

LUCAS COFFONI MORRONE

**OTIMIZAÇÃO DOS PLANOS DE IMPRESSÃO E
CORTE EM UMA GRÁFICA**

São Paulo
2023

LUCAS COFFONI MORRONE

**OTIMIZAÇÃO DOS PLANOS DE IMPRESSÃO E
CORTE EM UMA GRÁFICA**

Trabalho apresentado à Escola Politécnica
da Universidade de São Paulo para obtenção
do Título de Engenheiro de Produção.

São Paulo
2023

LUCAS COFFONI MORRONE

**OTIMIZAÇÃO DOS PLANOS DE IMPRESSÃO E
CORTE EM UMA GRÁFICA**

Trabalho apresentado à Escola Politécnica
da Universidade de São Paulo para obtenção
do Título de Engenheiro de Produção.

Orientador:

Leonardo Junqueira

São Paulo
2023

Aos meus pais.

AGRADECIMENTOS

Agradeço aos meus pais, Stiven e Perlla, por serem o melhor exemplo que eu poderia ter. Muito obrigado por serem a minha fonte de inspiração diária, tenho muito orgulho e admiração por vocês.

Agradeço à Marina por todo o seu amor, dedicação e apoio ao longo dos últimos anos. Sem você, essa jornada teria sido muito mais difícil. Muito obrigado por estar sempre ao meu lado.

Agradeço aos meus amigos por todas as conversas, aprendizados e discussões ao longo dos últimos anos na Escola Politécnica. Com certeza a minha vida acadêmica teria sido muito pior sem vocês.

Finalmente, agradeço ao meu orientador Leonardo Junqueira pela sua paciência, atenção e ensinamentos, que foram essenciais para a conclusão deste trabalho.

*“Once you know what failure feels like,
determination chases success.”*

-- Kobe Bryant

RESUMO

Este trabalho tem como objetivo aplicar os conceitos dos Problemas de Corte e Empacotamento e de Pesquisa Operacional para otimizar o processo de montagem do Plano de Impressão e Corte em uma pequena gráfica familiar localizada na cidade de São Paulo. A empresa não possui tecnologia para auxiliá-la na montagem do Plano, o que torna a tarefa complexa devido à ampla gama de papéis disponíveis, grande volume de pedidos únicos e prazo curto de execução. O trabalho visa explorar modelos para Problemas de Corte Bidimensional Guilhotinado já desenvolvidos em trabalhos publicados e estendê-los para a realidade da empresa, implementando duas inovações em relação aos modelos originais: a possibilidade de rotação dos itens e um critério de desempate das soluções ótimas visando maior eficiência operacional e redução do desperdício de papel. O principal objetivo do trabalho é propor uma solução capaz de gerar os Planos de Impressão e Corte de forma automática, priorizando a solução com o menor custo. A linguagem de programação usada para implementar os modelos matemáticos é o Python, com o Gurobi sendo o resolvidor de propósito geral. Por fim, realiza-se uma análise comparativa entre soluções adotadas pela gráfica na prática com as soluções encontradas pelos modelos, evidenciando as oportunidades de melhorias da operação atual e fornecendo recomendações práticas para a empresa.

Palavras-Chave – Pesquisa Operacional, Corte e Empacotamento, Corte Guilhotinado, Plano de Impressão e Corte.

ABSTRACT

This work aims to apply the concepts of Cutting and Packing Problems and Operations Research to optimize the process of assembling the Printing and Cutting Plan in a small family-owned printing company located in São Paulo, Brazil. The company lacks technology to assist it in the plan assembly, making the task complex due to the wide range of available paper types, the high volume of unique orders, and the tight deadlines. The goal of this work is to explore existing two-dimensional guillotine cutting problem models from published research and extend them to the company's reality by implementing two innovations compared to the original models: the possibility of item rotation and a tie-breaking criterion for optimal solutions to enhance operational efficiency and reduce paper waste. The main objective of the work is to propose a solution capable of generating Printing and Cutting Plans automatically, prioritizing the solution with the lowest cost. Python is the programming language used to implement the mathematical models, with Gurobi as the general-purpose solver. Finally, a comparative analysis is carried out between the solutions adopted by the printing company in practice and the solutions provided by the models, highlighting opportunities for improving the current operation and providing practical recommendations for the company.

Keywords – Operations Research, Cutting and Packing, Guillotine Cutting, Printing and Cutting Plan.

LISTA DE FIGURAS

1	Plano de Impressão e Corte.	13
2	Estrutura do Capítulo 2.	17
3	Estruturação da tipologia de WÄSCHER et al. (2007).	22
4	Tipos básicos de Problemas de Corte e Empacotamento.	24
5	Problemas de maximização das saídas.	25
6	Problemas de minimização das entradas.	26
7	Representação de objeto, item e sobra.	28
8	Empacotamento de itens irregulares.	28
9	Empacotamento não-ortogonal.	29
10	Padrões guilhotinado e não-guilhotinado.	30
11	Estágios de corte.	30
12	Padrões exato e não-exato.	31
13	Padrões orientado e não-orientado.	31
14	Representação de níveis em um objeto.	32
15	Cortes guilhotinados em dois estágios, exatos e não-exatos.	34
16	Principais tipos de problemas com itens distintos.	34
17	Empacotamentos normalizado e não-normalizado.	39
18	Três estratégias clássicas (NFDH, FFDH e BFDH).	41
19	Algoritmo HFF.	43
20	Algoritmo FFF.	44
21	Estrutura do Capítulo 3.	45
22	Impressora utilizada pela gráfica.	47
23	Gaveta 1 da impressora.	48

24	Formatos A a partir da folha A0.	49
25	Formatos a partir da matriz S.	50
26	Cálculo dos cliques para os formatos mais utilizados.	52
27	Exemplo de como a montagem do Plano influencia na quantidade de folhas utilizadas.	54
28	Guilhotina elétrica utilizada pela empresa.	55
29	Estrutura do Capítulo 4.	58
30	Representação dos índices de níveis abertos.	61
31	Representação dos cenários sem e com critério de desempate.	66
32	Estrutura do Capítulo 5.	69
33	Exemplo de solução da instância 3 pelo modelo R^D	76
34	Comparação das soluções da instância 5.	77
35	Comparação do tempo de execução para os quatro modelos.	78
36	Comparação da quantidade de nós para os quatro modelos.	78
37	Comparação da quantidade de iterações para os quatro modelos.	79
38	Exemplo de instância <i>infeasible</i>	81
39	Comparação das soluções encontradas pelo modelo com as soluções adotadas na prática.	81
40	Comparação entre os modelos sem rotação e com rotação.	82
41	Comparação da instância 14 entre os modelos com rotação e sem rotação.	83
42	Comparação da quantidade total de níveis abertos nos modelos.	85
43	Comparação da quantidade de níveis abertos na instância 4, com e sem critério de desempate.	86
44	Comparação da quantidade total de objetos utilizados.	87
45	Comparação da quantidade de objetos na instância 17.	88

LISTA DE TABELAS

1	Comparação entre impressão <i>offset</i> e digital.	46
2	Formatos das folhas mais utilizadas pela empresa.	50
3	Custos de matéria-prima e de impressão para os Formatos mais utilizados.	51
4	Notação dos modelos matemáticos	62
5	Instâncias utilizadas nos testes computacionais.	70
6	Objetos escolhidos pela gráfica para solucionar as instâncias.	71
7	Siglas dos 4 modelos.	72
8	Soluções do Modelo S	74
9	Soluções do Modelo S^D	74
10	Soluções do Modelo R	75
11	Soluções do Modelo R^D	75
12	Comparação dos custos totais entre a solução da gráfica e as soluções dos modelos.	80
13	Quantidade de níveis nos 4 modelos.	84

SUMÁRIO

1	Introdução	12
1.1	Motivações, Objetivos e Contribuições	14
1.2	Estrutura do Trabalho	15
2	Fundamentação Teórica	17
2.1	Pesquisa Operacional	18
2.2	Problemas de Corte e Empacotamento	19
2.2.1	Critérios da Tipologia de WÄSCHER et al. (2007)	22
2.2.2	Tipos de Problemas de Corte e Empacotamento	24
2.3	Problemas de Corte Bidimensional	27
2.3.1	Conceitos Relevantes	28
2.4	Problemas de Corte Bidimensional Guilhotinado	33
2.4.1	Histórico de Abordagens	35
2.4.2	Estratégias e Algoritmos	40
3	Descrição do Problema	45
3.1	Apresentação do Problema	46
3.2	Definição do Problema	53
4	Modelagem do Problema	58
4.1	Definições e Notação	59
4.2	Modelos sem Critério de Desempate	63
4.2.1	Modelo S - Sem Rotação	63
4.2.2	Modelo R - Com Rotação	64
4.3	Modelos com Critério de Desempate	66

4.3.1	Critério de Desempate	66
4.3.2	Modelos S^D e R^D	67
5	Testes Computacionais e Análises	69
5.1	Instâncias	70
5.2	Resultados	72
5.3	Análises	77
5.3.1	Esforço Computacional	77
5.3.2	Custos Totais	79
5.3.3	Critério de Desempate	84
5.3.4	Principais <i>Insights</i> e Recomendações Práticas	86
6	Conclusões e Perspectivas Futuras	89
	Referências	91

1 INTRODUÇÃO

A indústria de impressão tem passado por diversas transformações nos últimos anos. As tecnologias presentes no setor evoluíram de forma significativa devido à crescente demanda dos clientes por produtos personalizados e de alta qualidade, representando a criação de novas máquinas, processos e opções para o consumidor. Outra característica observada é que, com o avanço das plataformas digitais, a indústria tem notado uma mudança nos padrões dos pedidos, com redução no volume de impressões e prazos de entrega mais curtos, além da maior variedade de materiais disponíveis (MOSTAJABDAVEH et al., 2022).

Em uma indústria que é fortemente dependente da experiência do cliente, atender às suas necessidades no tempo desejado é um fator fundamental para garantir a fidelização da base de consumidores. Por isso, essas transformações impactam diretamente na tomada de decisões das empresas, especialmente no planejamento e na execução dos pedidos. Do ponto de vista das empresas, as escolhas definem a margem de lucro do produto, as etapas necessárias no processo e a garantia de que as especificações solicitadas serão atendidas.

Dessa forma, tanto as gráficas (que serão o foco deste trabalho) como as empresas que trabalham com comunicação visual passaram a enfrentar um desafio que tornou-se mais complexo: a preparação do Plano de Impressão e Corte das ordens de serviço, que são as duas principais atividades para a entrega de um pedido. As decisões no momento da montagem do Plano interferem diretamente em diversos aspectos da cadeia produtiva.

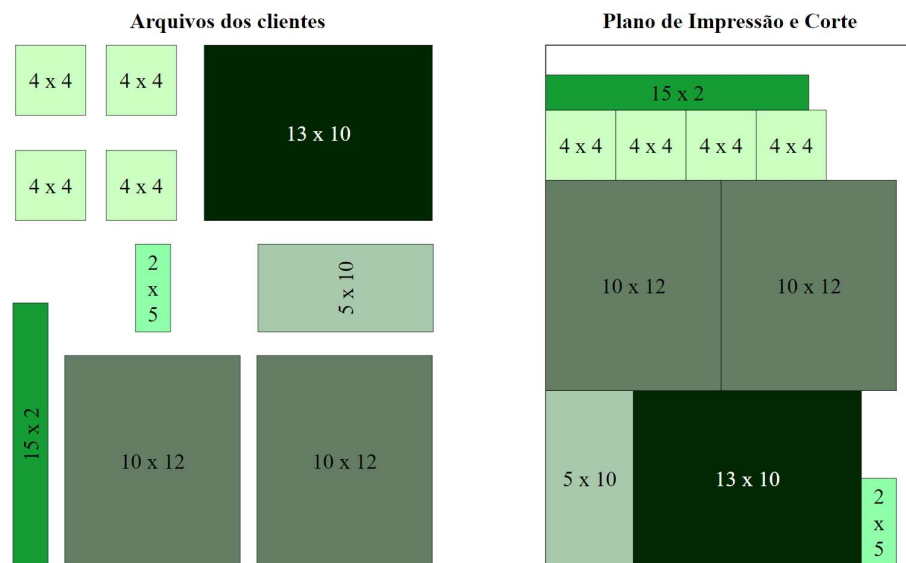
O primeiro fator afetado é a matéria prima. Com a amplificação da oferta de produtos personalizados, diversos materiais estão à disposição do cliente e das empresas: tipos de papéis variados (p.e., couché, *offset*, pólen, fotográfico, etc.); tamanhos de papel variados (A3, A4, formato 6, formato 9, etc.); e gramaturas variadas (75 g, 90 g, 120 g, 170 g, etc.). Logo, o primeiro grande desafio é conciliar as diversas combinações possíveis de opções dentro da operação da empresa para executar os múltiplos pedidos conforme as especificações solicitadas, garantindo a disponibilidade dos materiais e assegurando o atendimento da demanda dos clientes corretamente.

Outro fator afetado é o processo. Como dito anteriormente, o alto grau de personalização combinado com o aumento das opções disponíveis resulta em um baixo nível de padronização dos pedidos. Logo, cada pedido é único em termos de dimensões, quantidades e especificações. O principal desafio, portanto, é garantir que o produto seja entregue no prazo, mantendo a eficiência operacional e as especificações desejadas. Ou seja, deve-se assegurar a execução do prazo combinado com o cliente, respeitando fatores como o tempo gasto para a montagem do Plano (que varia a cada pedido), as regras de corte necessárias e a utilização de diversos materiais (com tamanhos variados) durante todo o processo.

O último fator afetado é o custo da ordem de serviço. A cada pedido impresso uma certa quantidade de folhas é utilizada, dependendo da configuração final do Plano de Impressão e Corte. Do ponto de vista das empresas do setor é essencial buscar a redução desse custo, devido à alta competitividade no setor que é muito sensível ao preço.

Para ilustrar, a Figura 1 a seguir é uma representação de um Plano de Impressão e Corte para uma gráfica. Os itens à esquerda simulam um pedido que uma gráfica recebeu, ou seja, um conjunto de arquivos solicitados pelos clientes para impressão, com suas respectivas dimensões. A imagem à direita representa o Plano já montado com todos os arquivos. Logo, nota-se que o objetivo é alocar os arquivos dentro de uma ou mais folhas padronizadas, da forma mais eficiente e econômica possível. A região em branco representa a *sobra*, que é a área não utilizada das folhas. Ela pode ser reaproveitada ou caracterizada como perda dependendo do tamanho/formato, e também da operação da empresa.

Figura 1: Plano de Impressão e Corte.



Fonte: o Autor.

Considerando o exemplo do Plano apresentado anteriormente, percebe-se que a montagem não é uma tarefa trivial, pois exige a definição do tamanho e da quantidade de folhas a serem utilizadas, bem como o posicionamento das figuras e sua orientação. Além disso, pensando no contexto das gráficas, a ampla gama de papéis disponíveis (com tamanhos variados), o grande volume de pedidos com especificações únicas e o prazo curto de execução evidenciam a grande complexidade de montar os Planos na operação diária de uma gráfica.

Desta maneira, percebe-se que há uma grande oportunidade no setor para aplicar abordagens de otimização que auxiliem no processo de tomada de decisões, especialmente na montagem dos Planos de Impressão e Corte.

1.1 Motivações, Objetivos e Contribuições

O objeto de estudo deste trabalho é uma pequena gráfica familiar localizada na zona sul da cidade de São Paulo, que atualmente não possui nenhum tipo de tecnologia capaz de auxiliá-la na montagem dos Planos de Impressão e Corte. Portanto, o esperado é que a aplicação das soluções propostas neste trabalho seja capaz de apoiar as decisões da empresa, visando a redução dos custos com materiais em relação à situação atual. Uma comparação será feita entre os dois momentos da produção (pré e pós implementação da solução) para o mesmo conjunto de instâncias, evidenciando os custos e comparando as soluções propostas com as soluções anteriormente adotadas na prática.

Entende-se que a aplicação de conceitos de Pesquisa Operacional, em especial o tópico de otimização aplicada na classe de Problemas de Corte e Empacotamento é uma proposta adequada para auxiliar a produção da empresa. Como base de conhecimento para pesquisas e desenvolvimento do trabalho, segundo a tipologia de Problemas de Corte e Empacotamento apresentada em WÄSCHER et al. (2007), o problema enquadra-se como um *Multiple Bin Size Bin Packing Problem*. Alguns trabalhos relevantes que abordam este problema podem ser encontrados em GILMORE e GOMORY (1965), LODI et al. (2002a), LODI e MONACI (2003), LODI et al. (2004), ABRANTES (2012) e NASCIMENTO (2022).

Para alcançar o objetivo de redução de custos, o presente trabalho visa explorar alguns modelos de otimização já desenvolvidos em trabalhos publicados e estendê-los para a realidade da empresa estudada, considerando as especificidades do seu processo produtivo.

Por fim, a implementação dos modelos é realizada em Python, utilizando o Gurobi como o resolvidor de propósito geral. Além disso, a biblioteca Matplotlib é empregada para a geração visual das representações gráficas dos Planos obtidos. O ambiente de desenvolvimento escolhido é o *Jupyter Notebook*, que proporciona ao usuário a capacidade de inserir dados, incluindo as especificações das folhas disponíveis (tamanho e quantidade), bem como as especificações dos arquivos destinados à impressão (tamanho e quantidade). Esse arquivo gera imagens das soluções encontradas pelo modelo, permitindo a visualização do Plano de Impressão e Corte recomendado.

De forma geral, o objetivo do presente trabalho é propor uma solução capaz de gerar os Planos de Impressão e Corte de forma automática priorizando a solução com o menor custo, resolvendo, portanto, a situação atual da empresa. Logo, os objetivos específicos do trabalho podem ser elencados da seguinte forma:

1. Reduzir o custo total na produção dos pedidos por meio de algoritmos que sejam capazes de identificar qual conjunto de objetos é o mais vantajoso financeiramente para empacotar os itens pedidos, respeitando as condições impostas pela operação da gráfica.
2. Mensurar e comparar os impactos financeiros entre as soluções adotadas na prática pela gráfica e as soluções obtidas pelos modelos desenvolvidos, para o mesmo conjunto de instâncias.
3. Desenvolver um arquivo que tem como *output* o resultado do Plano sugerido pelos modelos propostos de forma visual, indicando a disposição dos arquivos para impressão dentro das folhas.

1.2 Estrutura do Trabalho

- **Capítulo 1: Introdução** - contextualização do tema abordado; detalhamento dos objetivos, motivações, e contribuições; apresentação do que será feito/implementado, e descrição da estrutura do trabalho.
- **Capítulo 2: Fundamentação Teórica** - apresentação dos temas mais relevantes relacionados ao problema abordado e representação gráfica dos principais conceitos.
- **Capítulo 3: Descrição do Problema** - apresentação das características do problema abordado, esquematizando o contexto da empresa, definindo o objetivo da

modelagem, detalhando as variáveis importantes e estruturando as restrições do processo.

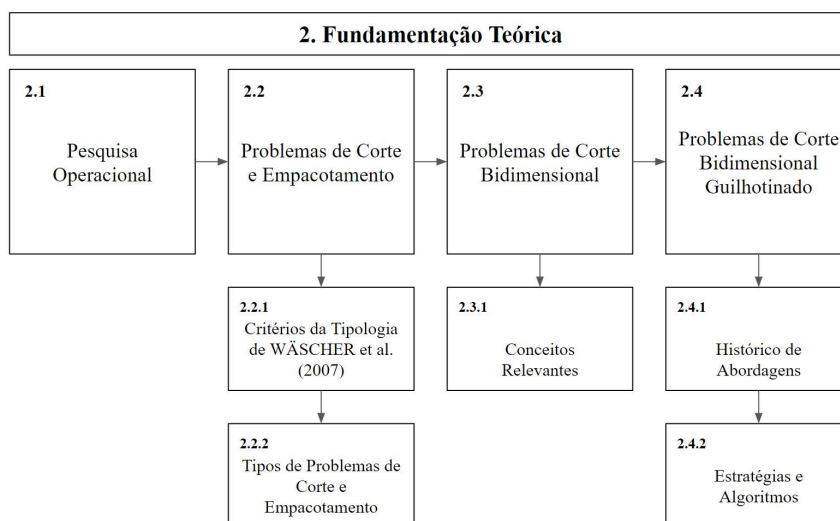
- **Capítulo 4: Modelagem do Problema** - declaração dos modelos desenvolvidos, apresentando a notação utilizada, bem como as funções objetivo e restrições modeladas.
- **Capítulo 5: Testes Computacionais e Análises** - apresentação das instâncias, testes computacionais e análises comparativas entre a solução adotada pela gráfica na prática e os resultados obtidos dos modelos de otimização.
- **Capítulo 6: Conclusões e Perspectivas Futuras** - considerações finais a respeito das principais contribuições do trabalho e discussão das perspectivas futuras acerca da modelagem e da solução desenvolvida.

2 FUNDAMENTAÇÃO TEÓRICA

O objetivo deste capítulo é explorar os principais tópicos relacionados ao problema estudado. Com isso, uma base de conhecimento será formada para garantir o entendimento dos conceitos fundamentais sobre o tema e permitir o desenvolvimento do trabalho.

Primeiramente, é apresentada uma breve introdução sobre a Pesquisa Operacional. Ela é seguida por um aprofundamento inicial sobre os Problemas de Corte e Empacotamento, apresentando um estudo detalhado sobre a tipologia desses problemas com o intuito de garantir as definições claras sobre o problema explorado, permitindo pesquisas direcionadas dentro da área. Em seguida, uma série de conceitos relevantes acerca dos problemas de corte bidimensional são apresentados, com o objetivo de formar uma base sólida de conhecimento para modelar o problema da gráfica. Por fim, é feita uma exploração da vizinhança mais próxima do problema que este trabalho busca solucionar com a especificação de suas principais características, com o histórico de abordagens adotadas e com as estratégias de resolução e algoritmos mais relevantes. A Figura 2 apresenta a estruturação do capítulo.

Figura 2: Estrutura do Capítulo 2.



Fonte: o Autor.

2.1 Pesquisa Operacional

A primeira aparição formal do termo Pesquisa Operacional foi na Segunda Guerra Mundial, pelo exército britânico. O objetivo era decidir a melhor forma de distribuir os recursos bélicos baseando-se em ciência, deixando de lado as opiniões pessoais (TAHA, 2017). Os avanços desse estudo na esfera militar foram, então, direcionados para aumentar a produtividade e eficiência na esfera civil.

A área de Pesquisa Operacional, assim como no método científico, é estruturada por uma sequência de etapas que visa investigar e fornecer respostas sobre um problema (HILLIER; LIEBERMAN, 2010). O primeiro passo é a análise e formulação desse problema, que envolve a coleta de dados e o entendimento completo do contexto no qual ele está inserido. O próximo passo é a construção do modelo, que visa descrever o cenário estudado em uma linguagem matemática. Posteriormente, é realizada uma sessão de validação do modelo através de experimentos e, por fim, é realizado o refinamento do modelo. Caso o problema estudado seja de otimização (como neste trabalho), todo esse processo tem como objetivo encontrar a melhor solução (solução ótima) para o caso.

Ou seja, a área de Pesquisa Operacional pode ser entendida como um ramo da engenharia (e da matemática) que visa fornecer uma base científica para auxiliar a tomada de decisões (MURTHY, 2007) através da estruturação do problema por meio de um modelo. Especificamente sobre o modelo matemático, ele é composto por expressões (p.e., equações, inequações) que visam descrever com precisão o problema estudado. Ele possui três principais componentes: a função objetivo, as variáveis de decisão e o conjunto de restrições. A função objetivo pode ser de maximização ou minimização, a depender do propósito da resolução (p.e., maximizar o lucro de uma operação, ou minimizar os custos). As variáveis de decisão representam as tomadas de decisão que o modelo irá processar, e portanto, alteram o valor final da função objetivo. As restrições, por sua vez, visam aplicar as condições impostas pelo cenário no qual o problema está envolvido, afetando diretamente o valor das variáveis de decisão. Logo, o grande objetivo é escolher os valores das variáveis de decisão que atingem o melhor valor possível para a função objetivo, respeitando as restrições impostas à modelagem.

Determinar esses valores de forma apropriada não é algo trivial, tanto pela coleta de informações (nem sempre os problemas estudados possuem dados estruturados para a análise), quanto pela modelagem em si. Por isso, outra prática comum na área de Pesquisa Operacional é a análise de sensibilidade, que busca refinar o modelo a partir de testes com as variáveis de decisão e com os parâmetros (as constantes), por meio

da alteração, retirada ou inclusão de outros valores plausíveis. Ou seja, a análise de sensibilidade permite o entendimento de como as mudanças nos parâmetros alteram a solução ótima de um problema (WINSTON, 2004).

Outro ponto interessante a ser mencionado é que nem sempre a solução ótima é possível de ser encontrada: seja por uma restrição de capacidade computacional ou de tempo. Dessa forma, heurísticas são desenvolvidas para atender às condições limitantes que o contexto do problema impõe. Em geral, as heurísticas descrevem uma classe de procedimentos para encontrar soluções aceitáveis para uma variedade de problemas complexos, ou seja, são uma coleção de regras ou etapas que orientam alguém para uma solução que pode ou não ser a solução ótima (GASS; FU, 2013).

2.2 Problemas de Corte e Empacotamento

Dentro da extensa área de Pesquisa Operacional há a classe de Problemas de Corte e Empacotamento, que se destaca por seu amplo repertório teórico combinado com uma vasta aplicação em problemas de produção, como por exemplo nos setores metalúrgico, têxtil ou gráfico. Ela já foi discutida em diversos trabalhos, como em DYCKHOFF (1990), DOWSLAND e DOWSLAND (1992), SWEENEY e PATERNOSTER (1992), MARTELLO (1994), BISCHOFF e WÄSCHER (1995), LODI et al. (2002b), WANG e WÄSCHER (2002), WÄSCHER et al. (2007), MORABITO et al. (2009), IORI et al. (2021) e OLIVEIRA et al. (2022). Em SWEENEY e PATERNOSTER (1992) há um compilado de livros, artigos e dissertações que, juntos, até então, somavam mais de 400 trabalhos envolvendo o tópico, o que já dá dimensão sobre a importância e desenvolvimento do tema.

Um ponto interessante em relação a essa classe de problemas é que a forte relação entre corte e empacotamento se dá pela dualidade entre o corpo material e o espaço ocupado por ele (FAINA, 2020). Ou seja, há uma dualidade entre cortar material/empacotar espaço e cortar espaço/empacotar material (JUNQUEIRA, 2009). A mesma lógica vale para o universo gráfico: cortar uma folha de impressão em pequenas partes (imagens), é o mesmo que empacotar essas imagens dentro de uma folha. Por isso, os termos Corte e Empacotamento caminham juntos nessa família de problemas.

A primeira tipologia amplamente reconhecida sobre os Problemas de Corte e Empacotamento foi apresentada em DYCKHOFF (1990) com o intuito de unificar as distintas notações na literatura e concentrar a pesquisa futura em problemas específicos. Porém,

com a evolução dos estudos sobre os Problemas de Corte e Empacotamento, a categorização tornou-se insuficiente para cobrir todas as especificidades nas quais a área já havia avançado. Assim, tomando como base o trabalho de Dyckhoff, uma nova proposta foi apresentada em WÄSCHER et al. (2007), que será detalhada a seguir.

A estrutura utilizada no trabalho define que existem apenas dois conjuntos de elementos, que são os *objetos* grandes (equivalentes a entrada ou oferta) e os *itens* pequenos (que são a saída ou demanda). Os elementos são geométricos, e devem obedecer a duas condições: os itens não podem se sobrepor, e devem estar inteiramente contidos dentro do conjunto de objetos grandes. Além disso, assim como um problema tradicional de Pesquisa Operacional, uma função objetivo deve ser otimizada, seja ela mono ou multi-objetivo. Dependendo do contexto do problema estudado, a solução do modelo não precisa necessariamente utilizar todos os objetos ou itens, ela pode selecionar apenas uma porção. Por fim, para atingir o ótimo global há 5 sub-problemas que devem ser resolvidos simultaneamente:

- Problema de seleção em relação aos *objetos*.
- Problema de seleção em relação aos *itens*.
- Problema de agrupamento em relação aos *itens* escolhidos.
- Problema de alocação em relação à atribuição dos *itens* aos *objetos*.
- Problema de layout em relação ao arranjo dos *itens* em cada um dos *objetos* selecionados, respeitando as condições geométricas.

