

Método de Benders Lógico para o *Two-Dimensional Bin Packing Problem*

Integrante(s): Arthur Henrique Sousa Cruz¹, Walison Adrian de Oliveira².

E-mail(s): arthurhscruz@usp.br¹, walisonadrian201@usp.br².

Resumo

Diversos setores industriais enfrentam desafios significativos ao cortar peças retangulares com dimensões específicas exigidas pelos seus clientes. Nesse cenário, grandes placas de matéria-prima, como painéis de tecido e chapas de metal, possuem dimensões fixas, e é necessário cortar todos os itens de maneira a minimizar o número total de placas utilizadas. Esse problema, conhecido como *Two-Dimensional Bin Packing Problem* (2D-BPP), é amplamente estudado na literatura e apresenta diferentes tipologias. O 2D-BPP é NP-difícil, uma vez que o *One-Dimensional Bin Packing Problem* (1D-BPP) é um caso particular do 2D-BPP. Neste trabalho, propõe-se uma decomposição lógica de Benders para o 2D-BPP. A separação entre o problema mestre e os subproblemas foi realizada de forma que o problema mestre é responsável por determinar a alocação das peças a serem cortadas em cada placa. Os subproblemas verificam a viabilidade de realizar os cortes das peças alocadas em cada placa. Experimentos computacionais foram conduzidos com instâncias da literatura e o método foi comparado ao modelo matemático padrão do problema. A partir dos resultados, foi possível analisar o comportamento da decomposição proposta. Observou-se que, em instâncias pequenas e médias, houve dificuldade em comprovar a otimalidade. Em instâncias grandes, identificou-se um potencial para encontrar soluções factíveis, enquanto o modelo matemático padrão esgotou a memória.

Palavras-chave: Decomposição Lógica de Benders. Decomposição de Benders. Single Bin Size Bin Packing Problem.

1 Introdução e trabalhos relacionados

Diversos setores industriais, como o têxtil e o siderúrgico, enfrentam desafios significativos ao cortar peças retangulares com dimensões específicas exigidas por seus clientes. Nesse cenário, as placas de matéria-prima, como painéis de tecido e chapas de metal, têm dimensões fixas. Todas as peças devem ser cortadas dessas placas e o desafio é minimizar o número total de placas utilizadas, reduzindo o desperdício de material, aumentando a lucratividade e a sustentabilidade das indústrias, como destacado em [Dyckhoff \(1990\)](#).

Esse problema, conhecido como *Two-Dimensional Bin Packing Problem* (2D-BPP), é amplamente estudado na literatura e apresenta diferentes tipologias. Na tipologia proposta por [Lodi et al. \(1999\)](#), o 2D-BPP é classificado como 2BP|O|F, considerando variantes que permitem a rotação ortogonal dos itens ou cortes guilhotinados. Posteriormente, [Wäscher et al. \(2007\)](#) referiram-se ao 2D-BPP como *Single Bin Size Bin Packing Problem* (SBSBPP), destacando a característica única das dimensões dos recipientes.

Como evidenciado em [Iori et al. \(2021\)](#), o 2D-BPP é NP-difícil, pois o *One-Dimensional Bin Packing Problem* (1D-BPP), reconhecido como NP-difícil ([Garey and Johnson \(1979\)](#)), é um caso particular do 2D-BPP. Considerando um conjunto de instâncias do 2D-BPP onde as placas têm comprimento 1 e largura W e todos os itens têm comprimento também igual a 1, o problema torna-se unidimensional e pode ser tratado como uma instância particular do 1D-BPP.

Embora muitos algoritmos exatos e heurísticos tenham sido desenvolvidos para resolver o 2D-BPP, o problema continua sendo extremamente complexo na prática. Instâncias com apenas 40 itens ainda não foram resolvidas até a otimalidade comprovada. O estado da arte do problema é o algoritmo proposto em [Pisinger and Sigurd \(2007\)](#), que utiliza uma decomposição dual do 2D-BPP, baseada na decomposição de Dantzig-Wolfe. Neste método, o problema é modelado como um problema de cobertura de conjuntos, onde cada coluna representa um recipiente 2D viável de uma placa contendo um ou mais itens. Devido ao grande número de possíveis recipientes, os autores implementaram um algoritmo de geração de colunas. Este algoritmo resolve, inicialmente, o problema mestre com um conjunto reduzido de colunas e, em seguida, verifica se existem colunas com custo reduzido negativo. Se tais colunas forem encontradas, elas são adicionadas ao problema mestre. O processo é repetido até que um conjunto de recipientes que abranja todos os itens e exija o menor número possível de placas.

No trabalho de [Cid-Garcia and Rios-Solis \(2020\)](#), é apresentado um novo método de duas etapas, denominado *Positions and Covering* (P&C), visando obter uma solução exata para o 2D-BPP. A primeira etapa do método consiste na geração de um conjunto de posições possíveis para cada item em uma malha discretizada. A segunda etapa envolve a resolução de um novo modelo de cobertura de conjuntos para determinar o corte ótimo para cada placa. O método P&C avalia a viabilidade das soluções ao considerar a versão de decisão do 2D-BPP e um limite inferior no número de placas necessárias. O estudo também propõe três conjuntos de desigualdades válidas e compara o desempenho da nova formulação matemática com a inclusão de cada desigualdade, avaliando soluções para diferentes tamanhos de itens e malhas e obtendo resultados satisfatórios em instâncias de porte não muito grandes.

