

**UNIVERSIDADE ESTADUAL PAULISTA "JÚLIO DE MESQUITA FILHO"**

FACULDADE DE CIÊNCIAS - CAMPUS BAURU

DEPARTAMENTO DE COMPUTAÇÃO

BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO

KAWÉ ANTÔNIO DOS SANTOS MARCELINO

**GERAÇÃO DE PADRÕES DE CORTE BIDIMENSIONAIS COM  
ITENS REGULARES E IRREGULARES DO TIPO L**

BAURU

Novembro/2019

Kawe Antônio dos Santos Marcelino

## **GERAÇÃO DE PADRÕES DE CORTE BIDIMENSIONAIS COM ITENS REGULARES E IRREGULARES DO TIPO L**

Trabalho de Conclusão de Curso do Curso  
de Ciência da Computação da Universidade  
Estadual Paulista “Júlio de Mesquita Filho”,  
Faculdade de Ciências, Campus Bauru.  
Orientador: Profa. Dra. Andréa Carla Gonçalves  
Vianna

BAURU  
Novembro/2019

Kawe Antônio dos Santos Marcelino      Geração de padrões de corte bidimensionais com itens regulares e irregulares do tipo L/ Kawe Antônio dos Santos Marcelino. – Bauru, Novembro/2019-      42 p. : il. (algumas color.) ; 30 cm.  
Orientador: Profa. Dra. Andréa Carla Gonçalves Vianna  
Trabalho de Conclusão de Curso – Universidade Estadual Paulista “Júlio de Mesquita Filho”  
Faculdade de Ciências  
Ciência da Computação, Novembro/2019.  
1. Tags 2. Para 3. A 4. Ficha 5. Catalográfica

Kawe Antônio dos Santos Marcelino

## **Geração de padrões de corte bidimensionais com itens regulares e irregulares do tipo L**

Trabalho de Conclusão de Curso do Curso de Ciência da Computação da Universidade Estadual Paulista "Júlio de Mesquita Filho", Faculdade de Ciências, Campus Bauru.

Banca Examinadora

---

**Profa. Dra. Andréa Carla Gonçalves  
Vianna**

Orientador

Universidade Estadual Paulista "Júlio de  
Mesquita Filho"

Faculdade de Ciências

Departamento de Computação

---

**Profa. Dra. Simone das Graças  
Domingues Padro**

Universidade Estadual Paulista "Júlio de  
Mesquita Filho"

Faculdade de Ciências

Departamento de Computação

---

**Profa. Dra. Adriana Cristina Cherri**

Universidade Estadual Paulista "Júlio de  
Mesquita Filho"

Faculdade de Ciências

Departamento de Computação

Bauru, \_\_\_\_\_ de Novembro de 2019.

*À minha família que me proporcionou tudo o que precisava para chegar até onde cheguei.*

# Agradecimentos

Agradeço primeiramente à minha família que sempre me inspiraram e deram todo o suporte necessário durante toda a minha infância até este momento para que pudesse perseguir e conquistar meus sonhos.

Gostaria de agradecer também a todos os meus amigos de sala e de faculdade, que sempre me ajudaram em tudo que precisei durante esses anos de faculdade. Conheci pessoas maravilhosas e irei levar vocês para a vida toda.

Agradeço também à minha orientadora por todo o suporte para que eu pudesse começar, desenvolver e concluir este trabalho, me atendendo em todas as horas que necessitei e me ensinando a todo momento.

Agradeço, por fim, à Unesp, o Departamento de Computação e aos professores que dispuseram de seu tempo para mim e todo à minha turma e ter proporcionado todas as condições para que eu pudesse concluir o curso.

*O passado pode doer, mas do jeito que eu vejo você pode fugir dele ou aprender com ele*  
Rafiki (Rei Leão, 1994)

# Resumo

Diversos tipos de indústrias têm como característica o corte de matéria prima em itens menores de diferentes tamanhos e com pouca padronização, de forma à atender a demanda interna ou externa. Para elas é essencial um bom planejamento na linha de produção, afim de evitar desperdícios de material e, conseqüentemente, aumentar lucros e melhorar a competitividade. Na literatura, este problema é conhecido como Problema de Corte e Empacotamento, que consiste em encontrar a melhor forma de alocar itens menores em objetos maiores otimizando um determinado objetivo (e.g., maximizar o valor de utilidade total, minimizar a perda, etc.). Um tipo de Problema de Corte pouco estudado é o Problema de Corte Bidimensional com peças do tipo-L. Este tipo de problema, em razão das presença de peças irregulares, tem alto grau de complexidade de resolução, mas também um alto grau de utilidade para as empresas. Este trabalho se propõe a estudar mais o tema e apresentar um método 2-estágios com combinação de peças irregulares para implementar um gerador de padrões de corte para problemas de corte bidimensional com peças regulares e irregulares do tipo-L. Além da proposição do método, foi realizado um levantamento de instâncias da literatura para avaliar o desempenho do método perante os problemas de teste.

**Palavras-chave:** Problema de Corte Bidimensional; Análise combinatória; geração de padrão de corte.



# Abstract

Several types of industries are characterized by the process of cutting raw material into smaller items of different sizes and with little standardization, in order to fulfill internal or external demand. For them a good production planning is essential in order to avoid material waste and, consequently, increase profits and improve competitiveness. In the literature, this problem is known as Cutting Stock Problem, which consists of finding the best way to allocate smaller items into larger objects by optimizing a given objective (e.g., maximize the total utility value, minimize loss, etc.). One class of Cutting Problem poorly studied in the literature is the 2-dimensional Cutting Problem with L-shaped pieces. This problem, due to the L-shaped pieces, has a high degree of complexity to be solved, but also has a high degree of interest to companies. This paper aims to study the 2-dimensional Cutting Problem with L-shaped pieces and propose a new method of 2-stages with combined pieces to implement a cutting pattern generator for 2-dimensional cutting problems with irregular pieces. In addition to the proposition of the new method, a collection of 2-dimensional cutting problem instances was collected in the literature to evaluate the method performance and compare the results with the literature.

**Keywords:** Two-dimensional Cutting Problem; Combinatory analysis; cutting pattern generation.

# Lista de figuras

Figura 1 – Exemplo de PC2D. . . . .	14
Figura 2 – Exemplo de padrão de corte. . . . .	15
Figura 3 – Exemplos de corte guilhotinado. . . . .	15
Figura 4 – Padrões de corte guilhotinado. . . . .	16
Figura 5 – Padrões de corte guilhotinado de (a) 2-estágios e (b) 4-estágios. . . . .	16
Figura 6 – Tipos de corte degrau. . . . .	21
Figura 7 – Exemplo de problema. . . . .	21
Figura 8 – Exemplo de padrão de corte com peças retangulares e do tipo-L. . . . .	22
Figura 9 – Processo de transformação de peças do tipo-L. . . . .	24
Figura 10 – Combinação de peças L com peças retangulares: (a) combinação na região de “dentro”; (b) combinação no “topo”. . . . .	25
Figura 11 – Combinações LR permitidas. . . . .	25
Figura 12 – Combinações LR não permitidas. . . . .	26
Figura 13 – Exemplos de combinação LL. . . . .	26
Figura 14 – Exemplo de solução homogênea. . . . .	27
Figura 15 – Exemplo de solução inicial homogênea composta. . . . .	28
Figura 16 – Padrão de corte para a instância <i>gcut13</i> . . . . .	34
Figura 17 – Padrão de corte para a instância <i>Xu</i> (irrestrito). . . . .	38
Figura 18 – Padrão de corte para a instância <i>Xu</i> (restrito). . . . .	38
Figura 19 – Padrão de corte para a instância <i>Nakatake</i> (irrestrito). . . . .	39
Figura 20 – Padrão de corte para a instância <i>Nakatake</i> (restrito). . . . .	39

# Lista de tabelas

Tabela 1 – Resultados para problemas do grupo 1. . . . .	33
Tabela 2 – Dados para os problemas do grupo 2. . . . .	35
Tabela 3 – Resultados para o grupo 2. . . . .	35
Tabela 4 – Dados para a instância <i>Nakatake</i> . . . . .	36
Tabela 5 – Dados para a instância <i>Xu</i> . . . . .	37

# Lista de abreviaturas e siglas

PC1D	Problema de Corte Unidimensional
PC2D	Problema de Corte Bidimensional
PC2DI	Problema de Corte Bidimensional Irrestrito
PC2DR	Problema de Corte Bidimensional Restrito
PC2D2E	Problema de Corte Bidimensional 2-estágios

# Sumário

<b>1</b>	<b>INTRODUÇÃO</b>	<b>13</b>
<b>2</b>	<b>PROBLEMA DE CORTE BIDIMENSIONAL</b>	<b>14</b>
2.1	Descrição do Problema	14
2.2	Resolução do Problema de Corte 2-estágios	16
2.2.1	Resolução do Problema de Corte 2-estágios Restrito	19
2.2.2	Problema de Corte Bidimensional com peças do tipo-L	21
<b>3</b>	<b>MÉTODO PROPOSTO</b>	<b>23</b>
3.1	Problema da Mochila	23
3.2	Resolução do Problema de Corte Bidimensional com peças do tipo-L	24
<b>4</b>	<b>IMPLEMENTAÇÃO COMPUTACIONAL</b>	<b>30</b>
4.1	Ferramentas utilizadas	30
4.1.1	Linguagem <i>Python</i>	30
4.1.2	CPLEX	30
4.1.3	TeXLive e TikZ	31
4.2	Estrutura do programa	31
<b>5</b>	<b>RESULTADOS</b>	<b>33</b>
<b>6</b>	<b>CONCLUSÃO</b>	<b>40</b>
	<b>REFERÊNCIAS</b>	<b>41</b>

# 1 Introdução

Diversos tipos de indústrias têm como característica o corte de matéria prima em itens menores de diferentes tamanhos e com pouca padronização, de forma à atender a demanda interna ou externa. Indústrias de aço, alumínio, vidro, madeira e tecido são alguns exemplos que têm esta característica. Para elas é essencial um bom planejamento na linha de produção, afim de evitar desperdícios de material e, conseqüentemente, aumentar lucros e melhorar a competitividade.

