

## **Um modelo para o problema de empacotamento de *bins* bidimensional guilhotinado 2-estágios com espalhamento de pedidos de clientes**

**Mateus Martin**

mpmartin@id.uff.br

Departamento de Engenharia de Produção, Universidade Federal Fluminense  
Rua Domingos Silvério 135, 25650-050, Petrópolis-RJ, Brazil

**Horacio Hideki Yanasse**

horacio.yanasse@unifesp.br

Universidade Federal de São Paulo - UNIFESP  
Av. Cesare Mansueto Giulio Lattes, 1201 - Eugênio de Melo, São José dos Campos - SP,  
12247-014

**Maristela O. Santos**

mari@icmc.usp.br

Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo  
Avenida Trabalhador São-carlense 400, 13566-590, São Carlos-SP, Brazil

**Reinaldo Morabito**

morabito@dep.ufscar.br

Departamento de Engenharia de Produção, Universidade Federal de São Carlos  
Via Washington Luiz km 235, 13565-905, São Carlos-SP, Brazil

### **RESUMO**

Nesse estudo, aborda-se uma variante do problema de empacotamento de *bins* bidimensional guilhotinado 2-estágios (2BPP) que lida com o espalhamento de pedidos de clientes (2BPP-OS). O 2BPP-OS considera um conjunto de itens retangulares, requeridos de diferentes pedidos de clientes, a ser cortado a partir de um conjunto de *bins* retangulares. O objetivo é minimizar o número de *bins* usadas e o espalhamento dos pedidos dos clientes ao longo do processo de corte. O 2BPP-OS tem relevância em sistemas de manufatura que buscam soluções de desperdício mínimo com níveis satisfatórios de serviço de cliente. A principal contribuição do trabalho é a proposição de um modelo de Programação Linear Inteira (PLI) ao 2BPP-OS. Usando um *solver* de PLI de propósito geral, os resultados mostram que o 2BPP-OS necessita de mais esforço computacional para resolver do que o 2BPP, visto que ele precisa considerar diversas simetrias que são frequentemente eliminadas em abordagens tradicionais para o 2BPP.

**PALAVRAS CHAVE.** Corte & Empacotamento, Programação Linear Inteira Mista, Minimização de espalhamento de pedidos.

**POI – PO na Indústria; PM – Programação Matemática**

### **ABSTRACT**

In this study, we address a variant of the two-dimensional bin packing problem of 2-stage patterns (2BPP) that considers the spread of customer orders (2BPP-OS). The 2BPP-OS addresses a set of rectangular items, required from different customer orders, to be cut from a set of rectangular bins. The objective is to minimize the number of bins used and the spread of customer orders

over the cutting process. The 2BPP-OS gains relevance in manufacturing environments that seek minimum waste solutions with satisfactory levels of customer service. The main contribution of this work is the proposition of an integer linear programming (ILP) model for the 2BPP-OS. Using a general-purpose ILP solver, the results show that the 2BPP-OS takes more computational effort to solve than the 2BPP, as it has to consider several symmetries that are often disregarded by the traditional 2BPP approaches.

**KEYWORDS.** Cutting & Packing. Mixed-integer linear programming. Order spread minimization

**ORI – OR in Industry; MP – Mathematical Programming**

### 1. Introdução

O problema de empacotamento de *bins* bidimensional (2BPP, do inglês, *Two-dimensional Bin Packing Problem*) é um problema combinatório que considera um conjunto de itens retangulares a ser cortado a partir de um conjunto de *bins* (objetos) retangulares e idênticas, enquanto se minimiza o número de *bins* usadas. O problema é NP-Difícil, dado que é uma extensão do problema de empacotamento de *bins* unidimensional [Garey e Johnson, 1979]. O 2BPP surge em operações de indústrias de manufatura, como no corte de painéis de vidro, chapas de madeira, chapas metálicas, e sistemas logísticos, como no carregamento de caminhões e projeto de embalagens [Scheithauer, 2018].

As abordagens de solução para o 2BPP são revisadas nos trabalhos de Lodi et al. [2002a], Scheithauer [2018], e Iori et al. [2021]. Essas abordagens podem ser categorizadas como: (i) algoritmos exatos que exploram todas as possíveis soluções, como algoritmos *branch and bound* e modelos de Programação Linear Inteira (PLI) [Lodi et al., 2004; Macedo et al., 2010; Silva et al., 2010; Puchinger e Raidl, 2007]; (ii) algoritmos aproximativos com garantias de desempenho de pior caso baseados em estratégias de alocação do tipo prateleira, como *first fit decreasing height* e *best fit decreasing height* [Coffman et al., 1980; Lodi et al., 2002c]; (iii) algoritmos heurísticos e meta-heurísticos para encontrar soluções de boa qualidade ao problema em tempo de processamento razoável [Lodi et al., 1999; Alvelos et al., 2009; Cui et al., 2015].