Em [F. Parreño and Tamarit \(2010\)](#), é proposto um algoritmo GRASP (*Greedy Randomized Adaptive Search Procedure*) para abordar problemas de empacotamento tanto bidimensionais quanto tridimensionais. A fase construtiva do algoritmo é fundamentada em uma heurística previamente desenvolvida pelos autores para o problema de carregamento de contêineres, enquanto a fase de aprimoramento incorpora novos movimentos, que são testados e combinados em uma estrutura VND (*Variable Neighborhood Descent*). Nos testes computacionais, foi utilizado um total de 500 instâncias, variando entre 20, 40, 60, 80 e 100 itens. Os resultados apresentados mostram que o algoritmo híbrido GRASP/VND, além de ser simples e eficiente, demonstra um bom desempenho computacional, alcançando soluções de qualidade igual ou superior às obtidas pelos métodos mais complexos da literatura.

Côté et al. (2019) abordam o 2D-BPP utilizando uma decomposição combinatória de Benders baseada em um modelo simplificado, no qual os itens e placas bidimensionais são representados apenas por suas áreas, e cortes infactíveis são tratados com *no-good cuts*. Apesar da decomposição ser relativamente simples, ela é aprimorada com técnicas de pré-processamento, desigualdades válidas, métodos de estimativa inferior e algoritmos avançados para gerar cortes mais eficazes. O algoritmo desenvolvido demonstrou um desempenho superior em comparação com algoritmos anteriores da literatura, resolvendo pela primeira vez algumas instâncias em aberto. No entanto, os resultados apresentados ainda mostram instâncias, incluindo algumas com apenas 40 itens, que permanecem sem solução ótima encontrada.

Neste trabalho, propõe-se uma decomposição lógica de Benders para o 2D-BPP. A separação entre problema mestre e subproblemas foi realizada de forma que o primeiro seja responsável por determinar a alocação das peças a serem cortadas em cada placa, enquanto os subproblemas verificam se é possível realizar os cortes de todas as peças alocadas em cada placa. O objetivo deste trabalho é analisar o comportamento da decomposição proposta e verificar seu potencial para a resolução do 2D-BPP. Para isso, experimentos computacionais foram conduzidos com instâncias da literatura e o método foi comparado ao modelo matemático padrão do problema. A partir dos resultados, percebeu-se que, para instâncias pequenas e médias, há dificuldade de comprovar otimalidade. Para instâncias grandes, observou-se haver potencial para encontrar soluções factíveis, enquanto o modelo matemático padrão esgotou a memória.

A seguir, na Seção 2, é definido o problema de empacotamento de retângulos estudado, além de ser formulado um modelo matemático para resolvê-lo. Na Seção 3, a decomposição lógica de Benders proposta é descrita, junto aos modelos do problema mestre e subproblemas. Na Seção 4, são apresentados os experimentos computacionais realizados com o *solver Gurobi* para resolver os modelos propostos em instâncias da literatura, bem como os resultados obtidos. Por fim, na Seção 5, as considerações finais e propostas de trabalhos futuros com base na análise dos resultados são apresentadas.

2 Definição do problema

Para o 2D-BPP, considera-se um conjunto \mathcal{P} de placas retangulares idênticas, com comprimento $L \in \mathbb{Z}^+$ e largura $W \in \mathbb{Z}^+$. No problema, assume-se um número ilimitado de placas disponíveis, contudo, neste trabalho, utiliza-se uma quantidade limitada que é suficientemente grande. Assim, o objetivo é minimizar o número total de placas necessárias para cortar $m \in \mathbb{Z}^+$ peças, de dimensões $l_i \in \mathbb{Z}^+$ e $w_i \in \mathbb{Z}^+$, com $i = 1, \dots, m$. Deve-se garantir que as peças não extrapolem as bordas das placas e que não haja sobreposição entre as peças em uma mesma placa.

Na Figura 1, é exemplificada uma instância do 2D-BPP, na qual todas as seis peças à esquerda devem ser cortadas nas placas à direita. Uma das possíveis soluções ótimas é ilustrada pela Figura 2, que apresenta uma solução que utiliza apenas duas placas.

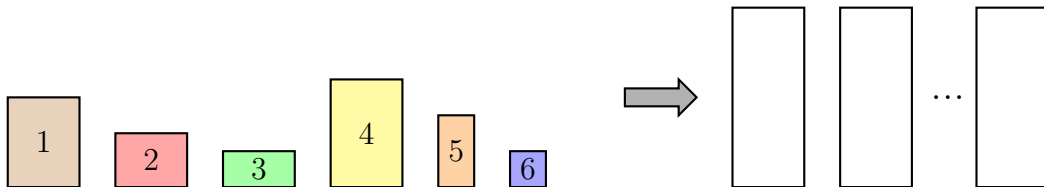


Figura 1: Exemplo de uma instância do 2D-BPP.

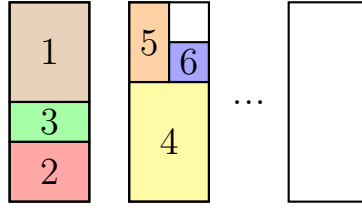


Figura 2: Exemplo de uma solução ótima para a instância apresentada na Figura 1.

Este problema pode ser formulado como um problema de otimização linear inteira. Para isso, apresenta-se a seguir uma modelagem baseada no modelo descrito em Arenales et al. (2015). No decorrer deste trabalho, esse modelo é referido como Modelo Padrão. Nessa abordagem, assume-se que as placas foram discretizadas em malhas de tamanho unitário, que os cortes são ortogonais a um dos lados da placa e a orientação das peças é fixa. Isso significa que uma peça com comprimento l_i e largura w_i é considerada distinta de uma peça com comprimento w_i e largura l_i .

