge o valor 0 sem encontrar padrão de corte factível, é gerado um padrão de corte que produza o máximo possível (considerando-se a largura do rolo-mestre, o parâmetro $MAXR$ e a terceira restrição da Seção 5.2) de itens em ordem decrescente de demanda restante. Mesmo que esse padrão não seja considerado factível, ele é utilizado.

Uma vez encontrado o próximo padrão, seguem-se os próximos Passos do algoritmo, utilizando o padrão de corte gerado e atualizando o vetor de demandas restantes. Realizam-se buscas iterativas por padrões de corte até a satisfação total das demandas.

É importante observar que não há excesso de produção devido à forma que o procedimento é desenvolvido. Somente são incluídos nos padrões de corte os itens que ainda não tiveram suas demandas satisfeitas e, uma vez encontrado um padrão de corte, ele é utilizado um número de vezes que não exceda a demanda restante de nenhum item, impedindo, também, o excedente produtivo.

6 Testes

Os testes foram implementados com "pedidos" de diferentes características a fim de comparar o desempenho de cada um dos métodos avaliados. Em (HAESSLER, 1975) é explicado que a heurística *SHP* proposta é útil para problemas com duas características: média de largura dos itens inferior a um quarto do comprimento do rolo-mestre e quantidade variada de itens.

O primeiro teste foi implementado com o que é proposto em (HAESSLER, 1975). Os quatro seguintes foram criados a fim de testar o desempenho de todos os métodos em testes com as propriedades exigidas pela *SHP*. Assim, há um que satisfaz as duas exigências, um que não satisfaz nenhuma delas e os outros dois satisfazem apenas uma das duas.

Essa seção é dividida entre os cinco testes realizados com as implementações feitas para os métodos e apresentam uma descrição das principais características dos pedidos, dentre as quais figuram a relação entre largura e demanda dos itens, largura do rolo-mestre, número total de itens no pedido, médias (aritmética e ponderada pelas demandas) das larguras dos itens e demanda total.

Ademais, são apresentados os planos de corte, cujas descrições são feitas por meio de tabelas, em que os padrões de corte utilizados são numerados e descritos a partir do comprimento dos itens que eles produzem e, entre parênteses, o número de vezes que cada item aparece no padrão. São apresentados, também, a utilização de cada padrão e o desperdício de matéria prima gerado por cada uso dele.

Outrossim, são apresentadas outras informações relativas à solução encontrada: uso total de rolos-mestre, número de *setup*, desperdício total, excesso produtivo (com exceção da *SHP* que não produz excesso) e tempo de execução. Para o método de Gilmore e Gomory e

Algoritmo 1: Busca Lexicográfica Decrescente

Entrada: Iteração j, largura L, vetores **l** e **r**
Saída: Vetor correspondente ao padrão de corte j ($\mathbf{a_j}$)
início
Compute parâmetros $I_1, I_2, MAXR, MINR$ e $MINU$ e defina $rest \leftarrow L$
repita
se $MINU > 0$ **então**
$a_{0,j} \leftarrow \min \left\{ \left\lfloor \frac{L}{l_0} \right\rfloor, \left\lfloor \frac{r_0}{MINU} \right\rfloor \right\}$
$rest \leftarrow rest - a_{0,j} * l_0$
para *todo* $i \in \{1, 2, \ldots, (m-1)\}$ **faça**
se $rest \geq \left\lfloor \frac{r_i}{MINU} \right\rfloor * l_i$ **então**
$a_{i,j} \leftarrow \left\lfloor \frac{r_i}{MINU} \right\rfloor$ e $rest \leftarrow rest - a_{i,j} * l_i$
fim
fim
se *Padrão de corte obedece às restrições* **então**
Pare. Padrão de corte encontrado
senão
repita
Defina o índice k de forma que $a_{k,j} >= 2$ e k é máximo e faça $a_{k,j} \leftarrow a_{k,j} - 1$ e $rest \leftarrow rest + l_k$
para *todo* $i > k$ **faça**
se $l_i < rest$ *e* $a_{i,j} \leq \left\lfloor \frac{r_i}{MINU} - 1 \right\rfloor$ **então**
$a_{i,j} \leftarrow a_{i,j} + 1$ e $rest \leftarrow rest - l_i$
fim
fim
até $\mathbf{a_j}$ *obedecer às restrições ou* $a_{0,j} = 0$
fim
se $\mathbf{a_j}$ *obedece às restrições* **então**
Pare. Padrão de corte encontrado
senão
$MINU \leftarrow MINU - 1$
fim
senão
Defina $N_{itens} \leftarrow 0$ e faça $rest \leftarrow 0$
para *todo* $i \in \{0, 1, \ldots, (m-1\}$ **faça**
se $rest > l_i$ **então**
$a_{i,j} \leftarrow \min \left\{ MAXR - N_{itens}, r_i, \left\lfloor \frac{rest}{l_i} \right\rfloor \right\}$
$N_{itens} \leftarrow N_{itens} + a_{i,j}$
$rest \leftarrow rest - a_{i,j} * l_i$
fim
fim
$MINU \leftarrow -10$
fim
até $MINU \leq -1$ *ou o padrão de corte obedecer às restrições*
fim

a respectiva modificação proposta por Haessler, é apresentado também o número final de padrões de corte gerados, que representa quantas colunas da matriz **A** são conhecidas no fim do processamento do algoritmo, ou seja, a soma do número de vezes que o problema da mochila é acionado com o número de padrões de corte homogêneos originais (m).

Por fim, tais soluções são comentadas e comparadas entre si, discutindo o desempenho de cada uma e em que aspecto um método gera uma solução melhor que o outro para um mesmo teste.

6.1 Teste 1 - Proposta para implementação da *SHP*

Esse teste é proposto em (HAESSLER, 1975). As características do pedido são descritas abaixo e as demandas unitárias são apresentadas na Tabela 1.

- Largura do rolo-mestre: L=141,000.
- Número de itens: 27.
- Média aritmética das larguras dos itens: $M_A \simeq 19,129 \simeq 13,6\%L$.
- Média ponderada das larguras dos itens: $M_P \simeq 16,558 \simeq 11,7\%L$.
- Demanda total: 212 itens.

Tabela 1: Larguras e demandas dos itens - Teste 1

Largura	Demanda	Largura	Demanda	Largura	Demanda
54,000	4	19,500	10	11,625	7
52,500	2	18,250	2	11,250	3
47,500	3	17500	16	10,125	3
26,500	4	15,250	6	10,000	37
25,000	8	13,875	2	9,250	4
24,750	10	13,750	2	9,125	3
23,250	13	12,500	28	8,750	1
22,500	8	12,250	5	8,500	14
20,000	5	12,000	10	7,000	2

Fonte: (HAESSLER, 1975)

6.1.1 Resultado da aplicação do método de Gilmore e Gomory - Teste 1

Abaixo são apresentados os resultados obtidos para o plano de corte ótimo referente a esse "pedido" através do algoritmo do método de Geração de Colunas. Na Tabela 2 são apresentados os padrões de corte realmente utilizados e quanto eles foram utilizados, enquanto na Tabela 3 é apresentado o quanto cada item foi produzido além de sua demanda.

