LI Simpósio Brasileiro de Pesquisa Operacional
Limeira–SP, 02 a 06 de setembro de 2019

HEURÍSTICAS CONSTRUTIVAS PARA O PROBLEMA DE CORTE GUILHOTINADO BIDIMENSIONAL EM 3 ESTÁGIOS COM RESTRIÇÕES DE PRECEDÊNCIA

Marcos Vinícius Almeida Guimarães[1]
Eduardo Theodoro Bogue[1]
Armando Honorio Pereira[1]
Iago Augusto Carvalho[1]
Thiago Ferreira de Noronha[1]
Sebastián Alberto Urrutia[1]
[1]*Departamento de Ciência da Computação / Instituto de Ciências Exatas / Universidade Federal de Minas Gerais*

RESUMO

Nesse trabalho abordamos heurísticas construtivas para o Problema de Corte Bidimensional em Três Estágios com Restrições de Precedência. No problema, os pedidos de corte são feitos através de pilhas que definem a precedência de corte dos itens, dessa forma, um item no topo da pilha deve ser cortado antes de todos os outros. Para cortar os itens são utilizados cortes guilhotinados em três estágios. O objetivo do problema é minimizar o desperdício de vidro. No melhor do nosso conhecimento, não existem trabalhos na literatura que abordam o aspecto de precedência em problemas bidimensionais de cortes guilhotinados. Adaptamos três heurísticas presentes na literatura para resolver o problema proposto. Executamos experimentos computacionais com o objetivo de compará-las. Os resultados obtidos mostram que nossa abordagem é capaz de obter, em média, padrões de cortes com menos de 25% de desperdício.

PALAVRAS CHAVE. Problema do empacotamento. Corte bidimensional. Heurísticas construtivas.

Área principal: Heurísticas.

ABSTRACT

In this work we deal with constructive heuristics for the Two-dimensional Three-staged Cutting Stock Problem with Precedence Constraints. In the problem, cutting requests are made through stacks that define the precedence in which the items must be cut, therefore, the item at the top of the stack must be cut before all the others. To cut the items guillotine cuts in three stages are used. The aim of the problem is to minimize the glass waste. To the best of our knowledge, there are no works in the literature that address the aspect of precedence in two-dimensional problems of guillotined cuts. We adapted three heuristics in the literature to solve the proposed problem. We perform computational experiments in order to compare them. The results show that our approach is able to obtain, on average, cutting patterns with less than 25% of waste.

KEYWORDS. Bin packing. Bidimensional cut. Constructive heuristics.

Main area: Heuristics.

1. Introdução

O vidro está presente em todos lugares que olhamos, seja em janelas, fachadas de residências, prédios, hotéis, vitrines, hospitais ou universidades. Seu alcance e aplicabilidade se tornou um aspecto primordial em nossas vidas. Atualmente existe uma variedade de tipos de vidro com diferentes funções: transparência, isolamento térmico e acústico, segurança, controle solar, decoração, entre outras, de modo que cada tipo de vidro possui uma gama de aplicações tanto domésticas quanto comerciais. Consequentemente, com o avanço de sua utilização, torna-se necessário métodos eficazes para lidar e acompanhar a ampliação de sua aplicação.

As placas de vidro são produzidas principalmente através de um processo chamado *flutuação* (do inglês, *float process*, [Takahashi et al., 2013]). Nesse processo, vários materiais, como por exemplo, areia e soda, são derretidos juntos dentro de um forno grande para criar uma fita de vidro líquido, que é espalhada sobre uma bacia de estanho e depois resfriada para solidificar. A fita obtida é então cortada em grandes placas de vidro com tamanho fixo. Posteriormente, essas placas são empilhados em paletes para serem enviadas para os chamados *transformadores*, que por sua vez são responsáveis por cortá-las em peças retangulares menores, aqui chamadas de itens, adaptados às necessidades dos clientes. Os pedidos dos clientes são organizados em lotes e a ordem dos itens dentro de cada lote deve ser mantida no processo de corte das placas. Em outras palavras, se a posição de um item i é menor que a posição de um item j dentro de um lote L, i deve ser cortado antes de j. Para que essa ordem seja mantida, usa-se a notação de pilha para a representação dos lotes, de forma que, quanto menor o índice do item na pilha, mais próximo do topo da mesma ele se encontra. É importante destacar que não existe restrição de precedência entre os lotes, apenas a ordem dos itens dentro de cada lote deve ser considerada no processo de corte.