Assim, utiliza-se para o modelo os seguintes parâmetros: o conjunto de placas é representado por \mathcal{P} , em que cada placa possui comprimento $L \in \mathbb{Z}^+$ e largura $W \in \mathbb{Z}^+$. A área das placas é indicada por $Q \in \mathbb{Z}^+$. O parâmetro $m \in \mathbb{Z}^+$ representa a quantidade total de peças, com $l_i \in \mathbb{Z}^+$ e $w_i \in \mathbb{Z}^+$ sendo o comprimento e a largura da peça i , respectivamente, para $i = 1, \dots, m$. A área da peça i é denotada por $A_i \in \mathbb{Z}^+$. Os conjuntos X e Y contêm os pontos referentes às discretizações do comprimento e da largura da placa, respectivamente. O parâmetro $a_{ilwrs} \in \{0, 1\}$ assume valor 1 se o ponto (r, s) é excluído quando o canto inferior esquerdo de uma peça i é cortado no ponto (l, w) , com $i = 1, \dots, m$, $l = 0, \dots, L - l_i$, $w = 0, \dots, W - w_i$, $r \in X$ e $s \in Y$. Caso contrário, a_{ilwrs} assume 0.

Na Figura 3, é ilustrado um item i com dimensões $l_i = 4$ e $w_i = 3$, sendo cortado com seu canto inferior esquerdo posicionado no ponto (l, w) , excluindo o ponto (r, s) , em uma placa discretizada com dimensões $L = 9$ e $W = 7$.

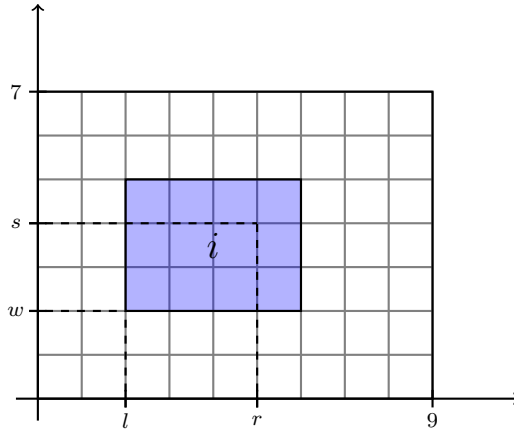


Figura 3: Ilustração de um corte de peça na malha. Adaptado de Arenales et al. (2015).

Como variáveis de decisão, define-se $x_{ijlw} \in \{0, 1\}$, cujo valor é 1 se, e somente se, o canto inferior esquerdo de uma peça i , é cortado na placa j no ponto (l, w) , tal que $i = 1, \dots, m$, $j \in \mathcal{P}$, $l = 0, \dots, L - l_i$ e $w = 0, \dots, W - w_i$. As variáveis são definidas somente para pontos que não geram cortes fora da placa, garantindo que todas as peças sejam inteiramente cortadas. As variáveis $z_j \in \{0, 1\}$, por sua vez, assumem valor 1 se a placa $j \in \mathcal{P}$ é utilizada, e 0 caso contrário.

Desse modo, o modelo proposto é descrito da seguinte maneira:

$$\text{Min.} \quad \sum_{j \in \mathcal{P}} j z_j \quad (1)$$

$$\text{s. a} \quad \sum_{i=1}^m \sum_{l=0}^{L-l_i} \sum_{w=0}^{W-w_i} a_{ilwrs} x_{ijlw} \leq z_j, \quad r \in X, s \in Y, j \in \mathcal{P}, \quad (2)$$

$$\sum_{j \in \mathcal{P}} \sum_{l=0}^{L-l_i} \sum_{w=0}^{W-w_i} x_{ijlw} = 1, \quad i = 1, \dots, m, \quad (3)$$

$$\sum_{i=1}^m \sum_{l=0}^{L-l_i} \sum_{w=0}^{W-w_i} A_i x_{ijlw} \leq Q, \quad j \in \mathcal{P}, \quad (4)$$

$$z_j \leq \sum_{i=1}^m \sum_{l=0}^{L-l_i} \sum_{w=0}^{W-w_i} x_{ijlw}, \quad j \in \mathcal{P}, \quad (5)$$

$$z_j \leq z_{j-1}, \quad j \in \mathcal{P} \setminus \{1\}, \quad (6)$$

$$x_{ijlw} \in \{0, 1\}, \quad i = 1, \dots, m, l = 0, \dots, L - l_i, \quad (7)$$

$$w = 0, \dots, W - w_i, j \in \mathcal{P},$$

$$z_j \in \{0, 1\}, \quad j \in \mathcal{P}. \quad (8)$$

A função objetivo (1) visa minimizar a quantidade de placas necessárias para cortar todos os itens, ao mesmo tempo que visa reduzir a simetria, priorizando as placas de menores índices. As restrições (2) garantem que qualquer ponto (r, s) de uma placa j seja excluído por, no máximo, uma peça i , evitando a sobreposição das peças. As restrições (3) asseguram que todos os itens sejam cortados em alguma placa j . As restrições (4) apresentam cortes baseados nas áreas dos itens, impedindo que a soma das áreas das peças cortadas em uma mesma placa ultrapasse a área da placa. As restrições (5) - (6) apresentam cortes de simetria para reduzir o número de soluções factíveis do problema que possuem o mesmo valor na função objetivo. Por fim, as restrições (7) - (8) indicam o domínio das variáveis.