Tabela 2: Solução gerada pelo método de Gilmore e Gomory - Teste 1

	Padrões de corte	**Uso**	**Desperdício**
1	52,500 (2); 24,750 (1); 11,250 (1)	1	0
2	25,000 (3); 18,250 (3); 11,250 (1)	1	0
3	12,000 (5); 10,125 (8)	1	0
4	26,500 (5); 8,500 (1)	1	0
5	12,250 (6); 11,250 (6)	1	0
6	22,500 (3); 12,250 (6)	1	0
7	19,500 (2); 8,500 (12)	1	0
8	19,500 (5); 9,125 (4); 7,000 (1)	1	0
9	54,000 (2); 13,750 (1); 10,125 (1); 9,125 (1)	2	0
10	19,500 (5); 12,500 (2); 9,250 (2)	2	0
11	15,250 (7); 12,500 (2); 9,250 (1)	1	0
12	13,750 (6); 12,500 (4); 8,500 (1)	1	0
13	47,500 (1); 12,500 (6); 9,250 (2)	1	0
14	13,875 (9); 9,125 (1); 7,000 (1)	1	0
15	24,750 (1); 23,250 (5)	3	0
16	20,000 (1); 17,500 (3); 10,000 (6); 8,500 (1)	4	0
17	17,500 (7); 10,000 (1); 8,500 (1)	1	0
18	12,500 (1); 10,000(12); 8,500 (1)	1	0
19	25,000 (4); 12,000 (2); 10,000 (1); 7,000 (1)	2	0
20	19,500 (1); 12,500 (6); 11,625 (4)	2	0
21	24,750 (2); 22,500 (3); 12,000 (2)	3	0
22	12,500 (1); 12,000 (10); 8,500 (1)	1	0
23	20,000 (1); 12,500 (9); 8,500 (1)	2	0
24	47,500 (1); 24,750 (3); 12,250 (1); 7,000 (1)	1	0
25	47,500 (2); 25,000 (1); 12,250 (1); 8,750 (1)	1	0
26	25,000 (1); 24,750 (4); 8,500 (2)	1	0
27	47,500 (2); 19,500 (1); 15,250 (1); 11,250 (1)	1	0
	Total	39	0

- Uso total de rolos-mestre: 39.
- Número de *setup*: 27.
- Número final de padrões de corte gerados: 72.
- Desperdício total: 0.
- Excesso de produção: 133. itens, correspondentes a 62,8% da demanda total.
- Tempo de execução: 4,015 s.

É interessante observar que o desperdício encontrado pela solução é nulo, algo muito interessante se esse é o único objetivo da minimização. Isso ocorre, pois esse método fornece a solução ótima do Problema P1. Contudo, o número de *setup* é muito alto comparativamente

Tabela 3: Excesso de produção do método de Gilmore e Gomory - Teste 1

Largura	Excesso	Largura	Excesso	Largura	Excesso	Largura	Excesso
54,000	0	22,500	4	13,750	6	10,000	2
52,500	0	20,000	1	12,500	20	9,250	3
47,500	3	16,500	10	12,250	9	9,125	4
26,500	1	18,250	1	12,000	15	8,750	0
25,000	5	17,500	3	11,625	1	8,500	11
24,750	7	15,250	2	11,250	6	7,000	3
23,250	2	13,875	7	10,125	7	Total	133

ao número de rolos-mestre gastos, coincidindo com o número de itens distintos demandados. Conclui-se, assim, conforme discutido na Seção 3.1, que a solução ótima para o problema contínuo, isto é, a solução encontrada pelo método de Gilmore e Gomory antes da utilização da heurística de arredondamento, não apresentava excedente. Assim, a totalidade do excedente produtivo é proveniente do arredondamento, e ele é muito alto, podendo gerar grande custo de estoque.

6.1.2 Resultado da aplicação da modificação do método de Gilmore e Gomory - Teste 1

Abaixo, encontra-se o resultado para o mesmo "pedido" obtido com a implementação da variação do método de geração de colunas proposta em (HAESSLER, 1980). A Tabela 4 apresenta o plano de corte encontrado, com os padrões de corte realmente utilizados referenciados através dos itens que produzem, assim como o desperdício gerado por cada um deles e a quantidade de vezes que devem ser aplicados. Ademais, na Tabela 5 se encontram os excessos de produção de cada item.

- Uso total de rolos-mestre: 42.
- Número de *setup*: 27.
- Número final de padrões de corte gerados: 110.
- Desperdício total: 0.
- Excesso de produção: 147. itens, correspondentes a 69,3% da demanda total.
- Tempo de execução: 8,125 s.

A solução encontrada por esse método é pior do que a encontrada pelo método de geração de colunas original, pois o desperdício e o número de *setup* são os mesmos, todavia são gastos três rolos a mais. Outro fator indicativo de uma solução pior é o excesso de produção ter aumentado em quase 7% do número de itens demandados.

Outrossim, o tempo gasto com o processamento é próximo do dobro do que foi gasto

Tabela 4: Solução gerada pela modificação do método de Gilmore e Gomory - Teste 1

	Padrões de corte	**Uso**	**Desperdício**
1	47,5(1);25,0(1);23,25(1);12,5(1);12,25(1);12,0(1);8,5(1)	1	0
2	23,25(1);22,5(1);19,5(1);17,5(1);12,5(2); 12,0(1);11,25(1);10,0(1)	1	0
3	23,25(1);19,5(1);17,5(1);15,25(1);12,5(2);12,0(1);10,0(2);8,5(1)	6	0
4	25,0(1);24,75(1);23,25(1);20,0(1);19,5(1);10,0(2);8,5(1)	3	0
5	52,5(1);26,5(1);15,25(1);12,5(2);11,625(1);10,125(1)	1	0
6	25,0(1);22,5(1);18,25(1);17,5(1);11,625(1);10,0(3);9,125(1);7,0(1)	1	0
7	24,75(1);22,5(1);18,25(1);17,5(1);15,25(1); 13,75(1);11,25(1);9,25(1);8,5(1)	1	0
8	26,5(1);24,75(1);17,5(1);12,0(1); 11,625(1);10,125(1);10,0(3);8,5(1)	2	0
9	22,5(1);13,75(1);12,5(2);12,25(1);11,625(1); 11,25(1);10,0(2);9,125(1);8,5(1);7,0(1)	1	0
10	47,5(1);22,5(1);20,0(1);17,5(1);12,5(2);8,5(1)	3	0
11	24,75(1);23,25(1);22,5(1);17,5(1);13,75(1);10,0(3);9,25(1)	2	0
12	54,0(1);22,5(1);13,875(1);12,25(1);10,0(2);9,25(1);9,125(1)	1	0
13	26,5(1);24,75(1);22,5(1);12,5(2);12,25(1);11,625(1);9,25(1);9,125(1)	1	0
14	25,0(1);24,75(1);23,25(1);17,5(1);11,625(1); 11,25(1);10,0(1);9,125(1);8,5(1)	2	0
15	24,75(1);19,5(1);13,875(1);12,5(2);11,625(1);10,0(3);9,25(1);7,0(1)	1	0
16	25,0(1);22,5(1);19,5(1);17,5(1);13,875(1); 12,5(1);11,625(1);10,0(1);8,5(1)	1	0
17	54,0(1);25,0(1);13,75(1);12,5(2);12,0(1);11,25(1)	1	0
18	26,5(1);24,75(1);18,25(1);17,5(1);12,5(2);11,25(1);9,25(1);8,5(1)	1	0
19	26,5(1);25,0(1);24,75(1);17,5(1);12,5(2);12,25(1);10,0(1)	2	0
20	54,0(1);26,5(1);23,25(1);12,5(1);9,25(1);8,5(1);7,0(1)	1	0
21	54,0(1);23,25(1);17,5(1);12,5(1);12,0(1);11,625(1);10,125(1)	2	0
22	52,5(1);25,0(1);24,75(1);22,5(1);9,25(1);7,0(1)	1	0
23	54,0(1);23,25(1);22,5(1);12,5(1);10,0(2);8,75(1)	1	0
24	52,5(1);19,5(1);18,25(1);12,25(1);10,0(3);8,5(1)	2	0
25	24,75(1);20,0(1);15,25(1);13,875(1);12,25(1); 12,0(1);11,625(1);11,25(1);10,0(2)	1	0
26	25,0(1);19,5(1);18,25(1);15,25(1);12,5(1);11,25(1);10,0(3);9,25(1)	1	0
27	54,0(1);47,5(1);19,5(1);10,0(2)	1	0
	Total	42	0

Tabela 5: Excesso de produção da modificação do método de Gilmore e Gomory - Teste 1

Largura	**Excesso**	**Largura**	**Excesso**	**Largura**	**Excesso**	**Largura**	**Excesso**
54,000	3	22,500	6	13,750	3	10,000	24
52,500	2	20,000	2	12,500	15	9,250	6
47,500	2	16,500	6	12,250	4	9,125	3
26,500	4	18,250	4	12,000	4	8,750	0
25,000	5	17,500	8	11,625	6	8,500	10
24,750	7	15,250	4	11,250	6	7,000	3
23,250	6	13,875	2	10,125	2	Total	147

com o método original e o número de padrões gerados, o qual possui uma correlação com o número de vezes em que se resolve o Problema da Mochila, é mais de 50% maior. Esse resultado é coerente, uma vez que Haessler argumenta que para problemas desse escopo é melhor utilizar a heurística proposta em (HAESSLER, 1975).