Devido à fragilidade do vidro, apenas cortes guilhotinados são permitidos no processo de corte das placas, ou seja, um pedaço de vidro só pode ser cortado de uma extremidade a outra através da propagação de uma abertura ao longo de uma linha reta. Logo, um corte é guilhotinado se, quando aplicado em uma placa retangular, ele produz duas novas placas retangulares. Esse tipo de corte é obrigatório para se cortar vidro, caso contrário, rachaduras podem acontecer. A Figura 1 mostra o que seriam cortes guilhotinados e não guilhotinados.

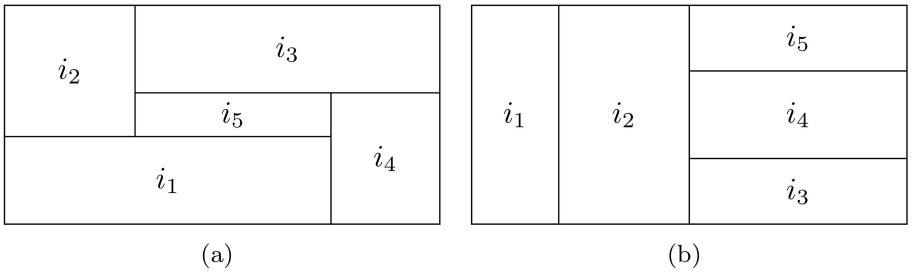

Figura 1: Não guilhotinado (a) e guilhotinado (b). A ordem dos itens não está sendo considerada.

Devido à limitações técnicas, o número de cortes guilhotinados que podem ser realizados para cortar um item é geralmente limitado por um dado número. Por exemplo, um padrão de corte é dito ser de 2 estágios quando um item deve ser obtido a partir de 2 cortes guilhotinados. A notação $\alpha - cut$ é usada para indicar o número de cortes realizados até o momento. O padrão de corte na Figura 1(b) possui 2 estágios, pois é necessário um corte vertical para extrair os itens i_1 e i_2, seguido de dois cortes horizontais para extrair os itens i_3, i_4 e i_5.

Os itens são obtidos de acordo com um padrão de corte que pode ser visto como uma

divisão das placas de vidro em pedaços retangulares de vários tamanhos posicionados de tal forma que as perdas geométricas de vidro (superfícies restantes do vidro pequenas demais para cortar um novo item) são tão baixas quanto possível. De outro modo, um padrão de corte para uma determinada placa pode ser visto como um plano bidimensional com indicações de como cortar as placas para se obter os itens. No problema aqui discutido, um padrão de corte é sempre composto de cortes guilhotinados. A Figura 2 mostra padrões de corte guilhotinado para uma placa.