3 Decomposição lógica de Benders

Para solucionar o problema, foi proposta uma decomposição lógica de Benders. Neste trabalho, para simplificação, essa abordagem é referida apenas como Decomposição de Benders. A separação entre problema mestre e subproblemas foi realizada de forma que o primeiro seja responsável por determinar a alocação das peças a serem cortadas em cada placa, enquanto os subproblemas verificam se é possível realizar os cortes de todas as peças alocadas em cada placa. Assim, o problema mestre visa minimizar o número de placas necessárias para os cortes, e os subproblemas tornam-se problemas de factibilidade. Os subproblemas são executados sempre que uma solução factível é encontrada para o problema mestre.

A seguir, na Seção 3.1, é apresentada a modelagem matemática do problema mestre e os cortes de Benders aplicados. Na Seção 3.2, é apresentada a modelagem de um subproblema.

3.1 Problema mestre

O objetivo do problema mestre é minimizar o número de placas utilizadas para realizar os cortes de todas as peças. Para fins práticos, define-se um conjunto \mathcal{P} contendo um número suficientemente grande de placas homogêneas. Dessa forma, a decisão a ser tomada é se a peça i , com $i = 1, \dots, m$, deve ser cortada na placa $j \in \mathcal{P}$. Cada placa possui uma área $Q \in \mathbb{Z}^+$, enquanto cada peça possui uma área $A_i \in \mathbb{Z}^+$, com $i = 1, \dots, m$.

Como variáveis de decisão, tem-se $b_{ij} \in \{0, 1\}$, que assume o valor 1 se a peça i , com $i = 1, \dots, m$, for cortada em uma placa $j \in \mathcal{P}$, e 0 caso contrário. Além disso, para a contagem de

placas utilizadas, define-se, para toda placa $j \in \mathcal{P}$, a variável $\rho_j \in \{0, 1\}$, que assume o valor 1 se a placa j foi utilizada, e 0 caso contrário.

Quando uma solução factível é encontrada para o problema mestre, é gerado um conjunto $\mathbb{S} = \{\bar{b}_{ij} \mid i = 1, \dots, m, j \in \mathcal{P}\}$, com \bar{b}_{ij} representando os valores das variáveis da solução encontrada, ou seja, $\bar{b}_{ij} = 1$ se a peça i foi cortada na placa j , e 0 caso contrário. Com esses valores, é criado um subproblema para cada placa $j \in \mathcal{P}$ para verificar se é possível cortar todas as peças alocadas. Após a verificação de factibilidade, para todo subproblema de uma placa $p \in \mathcal{P}$, cuja solução seja infactível, obtém-se o conjunto $\mathcal{V}_p \in \mathcal{K}$, com $\mathcal{V}_p = \{\bar{b}_{ip} \mid i = 1, \dots, m\}$.

Com as definições estabelecidas, apresenta-se a seguinte modelagem matemática para a resolução do problema mestre:

$$\text{Min.} \quad \sum_{j \in \mathcal{P}} j \rho_j \quad (9)$$

$$\text{s. a} \quad \sum_{j \in \mathcal{P}} b_{ij} = 1, \quad i = 1, \dots, m, \quad (10)$$

$$\sum_{i=1}^m A_i b_{ij} \leq Q \rho_j, \quad j \in \mathcal{P}, \quad (11)$$

$$\sum_{\bar{b}_{ip} \in \mathcal{V}_p \mid \bar{b}_{ip}=1} (1 - b_{ij}) \geq 1, \quad j \in \mathcal{P}, \quad \mathcal{V}_p \in \mathcal{K}, \quad (12)$$

$$\rho_j \leq \sum_{i=1}^m b_{ij}, \quad j \in \mathcal{P}, \quad (13)$$

$$\rho_j \leq \rho_{j-1}, \quad j \in \mathcal{P} \setminus \{1\}, \quad (14)$$

$$\rho_j \in \{0, 1\}, \quad j \in \mathcal{P}, \quad (15)$$

$$b_{ij} \in \{0, 1\}, \quad i = 1, \dots, m, \quad j \in \mathcal{P}. \quad (16)$$

A função objetivo (9) é análoga à função objetivo do Modelo Padrão, definido na Seção 2. Ela minimiza o número de placas utilizadas, ao mesmo tempo que o índice da variável é utilizado para reduzir o número de soluções simétricas. As restrições (10) asseguram que cada peça seja cortada em uma única placa. As restrições (11) garantem que, se uma peça for alocada em uma placa, então essa placa será contabilizada na função objetivo. Além disso, essas restrições adicionam um corte ao limitar a soma das áreas das peças à área da placa.

As restrições (12) representam os cortes de Benders adicionados quando é encontrado uma solução infactível no subproblema. São criados cortes monótonos para cada subproblema \mathcal{V}_p , considerando que, ao adicionar mais peças ao subproblema, ele continuaria infactível. Um detalhe importante a ser destacado é que, como as placas são homogêneas, o corte gerado para o subproblema com solução infactível de uma placa $p \in \mathcal{P}$ é replicado para toda placa $j \in \mathcal{P} \setminus \{p\}$.

As restrições (13)-(14) são cortes adicionados para evitar soluções simétricas. Por fim, as restrições (15)-(16) definem o domínio das variáveis.