Segundo os autores, existem 3 grandes motivações geradas pela tipologia desenvolvida por Dyckhoff que os fizeram apresentar uma nova proposta: *i)* nem todos os Problemas de Corte e Empacotamento podem ser atribuídos exclusivamente a um tipo de sub-problema; *ii)* a tipologia é parcialmente inconsistente, e sua aplicação pode levar a resultados confusos; e *iii)* a aplicação da tipologia não resulta necessariamente em uma categorização homogênea dos problemas estudados. Assim, a nova tipologia foi desenvolvida com base em 5 critérios, que serão apresentados na Subseção 2.2.1.

A primeira definição trazida pela tipologia de Wäscher diz respeito aos aspectos do problema estudado. Caso ele envolva apenas aspectos de Problemas de Corte e Empacotamento, então, diz-se que ele é do tipo *puro*; caso ele envolva aspectos adicionais, então ele é uma *extensão* do problema, como vemos nos seguintes trabalhos: problema

de minimização de padrões em VANDERBECK (2000); problema de sequenciamento de padrões em FOERSTER e WÄSCHER (1998), YANASSE (1997) e YUEN (1995); ou no problema de dimensionamento de lote em NONÅS e THORSTENSON (2000).

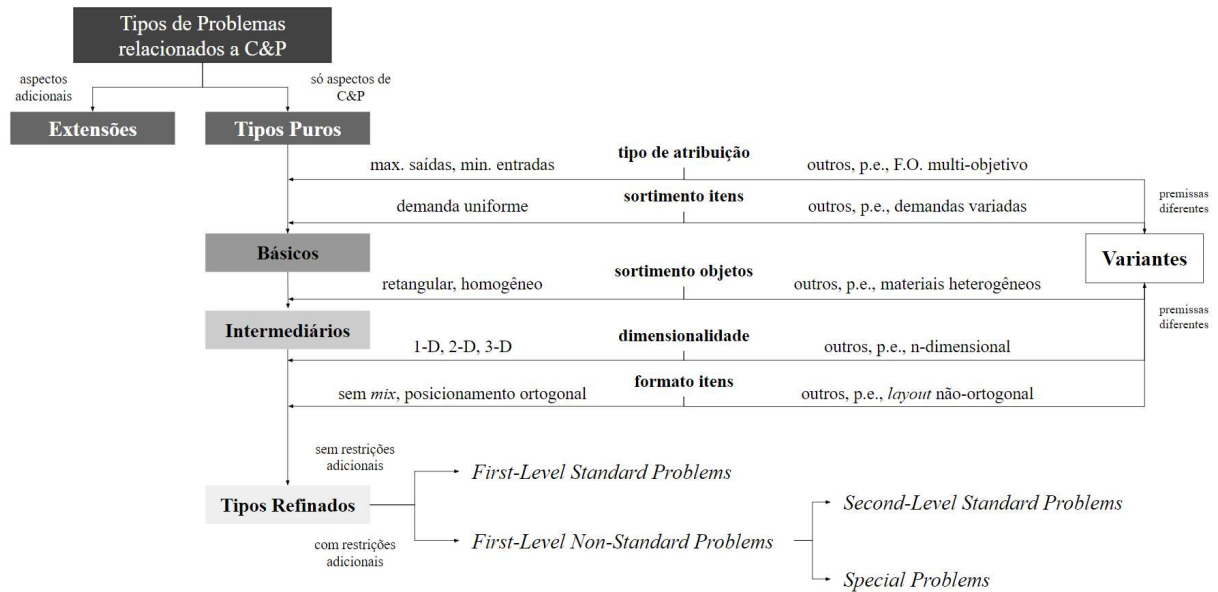
Dentro dos problemas *puros*, 5 critérios são definidos para categorizá-los: tipo de atribuição, sortimento dos itens pequenos, sortimento dos objetos grandes, dimensionalidade e formato dos itens pequenos. Caso apenas os dois primeiros sejam definidos durante o estudo do caso, temos um problema do tipo *básico*; caso os três primeiros sejam definidos, temos um problema do tipo *intermediário*; caso todos os critérios sejam utilizados na definição, temos um problema do tipo *refinado*.

Cada um dos critérios possui algumas premissas padronizadas, que serão apresentadas na Subseção 2.2.1 a seguir. Caso elas sejam diferentes no caso estudado, então temos uma *variante* do problema *puro*, como vemos nos trabalhos a seguir: problemas com múltiplos objetivos em WÄSCHER (1990); problemas estocásticos em DAS e GOSH (2003); ou problemas *online* em HEMMINKI et al. (1998).

Por fim, dentro dos problemas refinados, há uma distinção entre dois grupos. Caso o problema não apresente nenhuma característica ou restrição adicional além dos 5 critérios já definidos, temos um *problema padrão de primeiro nível*. Em contrapartida, caso ele envolva restrições adicionais, então, ele é chamado de *problema não padronizado de primeiro nível*.

A estruturação apresentada anteriormente pode ser representada pela Figura 3, que resume a categorização proposta para os Problemas de Corte e Empacotamento. A primeira divisão representa se os problemas são puros ou não. Dentro dos problemas puros, dado o número de critérios definidos, o problema pode ser básico, intermediário ou refinado (representado pelas setas na porção esquerda) se seguir as premissas padronizadas que estão elencadas na imagem; caso contrário, caracteriza-se como uma variante (setas apontando para o retângulo à direita). Finalmente, as últimas divisões analisam se há restrições adicionais ou não.

Figura 3: Estruturação da tipologia de WÄSCHER et al. (2007).



Fonte: adaptado de WÄSCHER et al. (2007).

2.2.1 Critérios da Tipologia de WÄSCHER et al. (2007)

Os 5 critérios usados como base da tipologia apresentada em WÄSCHER et al. (2007) serão descritos a seguir.

O primeiro critério é o **Tipo de Atribuição**. Há um conjunto de itens que precisa ser cortado ou empacotado em um conjunto de objetos, e com isso, a tipologia define dois casos: o de *maximização das saídas* e o de *minimização das entradas*. No primeiro caso, o conjunto de objetos não é capaz de acomodar todos os itens, ou seja, não há um problema de seleção de objetos, já que todos serão utilizados. Logo, o desafio é selecionar o subconjunto de itens de maior valor a ser cortado/empacotado. Já no segundo cenário, o conjunto de objetos é capaz de acomodar todos os itens. Dessa forma, há um desafio de seleção dos objetos, que visa escolher o subconjunto de menor valor capaz de acomodar todos os itens. O *valor* mencionado deve ser especificado no problema estudado (já que pode representar custos, lucros, materiais utilizados, etc.), mas, normalmente, ele é diretamente proporcional ao tamanho do item ou do objeto em questão. Caso o problema envolva uma função objetivo com mais de um objetivo, ou que exija a seleção tanto dos objetos quanto dos itens, ele é tratado como uma variante.

O segundo critério é o **Sortimento dos Itens Pequenos**. Há três possíveis casos:

itens idênticos, *sortimento fracamente heterogêneo* e *sortimento fortemente heterogêneo*. No primeiro caso, todos os itens são idênticos em formato e tamanho. No segundo caso, há alguns tipos de itens idênticos (cada um com uma demanda relativamente grande), e eles podem ser agrupados em poucos grupos (ao comparar com o número total); por definição, se a orientação dos itens mudar, eles são itens distintos e, portanto, são de tipos distintos (consequentemente, são agrupados separadamente). No terceiro caso, o conjunto de itens é caracterizado pelo fato de que raramente os itens compartilham semelhanças entre si, ou seja, não há itens idênticos em formato e tamanho (e, por isso, a demanda de cada um é unitária). Por fim, vale ressaltar que para os problemas padrão, a demanda dos itens é uniforme (ou seja, não há itens com alta demanda e outros itens com baixa demanda). Os problemas que possuem demandas muito diferentes são tratados como variantes.

O terceiro critério é o **Sortimento dos Objetos Grandes**. Há dois possíveis casos, que englobam outras categorias mais específicas internamente. O primeiro caso é o de *um único objeto*, que pode ser estendido para duas categorias específicas: o caso onde todas as dimensões são fixas ou o caso onde pelo menos uma dimensão pode variar. Já o segundo caso é o de *múltiplos objetos*, que pode ser estendido para três categorias específicas: todos os objetos são idênticos, eles são fracamente heterogêneos ou são fortemente heterogêneos. É importante mencionar que no caso de *múltiplos objetos*, as dimensões são sempre fixas (nenhuma dimensão pode variar). Para os problemas *básicos* (com os dois critérios iniciais definidos), tanto em duas como em três dimensões considera-se que os objetos possuem formatos retangulares (retângulos no caso bidimensional e cubóides no caso tridimensional) e são do mesmo material. Os casos nos quais essas premissas não são verdadeiras são tratados como variantes.

O quarto critério é a **Dimensionalidade**. Na tipologia apresentada, as dimensões consideradas como padrão são: *unidimensional* (p.e., corte de um rolo de papel em partes menores, na qual a única dimensão relevante é o comprimento do rolo); *bidimensional* (p.e., elaboração de um Plano de Corte para uma gráfica, na qual as dimensões de altura e largura são relevantes); *tridimensional* (p.e., arranjo de caixas dentro de um contêiner, onde as 3 dimensões são relevantes). Casos com 4 ou mais dimensões, como em (LINS et al., 2002), são tratados como variantes. Os dois exemplos mais comuns de variantes são: *multidimensional* (p.e., inclusão do fator temporal em uma análise) e *dimensão-aberta*, quando não há restrição em uma das dimensões relevantes (p.e., um caso 1.5D significa que uma das dimensões não está fixa, seja ela a largura ou a altura; um exemplo pode ser encontrado em (KOKTEN; SEL, 2022) para a indústria madeireira).

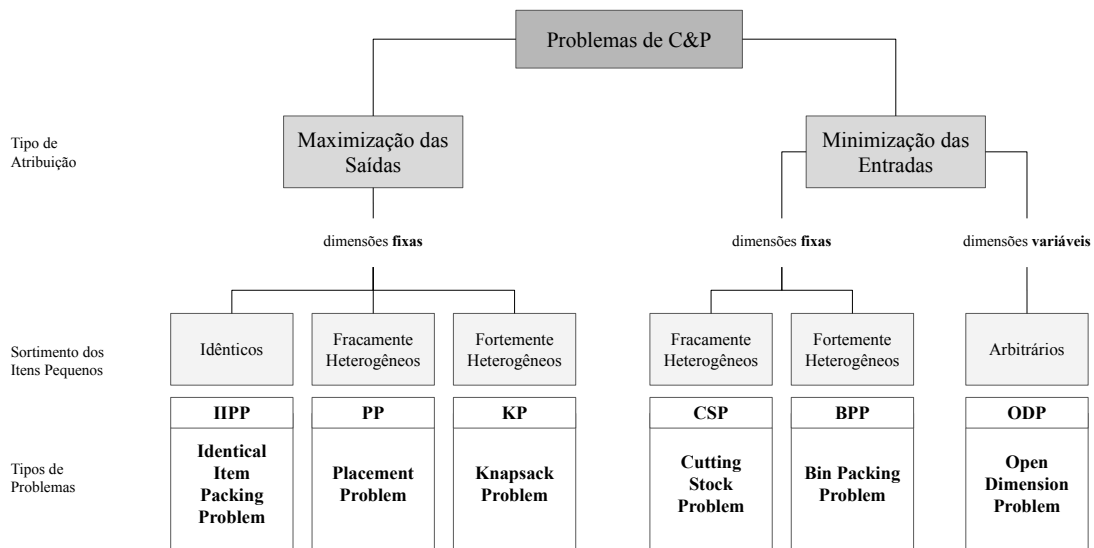
O quinto critério é o **Formato dos Itens Pequenos**. Há dois possíveis casos que

normalmente são aplicados em duas ou três dimensões: itens *regulares* (p.e., retângulos, círculos, etc.) e itens *irregulares*. Tratando mais especificamente sobre os retângulos, que são os itens mais estudados, geralmente eles são posicionados de forma ortogonal. Além disso, de forma padronizada, o conjunto de itens é inteiramente de itens regulares, ou inteiramente de itens irregulares. Os casos que consideram um posicionamento não-ortogonal ou um conjunto de itens que aceita os dois formatos são tratados como variantes.

2.2.2 Tipos de Problemas de Corte e Empacotamento

Os tipos de Problemas de Corte e Empacotamento são definidos mediante a especificação dos cinco critérios apresentados anteriormente. Os problemas *básicos*, como já mencionado, combinam a definição apenas dos dois primeiros critérios apresentados para segmentar o caso estudado: tipo de atribuição e sortimento dos itens pequenos. Com essas definições, esses problemas formam o alicerce da nomenclatura dos tipos básicos de Problemas de Corte e Empacotamento. A Figura 4 ilustra as combinações relevantes de critérios e seus respectivos tipos.

Figura 4: Tipos básicos de Problemas de Corte e Empacotamento.



Fonte: adaptado de WÄSCHER et al. (2007).

Com o intuito de definir tipos de problemas mais homogêneos, a definição do terceiro critério (sortimento dos objetos grandes) implica em problemas *intermediários*, que formam o panorama geral dos Problemas de Corte e Empacotamento, com as nomenclaturas mais utilizadas pela área no desenvolvimento dos trabalhos. Para cada tipo de problema definido, uma sigla foi elaborada, tornando a identificação de cada caso única.

As Figuras 5 e 6 a seguir detalham os tipos de problemas intermediários seguindo a estruturação nos três critérios considerados. Nas linhas temos as características dos objetos grandes, enquanto nas colunas temos o sortimento dos itens pequenos. Porém, como a definição do primeiro critério (tipo de atribuição) é fundamental para definir o problema estudado, a Figura 5 representa os problemas de maximização das saídas, enquanto a Figura 6 ilustra os problemas de minimização das entradas.

Figura 5: Problemas de maximização das saídas.

características dos objetos \ sortimento dos itens		idênticos	fracamente heterogêneos	fortemente heterogêneos
dimensões fixas	um único objeto	IIPP	SLOPP	SKP
		Identical Item Packing Problem	Single Large Object Placement Problem	Single Knapsack Problem
	idênticos		MILOPP	MIKP
			Multiple Identical Large Object Placement Problem	Multiple Identical Knapsack Problem
	diferentes		MHLOPP	MHKP
			Multiple Heterogeneous Large Object Placement Problem	Multiple Heterogeneous Knapsack Problem

Fonte: adaptado de WÄSCHER et al. (2007).

Figura 6: Problemas de minimização das entradas.

características dos objetos \ sortimento dos itens		fracamente heterogêneos	fortemente heterogêneos
dimensões fixas	idênticos	SSSCSP	SBSBPP
		Single Stock Size Cutting Stock Problem	Single Bin Size Bin Packing Problem
	fracamente heterogêneos	MSSCSP	MBSBPP
		Multiple Stock Size Cutting Stock Problem	Multiple Bin Size Bin Packing Problem
	fortemente heterogêneos	RCSP	RBPP
		Residual Cutting Stock Problem	Residual Bin Packing Problem
um único objeto dimensões variáveis		ODP	
		Open Dimension Problem	

Fonte: adaptado de WÄSCHER et al. (2007).

Por fim, caso os dois últimos critérios sejam definidos (dimensionalidade e formato dos itens pequenos), temos os problemas *refinados*. As nomenclaturas resultantes são caracterizadas por adjetivos que são adicionados aos nomes já apresentados nos problemas intermediários. Generalizando, segue-se um padrão que adiciona primeiro a dimensionalidade (D), depois o formato (F) e por último o tipo de problema intermediário (TPI), resultando em uma terminologia genérica D-F-TPI.

Logo, se considerarmos um problema de minimização das entradas bidimensional, com múltiplos objetos fracamente heterogêneos e um conjunto de itens retangulares com sortimento fortemente heterogêneo, temos um *Two-Dimensional Rectangular Multiple Bin Size Bin Packing Problem*, ou seja, um 2D-R-MBSBPP. Este é o tipo de problema refinado que constitui o foco deste trabalho pois melhor se alinha à operação de uma gráfica, já que reflete o cenário em que há múltiplos arquivos retangulares fortemente heterogêneos para impressão (cada pedido é único em termos de quantidade e dimensões), que necessitam ser cortados em um conjunto fracamente heterogêneo de folhas padronizadas (que possuem poucas opções de tamanho).

2.3 Problemas de Corte Bidimensional

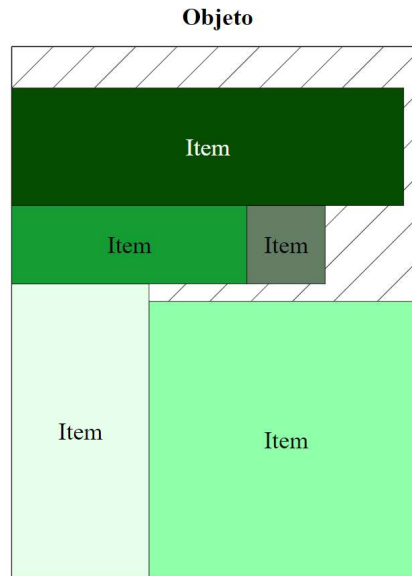
Como o objetivo do presente trabalho é otimizar os Planos de Impressão e Corte de uma gráfica, os Problemas de Corte e Empacotamento Bidimensional (como apresentados na tipologia vista na seção anterior) estão diretamente relacionados aos modelos desenvolvidos. Portanto, eles serão o foco principal do capítulo a partir deste momento.

Inicialmente, vale destacar dois trabalhos interessantes acerca do tema. Recentemente, em IORI et al. (2021), um novo trabalho de revisão bibliográfica sobre os Problemas de Corte e Empacotamento Bidimensional foi realizado, abordando tanto os métodos exatos desenvolvidos quanto as heurísticas mais aplicadas. O trabalho também aborda técnicas de pré-processamento e métodos de relaxação. Além dele, em OLIVEIRA et al. (2022) há uma lista de trabalhos focada em Problemas de Corte e Empacotamento que envolvem elementos retangulares, que estão relacionados com o problema abordado no presente trabalho.

Primeiramente, é importante ressaltar que, para fins de uniformização, existe uma nomenclatura padrão de alguns termos sempre presentes nesse tipo de problema. Os *itens* são os corpos pequenos, que precisam ser empacotados/cortados; já os *objetos* são os corpos grandes, nos quais os itens pequenos serão alocados (WÄSCHER et al., 2007). A modelagem, então, é responsável por definir a distribuição dos itens dentro dos objetos, e caso alguma parte não seja utilizada, ela é chamada de *sobra*. Na Figura 7 a seguir é possível observar que o retângulo maior é o objeto, os elementos coloridos menores são os itens e a área branca hachurada é a sobra.

Um tema relevante acerca dessa família de problemas é que essas alocações (dos itens dentro dos objetos), em termos matemáticos, podem ser traduzidas para uma série de combinações que precisam ser testadas até o modelo encontrar a melhor configuração possível. Por isso, esse tipo de problema é comumente caracterizado como *NP-hard*, que possui a característica de não ser facilmente solucionável. Ou seja, isso significa que os algoritmos que resolvem o problema de maneira ótima terão que gerar e comparar muitas soluções candidatas e, portanto, podem consumir muito tempo para encontrar a melhor solução (FAINA, 2020), especialmente em problemas nos quais as instâncias envolvem diversos itens e objetos. Por esse motivo, a utilização de heurísticas é comum em problemas dessa família.

Figura 7: Representação de objeto, item e sobra.



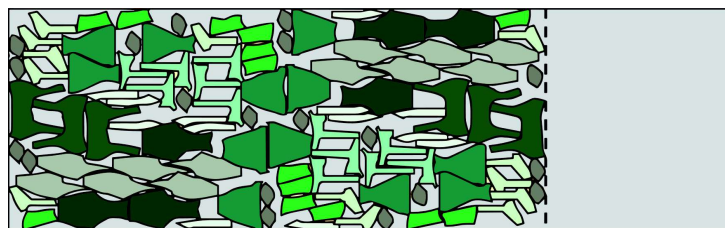
Fonte: o Autor.

2.3.1 Conceitos Relevantes

Dentro da família de Problemas de Corte e Empacotamento Bidimensional, há uma série de conceitos relevantes que se estendem por inúmeras aplicações e variantes. Os principais serão apresentados a seguir, com o intuito de formar uma base sólida de conhecimento para o desenvolvimento deste trabalho.

O primeiro conceito envolve a forma dos itens a serem empacotados. Eles podem assumir formas *regulares* (p.e., retângulos, círculos, etc.), que envolvem os estudos mais comuns dentro da área; mas também podem assumir formas *irregulares*, uma variante do problema muito mais complexa de ser resolvida por conta do formato irregular dos itens, que não podem se sobrepor (LEAO et al., 2020). A Figura 8 exemplifica um empacotamento de itens irregulares.

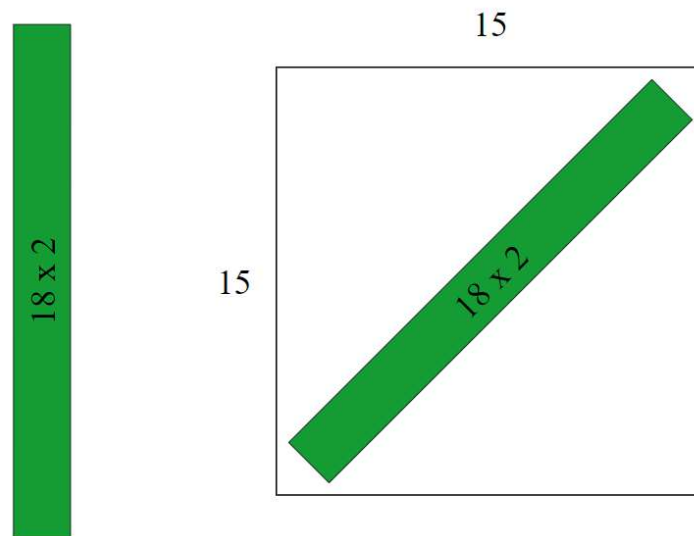
Figura 8: Empacotamento de itens irregulares.



Fonte: adaptado de BENNEL e OLIVEIRA (2008).

O segundo conceito envolve o posicionamento dos itens. A maioria dos problemas estudados envolvem o empacotamento de itens com posicionamento *ortogonal*, ou seja, os lados do item estão paralelos ou perpendiculares aos lados do objeto. Porém, em DE CANI (1978) há uma demonstração de que, na prática, o posicionamento *não-ortogonal* também deve ser considerado: o posicionamento não-ortogonal pode levar a melhores arranjos, ou até mesmo à inclusão de itens que não podem ser empacotados ortogonalmente no objeto. A Figura 9 exemplifica esse caso, evidenciando que, para o mesmo objeto, somente o posicionamento não-ortogonal é capaz de empacotar o retângulo alongado, já que ele possui um comprimento maior do que os lados do objeto.

Figura 9: Empacotamento não-ortogonal.

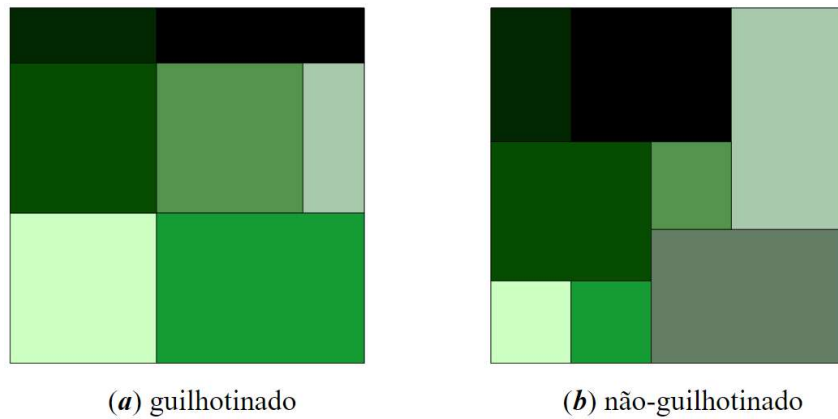


Fonte: o Autor.

O terceiro conceito envolve o padrão de corte e empacotamento. Em problemas nos quais os itens só podem ser obtidos ao realizar cortes ortogonais de um lado ao outro do objeto, sem passar por cima de qualquer item, diz-se que o padrão utilizado é o *guilhotinado* (SCHEITHAUER, 2017). Caso os cortes não sejam ortogonais, ou possam ser interrompidos antes de atingir o outro lado do objeto (como cortes com uma tesoura, por exemplo), temos o caso *não-guilhotinado*. A Figura 10 ilustra os dois padrões: guilhotinado (a) e não guilhotinado (b).

O quarto conceito envolve os estágios de corte. Caso o problema estudado imponha um limite máximo de cortes para a obtenção de um item, supondo que esse limite seja n , então, ele é classificado como um corte em n -estágios. Nessa definição, a sequência de cortes sempre segue um padrão: o primeiro corte normalmente é na horizontal; o

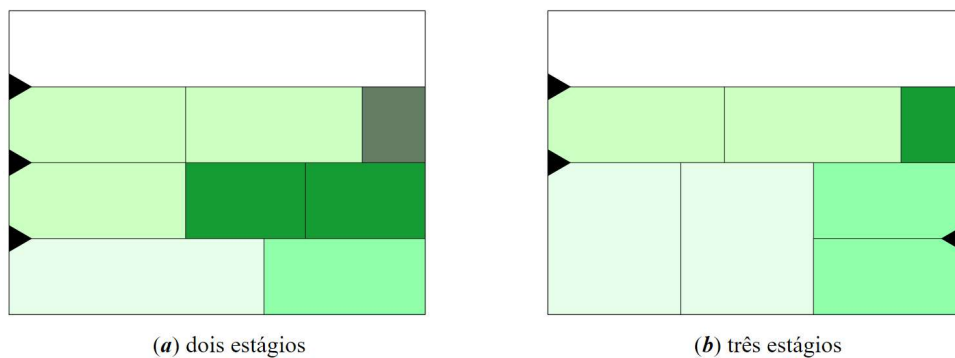
Figura 10: Padrões guilhotinado e não-guilhotinado.



Fonte: adaptado de DO NASCIMENTO et al. (2022).

segundo corte é na vertical; o próximo corte é na horizontal novamente; e assim por diante, alternadamente. Caso ele não possua restrições quanto ao número máximo de cortes, então a classificação de estágios não é aplicada (OLIVEIRA et al., 2022). A Figura 11 ilustra dois exemplos: corte em dois estágios (a) e corte em três estágios (b). O primeiro estágio é identificado pela seta apontando para a direita, o segundo estágio são os cortes na vertical, e o terceiro estágio (presente somente em b) é representado pela seta apontando para a esquerda.

Figura 11: Estágios de corte.

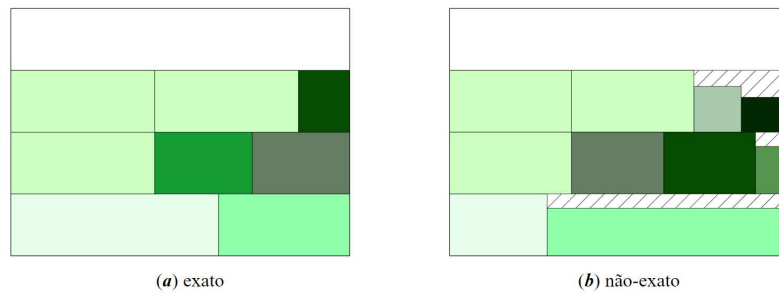


Fonte: adaptado de OLIVEIRA et al. (2022).

O quinto conceito envolve a exatidão dos cortes. Problemas com restrição no número de cortes podem envolver a necessidade de um corte adicional para extrair o item, nos cenários em que as dimensões do item não são compatíveis com os cortes realizados pelos estágios anteriores. A esse corte adicional dá-se o nome de *trimming*, e ele não se configura como um novo estágio. Portanto, diz-se que um problema é *exato* caso ele não permita o

trimming, e *não-exato* caso ele permita (OLIVEIRA et al., 2022). A Figura 12 representa os dois tipos de problema: exatos (a) e não-exatos (b). A área em branco hachurada é caracterizada como *sobra*, já que essa área é retirada para extrair o item, mas não possui as dimensões de um objeto padrão. Dependendo do formato da sobra, de seu tamanho e também levando em conta a operação da empresa, as sobras podem ser *reaproveitáveis*, caso elas possam ser incluídas novamente no processo (ANDRADE et al., 2016), ou serem caracterizadas como *perdas*, caso elas não possam retornar à operação.