Um ponto que merece destaque é o desperdício ter permanecido nulo, apesar de terem sido adicionadas novas restrições ao Problema da Mochila. Observa-se, também, que o número de itens por padrão de corte aumentou, conforme esperado, uma vez que a modificação busca esse resultado, visando a um maior uso de cada padrão. Esse objetivo também é alcançado, pois aumentou-se o número de rolos mestre gastos sem variar a quantidade de *setup* realizados.

É possível concluir que, para problemas similares a esse, se o objetivo for principalmente a minimização do desperdício, é melhor utilizar o algoritmo sem modificações.

6.1.3 Resultado da aplicação da *SHP* - Teste 1

O plano de corte encontrado pela implementação feita nesse estudo da heurística proposta por Haessler é dado abaixo. Não existe excesso de produção para esse algoritmo, já que ele não o admite. Também não faz sentido apresentar o número de padrões gerados, pois esse número é igual ao número de *setup* devido ao fato de todos os padrões gerado serem utilizados. Na Tabela 6 são apresentados os padrões de corte utilizados.

- Uso total de rolos-mestre: 25.
- Número de *setup*: 8.
- Desperdício total: 14,625, equivalente a 4,15% da largura total processada.
- Tempo de execução: 0,984 s.

O resultado encontrado é bastante satisfatório se a minimização do número de *setup* é relevante. Comparativamente com outros dois resultados, esse número é cerca de 30% do valor encontrado nos demais métodos, uma redução expressiva. Além disso, o número de rolos-mestre gastos é muito inferior, pelo fato de não ser necessário aplicar uma heurística de

Tabela 6: Solução gerada pela *SHP* - Teste 1

	Padrões de corte	**Uso**	**Desperdício**
1	24,75(1);23,25(1);17,5(1);12,5(2);12,0(1);10,0(3);8,5(1)	10	0
2	25,0(1);22,5(1);19,5(1);17,5(1);12,5(2);11,625(1);10,0(1);9,25(1)	4	0,625
3	26,5(1);25,0(1);22,5(1);20,0(1);19,5(1);15,25(1);12,25(1)	4	0
4	54,0(2);11,625(1);11,25(1);8,5(1)	2	1,625
5	47,5(2);10,125(3);7,0(2)	1	1,625
6	23,25(1);19,5(1);18,25(1);17,5(1);15,25(1); 13,875(1);13,75(1);10,0(1);9,125(1)	2	0,5
7	52,5(1);47,5(1);12,25(1);11,625(1);8,5(2);	1	0,125
8	52,5(1);23,25(1);20,0(1);11,25(1);10,0(1);9,125(1);8,75(1)	1	6,125
	Total	25	14,625

arredondamento no final, uma vez que o método já considera a integralidade no fornecimento da solução. Outro ponto a ser considerado, porém, é o maior desperdício (os resultados anteriores não apresentam desperdício), mas ainda assim é pequeno, representando apenas 4,15% do total processado.

É importante observar, também, que o resultado difere do apresentado em (HAESSLER, 1975), pois parte dos padrões de corte encontrados é diferente. Tal diferença pode ser explicada pelo algoritmo de ordenação utilizado, o qual não é explicitado no artigo, pois diferentes algoritmos podem deixar itens com mesma demanda restante em diferentes posições na ordenação, que é percorrida pela busca lexicográfica. Outro fator que pode causar essa variação do resultado é a implementação da própria busca lexicográfica decrescente, pois o algoritmo descrito anteriormente pode diferir do utilizado por Haessler. Contudo, o número de *setup*, o gasto de rolos-mestre e o desperdício são iguais, indicando que as soluções são equivalentes para os fins de minimização desejados.

6.2 Teste 2 - Poucos itens com média pequena de larguras

Esse teste satisfaz a condição de que a média das larguras dos itens deve ser inferior a um quarto da largura do rolo-mestre. Porém, diferentemente do que seria ideal para a *SHP*, há poucos itens. A Tabela 7 relaciona as demandas e larguras de cada item.

- Largura do rolo-mestre: L=1000.
- Número de itens: 7.
- Média aritmética das larguras dos itens: $M_A \simeq 98,3 \simeq 9,8\%L$.
- Média ponderada das larguras dos itens: $M_P \simeq 101,8 \simeq 10,2\%L$.
- Demanda total: 178 itens.

Tabela 7: Larguras e demandas dos itens - Teste 2

Largura	163	133	118	97	88	67	22
Demanda	32	15	26	27	38	22	18

6.2.1 Resultado da aplicação do método de Gilmore e Gomory - Teste 2

A Tabela 8 apresenta quais são os padrões de corte utilizados para solução desse problema utilizando-se o método de geração de colunas proposto por Gilmore e Gomory. Já a Tabela 9 explicita os excessos de produção de cada item. Abaixo são apresentadas as demais informações concernentes à solução.

- Uso total de rolos-mestre: 22.
- Número de *setup*: 7.
- Número final de padrões de corte gerados: 19.
- Desperdício total: 0.
- Excesso de produção: 66. itens, correspondentes a 37% da demanda total.
- Tempo de execução: 2,765 s.

Tabela 8: Solução gerada pelo método de Gilmore e Gomory - Teste 2

	Padrões de corte	**Uso**	**Desperdício**
1	118 (3); 88 (1); 67 (8); 22 (1)	2	0
2	163 (6); 22 (1)	6	0
3	97 (1); 67 (1); 22 (38)	1	0
4	133 (3); 97 (3); 88 (2); 67 (2)	5	0
5	118 (4); 88 (6)	4	0
6	118 (5); 97 (4); 22 (1)	2	0
7	97 (6); 88 (4); 22 (3)	2	0
	Total	22	0

Tabela 9: Excesso de produção do método de Gilmore e Gomory - Teste 2

Largura	163	133	118	97	88	67	22
Excesso	4	0	6	9	6	5	36

Novamente, o desperdício resultante é nulo, sendo uma boa solução para esse objetivo de otimização. Todavia, pode ser que, em condições reais, fosse impossível utilizar um padrão de corte que gera 40 itens, pois, no caso da indústria de papel, isso corresponderia à utilização de 39 facas, podendo ser além da capacidade produtiva do maquinário disponível. No caso de corte de fios, por outro lado, isso não representaria problema.

Observa-se, pelos padrões de corte, que esse grande número de itens por padrão se deve ao fato de que o item de largura 22 foi utilizado repetidas vezes nos padrões a fim de que o desperdício fosse nulo. Porém, apesar dessa alta utilização de tal item, não há excedente na solução ótima do problema contínuo, pois o número de *setup* coincide com o número de itens distintos demandados. Por conseguinte, a totalidade dos excedentes provém da heurística de arredondamento.

Comparativamente com o primeiro teste, o desempenho dessa implementação é superior com respeito aos arredondamentos, pois o excesso de itens é apenas 37% do total de itens demandados.

6.2.2 Resultado da aplicação da modificação do método de Gilmore e Gomory - Teste 2

Com a modificação do método proposta por Haessler, obtém-se a nova solução descrita abaixo. A relação de padrões de corte utilizados é apresentada na Tabela 10 e o excedente produtivo na Tabela 11.