3.2 Subproblema

Após encontrar uma solução factível para o problema mestre, é necessário verificar se é possível cortar as \bar{m} peças alocadas em uma determinada placa. Assim, deve-se garantir que cada peça i , com $i = 1, \dots, \bar{m}$, de comprimento $l_i \in \mathbb{Z}^+$ e largura $w_i \in \mathbb{Z}^+$, possa ser inteiramente cortada, sem sobreposição, em uma placa com comprimento $L \in \mathbb{Z}^+$ e largura $W \in \mathbb{Z}^+$. Assim como no Modelo Padrão definido na Seção 2, a placa é discretizada, de modo que os conjuntos X e Y contêm os pontos referentes às discretizações do comprimento e da largura da placa, respectivamente. O parâmetro $c \in \mathbb{R}$ é um valor constante qualquer. Na implementação realizada, assume-se $c = 0$. Além disso, o parâmetro a_{ilwrs} é utilizado para verificar a não sobreposição das peças. Dessa forma,

a_{ilwrs} assume o valor 1 se o ponto (r, s) é excluído quando o canto inferior esquerdo da peça i é posicionado no ponto (l, w) da placa, e 0 caso contrário, com $i = 1, \dots, m$, $j \in \mathcal{P}$, $l = 0, \dots, L - l_i$ e $w = 0, \dots, W - w_i$.

Têm-se as variáveis de decisão $\mu_{ilw} \in \{0, 1\}$, $i = 1, \dots, m$, $l = 0, \dots, L - l_i$ e $w = 0, \dots, W - w_i$. O valor de cada variável é 1 se, e somente se, o canto inferior esquerdo da peça i é cortado na posição (l, w) da placa. As variáveis são definidas somente para pontos que não geram cortes fora da placa, garantindo que todas as peças sejam inteiramente cortadas. As variáveis μ_{ilw} são análogas às variáveis x_{ijlw} do Modelo Padrão, porém para apenas uma placa.

Com as definições apresentadas, pode-se construir o seguinte modelo matemático para representar um subproblema da Decomposição de Benders proposta:

$$\text{Min.} \quad c \quad (17)$$

$$\text{s. a} \quad \sum_{l=0}^{L-l_i} \sum_{w=0}^{W-w_i} \mu_{ilw} = 1, \quad i = 1, \dots, \bar{m}, \quad (18)$$

$$\sum_{i=1}^{\bar{m}} \sum_{l=0}^{L-l_i} \sum_{w=0}^{W-w_i} a_{ilwrs} \mu_{ilw} \leq 1, \quad i = 1, \dots, \bar{m}, \quad r \in X, \quad s \in Y, \quad (19)$$

$$\mu_{ilw} \in \{0, 1\}, \quad i = 1, \dots, \bar{m}, \quad l = 0, \dots, L - l_i, \quad w = 0, \dots, W - w_i. \quad (20)$$

A função objetivo (17) minimiza uma constante c , pois o intuito do modelo é verificar se existe uma solução factível para o subproblema. As restrições (18) definem que toda peça deve ser cortada. As restrições (19) evitam a sobreposição de peças ao impedir que um ponto (r, s) seja cortado mais de uma vez. Por fim, as restrições (20) definem o domínio das variáveis.

4 Experimentos computacionais

Para avaliar o desempenho da Decomposição de Benders proposta, foram realizados experimentos computacionais com instâncias da literatura. A comparação dos resultados foi feita com o Modelo Padrão proposto na Seção 2. O código-fonte, com instruções para execução e reprodução dos experimentos, pode ser encontrado no seguinte repositório: <https://github.com/thuza/x/Benders-2D-Cutting>. Os resultados dos experimentos também podem ser encontrados no repositório.

Na sequência, na Seção 4.1, são descritos o ambiente computacional e o *hardware* utilizados nos experimentos. Na Seção 4.2, são detalhadas as instâncias e o cálculo de algumas das métricas utilizadas na avaliação dos resultados. Em seguida, na Seção 4.3, os resultados obtidos para as instâncias pequenas são apresentados. Posteriormente, na Seção 4.4, são discutidos os resultados para as instâncias médias, enquanto na Seção 4.5 são detalhados os resultados para as instâncias grandes. Por fim, na Seção 4.6, o comportamento da Decomposição de Benders considerando o problema mestre e subproblema é analisado com base na quantidade de nós explorados e tempo em *callback*.

4.1 Ambiente computacional

Tanto o Modelo Padrão quanto a Decomposição de Benders foram implementadas em *Python* na versão 3.10.12. Para a execução dos modelos matemáticos, foi utilizado o *solver Gurobi* versão 11.01 em sua configuração padrão. A implementação do modelo foi feita através do pacote *gurobipy*. O programa foi executado no sistema operacional *Ubuntu* versão 22.04 com um processador Intel *i7* de 12ª geração e 16 GB de memória RAM. Cada execução foi limitada a 30 minutos.

4.2 Instâncias e definições de métricas de avaliação

Foram escolhidas 15 instâncias classificadas por tamanho, sendo 5 pequenas, 5 médias e 5 grandes. Na Tabela 1, são apresentadas as dimensões da placa e a quantidade de peças de cada uma das classes das instâncias. Todas as instâncias foram selecionadas da biblioteca 2DPackLib (Iori et al., 2022). O número de placas foi calculado como $|\mathcal{P}| = \left\lceil 1,2 \cdot \left\lceil \frac{\sum_{i=1}^m A_i}{Q} \right\rceil \right\rceil$, ou seja, a quantidade de placas $|\mathcal{P}|$ é igual ao teto de 20% a mais do teto do somatório das áreas dos itens dividido pela área da placa. A discretização da placa e das peças foi de uma unidade da medida utilizada em cada instância.