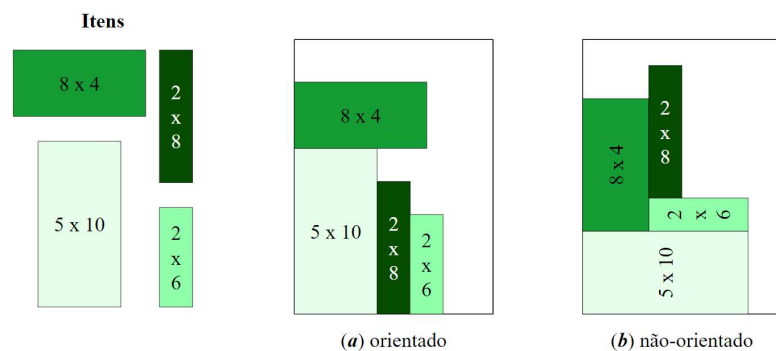
Figura 12: Padrões exato e não-exato.



Fonte: adaptado de OLIVEIRA et al. (2022).

O sexto conceito envolve a orientação dos itens. Os problemas podem permitir ou proibir a rotação dos itens em 90° . Na indústria de impressão (o foco deste trabalho), no caso de problemas bidimensionais, os termos mais utilizados para descrever a orientação de um item são: *paisagem*, quando a largura é maior que a altura; ou *retrato*, quando a altura é maior que a largura. Quando o problema proíbe as rotações, ele é chamado de *orientado*; caso permita as rotações em 90° , diz-se que ele é *não-orientado* (DELL'AMICO et al., 2002). A Figura 13 a seguir representa os dois cenários, para um conjunto de itens que está definido à esquerda.

Figura 13: Padrões orientado e não-orientado.

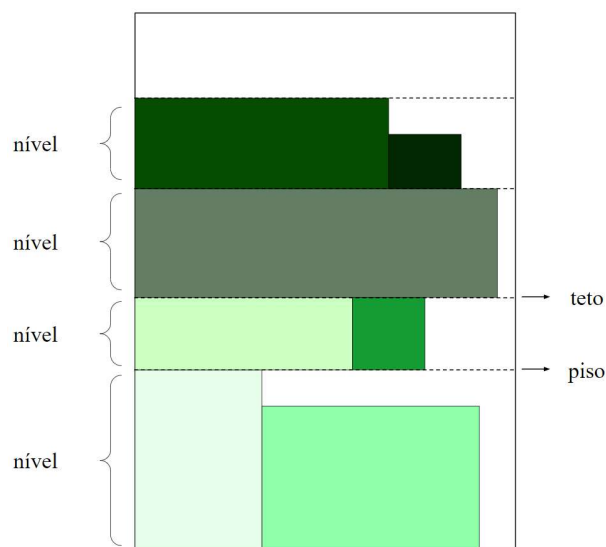


Fonte: o Autor.

O sétimo conceito envolve limitações sobre a quantidade de itens cortados e está presente em OLIVEIRA et al. (2022). Um problema é considerado *restrito* quando um limite sobre a quantidade de itens a serem cortados é definido, seja ele inferior ou superior. Caso ambos limites sejam definidos, então ele é *duplamente-restrito*. Caso nenhum dos limites seja definido, ele é *irrestrito*. Além disso há outro conceito interessante em OLIVEIRA et al. (2022), o de ponderação: caso o valor de todos os itens seja igual às suas dimensões (p.e., área no caso bidimensional) temos um caso *não-ponderado*; caso contrário, ou seja, se o item possuir um outro valor associado que leve em conta fatores distintos (p.e., custo ou nível de prioridade), então temos um caso *ponderado*.

O oitavo e último conceito envolve a acomodação dos itens. Em diversos casos estudados, há um grande número de itens a serem cortados dentro de um objeto. Há uma abordagem clássica, e muito utilizada, que envolve empacotar os itens em *prateleiras*, ou em *níveis*, que são faixas delimitadas ao longo de todo o objeto. Para fins de padronização, o presente trabalho optará por utilizar o termo *níveis*. Segundo LODI et al. (2002b) o primeiro nível é a porção inferior do objeto, e os próximos níveis são produzidos pela linha horizontal que coincide com o topo do item mais alto colocado no nível abaixo. Ou seja, a altura do item mais alto de um nível define o *teto* de seu nível, e consequentemente o *piso* do próximo nível. Não é exigido que os níveis possuam a mesma altura, e nem que os itens estejam sempre alocados no piso (embora essa seja a situação mais comum). A Figura 14 ilustra diversos níveis em um objeto.

Figura 14: Representação de níveis em um objeto.



Fonte: o Autor.

2.4 Problemas de Corte Bidimensional Guilhotinado

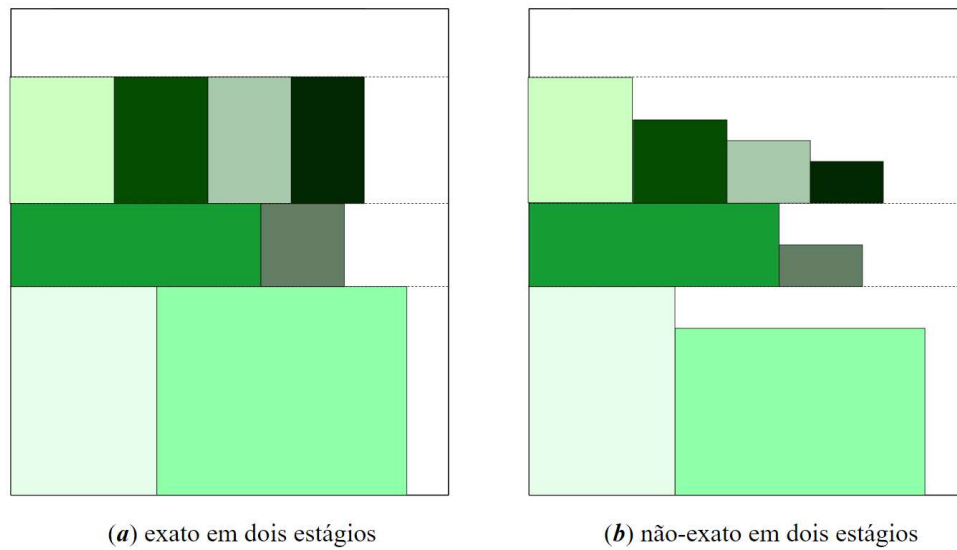
Dado que o objeto de estudo do presente trabalho é uma gráfica, os Problemas de Corte Bidimensional Guilhotinado passam a ser o foco do Capítulo 2 a partir deste momento, já que o setor gráfico, assim como outras indústrias (p.e., indústria moveleira ou de chapas de madeira), utiliza a estratégia de cortes guilhotinados na montagem dos Planos de Impressão e Corte. Diversos trabalhos já foram escritos acerca dessa vizinhança de problemas nessas indústrias, como em GILMORE e GOMORY (1965), FARLEY (1988), YANASSE et al. (1991), CARNIERI et al. (1994), MORABITO e ARENALES (2000) e BELLUZZO e MORABITO (2005).

Apenas lembrando alguns conceitos já mencionados na Subseção 2.3.1, um corte é dito guilhotinado quando os itens são obtidos por meio de cortes ortogonais de um lado ao outro do objeto, sem serem interrompidos. Ou seja, no caso dos problemas bidimensionais com objetos e itens retangulares, um corte guilhotinado produz dois novos retângulos.

Outro conceito importante, especialmente para a indústria gráfica, é a quantidade de estágios de corte. Usualmente, os itens devem ser obtidos em até dois estágios. No primeiro estágio, cortes guilhotinados horizontais (paralelos à base) são realizados no objeto, para a obtenção de um conjunto de tiras, que são os níveis. No segundo estágio, então, são realizados cortes guilhotinados verticais (paralelos ao lado) nas tiras, para a obtenção dos itens. Se não houver necessidade de cortes adicionais (ou seja, se dentro de cada nível os itens possuírem alturas idênticas), diz-se que o padrão de corte guilhotinado é *exato*; caso ainda haja necessidade de realizar o *trimming*, diz-se que o padrão é *não-exato*. A Figura 15 a seguir representa os dois padrões de corte guilhotinados em dois estágios, exato (*a*) e não-exato (*b*).

Seguindo a tipologia apresentada em WÄSCHER et al. (2007) na Subseção 2.2.2, há 4 tipos principais dos Problemas de Corte e Empacotamento Bidimensional que consideram a alocação de itens distintos: *Placement Problem*, caso o problema seja de maximização das saídas com um sortimento fracamente heterogêneo de itens; *Knapsack Problem*, problema de maximização das saídas, mas com um sortimento fortemente heterogêneo de itens; *Cutting Stock Problem*, caso o problema seja de minimização das entradas com um sortimento fracamente heterogêneo de itens; e por fim o *Bin Packing Problem*, problema de minimização das entradas, mas com um sortimento fortemente heterogêneo de itens. A Figura 16 a seguir resume os quatro principais tipos de problemas.

Figura 15: Cortes guilhotinados em dois estágios, exatos e não-exatos.



Fonte: o Autor.

Figura 16: Principais tipos de problemas com itens distintos.

	itens fracamente heterogêneos	itens fortemente heterogêneos
maximização das saídas	PP Placement Problem	KP Knapsack Problem
minimização das entradas	CSP Cutting Stock Problem	BPP Bin Packing Problem

Fonte: o Autor.

Olhando para os Problemas de Corte Bidimensional Guilhotinado em dois estágios, que serão o foco do presente trabalho, a maioria dos trabalhos presentes na literatura lidam ou com o *Knapsack Problem* em dois estágios, ou com o *Cutting Stock Problem* em dois estágios (ANDRADE et al., 2016). A primeira tentativa de modelagem para esses problemas foi vista em GILMORE e GOMORY (1965). Através de uma extensão da modelagem proposta pelos dois autores para o caso unidimensional, eles propuseram uma

abordagem através do método SIMPLEX com um procedimento de geração de colunas para gerar o padrão de corte em dois estágios, utilizando níveis para alocar os itens.

Outros trabalhos relevantes que buscam apresentar uma proposta de solução para o caso bidimensional com cortes guilhotinados em dois estágios podem ser encontrados em BEASLEY (1985), HADJICONSTANTINOU e CHRISTOFIDES (1995), MORABITO e ARENALES (1996), RIEHME et al. (1996), MORABITO e GARCIA (1998), CUI et al. (2005), YANASSE e KATSURAYAMA (2005), BELOV e SCHEITHAUER (2006), CINTRA et al. (2008) e HIFI et al. (2009).

Como o foco do presente trabalho está relacionado ao *Two-Dimensional Rectangular Multiple Bin Size Bin Packing Problem* (2D-R-MBSBPP), as subseções a seguir finalizam a fundamentação teórica, apresentando os principais modelos desenvolvidos, estratégias e algoritmos utilizados para resolver problemas dessa vizinhança. Elas servem como uma base de conhecimento sólida dos problemas bidimensionais guilhotinados em dois estágios, buscando consolidar os principais conceitos e preparar-se para a resolução do problema estudado na gráfica, já que a base de formulação e modelagem será a mesma, porém, aplicando premissas e restrições específicas ao estudo de caso.

2.4.1 Histórico de Abordagens

A seguir, serão apresentadas três abordagens desenvolvidas sobre o tema que formam uma base sólida para entender essa classe de problemas. A ideia é apresentar os principais trabalhos que dão origem à modelagem desenvolvida no Capítulo 4. Vale ressaltar que as três abordagens tratam apenas os problemas com orientação fixa (i.e., sem a possibilidade de rotação em 90°). Uma das contribuições do presente trabalho, que será observada no Capítulo 3, é a extensão para o caso não-orientado (que permite a rotação em 90° dos itens), que se assemelha à operação da gráfica estudada.

A primeira abordagem é o modelo $M1$ apresentado em LODI e MONACI (2003), um trabalho que trata sobre o *Two-Dimensional Knapsack Problem* (TDK), ou seja, o caso bidimensional do problema de maximização das saídas com itens fortemente heterogêneos e um único objeto. Essa abordagem já utiliza o conceito de níveis. Generalizando o TDK segundo LODI e MONACI (2003), temos um único objeto retangular S com largura W e altura H , e uma lista de itens retangulares m com diversos tipos de formato i que devem ser cortados a partir de S . Cada tipo de formato i ($i = 1, \dots, m$) é caracterizado por uma largura \bar{w}_i , altura \bar{h}_i , lucro \bar{p}_i e um limite superior \bar{u}_i indicando o número máximo de itens do tipo i que podem ser cortados. O objetivo do problema é determinar um padrão

de corte de S maximizando a soma dos lucros dos itens cortados.

No modelo $M1$ considera-se que todos os itens de um mesmo tipo são distintos; logo, para cada tipo de formato i ($i = 1, \dots, m$), define-se um limite superior ub_i com itens idênticos j de forma que $h_j = \bar{h}_i$, $w_j = \bar{w}_i$ e $p_j = \bar{p}_i$. Temos $n = \sum_{i=1}^m ub_i$ indicando o número total de itens, e consideramos que os itens estão ordenados de forma que $h_1 \geq h_2 \geq \dots \geq h_n$. O modelo assume que n potenciais níveis podem ser *inicializados* (ou seja, o objeto pode chegar a ter um item alocado em cada nível): o nível k , se utilizado, é inicializado pelo item k ($k = 1, \dots, n$). Desta forma, a seguinte variável binária pode ser definida para indicar os possíveis itens n cortados de um nível:

$$x_{jk} = \begin{cases} 1, & \text{se o item } j \text{ é cortado do nível } k \\ 0, & \text{caso contrário} \end{cases} \quad (k = 1, \dots, n; j = k, \dots, n) \quad (2.1)$$

Logo, o modelo $M1$ de LODI e MONACI (2003) pode ser representado como um modelo de Programação Linear Inteira para descrever o *Two-Dimensional Knapsack Problem*:

$$\max \quad \sum_{j=1}^n p_j \sum_{k=1}^j x_{jk} \quad (2.2)$$

$$\text{s.a.} \quad \sum_{k=1}^j x_{jk} \leq 1 \quad (j = 1, \dots, n) \quad (2.3)$$

$$\sum_{j=k+1}^n w_j x_{kj} \leq (W - w_k) x_{kk} \quad (k = 1, \dots, n-1) \quad (2.4)$$

$$\sum_{k=1}^n h_k x_{kk} \leq H \quad (i = 1, \dots, n) \quad (2.5)$$

$$x_{jk} \in \{0, 1\} \quad (k = 1, \dots, n; j = k, \dots, n) \quad (2.6)$$

A função objetivo (2.2) maximiza a soma dos lucros dos itens cortados. As restrições (2.3) garantem que cada item só é cortado no máximo uma única vez, e somente em níveis cuja altura seja no mínimo igual à altura do item. As restrições (2.4) garantem que a largura do nível não pode exceder a largura do objeto, e que ou o item k está no nível k , ou o nível k está vazio. As restrições (2.5) impõem que a soma das alturas dos níveis abertos não podem exceder a altura do objeto. Note que as variáveis x_{kk} ($k = 1, \dots, n$) possuem dois significados: $x_{kk} = 1$ implica que o item k é cortado do nível k ; logo, o nível k é utilizado e inicializado pelo item k correspondente. Por fim, as restrições (2.6) definem o domínio de x_{jk} .

A segunda abordagem é o modelo $M2$, ainda presente em LODI e MONACI (2003). A principal diferença em relação ao modelo $M1$ é o agrupamento de itens semelhantes. Nele, considera-se que os itens com o mesmo formato são agrupados; a separação ocorre somente nos itens que inicializam um nível. Portanto, é necessário definir um mapeamento entre os diversos tipos de formato i ($i = 1, \dots, m$), e dos potenciais níveis k ($k = 1, \dots, n$).

Observa-se, portanto, que qualquer item do tipo i pode ser cortado no intervalo de níveis $[1, \sum_{s=1}^i ub_s]$. Além disso, define-se $\alpha_i = \sum_{s=1}^i ub_s$ ($i = 1, \dots, m$), com $\alpha_0 = 0$. Por outro lado, qualquer nível k pode ser utilizado para obter itens cujo tipo está no intervalo $[\beta_k, m]$, com $\beta_k = \min \{r : 1 \leq r \leq m, \alpha_r \geq k\}$ ($k = 1, \dots, n$). Assim, β_k ($k = 1, \dots, n$) denota o tipo de formato que deve inicializar o nível k . Como assume-se que $h_1 \geq h_2 \geq \dots \geq h_n$, temos dois conjuntos separados de variáveis.

O primeiro conjunto é composto das seguintes variáveis inteiras (não-binárias), onde $i = 1, \dots, m$, $k \in [1, \alpha_i]$, e o termo *adicional* indica que o item do tipo i inicializando o nível k é considerado separadamente (se o nível corresponder a este tipo de item):

$$x_{ik} = \begin{cases} \text{número de itens do tipo } i \text{ cortados do nível } k & \text{se } i \neq \beta_k \\ \text{número de itens adicionais do tipo } i \text{ cortados do nível } k & \text{se } i = \beta_k \end{cases} \quad (2.7)$$

O segundo conjunto envolve as seguintes variáveis binárias:

$$q_k = \begin{cases} 1, & \text{se o nível } k \text{ é utilizado} \\ 0, & \text{caso contrário} \end{cases} \quad (k = 1, \dots, n) \quad (2.8)$$

Logo, o modelo $M2$ apresentado em LODI e MONACI (2003) pode ser representado como um modelo de Programação Linear Inteira para descrever o *Two-Dimensional Knapsack Problem*:

$$\max \quad \sum_{i=1}^m \bar{p}_i \left(\sum_{k=1}^{\alpha_i} x_{ik} + \sum_{k=\alpha_{i-1}+1}^{\alpha_i} q_k \right) \quad (2.9)$$

$$\text{s.a.} \quad \sum_{k=1}^{\alpha_i} x_{ik} + \sum_{k=\alpha_{i-1}+1}^{\alpha_i} q_k \leq ub_i \quad (i = 1, \dots, m) \quad (2.10)$$

$$\sum_{i=\beta_k}^m \bar{w}_i x_{ik} \leq (W - \bar{w}_{\beta_k}) q_k \quad (k = 1, \dots, n) \quad (2.11)$$

$$\sum_{k=1}^n h_{\beta_k} q_k \leq H \quad (i = 1, \dots, n) \quad (2.12)$$

$$\sum_{s=k}^{\alpha_i} x_{is} \leq ub_i - (k - \alpha_{i-1}) \quad (i = 1, \dots, m; k \in [\alpha_{i-1} + 1, \alpha_i]) \quad (2.13)$$

$$0 \leq x_{ik} \leq ub_i \quad (i = 1, \dots, m; k \in [1, \alpha_i]) \quad (2.14)$$

$$q_k \in \{0, 1\} \quad (k = 1, \dots, n) \quad (2.15)$$

A função objetivo (2.9) corresponde à (2.2) do modelo $M1$, porém, com a adição da nova variável de decisão. As restrições (2.10), (2.11) e (2.12) correspondem respectivamente à (2.3), (2.4) e (2.5) do modelo $M1$, com as devidas adaptações, garantindo que o item só é cortado no máximo uma vez (2.10), que a largura do objeto não é excedida (2.11) e que a altura do objeto não é excedida (2.12). O propósito das restrições (2.13), que são *redundantes* em termos de formulação da Programação Linear Inteira (já que podem ser omitidas sem alterar a solução viável (TELGEN, 1983)), é fortalecer o limite nas variáveis x_{ik} , dado pelas restrições (2.14) (LODI; MONACI, 2003). Ou seja, sem as restrições (2.13), parte da estrutura obtida no modelo $M1$ que considera separadamente os itens com a mesmo formato seria perdida no modelo $M2$. Por fim, as restrições (2.15) definem o domínio de q_k .

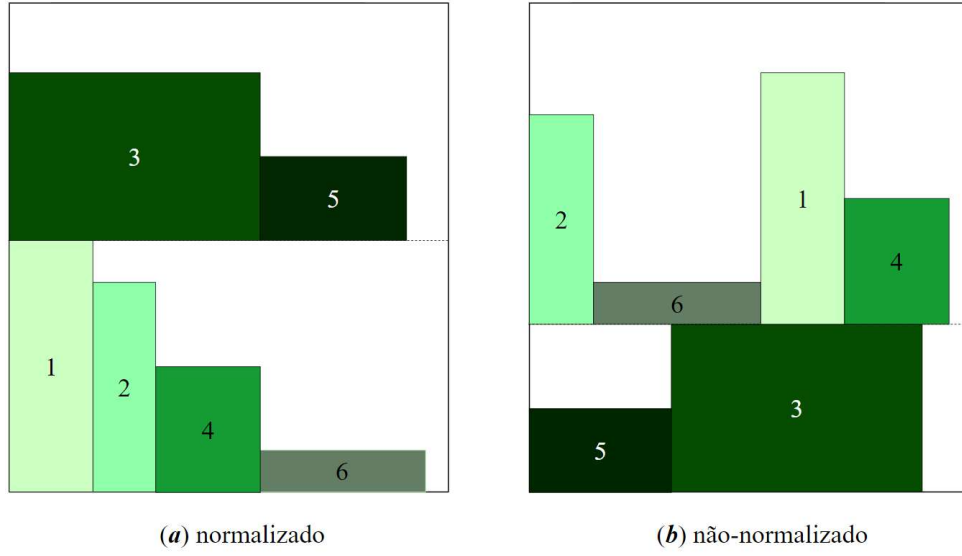
Por fim, a terceira e última abordagem envolve um modelo de Programação Linear Inteira para o empacotamento em níveis desenvolvido em LODI et al. (2004). Essa formulação tornou-se muito popular para resolver o *Bin Packing Problem*, já que descreve o caso bidimensional do problema de maximização das saídas com itens fortemente heterogêneos e com múltiplos objetos (diferentemente dos modelos $M1$ e $M2$ apresentados anteriormente, que tratam o caso de um único objeto).

Antes de detalhar essa abordagem, devemos assumir, sem perda de generalidade, que os empacotamentos nessa modelagem seguem as três premissas elencadas a seguir:

- (i) em cada nível, o item mais à esquerda é o que possui a maior altura.
- (ii) o primeiro nível (o mais baixo) de cada objeto é o nível com maior altura.
- (iii) os itens são ordenados de forma que $h_1 \geq h_2 \geq \dots \geq h_n$.

A esse empacotamento damos o nome de *normalizado*. A Figura 17 a seguir evidencia em (a) um empacotamento normalizado, enquanto em (b) apresenta um empacotamento por níveis sem essa condição. Os itens foram numerados em ordem decrescente de altura, ou seja, o item 1 é o que possui a maior altura, enquanto o item 6 é o que possui a menor altura.

Figura 17: Empacotamentos normalizado e não-normalizado.



Fonte: o Autor.

O modelo utiliza quatro conjuntos de variáveis: os dois primeiros se referem ao empacotamento dos itens dentro dos níveis; já os dois últimos se referem ao empacotamento dos níveis dentro dos objetos. A modelagem assume que há n potenciais níveis, cada um com um item i o inicializando, e portanto, cada nível com sua altura h_i . A seguir, temos a primeira variável de decisão.

$$y_i = \begin{cases} 1, & \text{se o item } i \text{ inicializa o nível } i \\ 0, & \text{caso contrário} \end{cases} \quad (i = 1, \dots, n) \quad (2.16)$$

Por conta de (i) e (iii) apresentados anteriormente, somente itens j que satisfazem $j > i$ podem ser empacotados no nível i (se este nível for realmente inicializado pelo item i). Assim, temos a segunda variável de decisão:

$$x_{ij} = \begin{cases} 1, & \text{se o item } j \text{ é empacotado dentro do nível } i \\ 0, & \text{caso contrário} \end{cases} \quad (i = 1, \dots, n-1; j > i) \quad (2.17)$$

De forma similar, assumimos que há n potenciais objetos, cada um com um potencial nível k que o inicializa. Assim, temos a terceira variável de decisão.

$$q_k = \begin{cases} 1, & \text{se o nível } k \text{ inicializa o objeto } k \\ 0, & \text{caso contrário} \end{cases} \quad (k = 1, \dots, n) \quad (2.18)$$

Por conta de (ii) e (iii) apresentados anteriormente, somente níveis i que satisfazem $i > k$ podem ser empacotados no objeto k (se este objeto for realmente inicializado pelo nível k). Assim, temos a quarta variável de decisão:

$$z_{ki} = \begin{cases} 1, & \text{se o nível } i \text{ é alocado ao objeto } k \\ 0, & \text{caso contrário} \end{cases} \quad (k = 1, \dots, n-1; i > k) \quad (2.19)$$

Dessa forma, o modelo completo de Programação Linear Inteira desenvolvido no trabalho de LODI et al. (2004) que descreve o *Bin Packing Problem* está detalhado a seguir:

$$\min \sum_{k=1}^n q_k \quad (2.20)$$

$$\text{s.a.} \quad \sum_{i=1}^{j-1} x_{ij} + y_j = 1 \quad (j = 1, \dots, n) \quad (2.21)$$

$$\sum_{j=i+1}^n w_j x_{ij} \leq (W - w_i) y_i \quad (j = 1, \dots, n-1) \quad (2.22)$$

$$\sum_{k=1}^{i-1} z_{ki} + q_i = y_i \quad (i = 1, \dots, n) \quad (2.23)$$

$$\sum_{i=k+1}^n h_i z_{ki} \leq (H - h_k) q_k \quad (k = 1, \dots, n-1) \quad (2.24)$$

$$y_i, x_{ij}, q_k, z_{ki} \in \{0, 1\} \quad \forall \quad i, j, k \quad (2.25)$$

A função objetivo (2.20) minimiza o número de objetos utilizados. As restrições (2.21) impõem que cada item é empacotado uma única vez. As restrições (2.22) impõem a limitação de largura para cada nível usado (a soma das larguras dos itens não pode ultrapassar a largura do objeto ao qual estão alocados). As restrições (2.23) impõem que cada nível usado é alocado para um único objeto. As restrições (2.24) impõem a limitação de altura para o objeto (a soma das alturas dos níveis não pode ultrapassar a altura do objeto ao qual estão alocados). Por fim, as restrições (2.25) definem o domínio das variáveis de decisão.

2.4.2 Estratégias e Algoritmos

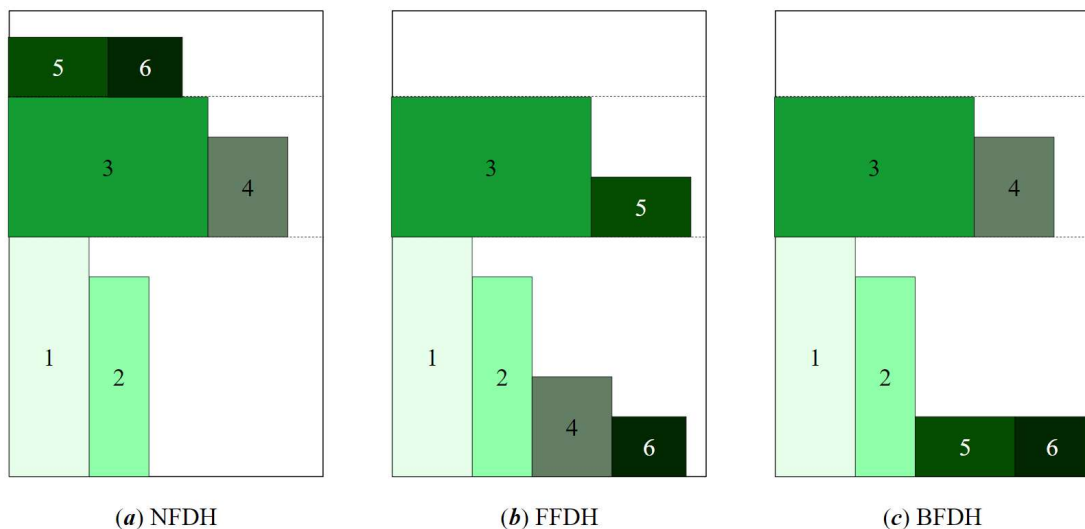
As primeiras estratégias desenvolvidas para solucionar os Problemas de Corte Empacotamento Bidimensional Guilhotinado surgiram a partir do caso unidimensional do *Bin Packing Problem*, já utilizando o conceito de níveis e do empacotamento normalizado. Apesar de serem estratégias que surgiram para resolver o caso unidimensional, elas

são a base (a primeira etapa) de resolução dos problemas bidimensionais do *Bin Packing Problem*.