- Uso total de rolos-mestre: 22.
- Número de *setup*: 7.
- Número final de padrões de corte gerados: 30.
- Desperdício total: 129, correspondente a 0,6% da largura total processada.
- Excesso de produção: 36 itens, correspondentes a 20% da demanda total.
- Tempo de execução: 4,215 s.

Tabela 10: Solução gerada pela modificação do método de Gilmore e Gomory para o Teste 2

	Padrões de corte	**Uso**	**Desperdício**
1	133 (1); 118 (2); 97 (2); 88 (3); 67 (2); 22 (1)	5	17
2	163 (3); 118 (1); 97 (2); 88 (2); 22 (1)	2	1
3	163 (2); 133 (1); 118 (1); 88 (3); 67 (2); 22 (1)	5	3
4	163 (3); 133 (1); 97 (2); 88 (2)	1	8
5	163 (3); 133 (1); 97 (2); 88 (1); 67 (1); 22 (1)	2	7
6	163 (2); 133 (1); 118 (2); 97 (2); 88 (1); 22 (1)	5	1
7	163 (2); 118 (2); 97 (2); 88 (1); 67 (2); 22 (1)	2	0
	Total	22	129

Os números de *setup* e rolos-mestre gastos são os mesmos que os encontrados para a implementação do método de Gilmore e Gomory. Há, porém, um desperdício pequeno, correspondendo a apenas 0,6% do total processado. Observa-se, também, um menor excedente de

Tabela 11: Excesso de produção obtido com método de Gilmore e Gomory - Teste 2

Largura	163	133	118	97	88	67	22
Excesso	7	3	5	7	7	4	3

produção, correspondendo a apenas 20% da demanda total de itens, condizendo com a proposta de melhoria com problemas de arredondamento, pois novamente o número de *setup* coincide com o número de itens distintos, fazendo com que a solução ótima encontrada para P1 não apresente excessos.

Outrossim, o máximo de itens por padrão é onze, o que é mais realista considerando-se a indústria de papel, pois serão gastas apenas onze facas, comparativamente com as trinta e nove do método proposto por Gilmore e Gomory. Observando a solução, é possível notar que isso é feito com uma maior variedade de itens em cada padrão de corte, resultado garantido pela modificação do Problema da Mochila. Nesse plano de corte, em vez de utilizar o item de largura 22 para evitar desperdício, utilizam-se itens de largura maior em cada um dos padrões de forma a evitar o desperdício e satisfazer a nova restrição do Problema da Mochila.

O desempenho computacional continua pior que o do algoritmo do método de geração de colunas, com o tempo cerca de 50% maior e o número de padrões gerados 58% maior. Algo que também é esperado devido à maior complexidade computacional.

6.2.3 Resultado da aplicação da *SHP* - Teste 2

O plano de corte é apresentado na Tabela 12, enquanto as demais informações concernentes à solução encontrada são apresentadas abaixo.

- Uso total de rolos-mestre: 18.
- Número de *setup*: 6.
- Desperdício total: 888, equivalente a 4,67% da largura total processada.
- Tempo de execução: 0,344 s.

O resultado da heurística é muito inferior aos demais métodos com relação ao desperdício, o que não é justificado pela melhoria com relação ao *setup*, que reduz apenas em uma unidade. Além disso, é evidente que duas vezes a busca lexicográfica não encontrou um padrão de corte factível antes de obter $MINU = 0$ e, assim, foi necessário encontrar padrões de corte utilizando o algoritmo do "mecanismo de segurança" descrito nesse trabalho para encontrar padrões nesses casos, não verificando a factibilidade do padrão e gerando desperdício excessivo.

Conclui-se que, de fato, o desempenho da *SHP* foi ruim, como esperado uma vez que o autor do artigo apresenta as características necessárias ao "pedido" para que ela apresente bons resultados, e uma dessas características não é satisfeita. A heurística da forma que é apresentada

Tabela 12: Solução gerada pela *SHP* - Teste 2

	Padrões de corte	Uso	Desperdício
1	163 (2); 118 (2); 97 (2); 88 (2); 67 (1)	13	1
2	163 (2); 133 (2); 97 (1); 88 (2); 67 (1); 22 (3)	1	2
3	22 (11)	1	758
4	163 (2); 133 (3); 88 (2); 67 (1); 22 (1)	2	10
5	133 (6); 88 (1); 67 (1); 22 (2)	1	3
6	133 (1); 88 (5); 67 (5)	1	92
	Total	19	888

em (HAESSLER, 1975) é incapaz de encontrar plano de corte para esse problema, foi essencial o algoritmo desenvolvido para garantir que seria encontrado um padrão de corte factível ao atingir $MINU = 0$ sem padrão que satisfaça as restrições apresentadas na Seção 5.2.

6.3 Teste 3 - Poucos itens com média grande de larguras

Nesse teste, nenhuma das condições de adequação da *SHP* são satisfeitas. Assim, espera-se um desempenho muito superior dos outros métodos. A Tabela 13 apresenta a relação de itens e suas características.

- Largura do rolo-mestre: L=1000.
- Número de itens: 7.
- Média aritmética das larguras dos itens: $M_A \simeq 403,4 \simeq 40,3\%L$.
- Média ponderada das larguras dos itens: $M_P \simeq 426,2 \simeq 42,6\%L$.
- Demanda total: 208 itens.

Tabela 13: Larguras e demandas dos itens - Teste 3

Largura	698	607	519	462	214	173	151
Demanda	43	26	38	17	29	35	20

6.3.1 Resultado da aplicação do método de Gilmore e Gomory - Teste 3

O resultado deste teste obtido com a implementação do método de geração de colunas é apresentado a seguir, com o plano de corte descrito na Tabela 14 e os excessos de produção na Tabela 15.

- Uso total de rolos-mestre: 108.
- Número de *setup*: 6.

- Número final de padrões de corte gerados: 11.
- Desperdício total: 18441, correspondente a 17% da largura total processada.
- Excesso de produção: 3 itens, correspondentes a 1,4% da demanda total.
- Tempo de execução: 0,312 s.

Tabela 14: Solução gerada pelo método de Gilmore e Gomory - Teste 3

	Padrões de corte	**Uso**	**Desperdício**
1	698 (1)	43	302
2	519 (1)	10	481
3	519 (1); 173 (1); 151 (2)	10	6
4	519 (1); 462 (1)	17	19
5	607 (1); 214 (1); 173 (1)	26	6
6	519 (1); 214 (2)	2	53
	Total	108	18441

Tabela 15: Excesso de produção do método de Gilmore e Gomory - Teste 3

Largura	698	607	519	462	214	173	151
Excesso	0	0	1	0	1	1	0

Observa-se que, diferentemente dos testes anteriores, o número de padrões utilizados (seis) é inferior ao número de itens distintos (sete). Isso indica que, no ótimo contínuo, isto é, desconsiderando-se as restrições de integralidade do uso dos padrões de corte (x_j), há variáveis de folga básicas, representando um excedente produtivo que não é devido, apenas, ao arredondamento de um dos itens. Além disso, os problemas de arredondamento foram mínimos, pois o excedente foi de três itens, sendo um originário da variável de folga básica e apenas dois de arredondamentos.

Outrossim, o desperdício dessa solução é bastante expressivo, devido às grandes larguras dos itens e pequena variedade entre eles, reduzindo a quantidade de combinações possíveis com pequeno desperdício. Essa baixa variedade pode ser vista também no pequeno número final de colunas geradas.

6.3.2 Resultado da aplicação da modificação do método de Gilmore e Gomory - Teste 3

Para a modificação do método de geração de colunas proposto em (HAESSLER, 1980), obteve-se o resultado descrito abaixo, com os padrões de corte apresentados na Tabela 16 e os excessos na Tabela 17.

- Uso total de rolos-mestre: 110.