Na literatura, este problema é conhecido como Problema de Corte e Empacotamento (DYCKHOFF, 1990). O Problema de Corte consiste, basicamente, em cortar uma unidade maior (objeto ou placa) dividindo-a em unidades menores (itens) otimizando determinado objetivo, como minimizar a perda de material. De forma relacionada, o Problema de Empacotamento consiste na alocação de unidades menores em uma unidade maior - como um contêiner, por exemplo - otimizando também uma função objetivo, como minimizar o espaço ocioso.

Em razão da alta aplicabilidade para pesquisa, diversidade de problemas reais em grande variedade e complexidade na solução destes problemas, o Problema de Corte e Empacotamento atrai muitos pesquisadores por todo mundo, como apontado por Bischoff e Wäscher (1995) e os estudos de Brown (1971), Golden (1976), Arenales, Morabito e Yanasse (1999) e Dyckhoff e Finke (1992).

Há vários tipos de problemas de corte, entretanto há um tópico pouco estudado pela literatura (VIANNA, 2000; LINS; LINS; MORABITO, 2003) que envolve cortes bidimensionais com peças regulares e irregulares, em especial peças com formato em L. Este tópico é abordado neste trabalho.

No Capítulo 2, o Problema de Corte Bidimensional é apresentado com mais detalhes. Além disso, há uma definição do Problema de Corte Bidimensional com peças do tipo-L, tema deste trabalho, e um método de resolução em 2-estágios muito utilizado na indústria para problemas da tipo bidimensional.

O Capítulo 3 e 4 tratam, respectivamente, do método proposto para a resolução de problemas de corte bidimensional com peças L e da implementação computacional do método.

O Capítulo 5 discorre sobre o desempenho do método implementado comparando os resultados com algumas instâncias da literatura.

Por fim, o Capítulo 6 apresenta as conclusões do trabalho e passos que podem ser realizados no futuro.

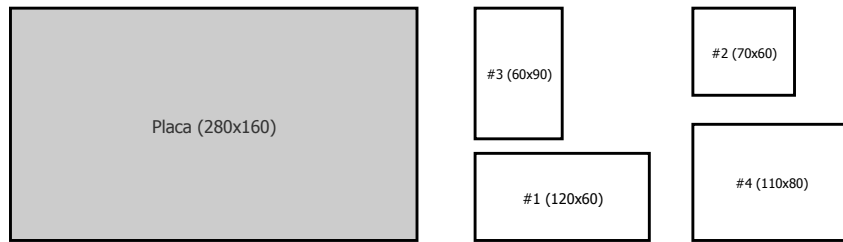
## 2 Problema de Corte Bidimensional

Neste Capítulo é apresentado uma descrição mais detalhada do Problema de Corte Bidimensional e o método de resolução em dois estágios.

### 2.1 Descrição do Problema

O Problema de Corte Bidimensional (PC2D) consiste em cortar uma placa retangular (objeto) em peças retangulares menores (itens) otimizando algum objetivo (e.g., minimizar a perda de matéria-prima, maximizar o valor de utilidade, entre outros). A Figura 1 mostra um exemplo de PC2D com placa e itens retangulares.

Figura 1 – Exemplo de PC2D.



Fonte: Elaborado pelo autor.

Considere uma placa retangular de dimensões  $(L, W)$ , onde  $L$  e  $W$  representam, respectivamente, o comprimento e a largura do objeto; e um conjunto de  $m$  peças retangulares menores de dimensões  $(l_i, w_i)$ , onde  $l_i$  representa o comprimento da peça  $i$  e  $w_i$  a largura da mesma. Além disso, cada peça  $i$  tem um valor  $\alpha_i$  associado, que neste caso será a área da peça. O objetivo é maximizar a área total ocupada pelas peças, portanto estaremos também minimizando a perda de matéria-prima. Seja  $x_i$  o número de vezes que a peça  $i$  é produzida, temos:

$$\begin{aligned} & \text{maximizar } \sum_{i=1}^m \alpha_i \cdot x_i \\ & \text{sujeito a: } (x_1, x_2, \dots, x_m) \text{ correspondente a um padrão de corte} \\ & x_i \geq 0, \text{ inteiro, } i = 1, \dots, m \end{aligned}$$

O resultado do modelo matemático descrito anteriormente define um padrão de corte. Padrão de corte é o modo como as peças estão organizadas ao longo da placa (Figura 2).

Figura 2 – Exemplo de padrão de corte.



Fonte: Elaborado pelo autor.

Considerando apenas essas informações (dimensão da placa, conjunto de itens e o valor da área de cada item), o problema é classificado como Problema de Corte Bidimensional Irrestrito (PC2DI). Ou seja, não há um limite (máximo ou mínimo, dependendo do tipo de problema) da quantidade que cada peça pode ser produzida. Caso haja restrição na quantidade de peças a ser produzida, o problema é chamado de Problema de Corte Bidimensional Restrito (PC2DR), que pode ser definido da seguinte maneira:

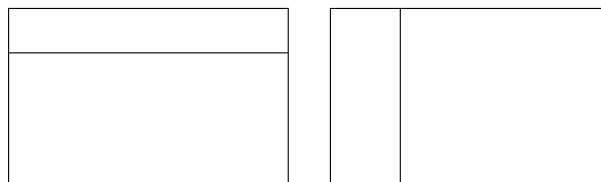
$$\begin{aligned} & \text{maximizar } \sum_{i=1}^m \alpha_i \cdot x_i \\ & \text{sujeito a: } (x_1, x_2, \dots, x_m) \text{ correspondente a um padrão de corte} \\ & 0 \leq x_i \leq b_i, \text{ inteiro, } i = 1, \dots, m \end{aligned}$$

em que  $b_i$  é a quantidade máxima de peças do tipo  $i$  no padrão de corte.

Apesar de ser uma alteração mínima no modelo matemático em relação ao PC2DI, a limitação na quantidade de peças no padrão acrescenta bastante complexidade na resolução do problema (VIANNA, 2000).

Em muitos Problemas de Corte, o maquinário (serras) utilizado aplica cortes nos objetos denominados *cortes guilhotinados*.

Figura 3 – Exemplos de corte guilhotinado.

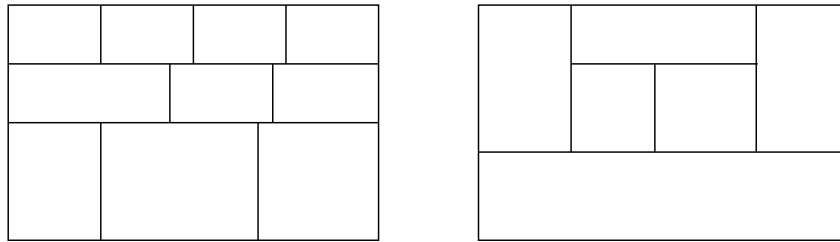


Fonte: Elaborado pelo autor.

O corte é chamado guilhotinado quando, aplicado em um retângulo, produz dois novos retângulos, ou seja, o retângulo é cortado somente de modo horizontal ou vertical, conforme a Figura 3. Um padrão de corte é do tipo guilhotinado se for obtido por cortes guilhotinados sucessivos (Figura 4).



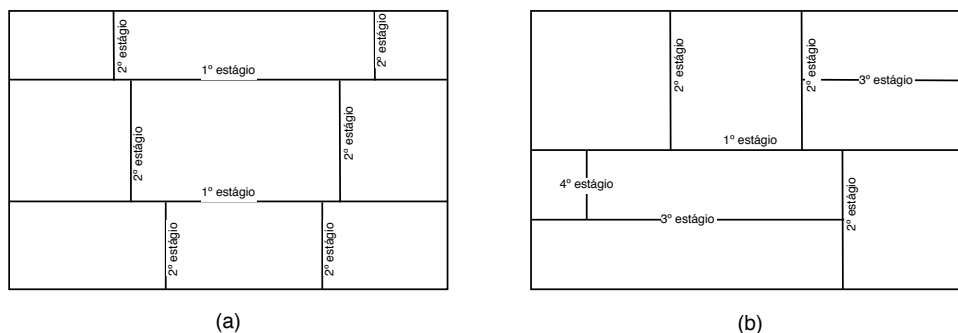
Figura 4 – Padrões de corte guilhotinado.



Fonte: Elaborado pelo autor.

Para gerar um padrão de corte guilhotinado, os cortes são feitos, no primeiro estágio, de forma paralela a um dos lados da placa retangular; no próximo estágio, os cortes são perpendiculares aos cortes do estágio anterior, e assim por diante até finalizar o processo da serra. Caso haja um limite  $k$  de estágios, o problema é chamado de Problema de Corte Guilhotinado  $k$ -estagiado. Caso contrário, é chamado de Problema de Corte Guilhotinado não-estagiado. A Figura 5 ilustra padrões de corte guilhotinado de 2 e 4 estágios.

Figura 5 – Padrões de corte guilhotinado de (a) 2-estágios e (b) 4-estágios.



Fonte: Elaborado pelo autor.

O Problema de Corte Guilhotinado foi definido no artigo de [Gilmore e Gomory \(1965\)](#) e, através dos estudos dos dois autores, foi apresentado um método de resolução do Problema de Corte Bidimensional 2-estágios (PC2D2E). Na próxima seção é apresentado o método e sua aplicação em problemas de corte bidimensional.

## 2.2 Resolução do Problema de Corte 2-estágios

O método de 2-estágios de [Gilmore e Gomory](#) é um método para resolução de problemas de corte bidimensional, como mostrado no artigo de [1965](#). Um dos motivos da proposição deste método é a complexidade computacional do método de geração de colunas, que apresenta um número muito grande de colunas no seu modelo matemático.