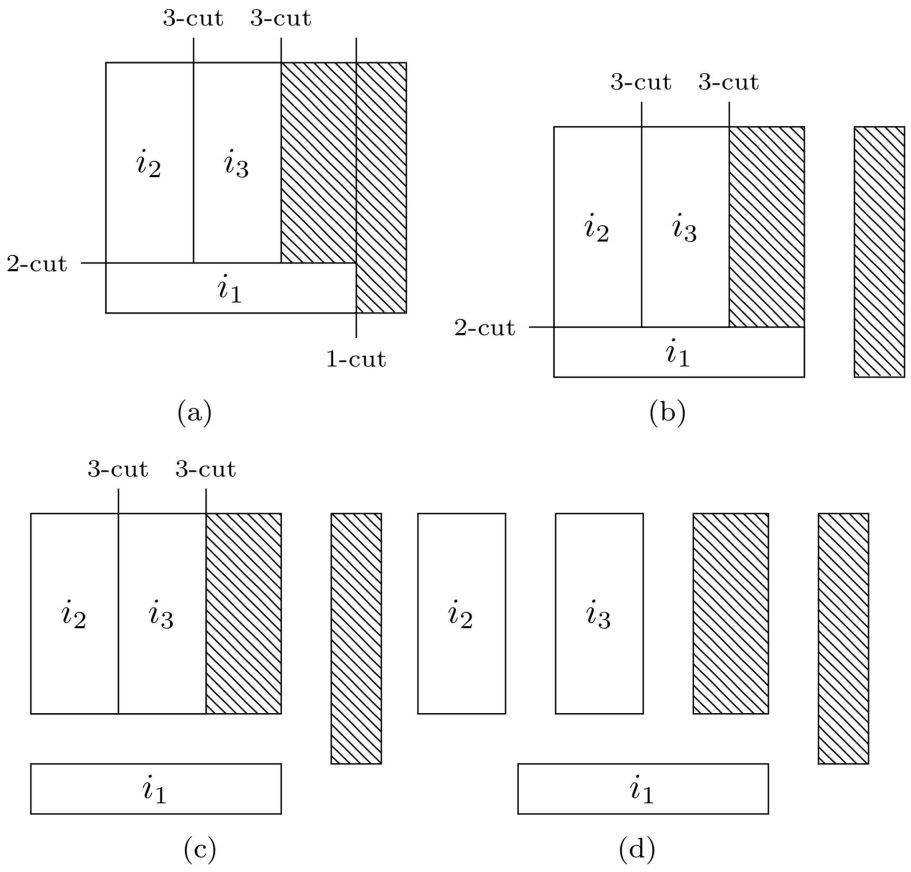

Figura 2: Um padrão inicial (a) e seus variantes a partir dos cortes (b-c-d). A ordem dos itens não está sendo considerada.

Ao final do processo de corte, as partes retangulares que não correspondem a nenhum dos itens são chamadas de desperdício. Desperdícios equivalem à perda de material e devem ser evitadas ao máximo. A parte retangular à direita do último corte do tipo *1-cut* realizado na última placa e que não corresponde a nenhum dos itens é denominada resíduo. Áreas residuais não são consideradas desperdícios pois podem ser usadas posteriormente para a obtenção de novos itens. Na Figura 2, as áreas retangulares com linhas diagonais, à esquerda e à direita do último corte *1-cut*, são consideradas desperdício e resíduo, respectivamente.

Um padrão de corte pode ser representado por uma árvore. Isso é possível porque um corte guilhotinado sempre divide uma placa retangular em duas ou mais placas retangulares menores. Sua raiz corresponde a placa inicial e suas folhas correspondem aos itens, desperdícios ou resíduos. Os filhos de um nó dado (com exceção das folhas) são os componentes obtidos depois de realizar um corte de profundidade α do nó raiz. A representação em árvore do padrão de corte da Figura 2 é

dada na Figura 3.

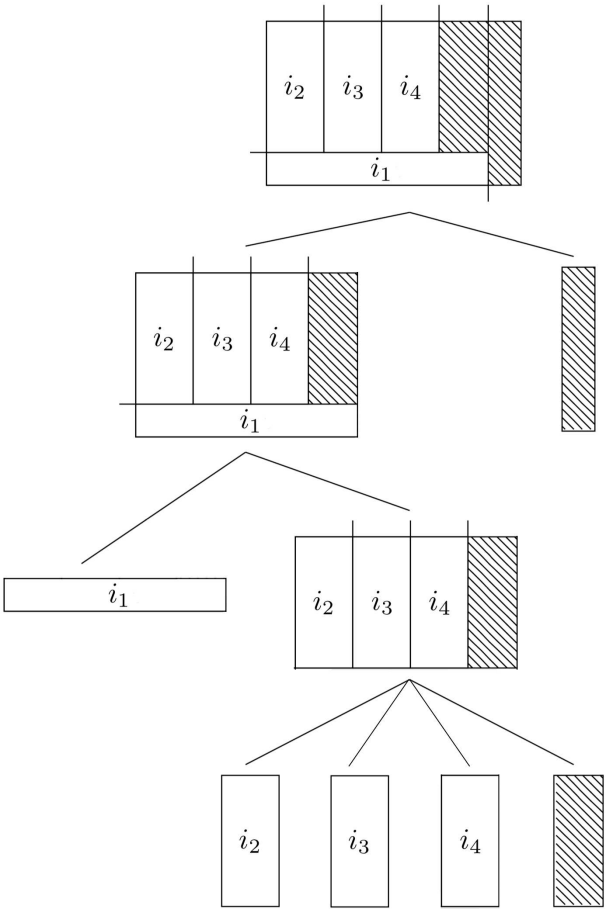

Figura 3: Representação em árvore do padrão de corte da Figura 2.