Tabela 1: Dimensões da placa e quantidade de peças a serem cortadas em cada classe de instância.

Classe	Dimensões da placa ($L \times W$)	Quantidade de peças (m)
Pequena	10×10	60
Média	30×30	60
Grande	40×40	60

Cada instância foi executada uma vez. Observa-se que os valores das funções objetivo do Modelo Padrão e da Decomposição de Benders podem diferir em relação ao número de placas obtidas na solução, visto que há um peso adicional na função que visa evitar simetria na alocação de peças nas placas. Consequentemente, o valor do *gap* gerado pelo *Gurobi* não reflete o *gap* do problema. Portanto, o *gap* analisado foi calculado como $gap = \frac{|\theta_P - \theta_D|}{|\theta_P|}$, de forma que θ_P é o número de placas obtidas na melhor solução factível encontrada e θ_D é o número de placas do melhor limitante dual obtido.

Tanto o limitante dual quanto o número de placas foram calculados a partir da função objetivo obtida pelo *Gurobi*, pois as restrições de simetria de placa (consulte as restrições (5) e (6) da Seção 2 e as restrições (13) e (14) da Seção 3.1) obrigam que as placas sejam escolhidas em ordem crescente de índices. Dessa forma, θ_P (menor número de placas encontrado) e θ_D (melhor limitante dual) assumem, respectivamente, valores tal que $\sum_{\theta=1}^{\theta_P} \theta$ seja igual ao melhor valor de função objetivo obtido pelo *Gurobi*, e $\sum_{\theta=1}^{\theta_D} \theta$ seja o valor do melhor limitante dual obtido pelo *Gurobi*.

4.3 Resultados das instâncias pequenas

Na Tabela 2, são apresentados os resultados obtidos para as instâncias da classe pequena. Na primeira coluna, é listado o nome de cada instância. Nas colunas seguintes, da esquerda para a direita, são apresentados, para o Modelo Padrão, o número de placas da melhor solução encontrada, o *gap* (em porcentagem) e o tempo de execução (em segundos). Por fim, nas três últimas colunas, são apresentados, da esquerda para a direita, o número de placas da melhor solução encontrada, o *gap* (em porcentagem) e o tempo gasto (em segundos) na execução de cada instância pela Decomposição de Benders. Em negrito, estão marcados os melhores valores para o *gap* e os melhores valores para o número de placas.

Como pode ser visto, o Modelo Padrão encontrou o valor ótimo para todas as instâncias. No caso da Decomposição de Benders, três instâncias alcançaram o valor da solução ótima, contudo, não foi comprovada a otimalidade. Além disso, para uma das instâncias, o número de placas foi de apenas um a mais do que a solução ótima.

O resultado obtido pela Decomposição de Benders para a instância *cl_01_060_05*, não foi considerado ótimo, pois, apesar de ter alcançado o valor ótimo, não foi possível coletar os valores das variáveis dos subproblemas. Isso ocorreu por um dos dois motivos: (i) uma solução foi encontrada, porém, o tempo limite de execução foi alcançado antes da verificação de factibilidade dos subproblemas; ou (ii) a factibilidade da solução foi verificada, mas a execução do modelo foi interrompida antes do armazenamento dos valores das soluções dos subproblemas. Ao analisar o histórico de

Tabela 2: Resultados obtidos para as instâncias pequenas pelo Modelo Padrão e pela Decomposição de Benders.

Instância	Padrão			Benders		
	Placas	Gap (%)	Tempo (s)	Placas	Gap (%)	Tempo (s)
cl.01_060_01	23	0	4	24	12,50	1800
cl.01_060_02	19	0	6	19	5,26	1800
cl.01_060_03	21	0	1	21	9,52	1800
cl.01_060_04	22	0	2	22	13,64	1800
cl.01_060_05	19	0	5	19*	10,53	1800

execução gerado pelo *Gurobi*, é possível verificar que nenhuma outra solução factível foi encontrada, indicando dificuldade em gerar uma solução factível. Dessa forma, há maior probabilidade de que a situação (i) tenha ocorrido.

4.4 Resultados das instâncias médias

Na Tabela 3, são apresentados os resultados obtidos para as instâncias da classe média. Seu formato segue o mesmo padrão da Tabela 2 descrita na Seção 4.3.

Tabela 3: Resultados obtidos para as instâncias médias pelo Modelo Padrão e pela Decomposição de Benders.

Instância	Padrão			Benders		
	Placas	Gap (%)	Tempo (s)	Placas	Gap (%)	Tempo (s)
cl.02_060_01	3	0	86	3	0	10
cl.02_060_02	2	0	1391	2	0	13
cl.02_060_03	3	0	68	3	0	60
cl.02_060_04	3	0	63	3	33,33	1800
cl.02_060_05	2	0	105	2	0	197

Assim como para as instâncias pequenas, o Modelo Padrão alcançou o valor ótimo para todas as instâncias. Apesar do maior tempo de execução, quatro das cinco instâncias foram resolvidas em menos de dois minutos. Somente a instância *cl.02_060_02* precisou de mais tempo para a exploração, visto que foi necessária maior exploração da árvore do *branch-and-bound* (consulte a Seção 4.6 para mais detalhes).