Há três heurísticas clássicas que envolvem o empacotamento por níveis. A seguir, na Figura 18, as três estratégias estão ilustradas. Nos três casos, os itens são inicialmente ordenados em alturas decrescentes. Seja j o item atual e s o último nível criado, seguindo o trabalho de LODI (1999), temos:

- **Next-Fit Decreasing Height (NFDH)**: item j é empacotado justificado à esquerda no nível s , se nele couber. Caso contrário, um novo nível ($s = s + 1$) é criado, e j é empacotado justificado à esquerda nesse novo nível.
- **First-Fit Decreasing Height (FFDH)**: item j é empacotado justificado à esquerda no primeiro nível (de baixo para cima) onde ele couber, se houver. Se nenhum nível puder acomodar j , um novo é inicializado, assim como no NFDH.
- **Best-Fit Decreasing Height (BFDH)**: o item j é empacotado justificado à esquerda naquele nível em que dentre aqueles onde ele couber, resulte no menor espaço horizontal não utilizado. Se nenhum nível puder acomodar j , um novo é inicializado, assim como no NFDH.

Figura 18: Três estratégias clássicas (NFDH, FFDH e BFDH).



Fonte: adaptado de LODI (1999).

Em relação ao NFDH, é possível notar que, como o item 3 não cabia no primeiro nível (por conta de sua largura), um novo nível foi aberto para ele ser alocado. O mesmo

aconteceu com o item 5, que não cabia no segundo nível, e por isso um novo nível foi aberto. Ou seja, toda vez que um item não cabe em um nível, uma nova é aberto, e o próximo item é testado somente no nível atual, mas não nos anteriores.

Já em relação ao FFDH, é possível notar que, como o item 3 não cabia no primeiro nível (por conta de sua largura), um novo nível foi aberto para ele ser alocado, assim como no NFDH. Porém, o item 4 foi alocado no primeiro nível, já que ele cabia nesse nível. Ou seja, toda vez que um item não cabe em uma nível, um novo é aberto assim como no NFDH. Porém, o próximo item é testado em todos os níveis abertos (de baixo para cima), e é alocado no primeiro nível que couber.

Por fim, em relação ao BFDH, também é possível notar que um novo nível foi aberto para o item 3 por conta de sua largura, assim como nas duas heurísticas anteriores. Entretanto, o item 4 foi alocado no segunda nível (e não no primeiro como ocorreu no FFDH), já que o espaço horizontal remanescente ao ser alocado no segundo nível é menor do que no primeiro.

Além das três abordagens clássicas, há diversos outros algoritmos que são utilizados para resolver problemas relacionados: *Split-Fit* (SF) (COFFMAN et al., 1980), *Reverse-Fit* (RF) (SCHIERMEYER, 1994), *Algoritmo de Steinberg* (STEINBERG, 1997) e *Algoritmo de Sleator* (SLEATOR, 1980).

Outro conceito relevante em relação a caracterização das heurísticas é que elas podem ser divididas em duas famílias de algoritmos segundo LODI (1999):

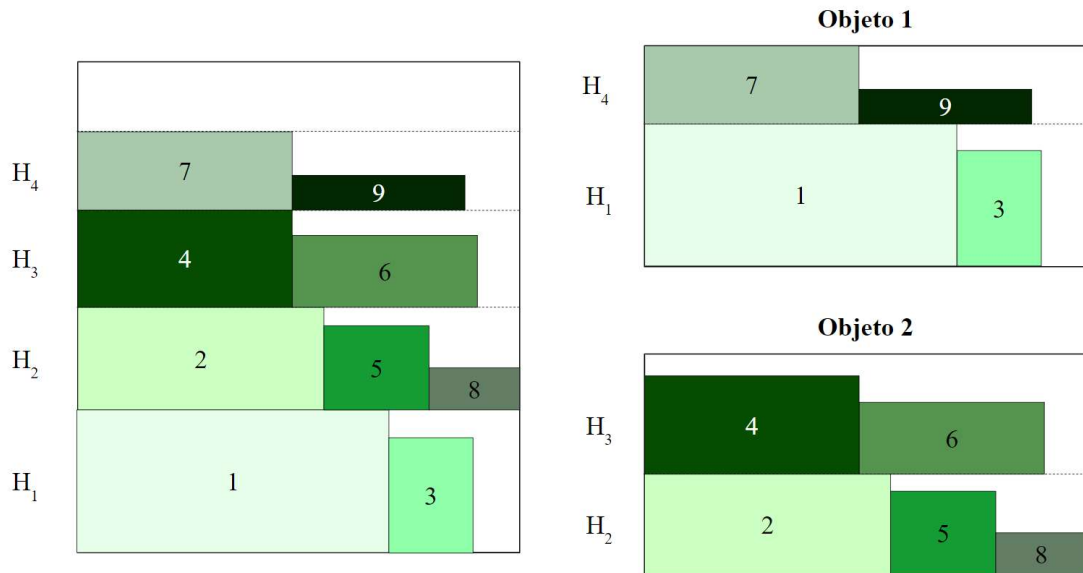
- **Algoritmos de duas fases:** começam empacotando os itens em um único nível de largura W . Na segunda fase, a solução por níveis (ou *strips*) é usada para construir um empacotamento em um número finito de objetos com dimensões $W \times H$, com H representando a altura.
- **Algoritmos de uma fase:** empacotam os itens diretamente em um número finito de objetos.

Em relação à família de algoritmos de duas fases, os principais são o *Hybrid First-Fit* (HFF), o *Finite Best-Strip* (FBS) e o *Hybrid Next-Fit* (HNF). Os três algoritmos são apresentados a seguir.

O HFF foi proposto em CHUNG et al. (1982). A primeira fase do HFF obtém um empacotamento de níveis por meio do FFDH. As alturas resultantes dos níveis após a aplicação do FFDH são decrescentes, ou seja, $H_1 \geq H_2 \geq \dots \geq H_n$. Então, basta resolver

um Bin Packing unidimensional por meio da heurística FFDH novamente: inicializar o objeto 1 para empacotar o nível 1, e, para um i crescente, empacotar o nível i atual no objeto com menor índice possível em que ela consiga ser empacotada; se não houver nenhum objeto capaz de empacotar o nível i , então, deve-se inicializar um novo objeto. O HFF está representado na Figura 19 a seguir. Nota-se que há 4 níveis, sendo que $H_1 \geq H_2 \geq H_3 \geq H_4$. Porém, como não é possível empacotar H_2 no mesmo objeto que H_1 , inicializou-se outro objeto. Assim como H_2 , não era possível alocar o nível com altura H_3 no primeiro objeto, somente no objeto 2; porém, foi possível alocar o nível com altura H_4 no objeto 1.

Figura 19: Algoritmo HFF.



Fonte: adaptado de LODI (1999).

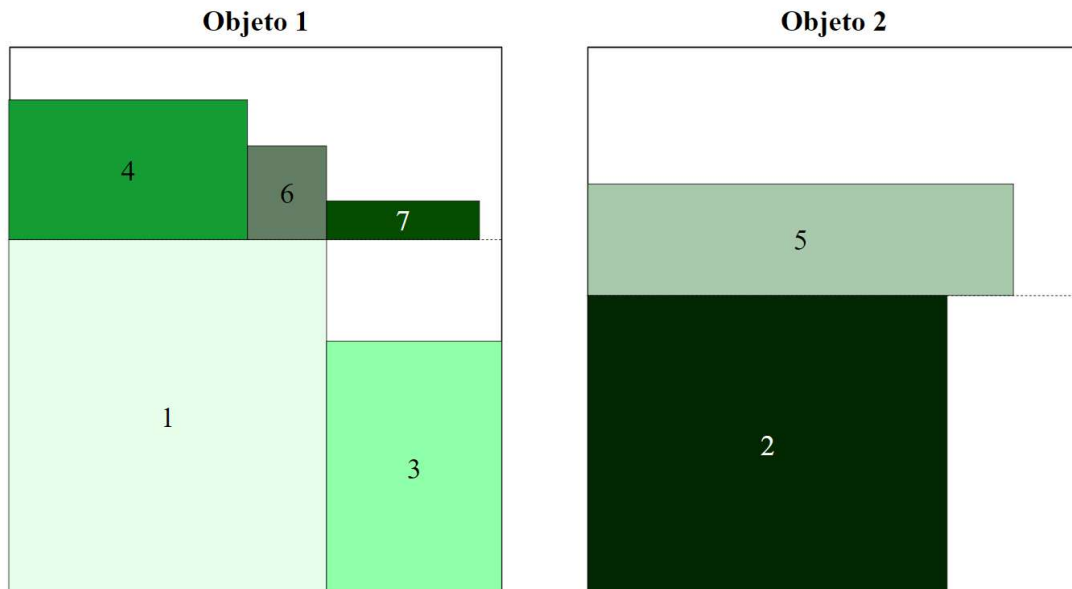
O FBS é uma variação do HFF, e foi proposto em BERKEY e WANG (1987). Na primeira fase, aplica-se o BFDH. Na segunda fase, basta resolver um *Bin Packing* unidimensional por meio da heurística BFDH novamente: empacotar o nível atual no objeto cuja altura remanescente fosse mínima; se não houver nenhum objeto capaz de empacotar o nível atual, então, deve-se inicializar um novo objeto.

Por fim, temos o HNF, que é outra variação do HFF. Na primeira fase, adota-se o NFDH. Na segunda fase, basta resolver um *Bin Packing* unidimensional considerando o NFDH novamente: alocar o nível atual no objeto atual (já aberto), se couber, ou inicializar um objeto novo caso contrário. Devido à característica *next-fit*, os novos níveis nunca serão testados em objetos anteriores. Ou seja, caso o nível não caiba no objeto

atual, um objeto novo será aberto, e os próximos níveis não serão testados nos objetos previamente abertos.

Finalmente, em relação à família de uma fase, o principal algoritmo é o *Finite First-Fit* (FFF). Ele foi desenvolvido e testado em BERKEY e WANG (1987). O FFF adota a mesma estratégia do algoritmo FFDH apresentado anteriormente. O item atual é empacotado no nível mais baixo do primeiro objeto onde é possível empacotá-lo; se nenhum nível puder acomodá-lo, então, há duas opções: um novo nível capaz de acomodá-lo é criado em algum objeto que comporte esse novo nível ou um novo objeto é inicializado (se nenhum dos objetos disponíveis possuir o espaço vertical necessário para acomodar esse novo nível). O FFF está representado a seguir na Figura 20.

Figura 20: Algoritmo FFF.



Fonte: adaptado de PIETROBUONI (2015).

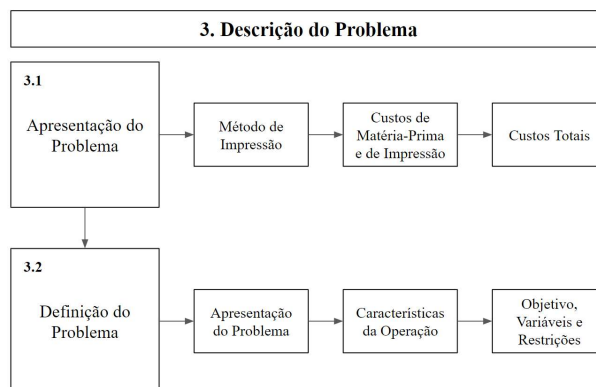
É possível notar que o item 2 não cabia no objeto 1, e, por isso, um novo objeto foi inicializado. Porém, o item 3 ainda cabia no objeto 1, assim como os itens 4, 6 e 7 que foram alocados ao objeto 1, já que ele é o primeiro (o objeto com menor índice) no qual eles podem ser alocados.

3 DESCRIÇÃO DO PROBLEMA

O desafio que o presente trabalho busca resolver, como já mencionado no Capítulo 1, é a montagem dos Planos de Impressão e Corte para uma pequena gráfica que atualmente não possui nenhuma ferramenta para auxiliá-la nesse processo. O principal objetivo é a redução de custos na produção de um pedido, que é estritamente dependente da quantidade de folhas utilizadas na execução da ordem de serviço. Por isso, o intuito deste capítulo é apresentar o problema inserido no contexto da gráfica, esquematizando os conceitos importantes e listando as principais características relevantes para a modelagem.

Em relação à organização do capítulo, ele está dividido em duas partes: apresentação do problema e definição do objetivo, das variáveis e das restrições. Na primeira parte, descreve-se o método de impressão utilizado na gráfica (que interfere diretamente nos custos e materiais utilizados), as especificações das folhas disponíveis para impressão e uma análise sobre os custos totais. Já na segunda parte, o foco está no processo de montagem do Plano e na execução dos pedidos, para que, então, o objetivo, as variáveis e as restrições sejam definidas. A estrutura do capítulo pode ser observada na Figura 21 a seguir.

Figura 21: Estrutura do Capítulo 3.



Fonte: o Autor.

3.1 Apresentação do Problema

No setor gráfico existem duas formas principais de impressão de uma imagem: *offset* ou digital. A gráfica estudada utiliza a impressão digital para executar os pedidos dos seus clientes. Ela é adequada para pequenas/médias tiragens, por possuir alta capacidade de personalização em relação à *offset* (já que a digital, p.e., imprime arquivos com dados variáveis ao mesmo custo; enquanto a *offset* implica em custos adicionais para cada alteração feita no arquivo a ser impresso). Além disso, em relação ao fator econômico, o custo de *setup* da digital é muito mais baixo que o da *offset*, tanto em termos financeiros quanto em termos de tempo utilizado para preparação. A Tabela 1 a seguir compara as duas formas principais de impressão, *offset* e digital, com base em 5 indicadores: quantidade, qualidade, tempo, custo e personalização:

Tabela 1: Comparação entre impressão *offset* e digital.

Indicador	Digital	Offset
Quantidade	ideal para pequenas ou médias tiragens	ideal para grandes tiragens
Qualidade	alta qualidade, porém limitada à combinação das 4 cores padrão, que adotam a sigla CMYK: ciano (C), magenta (M), amarelo (Y) e preto (K)	alta qualidade, com a possibilidade de inclusão de cores personalizadas
Tempo	tempo de <i>setup</i> muito baixo	tempo de <i>setup</i> elevado
Custo	custo sempre padronizado, independentemente da quantidade	baseado em escala, ou seja, o custo/unidade cai à medida que o pedido é maior, tornando-se mais barato que a digital
Personalização	ilimitada, sem custos adicionais	custos adicionais à cada alteração feita no arquivo

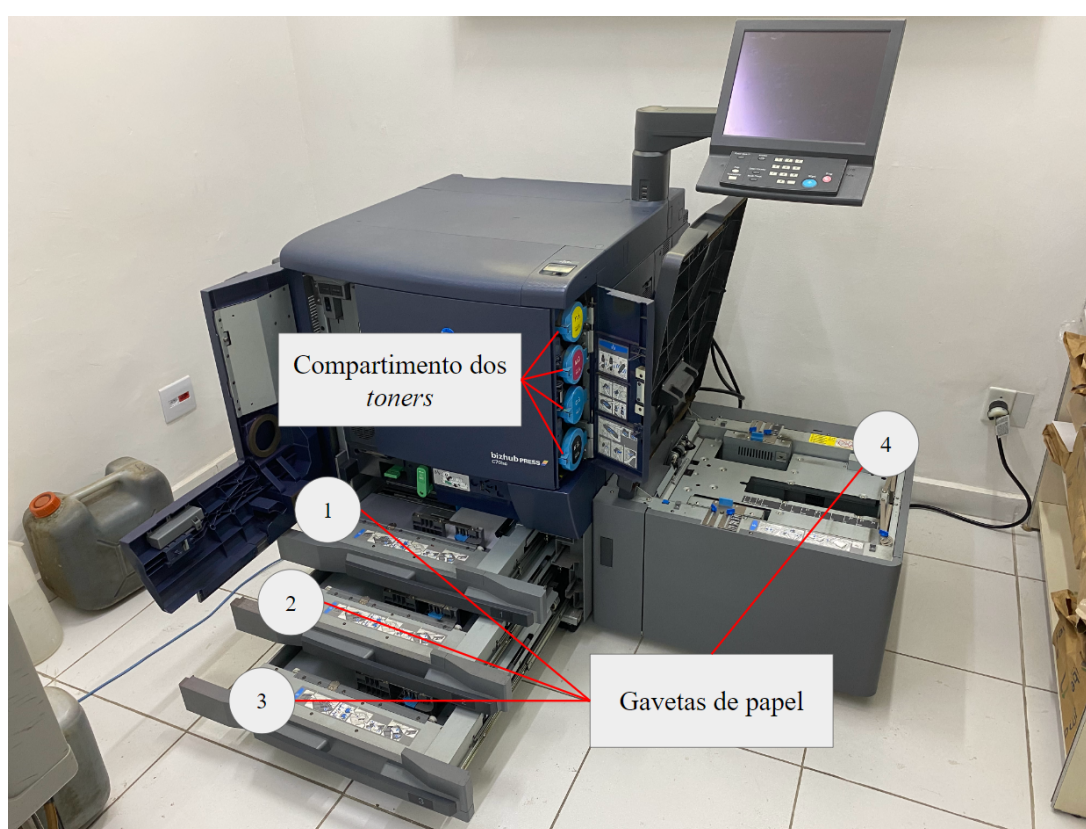
Fonte: o Autor.

A impressão digital, por sua vez, também possui duas formas de ser executada: por impressoras a jato de tinta, ou por impressoras a *laser*. A gráfica utiliza uma impressora

a *laser* para rodar os pedidos, por conta de sua maior velocidade de impressão e maior qualidade das imagens impressas quando comparada com impressoras a jato de tinta para essa finalidade gráfica.

Na Figura 22 a seguir encontra-se o equipamento utilizado pela empresa para a realização das impressões dos pedidos. É possível observar duas partes principais: as gavetas de papel (que armazenam as folhas utilizadas para imprimir os arquivos) e o compartimento dos *toners*, que são os recipientes que armazenam a tinta em pó utilizada em impressoras a *laser*.

Figura 22: Impressora utilizada pela gráfica.



Fonte: o Autor.

Dado que a impressora possui gavetas, o papel utilizado para imprimir um arquivo já está cortado (e não está em rolo, como seria na impressão *offset*). As gavetas possuem delimitadores que cercam as bordas do papel utilizado, como observado na Figura 23 a seguir. Vale ressaltar que as gavetas estão delimitadas por um tamanho máximo que a estrutura comporta, ou seja, elas não aceitam qualquer tamanho de papel. No caso da impressora utilizada, as primeiras três gavetas aceitam uma folha com, no máximo, a dimensão 33 cm \times 48 cm. A gaveta 4, por poder trabalhar aberta, está limitada apenas pela largura do papel, que deve ser de no máximo 33 cm.

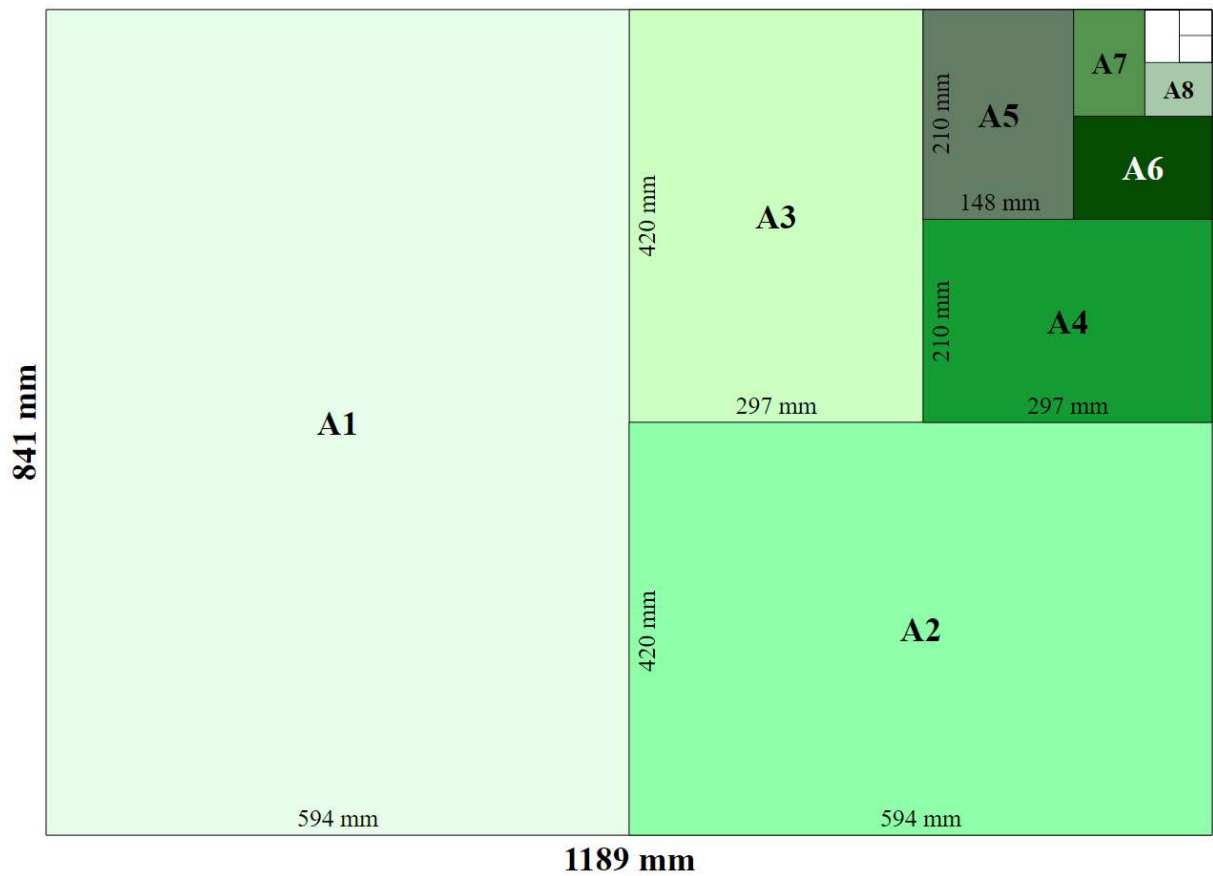
Figura 23: Gaveta 1 da impressora.



Fonte: o Autor.

Quando a folha utilizada já possui um padrão em suas dimensões, dá-se o nome de *Formato*. Os Formatos normalmente surgem a partir de uma folha *matriz*, ou seja, uma folha suficientemente grande que é cortada em n partes, formando n objetos menores com dimensões padronizadas. Os Formatos mais conhecidos são o A, p.e. o Formato A4, muito popular em diversos segmentos. A Figura 24 a seguir ilustra os diversos Formatos que são derivados da matriz A0, ou seja, a folha matriz do formato A, que possui dimensões 1189 mm \times 841 mm. Nota-se que os Formatos são originados a partir da divisão do Formato anterior pela metade. Ou seja, uma folha matriz A0, se dividida ao meio, gera duas folhas com Formato A1; esta, por sua vez, se dividida ao meio, gera duas folhas com Formato A2, e assim sucessivamente.

Figura 24: Formatos A a partir da folha A0.

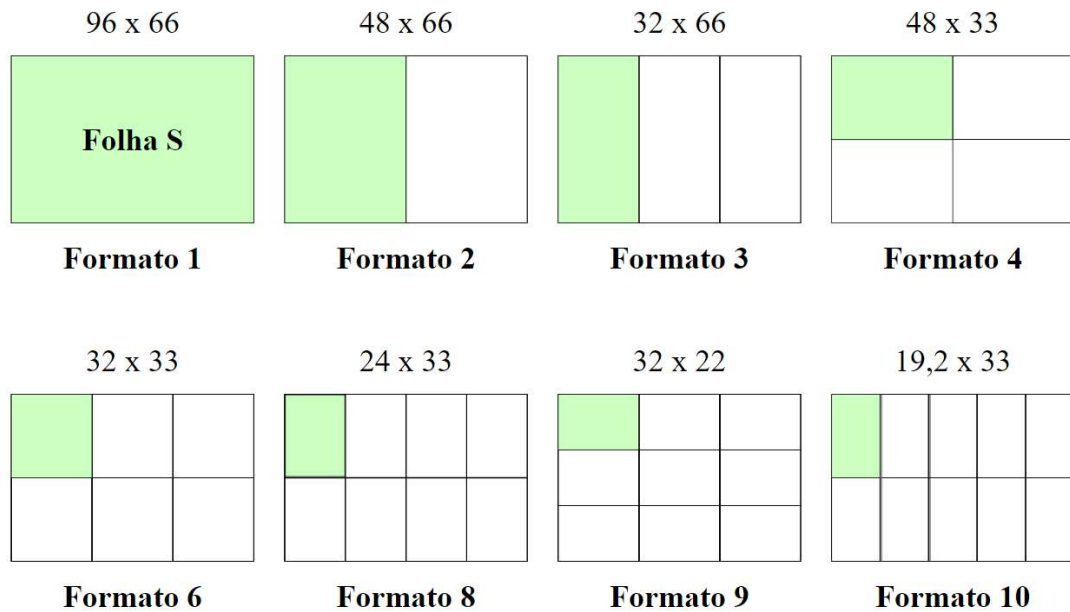


Fonte: o Autor.

Porém, os Formatos mais utilizados pela empresa não derivam da matriz A0, mas sim de outras duas folhas matrizes com as quais os seus fornecedores trabalham: a folha com dimensões 96 cm \times 66 cm (a qual chamaremos de *folha S*), ou a folha 112 cm \times 76 cm (a qual chamaremos de *folha G*). Ambas matrizes podem possuir diversos Formatos, assim como os Formatos A apresentados anteriormente.

A Figura 25 a seguir ilustra os formatos mais comuns da matriz S. A mesma lógica pode ser aplicada para a matriz G, porém, as dimensões seriam diferentes. O número que sucede o Formato indica em quantas partes aquela folha matriz foi cortada para obter a dimensão do Formato (p.e.: Formato 4 indica que a matriz *S* foi cortada em 4 partes para obter aquelas dimensões).

Figura 25: Formatos a partir da matriz S.



Fonte: o Autor.

Como mencionado anteriormente, a largura máxima que o equipamento da gráfica é capaz de comportar é 33 cm. Por isso, a empresa precisa utilizar os Formatos menores que são derivados das folhas matrizes (p.e., para a matriz S, a máquina é capaz de comportar folhas do Formato 3 em diante).

Sendo assim, a modelagem deve considerar os principais formatos que a empresa utiliza para montar os Planos de Impressão e Corte. Na Tabela 2 a seguir estão os 5 formatos mais utilizados pela gráfica, com suas dimensões (Largura \times Altura) em centímetros. A letra S ou G foi incluída após o nome do Formato para a identificação de sua folha matriz.

Tabela 2: Formatos das folhas mais utilizadas pela empresa.

Formato	Dimensões (L \times A)
3 S	32 \times 66
4 S	48 \times 33
6 S	32 \times 33
8 G	28 \times 38
9 S	32 \times 22

Fonte: o Autor.

Em relação aos custos totais para rodar um pedido, eles podem ser divididos em duas partes: custo de matéria-prima e custo de impressão. Ambas serão abordadas a seguir, com a Tabela 3 compilando os custos para cada Formato.

O custo de matéria-prima está associado a quanto a empresa gasta por folha (de acordo com o Formato pedido) em seus fornecedores. Para todos os formatos listados na Tabela 3, considerou-se a mesma gramatura do papel (170 g) e o mesmo tipo de papel (couché), para fins de padronização. Esta é uma consideração razoável na prática, já que essa é a combinação de especificações mais utilizada para impressão dos pedidos na empresa. De qualquer forma, o problema pode ser transposto a outras gramaturas e tipos, já que as dimensões do papel permanecem as mesmas; o que irá variar é o custo por folha, a depender do material escolhido.

Já o custo de impressão está relacionado à quanto a empresa gasta por folha (de acordo com o Formato utilizado) para realizar a impressão de um lado do papel (uma página). Nessa gráfica, esse custo é padronizado por uma unidade de medida que será apresentada a seguir.

Na Tabela 3 a seguir estão consolidados os custos de matéria-prima e os custos de impressão para cada um dos Formatos mais utilizados. Para a modelagem matemática do problema, é interessante que os custos de cada Formato estejam expressos pela métrica de custo por unidade de área ao invés do custo por folha. Essa métrica foi calculada dividindo o custo total por folha pela área (em m^2) de cada Formato.

Tabela 3: Custos de matéria-prima e de impressão para os Formatos mais utilizados.