1.1. Caracterização do problema

Dado um conjunto de placas B de largura $W = 6m$ e altura $H = 3m$, seja S o conjunto de lotes que representa os pedidos dos clientes. Um conjunto de lotes I onde cada lote é representado por uma pilha $s = (i_1, i_2, ..., i_n)$, sendo uma sequência ordenada de itens de forma que $i_1 <_{cut} i_2 <_{cut} ... <_{cut} i_n$, onde $<_{cut}$ representa um operador de ordem parcial. Logo, para dois itens i_x e i_y, tal que $1 \leq x,y \leq n$, se $i_x <_{cut} i_y$, temos que o item i_x deve ser cortado antes do item i_y. Essa ordem vem de restrições de agendamento relacionadas à entregas e processamento dos itens. Cada item i é caracterizado como um par (w_i, h_i), representando, respectivamente, seu comprimento e altura.

Como as placas são empilhadas na fábrica, a ordem do conjunto de placas B deve ser mantida. Seja $\{0, \ldots, |B|\}$ os índices das placas, temos então que para duas placas b_i e b_j, $0 \leq i,j \leq |B|$, se $b_i <_{cut} b_j$ então b_i deve ser usada para o corte de itens antes de b_j. Assume-se que a quantidade de placas é grande o suficiente para se cortar todos os itens, a rotação das mesmas não é permitida.

O corte de vidro é sujeito a algumas restrições que estão relacionadas aos itens, as placas, o processo de corte guilhotinado, entre outros fatores. Uma restrição pode ser física, relacionada

com o corte do vidro, ou organizacional para satisfazer as precedências. As soluções geradas nesse trabalho devem satisfazer todas as seguintes restrições:

- Apenas rotações de 90° nos itens são permitidas.
- Todos os itens do conjunto de lotes *I* devem ser cortados.
- A superprodução de itens não é permitida. Ou seja, apenas os itens de S devem ser cortados.
- A ordem de corte dos itens deve ser respeitada.
- As placas devem ser utilizadas na ordem dada.
- A sobreposição de itens não é permitida.
- Padrões de cortes são bidimensionais e devem ser obtidos através de cortes guilhotinados. O número de cortes que podem ser feitos para obter um item deve ser no máximo 3 (*1, 2, 3-cuts* somente).
- É permitido obter um item com menos de 3 cortes.
- Cortes do tipo *1-cut* e *3-cut* são sempre verticais e os cortes do tipo *2-cut* são sempre horizontais.

O objetivo do problema consiste em fornecer padrões de corte bidimensionais que permitam cortar todos os itens em *S* utilizando as placas disponíveis em *B*, minimizando o desperdício de vidro. Como mencionando anteriormente, a área do resíduo pode ser reutilizada após o corte, logo, é necessário desconsiderá-la no cálculo do desperdício. Para simplificar o gerenciamento residual, apenas o resíduo no último padrão de corte é considerado. Seja $P = \{p_1, p_2, \ldots, p_m\}$ uma solução viável, isto é, um conjunto de padrões de corte satisfazendo todas as restrições do problema. A perda total é medida somando a área de desperdício nos padrões de corte $\{p_1, p_2, \ldots, p_m\}$. Seja r_m a área residual no último padrão de corte p_m, temos então a seguinte função objetivo do problema:

$$min \quad HW|P| - Hr_m - \sum_{i \in I} w_i h_i \tag{1}$$

De modo que $HW|P|$ representa a área total de todas as placas utilizadas na solução gerada, Hr_m representa a área total da parte residual do último padrão de corte e $\sum_{i \in I} w_i h_i$ representa a área total de todos os itens a serem cortados das placas.

Esse problema está relacionado com o problema da mochila bidimensional, entretanto, possui restrições extras devido a requisitos industriais, tais como respeitar a ordem parcial em pilhas de itens e satisfazer restrições físicas como os cortes guilhotinados.

A motivação principal desse trabalho dá-se pelo fato do problema de corte bidimensional guilhotinado em três estágios se tratar de uma necessidade industrial real e que a sua versão, que leva em consideração todas as restrições acima descritas, não foi previamente abordada de forma conjunta em nenhum outro trabalho presente na literatura. O objetivo desse trabalho consiste em preencher essa lacuna propondo algoritmos heurísticos que permitam minimizar os desperdícios de vidro no processo de corte.

O restante do trabalho está organizado da seguinte maneira: na Seção 2 são exibidos os trabalhos relacionados. A Seção 3 apresenta trés heurísticas construtivas para o problema, e na Seção 4 são realizados os experimentos computacionais. Por fim, a Seção 5 apresenta as conclusões a respeito dos resultados obtidos no trabalho.