O método de 2-estágios se utiliza de duas ferramentas: o **Problema da Mochila** e o paradigma de **Divisão e Conquista**. O problema da mochila é um problema clássico da literatura de Otimização e Análise Combinatória que será explicado no próximo capítulo. O paradigma de divisão e conquista é uma ferramenta usado em algoritmos como *merge sort* e *quicksort* ([KHAN ACADEMY](#)), por exemplo, para facilitar a resolução de problemas de grande complexidade computacional. A ideia é quebrar um processo complexo em pequenos processos de menor complexidade e de fácil resolução computacional ([TUTORIALSPPOINT](#)). A seguir uma pequena explicação de como a Divisão e Conquista funciona:

- **Dividir** o problema em subproblemas menores similares ao problema original, mas muito mais fáceis e rápidos de solucionar;
- **Conquistar** os subproblemas ao resolvê-los separadamente;
- **Combinar** a solução de todos os subproblemas de forma que o resultado final seja a solução do problema original, caso o mesmo fosse resolvido como um problema único.

A resolução do PC2D2E utiliza o mesmo conceito: divide um PC2D em vários subproblemas de Problema de Corte Unidimensional (PC1D); cada um desses subproblemas será resolvido como um problema da mochila; e, após a solução de todos os subproblemas, seus resultados serão usados para resolver um último PC1D.

Seja uma placa retangular de dimensões  $(L, W)$  e um conjunto de  $m$  peças de dimensões  $(l_i, w_i)$ ,  $i = 1, \dots, m$ , que devem ser produzidas a partir da placa. Este problema pode ser formulado como:

$$\begin{aligned} &\text{maximizar } \sum_{i=1}^m \alpha_i \cdot x_i \\ &\text{sujeito a: } (x_1, x_2, \dots, x_m) \text{ correspondente a um padrão de corte} \\ &x_i \geq 0, \text{ inteiro, } i = 1, \dots, m \end{aligned}$$

em que  $\alpha_i$  e  $x_i$  representam, respectivamente, a área da peça  $i$  e o número de vezes que a peça  $i$  aparece no padrão de corte. O padrão de corte será representado pelo vetor  $x = (x_1, \dots, x_m)$ .

A resolução para o mesmo problema de corte formulado anteriormente pelo método de 2-estágios pode ser realizada da seguinte forma:

*Etapa 1 - Dividir o problema original em subproblemas PC1D:*

Através do conjunto  $m$  de peças do problema original, definir o conjunto  $F$  de faixas; cada faixa terá dimensões de comprimento  $L$  e largura  $w_i$ ,  $(i = 1, \dots, m)$ . Como podem haver peças de mesma largura, selecionar as  $p$  faixas com larguras diferentes. Desta forma, o conjunto  $F$  é definido por  $F = \{L \times w_1, L \times w_2, \dots, L \times w_p\}$ . Após isso, para cada faixa  $L \times w_f$ ,

define-se o conjunto  $W_f = \{i \text{ tal que } w_i \leq w_f\}$ . Este conjunto restringe os itens que podem ser cortados numa faixa, ou seja, que respeitam a largura  $w_f$  da faixa.

Depois de definido o conjunto de faixas e o conjunto de itens para cada uma das faixas, o objetivo é cortar as faixas de modo que cada uma seja aproveitada da melhor maneira possível. Assim, para cada faixa  $f$  do conjunto  $F$ , deve ser resolvido o seguinte problema da mochila:

$$\begin{aligned} &\text{maximizar } V_f = \sum_{i \in W_f} \alpha_i \cdot x_i \\ &\text{sujeito a: } \sum_{i \in W_f} l_i \cdot x_i \leq L \\ &x_i \geq 0, \text{ inteiro, } i \in W_f \end{aligned}$$

em que  $\alpha_i$  é a área da peça  $i$  e  $x_i$  é o número de vezes que a peça  $i$  foi produzida na faixa  $f$ . Após a resolução dos  $p$  problemas, realiza-se a segunda etapa.

*Etapa 2 - Determinar o padrão de corte bidimensional selecionando as melhores faixas encontradas:*

Com os resultados dos  $p$  problemas da mochila, resolve-se mais um problema da mochila, agora combinado as faixas geradas no comprimento  $W$ . Na etapa anterior, o objetivo foi alocar da melhor maneira os itens limitando-os pela largura da faixa ( $L$ ). Agora, o foco está em como alocar as faixas da melhor maneira, limitando-as pelo comprimento da placa ( $W$ ). Desta forma, podemos selecionar as melhores faixas e construir um padrão de corte bidimensional, solucionando o problema original.

Com todas essas considerações, o problema da mochila final será:

$$\begin{aligned} &\text{maximizar } V = \sum_{i=1}^p V_i \cdot x_i \\ &\text{sujeito a: } \sum_{i=1}^p w_i \cdot x_i \leq W \\ &x_i \geq 0, \text{ inteiro, } i = 1, \dots, p \end{aligned}$$

em que  $V_i$  é a área alocada por itens da faixa  $i$  e  $x_i$  é o número de vezes que a faixa  $i$  é produzida. O resultado do problema fornece o padrão de corte 2-estágios que melhor aproveita a área da placa.

Este método geralmente é aplicado em problemas de corte com placa e peças retangulares, mas há outros problemas de corte em que o mesmo pode ser utilizado. Neste trabalho, o objetivo é utilizar o método de 2-estágios para resolver problemas de corte bidimensional com peças retangulares e peças irregulares do tipo-L. Essa subclasse dos problemas de corte bidimensional é apresentada na Seção 2.2.2.

### 2.2.1 Resolução do Problema de Corte 2-estágios Restrito

O método de 2-estágios com restrição de demanda pode ser descrito da seguinte maneira, conforme (VIANNA; ARENALES; GRAMANI, 2003):

$$\begin{aligned}
 & \text{maximizar } \sum_{k=1}^K \sum_{p=1}^{P_k} \sum_{i \in I_k} v_i \cdot \lambda_{kp}^i \cdot \mu_{kp} \\
 & \text{sujeito a: } \sum_{i \in I_k} l_i \cdot \lambda_{kp}^i \leq L, \text{ para } k = 1, \dots, K, p \in P_k \\
 & \sum_{k=1}^K w_k \sum_{p=1}^{P_k} \mu_{kp} \leq W \\
 & \sum_{k=1}^K \sum_{p=1}^{P_k} \lambda_{kp}^i \cdot \mu_{kp} \leq b_i, \text{ para } i = 1, \dots, m \\
 & \lambda_{kp}^i, \mu_{kp} \geq 0, \text{ inteiro, } i = 1, \dots, m, k = 1, \dots, K, p = 1, \dots, P_k
 \end{aligned}$$

em que:

- $L$  e  $W$  representam, respectivamente, o comprimento e largura da placa;
- $l_i$ ,  $w_i$  e  $b_i$  definem, respectivamente, o comprimento, largura e demanda da peça do tipo  $i$ ;
- $K$  é o número de larguras diferentes  $w_i$ ;
- $P_k$  define o número máximo possível de faixas  $L \times w_k$  que podem ser alocadas na placa;
- $I_k$  é o conjunto de itens que podem ser alocadas na faixa  $k$  ( $I_k = \{i \mid w_i \leq w_k\}$ );
- $v_i$  é um valor associado à peça do tipo  $i$ ;
- $\lambda_{kp}^i$  é o número de vezes que a peça  $i$  é alocada no  $p$ -ésimo padrão da faixa  $k$ ;
- $\mu_{kp}$  define o número de vezes que o  $p$ -ésimo padrão da faixa  $k$  foi alocada ao longo da largura  $W$  da placa;

Conforme mostrado pelo modelo matemático, a aplicação do método de 2-estágios para resolver problemas de corte bidimensional restrito cria um problema de otimização não linear. Para contornar a não linearidade e não realizar nenhuma linearização, Vianna, Arenales e Gramani (2003) utilizaram uma heurística, que tem como base as etapas citadas na seção anterior, entretanto com algumas alterações para viabilizar a resolução do problema.

Seja uma placa de dimensões  $L \times W$ , um conjunto de  $m$  peças (itens) de dimensões  $l_i \times w_i$  com demanda  $b_i$  e valor de utilidade  $v_i$ , o objetivo é maximizar a área total ocupada pelos itens na placa respeitando as demandas de cada uma das peças.

Segue uma heurística para resolução do problema:

**Passo 1** Determinar um vetor de demandas e uma variável de controle.

Criar um vetor  $B$  com todas as demandas das peças ( $b_i, i = 1, \dots, m$ ) e uma variável  $controle = W$ . A variável de controle será responsável por limitar a geração de faixas que não podem ser encaixadas na placa, conforme o procedimento está sendo executado.

**Passo 2** Criar  $F$  faixas de largura diferente conforme variável de controle.

Criar  $F$  faixas para as larguras diferente no conjunto de peças, desde que  $w_f \leq controle$  para  $f = 1, \dots, F$ . Além das faixas, determinar o conjunto de itens que cada faixa  $f$  poderá utilizar:

$$W_f = \{i \mid w_i \leq w_f\}$$

**Passo 3** Se nenhuma faixa foi gerada, ir para o **Passo 7**.

**Passo 4** Resolver cada faixa como um problema da mochila.

Para cada uma das faixas geradas no **Passo 2**, resolver um problema da mochila conforme o seguinte modelo matemático:

$$\begin{aligned} &\text{maximizar } V_f = \sum_{i \in W_f} v_i \cdot x_i \\ &\text{sujeito a: } \sum_{i \in W_f} l_i \cdot x_i \leq L \\ &0 \leq x_i \leq b_i, \text{ inteiro, } i \in W_f \end{aligned}$$

em que  $x_i$  é o número de vezes que a peça do tipo  $i$  foi produzida e  $V_f$  é valor ótimo encontrado resolvendo o problema da mochila. Observa-se que a demanda  $b_i$  da peça  $i$  é obtida através do vetor de demandas  $B$  e caso a mesma seja igual a zero, a mesma pode ser descartada do modelo, pois sua demanda já foi atendida.

**Passo 5** Selecionar a faixa com o maior valor de utilidade e alocá-la na placa.

Selecionar a faixa com o maior valor ótimo e alocá-la na placa, decrementando do vetor de demandas e da variável de controle o vetor  $x$  da solução e a largura  $w$  da melhor faixa.

**Passo 6** Repetir os **Passos 2-6** até que a variável de controle não permita mais que uma faixa seja gerada, ou seja, as larguras das faixas são todas maiores que a variável de controle.