Formato	Matéria-prima	Impressão	Custo Total	
#	R\$/Folha	R\$/Folha	R\$/Folha	R\$/ m^2
3 S	0,48	1,60	2,08	9,85
4 S	0,36	0,80	1,16	7,32
6 S	0,24	0,40	0,64	6,06
8 G	0,20	0,40	0,60	5,64
9 S	0,16	0,40	0,56	7,95

Fonte: o Autor.

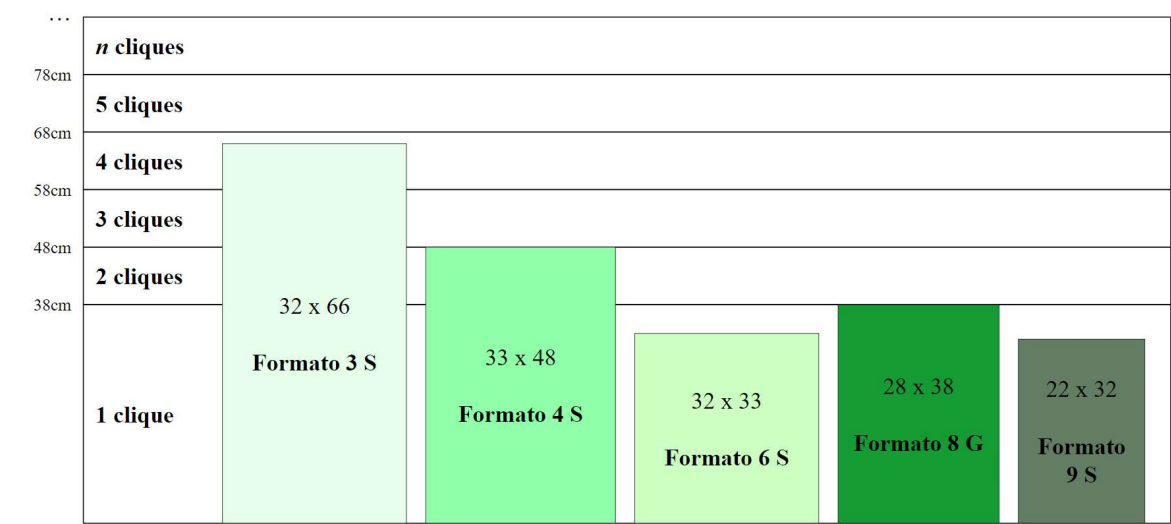
A unidade de medida que padroniza o custo de impressão é o *cliq*ue, conceito amplamente utilizado no setor gráfico. Ele é uma unidade de medida de área de impressão,

que determina a quantidade de vezes que o sistema de gravação da imagem (um cinto, ou *belt*, de impressão) necessita ser acionado para cobrir toda a extensão da folha, independentemente da área ocupada pelo(s) arquivo(s) impresso(s). Ou seja, o custo é o mesmo se a folha possuir apenas um pequeno arquivo, ou se estiver repleta de arquivos; o que importa, de fato, é quantas vezes esse sistema de gravação é acionado no processo de impressão, que é determinado pela altura do Formato utilizado (a largura do Formato não é relevante para definir o custo).

Logo, cada Formato exige um número de *cliques* diferente, pois folhas com maior altura necessitam de mais rotações desse cinto de impressão para serem cobertas por completo. Nessa gráfica, o custo do clique é padronizado em R\$ 0,40. Ou seja, para chegar no custo de impressão final, basta multiplicar o custo do clique pela quantidade de cliques que cada Formato exige.

A Figura 26 a seguir ilustra como a relação de cliques para os formatos mais utilizados é calculada para o equipamento que a empresa possui. Basicamente, o primeiro clique é contabilizado até a altura de 38 cm. Depois dele, os próximos cliques são contabilizados a cada avanço de 10 cm (ou seja, o segundo clique é contabilizado na altura 48 cm, o terceiro na altura 58 cm, e assim por diante). Por isso a relação de cliques para os formatos mais utilizados é: **4** (Formato 3 S), **2** (Formato 4 S), **1** (Formato 6 S), **1** (Formato 8 G), **1** (Formato 9 S).

Figura 26: Cálculo dos cliques para os formatos mais utilizados.



Fonte: o Autor.

Esse custo padronizado de R\$ 0,40 está definido em um contrato entre a empresa e a fabricante do equipamento. Na prática, a cada página impressa, o equipamento

de impressão possui um sistema de contagem de cliques, que é acessado remotamente pela fabricante. Mensalmente, uma cobrança é realizada pela fabricante com base na quantidade de *cliques* do período. Nesse valor estão inclusas a tinta (que é reposta à medida que a gráfica realiza o pedido), as peças de reposição e as manutenções corretiva e preventiva da máquina. Ou seja, a empresa é dona do equipamento (não é um contrato de *leasing*) e paga uma taxa mensal proporcional à utilização para manter a máquina operando.

3.2 Definição do Problema

Dada a extensa Fundamentação Teórica apresentada no Capítulo 2, esta seção tem como objetivo avaliar as características do problema na prática, traçando um paralelo com os conceitos apresentados. Para fins de padronização, define-se que a nomenclatura dos Problemas de Corte e Empacotamento apresentada no Capítulo 2 será utilizada: os Formatos (3 S, 4 S, 6 S, 8 G e 9 S) serão denominados *objetos*, os arquivos enviados pelos clientes (as imagens para impressão) serão denominados *itens* e as áreas inutilizadas na impressão serão denominadas *sobras*.

Atualmente, a montagem do Plano de Impressão e Corte é feita manualmente por um colaborador, que deve levar em consideração diversos fatores que serão abordados a seguir: padrão de corte, orientação/posicionamento dos itens e geração das sobras. O principal objetivo deste colaborador é selecionar o conjunto de objetos que apresentem o menor custo para a operação, conseguindo alocar todos os itens que precisam ser impressos. Portanto, estamos falando de um Plano bidimensional com itens e objetos retangulares.

Além disso, os itens a serem empacotados são muito heterogêneos entre os pedidos (já que podem assumir qualquer dimensão), o que gera uma complexidade ainda maior na montagem, pois, a cada pedido, um novo Plano precisa ser elaborado sem seguir um padrão pré-estabelecido. Os objetos, todavia, não apresentam essa heterogeneidade, já que são apenas 5 Formatos padronizados.

Ou seja, seguindo a tipologia de WÄSCHER et al. (2007), vemos que o problema é um 2D-R-MBSBPP, já que é caso bidimensional com itens retangulares do *Multiple Bin Size Bin Packing Problem*, por ser um problema de minimização das entradas (dos custos), com um sortimento fortemente heterogêneo de itens, mas fracamente heterogêneo de objetos. A seguir, veremos outras características relevantes do problema na prática, que ajudarão a definir as principais variáveis e restrições da modelagem.

Como a montagem do Plano de Impressão e Corte é feita manualmente, percebe-se que há espaço para otimizar esse problema complexo na operação da empresa, com o intuito de reduzir os custos operacionais de maneira consistente. A Figura 27 a seguir, originada de uma demanda real da gráfica (a qual necessitava imprimir 60 itens), evidencia que uma simples alteração na rotação de um item permite que um aproveitamento melhor seja obtido, consequentemente utilizando menos objetos, resultando em um custo menor para executar o pedido. Para os mesmos tamanhos de objeto e item, o cenário à esquerda alocou apenas 3 itens, enquanto o da direita conseguiu alocar 4 itens. Nesse pedido de 60 itens, o cenário à esquerda utilizaria 20 objetos, enquanto o cenário à direita utilizaria apenas 15.

Figura 27: Exemplo de como a montagem do Plano influencia na quantidade de folhas utilizadas.



Fonte: o Autor.

Além disso, na gráfica estudada todos os cortes em papel são guilhotinados (Figura 10) e os itens são empacotados ortogonalmente (Figura 9), por dois principais motivos: precisão nos cortes e agilidade. Precisão porque esse padrão permite que os cortes sejam realizados de uma ponta a outra da folha sem interrupções, e, portanto, sem correr o risco de cortar um arquivo impresso que estaria na linha do corte. Agilidade porque

a empresa utiliza uma guilhotina elétrica, que está representada na Figura 28 a seguir. Este equipamento permite que os cortes sejam muito mais ágeis do que uma refiladora, a outra opção disponível no setor gráfico (que, dependendo do modelo, consegue lidar de certa forma com os cortes não-guilhotinados). Com os itens posicionados ortogonalmente seguindo o padrão guilhotinado, basta alinhar a base da Folha e definir a altura desejada do corte.

Figura 28: Guilhotina elétrica utilizada pela empresa.



Fonte: o Autor.

Outra condição importante por conta da produtividade nos cortes é o empacotamento dos itens em níveis, respeitando os cortes em no máximo 2 estágios (Figura 11), com a retirada de sobras (*trimming*) permitida. Ou seja, este é um caso não-exato dos Problemas de Corte e Empacotamento Bidimensional. O número de níveis também é uma preocupação relevante durante a montagem do Plano: é interessante concentrar os itens no menor número de níveis possível, já que isso significa uma economia de tempo no momento dos cortes.

Em relação à orientação dos arquivos dentro da folha de impressão, eles podem assumir as orientações tanto *paisagem* quanto *retrato*. No setor gráfico, estes são os termos mais utilizados para descrever a orientação de um item: paisagem é quando a largura do item é maior que a altura; retrato é quando a altura é maior do que a largura. Ou seja, estamos tratando de um padrão não-orientado (que permite a rotação dos itens em 90°), como já visto na Figura 13 anteriormente.

No que se refere às sobras geradas pelos cortes dessas folhas, elas não são aproveitáveis para a gráfica por dois principais motivos: processo trabalhoso e incertezas nas demandas. Em relação ao processo, trabalhar com sobras não é uma tarefa trivial, pois isso exige diversas preparações extras na máquina de impressão, que dificultam o processo e consomem mais tempo da produção. Esses ajustes são tanto físicos (na bandeja de impressão), quanto digitais (configurações no *software* de impressão). Já em relação aos pedidos, eles variam bastante entre as ordens de impressão. Por isso, a definição das dimensões de uma sobra aproveitável seria muito imprecisa. Além disso, não haveria garantia de que ela seria utilizada, gerando um estoque improdutivo na empresa.

Por fim, sabe-se que na operação atual a montagem dos Planos de Impressão e Corte é feita o mais rapidamente possível, por conta dos prazos curtos de entrega. Ou seja, não é possível estender a janela de tempo da montagem dos Planos por um longo período, apenas em algumas horas (o que já é suficiente para agregar alguns pedidos para a produção).

Dessa forma, o objetivo da modelagem, assim como os principais parâmetros, variáveis e restrições relevantes são apresentados a seguir.

Objetivo da Modelagem:

- Alocar os itens no conjunto de objetos que apresentar o menor custo total, priorizando as soluções com o menor número de níveis possível.

Parâmetros do Problema:

- Quantidade de itens e de objetos.
- Dimensões dos itens e dos objetos.
- Custo/área dos objetos.
- Orientação dos itens.

Variáveis do Problema:

- Indicar se um objeto é utilizado (ou não).
- Indicar se um item utilizado foi rotacionado (ou não).

Restrições do Problema:

- Os itens alocados em um objeto não podem ultrapassar o limite de altura do mesmo.
- Os itens alocados em um objeto não podem ultrapassar o limite de largura do mesmo.
- Os itens não podem se sobrepor.
- A demanda dos itens deve ser atendida.
- Um item não pode ser cortado mais de uma vez.
- A quantidade de objetos disponíveis deve ser respeitada (a quantidade de cada Formato é finita, sujeita a disponibilidade na gráfica).

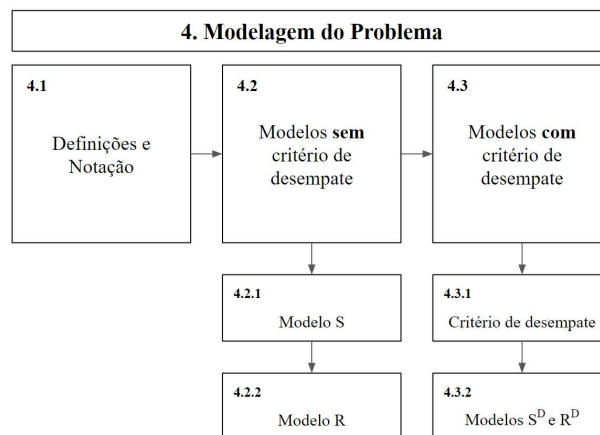
4 MODELAGEM DO PROBLEMA

Neste capítulo são apresentados os modelos matemáticos desenvolvidos para auxiliar a gráfica na montagem dos Planos de Impressão e Corte. Os modelos são extensões diretas dos modelos apresentados em ANDRADE et al. (2016), que, por sua vez, são fortemente inspirados no trabalho desenvolvido em LODI e MONACI (2003), que trata sobre um *Knapsack Problem* bidimensional com corte em dois estágios.

As principais contribuições da presente modelagem proposta foram a inclusão da possibilidade de rotação dos itens para simular as condições reais da gráfica e a inclusão de um critério de desempate das soluções ótimas para gerar o menor número de níveis (e, portanto, cortes) possível, impactando no tempo de execução das ordens de serviço.

Em relação à organização do capítulo, ele está dividido em três partes principais: a apresentação das definições e notação, o desenvolvimento dos modelos sem o critério de desempate (sem e com rotação dos itens) e o desenvolvimento dos modelos com o critério de desempate (sem e com rotação dos itens). A estrutura simplificada do capítulo pode ser observada na Figura 29 a seguir.

Figura 29: Estrutura do Capítulo 4.



Fonte: o Autor.

4.1 Definições e Notação

Inicialmente, vale lembrar que existe uma nomenclatura padrão de alguns termos sempre presentes nesse tipo de problema. Os *itens* são os corpos pequenos, que precisam ser empacotados/cortados. Já os *objetos* são os corpos grandes, nos quais os itens pequenos serão alocados. No caso do problema estudado, os itens são os arquivos enviados pelos clientes da gráfica, que devem ser alocados no conjunto de objetos que a gráfica possui, ou seja, os cinco formatos disponíveis de folhas (conforme Tabela 2).

Consideram-se p objetos retangulares, onde cada objeto l possui uma largura W_l e altura H_l ; além disso, como já visto na Tabela 3, cada objeto possui um custo/área c_l ($l = 1, \dots, p$) definido. Em relação aos itens, considera-se n itens retangulares, onde cada item i possui largura w_i e altura h_i ($i = 1, \dots, n$). Os itens devem ser alocados seguindo o padrão guilhotinado de corte.

Sem perda de generalidade, assume-se que $h_1 \geq h_2 \geq \dots \geq h_n$. Além disso, assim como em ANDRADE et al. (2016), assume-se que os cortes nos objetos são infinitamente finos; caso contrário, considera-se que a espessura da lâmina de corte já foi adicionada às dimensões dos objetos e itens, sem perda de generalidade (GILMORE; GOMORY, 1965), (MORABITO; ARENALES, 2000). Por fim, assume-se que todas as dimensões de itens e objetos e as unidades de custo/área são números inteiros. Isso não é uma suposição muito restritiva para lidar com as instâncias do problema na prática, porque a precisão finita das ferramentas de corte e medição (no máximo em milímetros), bem como a precisão finita usada no Real (no máximo em centavos), a moeda considerada para definir os custos dos objetos, significa que as suposições podem ser facilmente satisfeitas por uma mudança de escala.

Em ambos modelos, sem e com critério de desempate, utiliza-se a variável binária u_l ($l = 1, \dots, p$), que indica se o objeto l é utilizado ou não:

$$u_l = \begin{cases} 1, & \text{se o objeto } l \text{ é usado} \\ 0, & \text{caso contrário} \end{cases} \quad (4.1)$$

Além de u_l , em ambos modelos, utiliza-se outra variável binária, a x_{ikl} ($k = 1, \dots, n$, $i = k, \dots, n$, $l = 1, \dots, p$), que indica se o item i não-rotacionado é alocado (ou não) no nível k do objeto l :

$$x_{ikl} = \begin{cases} 1, & \text{se o item } i \text{ não-rotacionado está no nível } k \text{ do objeto } l \\ 0, & \text{caso contrário} \end{cases} \quad (4.2)$$

O item i não-rotacionado indica que ele será alocado da forma como ele for declarado para o modelo, ou seja, não haverá a possibilidade de rotação.

Porém, nos modelo com rotações, utiliza-se também a variável binária y_{ikl} ($k = 1, \dots, n$, $i = k, \dots, n$, $l = 1, \dots, p$), que indica se o item i rotacionado é alocado (ou não) no nível k do objeto l :

$$y_{ikl} = \begin{cases} 1, & \text{se o item } i \text{ rotacionado está no nível } k \text{ do objeto } l \\ 0, & \text{caso contrário} \end{cases} \quad (4.3)$$

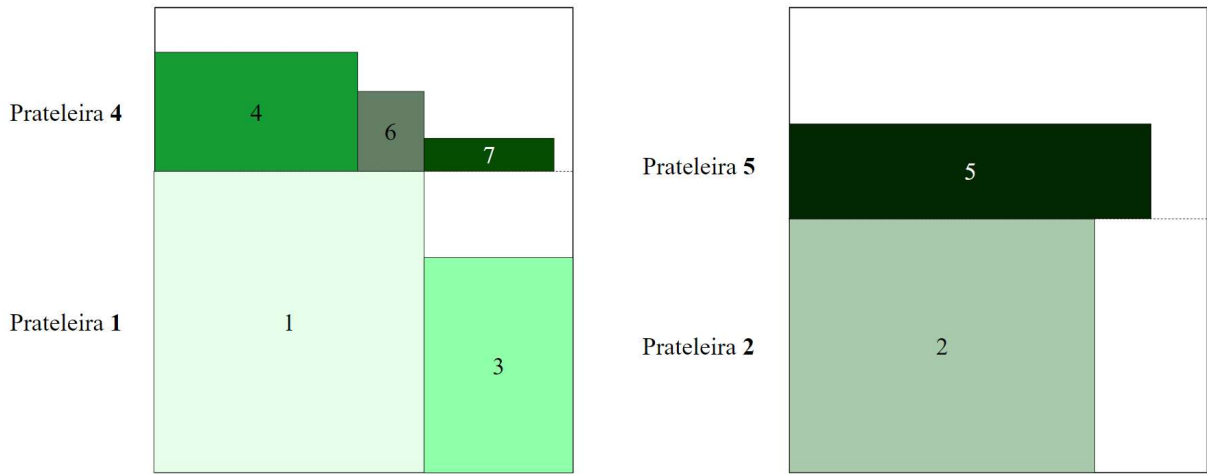
O item i rotacionado indica que ele será alocado com uma rotação de 90° em relação à maneira como foi declarado para o modelo, na qual a sua dimensão ao longo da largura do objeto passa a ser a sua altura, e vice-versa (ou seja, a sua dimensão ao longo da altura do objeto passa a ser a sua largura): p.e., se temos um item i que foi declarado com largura de 15 cm e altura de 10 cm, no cenário em que ele é rotacionado a sua altura passa a ser 15 cm e a sua largura passa a ser 10 cm.

A solução adotada para respeitar o padrão de cortes guilhotinados foi a adoção de níveis. Seguindo a terminologia utilizada em LODI e MONACI (2003) que foi apresentada no Capítulo 2, dado um objeto l , um nível é definida como uma faixa do objeto com largura W_l e altura igual à altura do item mais alto alocado nela.

Nos modelos propostos, considera-se que pode haver no máximo n níveis, cada um definido por um item i , com sua parte inferior sempre no chão do nível. O índice de cada nível é definido como o índice do primeiro item alocado nele, como pode ser observado na Figura 30 a seguir.

Ou seja, um nível k está aberto (ou utilizado) se o item k for o item de menor índice alocado nele. Nesse caso, todo nível aberto é representada por $x_{kkl} = 1$ ou $y_{kkl} = 1$ (dependendo se o item está rotacionado ou não), já que, nesse caso, o índice do item que abriu o nível torna-se o próprio índice do nível.

Figura 30: Representação dos índices de níveis abertos.



Fonte: o Autor.

Por fim, vale ressaltar que qualquer padrão ótimo de corte em dois estágios tem uma solução equivalente na qual o item de maior altura em cada nível é o primeiro item colocado à esquerda desse nível. Ambos modelos seguem essa lógica, sempre alocando o item de maior altura do nível à esquerda. Dessa forma, os cortes sempre respeitam o padrão de corte guilhotinado em dois estágios (mais o *trimming* se for necessário).

O último conceito abordado em ambos modelos é o agrupamento de itens em tipos, introduzido por ANDRADE et al. (2016). Um tipo é definido pela combinação das dimensões de largura e altura dos itens: toda vez que uma nova combinação é identificada, surge um novo tipo. Em outras palavras, se houver 10 itens para serem alocados, cada um com dimensões diferentes entre si, teremos 10 tipos distintos. Caso existam 5 itens com dimensões idênticas (10×15) e outros 5 itens com dimensões idênticas (5×9), teremos 2 tipos. Além disso, é relevante observar que itens com dimensões iguais, mas rotacionados, são considerados tipos distintos. Portanto, se possuímos 2 itens, um com dimensões 13×18 e outro com dimensões 18×13 , teremos dois tipos diferentes.

Assume-se que existem m tipos diferentes de itens, que terão seus índices armazenados em uma lista α , com $\alpha_0 = 0$ e α_j sendo igual ao último índice dos itens do j -ésimo tipo. Isso significa que os índices dos itens do tipo j variam de $\alpha_{j-1} + 1$ a α_j . Por exemplo, se há dez itens do tipo 1 e cinco itens do tipo 2, então, $\alpha_1 = 10$ e $\alpha_2 = 15$, já que o valor de α_j é igual ao maior índice do item do tipo j , uma vez que os itens do mesmo tipo são numerados de forma consecutiva. Ou seja, os itens 1, 2, ..., 10 são todos do tipo 1 (e,

portanto, $\alpha_1 = 10$), enquanto os itens 11, 12, ..., 15 são todos do tipo 2 (e, portanto, $\alpha_2 = 15$).

A Tabela 4 a seguir resume a notação utilizada nos modelos matemáticos, indicando os índices, parâmetros e as variáveis de decisão (que, no caso dos quatro modelos, são todas binárias):

Tabela 4: Notação dos modelos matemáticos

Índices	
i	Índice do item
k	Índice do nível
l	Índice do objeto
j	Índice do tipo do item
α_j	Índice do último item do j -ésimo tipo
n	Quantidade total de itens
p	Quantidade total de objetos
m	Quantidade total de tipos de itens
Parâmetros	
w_i	Largura do item i
h_i	Altura do item i
w_k	Largura do nível k
h_k	Altura do nível k
W_l	Largura do objeto l
H_l	Altura do objeto l
c_l	Custo/área do objeto l
Variáveis de Decisão (binárias)	
u_l	Indica se o objeto l é utilizado (1), ou não (0)
x_{ikl}	Indica se o item i não-rotacionado é alocado no nível k no objeto l (1), ou não (0)
y_{ikl}	Indica se o item i rotacionado é alocado no nível k no objeto l (1), ou não (0)

Fonte: o Autor.

4.2 Modelos sem Critério de Desempate

Os modelos apresentados nessa seção não possuem o critério de desempate desenvolvido para priorizar as soluções que geram o menor número de níveis. A seguir, os modelos sem possibilidade de rotação (Modelo S) e com possibilidade de rotação (Modelo R) são apresentados.

4.2.1 Modelo S - Sem Rotação

O modelo sem rotação e sem critério de desempate desenvolvido pelo presente trabalho pode ser escrito como um modelo de Programação Linear Inteira:

$$\min \sum_{l=1}^p c_l W_l H_l u_l \quad (4.4)$$

$$\text{s.a.} \quad \sum_{k=1}^n h_k x_{kkl} \leq H_l u_l \quad l = 1, \dots, p \quad (4.5)$$

$$\sum_{i=k+1}^n w_i x_{ikl} \leq (W_l - w_k) x_{kkl} \quad k = 1, \dots, n, \quad l = 1, \dots, p \quad (4.6)$$

$$\sum_{l=1}^p \sum_{k=1}^i x_{ikl} = 1 \quad i = 1, \dots, n \quad (4.7)$$

$$\sum_{l=1}^p x_{k+1,k+1,l} \leq \sum_{l=1}^p x_{kkl} \quad j = 1, \dots, m, \quad k \in [\alpha_{j-1} + 1, \alpha_j - 1] \quad (4.8)$$

$$\sum_{l=1}^p \sum_{i=k+2}^{\alpha_j} x_{i,k+1,l} \leq \sum_{l=1}^p \sum_{i=k+1}^{\alpha_j} x_{ikl} \quad j = 1, \dots, m, \quad k \in [\alpha_{j-1} + 1, \alpha_j - 1] \quad (4.9)$$

$$u_l \in \{0, 1\} \quad l = 1, \dots, p \quad (4.10)$$

$$x_{ikl} \in \{0, 1\} \quad k = 1, \dots, n, \quad i = k, \dots, n, \quad l = 1, \dots, p \quad (4.11)$$

A função objetivo (4.4) minimiza o custo total dos objetos utilizados. As restrições (4.5) garantem que, para cada objeto utilizado, a soma das alturas dos níveis abertos não seja maior do que a altura do objeto; além disso, garantem que os níveis abertos sejam atribuídos apenas a objetos utilizados. As restrições (4.6) garantem que, para cada objeto, a soma das larguras dos itens alocados a cada nível não seja maior do que a largura do objeto, e que um item só pode ser alocado a um nível se o nível estiver aberto. As restrições (4.7) asseguram que a demanda de cada item seja atendida. As restrições (4.8) e (4.9) servem para eliminar simetrias de itens idênticos (i.e., de um mesmo tipo). As

restrições (4.8) estabelecem que um item que não é o primeiro de seu tipo só pode abrir um nível se o item anterior (do mesmo tipo) também abrir um nível. As restrições (4.9) estabelecem que, se dois itens consecutivos k e $k+1$ do tipo j abrirem um nível, o número de itens do tipo j no nível $k+1$ deve ser menor ou igual ao número de itens do tipo j no nível k . As restrições (4.10) e (4.11) definem o domínio das variáveis u_l e x_{ikl} .

4.2.2 Modelo R - Com Rotação

O modelo com rotação e sem critério de desempate desenvolvido pelo presente trabalho pode ser escrito como um modelo de Programação Linear Inteira:

$$\min \sum_{l=1}^p c_l W_l H_l u_l \quad (4.12)$$

$$\text{s.a.} \quad \sum_{k=1}^n (h_k x_{kkl} + w_k y_{kkl}) \leq H_l u_l \quad l = 1, \dots, p \quad (4.13)$$

$$\sum_{i=k+1}^n (w_i x_{ikl} + h_i y_{ikl}) \leq (W_l - w_k) x_{kkl} + (W_l - h_k) y_{kkl} \quad k = 1, \dots, n, \quad l = 1, \dots, p \quad (4.14)$$

$$\sum_{l=1}^p \sum_{k=1}^i (x_{ikl} + y_{ikl}) = 1 \quad i = 1, \dots, n \quad (4.15)$$

$$\sum_{l=1}^p (x_{k+1,k+1,l} + y_{k+1,k+1,l}) \leq \sum_{l=1}^p (x_{kkl} + y_{kkl}) \quad j = 1, \dots, m, \quad k \in [\alpha_{j-1} + 1, \alpha_j - 1] \quad (4.16)$$

$$\sum_{l=1}^p \sum_{i=k+2}^{\alpha_j} (x_{i,k+1,l} + y_{i,k+1,l}) \leq \sum_{l=1}^p \sum_{i=k+1}^{\alpha_j} (x_{ikl} + y_{ikl}) \quad j = 1, \dots, m, \quad k \in [\alpha_{j-1} + 1, \alpha_j - 1] \quad (4.17)$$

$$(h_k x_{kkl} + w_k y_{kkl}) \geq (h_{k'} x_{k',k,l} + w_{k'} y_{k',k,l}) \quad l = 1, \dots, p, \quad k = 1, \dots, n, \quad k' = k + 1, \dots, n \quad (4.18)$$

$$u_l \in \{0, 1\} \quad l = 1, \dots, p \quad (4.19)$$

$$x_{ikl} \in \{0, 1\} \quad k = 1, \dots, n, \quad i = k, \dots, n, \quad l = 1, \dots, p \quad (4.20)$$

$$y_{ikl} \in \{0, 1\} \quad k = 1, \dots, n, \quad i = k, \dots, n, \quad l = 1, \dots, p \quad (4.21)$$

A função objetivo (4.12) minimiza o custo total dos objetos utilizados. As demais restrições também são equivalentes ao modelo sem rotação, porém, com a adição da variável binária y que representa os itens rotacionados. Como as restrições (4.15) garantem que um item só pode assumir apenas uma orientação, então, basta adicionar a variável binária y na formulação.