A Decomposição de Benders, por sua vez, alcançou o ótimo em quatro das cinco instâncias. Em três delas, seu tempo de execução foi menor do que o do *Gurobi*. Além disso, para a única instância para qual o valor ótimo não foi encontrado, o número de placas na melhor solução foi igual ao do valor ótimo.

4.5 Resultados das instâncias grandes

Na Tabela 4, são apresentados os resultados obtidos para as instâncias da classe grande. Seu formato segue o mesmo padrão da Tabela 2 descrita na Seção 4.3.

Tabela 4: Resultados obtidos para as instâncias grandes pelo Modelo Padrão e pela Decomposição de Benders.

Instância	Padrão			Benders		
	Placas	Gap (%)	Tempo (s)	Placas	Gap (%)	Tempo (s)
cl_03_060_01	-	-	-	-	-	1800
cl_03_060_02	-	-	-	-	-	1800
cl_03_060_03	-	-	-	-	-	1800
cl_03_060_04	-	-	-	-	-	1800
cl_03_060_05	15*	21,42*	-	-	-	1800

O Modelo Padrão esgotou a memória em todos os casos. Contudo, para a instância cl_03_060_05, foi possível encontrar uma solução factível. No entanto, não foi possível extrair o valor das variáveis da solução, visto que a execução foi interrompida.

No caso da Decomposição de Benders, não foi possível encontrar solução factível para nenhuma das instâncias; entretanto, a memória não esgotou em nenhum caso. Isso indica que o uso da memória pela Decomposição de Benders foi menor e que há chances do método encontrar soluções factíveis (ou até mesmo ótimas) para as instâncias testadas, caso seja disponibilizado mais tempo.

4.6 Análise da decomposição de Benders

Na Tabela 5, são mostrados, da esquerda para a direita, o nome da instância, o número de nós explorados pelo Modelo Padrão, o número de nós explorados pelo problema mestre da Decomposição de Benders e a porcentagem de tempo gasta na execução de *callbacks* pela Decomposição de Benders. As classes das instâncias são representadas pelas letras P (pequena), M (média) e G (grande).

Tabela 5: Quantidade de nós explorados para cada instância pelo Modelo Padrão, quantidade de nós explorados pela Decomposição de Benders e porcentagem tempo gasto em *callbacks* pela Decomposição de Benders.

Instância	Nº de nós (Padrão)	Nº de nós (Benders)	CB% (Benders)
cl_01_060_01 (P)	1	98443	1,67
cl_01_060_02 (P)	1	96236	3
cl_01_060_03 (P)	1	106352	2,17
cl_01_060_04 (P)	1	115201	1,72
cl_01_060_05 (P)	1	125284	2,28
cl_02_060_01 (M)	1	1	100
cl_02_060_02 (M)	71	1	100
cl_02_060_03 (M)	1	1	100
cl_02_060_04 (M)	1	1	100
cl_02_060_05 (M)	1	1	100
cl_03_060_01 (G)	-	18548	100
cl_03_060_02 (G)	-	1	100
cl_03_060_03 (G)	-	5067	100
cl_03_060_04 (G)	-	7642	99,89
cl_03_060_05 (G)	-	10009	100

Como visto nas Seções 4.3 e 4.4, o Modelo Padrão obteve solução ótima para todas as instâncias pequenas e médias. Com exceção de uma instância média, todas as soluções ótimas foram encon-

tradas no primeiro nó, após o *presolve* e aplicação de heurísticas. Analisando a porcentagem de tempo gasto durante os *callbacks* pela Decomposição de Benders, é possível perceber que, para as instâncias pequenas, a verificação de factibilidade é rápida. Isso provavelmente se deve ao fato do subproblema ser um caso específico do Modelo Padrão, ou seja, as heurísticas aplicadas pelo *Gurobi* são tão efetivas para encontrar a solução do modelo do subproblema quanto para o Modelo Padrão. Esse comportamento se reflete tanto para as instâncias pequenas quanto médias.

Para as instâncias pequenas, em que o problema mestre da Decomposição de Benders explorou um grande número de nós, o tempo gasto em *callbacks* não superou 3%. O curto tempo nos subproblemas também se refletiu nas instâncias médias. Mesmo que o tempo de execução para esse conjunto de instâncias tenha sido gasto praticamente todo em *callbacks*, foi explorado apenas 1 nó no problema mestre. Somente a instância cl_02_060_04, que não comprovou a otimalidade, não conseguiu resolver com facilidade os subproblemas de sua última solução encontrada pelo problema mestre no primeiro nó. Provavelmente, trata-se de um caso similar ao que ocorreu durante a execução da instância cl_02_060_02 pelo Modelo Padrão, cujo tempo de execução (quase 1400 segundos) foi o maior dentre todas as instâncias que não esgotaram a memória.

No caso das instâncias grandes, o número de nós explorados pelo problema mestre foi menor em comparação às instâncias pequenas, e o tempo foi gasto praticamente todo em *callbacks*. Isso se deve à complexidade na resolução do subproblema. Além disso, a construção dos modelos dos subproblemas têm grande impacto no consumo de tempo, devido ao grande número de variáveis. É possível notar, entretanto, que a quantidade de nós explorados ainda é relativamente alta, provavelmente porque a maioria dos subproblemas foram resolvidos após a exploração de poucos nós com auxílio das heurísticas.