2. Trabalhos relacionados

Como mencionado, no melhor do nosso conhecimento, até o presente momento nenhum trabalho na literatura leva em consideração as restrições descritas na Seção 1.1 sobre a precedência entre itens. Dessa forma, a revisão bibliográfica realizada considera apenas problemas de corte bidimensional guilhotinado tradicionais. Em Puchinger et al. [2004] foram apresentadas heurísticas para resolver o problema do corte bidimensional guilhotinado em três estágios. Duas estratégias baseadas em *Branch & Bound* foram propostas para resolver o problema. Ambas as estratégias utilizam métodos heurísticos e portanto não garantem soluções ótimas. Também foram propostos algoritmos evolucionários que utilizam técnicas de recombinação e mutação para gerar soluções viáveis para o problema. Nos experimentos realizados foram utilizadas 31 instâncias reais, de modo que os algoritmos evolucionários apresentaram resultados superiores.

Suliman [2006] propõem uma heurística sequencial de três estágios para o problema. No primeiro estágio, é determinado um padrão de corte horizontal que produz perda de largura mínima. No segundo estágio, é determinado o *layout* associado das peças para produzir um padrão de corte. No estágio final, o número de vezes que o padrão de corte gerado será usado é determinado.

Aryanezhad et al. [2012] atribuíram pesos baseados na largura para os itens a serem cortados, de forma que os itens com maior largura tiveram maior prioridade. Essa suposição foi feita com base no princípio de que a maior parte do desperdício nos cortes guilhotinados é criada durante o processo de aparação (para separar os desperdícios dos itens), portanto, na abordagem sugerida, as larguras das peças em cada placa de vidro são iguais ou as mais próximas possíveis às larguras dos cortes. Para os experimentos realizados, esta abordagem foi utilizada ao lado de três outros métodos apresentados na literatura (*metaheuristic bottom-left-fill (MBLF), best-fit (BF) + MBLF* e *interactive approach (IA)*) e apenas o tempo de execução foi levado em consideração. A heurística proposta neste artigo apresentou os melhores resultados quando comparada com as outras três.

Uma heurística baseada em *Variable Neighborhood Search* (VNS) para o problema de corte bidimensional em três estágios foi proposta em Dusberger e Raidl [2015]. A heurística utiliza o princípio de destruir e reconstruir, e uma programação dinâmica para gerar vizinhanças. Para construir uma solução inicial, três heurísticas baseadas na abordagem *first-fit* (*3-staged First Fit Decreasing Height with rotations, matching step* e *Fill Strip*) foram usadas, de modo que a melhor solução fornecida por elas foi selecionada. Para reconstruir uma solução "destruída", foi utilizada uma programação dinâmica baseada na abordagem *raster points*. Para os experimentos computacionais foram utilizadas adaptações de instâncias fornecidas na literatura e foi concluído que essa abordagem supera as abordagens VNS que baseiam-se apenas em heurísticas de reconstrução.

Em Furini et al. [2016] foi proposta uma formulação em Programação Linear Inteira para o problema. Essa modelagem consegue lidar com instâncias grandes e requer um número pseudo-polinomial de variáveis e restrições. Todavia, a abordagem não limita o número de estágios que devem ser alcançados para obter um item ao final do processo de corte. Para os experimentos foram levados em consideração o tamanho, a efetividade e a qualidade das soluções obtidas. A abordagem proposta foi capaz de resolver de forma ótima várias instâncias da literatura.

Por fim, uma heurística *diving* baseada no método de geração de colunas de Dantzig-Wolfe para resolver o problema do corte guilhotinado bidimensional foi proposta em Clautiaux et al. [2017]. Considerando que a abordagem de Dantzig-Wolfe gera padrões de corte que não podem participar de uma solução inteira viável para o problema, foi mostrado uma forma de adaptar a heurística *diving* para casos como esse e também como minimizar o número de padrões de corte não adequados mantendo a sua eficácia. Instâncias baseadas em dados reais e da literatura foram utilizados para conduzir os experimentos com o objetivo de avaliar, principalmente, a qualidade

das soluções e o impacto da reutilização das áreas residuais em lotes reais. O método em questão foi capaz de obter resultados melhores que as heurísticas construtivas e evolutivas presentes na literatura.

3. Metodologia

Nesta seção são apresentadas as adaptações feitas nas heurísticas presentes em Puchinger et al. [2004], Suliman [2006] e Aryanezhad et al. [2012] para resolver o problema do corte guilhotinado em 3 estágios com restrições de precedência.