Caso haja alguma operação de multiplicação, então, basta acompanhar a variável y com a dimensão oposta àquela que acompanha x (p.e., se altura acompanha a variável x como em $h_k x_{kkl}$, então, basta trocar altura por largura quando multiplicarmos por y , resultando em $w_k y_{kkl}$). Isso se deve ao fato de que cada item rotacionado compartilha as mesmas dimensões de largura e altura que o item não-rotacionado correspondente, com os valores invertidos: a largura do item rotacionado é equivalente à altura do item não-rotacionado, e vice-versa (a altura do item rotacionado é igual à largura do item não-rotacionado).

Logo, as restrições (4.13) garantem que, para cada objeto utilizado, a soma das alturas dos níveis abertos, tanto por itens rotacionados ou não-rotacionados, não seja maior do que a altura do objeto; além disso, garantem que os níveis abertos sejam atribuídas apenas a objetos utilizados. As restrições (4.14) garantem que, para cada objeto, a soma das larguras dos itens, rotacionados ou não-rotacionados, alocados a cada nível não seja maior do que a largura do objeto; além disso, garantem que um item, tanto rotacionado quanto não-rotacionado, só pode ser alocado a um nível se a mesma estiver aberta. As restrições (4.15) asseguram que o item só pode assumir uma orientação e que a demanda por ele seja atendida. As restrições (4.16) e (4.17) servem para eliminar simetrias de itens do mesmo tipo, sejam eles rotacionados ou não-rotacionados. As restrições (4.16) estabelecem que, se um item que não é o primeiro de seu tipo só pode abrir um nível se o item anterior (do mesmo tipo) também abrir um nível. As restrições (4.17) estabelecem que, se dois itens consecutivos k e $k + 1$ do tipo j abrirem um nível, o número de itens do tipo j no nível $k + 1$ deve ser menor ou igual ao número de itens do tipo j no nível k . As restrições (4.18) garantem que os itens estão ordenados da maior para a menor altura em um nível aberto (i.e., o primeiro item alocado em um nível é o que possui a maior altura daquele nível; o próximo item possui a segunda maior altura, e assim por diante); essas restrições foram outra contribuição do presente trabalho para tornar possível a rotação dos itens, já que não havia nenhuma restrição no modelo original que limitava a altura dos itens em um nível (dado que os itens estavam ordenados por alturas decrescentes). As restrições (4.19), (4.20) e (4.21) definem o domínio das variáveis u_l , x_{ikl} e y_{ikl} .

4.3 Modelos com Critério de Desempate

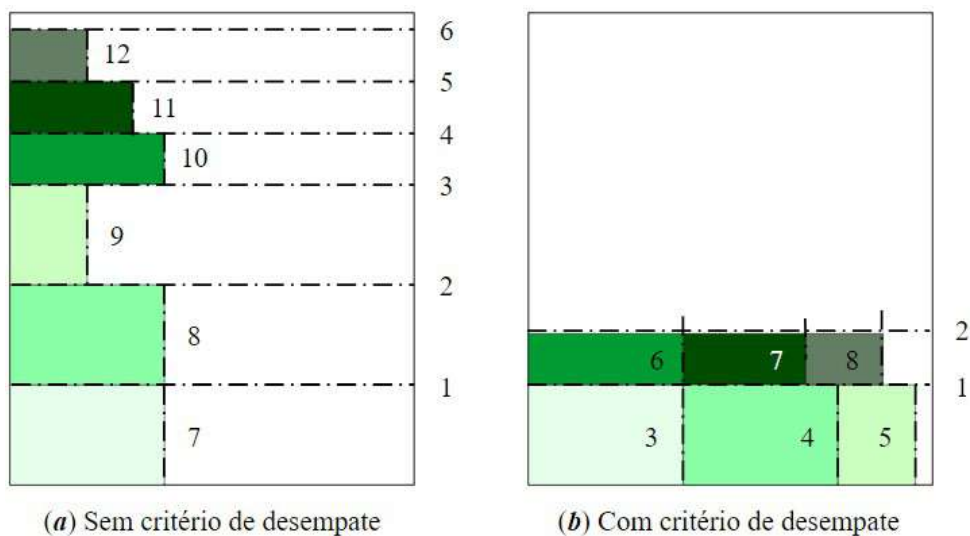
Nesta seção, o critério de desempate desenvolvido para priorizar as soluções que geram o menor número de níveis é apresentado para os dois modelos, sem (Modelo S^D) e com possibilidade de rotação (Modelo R^D).

4.3.1 Critério de Desempate

O critério de desempate foi uma adição em relação aos modelos S e R que o presente trabalho desenvolveu para lidar com casos em que a solução encontrada pelo modelo alocava os itens em múltiplos níveis, sendo que era possível alocá-los em uma quantidade menor de níveis, mantendo o mesmo valor (de custo dos objetos) da solução previamente encontrada. Ou seja, o critério serve apenas para desempatar as melhores soluções encontrados, priorizando a solução que fornece o menor número de níveis; logo, ele não altera o valor de custo de uma instância.

A Figura 31 a seguir representa os dois cenários, sem critério de desempate (a) e com o critério (b). É possível notar que, no cenário sem o critério, os itens foram alocados em 6 níveis, enquanto no outro cenário apenas 2 níveis foram abertos. Isso resultou, no exemplo abaixo, em uma diferença de 4 cortes: 12 cortes seriam necessários para separar os itens no cenário (a), enquanto apenas 8 seriam necessários no cenário (b). A diferença se deu por conta dos cortes de primeiro estágio: no cenário (a), 6 cortes foram necessários (um para cada nível); já no cenário (b), apenas 2 cortes foram necessários.

Figura 31: Representação dos cenários sem e com critério de desempate.



Fonte: o Autor.

O desenvolvimento desse critério de desempate para as soluções ótimas é pertinente por dois principais motivos: redução do tempo de corte na guilhotina e redução do desperdício de papel na gráfica.

O primeiro motivo para a introdução do critério de desempate é a otimização do tempo de corte na guilhotina. Quando os itens são alocados em múltiplos níveis sem critério de desempate, isso pode resultar em mais cortes na guilhotina, pois cada nível representa uma etapa de corte separada. Ao implementar o critério de desempate, que busca minimizar o número de níveis utilizados, é possível reduzir o número de cortes necessários na guilhotina, especialmente quando os itens alocados em um nível possuem a mesma altura (sem a necessidade de *trimming*).

Isso se traduz em economia de tempo durante o processo de produção, pois menos cortes significam uma produção mais eficiente e rápida. De acordo com o que foi coletado presencialmente na gráfica pelo Autor, o tempo médio de corte na empresa (considerando a medição do tamanho do nível, o ajuste do papel na bancada, a preparação da guilhotina e o corte em si) é de 30 segundos. Logo, cada corte economizado se traduz em uma economia média de 30 segundos no tempo de produção. No exemplo da Figura 31, a economia seria de 2 minutos, em apenas 1 folha.

O segundo motivo importante para a introdução do critério de desempate é a redução do desperdício de papel. Quando os itens são alocados em um maior número de níveis, há uma maior probabilidade de haver espaço não utilizado em cada nível, o que resulta em um desperdício de papel. Ao utilizar o critério de desempate para minimizar o número de níveis, os itens são alocados de maneira mais eficiente, ocupando o espaço disponível de forma mais otimizada. Isso leva a uma redução significativa no desperdício de papel, já que as sobras podem ser reutilizadas como papéis de rascunho para os colaboradores, o que é economicamente vantajoso e ambientalmente responsável.

4.3.2 Modelos S^D e R^D

O critério de desempate está inserido na função objetivo como uma parcela adicional à formulação inicial de cada modelo, S e R . Portanto, a única diferença na modelagem é a adição do critério de desempate, com as restrições continuando exatamente iguais.

O modelo sem rotação e com critério de desempate (S^D) pode ser obtido apenas alterando a função objetivo (4.4) pela função objetivo (4.22) a seguir:

$$\min \sum_{l=1}^p c_l W_l H_l u_l + \sum_{l=1}^p \frac{\sum_{k=1}^n x_{kkl}}{\frac{H_l}{\min_{i=1,\dots,n} \{w_i, h_i\}}} \quad (4.22)$$

A função objetivo (4.22) minimiza o custo total dos objetos utilizados, priorizando a solução que forneça o menor número de níveis possível. A primeira parcela é idêntica à da expressão (4.4). A segunda parcela indica o critério de desempate das soluções, que prioriza o menor número de níveis (somatório de todos os x_{kkl}), dentre a quantidade máxima possível de níveis de cada objeto (obtida através da divisão entre a altura H_l do objeto l pelo item com a menor dimensão possível, seja w_i ou h_i , já que, nesse caso, retornaria o maior valor possível de níveis).

Já o modelo com rotação e com critério de desempate (R^D) pode ser obtido apenas alterando a função objetivo (4.12) pela função objetivo (4.23) a seguir:

$$\min \sum_{l=1}^p c_l W_l H_l u_l + \sum_{l=1}^p \frac{\sum_{k=1}^n x_{kkl} + y_{kkl}}{\frac{H_l}{\min_{i=1,\dots,n} \{w_i, h_i\}}} \quad (4.23)$$

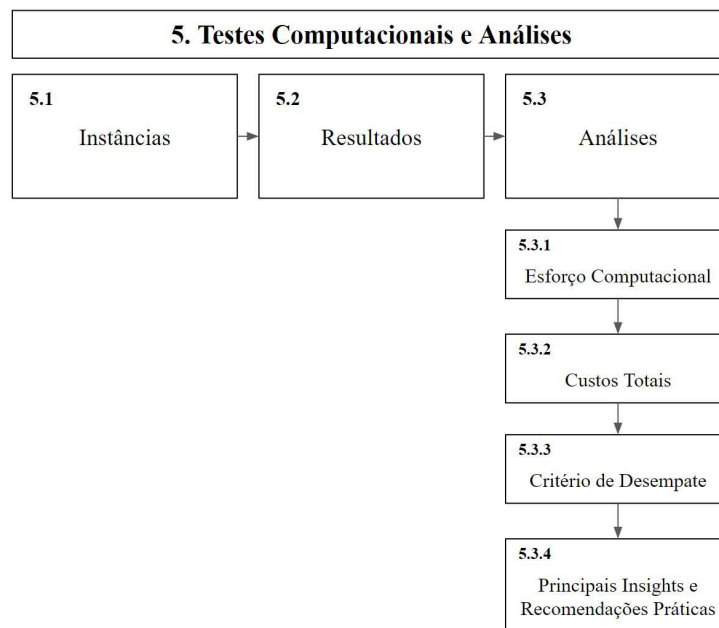
A função objetivo (4.23) minimiza o custo total dos objetos utilizados, priorizando a solução que forneça o menor número de níveis possível. A diferença em relação ao modelo sem rotação é a adição da variável y_{kkl} na parcela de critério de desempate, já que, agora, é possível que um item rotacionado abra um nível.

5 TESTES COMPUTACIONAIS E ANÁLISES

Neste capítulo são apresentadas as principais informações relacionadas aos testes computacionais realizados para testar os modelos desenvolvidos, bem como as análises realizadas sobre os resultados. O objetivo é apresentar o conjunto de instâncias que foram coletadas da operação diária da gráfica, comparando as soluções obtidas pelos modelos com as soluções adotadas na prática. Apenas lembrando, atualmente a empresa não possui nenhuma ferramenta que a auxilie no processo de montagem dos Planos de Impressão e Corte.

Em relação à organização do capítulo, ele está dividido em três seções: declaração das instâncias para teste, resultados obtidos e análises. A estrutura do capítulo pode ser observada na Figura 32 a seguir.

Figura 32: Estrutura do Capítulo 5.



Fonte: o Autor.

5.1 Instâncias

A Tabela 5 descreve as 20 instâncias testadas: a coluna p indica o número de objetos considerados, n indica o número de itens e as colunas $L \times A$ indicam as dimensões.

Tabela 5: Instâncias utilizadas nos testes computacionais.

Inst.	Objetos		Itens	
#	p	$L \times A$	n	$L \times A$
1	30	6(32 × 66), 6(33 × 48), 6(32 × 33), 6(28 × 38), 6(22 × 32)	7	1(40 × 25), 2(30 × 20), 2(60 × 30), 2(40 × 30)
2	15	3(32 × 66), 3(33 × 48), 3(32 × 33), 3(28 × 38), 3(22 × 32)	6	3(15 × 10), 2(18 × 13), 1(20 × 15)
3	25	5(32 × 66), 5(33 × 48), 5(32 × 33), 5(28 × 38), 5(22 × 32)	9	1(9 × 5), 1(9 × 6), 1(12 × 9), 1(15 × 10), 1(21 × 15), 1(18 × 13), 1(25 × 20), 1(30 × 20), 1(40 × 20)
4	45	9(32 × 66), 9(33 × 48), 9(32 × 33), 9(28 × 38), 9(22 × 32)	16	10(30 × 10), 2(30 × 21), 2(28 × 22), 2(30 × 30)
5	20	4(32 × 66), 4(33 × 48), 4(32 × 33), 4(28 × 38), 4(22 × 32)	16	8(9 × 6), 2(7 × 5), 6(5 × 5)
6	40	8(32 × 66), 8(33 × 48), 8(32 × 33), 8(28 × 38), 8(22 × 32)	12	5(30 × 20), 4(21 × 15), 3(15 × 10)
7	30	6(32 × 66), 6(33 × 48), 6(32 × 33), 6(28 × 38), 6(22 × 32)	19	8(9 × 6), 6(21 × 15), 4(7 × 7), 1(21 × 10)
8	20	4(32 × 66), 4(33 × 48), 4(32 × 33), 4(28 × 38), 4(22 × 32)	19	12(6 × 5), 5(20 × 5), 2(30 × 10)
9	20	4(32 × 66), 4(33 × 48), 4(32 × 33), 4(28 × 38), 4(22 × 32)	18	5(8 × 10), 6(10 × 10), 5(5 × 20), 2(10 × 15)
10	60	12(32 × 66), 12(33 × 48), 12(32 × 33), 12(28 × 38), 12(22 × 32)	30	10(30 × 10), 6(42 × 30), 4(21 × 15), 10(30 × 21)
11	55	11(32 × 66), 11(33 × 48), 11(32 × 33), 11(28 × 38), 11(22 × 32)	21	3(28 × 22), 1(43 × 28), 4(22 × 14), 2(20 × 15), 5(21 × 10), 6(15 × 10)
12	50	10(32 × 66), 10(33 × 48), 10(32 × 33), 10(28 × 38), 10(22 × 32)	25	5(12 × 9), 5(15 × 10), 5(20 × 15), 5(18 × 13), 5(25 × 20)
13	40	8(32 × 66), 8(33 × 48), 8(32 × 33), 8(28 × 38), 8(22 × 32)	19	2(6 × 5), 2(9 × 6), 4(9 × 9), 8(18 × 10), 3(30 × 21)
14	30	6(32 × 66), 6(33 × 48), 6(32 × 33), 6(28 × 38), 6(22 × 32)	24	18(15 × 10), 6(30 × 30)
15	70	14(32 × 66), 14(33 × 48), 14(32 × 33), 14(28 × 38), 14(22 × 32)	28	2(15 × 10), 3(21 × 10), 23(30 × 10)
16	45	9(32 × 66), 9(33 × 48), 9(32 × 33), 9(28 × 38), 9(22 × 32)	25	5(28 × 22), 10(9 × 5), 10(18 × 13)
17	35	7(32 × 66), 7(33 × 48), 7(32 × 33), 7(28 × 38), 7(22 × 32)	30	10(30 × 10), 20(9 × 5)
18	35	7(32 × 66), 7(33 × 48), 7(32 × 33), 7(28 × 38), 7(22 × 32)	30	10(20 × 15), 8(10 × 8), 12(9 × 5)
19	25	5(32 × 66), 5(33 × 48), 5(32 × 33), 5(28 × 38), 5(22 × 32)	42	30(6 × 6), 8(5 × 20), 4(7 × 7)
20	25	5(32 × 66), 5(33 × 48), 5(32 × 33), 5(28 × 38), 5(22 × 32)	50	15(9 × 6), 30(5 × 5), 5(20 × 5)

Fonte: o Autor.

É possível observar que as instâncias possuem um caráter fortemente heterogêneo de itens, com dimensões e quantidades variadas. O conjunto de objetos possui sempre as mesmas dimensões (os 5 Formatos já apresentados), por isso, apresenta um caráter fracamente heterogêneo.

A quantidade de objetos para teste foi obtida através de um cálculo rápido com a equipe da gráfica, buscando a quantidade mínima de objetos do Formato 9 S (22×32) que comportasse todos os itens daquela instância. Essa quantidade, então, foi replicada para os demais Formatos. Esse procedimento de buscar a quantidade de folhas do formato 9 S para atender o pedido é uma prática comum na gráfica, já que é a folha mais utilizada pela empresa na operação atual. O principal motivo levantado para essa escolha é que o seu custo/folha é o mais baixo entre as opções, além de ser o Formato mais abundante no estoque. A Seção 5.3 irá abordar algumas análises para avaliar se essa preferência pelo Formato 9 S é adequada.

A Tabela 6 a seguir resume as soluções adotadas pela gráfica na prática, para cada instância. As colunas representam a quantidade de objetos escolhidos de cada Formato que a empresa utilizou para atender as instâncias, bem como a quantidade total de objetos e um cálculo do custo total de cada instância.

Tabela 6: Objetos escolhidos pela gráfica para solucionar as instâncias.

Inst.	Solução adotada pela gráfica					
#	3 S	4 S	6 S	8 G	9 S	Total (R\$)
1	2	4				8,80
2					3	1,68
3		3				3,48
4			2		9	6,32
5		1				1,16
6					8	4,48
7		2			3	4,00
8		1			2	2,28
9					4	2,24
10	2	6			12	17,84
11		2		5	3	7,00
12			1	1	10	6,84
13	1	2			2	5,52
14	3				5	9,04
15					14	7,84
16					9	5,04
17	2				2	5,28
18				6	1	4,16
19			3			1,92
20		1	1		1	2,36

Fonte: o Autor.

É possível observar que as soluções adotadas pela gráfica concentram-se em dois objetos principalmente: o Formato 4 S e o Formato 9 S. Os outros Formatos foram escolhidos em poucas situações. O custo total de todas soluções na prática foi de R\$ 107,28.

5.2 Resultados

Os testes dos quatro modelos de otimização propostos neste trabalho foram realizados utilizando um computador pessoal com as seguintes especificações: processador 13th Gen Intel(R) Core(TM) i5-13400F @ 2.5 GHz, 16 GB de memória RAM, placa de vídeo de 8 GB e Windows 10 como o sistema operacional. Essas especificações são semelhantes às máquinas que a empresa possui, e, por isso, representam um cenário viável para a aplicação da solução. A aplicação foi desenvolvida em Python 3.11, com o Gurobi 10.0 sendo o resolvidor de propósito geral utilizado.

Para fins de padronização a respeito da apresentação dos resultados, a Tabela 7 a seguir resume as siglas utilizadas para cada um dos modelos desenvolvidos, como já apresentado no Capítulo 4 anteriormente.

Tabela 7: Siglas dos 4 modelos.

Modelo	Sigla
Modelo sem Rotação, sem Critério de Desempate	S
Modelo sem Rotação, com Critério de Desempate	S^D
Modelo com Rotação, sem Critério de Desempate	R
Modelo com Rotação, com Critério de Desempate	R^D

Fonte: o Autor.

Antes de apresentar os resultados dos testes vale destacar que, para fins de padronização, as tabelas que serão apresentadas na sequência apresentam indicadores comuns, que seguem a mesma estrutura:

1. **F.O.:** representa o valor da função objetivo ao final dos procedimentos de otimização realizados pelo Gurobi, ou seja, o custo total da instância que o modelo buscou minimizar.

2. **GAP (%)**: indica o quão distante da solução ótima está a melhor solução factível encontrada pelo resolvedor. Ou seja, representa a diferença entre o valor da função objetivo encontrado pelo resolvedor (dentro dos parâmetros estabelecidos) e o melhor valor possível (neste caso, um limitante inferior) que o resolvedor calculou. Portanto, se o valor for 0, isso indica que aquela instância chegou até o valor ótimo.
3. **Tempo (s)**: indica em quantos segundos o resolvedor chegou na solução ótima; caso não tenha chegado na solução ótima, ela apresentará o tempo máximo de 900s estabelecido pelo Autor para os testes. Este valor está próximo do tempo que os colaboradores da gráfica levam para preparar o Plano de Impressão e Corte e o *setup* do equipamento atualmente, portanto, é um tempo adequado para limitar os modelos. Além de ser um indicador relevante para avaliar o esforço computacional para resolver cada instância, ele também dá dimensão da possível economia de tempo na montagem dos Planos.
4. **Nós**: indica quantos nós da árvore do *Branch-and-Bound* (B&B) foram explorados pelo Gurobi durante o processo de otimização. Esse indicador é relevante para avaliar o esforço computacional para resolver cada instância.
5. **Iterações**: indica quantas iterações do método utilizado pelo resolvedor (estratégia *Branch-and-Cut*) foram realizadas pelo Gurobi durante o processo de otimização. Esse indicador também é relevante para avaliar o esforço computacional para resolver cada instância.
6. **Níveis**: indica em quantos níveis (no total) a solução encontrada pelo Gurobi alocou os itens, ou seja, essa coluna soma a quantidade de níveis para todos os objetos utilizados. Esse indicador é importante para avaliar o impacto do critério de desempate.
7. **Objetos**: esse conjunto de colunas indica a quantidade de objetos utilizados na solução encontrada para cada Formato. Essa série de colunas é relevante para avaliar as diferenças entre a solução encontrada pelo modelo e a solução adotada na prática pela gráfica.

As Tabelas 8, 9, 10 e 11 a seguir descrevem os resultados obtidos para os modelos S , S^D , R e R^D , respectivamente. Estes resultados serão analisados na Seção 5.3, que irá comparar de maneira consolidada os quatro modelos entre as dimensões coletadas nesta seção de resultados.

Tabela 8: Soluções do Modelo S .

Inst.	Solução		Info. Computacional			Nív.	Objetos				
#	F.O.	GAP (%)	Tempo (s)	Nós	Iter.	Qtd.	3 S	4 S	6 S	8 G	9 S
1	Inf.	-	-	-	-						
2	1,20	0,00	0,05	1	135	6				2	
3	Inf.	-	-	-	-						
4	5,12	0,00	0,37	238	2446	16			8		
5	0,60	0,00	0,25	1	2904	6				1	
6	4,60	0,00	0,17	165	1676	11		3			2
7	2,28	0,00	0,56	1	8372	11				1	3
8	1,20	0,00	0,97	55	11424	10			1		1
9	1,72	0,00	2,51	1361	50572	11		1			1
10	Inf.	-	-	-	-						
11	Inf.	-	-	-	-						
12	5,28	0,00	6,03	2900	108089	18		2		4	1
13	3,48	0,00	2,99	712	42977	13		2		1	1
14	6,68	0,00	0,12	1	3348	15		3	5		
15	5,68	0,00	0,15	1	328	27			8		1
16	4,68	0,00	4,95	918	71004	18				5	3
17	3,04	0,00	5,06	108	59333	19		1	2	1	
18	3,48	0,00	5,41	45	96087	18				3	3
19	1,72	0,00	65,11	4551	867800	18				1	2
20	1,72	0,00	900,00	235037	3352710	18				1	2

Tabela 9: Soluções do Modelo S^D .

Inst.	Solução		Info. Computacional			Nív.	Objetos				
#	F.O.	GAP (%)	Tempo (s)	Nós	Iter.	Qtd.	3 S	4 S	6 S	8 G	9 S
1	Inf.	-	-	-	-						
2	1,20	0,00	0,09	1	138	6				2	
3	Inf.	-	-	-	-						
4	5,12	0,00	0,43	131	2096	16			8		
5	0,60	0,00	1,40	1	5184	5				1	
6	4,60	0,00	1,27	1379	12556	11		3			2
7	2,28	0,00	15,27	323	61620	11				1	3
8	1,20	0,00	5,97	316	21011	10			1		1
9	1,72	0,00	10,28	3062	68973	10		1			1
10	Inf.	-	-	-	-						
11	Inf.	-	-	-	-						
12	5,28	0,00	58,57	34142	662663	18		2		4	1
13	3,48	0,00	9,66	5726	213447	12		2		1	1
14	6,68	0,00	0,72	1	8651	15		3	5		
15	5,68	0,00	1,31	1	1468	27			8		1
16	4,68	0,00	6,53	261	41739	17				5	3
17	3,04	0,00	51,24	83	94606	17		1	2	1	
18	3,48	0,00	102,33	1853	457586	18				3	3
19	1,72	0,00	288,40	5134	880224	18				1	2
20	1,72	0,00	900,00	27884	4172551	17				1	2

Fonte: o Autor.

Tabela 10: Soluções do Modelo R .

Inst.	Solução		Info. Computacional			Nív.	Objetos				
#	F.O.	GAP (%)	Tempo (s)	Nós	Iter.	Qtd.	3 S	4 S	6 S	8 G	9 S
1	8,76	0,00	0,03	1	56	7	2	3			2
2	1,16	0,00	0,15	1	686	3		1			
3	2,88	0,00	0,81	242	8913	6		2			1
4	5,12	0,00	8,41	2921	85688	16			8		
5	0,60	0,00	4,85	3238	129532	5				1	
6	4,04	0,00	2,78	2206	37944	10			1	1	5
7	1,80	0,00	4,85	1	29843	8				3	
8	1,16	0,00	12,70	81	69303	5				1	1
9	1,20	0,00	25,21	2700	566018	6				2	
10	15,28	0,00	27,28	4323	352783	25		6	6		8
11	5,28	0,00	196,51	7736	1291940	15		2		4	1
12	4,72	0,00	444,42	36548	6845847	16		1		5	1
13	2,92	0,00	379,51	35672	4775956	10		1		2	1
14	6,16	0,00	26,69	201	122915	14		2	6		
15	5,72	6,22	900,00	6363	1898533	13			8	1	
16	4,12	0,00	64,41	971	259288	13				5	2
17	3,00	0,00	270,65	1	311325	10				5	
18	2,96	0,00	106,75	1	284596	14				4	1
19	1,76	27,15	900,00	3025	3203342	12		1		1	
20	1,72	31,34	900,00	1651	2140141	14		1			1

Tabela 11: Soluções do Modelo R^D .

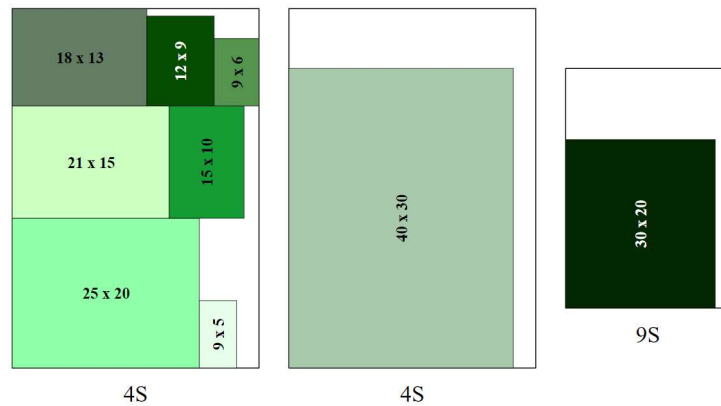
Inst.	Solução		Info. Computacional			Nív.	Objetos				
#	F.O.	GAP (%)	Tempo (s)	Nós	Iter.	Qtd.	3 S	4 S	6 S	8 G	9 S
1	8,76	0,00	0,07	1	194	7	2	3			2
2	1,16	0,00	0,47	1	1631	3		1			
3	2,88	0,00	1,75	442	19210	5		2			1
4	5,12	0,00	8,68	1891	98026	8			8		
5	0,60	0,00	9,33	3448	95559	4				1	
6	4,04	0,00	11,11	8864	191927	9			1	1	5
7	1,80	0,00	22,21	1	31431	8				3	
8	1,16	0,00	33,11	136	72417	5				1	1
9	1,20	0,00	64,20	2837	380850	6				2	
10	15,28	0,00	92,97	1770	196649	22		6	6		8
11	5,28	0,00	100,93	4018	709235	13		2		4	1
12	4,72	0,00	114,68	5300	1607788	15		1		5	1
13	2,92	0,00	677,96	14214	4588866	10		1		2	1
14	6,16	0,00	718,61	10099	7675711	14		2	6		
15	5,72	1,95	900,00	1778	1271456	10			8	1	
16	4,12	7,82	900,00	2773	1342170	12				5	2
17	3,00	9,43	900,00	1186	1036620	10		1	1	2	
18	3,04	15,09	900,00	1676	1164744	13			1	4	
19	1,76	30,50	900,00	367	1207194	9		1		1	
20	1,76	33,77	900,00	78	901905	12		1		1	

Fonte: o Autor.