**Passo 7** Solução encontrada e fim do procedimento.

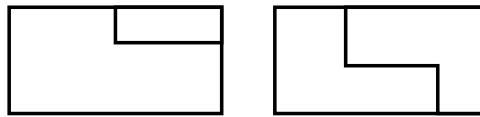
A solução final será encontrada no conjunto de vetores  $x$ s e a somatória do valor ótimo da cada faixa alocada na placa.

### 2.2.2 Problema de Corte Bidimensional com peças do tipo-L

Todos os exemplos de problema de corte mencionados até o momento possuem placa e peças de formato retangular e os padrões de corte guilhotinados. Considere um problema com peças retangulares e irregulares do tipo-L com objetivo de maximizar o valor de utilidade, entretanto com demanda por peças com geometrias diferentes.

Para a geração destes tipos de peças é necessário a utilização do corte de tipo degrau. Este tipo de corte gera uma peça retangular e uma peça do tipo-L ou duas peças do tipo-L, como ilustrado pela Figura 6.

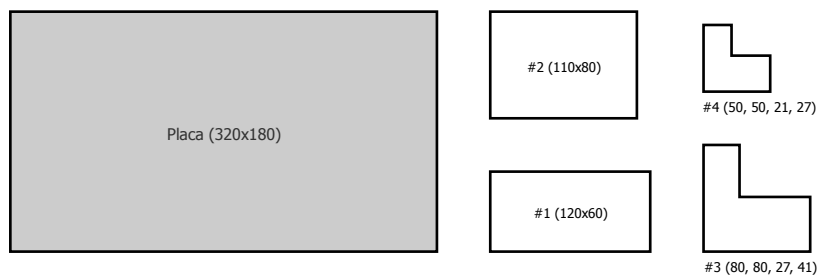
Figura 6 – Tipos de corte degrau.



Fonte: Elaborado pelo autor.

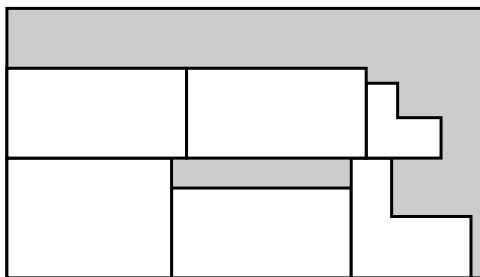
Considere uma placa retangular de dimensões  $L \times W$ , onde  $L$  e  $W$  correspondem ao comprimento e largura da placa, respectivamente. Considere também um conjunto de  $m$  peças de dimensões  $(l_{i1}, w_{i1}, l_{i2}, w_{i2})$ , onde  $l_{i1}$  e  $l_{i2}$  são, respectivamente, o comprimento inferior e superior e  $w_{i1}$  e  $w_{i2}$  correspondem, respectivamente, a largura à esquerda e largura à direita da peça  $i$  (Figura 7). Se a peça  $i$  possuir  $l_{i1} = l_{i2}$  ou  $w_{i1} = w_{i2}$ , a peça é caracterizada por ser retangular, caso contrário a peça é do tipo-L. A Figura 8 ilustra um padrão de corte para o problema descrito.

Figura 7 – Exemplo de problema.



Fonte: Elaborado pelo autor.

Figura 8 – Exemplo de padrão de corte com peças retangulares e do tipo-L.



Fonte: Elaborado pelo autor.

Problemas de corte com peças irregulares são comuns na indústria, entretanto há pouca literatura sobre o tema e métodos para solucioná-los. No próximo capítulo é apresentado uma solução 2-estágios para problemas de corte bidimensional com peças do tipo-L.

## 3 Método Proposto

Há pouca literatura sobre problema de corte com peças irregulares sejam elas do tipo L, U, T ou outro formato. Este trabalho propõe a resolução do problema utilizando método 2-estágios, desenvolvido por [Gilmore e Gomory \(1965\)](#), com a combinação das peças.

As próximas duas seções descrevem o problema da mochila utilizado no método de 2-estágios, e, em seguida, é apresentado o método proposto com mais detalhes.

### 3.1 Problema da Mochila

O Problema da Mochila (em inglês, *Knapsack Problem*) é um problema clássico de otimização combinatória com aplicação em diversas áreas como investimentos, orçamentos, problemas de corte e empacotamento, entre outros. O nome, Problema da Mochila, é dado com base na situação hipotética em que uma pessoa deseja carregar uma mochila com itens de diferentes pesos e valores. O objetivo é carregar a mochila com o maior valor possível sem ultrapassar o peso máximo permitido pela mesma.

O problema pode ser formulado como:

$$\begin{aligned} & \text{maximizar } \sum_{i=1}^m v_i \cdot x_i \\ & \text{sujeito a: } \sum_{i=1}^m w_i \cdot x_i \leq W \\ & x_i \geq 0, \text{ inteiro}, i = 1, \dots, m \end{aligned}$$

em que  $W$  é o peso máximo que a mochila suporta,  $w_i$ ,  $v_i$  são, respectivamente, o peso e o valor do item  $i$  e,  $x_i$  é quantidade de vezes que o item  $i$  foi carregado na mochila.

O Problema da Mochila pode ser estendido para duas dimensões (Problema da Mochila Bidimensional) e, consiste de um conjunto de  $m$  itens retangulares, cada um possuindo valor ( $v_i$ ) e dimensões de comprimento e largura ( $l_i \times w_i$ ), e um retângulo (representando a mochila) de dimensões  $L \times W$ . O objetivo é alocar o maior valor possível de itens no retângulo, desde que os itens não se sobreponham ([CAPRARA; MONACI, 2004](#)) e não ultrapassem as dimensões da mochila.

O problema pode ser descrito através do seguinte modelo matemático como um problema de otimização linear:



$$\begin{aligned}
& \text{maximizar } \sum_{i=1}^m v_i \cdot x_i \\
& \text{sujeito a: } \sum_{i=1}^m \alpha_i \cdot x_i \leq A \\
& x_i \geq 0, \text{ inteiro, } i = 1, \dots, m
\end{aligned}$$

em que  $v_i$ ,  $\alpha_i$  e  $x_i$  representam, respectivamente, o valor, a área ( $l_i \times w_i$ ) e a quantidade de vezes que o item  $i$  é alocado na mochila, e  $A$  representa a área total da mochila ( $L \times W$ ).

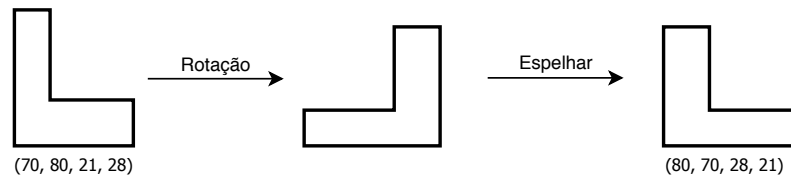
O Problema da Mochila aparece em diversas situações, desde hipotéticas até reais. Uma dessas situações é o problema de corte de itens menores em objetos maiores. O método apresentado a seguir utiliza o Problema da Mochila para auxiliar na obtenção dos resultados.

### 3.2 Resolução do Problema de Corte Bidimensional com peças do tipo-L

Este projeto tem por objetivo apresentar um método de resolução de problemas de corte bidimensional com peças regulares e irregulares do tipo-L. A ideia é aplicar o método 2-estágios de (GILMORE; GOMORY, 1965) combinando peças regulares e irregulares (tipo-L).

É necessário que todas as peças do tipo-L tenham formato tal que  $l_{i1} \geq w_{i1}$ . Caso a peça não atenda esta condição, a mesma deve ser transformada, como mostra a Figura 9:

Figura 9 – Processo de transformação de peças do tipo-L.



Fonte: Elaborado pelo autor.

A transformação possui duas etapas: a primeira é rotacionar a peça em 90 graus de forma que a largura à esquerda se torne o comprimento inferior, e a segunda etapa é espelhar a peça tal que a nova largura à esquerda obtida pela rotação se torna a largura à direita.

Após estes ajustes, inicia-se o método. A seguir é apresentado as etapas para a resolução do problema irrestrito:

#### (Etapa 1) Geração de peças combinadas

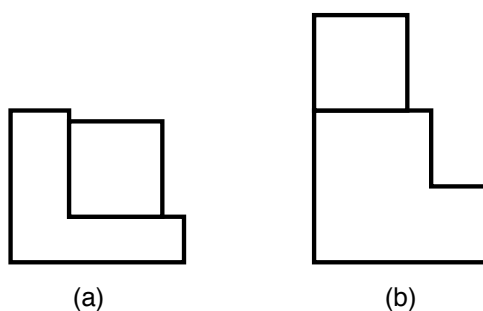
Após o levantamento das principais informações do problema de corte bidimensional: dimensões da placa, quantidade total de peças, demanda e dimensões de cada peça

(regular ou irregular do tipo-L), é realizada a combinação das peças. A combinação ocorre em duas situações:

a) **Peça do tipo-L com peça retangular (combinação LR)**

Para cada peça do tipo-L é verificada a maior região para tentar encaixar as peças retangulares. Esta região pode ser "dentro" da peça L (região entre  $l_1$  e  $l_2$ ) ou no "topo", em cima do comprimento superior  $l_2$ . A Figura 10 ilustra as duas regiões.

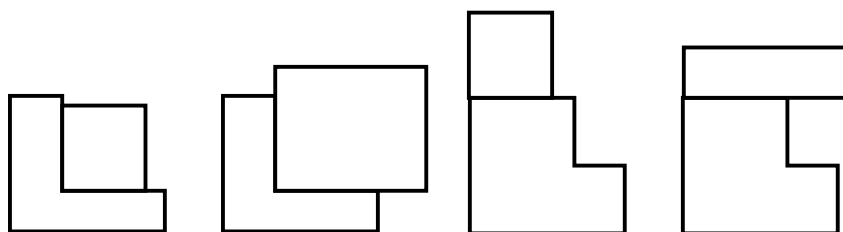
Figura 10 – Combinação de peças L com peças retangulares: (a) combinação na região de "dentro"; (b) combinação no "topo".