3.1. Uma heurística *first-fit* finita (FFFWS)

A primeira heurística desenvolvida consiste em uma adaptação da heurística FFFWS descrita em Puchinger et al. [2004]. A heurística FFFWS constrói uma solução item a item de maneira gulosa, de modo que os itens são rotacionados de forma a manter horizontalmente a dimensão que possui maior comprimento.

Uma vez que uma placa é selecionada, as placas anteriores não são mais consideradas pelo procedimento. Inicialmente os primeiros itens de todos os lotes são retirados e inseridos de forma ordenada em uma fila de prioridades, onde itens com maior largura são mantidos nas primeiras posições da fila. Em caso de empates, os itens com maior altura têm preferência na fila. Em seguida, o primeiro item da fila é posicionado no canto inferior esquerdo da placa e sua largura é utilizada para definir o tamanho do corte do tipo *1-cut*. Essa nova faixa de material que é definida por esse corte do tipo *1-cut* é então preenchida iterativamente da esquerda para direita.

Caso não seja possível posicionar itens dentro da faixa, uma nova faixa é criada seguindo a lógica anteriormente descrita. Sempre que um elemento é inserido na placa, o próximo elemento do seu lote é inserido de forma ordenada na fila de prioridades. Caso não seja mais possível posicionar itens dentro de uma placa, uma nova placa é inicializada. O Algoritmo 1 descreve o funcionamento da adaptação da heurística FFFWS.

Algorithm 1 Heurística *first-fit* finita (FFFWS)

Input: S = conjunto de lotes contendo os itens a serem cortados
Output: P = Padrão de corte para obtenção de todos os itens

```
Inserir os itens de forma ordenada na fila de prioridades;
Realizar o primeiro corte do tipo 1-cut;
repeat
    if Existe um 1-cut definido then
        if Item atual cabe horizontalmente e verticalmente dentro do 1-cut then
            Inserir item no padrão de corte;
            Remover item da fila de prioridades;
            Inserir próximo item do respectivo lote na fila de prioridades;
        else
            Continuar a iterar pelos itens da fila de prioridades;
        end if
    else
        if É possível realizar outro 1-cut na placa then
            Realizar 1-cut;
        else
            Inicializar outra placa;
        end if
    end if
until A fila de prioridades ficar vazia
```

3.2. Uma heurística sequencial (SHP)

De forma análoga à heurística FFFWS descrita, a adaptação realizada na heurística sequencial SHP presente em Suliman [2006], também insere, de forma ordenada pelo lado de maior comprimento, os primeiros itens de cada lote em uma fila de prioridades. Em seguida, os itens são posicionados, a partir do canto inferior esquerdo, um ao lado do outro preenchendo todo o comprimento da placa, até que não seja mais possível adicionar nenhum outro item. Dessa forma, o comprimento do primeiro corte do tipo *1-cut* é definido.

Para preencher a faixa de material definida pelo corte do tipo *1-cut*, o procedimento busca por itens de dimensões iguais aos utilizados anteriormente para repetir o padrão de corte que foi gerado. Caso não encontre, novos itens são posicionados de forma iterativa na faixa até que não seja mais possível incluir nenhum outro item. Os passos acima descritos se repetem até que seja gerado um padrão de corte para obter todos os itens necessários. O Algoritmo 2 apresenta a adaptação da heurística SHP, conforme procedimento descrito acima.

Algorithm 2 Um procedimento heurístico sequencial (SHP)

Input: S = conjunto de lotes contendo os itens a serem cortados
Output: P = Padrão de corte para obtenção de todos os itens

```
Inserir de forma ordenada na fila de prioridades;
repeat
    if Existe um 1-cut definido then
        if Item atual cabe dentro do 1-cut horizontalmente then
            if Item atual cabe dentro do 1-cut verticalmente then
                Inserir item no padrão de corte geral;
                Inserir item no padrão de corte temporário;
                Remover item da fila de prioridades;
                Inserir próximo item do lote na fila de prioridades;
            else
                Continuar a iterar pelos itens da fila de prioridades;
            end if
        else
            if É o último elemento da fila de prioridades then
                Realizar 1-cut;
                if A fila de prioridades têm itens iguais aos contidos no padrão temporário then
                    Inserir os itens contidos no padrão de corte temporário no padrão de corte geral;
                    Remover os itens inseridos da fila de prioridades;
                    Esvaziar padrão de corte temporário;
                else
                    Esvaziar padrão de corte temporário;
                end if
            else
                Continuar a iterar pelos itens da fila de prioridades;
            end if
        end if
    else
        if É possível realizar outro 1-cut na placa then
            Repetir procedimento para inserir itens;
        else
            Inicializar outra placa;
        end if
    end if
until A fila de prioridades ficar vazia
```

3.3. Uma heurística *first-fit* com atribuição de pesos (FFRPW)

A adaptação na heurística FFRPW, descrita em Aryanezhad et al. [2012], também utiliza-se de critérios para a fila de prioridades e é dividida em três etapas:

1. Atribuir pesos aos itens a serem cortados. O peso de um item i de largura w_i e altura h_i pode ser obtido através do cálculo da área do item.