As tabelas apresentadas anteriormente fornecem algumas informações valiosas de serem observadas. As Tabelas 8 e 9 indicam que 4 das 20 instâncias resultaram infactíveis, pois a rotação das peças não foi permitida; além disso, elas indicam que as 16 instâncias com resultados factíveis chegaram no valor ótimo, já que o GAP de todas é igual a zero. Ao observar as Tabelas 10 e 11, nota-se que as 20 instâncias apresentaram resultados factíveis, destacando a flexibilidade oferecida pelos modelos com a possibilidade de rotação na resolução do problema. No entanto, na Tabela 10, 17 das 20 instâncias tiveram a solução ótima comprovada, enquanto na Tabela 11, apenas 14 das 20 instâncias apresentaram um GAP igual a zero.

A Figura 33 a seguir apresenta um exemplo da resolução da instância 3 pelo modelo R^D , para ilustrar o que espera-se como resultado dos modelos implementados: a montagem do Plano de Impressão e Corte de forma visual, indicando o posicionamento dos itens e o conjunto de objetos que minimiza o custo total da instância. Quando os números estão dispostos verticalmente (p.e., o item com dimensões 30×20 alocado no objeto 9 S), isso sinaliza que o item foi rotacionado em 90° .

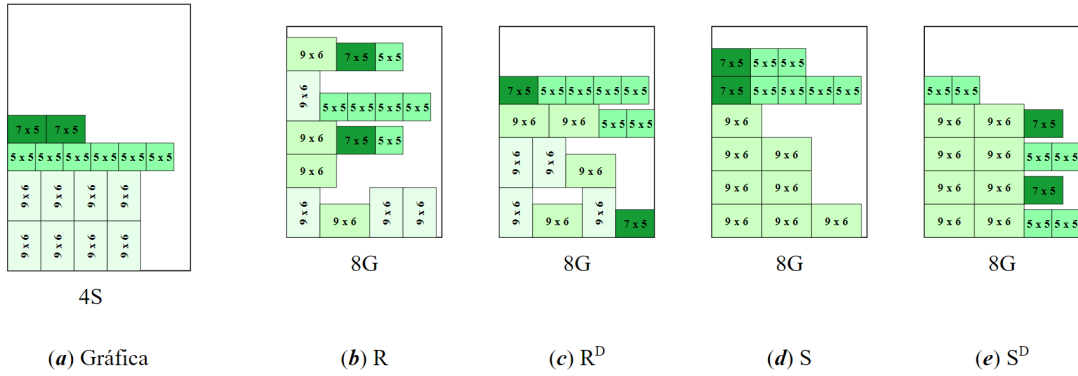
Figura 33: Exemplo de solução da instância 3 pelo modelo R^D .



Fonte: o Autor.

Por fim, a Figura 34 a seguir ilustra um exemplo da resolução da instância 5 pela gráfica (a), e pelos quatro modelos: R (b), R^D (c), S (d) e S^D (e). É possível notar que a solução da gráfica optou por utilizar um objeto maior (Formato 4 S), provavelmente por conta da maior facilidade na montagem do Plano de Impressão e Corte, enquanto os modelos sempre optaram pelo Formato 8 G, que é a opção disponível que apresenta o menor custo para resolver essa instância, mas envolve maior complexidade na montagem.

Figura 34: Comparação das soluções da instância 5.



Fonte: o Autor.

5.3 Análises

Esta seção apresenta as principais análises a partir dos resultados apresentados anteriormente na Seção 5.2. Ela está dividida em quatro partes: observação do esforço computacional das instâncias, comparação dos custos totais, avaliação do critério de desempate e consideração sobre os objetos utilizados.

5.3.1 Esforço Computacional

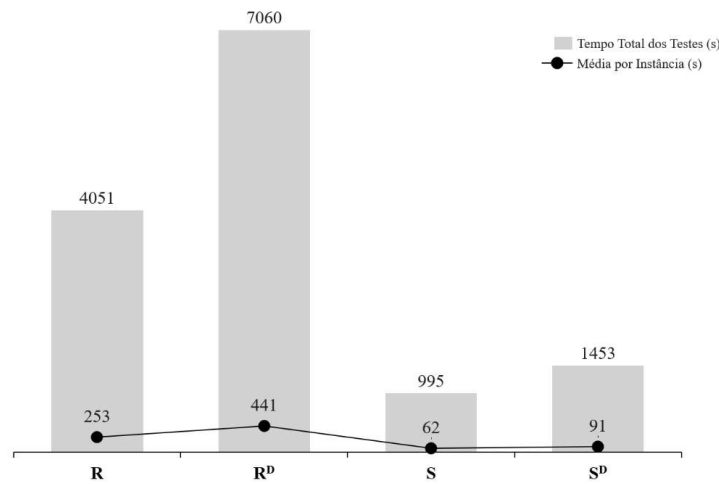
Esta subseção tem como objetivo apresentar brevemente os três principais indicadores relacionados ao esforço computacional para rodar as instâncias: o tempo de execução, a quantidade de nós explorados e o número de iterações realizadas.

A Figura 35 a seguir detalha, para cada modelo e desconsiderando as 4 instâncias com valor *infeasible* (as instâncias 1, 3, 10 e 11, já que não retornaram valores factíveis para os modelos sem possibilidade de rotação) o tempo total de execução em segundos (nas barras) e a média por instância em segundos (na linha). É possível notar que os modelos com rotação necessitam de mais tempo para rodar as instâncias, algo que faz sentido devido a sua maior complexidade. Além disso, os modelos com o critério de desempate também demonstraram consumir mais tempo para rodar os testes do que os modelos sem o critério (dentro da sua própria categoria de rotação ou não-rotação dos itens).

A Figura 36 a seguir detalha, para cada modelo e desconsiderando as 4 instâncias com valor *infeasible* (pelo mesmo motivo citado anteriormente), a quantidade total de nós do *Branch-and-Bound* (B&B) explorados (nas barras) e a média por instância (na

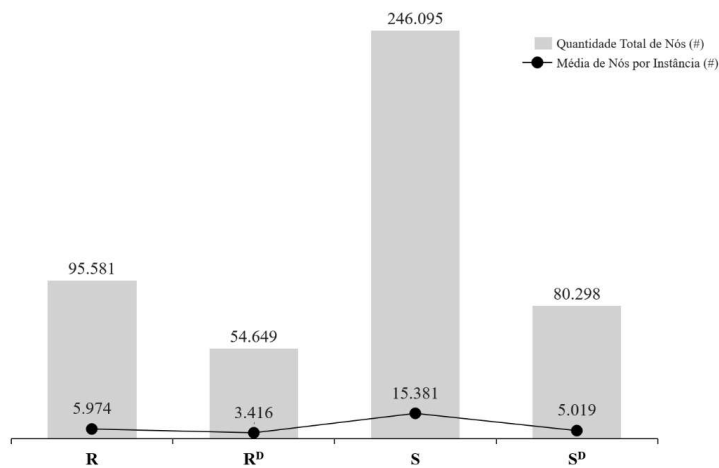
linha). Apesar dos modelos com rotação serem mais complexos, é possível notar que os modelos sem rotação exploraram, na média, mais nós do que os modelos com rotação, em um tempo mais curto (relacionando com a Figura 35 anterior). Também é possível notar que os modelos sem o critério de desempate percorreram mais nós do que os modelos com o critério (dentro da sua própria categoria de rotação ou não-rotação dos itens).

Figura 35: Comparação do tempo de execução para os quatro modelos.



Fonte: o Autor.

Figura 36: Comparação da quantidade de nós para os quatro modelos.

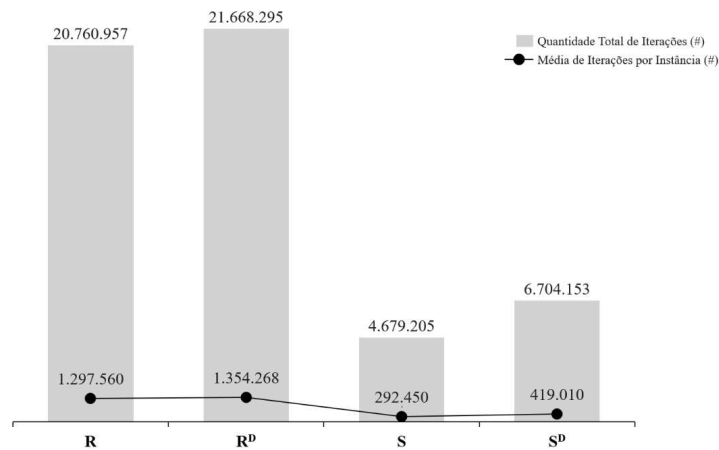


Fonte: o Autor.

Por fim, a Figura 37 a seguir detalha, para cada modelo e desconsiderando as 4 instâncias com valor *infeasible* (pelo mesmo motivo citado anteriormente), a quantidade total de iterações (nas barras) e a média por instância (na linha). É possível notar que

os modelos com rotação realizaram uma quantidade consideravelmente maior de iterações do que os modelos sem rotação, justamente por conta de sua maior complexidade. Além disso, os modelos com o critério de desempate apresentaram uma quantidade levemente superior de iterações realizadas em relação aos modelos sem o critério (dentro da sua própria categoria de rotação ou não-rotação dos itens), o que é razoável dado a sua maior complexidade.

Figura 37: Comparação da quantidade de iterações para os quatro modelos.



Fonte: o Autor.

5.3.2 Custos Totais

A Tabela 12 a seguir compila os valores de custo total encontrados pela gráfica na prática (coluna *Gráfica*) com os valores da função objetivo obtidos através dos procedimentos de otimização dos modelos propostos neste trabalho (colunas com os respectivos nomes dos 4 modelos). A coluna *Modelo* seleciona o melhor resultado entre os 2 modelos com rotação (como o problema é de minimização, seleciona o menor valor), já que estes são comparáveis à realidade da gráfica, que considera a rotação dos itens na montagem dos Planos de Impressão e Corte. Note que alguns valores possuem um asterisco (*) ao seu lado: isso indica que a solução não é comprovadamente ótima. A coluna *dif. abs.* representa a diferença em Reais (R\$) entre a coluna *Gráfica* e a coluna *Modelo*; por fim, a coluna *dif. %* representa a mesma diferença, só que percentualmente. Note também que a solução (ótima) dos modelos com desempate não é melhor do que a solução dos modelos sem desempate, como esperado segundo a Seção 4.3.

Tabela 12: Comparação dos custos totais entre a solução da gráfica e as soluções dos modelos.

Inst.	Comparação das Soluções				Com Rotação		Sem Rotação	
	Gráfica	Modelo	dif. abs.	dif. %	R	R^D	S	S^D
1	8,80	8,76	0,04	-0,5%	8,76	8,76	Inf.	Inf.
2	1,68	1,16	0,52	-31%	1,16	1,16	1,20	1,20
3	3,48	2,88	0,60	-17%	2,88	2,88	Inf.	Inf.
4	6,32	5,12	1,20	-19%	5,12	5,12	5,12	5,12
5	1,16	0,60	0,56	-48%	0,60	0,60	0,60	0,60
6	4,48	4,04	0,44	-10%	4,04	4,04	4,60	4,60
7	4,00	1,80	2,20	-55%	1,80	1,80	2,28	2,28
8	2,28	1,16	1,12	-49%	1,16	1,16	1,20	1,20
9	2,24	1,20	1,04	-46%	1,20	1,20	1,72	1,72
10	17,84	15,28	2,56	-14%	15,28	15,28	Inf.	Inf.
11	7,00	5,28	1,72	-25%	5,28	5,28	Inf.	Inf.
12	6,84	4,72	2,12	-31%	4,72	4,72	5,28	5,28
13	5,52	2,92	2,60	-47%	2,92	2,92	3,48	3,48
14	9,04	6,16	2,88	-32%	6,16	6,16	6,68	6,68
15	7,84	5,72*	2,12	-27%	5,72*	5,72*	5,68	5,68
16	5,04	4,12	0,92	-18%	4,12	4,12*	4,68	4,68
17	5,28	3,00	2,28	-43%	3,00	3,00*	3,04	3,04
18	4,16	2,96	1,20	-29%	2,96	3,04*	3,48	3,48
19	1,92	1,76*	0,16	-8%	1,76*	1,76*	1,72	1,72
20	2,36	1,72*	0,64	-27%	1,72*	1,76*	1,72	1,72

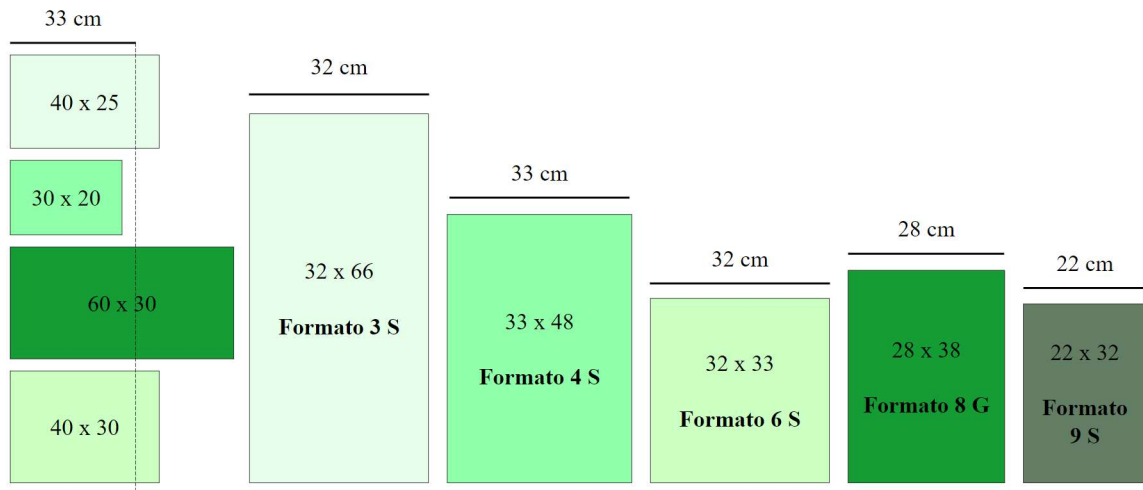
Fonte: o Autor.

O primeiro tópico a ser avaliado na Tabela 12 são as instâncias 1, 3, 10 e 11 que apresentam valores *infeasible* (inviáveis) nos modelos sem rotação. Esse comportamento se dá por conta das dimensões dos itens presentes nessas instâncias e pela característica dos modelos sem rotação. Quando um item possui largura maior do que 33 cm e não pode ser rotacionado, ele não consegue ser alocado em nenhum dos Formatos, que possuem no máximo 33 cm de largura. Isso ocorre em todas as instâncias citadas. Como exemplo, vamos citar a instância 1, composta pelos seguintes itens (com dimensões respeitando o padrão $L \times A$): 40×25 , 30×20 , 60×30 e 40×30 . Note que 3 dos 4 tipos de itens presentes nessa instância extrapolam o limite de 33 cm. Dessa forma, a solução torna-se *infeasible*. A Figura 38 a seguir ilustra essa situação.

O segundo tópico a ser avaliado na Tabela 12 é o desempenho dos modelos em comparação com as soluções adotadas pela gráfica. O objetivo é comparar, somando os custos de todas as instâncias, o melhor valor encontrado pelo Gurobi para os modelos com rotação (coluna *Modelo*) com as soluções adotadas na prática (coluna *Gráfica*). A Figura 39 a seguir ilustra os dois cenários de avaliação: considerando a soma dos custos das instâncias com valores *infeasible* (a), já que algum modelo foi capaz de encontrar um resultado para

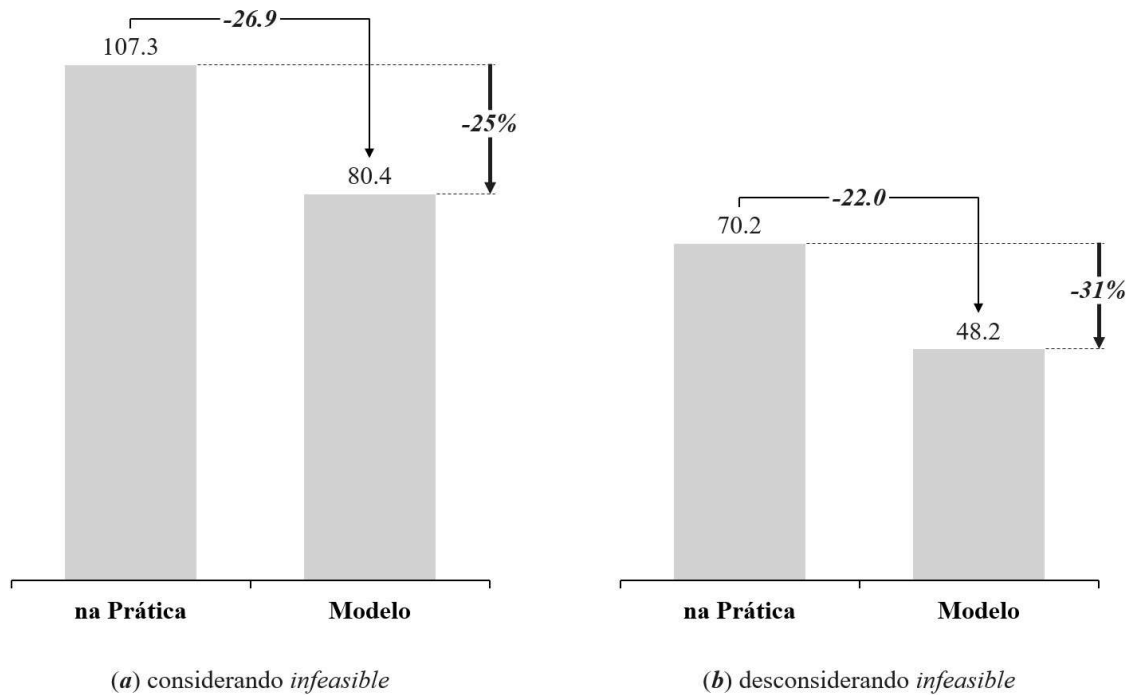
a instância; e desconsiderando (b). É possível notar que a utilização da melhor solução dos modelos apresentou uma economia em relação às soluções adotadas pela gráfica na faixa dos 25-31%.

Figura 38: Exemplo de instância *infeasible*.



Fonte: o Autor.

Figura 39: Comparação das soluções encontradas pelo modelo com as soluções adotadas na prática.

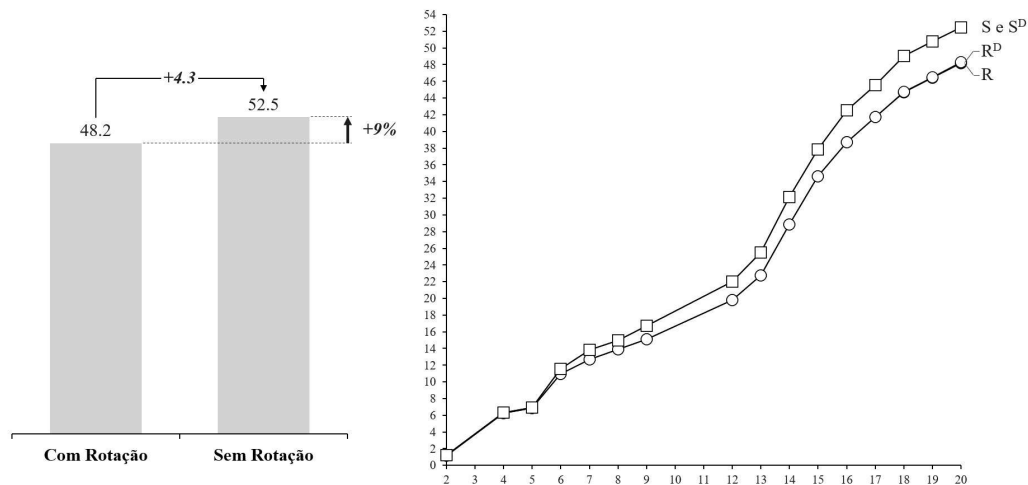


Fonte: o Autor.

Logo, observando a Figura 39, conclui-se que esse é um forte indicativo de que a adoção dos modelos de otimização propostos neste trabalho representa uma oportunidade de redução de custos para a empresa, que atualmente não possui nenhuma ferramenta para auxiliá-la na montagem dos Planos de Impressão e Corte.

Observando com mais atenção os valores das soluções encontradas por cada modelo, é possível avaliar que os modelos com rotações se mostram mais econômicos, ou seja, foram os modelos que mais conseguiram minimizar os custos totais das instâncias. A Figura 40 a seguir apresenta uma comparação entre a melhor solução obtida pelos modelos com rotação e a melhor solução obtida pelos modelos sem rotação. À esquerda é possível notar que, somando os custos totais (desconsiderando instâncias *infeasible*, já que a comparação de soma total seria injusta), constata-se que os modelos com rotação são de fato mais econômicos, aproximadamente 10% na média. À direita é possível observar a soma acumulada dos custos (representada pelos valores no eixo Y) para cada instância considerada (no eixo X). Percebe-se que, ao final, os modelos com rotações se distanciam dos modelos sem rotação, ilustrando a economia descrita no gráfico à esquerda.

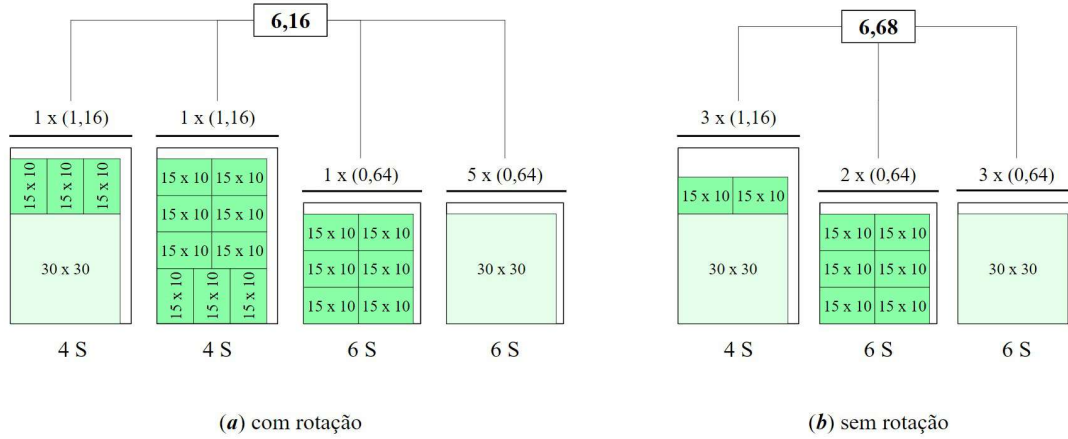
Figura 40: Comparação entre os modelos sem rotação e com rotação.



Fonte: o Autor.

A instância 14, representada pela Figura 41 a seguir, é um exemplo de como os modelos com rotação são mais econômicos que os modelos sem rotação. No cenário com rotação (a), o modelo conseguiu alocar todos os itens em 2 objetos do Formato 4 S e 6 objetos do Formato 6 S, totalizando R\$ 6,16. No cenário sem rotação (b), o modelo conseguiu alocar os itens em 3 objetos do Formato 4 S e em 5 objetos do Formato 6 S, totalizando R\$ 6,68, representando uma economia de 8% (R\$ 0,52).

Figura 41: Comparação da instância 14 entre os modelos com rotação e sem rotação.



Fonte: o Autor.

Por fim, o último tópico a ser comentado em relação aos valores encontrados pelos modelos propostos é o de instâncias que atingiram o limite máximo estabelecido pelo trabalho, de 900 s, e não retornaram uma solução comprovadamente ótima. Estes casos ocorreram, em sua grande maioria, nos modelos com rotação: instâncias 15, 19 e 20 do modelo R , e instâncias 15-20 do modelo R^D . Nessas instâncias, duas particularidades ocorreram:

- Os valores encontrados pelos modelos com rotação nas instâncias 18 e 20 são diferentes entre o modelo R e o modelo R^D , com o modelo R apresentando uma solução levemente melhor. Isso ocorreu justamente pois o modelo sem o critério de desempate, dentro da mesma limitação de tempo, apresentou um GAP menor do que o modelo com o critério, indicando que o modelo R ficou mais próximo da solução ótima. Em um cenário de soluções ótimas, isso não deve ocorrer, já que o critério de desempate não influencia no valor da função objetivo, apenas na alocação dos itens em menos níveis.
- Os valores encontrados nas instâncias 15 e 19 são menores nos modelos sem rotação do que nos modelos com rotação. Isso ocorreu justamente pois os modelos sem rotação chegaram a uma solução ótima (GAP igual a 0), enquanto os modelos com rotação não chegaram. Em um cenário de soluções ótimas, isso não deve ocorrer, já que os modelos sem rotações devem apresentar um resultado igual ou pior aos modelos com rotação.

A primeira particularidade explica o leve descasamento dos custos totais acumulados

observado na Figura 40 entre os modelos R e R^D ao final das instâncias. Já a segunda particularidade evidencia que o tempo limite de 900 s nessas instâncias foi um fator limitante para os modelos com rotação encontrarem uma solução melhor.

5.3.3 Critério de Desempate

Esta subseção tem como objetivo avaliar o impacto do critério de desempate desenvolvido pelo presente trabalho nas soluções encontradas pelos modelos, através da comparação do número de níveis abertos. A Tabela 13 a seguir compila a quantidade de níveis abertos pelos quatro modelos, em todas as instâncias. É possível notar que o número de níveis dos modelos com critério de desempate é sempre menor ou igual ao número de níveis encontrado pelos modelos sem o critério.

Tabela 13: Quantidade de níveis nos 4 modelos.

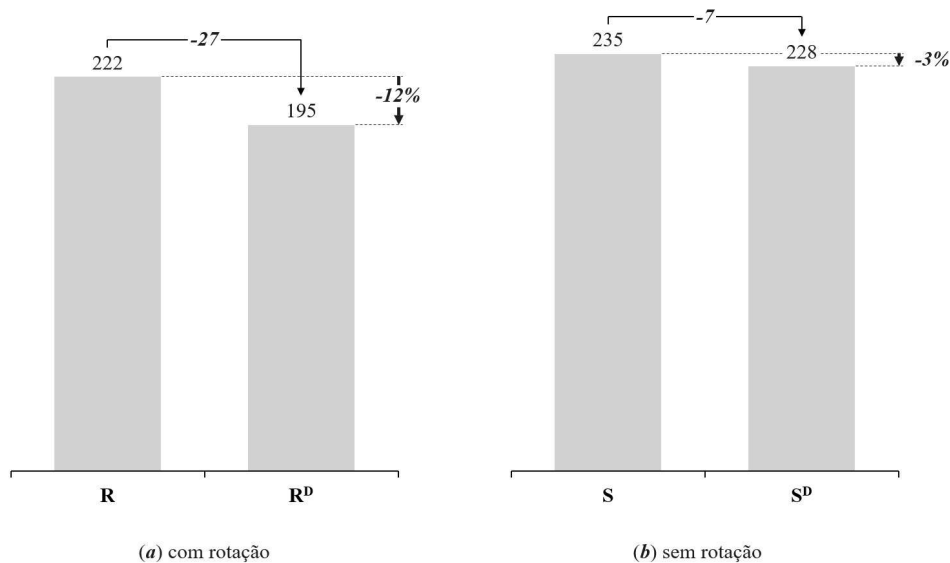
Inst.	R	R^D	S	S^D
#	Qtd.	Qtd.	Qtd.	Qtd.
1	7	7	-	-
2	3	3	6	6
3	6	5	-	-
4	16	8	16	16
5	5	4	6	5
6	10	9	11	11
7	8	8	11	11
8	5	5	10	10
9	6	6	11	10
10	25	22	-	-
11	15	13	-	-
12	16	15	18	18
13	10	10	13	12
14	14	14	15	15
15	13	10	27	27
16	13	12	18	17
17	10	10	19	17
18	14	13	18	18
19	12	9	18	18
20	14	12	18	17

Fonte: o Autor.