Fonte: Elaborado pelo autor.

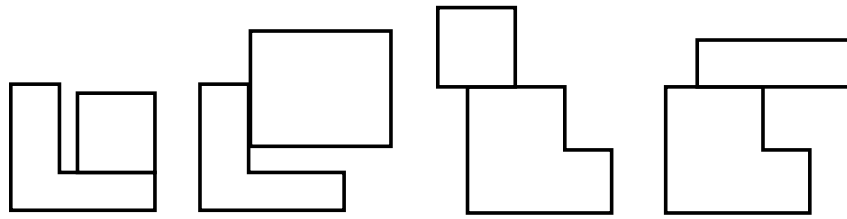
Determinada a região, a peça retangular é encaixada na peça L sempre partindo da largura à esquerda, no caso da combinação pelo "topo", ou sempre partindo pela largura interna, caso a combinação seja por "dentro". É calculado as novas dimensões  $l$  e  $w$  da peça e a área ocupada da mesma (que será a área das duas peças somadas). Desta forma não é necessário salvar em detalhes as dimensões de cada aresta da nova peça. As Figuras 11 e 12 mostram, respectivamente, combinações permitidas e não permitidas.

Figura 11 – Combinações LR permitidas.



Fonte: Elaborado pelo autor.

Figura 12 – Combinações LR não permitidas.



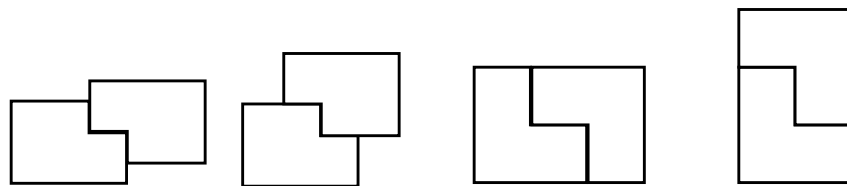
Fonte: Elaborado pelo autor.

Caso a perda da nova peça seja menor ou igual que uma percentagem  $\alpha$  da área da nova peça, a mesma será considerada aceita e adicionada ao conjunto de peças. Caso contrário, a nova peça é descartada.

b) **Peça do tipo-L combinada com ela mesma de forma "espelhada"(combinação LL)**

Para esta combinação também há as regiões de "dentro" e do "topo", entretanto a combinação é realizada para as duas regiões e não há nenhuma condição de aceite da nova peça no conjunto de itens do problema. Este tipo de combinação também segue as regras das peças combinadas não ultrapassarem a largura à esquerda e estarem juntas à peça que serve como base. Na combinação LL, as duas peças são iguais, entretanto a segunda peça é a versão "espelhada" da primeira peça, ou seja, a primeira peça rotacionada em 180 graus. A Figura 13 mostra algumas combinações LL.

Figura 13 – Exemplos de combinação LL.



Fonte: Elaborado pelo autor.

Após todo o processo de geração de combinação de itens, o conjunto de  $m$  itens iniciais aumentará em  $c$  itens, respectivos aos novos itens combinados criados. Portanto, teremos um conjunto de  $n = m + c$  itens.

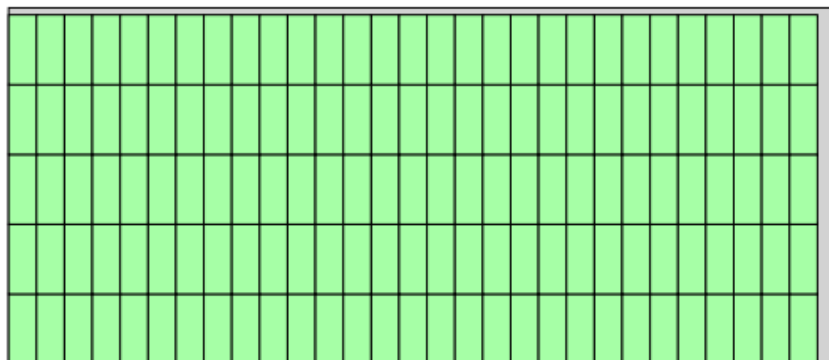
(Etapa 2) **Obtenção da solução inicial**

Nesta etapa tem como objetivo obter uma solução inicial para o problema através de um algoritmo "guloso".

Como o problema é irrestrito, a solução inicial é alocar o maior número de peças do mesmo tipo na placa e determinar o valor de utilidade total, retornando a peça que

teve o melhor valor de utilidade total. Essa solução é chamada de solução homogênea, ilustrado na Figura 14.

Figura 14 – Exemplo de solução homogênea.



Fonte: Elaborado pelo autor.

Determinada a solução inicial, inicia-se a fase da busca pela solução ótima. Esta fase engloba as etapas 3, 4 e 5.

(Etapa 3) **Geração das faixas**

Gerar as faixas para as  $p$  larguras diferentes do conjunto de  $n$  itens, conforme detalhado no Capítulo 2.

(Etapa 4) **Resolução das faixas como Problema da Mochila**

Resolver os diversos Problemas da Mochila conforme mencionado na Seção 2.2.

(Etapa 5) **Resolução do último Problema da Mochila**

Resolver o último Problema da Mochila, otimizando as faixas que serão alocadas na placa, conforme descrito no Capítulo 2. Caso a solução encontrada seja melhor que a solução inicial, retornar a solução encontrada. Se não, retornar solução inicial.

Para o Problema de Corte Restrito, o método é o seguinte:

(Etapa 1) **Geração de peças combinadas**

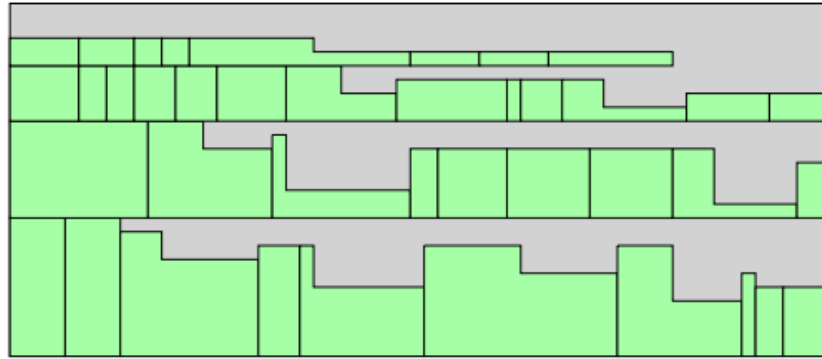
Da mesma forma que o problema irrestrito.

(Etapa 2) **Obtenção da solução inicial**

Nesta etapa tem como objetivo obter uma solução inicial para o problema através de um algoritmo "guloso". A solução inicial será do tipo homogênea composta para o Problema Restrito. Como as demandas devem ser atendidas, o algoritmo alocará, em faixas, o maior número de peças 1 até atender a demanda. Logo depois, para a segunda peça repetirá o processo até alocar todas as peças. A alocação poderá ser em uma nova faixa

ou numa faixa já preenchida e que possui espaço suficiente para a alocação da peça. A alocação das peças é feita pela ordem das peças de maior valor. A Figura 15 ilustra uma solução homogênea composta.

Figura 15 – Exemplo de solução inicial homogênea composta.



Fonte: Elaborado pelo autor.

Determinada a solução inicial, inicia-se a fase da busca pela solução ótima. Esta fase engloba as etapas 3, 4 e 5.

(Etapa 3) **Geração das faixas**

Gerar as faixas para as  $p$  larguras distintas do conjunto de  $n$  itens caso ainda haja espaço na placa.

(Etapa 4) **Resolução das faixas como Problema da Mochila**

Resolver as faixas como Problema da Mochila, conforme modelo matemático da problema irrestrito, e adicionando três tipos de restrições:

a) Restrição de demanda

$$x_i \leq b_i, i = 1, \dots, n$$

em que  $x_i$  e  $b_i$  é o número de vezes que a peça de item  $i$  é produzida e a demanda da mesma.

b) Restrição de combinação LR

Seja a peça 1 e 2, respectivamente, peças do tipo L e R, e a peça 3 a combinação das duas, a restrição de combinação será:

$$x_1 + x_2 + 2 \cdot x_3 \leq b$$

em que  $x_1$ ,  $x_2$  e  $x_3$  representam a quantidade de vezes que as peças 1, 2 e 3 são produzidas.  $b$  representa a menor demanda entre as peças 1 e 2.

c) Restrição de combinação LL

Seja a peça 1 uma peça do tipo L e a peça 2 a combinação LL da peça 1, a restrição de combinação será:

$$x_1 + 2 \cdot x_2 \leq b_1$$

em que  $x_1$  e  $x_2$  representam a quantidade de vezes que as peças 1 e 2 foram produzidas, e  $b_1$  representa a demanda da peça 1, que será a demanda da peça 2.

O modelo matemático do problema restrito não comporta totalmente as demandas, o que cria a necessidade de torná-lo um problema de otimização não linear. Para evitar isso, foi definido o seguinte procedimento:

- i) Gerar as faixas como problemas de otimização linear, conforme modelo do Problema da Mochila;
- ii) Resolver todos os problemas até o primeiro que não tenha solução viável;
- iii) Selecionar, dentre as faixas com solução viável, a faixa com melhor valor de utilidade. Caso haja mais de uma faixa com o mesmo resultado, verificar a faixa que possui o maior número de peças alocadas. Caso haja mais de uma faixa com o mesmo número de peças alocadas, retornar qualquer faixa;
- iv) Atualizar a demanda das peças;
- v) Repetir os passos **i** a **iv** até que todas as demandas sejam atendidas ou não ser possível gerar mais faixas por falta de espaço na placa.

(Etapa 5) **Resolução do problema**

Terminada a etapa anterior, retornar a solução encontrada, caso a mesma seja melhor que a solução inicial. Se não, retornar a solução inicial.

## 4 Implementação Computacional

Este capítulo apresenta as ferramentas e estruturação do programa para a geração dos padrões de corte.

### 4.1 Ferramentas utilizadas

Para a implementação foram utilizadas três ferramentas: a Linguagem *Python*, o pacote de otimização CPLEX e o pacote do TeXLive com o pacote TikZ.