2. Os itens são inseridos em ordem decrescente na fila de prioridades. Caso os itens possuam mesmo peso, aquele com o maior lado é inserido primeiro na fila.

3. Posicionar o item de maior peso no canto inferior esquerdo da placa, definindo assim o comprimento do primeiro corte do tipo *1-cut* e iterativamente preencher a faixa de material definida pelo corte do tipo *1-cut* através da inclusão dos demais itens presentes na fila de prioridades no padrão de corte a ser gerado.

Esse procedimento é repetido até que todos os itens estejam inclusos no padrão de corte final gerado pelo algoritmo, conforme exibido no Algoritmo 3.

Algorithm 3 Uma heurística *first-fit* com atribuição de pesos (FFRPW)

Input: S = conjunto de lotes contendo os itens a serem cortados
Output: P = Padrão de corte para obtenção de todos os itens

```
Calcular peso dos itens;
Inserir de forma ordenada na fila de prioridades;
Realizar o primeiro corte do tipo 1-cut;
repeat
    if Existe um 1-cut definido then
        if Item atual cabe dentro do 1-cut horizontalmente then
            if Item atual cabe dentro do 1-cut verticalmente then
                Inserir item no padrão de corte;
                Remover item da fila de prioridades;
                Inserir próximo item do lote na fila de prioridades;
            else
                Continuar a iterar pelos itens da fila de prioridades;
            end if
        else
            Continuar a iterar pelos itens da fila de prioridades;
        end if
    else
        if É possível realizar outro 1-cut na placa then
            Realizar 1-cut;
        else
            Inicializar outra placa;
        end if
    end if
until A fila de prioridades ficar vazia
```

4. Experimentos Computacionais

Para a realização dos experimentos utilizou-se um computador com processador Intel Core i7-4510U 2.60GHZ com 8 GB de Memória RAM. As instâncias utilizadas nos experimentos realizados foram fornecidas pela *Saint-Gobain*, multinacional francesa que trabalha com manufatura de vidro, para o desafio ROADEF 2018, que foi organizado pela sociedade francesa de Pesquisa Operacional e apoio à decisão (*French Operations Research & Decision Support Society*).

Todas as três heurísticas apresentadas na seção anterior foram avaliadas utilizando as 20 instâncias presentes no desafio, de modo que o percentual de desperdício de vidro e o tempo de execução dos algoritmos foram utilizados como métricas para avaliar a qualidade das soluções geradas. A Tabela 1 abaixo apresenta o percentual de desperdício e o tempo de execução para cada uma das três heurísticas propostas.

Instância	FFFWS		SHP		FFRPW	
	Desemp. (%)	Tempo (s)	Desemp. (%)	Tempo (s)	Desemp. (%)	Tempo (s)
A1	53.14	**1.27**	29.53	1.78	**22.31**	1.38
A2	**31.67**	**1.01**	33.41	1.04	32.63	1.51
A3	39.94	2.24	**22.85**	1.35	25.16	**0.97**
A4	39.21	1.21	25.71	**1.18**	**21.84**	2.09
A5	34.75	**1.25**	31.87	1.45	**24.23**	1.35
A6	32.33	1.1	27.35	1.61	**20.65**	**0.86**
A7	43.62	1.07	34.32	**1.01**	**25.52**	1.58
A8	35.85	1.76	25.73	**0.96**	**17.74**	1.05
A9	46.11	**1.03**	23.47	1.54	**20.83**	1.97
A10	38.91	2.58	26.84	2.01	**23.81**	**1.36**
A11	33.27	1.05	27.26	**0.81**	**19.28**	1.19
A12	45.16	1.07	31.86	**0.87**	**26.76**	2.13
A13	28.94	**0.94**	25.05	1.79	**20.82**	1.99
A14	28.43	1.89	24.92	1.52	**21.26**	**1.34**
A15	34.69	1.36	**19.45**	1.66	28.42	**1.05**
A16	31.41	1.31	**16.83**	**1.06**	21.36	1.44
A17	46.28	1.63	34.95	1.05	**21.57**	**0.77**
A18	42.5	1.58	27.48	**0.82**	**24.82**	0.9
A19	44.12	**0.89**	34.82	2.65	**25.72**	1.21
A20	54.07	**0.88**	43.23	1.69	**28.42**	1.1
Média	39.22	**1.36**	28.35	1.4	**23.66**	1.37