- Número de *setup*: 7.
- Número final de padrões de corte gerados: 15.
- Desperdício total: 18752, correspondente a 17% da largura total processada.
- Excesso de produção: 8 itens, correspondentes a 3,8% da demanda total.
- Tempo de execução: 1,297 s.

Tabela 16: Solução gerada pela modificação do método de Gilmore e Gomory - Teste 3

	Padrões de corte	**Uso**	**Desperdício**
1	698 (1)	43	302
2	607 (1)	9	393
3	519 (1); 214 (2)	15	53
4	519 (1); 173 (1); 151 (2)	7	6
5	519 (1); 462 (1)	17	19
6	607 (1); 173 (2)	15	47
7	607 (1); 151 (2)	4	91
	Total	110	18752

Tabela 17: Excesso de produção da modificação do método de Gilmore e Gomory - Teste 3

Largura	698	607	519	462	214	173	151
Excesso	0	2	1	0	1	2	2

Comparando com a solução do método original, há apenas três padrões utilizados distintos entre elas e a outra variação corresponde ao número de vezes que cada um deles é utilizado.

Outrossim, a solução dessa modificação é pior em vários sentidos, pois tem maior número de *setup*, rolos-mestre gastos, excedente, padrões de corte gerados e tempo de processamento.

Outro ponto a ser observado é a coincidência do número de itens distintos e *setup*, indicando que no problema contínuo não há variáveis de folga básicas e, por conseguinte, excesso de produção. Todo item produzido além do que foi demandado é proveniente do arredondamento, o qual é, portanto, pior do que o do método de Gilmore e Gomory.

6.3.3 Resultado da aplicação da *SHP* - Teste 3

O último resultado do terceiro teste analisado é obtido através da utilização da heurística proposta por Haessler em (HAESSLER, 1975) e espera-se que não seja encontrado um plano de corte com ela, dependendo do "mecanismo de segurança" criado para obtenção da solução. Essa expectativa se deve ao fato de esse "pedido" não satisfazer nenhum dos requisitos para a boa utilização da *SHP*.

Abaixo estão descritos os resultados e na Tabela 18 é apresentada a forma como o estoque deve ser processado a fim de satisfazer as demandas dadas.

- Uso total de rolos-mestre: 107.
- Número de *setup*: 8.
- Desperdício total: 18347, correspondente a 17% da largura total processada.
- Tempo de execução: 1,828 s.

Tabela 18: Solução gerada pela *SHP* - Teste 3

	Padrões de corte	**Uso**	**Desperdício**
1	698 (1); 151 (2)	10	0
2	519 (1); 173 (2)	17	135
3	698 (1); 214 (1)	29	88
4	607 (1); 173 (1)	1	220
5	607 (1)	25	393
6	519 (1); 462 (1)	17	19
7	698 (1)	4	302
8	519 (1)	4	481
	Total	107	18347

Dados os desperdícios de cada padrão, observa-se que apenas o primeiro deles foi obtido a partir da busca lexicográfica decrescente, pois satisfaz as três restrições apresentadas na Seção 5.2. Os demais foram encontrados utilizando-se o algoritmo para $MINU = 0$.

Para fins de minimização do número de padrões distintos utilizados, esta foi a pior solução encontrada. Por outro lado, devido à natureza da heurística que lida apenas com números inteiros e não apresenta problemas de arredondamento, o desperdício e o gasto de rolos-mestre são os menores encontrados dentre os três métodos utilizados, apesar de a melhoria ser pequena com relação aos demais e não compensar devido ao custo de *setup*.

Um ponto interessante é a busca lexicográfica ter gerado um padrão de corte de desperdício nulo, o qual não aparece nas demais soluções, que minimizam o gasto de estoque. Elas não apresentam tal solução pois a presença de tal padrão no plano de corte prejudica a qualidade dos demais e, portanto, produz uma solução final contínua pior, apesar de a heurística ter apresentado uma melhor (no que diz respeito à minimização de gasto e desperdício de estoque).

6.4 Teste 4 - Muitos itens com média grande de larguras

Esse "pedido" satisfaz a grande variedade de itens necessária ao bom funcionamento da heurística proposta em (HAESSLER, 1975), porém a média de larguras dos itens é muito alta, dificultando as combinações com baixo desperdício. A relação entre larguras e demandas é dada na Tabela 19.

- Largura do rolo-mestre: L=1000.
- Número de itens: 27.
- Média aritmética das larguras dos itens: $M_A \simeq, 450,8 \simeq 45\%L$.
- Média ponderada das larguras dos itens: $M_P \simeq 458,8 \simeq 46\%L$.
- Demanda total: 412 itens.

Tabela 19: Larguras e demandas dos itens - Teste 4

Largura	**Demanda**	**Largura**	**Demanda**	**Largura**	**Demanda**
715	22	545	28	397	13
702	8	541	13	382	7
683	10	531	34	329	10
644	7	489	27	305	10
621	8	482	7	247	15
607	16	464	17	189	5
600	16	454	18	133	14
581	15	421	14	98	10
564	16	402	32	45	20

6.4.1 Resultado da aplicação do método de Gilmore e Gomory - Teste 4

A Tabela 20 apresenta o plano de corte encontrado pelo método de geração de colunas proposto em (GILMORE; GOMORY, 1961) e (GILMORE; GOMORY, 1963), enquanto a Tabela 21 expõe os excessos de cada item. Abaixo são dadas informações complementares.

- Uso total de rolos-mestre: 217.
- Número de *setup*: 25.
- Número final de padrões de corte gerados: 53.
- Desperdício total: 20538, equivalente a 9,45% da largura total processada.
- Excesso de produção: 38 itens, correspondentes a 9,2% da demanda total.
- Tempo de execução: 12,669 s.

Tabela 20: Solução gerada pelo método de Gilmore e Gomory - Teste 4

	Padrões de corte	Uso	Desperdício
1	715 (1)	22	285
2	702 (1)	8	298
3	644 (1)	7	356
4	621 (1)	8	379
5	607 (1)	5	393
6	489 (2)	14	22
7	482 (2)	4	36
8	545 (1); 402 (1)	13	53
9	541 (1); 402 (1); 45 (1)	5	12
10	545 (1); 421 (1)	14	34
11	531 (1); 382 (1)	7	87
12	531 (1); 454 (1)	10	15
13	581 (1); 402 (1)	15	17
14	531 (1); 464 (1)	17	5
15	541 (1); 454 (1)	9	5
16	531 (1); 189 (1); 133 (2)	1	14
17	564 (1); 247 (1); 189 (1)	4	0
18	600 (1); 397 (1)	13	3
19	564 (1); 133 (1); 98 (3)	4	9
20	600 (1) 133 (3)	3	1
21	683 (1); 305 (1)	10	12
22	607 (1); 247 (1); 45 (3)	12	11
23	545 (1); 329 (1)	2	126
24	564 (1); 329 (1)	9	107
25	607 (1); 189 (2)	1	15
	Total	217	20538

Tabela 21: Excesso de produção do método de Gilmore e Gomory - Teste 4

Largura	Excesso	Largura	Excesso	Largura	Excesso	Largura	Excesso
715	0	581	0	464	0	305	0
702	0	564	1	454	1	247	1
683	0	545	1	421	0	189	2
644	0	541	1	402	1	133	1
621	0	531	1	397	0	98	2
607	2	489	1	382	0	45	21
600	0	482	1	329	1	Total	38

Novamente, há menos padrões de corte utilizados do que itens demandados, indicando a presença de variáveis de folga básicas na solução ótima contínua. Ademais, o item de largura 45 foi amplamente utilizado para redução de desperdício de vários padrões. Nota-se porém, que alguns deles poderiam ter incluído itens dessa largura por apresentarem desperdício superior a 45, mas isso não ocorre por prejudicar a minimização final de consumo de estoque, para o problema sem restrições de integralidade.

Comparativamente com o teste da Seção 6.2, o qual também apresenta média de largura dos itens elevada, o desempenho com relação aos excessos e desperdícios foi superior, devido à maior quantidade de itens, que permite maior variabilidade de combinações.