#### 4.1.1 Linguagem *Python*

A linguagem de programação *Python* é uma das linguagens mais utilizadas e com grande suporte da comunidade de desenvolvedores oferecendo uma vasta gama de pacotes para, praticamente, qualquer coisa. Além disso, sua curva de aprendizagem é muito baixa, suporta diversos paradigmas de programação (i.e., estruturada e orientada a objetos), e tem uma sintaxe muito próxima a de pseudo-algoritmos. Todas essas vantagens mostraram de grande relevância para a escolha desta ferramenta para o projeto. Para o desenvolvimento do gerador foi utilizada a versão 3.7.3.

#### 4.1.2 CPLEX

O pacote CPLEX é um produto proprietário da IBM que oferece recursos para se desenvolver aplicações que necessitem da resolução de problemas de otimização linear, linear inteiro, quadrático, entre outras variações. Foi criado em 1988 por Robert E. Bixby como o Método Simplex implementado na Linguagem C ([BIXBY, 2012](#)), daí a origem do nome do programa, e em 2009 foi adquirido pela IBM através da empresa ILOG ([IBM-NEWSROOM, 2009](#)).

Através dos anos, além de continuar o suporte a linguagem C, foi criada APIs (*Application Programming Interface* em inglês, traduzida como Interface de Programação de Aplicação) para as linguagens Java, C++, C# (C-Sharp), Python, entre outras.

Devido a API para Python e da robustez do programa, o mesmo foi escolhido para resolver os problemas da mochila criados pelo gerador durante a execução do método. Para este trabalho foi utilizada a versão 12.9 utilizando uma licença para estudante fornecida pela própria IBM através do seu programa AI (sigla para *Academic Initiative*).

### 4.1.3 TeXLive e TikZ

Para o módulo de desenho do gerador de padrões foi necessário a utilização da distribuição TeXLive. Esta distribuição instala um ambiente completo do sistema LaTeX, conhecida plataforma de preparação de documentos por texto simples e comandos de formatação. A escolha pela distribuição TeXLive se deu pela facilidade de instalação em qualquer sistema operacional (S.O.) e pela presença do pacote TikZ e do *script* pdflatex.

O pacote TikZ é uma biblioteca que permite produzir gráficos e desenhos utilizando descrição geométrica/algébrica através de comandos como:

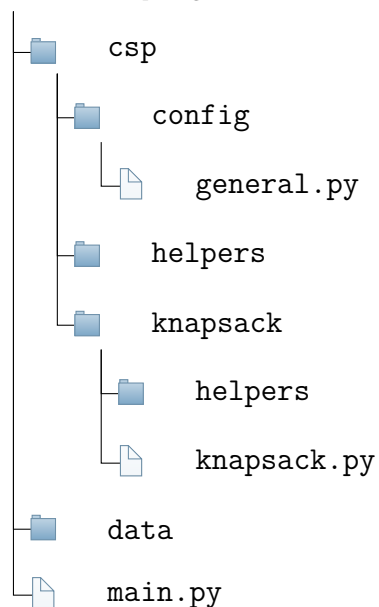
```
\tikz \draw (0,0) rectangle (3,1);
\tikz \draw (0,0) circle (0.5);
```

É possível desenhar, por exemplo, retângulos e círculos de forma simples e rápida. No primeiro, descreve-se o retângulo apontando o ponto inicial e final e entre eles a palavra-chave *rectangle*. Para o círculo, a sintaxe é a mesma, mas o último comando é o raio que círculo possui.

## 4.2 Estrutura do programa

A estrutura do programa é organizada da seguinte forma:

raíz do projeto



A pasta data armazena alguns exemplos de problemas de corte para testar o gerador executando o arquivo `main.py` no terminal. A execução no terminal é bem simples e recebe



parâmetros como o caminho do arquivo de teste no sistema, se é para rotacionar ou não, gerar o padrão de corte desenhado, se o problema possui restrição.

A pasta `csp` é onde todo o código do gerador está armazenado e é chamado pelo arquivo `main.py`. Cada subdiretório da pasta `csp` possui um arquivo `__init__.py` para torná-los módulos do Python. Isto permite que cada pasta possa ser chamada por outros arquivos em lugares diferentes, facilitando a modularização. Dentro da pasta `csp` há três módulos: `config`, `helpers` e `knapsack`.

O módulo `config` tem responsabilidade de armazenar, através do arquivo `general.py`, as variáveis globais de todo o programa. Neste módulo que é declarada a estrutura de dados para as informações da placa e das peças. As informações da placa são armazenadas numa variável chamada `plate` que terá os atributos `L` e `W`. As informações das peças são armazenadas através da instância de uma classe de dados (*dataclass* no Python) chamada `Piece`.

A classe `Piece` tem como atributos a identificação, tipo (retangular, irregular ou combinada), dimensões (diferente para cada tipo de peça), área, demanda e perda de cada peça. Além disso, há atributos para verificar se a peça foi rotacionada (somente para peças retangulares), transformada (somente para peças irregulares, e indica se foi aplicada o processo de transformação relatada no capítulo 2.2.2) e é fruto de uma combinação, que conterá informações sobre as peças originais e o tipo de combinação (combinação LR ou LL). Há duas funções que realizam o cálculo da área e perda, conforme tipo da peça.

O módulo `helpers` é responsável por facilitar operações comuns a todo o programa, como desenho de padrões de corte, leitura dos arquivos e determinação da solução inicial para problemas irrestritos e restritos.

Finalmente, o último módulo, `knapsack`, é responsável por implementar o método proposto conforme descrição no capítulo 2.2.2. A forma como este módulo foi construído é de emular uma API que, após a declaração do problema de corte a ser solucionado, basta instanciar a classe correspondente ao tipo de problema a ser tratado (restrito ou irrestrito) e chamar as funções `solve` e `get_solution`.

Neste módulo, as classes para os dois tipos de problema de corte (restrito ou irrestrito) estão implementadas no arquivo `knapsack.py`. A pasta `helpers` interna à pasta do módulo `knapsack` é responsável por criar a base dos problemas da mochila. Esta base tem como objetivo implementar os problemas da mochila criadas pelas classes do arquivo `knapsack.py` e usar a API para Python do CPLEX e retornar a solução encontrada.

## 5 Resultados

Para avaliar o desempenho do método descrito na seção 3.2, foi utilizado instâncias da literatura. Apesar de algumas instâncias terem solução ótima obtida por meio de métodos diferentes, os resultados obtidos mostram que o método consegue encontrar um resultado muito próximo do ótimo, quando não o encontram.

As instâncias foram separadas em dois grupos: problemas de corte com peças retangulares e problemas de corte com peças retangulares e do tipo-L. Todos os problemas tem como solução ótima a perda encontrada para o problema na forma irrestrita.

Para o primeiro grupo foi selecionado 13 problemas da *OR-Library* (*gcut1*, ..., *gcut13*) e que foram analisados por [Beasley \(1985\)](#), [Cintra et al. \(2008\)](#) e [Yanasse e Morabito \(2013\)](#). A Tabela 1 apresenta, para cada instância, a solução ótima fornecida pela literatura e a solução do método implementado.

Tabela 1 – Resultados para problemas do grupo 1.

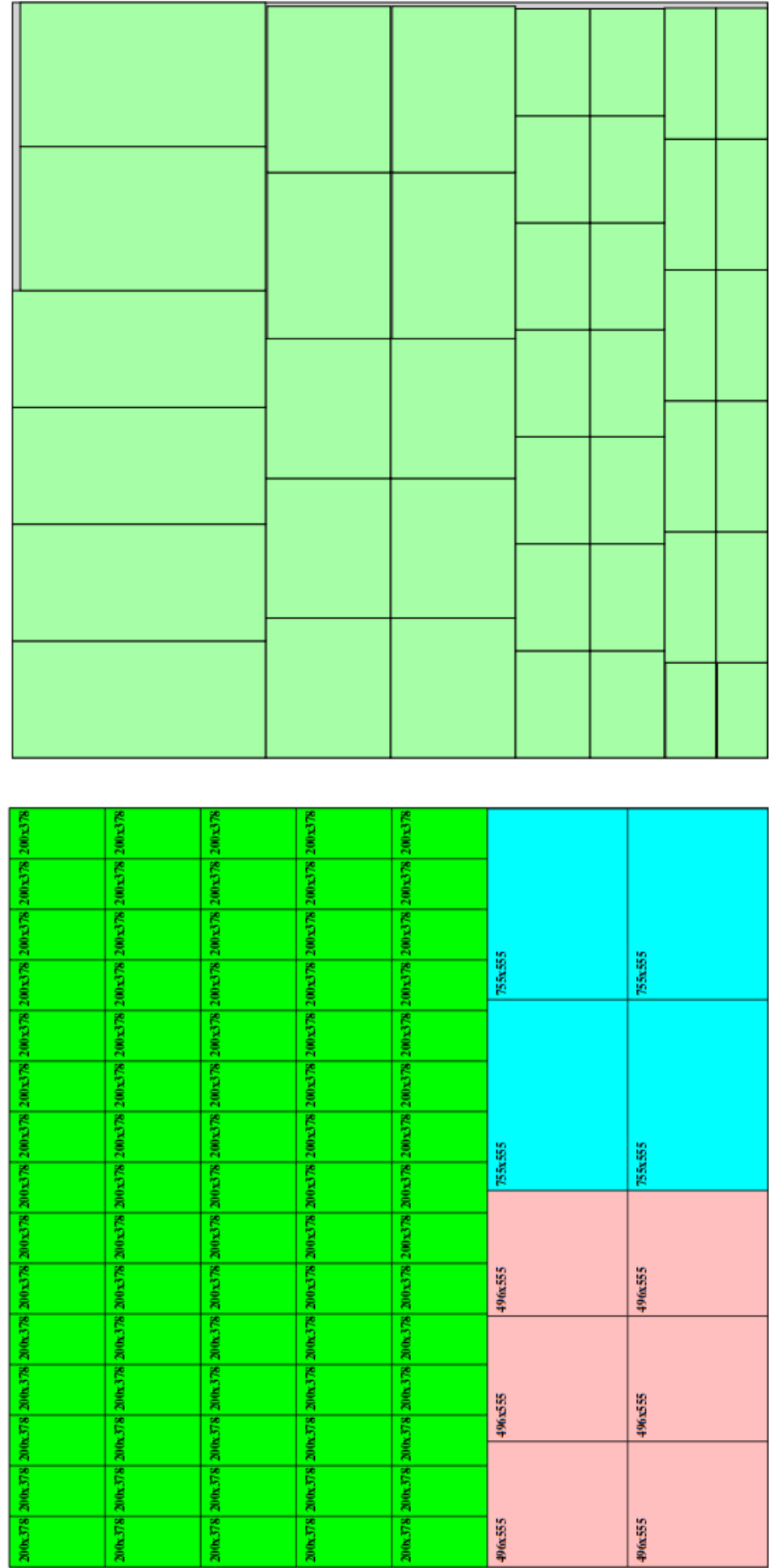
#	Instância	Dimensão da Placa	Número de itens	Solução Ótima	Solução do método
1	<i>gcut1</i>	250 x 250	10	9,664%	9,664%
2	<i>gcut2</i>		20	3,142%	3,878%
3	<i>gcut3</i>		30	2,342%	3,787%
4	<i>gcut4</i>		50	1,283%	1,283%
5	<i>gcut5</i>	500 x 500	10	1,600%	1,600%
6	<i>gcut6</i>		20	4,401%	5,997%
7	<i>gcut7</i>		30	2,973%	2,973%
8	<i>gcut8</i>		50	1,347%	1,697%
9	<i>gcut9</i>	1000 x 1000	10	2,890%	2,890%
10	<i>gcut10</i>		20	1,798%	1,797%
11	<i>gcut11</i>		30	1,990%	2,536%
12	<i>gcut12</i>		50	2,001%	2,223%
13	<i>gcut13</i>	3000 x 3000	32	0,025%	1,042%