Tabela 1: Percentual de desperdício de vidro das três heurísticas para cada uma das vinte instâncias e percentual médio de desperdício.

Por meio da análise dos dados contidos na Tabela 1, é possível concluir que a adaptação na heurística FFRPW obteve os melhores resultados com uma média de 23.66% de desperdício de vidro e a adaptação na heurística FFFWS obteve os piores resultados com um percentual médio de desperdício de 39.22%. Também é possível perceber que a adaptação realizada em SHP, com percentual de desperdício médio de 28.35%, obteve resultados melhores que FFRPW em três instâncias. As médias dos tempos de execução (em segundos) de todas as 3 heurísticas foram similares, com FFFWS obtendo a melhor média de 1.36 segundos.

5. Conclusões

O presente trabalho apresentou adaptações de três heurísticas existentes na literatura para a resolução do problema de corte bidimensional em três estágios com restrições de precedência. Todas as três heurísticas são construtivas e geram padrões de cortes seguindo critérios gulosos.

Para comparar empiricamente os três algoritmos, experimentos utilizando uma base de dados oferecida pela *Saint-Gobain* para o desafio ROADEF 2018 foram planejados, executados e

analisados. Foram observados o percentual médio de desperdício de vidro e o tempo de execução (em segundos) em todas as vinte instâncias para as três heurísticas.

Após a realização de todos os experimentos foi possível concluir que, de modo geral, a adaptação realizada em FFRPW obteve os melhores resultados com uma média de 23.66% de desperdício. A modificação realizada em SHP obteve resultados bem próximos com uma média de desperdício de 28.35%. A adaptação feita em FFFWS obteve os piores resultados com um percentual médio de desperdício de 39.22%. A grande diferença nos resultados obtidos em FFRPW e SHP quando comparados a FFFWS pode ser justificada pelos critérios mais sofisticados atribuição de pesos e inserção dos itens nos padrões de corte gerados das duas primeiras heurísticas quando comparadas à terceira.

As médias dos tempos de execução de todas as 3 heurísticas foram similares, com SHP obtendo 1.4 segundos, FFRPW obtendo 1.37 segundos e FFFWS obtendo a melhor média de 1.36 segundos. Esse fato pode ser justificado pela semelhança estrutural das heurísticas comparadas.

Para trabalhos futuros, pode-se pensar em utilizar metaheurísticas que podem ser executadas a partir dos resultados obtidos nos procedimentos heurísticos aqui propostos a fim de melhorar as soluções encontradas.

Referências

Aryanezhad, M.-B., Hashemi, N. F., Makui, A., e Javanshir, H. (2012). A simple approach to the two-dimensional guillotine cutting stock problem. *Journal of Industrial Engineering International*, p. 8–21.

Clautiaux, F., Sadykov, R., Vanderbeck, F., e Viaud, Q. (2017). Pattern based diving heuristics for a two-dimensional guillotine cutting-stock problem with leftovers. p. 1–30.

Dusberger, F. e Raidl, G. R. (2015). Solving the 3-staged 2-dimensional cutting stock problem by dynamic programming and variable neighborhood search. *Electronic Notes in Discrete Mathematics*, 47:133–140.

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

Puchinger, J., Raidl, G. R., e Koller, G. (2004). Solving a real-world glass cutting problem. In *European Conference on Evolutionary Computation in Combinatorial Optimization*, p. 165–176. Springer.

Suliman, S. (2006). A sequential heuristic procedure for the two-dimensional cutting-stock problem. *International Journal of Production Economics*, 99(1-2):177–185.

Takahashi, S., Horiuchi, K., Tatsukoshi, K., Ono, M., Imajo, N., e Mobely, T. (2013). Development of through glass via (tgv) formation technology using electrical discharging for 2.5/3d integrated packaging. In *Electronic Components and Technology Conference (ECTC), 2013 IEEE 63rd*, p. 348–352. IEEE.