O custo computacional desse teste também foi expressivamente maior, como pode ser observado pelo tempo gasto muito superior aos demais testes.

6.4.2 Resultado da aplicação da modificação do método de Gilmore e Gomory - Teste 4

A Tabela 22 apresenta o plano de corte encontrado com a alteração feita por Haessler para o método de geração de colunas e a Tabela 23 os consequentes excessos. Mais informações sobre a solução também são apresentadas.

- Uso total de rolos-mestre: 213.
- Número de *setup*: 25.
- Número final de padrões de corte gerados: 68.
- Desperdício total: 20918, equivalente a 9,82% da largura total processada.
- Excesso de produção: 11 itens, correspondentes a 2,7% da demanda total.
- Tempo de execução: 14,418 s.

Comparativamente ao resultado da aplicação do método de geração de colunas original, há melhoria considerável no excesso produtivo, indicando melhor desempenho com o arredondamento, conforme é proposto em (HAESSLER, 1975). Há também uma redução no consumo de rolos-mestre que corroboram para esse resultado.

Não há, contudo, variação do *setup*. Tal resultado indica que, novamente, há duas variáveis de folga básicas para a solução final contínua.

Outro ponto comparativo que merece destaque é o pequeno aumento do tempo de execução, o qual apresentou aumento relativo muito maior para os demais testes.

6.4.3 Resultado da aplicação da *SHP* - Teste 4

Os resultados encontrados através da heurística proposta por Haessler são apresentados abaixo, com o plano de corte dado pela Tabela 24.

Tabela 22: Solução gerada pela modificação do método de Gilmore e Gomory - Teste 4

	Padrões de corte	Uso	Desperdício
1	715 (1)	22	285
2	702 (1)	8	298
3	683 (1)	10	317
4	644 (1)	7	356
5	621 (1)	1	379
6	607 (1)	4	393
7	489 (2)	10	22
8	541 (1); 189 (1); 133 (1); 98 (1)	5	39
9	531 (1); 421 (1); 45 (1)	14	3
10	531 (1); 382 (1)	2	87
11	545 (1); 329 (1)	9	126
12	545 (1); 454 (1)	18	1
13	531 (1); 464 (1)	17	5
14	541 (1); 247 (1); 133 (1)	8	79
15	600 (1); 397 (1)	13	3
16	489 (1); 482 (1)	7	29
17	541 (1); 329 (1)	1	130
18	581 (1); 402 (1)	15	17
19	621 (1); 247 (1); 98 (1)	8	34
20	564 (1); 402 (1)	16	34
21	600 (1); 305 (1)	3	95
22	607 (1); 305 (1); 45 (1)	7	43
23	607 (1); 382 (1)	5	11
24	531 (1); 329 (1); 133 (1)	2	7
25	545 (1); 402 (1)	1	53
	Total	213	20918

Tabela 23: Excesso de produção da modificação do método de Gilmore e Gomory - Teste 4

Largura	Excesso	Largura	Excesso	Largura	Excesso	Largura	Excesso
715	0	581	0	464	0	305	0
702	0	564	0	454	0	247	1
683	0	545	0	421	0	189	0
644	0	541	1	402	0	133	1
621	1	531	1	397	0	98	3
607	0	489	0	382	0	45	1
600	0	482	0	329	2	Total	11

Tabela 24: Solução gerada pela *SHP* - Teste 4

	Padrões de corte	**Uso**	**Desperdício**
1	531 (1); 133 (2); 98 (1); 45 (2)	7	15
2	545 (1); 402 (1); 45 (1)	6	8
3	531 (1); 454 (1)	18	15
4	489 (1); 402 (1); 98 (1)	3	11
5	489 (2)	12	22
6	402 (2); 189 (1)	5	7
7	715 (1); 247 (1)	15	38
8	545 (1); 421 (1)	14	34
9	464 (2)	8	72
10	607 (1); 382 (1)	7	11
11	600 (1); 397 (1)	13	3
12	564 (1); 402 (1)	13	34
13	581 (1); 329 (1)	10	90
14	541 (1); 305 (1)	10	154
15	683 (1)	10	317
16	607 (1)	9	393
17	531 (1); 464 (1)	1	5
18	702 (1)	8	298
19	621 (1)	8	379
20	545 (1)	8	455
21	531 (1)	8	469
22	715 (1)	7	285
23	644 (1)	7	356
24	482 (2)	3	36
25	581 (1)	5	419
26	600 (1)	3	400
27	564 (1)	3	436
28	541 (1)	3	459
29	482 (1)	1	518
	Total	225	35988

- Uso total de rolos-mestre: 225.
- Número de *setup*: 29.
- Desperdício total: 35988, correspondente a 16% da largura total processada.
- Tempo de execução: 1,781 s.

Diversos padrões de corte da solução dada não obedecem às restrições apresentadas na Seção 5.2, o que indica que a busca lexicográfica foi incapaz de encontrar padrões de corte factíveis que satisfizessem as demandas dos itens apresentados, sendo necessário recorrer ao algoritmo para encontrar padrões de corte quando $MINU = 0$. Assim, a heurística original seria incapaz de apresentar resultado para esse problema com a busca lexicográfica da forma como

Tabela 25: Larguras e demandas dos itens - Teste 5

Largura	**Demanda**	**Largura**	**Demanda**	**Largura**	**Demanda**
358	18	213	5	98	10
341	13	202	20	77	15
324	10	200	15	66	17
304	27	193	10	56	28
282	6	176	4	45	14
255	7	148	5	38	4
232	10	145	44	28	14
224	7	125	9	11	21
217	6	109	21	6	35

foi implementada, seguindo o Algoritmo 1.

Ademais, o resultado encontrado foi o pior dentre os três métodos utilizados, o que se justifica pela não satisfação da premissa de a média de largura dos itens ser inferior a um quarto da largura dos rolos-mestre.

Vale observar que os primeiros padrões encontrados são muito bons, pois apresentam uma quantidade maior de itens (podendo reduzir o *setup*) e pequeno desperdício. Todavia, comparativamente com as soluções encontradas pelo método de Gilmore e Gomory e sua modificação proposta por Haessler para esse teste, a presença desses padrões pioraria a solução geral final de minimização de gasto de estoque quando desconsideram-se as restrições de integralidade das variáveis x_j, o que explica a não utilização desses padrões nas soluções encontradas pelos outros métodos para o mesmo "pedido".

6.5 Teste 5 - Muitos itens com média pequena de larguras pequena

Esse teste satisfaz as condições de utilidade da heurística apresentadas em (HAESSLER, 1975), assim como o primeiro. O objetivo desse novo teste é comparar os resultados e ver se a heurística apresenta bom desempenho em outros "pedidos" com características similares.

A Tabela 25 relaciona as demandas e larguras dos itens.

- Largura do rolo-mestre: L=1000.
- Número de itens: 27.
- Média aritmética das larguras dos itens: $M_A \simeq, 165,7 \simeq 17\%L$.
- Média ponderada das larguras dos itens: $M_P \simeq 148,6 \simeq 15\%L$.
- Demanda total: 395 itens.

6.5.1 Resultado da aplicação do método de Gilmore e Gomory - Teste 5

Os resultados desse teste utilizando-se o método de geração de colunas são descritos abaixo. A Tabela 26 apresenta o plano de corte encontrado e a Tabela 27, o excedente de produção.

- Uso total de rolos-mestre: 72.
- Número de *setup*: 27.
- Número final de padrões de corte gerados: 71.
- Desperdício total: 0.
- Excesso de produção: 158 itens, correspondentes a 40% da demanda total.
- Tempo de execução: 5,515 s.

Esse resultado confirma a maior facilidade de encontrar baixos deperdícios para problemas em que a média de larguras é pequena. Confirma-se, também a inexistência de qualquer resultado que otimize o *setup*. Todavia, há padrões gerando quantidades muito grandes de itens (até 28), o que pode tornar impossível de se atingir, na prática, em uma indústria de papel, por exemplo, devido às limitações de maquinário.