Fonte: Elaborado pelo autor.

Nota-se que dos 13 problemas, o método determinou a solução ótima das instâncias #1, #4, #5, #7 e #9. Vale destacar que para o problema *gcut10*, o resultado foi o mesmo também, apesar da diferença de 0,001%. Além disso, para as outras instâncias foi possível obter um resultado bem próximo do ótimo (instâncias #2, #8 e #12).

Na Figura 16 é ilustrado o padrão de corte da solução ótima ([CINTRA et al., 2008](#)) e o padrão de corte obtido pelo método proposto para o problema *gcut13*, que tiveram, conforme comentado anteriormente, uma diferença de 1% de perda entre as soluções.

Figura 16 – Padrão de corte para a instância *gcut13*.



(a) padrão de corte ótimo

(b) padrão de corte encontrado pelo método

Fonte: (a) Elaborado pelo autor, (b) Cintra et al. (2008)

Para o grupo 2 foi selecionado dois problemas. As instâncias #1 e #2 foram analisadas por Nakatake et al. (1996), Xu, Guo e Cheng (1998) e Cherri e Vianna (2012).

As Tabelas 2 e 3 apresentam, respectivamente, as informações sobre cada instância do grupo 2 e as soluções para os casos irrestrito e restrito.

Tabela 2 – Dados para os problemas do grupo 2.

#	Instância	Dimensão da Placa	Número de itens	Número de itens retangulares	Número de itens do tipo-L
1	Nakatake	594 x 255	35	25	10
2	Xu	6314 x 5922	28	7	21

Fonte: Elaborado pelo autor.

Tabela 3 – Resultados para o grupo 2.

#	Instância	Solução ótima	Solução do método (irrestrito)	Solução do método (restrito)
1	Nakatake	13,000%	2,621%	38,932%
2	Xu	5,200%	0,716%	31,339%

Fonte: Elaborado pelo autor.

Nota-se que método teve ótimos resultados para todas as instâncias no caso irrestrito, obtendo perda próxima de zero para os problemas #2, #3 e #4. Outro destaque é para solução da instância #1 que teve uma diminuição da perda de 13% para menos de 3%.

Para o caso restrito, as soluções não se mostraram muito boas. Isso se deve ao fato destes problemas apresentarem peças com demanda muito baixa, tornando a perda automaticamente grande. Outro problema é que a demanda baixa impede o uso das peças combinadas, já que as mesmas representam as peças originais duas vezes no modelo matemático. Isto acarreta que, caso a demanda de uma peça  $k$  qualquer seja 1, é impossível aplicar uma peça combinada da peça  $k$  sem violar a demanda da mesma.

As Tabelas 4 e 5 ilustra as demandas e dimensões de cada peça para as instâncias #1 e #2.

Tabela 4 – Dados para a instância *Nakatake*.

Peças retangulares				Demanda
50		40		1
20		40		1
40		100		2
10		60		1
20		50		1
30		80		1
20		40		2
30		50		1
80		30		1
30		40		2
10		30		1
30		30		1
20		50		1
60		20		1
50		20		1
50		40		1
50		50		1
40		20		1
60		50		1
100		70		1
40		20		1
50		10		2
20		20		2
90		10		1
60		50		1
Peças do tipo-L				Demanda
80	90	50	10	1
90	50	30	10	1
160	20	90	10	1
60	100	20	10	1
90	30	30	10	1
40	80	20	40	1
140	80	70	60	1
90	80	40	40	1
100	90	30	70	1
70	90	50	40	1

Fonte: Elaborado pelo autor.

Tabela 5 – Dados para a instância  $Xu$ .

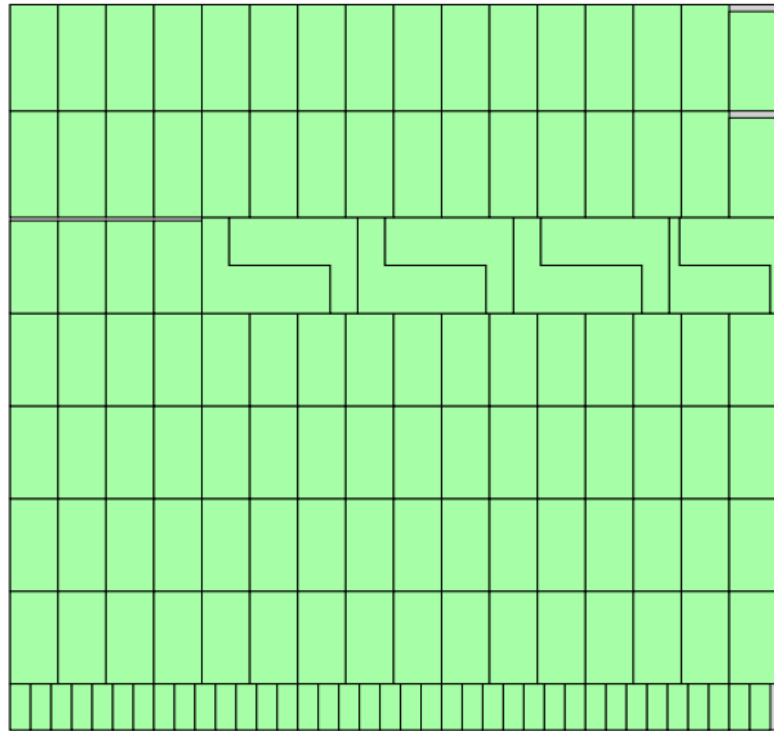
Peças retangulares				Demanda
1708		3234		1
1862		882		1
392		756		1
798		266		1
392		868		1
168		378		1
434		812		1
Peças do tipo-L				Demanda
1302	1554	672	1302	1
3080	2618	2184	1008	1
1386	1260	84	532	1
868	756	28	350	1
910	1778	266	532	1
1064	980	980	518	1
826	672	616	378	1
672	1064	406	266	1
924	798	854	406	1
854	686	798	322	1
1862	392	784	145	1
1050	784	224	392	1
1582	364	784	252	1
826	784	84	392	1
952	728	840	364	1
2128	490	1134	448	1
784	672	728	350	1
1148	1050	994	490	1
1890	1680	1302	728	1
1722	1218	560	672	1
1204	952	238	448	1

Fonte: Elaborado pelo autor.

Mesmo com estes resultados para os problemas restritos, é perceptível que a combinação das peças irregulares com ela mesma e com peças retangulares ajudou a obter soluções com bom resultado para os problemas irrestritos. Para problemas restritos onde a demanda das peças é maior, o método terá mais recursos para gerar um padrão de corte mais otimizado utilizando-se das peças combinadas.

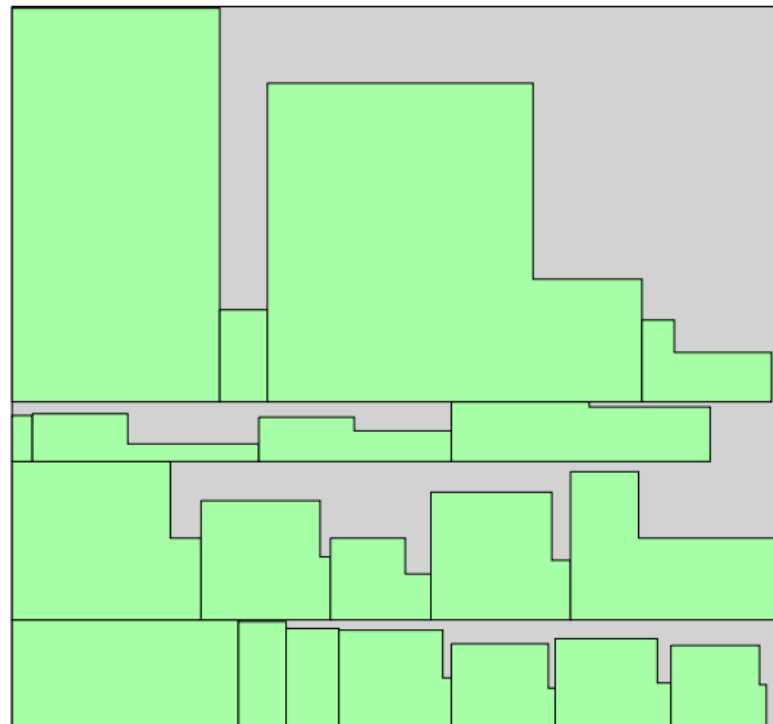
Na Figura 17 e 18 é mostrado, respectivamente, o padrão de corte irrestrito e restrito da instância #2 do grupo 2.