A Figura 42 a seguir ilustra de forma comparativa a quantidade total de níveis abertos nos cenários com rotação (a) e sem rotação (b); no segundo cenário, desconsiderou-se as instâncias com valores *infeasible* (já que não há solução factível). É possível notar que os modelos com critério de desempate, em ambos cenários, alocam os itens em uma quantidade menor de níveis. Nos modelos com rotação os resultados parecem ter mais impacto,

com uma média de redução de 12% no número de níveis (contra apenas 3% de redução no cenário sem rotação), por conta da flexibilidade que os modelos permitem na alocação dos itens. Aparentemente, a possibilidade de rotação dos itens favorece o desempate no número de níveis, já que a redução nestes modelos é consideravelmente maior do que nos modelos sem a possibilidade de rotação. Se desconsiderarmos os valores *infeasible* em ambos cenários (para a comparação de soma total dos níveis ser justa), chegamos a outra conclusão: a quantidade de níveis nos modelos sem rotação é consideravelmente maior do que nos modelos com rotação. Olhando somente para os modelos com o critério de desempate, temos 228 níveis no modelo sem rotação contra 148 níveis no modelo com rotação (para chegar a esse valor, basta somar a quantidade de níveis na Tabela 13, desconsiderando as instâncias 1, 3, 10 e 11), o que representa uma redução de 35% do número total de níveis.

Figura 42: Comparação da quantidade total de níveis abertos nos modelos.

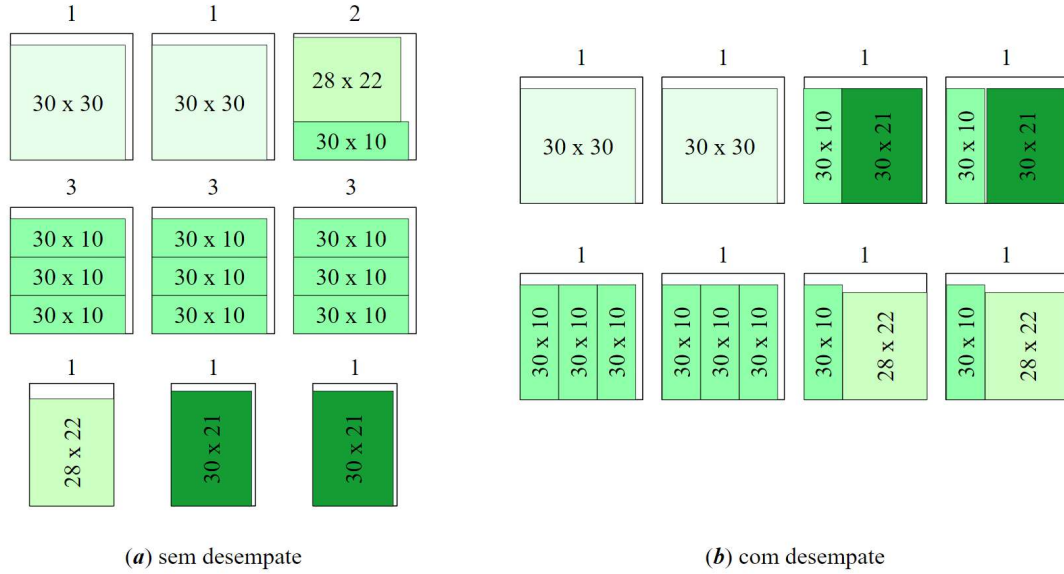


Fonte: o Autor.

A instância 4, representada pela Figura 43 a seguir, é um exemplo de redução do número de níveis causada pelo critério de desempate. Ela apresenta os dois cenários de resolução pelos modelos com rotação: sem o critério de desempate (a) e com o critério (b). Os números na parte superior dos objetos representam os níveis abertos. É possível notar que 16 níveis foram abertos no primeiro cenário, enquanto apenas 8 foram abertos no segundo, representando uma redução de 50% no número de níveis, o que favorece na prática o corte. Como visto anteriormente na Subseção 4.3.1, cada corte economizado se traduz em uma economia média de 30 segundos no tempo de produção, e, portanto,

o desenvolvimento do critério de desempate proposto neste trabalho mostra-se relevante para a otimização da operação da gráfica.

Figura 43: Comparação da quantidade de níveis abertos na instância 4, com e sem critério de desempate.



Fonte: o Autor.

5.3.4 Principais *Insights* e Recomendações Práticas

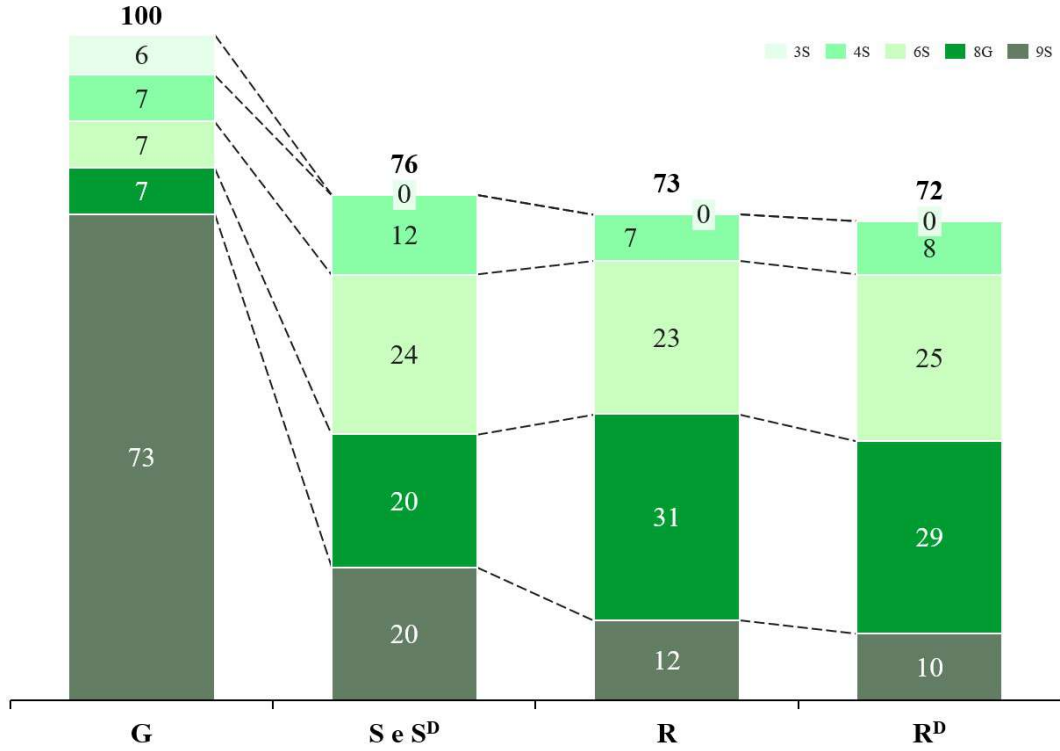
Esta subseção tem como objetivo discutir os principais *insights* obtidos após a análise dos resultados e realizar recomendações práticas para a operação da gráfica a respeito dos objetos escolhidos na montagem dos Planos de Impressão e Corte.

Primeiramente, a operação atual da gráfica, por realizar a montagem dos Planos de Impressão e Corte de forma manual, é ineficiente ao alocar os itens nos objetos, já que os modelos propostos neste trabalho apresentam uma economia média na faixa dos 25-30%. Logo, a primeira recomendação é que a gráfica implemente o modelo R^D em sua operação. Ele se mostrou o mais adequado entre os modelos desenvolvidos para resolver as instâncias reais coletadas, já que possibilita a rotação dos itens (resultando em soluções mais econômicas) e prioriza as soluções com o menor número de prateleiras (resultando em maior eficiência operacional por conta da redução do número de cortes).

Além da sugestão de adoção do modelo R^D , outra recomendação relevante surge a partir da análise comparativa da quantidade total de objetos utilizados. A Figura 44 a seguir compila, por Formato, a quantidade total de objetos utilizados pela gráfica (em

sua solução) e pelos quatro modelos, desconsiderando as instâncias com valores *infeasible*. Os modelos S e S^D , por apresentarem a mesma distribuição de objetos utilizados, foram unificados em apenas uma série.

Figura 44: Comparação da quantidade total de objetos utilizados.



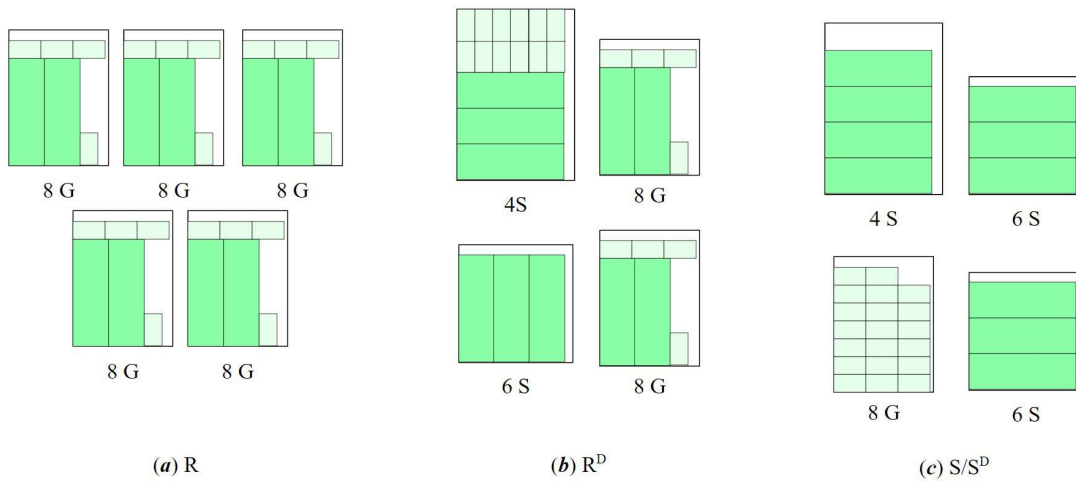
Fonte: o Autor.

É possível notar que os modelos proporcionam uma notável redução no número de objetos, passando de 100 objetos na solução prática da gráfica para 72 no caso do modelo R^D , representando uma redução de aproximadamente 30%. É interessante observar a preferência da gráfica pelo Formato 9 S, enquanto o modelo R^D tende a favorecer os Formatos 8 G e 6 S. Dessa forma, a segunda recomendação é que a gráfica, até a implementação do modelo R^D , considere a utilização dos Formatos 8 G e 6 S durante a montagem dos Planos de Impressão e Corte, dado que esses formatos apresentam os menores custos por metro quadrado, conforme evidenciado na Tabela 3.

Para complementar essa recomendação, uma instância interessante de ser observada é a 17, representada pela Figura 45 a seguir. Ela possui 3 distribuições de objetos, representadas pelos 3 cenários (a , b e c). Nota-se que em (a) foram utilizados cinco objetos do Formato 8 G; em (b) foram utilizados um 3 S, um 4 S e dois 8 G; em (c) foram utilizados um 3 S, dois 4 S e um 8 G. Vale ressaltar que a instância 17 é subótima para o

modelo R^D , como visto na Tabela 12. Na figura, é possível notar que diferentes Formatos produzem arranjos distintos de alocação dos itens em cada um dos modelos, dependendo dos objetivos e restrições aos quais estão submetidos (p.e., rotação dos itens e critério de desempate no número de níveis). Por isso, é importante que a gráfica opte pela escolha dos objetos mais econômicos, e de preferência, em uma alocação que reduza o número total de níveis.

Figura 45: Comparação da quantidade de objetos na instância 17.



Fonte: o Autor.

Por fim, a terceira e última recomendação é que a gráfica mantenha os Formatos 3 S e 4 S (mesmo sendo mais caros em termos de custo por metro quadrado quando comparados aos Formatos 6 S e 8 G), e reconsidere a priorização pelo Formato 9 S. Os Formatos 3 S e 4 S possibilitam à empresa atender à demanda por itens com dimensões superiores a 38 cm (algo que os outros Formatos não permitem). Assim, a sugestão é não eliminar esses formatos devido ao custo elevado, mas utilizá-los apenas em situações apropriadas, idealmente com o suporte do modelo R^D . Quanto ao Formato 9 S, é recomendável que a empresa reconsidere sua priorização, especialmente se não contar com o suporte do modelo proposto neste trabalho, já que este formato apresenta o segundo maior custo por metro quadrado entre as opções disponíveis. Além disso, sua preferência não parece conferir uma vantagem competitiva, como, por exemplo, atender a uma demanda específica (já que ele pode ser alocado dentro dos outros 4 Formatos). Nesse contexto, uma revisão dessa priorização pode contribuir significativamente para uma redução dos custos na operação atual.

6 CONCLUSÕES E PERSPECTIVAS FUTURAS

Este trabalho abordou os diversos desafios enfrentados por uma gráfica na preparação do Plano de Impressão e Corte das ordens de serviço, especialmente para o caso da empresa estudada, que atualmente não possui nenhum tipo de tecnologia capaz de auxiliá-la na montagem desses Planos. O trabalho foi capaz de estabelecer uma relação direta entre o Problema de Corte Bidimensional Guilhotinado com a realidade da companhia, definindo o problema como um 2D-R-MBSBPP, ou seja, o caso bidimensional com itens retangulares do *Multiple Bin Size Bin Packing Problem*, considerando cortes guilhotinados.

De forma geral, o presente trabalho desenvolveu quatro modelos de otimização que tem como objetivo minimizar os custos totais de produção dos pedidos, contribuindo com duas inovações em relação aos modelos originais que foram estendidos: a possibilidade de rotação dos itens e a criação de um critério de desempate de soluções, visando a eficiência operacional e a redução do desperdício de papel. Os modelos foram desenvolvidos e implementados de forma que a sua execução fornece os Planos de Impressão e Corte sugeridos de forma visual, indicando a disposição dos arquivos para impressão dentro das folhas. Por fim, este trabalho também cumpriu o objetivo de comparar os impactos financeiros entre as soluções adotadas na prática pela gráfica com as soluções obtidas pelos modelos propostos.

Após os diversos testes computacionais realizados e as análises comparativas, conclui-se que a adoção do modelo R^D parece ser a alternativa mais adequada para apoiar os processos de impressão e corte. Além de simular a realidade da empresa (permitindo a rotação dos itens), ele introduz o critério de desempate que minimiza o número de cortes e a área desperdiçada de papel. Outra característica relevante é o seu desempenho: ele é capaz de reduzir consideravelmente os custos em relação às soluções adotadas na prática, e, conseqüentemente, a quantidade de objetos utilizados pela empresa.

Porém, o modelo apresenta limitações. A principal é o tempo para rodar uma instância complexa (com múltiplos objetos e itens): nos resultados observados, nota-se que ele não foi capaz de chegar em uma solução ótima em 6 das 20 instâncias com a restrição de

900s. O principal fator observado para essa limitação é o fato do modelo, atualmente, não agrupar objetos idênticos, fazendo com que ele teste todas as combinações de itens nos n objetos do mesmo tipo. Ou seja, entende-se que a inclusão de desigualdades válidas para eliminação de simetrias entre os objetos, assim como feito com os itens, poderia ajudar o modelo desenvolvido a alcançar um tempo de execução menor, e portanto, seria uma primeira perspectiva futura a ser considerada para aperfeiçoar o trabalho desenvolvido.

Adicionalmente, outras possibilidades interessantes podem ser exploradas acerca dos modelos desenvolvidos: uma possível extensão dos modelos para considerar múltiplos períodos (simulando uma operação em que isso ocorra), o desenvolvimento de uma heurística rápida e eficiente que consiga ser competitiva com o Gurobi (principalmente no caso de instâncias mais complexas), e até mesmo a extensão do modelo para considerar sobras reaproveitáveis (para tratar situações em que isso seja relevante na prática), algo que não é relevante na empresa estudada atualmente.

Por fim, a última perspectiva futura mapeada é o desenvolvimento de uma interface gráfica para o arquivo desenvolvido, com o intuito de facilitar a entrada de dados pelo usuário (já que atualmente é necessário utilizar a própria interface do *Jupyter Notebook*).

REFERÊNCIAS

- ABRANTES, R. L. D. A. *Problemas de corte com sobras aproveitáveis e eliminação de simetrias*. Tese (Doutorado em Ciência da Computação) — Instituto de Matemática e Estatística, Universidade de São Paulo, 2012.
- ANDRADE, R.; BIRGIN, E. G.; MORABITO, R. Two-stage two-dimensional guillotine cutting stock problems with usable leftover. *International Transactions in Operational Research*, v. 23, n. 1-2, p. 121–145, 2016.
- BEASLEY, J. E. An exact two-dimensional non-guillotine cutting tree search procedure. *Operations Research*, v. 33, n. 1, p. 49–64, 1985.
- BELLUZZO, L.; MORABITO, R. Otimização nos padrões de corte de chapas de fibra de madeira reconstituída: um estudo de caso. *Pesquisa Operacional*, v. 25, n. 3, p. 391–415, 2005.
- BELOV, G.; SCHEITHAUER, G. A branch-and-cut-and-price algorithm for one-dimensional stock cutting and two-dimensional two-stage cutting. *European Journal of Operational Research*, v. 171, n. 1, p. 85–106, 2006.
- BENNEL, J. A.; OLIVEIRA, J. F. The geometry of nesting problems: A tutorial. *European Journal of Operational Research*, v. 184, n. 2, p. 397–415, 2008.
- BERKEY, J. O.; WANG, P. Y. Two-dimensional finite bin-packing algorithms. *The Journal of the Operational Research Society*, v. 38, n. 5, p. 423, 1987.
- BISCHOFF, E. E.; WÄSCHER, G. Cutting and packing. *European Journal of Operational Research*, v. 84, n. 3, p. 503–505, 1995.
- CARNIERI, C.; MENDOZA, G. A.; GAVINHO, L. G. Solution procedures for cutting lumber into furniture parts. *European Journal of Operational Research*, v. 73, n. 3, p. 495–501, 1994.
- CHUNG, F. R. K.; GAREY, M. R.; JOHNSON, D. S. On packing two-dimensional bins. *SIAM Journal on Algebraic and Discrete Methods*, v. 3, n. 1, p. 66–76, 1982.
- CINTRA, G. F. 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.
- COFFMAN, E. G. et al. Performance bounds for level-oriented two-dimensional packing algorithms. *SIAM Journal on Computing*, v. 9, n. 4, p. 808–826, 1980.
- CUI, Y.; WANG, Z.; LI, J. Exact and heuristic algorithms for staged cutting problems. *Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture*, v. 219, n. 2, p. 201–207, 2005.

- DAS, S.; GOSH, D. Binary knapsack problems with random budgets. *The Journal of the Operational Research Society*, v. 54, n. 9, p. 970–983, 2003.
- DE CANI, P. A note on the two-dimensional rectangular cutting-stock problem. *The Journal of the Operational Research Society*, v. 29, n. 7, p. 703, 1978.
- DELL'AMICO, M.; MARTELLO, S.; VIGO, D. A lower bound for the non-oriented two-dimensional bin packing problem. *Discrete Applied Mathematics*, v. 118, n. 1-2, p. 13–24, 2002.
- DO NASCIMENTO, O. X.; DE QUEIROZ, T.; JUNQUEIRA, L. A mip-cp based approach for two- and three-dimensional cutting problems with staged guillotine cuts. *Annals of Operations Research*, v. 316, n. 2, p. 805–835, 2022.
- DOWSLAND, K. A.; DOWSLAND, W. B. Packing problems. *European Journal of Operational Research*, v. 56, n. 1, p. 2–14, 1992.
- DYCKHOFF, H. A typology of cutting and packing problems. *European Journal of Operational Research*, v. 44, n. 2, p. 145–159, 1990.
- FAINA, L. A survey on the cutting and packing problems. *Bollettino dell Unione Matematica Italiana*, v. 13, n. 4, p. 567–572, 2020.
- FARLEY, A. A. Practical adaptations of the gilmore-gomory approach to cutting stock problems. *OR Spectrum: Quantitative Approaches in Management*, v. 10, n. 2, p. 113–123, 1988.
- FOERSTER, H.; WÄSCHER, G. Simulated annealing for order spread minimization in sequencing cutting patterns. *European Journal of Operational Research*, v. 110, n. 2, p. 272–281, 1998.
- GASS, S. I.; FU, M. C. *Encyclopedia of Operations Research and Management Science*. 3. ed. Boston, United States: Springer, 2013.
- GILMORE, P. C.; GOMORY, R. E. Multistage cutting stock problems of two and more dimensions. *Operations Research*, v. 13, n. 1, p. 94–120, 1965.
- HADJICONSTANTINO, E.; CHRISTOFIDES, N. An exact algorithm for general, orthogonal, two-dimensional knapsack problems. *European Journal of Operational Research*, v. 83, n. 1, p. 39–56, 1995.
- HEMMINKI, J.; LEIPALA, T.; NEVALAINEN, O. On-line packing with boxes of different sizes. *International Journal of Production Research*, v. 36, n. 8, p. 2225–2245, 1998.
- HIFI, M. M.; M'HALLAH, R.; SAADI, T. Approximate and exact algorithms for the double-constrained two-dimensional guillotine cutting stock problem. *Computational Optimization and Applications*, v. 42, n. 2, p. 303–326, 2009.
- HILLIER, F. S.; LIEBERMAN, G. J. *Introduction to Operations Research*. 9. ed. New York, United States: McGraw-Hill, 2010.
- IORI, M. et al. Exact solution techniques for two-dimensional cutting and packing. *European Journal of Operational Research*, v. 289, n. 2, p. 399–415, 2021.

- JUNQUEIRA, L. *Modelos de programação matemática para problemas de carregamento de caixas dentro de contêineres*. Dissertação (Mestrado em Ciências Exatas e da Terra) — Universidade Federal de São Carlos, 2009.
- KOKTEN, E. S.; SEL, C. A cutting stock problem in the wood products industry: a two-stage solution approach. *International Transactions in Operational Research*, v. 29, n. 2, p. 879–907, 2022.
- LEAO, A. A. S. et al. Irregular packing problems: A review of mathematical models. *European Journal of Operational Research*, v. 282, n. 3, p. 803–822, 2020.
- LINS, L.; LINS, S.; MORABITO, R. An n-tet graph approach for non-guillotine packings of n-dimensional boxes into an n-container. *European Journal of Operational Research*, v. 141, n. 2, p. 421–439, 2002.
- LODI, A. *Algorithms for Two-Dimensional Bin Packing and Assignment Problems*. [S.l.], 1999. Disponível em: https://www.dei.unipd.it/~fisch/ricop/tesi/tesi_dottorato-Lodi_1999.pdf.
- LODI, A.; MARTELLO, S.; MONACI, M. Two-dimensional packing problems: A survey. *European Journal of Operational Research*, v. 141, n. 2, p. 241–252, 2002.
- LODI, A.; MARTELLO, S.; VIGO, D. Recent advances on two-dimensional bin packing problems. *Discrete Applied Mathematics*, v. 123, n. 1-3, p. 379–396, 2002.
- LODI, A.; MARTELLO, S.; VIGO, D. Models and bounds for two-dimensional level packing problems. *Journal of Combinatorial Optimization*, v. 8, n. 3, p. 363–379, 2004.
- LODI, A.; MONACI, M. Integer linear programming models for 2-staged two-dimensional knapsack problems. *Mathematical Programming*, v. 94, n. 2-3, p. 257–278, 2003.
- MARTELLO, S. Knapsack, packing and cutting. *Information Systems and Operational Research*, v. 32, n. 3, p. 121–123, 1994.
- MORABITO, R.; ARENALES, M. N. Staged and constrained two-dimensional guillotine cutting problems: An and/or-graph approach. *European Journal of Operational Research*, v. 94, n. 3, p. 548–560, 1996.
- MORABITO, R.; ARENALES, M. N. Optimizing the cutting of stock plates in a furniture company. *International Journal of Production Research*, v. 38, n. 12, p. 2725–2742, 2000.
- MORABITO, R.; ARENALES, M. N.; YANASSE, H. H. Special issue on cutting, packing and related problems. *International Transactions in Operational Research*, v. 16, n. 6, p. 659–659, 2009.
- MORABITO, R.; GARCIA, V. The cutting stock problem in a hardboard industry: A case study. *Computers & Operations Research*, v. 25, n. 6, p. 469–485, 1998.
- MOSTAJABDAVEH, M.; SALMAN, F. S.; TAHMASBI, N. Two dimensional guillotine cutting stock and scheduling problem in printing industry. *Computers and Operations Research*, v. 148, n. 106014, p. 106014, 2022.

- MURTHY, P. R. *Operations Research*. 2. ed. New Delhi, India: New Age International, 2007.
- NASCIMENTO, D. N. D. *O problema de corte bidimensional com sobras aproveitáveis e incerteza na demanda*. Tese (Doutorado em Engenharia de Produção) — Faculdade de Engenharia da Universidade Estadual Paulista (UNESP), 2022.
- NONÁS, S. L.; THORSTENSON, A. A combined cutting-stock and lot-sizing problem. *European Journal of Operational Research*, v. 120, n. 2, p. 327–342, 2000.
- OLIVEIRA, O.; GAMBOA, D.; SILVA, E. An introduction to the two-dimensional rectangular cutting and packing problem. *International Transactions in Operational Research*, 2022.
- PIETROBUONI, E. *Two-dimensional bin packing problem with guillotine restrictions*. [S.l.], 2015. Disponível em: <http://dx.doi.org/10.6092/unibo/amsdottorato/6810>.
- RIEHME, J.; SCHEITHAUER, G.; TERNO, J. The solution of two-stage guillotine cutting stock problems having extremely varying order demands. *European Journal of Operational Research*, v. 91, n. 3, p. 543–552, 1996.
- SCHEITHAUER, G. *Introduction to cutting and packing optimization: Problems, modeling approaches, solution methods*. 1. ed. Basel, Switzerland: Springer International Publishing, 2017.
- SCHIERMEYER, I. Reverse-fit: A 2-optimal algorithm for packing rectangles. *Em: Algorithms — ESA '94. Berlin, Heidelberg: Springer Berlin Heidelberg*, p. 290–299, 1994.
- SLEATOR, D. D. K. A 2.5 times optimal algorithm for packing in two dimensions. *Information Processing Letters*, v. 10, n. 1, p. 37–40, 1980.
- STEINBERG, A. A strip-packing algorithm with absolute performance bound 2. *SIAM Journal on Computing*, v. 26, n. 2, p. 401–409, 1997.
- SWEENEY, P. E.; PATERNOSTER, E. R. Cutting and packing problems: A categorized, application-orientated research bibliography. *The Journal of the Operational Research Society*, v. 43, n. 7, p. 691–706, 1992.
- TAHA, H. A. *Operations Research: an Introduction*. 10. ed. Fayetteville, United States: Pearson, 2017.
- TELGEN, J. Identifying redundant constraints and implicit equalities in systems of linear constraints. *Management Science*, v. 29, n. 10, p. 1209–1222, 1983.
- VANDERBECK, F. Exact algorithm for minimising the number of setups in the one-dimensional cutting stock problem. *Operations Research*, v. 48, n. 6, p. 915–926, 2000.
- WANG, P. Y.; WÄSCHER, G. Cutting and packing. *European Journal of Operational Researchs*, v. 141, n. 2, p. 239–240, 2002.
- WINSTON, W. L. *Operations Research: applications and algorithms*. 4. ed. Belmont, Canada: Brooks/Cole - Thomson Learning, 2004.

- WÄSCHER, G. An lp-based approach to cutting stock problems with multiple objectives. *European Journal of Operational Research*, v. 44, n. 2, p. 175–184, 1990.
- WÄSCHER, G.; HAUSSNER, H.; SCHUMANN, H. An improved typology of cutting and packing problems. *European Journal of Operational Research*, v. 183, n. 3, p. 1109–1130, 2007.
- YANASSE, H. H. On a pattern sequencing problem to minimize the maximum number of open stacks. *European Journal of Operational Research*, v. 100, n. 3, p. 454–463, 1997.
- YANASSE, H. H.; KATSURAYAMA, D. M. Checkerboard pattern: proposals for its generation. *International Transactions in Operational Research*, v. 12, n. 1, p. 21–45, 2005.
- YANASSE, H. H.; ZINOBER, A. S. I.; HARRIS, R. G. Two-dimensional cutting stock with multiple stock sizes. *The Journal of the Operational Research Society*, v. 42, n. 8, p. 673–683, 1991.
- YUEN, B. J. Improved heuristics for sequencing cutting patterns. *European Journal of Operational Research*, v. 87, n. 1, p. 57–64, 1995.