Ademais, o número de padrões de corte coincide com o número de itens distintos, o que indica a inexistência de variáveis de folga na solução ótima contínua. Assim, todo o excesso é proveniente de problemas de arredondamento, que ainda assim é mais eficiente do que o apresentado para o primeiro teste.

6.5.2 Resultado da aplicação da modificação do método de Gilmore e Gomory - Teste 5

Abaixo são apresentadas as informações relativas à solução encontrada pelo algoritmo apresentado que altera o método de geração de colunas. Na Tabela 28 são apresentados os padrões de corte e, na Tabela 29, o excesso de produção.

- Uso total de rolos-mestre: 72.
- Número de *setup*: 27.
- Número final de padrões de corte gerados: 96.
- Desperdício total: 0.
- Excesso de produção: 101. itens, correspondentes a 26% da demanda total.
- Tempo de execução: 5,814 s.

Tabela 26: Solução gerada pelo método de Gilmore e Gomory - Teste 5

	Padrões de corte	Uso	Desperdício
1	200 (5)	3	0
2	125 (8)	2	0
3	358 (2); 255 (1); 11 (1); 6 (3)	5	0
4	358 (1); 341 (1); 202 (1); 77 (1); 11 (2)	6	0
5	217 (3); 176 (1); 145 (1); 28 (1)	2	0
6	176 (4); 145 (2); 6 (1)	1	0
7	148 (6); 56 (2)	1	0
8	77 (12); 38 (2)	1	0
9	358 (1); 193 (3); 45 (1); 6 (3)	4	0
10	324 (3); 28 (1)	4	0
11	304 (1); 202 (3); 45 (2)	5	0
12	255 (3); 28 (8); 11 (1)	1	0
13	145 (2); 109 (6); 56 (1)	4	0
14	66 (13); 56 (2); 6 (5)	1	0
15	255 (3); 224 (1); 11 (1)	1	0
16	145 (1); 45 (19)	1	0
17	213 (3); 145 (1); 66 (2); 28 (3)	1	0
18	304 (1); 232 (3)	4	0
19	145 (2); 77 (2); 66 (8); 28 (1)	2	0
20	304 (2); 98 (4)	2	0
21	304 (3); 77 (1); 11 (1)	5	0
22	38 (26); 6 (2)	1	0
23	282 (3); 98 (1); 28 (2)	2	0
24	213 (1); 145 (5); 38 (1); 6 (4)	3	0
25	341 (1); 224 (1); 145 (3)	7	0
26	66 (1); 56 (16); 38 (1)	2	0
27	341 (2); 255 (1); 45 (1); 6 (3)	1	0
	Total	72	0

Tabela 27: Excesso de produção do método de Gilmore e Gomory - Teste 5

Largura	Excesso	Largura	Excesso	Largura	Excesso	Largura	Excesso
358	2	224	1	148	1	56	12
341	2	217	0	145	10	45	20
324	2	213	1	125	7	38	29
304	1	202	1	109	3	28	9
282	0	200	0	98	0	11	3
255	5	193	2	77	12	6	15
232	2	176	2	66	16	Total	158

Tabela 28: Solução gerada pela modificação do método de Gilmore e Gomory - Teste 5

	Padrões de corte	**Uso**	**Desperdício**
1	200 (1); 145 (4); 125 (1); 77 (1); 6 (3)	3	0
2	304 (2); 176 (1); 148 (1); 56 (1); 6 (2)	3	0
3	304 (1); 282 (1); 232 (1); 98 (1); 66 (1); 6 (3)	3	0
4	358 (1); 341 (1); 145 (2); 11 (1)	11	0
5	358 (1); 324 (1); 217 (1); 56 (1); 45 (1)	5	0
6	202 (1); 193 (1); 176 (1); 145 (1); 109 (2); 66 (1)	2	0
7	324 (1); 255 (1); 200 (1); 109 (1); 66 (1); 28 (1); 6 (3)	1	0
8	200 (1); 145 (4); 98 (1); 56 (1); 38 (1); 28 (1)	2	0
9	232 (1); 224 (1); 213 (1); 200 (1); 109 (1); 11 (2)	1	0
10	282 (1); 200 (1); 193 (1); 109 (1); 77 (1); 45 (1); 38 (1); 28 (1); 11 (2); 6 (1)	2	0
11	304 (1); 224 (1); 193 (1); 148 (1); 125 (1); 6 (1)	1	0
12	232 (1); 202 (1); 200 (1); 77 (1); 66 (1); 56 (2); 45 (1); 38 (1); 28 (1)	1	0
13	341 (1); 324 (1); 145 (2); 45 (1)	3	0
14	324 (1); 213 (1); 193 (1); 109 (1); 77 (1); 56 (1); 28 (1)	3	0
15	304 (1); 224 (1); 217 (1); 77 (1); 66 (1); 56 (1)	2	0
16	282 (1); 213 (1); 202 (2); 56 (1); 45 (1)	2	0
17	358 (1); 255 (1); 193 (1); 77 (1); 66 (1); 45 (1); 6 (1)	4	0
18	202 (2); 200 (1); 193 (1); 98 (1); 77 (1); 11 (2); 6 (1)	1	0
19	232 (1); 224 (1); 202 (2); 56 (2); 28 (1)	5	0
20	304 (1); 232 (1); 148 (1); 145 (1); 77 (1); 56 (1); 38 (1)	1	0
21	304 (1); 202 (2); 125 (1); 77 (1); 56 (1); 28 (1); 6 (1)	3	0
22	200 (1); 148 (1); 145 (1); 125 (1); 109 (2); 98 (1); 66 (1)	1	0
23	304 (1); 255 (1); 125 (1); 109 (2); 98 (1)	3	0
24	282 (1); 200 (1); 193 (1); 125 (1); 66 (1); 56 (2); 11 (2)	1	0
25	304 (2); 200 (1); 109 (1); 66 (1); 11 (1); 6 (1)	6	0
26	232 (1); 217 (1); 200 (1); 148 (1); 98 (1); 77 (1); 28 (1)	1	0
27	324 (1); 200 (1); 176 (1); 125 (1); 98 (1); 77 (1)	1	0
	Total	72	0

Tabela 29: Excesso de produção da modificação do método de Gilmore e Gomory - Teste 5

Largura	**Excesso**	**Largura**	**Excesso**	**Largura**	**Excesso**	**Largura**	**Excesso**
358	2	224	2	148	2	56	9
341	1	217	2	145	8	45	3
324	3	213	1	125	4	38	2
304	4	202	5	109	4	28	4
282	2	200	6	98	2	11	6
255	1	193	4	77	7	6	9
232	2	176	2	66	4	Total	101

Comparativamente com o método original, o desperdício, o estoque gasto e o número de *setup* permanecem os mesmos. A principal variação é o excesso produtivo, que é 36% menor, cumprindo o principal papel da modificação. Para atingir tal objetivo, o custo computacional é expressivamente maior, como pode ser visto pelo maior tempo de execução e de vezes em que se resolve o Problema da Mochila Modificado.

Outra vantagem desse resultado é que o maior número de itens num mesmo padrão é dez, tornando muito maior a viabilidade desse resultado, na prática, do que o fornecido pelo método de Gilmore e Gomory, cuja solução apresentou padrões capazes de gerar 28 itens.

É importante observar que, novamente, não há variáveis de folga básicas, o que implica que todo o excedente é proveniente, apenas, do processo de arredondamento para ambos os métodos.

6.5.3 Resultado da aplicação da *SHP* - Teste 5

Por fim, é apresentado o resultado encontrado pela heurística para esse problema. Uma vez que ele satisfaz as condições para o bom funcionamento do algoritmo, espera-se um bom desempenho. O plano de corte se encontra na Tabela 30.

- Uso total de rolos-mestre: 60.
- Número de *setup*: 15.
- Desperdício total: 1295, correspondente a 2,2% da largura total processada.
- Tempo de execução: 2,343 s.