Figura 17 – Padrão de corte para a instância  $Xu$  (irrestrito).



Fonte: Elaborado pelo autor.

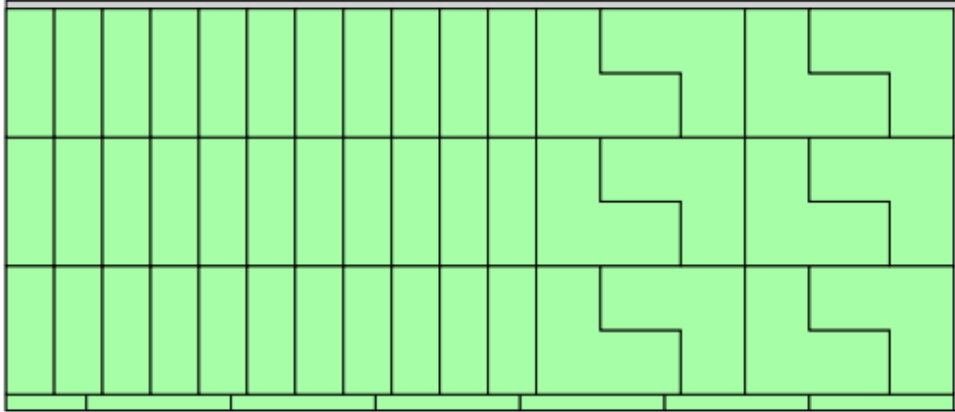
Figura 18 – Padrão de corte para a instância  $Xu$  (restrito).



Fonte: Elaborado pelo autor.

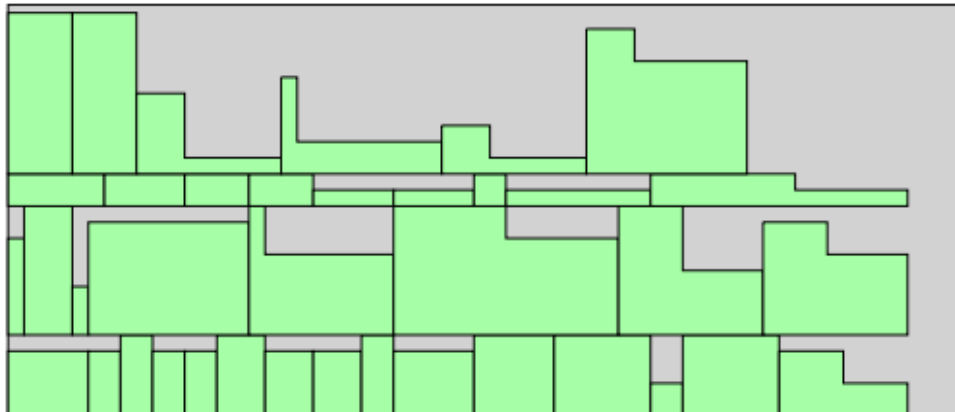
Os padrões de corte irrestrito e restrito para a instância #1 do grupo 2 estão ilustrados nas Figuras 19 e 20.

Figura 19 – Padrão de corte para a instância *Nakatake* (irrestrito).



Fonte: Elaborado pelo autor.

Figura 20 – Padrão de corte para a instância *Nakatake* (restrito).



Fonte: Elaborado pelo autor.



## 6 Conclusão

Neste trabalho foi abordado o Problema de Corte Bidimensional com especial foco nos problemas com itens regulares e irregulares do tipo-L. O objetivo foi apresentar um método 2-estágios com combinação de peças regulares e irregulares (aumentando o número de peças), de modo que a tentar diminuir ou eliminar a perda das peças L, melhorando a solução final.

O método apresentou bons resultados tanto para problemas somente com peças retangulares como em problemas com os dois tipos de peça. Vale destacar os ótimos resultados obtidos no segundo grupo de instâncias, onde o método foi capaz de gerar padrões de corte com perda próxima a zero.

A combinação de peças realmente ajudou o algoritmo de corte a encontrar uma solução melhor, em decorrência do aumento de variabilidade de itens disponíveis e que, se tratando das peças combinadas, se equivalem à peças originais. Como trabalho futuro, pode-se tentar aplicar a combinação em método como o grafo E/OU, algoritmos genéticos, entre outros.

## Referências

- ARENALES, M.; MORABITO, R.; YANASSE, H. Special issue: Cutting and packing problems. *Pesquisa Operacional*, v. 19, n. 2, p. 107–299, 1999.
- BEASLEY, J. E. Algorithms for unconstrained two-dimensional guillotine cutting. *Journal of the Operational Research Society*, Taylor & Francis, v. 36, n. 4, p. 297–306, 1985. Disponível em: <<https://doi.org/10.1057/jors.1985.51>>.
- BISCHOFF, E.; WÄSCHER, G. Cutting and packing. *European Journal of Operational Research*, v. 84, n. 3, p. 503 – 505, 1995. ISSN 0377-2217. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S037722179500018L>>.
- BIXBY, R. E. A brief history of linear and mixed-integer programming computation. *Documenta Mathematica*, 2012.
- BROWN, A. R. A. R. Book; Book/Illustrated. *Optimum packing and depletion : the computer in space - and resource-usage problems*. [S.l.]: London : Macdonald and Co. ; New York : American Elsevier, 1971. Bibliography: p. 106. ISBN 0444195882.
- CAPRARA, A.; MONACI, M. On the two-dimensional knapsack problem. *Operations Research Letters*, v. 32, n. 1, p. 5 – 14, 2004. ISSN 0167-6377. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S0167637703000579>>.
- CHERRI, A. C.; VIANNA, A. C. G. Problemas de corte com itens irregulares. In: XVI CLAIO/ XLIV SBPO - Congresso Latino-Iberoamericano de Investigación Operativa / Simpósio Brasileiro de Pesquisa Operacional. [S.l.: s.n.], 2012. (Anais do CLAIO/SBPO), p. 3327–3335.
- CINTRA, G. et al. Algorithms for two-dimensional cutting stock and strip packing problems using dynamic programming and column generation. *European Journal of Operational Research*, v. 191, n. 1, p. 61 – 85, 2008. ISSN 0377-2217. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S0377221707008831>>.
- DYCKHOFF, H. A typology of cutting and packing problems. *European Journal of Operational Research*, Elsevier, v. 44, n. 2, p. 145–159, 1990.
- DYCKHOFF, H.; FINKE, U. *Cutting and packing in production and distribution: A typology and bibliography*. [S.l.]: Springer Science & Business Media, 1992.
- GILMORE, P. C.; GOMORY, R. E. Multistage cutting stock problems of two and more dimensions. *Oper. Res.*, INFORMS, Institute for Operations Research and the Management Sciences (INFORMS), Linthicum, Maryland, USA, v. 13, n. 1, p. 94–120, fev. 1965. ISSN 0030-364X. Disponível em: <<http://dx.doi.org/10.1287/opre.13.1.94>>.
- GOLDEN, B. L. Approaches to the cutting stock problem. *AIIE transactions*, Taylor & Francis, v. 8, n. 2, p. 265–274, 1976.
- IBM-NEWSROOM. *IBM Completes Acquisition of ILOG*. 2009. Disponível em: <<https://web.archive.org/web/20120927094255/https://www-304.ibm.com/jct03002c/press/us/en/pressrelease/26403.wss>>. Acesso em: 21 out. 2019.

KHAN ACADEMY. *Divide and conquer algorithms*. Disponível em: <<https://www.khanacademy.org/computing/computer-science/algorithms/merge-sort/a/divide-and-conquer-algorithms>>. Acesso em: 19 out. 2019.

LINS, L.; LINS, S.; MORABITO, R. An l-approach for packing (l, w)-rectangles into rectangular and l-shaped pieces. *Journal of the Operational Research Society*, v. 54, 2003.

NAKATAKE, S. et al. Module placement on bsg-structure and ic layout applications. In: *Proceedings of the 1996 IEEE/ACM International Conference on Computer-aided Design*. Washington, DC, USA: IEEE Computer Society, 1996. (ICCAD '96), p. 484–491. ISBN 0-8186-7597-7. Disponível em: <<http://dl.acm.org/citation.cfm?id=244522.244865>>.

TUTORIALSPPOINT. *Data Structures - Divide and Conquer*. Disponível em: <[https://www.tutorialspoint.com/data\\_structures\\_algorithms/divide\\_and\\_conquer.htm](https://www.tutorialspoint.com/data_structures_algorithms/divide_and_conquer.htm)>. Acesso em: 19 out. 2019.

VIANNA, A. C. G. *Problemas de corte e empacotamento: uma abordagem em grafo E/OU*. Tese (Doutorado) — Universidade de São Paulo, 2000.

VIANNA, A. C. G.; ARENALES, M. N.; GRAMANI, M. C. N. Two-stage and constrained two-dimensional guillotine cutting problems. ICMC-USP, 2003.

XU, J.; GUO, P.-n.; CHENG, C.-K. Rectilinear block placement using sequence-pair. In: *Proceedings of the 1998 International Symposium on Physical Design*. New York, NY, USA: ACM, 1998. (ISPD '98), p. 173–178. ISBN 1-58113-021-X. Disponível em: <<http://doi.acm.org/10.1145/274535.274561>>.

YANASSE, H. H.; MORABITO, R. Modelos lineares e não lineares inteiros para problemas da mochila bidimensional restrita a 2 estágios. *Production*, scielo, v. 23, p. 887 – 896, 12 2013. ISSN 0103-6513. Disponível em: <[http://www.scielo.br/scielo.php?script=sci\\_arttext&pid=S0103-65132013000400018&nrm=iso](http://www.scielo.br/scielo.php?script=sci_arttext&pid=S0103-65132013000400018&nrm=iso)>.