Considerando o conjunto de instâncias testadas, é possível perceber que a Decomposição de Benders proposta não consegue se aproveitar tão bem das heurísticas do *Gurobi*, que foi um fator diferencial na resolução do problema. Assim, o Modelo Padrão se mostrou um método mais eficaz para o conjunto de testes do experimento. Entretanto, a escalabilidade em relação à memória da decomposição aparenta ser melhor, como visto na Seção 4.5. Não é possível afirmar que o método conseguiria encontrar uma solução factível ou ótima, pois, para verificar essa hipótese, seriam necessários novos experimentos com maior tempo de duração para as instâncias da classe grande.

5 Conclusão e trabalhos futuros

Este trabalho resolveu o 2D-BPP utilizando uma decomposição lógica de Benders. Foi definido um modelo matemático (referido como Modelo Padrão), com base em [Arenales et al. \(2015\)](#), que discretiza as posições da placa e tamanho das peças. Em seguida, foi proposta uma decomposição de Benders na qual o problema mestre ficou responsável pela minimização do número de placas utilizadas para os cortes e a definição de qual peça será cortada em cada placa. Com isso, os subproblemas foram modelados para verificar se é possível cortar as peças alocadas em cada placa.

Experimentos utilizando o *solver Gurobi* foram realizados com 15 instâncias da literatura, sendo elas divididas em três classes: pequenas, médias e grandes. A decomposição proposta foi comparada ao Modelo Padrão. A partir dos experimentos foi possível observar que o Modelo Padrão obteve soluções ótimas para as instâncias pequenas e médias. A Decomposição de Benders, por sua vez, alcançou a solução ótima para a maioria das instâncias pequenas e todas as instâncias médias, contudo comprovou a otimalidade em somente quatro instâncias médias. Para as instâncias grandes, o Modelo Padrão estourou a capacidade da memória, enquanto a decomposição proposta foi incapaz de achar soluções factíveis. Com isso, observou-se a possibilidade de que a decomposição de Benders possa encontrar soluções factíveis para as instâncias grandes, caso haja mais tempo disponível.

Também foi possível perceber a efetividade do *presolve* e das heurísticas *Gurobi* no Modelo Padrão. A solução ótima foi encontrada no primeiro nó explorado para todas as instâncias pequenas

e 4 das 5 instâncias médias. Desse modo, é possível concluir que o *presolve* e as heurísticas foram os fatores diferenciais nos experimentos, e que a decomposição de Benders não conseguiu explorar tão bem esses aspectos devido à divisão entre problema mestre e subproblema.

Considerando as conclusões descritas, trabalhos futuros poderiam focar em dois principais aspectos: (i) realização de mais experimentos para assegurar as conclusões e hipóteses deste trabalho; e (ii) proposição de melhoria na decomposição de Benders. No caso (i), poderiam ser realizados experimentos com instâncias mais heterogêneas e em maior número para aumentar a confiabilidade dos experimentos deste trabalho. Além disso, experimentos com mais tempo de execução podem ser feitos para verificar se a decomposição proposta pode encontrar soluções factíveis quando o Modelo Padrão estoura a memória. Seguindo a ideia (ii), poderiam ser propostas melhorias na modelagem e na decomposição. Explorar a característica de placas homogêneas para a geração de cortes mais efetivos, revisar os modelos propostos, adicionar cortes da literatura e utilizar cortes analíticos são possibilidades para a melhoria do método. Além disso, como visto para o caso das instâncias pequenas, estudos em métodos para encontrar melhores limitantes duais para o problema mestre podem auxiliar na comprovação de otimalidade.

Referências

- M. Arenales, R. Morabito, V. Armentano, and H. Yanasse. *Pesquisa Operacional: Para cursos de engenharia*. Elsevier Brasil, 2015.
- Nestor M. Cid-Garcia and Yasmin A. Rios-Solis. Positions and covering: A two-stage methodology to obtain optimal solutions for the 2d-bin packing problem. *PLOS ONE*, 2020.
- Jean-François Côté, Mohamed Haouari, and Manuel Iori. Combinatorial benders decomposition for the two-dimensional bin packing problem, 2019.
- Harald Dyckhoff. A typology of cutting and packing problems. *European Journal of Operational Research*, 44:145–159, 1990.
- J. F. Oliveira F. Parreño, R. Alvarez-Valdes and J. M. Tamarit. A hybrid grasp/vnd algorithm for two- and three-dimensional bin packing. *Annals of Operations Research*, 2010.
- Michael R Garey and David S Johnson. *Computers and intractability*, volume 174. freeman San Francisco, 1979.
- Manuel Iori, Vinícius L. de Lima, Silvano Martello, Flávio K. Miyazawa, and Michele Monaci. Exact solution techniques for two-dimensional cutting and packing. *European Journal of Operational Research*, 289:399–415, 2021.
- Manuel Iori, Vinícius Loti de Lima, Silvano Martello, and Michele Monaci. 2dpacklib: a two-dimensional cutting and packing library. *Optimization Letters*, pages 1–10, 2022.
- Andrea Lodi, Silvano Martello, and Daniele Vigo. Heuristic and metaheuristic approaches for a class of two-dimensional bin packing problems. *INFORMS Journal on Computing*, 11:345–357, 1999.
- David Pisinger and Mikkel Sigurd. Using decomposition techniques and constraint programming for solving the two-dimensional bin-packing problem. *INFORMS Journal on Computing*, 2007.
- G. Wäscher, H. Haußner, and H. Schumann. An improved typology of cutting and packing problems. *European Journal of Operational Research*, 183:1109 – 1130, 2007.