Caso a minimização do *setup* seja relevante, a solução encontrada pela heurística é muito superior às demais, pois reduz em mais de 45% esse número, além de manter o desperdício baixo: apenas 2,2%. O gasto de estoque também é consideravelmente inferior.

Dados os quinze padrões de corte da solução, apenas três não satisfazem as restrições apresentadas na Seção 5.2, tendo sido, assim, gerados pelo algoritmo utilizado ao obter $MINU = 0$. Contudo, ainda assim a solução apresentada tem bons indicadores e talvez seja possível fazer adaptações a fim de gerar uma solução um pouco melhor com esses dados. É possível que uma busca lexicográfica decrescente implementada de forma diferente apresente um resultado mais satisfatório.

Assim, apesar de não ser tão superior aos demais resultados quanto no primeiro teste, esse é o melhor método para solucionar esse problema e, de fato, a heurística proposta (HAESSLER, 1975) é a melhor solução para problemas de grande variedade de itens com média de larguras pequenas, se o *setup* for um fator importante a ser minimizado.

Tabela 30: Solução gerada pela *SHP* - Teste 5

	Padrões de corte	**Uso**	**Desperdício**
1	304 (1); 145 (3); 109 (1); 66 (1); 56 (1); 11 (1); 6 (2)	14	7
2	358 (1); 202 (2); 200 (1); 28 (1)	10	10
3	304 (1); 232 (1); 98 (1); 77 (2); 56 (2); 45 (2); 6(1)	7	4
4	341 (1); 324 (1); 193 (1); 125 (1); 11 (1)	7	6
5	358 (2); 255 (1); 28 (1)	4	1
6	224 (3); 109 (3)	2	1
7	341 (2); 213 (1); 66 (1); 38 (1)	3	1
8	304 (2); 282 (1); 98 (1)	3	12
9	255 (1); 217 (1); 200 (1); 176 (1); 148 (1)	3	4
10	324 (1); 282 (1); 232 (1); 148 (1)	2	14
11	217 (1); 193 (2); 145 (1); 125 (1); 77 (1); 38 (1)	1	12
12	217 (2); 213 (2); 125 (1)	1	15
13	324 (1); 232 (1); 200 (2)	1	44
14	282 (1); 224 (1); 193 (1); 176 (1); 109 (1)	1	16
15	145 (1)	1	855
	Total	60	1295

7 Conclusão

A conclusão obtida com esse estudo é que existem diversos métodos capazes de resolver o problema de corte de estoque unidimensional minimizando diferentes custos. Conclui-se, também, que as características das larguras e demandas dos itens que devem ser satisfeitos interfere no desempenho de cada método, assim como a prioridade de minimização.

Se o objetivo de minimização for apenas o desperdício, a melhor forma de obter uma solução satisfatória é através do método de geração de colunas apresentado em (GILMORE; GOMORY, 1961) e (GILMORE; GOMORY, 1963) e amplamente debatido ao longo do presente trabalho. Ele apresenta a melhor solução por obter uma solução ótima para o Problema P1.

Todavia, algumas das soluções encontradas por esse método podem ser difíceis de serem implementadas na prática. Um problema encontrado foi a geração de padrões de corte que produzem um número muito grande de itens. Isso pode impedir a utilização da solução em algumas indústrias, como a de papel, conforme discutido anteriormente, por limitações do maquinário. Por outro lado, uma indústria que realiza cortes perpendiculares ao comprimento dos itens de estoque, o que pode ser feito com uma única faca, não apresentaria esse problema e poderia implementar uma solução que tenha padrões de corte com essa característica.

Outro problema do método de Gilmore e Gomory é a necessidade de uma heurística de arredondamento. A utilizada nesse trabalho foi simples e seria possível utilizar uma mais poderosa. Porém, em alguns casos, o excedente gerado pelo arredondamento pode ser muito grande, como ocorre no primeiro teste da Seção 6, gerando custos de estocagem que precisam ser considerados.

Outra conclusão que se obtém ao analisar os testes realizados é que, por vezes, padrões de corte com mais desperdício geram planos de corte com maior gasto de estoque e, consequentemente, mais desperdício total, como acontece no terceiro teste da Seção 6, em que a *SHP* encontra um padrão de corte de desperdício nulo e apresenta uma solução pior do que a do método de Gilmore e Gomory, que não utiliza esse padrão. Assim, não basta encontrar os melhores padrões de corte, mas o melhor conjunto deles que satisfaça as demandas dos itens.

Para melhorar o desempenho do processo de arredondamento e reduzir o número de itens produzidos pelo mesmo padrão de corte, Haessler apresentou a sua adaptação do método de geração de colunas, descrita em (HAESSLER, 1980).

De fato, comparando os resultados dos testes, observa-se que, de modo geral, ocorre um melhor desempenho da sua modificação no que tange a esses quesitos. Por outro lado, essa adaptação do problema da mochila pode incorrer em maior desperdício e certamente causa maior custo computacional. Em problemas muito complexos, isso pode ser um problema.

Há, também, o *setup*, que é praticamente desconsiderado nos demais métodos e que apresenta, em alguns casos, custo significativo, sendo importante considerá-lo no processo de otimização da produção.

Para solucionar esse problema, é proposta a *sequential heuristic procedure* em (HAESSLER, 1975), que não utiliza programação linear e trabalha com algoritmos iterativos baseados em utilizar os mesmos padrões repetidas vezes e, com isso, reduzir o número de vezes em que se ajusta o maquinário.

Conclui-se que, ao menos com a busca lexicográfica decrescente implementada da forma que foi feita nesse estudo, a heurística é muito sensível às características dos itens demandados, sendo grande o número de problemas em que ela não obtém uma solução factível.

Contudo, com a implementação do "mecanismo de segurança" descrito na Seção 5.4, garante-se que a heurística encontre um plano de corte factível. Se forem satisfeitas as condições impostas em (HAESSLER, 1975) para o bom funcionamento da heurística, isto é, grande variedade de itens demandados e pequenas larguras comparativamente ao rolo-mestre, a solução encontrada, mesmo que recorra ao "mecanismo de segurança" para encontrar alguns padrões é superior aos outros dois métodos apresentados pela vantagem com relação ao número de *setup*.

Além disso, uma implementação da busca lexicográfica decrescente diferente da realizada nesse estudo e um algoritmo a que se recorrer caso não seja obtido padrão de corte considerado factível com $MINU = 0$ mais eficientes podem, também, melhorar o desempenho da heurística.

Portanto, dentre os diversos métodos analisados nesse estudo e as diferentes aplicações, o melhor é avaliar as características dos itens e suas demandas e quais os custos mais importantes de serem minimizados a fim de fazer a melhor escolha do método com que será buscada a solução.

Referências

BAZARAA, M. S.; JARVIS, J. J.; SHERALI, H. D. **Linear Programming and Network Flows**. 4. ed. [S.l.]: Wiley, 2010.

GILMORE, P. C.; GOMORY, R. E. A Linear Programming Approach to the Cuttting-Stock Problem. **Operations Research**, Vol. 9, 1961.

________. A Linear Programming Approach to the Cuttting-Stock Problem-Part II. **Operations Research**, Vol. 11, 1963.

HAESSLER, R. W. Controlling Cutting Pattern Changes in One-Dimensional Trim Problems. **Operations Research**, Vol. 23, 1975.

________. Selection and Design of Heuristic Procedures for Solving Roll Trim Problems. **Management Science**, Vol. 34, 1988.

________. Technical Note - A Note on Computational Modifications to the Gilmore-Gomory Cutting Stock Algorithm. **Operations Research**, Vol. 28, 1980.

MARTIN, M.; MORETTI, A.; GOMES-RUGGIERO, M.; SALLES NETO, L. Modification of Haessler's sequential heuristic procedure for the one-dimensional cutting stock problem with setup cost. **Production**, Vol. 28, 2018.