Dyson e Gregory [1974] e Madsen [1979, 1988] estiveram entre os primeiros autores a abordar problemas de corte que lidam com uma decisão de sequenciamento, buscando contemplar os pedidos de clientes em uma abordagem de duas fases [Martin et al., 2023]. Eles propuseram resolver primeiramente o problema de corte com a abordagem de geração de colunas de Gilmore e Gomory [1965] e, então, sequenciar os padrões de corte a fim de reduzir o número de descontinuidades (i.e., o número de vezes que um pedido de cliente é *re-iniciado* ao longo do processo de corte), a partir de abordagens para o problema do caixeiro viajante. Destaca-se que esses autores argumentaram que, na indústria de vidros, a tonalidade dos vidros pode ser ligeiramente diferente, o que justifica, sob aspectos estéticos, que os itens de um mesmo pedido devam ser cortados a partir do menor número de *bins*.

Nesse contexto, é apropriado otimizar o espalhamento dos pedidos dos clientes durante o processo de corte, dado que os itens pertencentes a um mesmo pedido são despachados juntos para o próximo estágio de produção ou distribuição. Note que as empresas geralmente seguem uma política organizacional de aguardar a consolidação da demanda de alguns pedidos de clientes antes de iniciar as operações de corte a fim de aumentar a taxa de utilização das *bins*. Essa política tende a prejudicar os tempos de entrega e faturamento de pedidos se a sequência de *bins* cortadas geram itens sem considerar que eles pertencem a pedidos de clientes diferentes. Essa decisão pode

ser abordada pelo Problema de Minimização do Espalhamento de Pedidos (MORP, do inglês, *Minimization of Order Spread Problem*), que é um problema combinatório que busca determinar a sequência de processamento de tarefas [Dyson e Gregory, 1974]. Note que uma solução do 2BPP que é ótima em termos da utilização de *bins* pode não ser satisfatória em termos de serviço ao cliente se os pedidos permanecem em processamento por longos períodos durante as operações de corte. O MORP também é um problema NP-Difícil [Linhares e Yanasse, 2002].

Nesse trabalho, propõe-se um modelo de PLI para o problema de empacotamento de *bins* bidimensional guilhotinado 2-estágios com espalhamento de pedidos de clientes, denotado como 2BPP-OS – as características do padrão de corte serão explicadas na próxima seção. O 2BPP-OS é um problema que integra as decisões do 2BPP e MORP. Destaca-se a importância do 2BPP-OS no contexto de sistemas produtivos de baixa escala de produção, dado que eles frequentemente competem não apenas em custos mas também na responsividade de entrega de produtos. A formulação de PLI proposta é derivada da formulação de Lodi et al. [2004] em relação ao 2BPP e dos estudos de Martin e Yanasse [2021] em relação ao MORP. Apesar do seu potencial prático, ao conhecimento desses autores, não existem abordagens integradas para o 2BPP-OS na literatura. Dessa forma, para fins de comparação, as soluções do modelo proposto são comparadas às soluções do modelo de Lodi et al. [2004], que não considera as decisões de espalhamento de pedidos de clientes. Usando um *solver* de PLI de propósito geral, os resultados mostram que o 2BPP-OS necessita de mais esforço computacional para resolver do que o 2BPP, visto que ele precisa considerar diversas simetrias que são frequentemente eliminadas em abordagens tradicionais para o 2BPP.

A estrutura do trabalho é como segue. Na Seção 2, o 2BPP-OS é descrito e um exemplo ilustrativo é apresentado para destacar a noção de espalhamento do pedido de cliente. Na Seção 3, o modelo de Lodi et al. [2002b] para o 2BPP é apresentado. Em seguida, esse modelo é adaptado para considerar as decisões de sequenciamento do 2BPP-OS. A partir da adaptação de exemplares de *benchmark* para o 2BPP disponíveis na literatura, o desempenho computacional da formulação proposta é avaliado e reportado na Seção 4. Na Seção 5, conclui-se o estudo e apresentam-se oportunidades de pesquisa futura.

## 2. Descrição do problema

O 2BPP-OS considera um conjunto  $I = \{1, \dots, n\}$  de itens retangulares a ser cortado a partir de um conjunto de *bins* retangulares de comprimento  $L$  e largura  $W$ . Cada item  $i \in I$  tem uma largura  $l_i$ , comprimento  $w_i$ , e está associado a um único pedido de cliente  $c_i \in C = \{1, \dots, m\}$ . Considera-se que as dimensões  $L$  e  $l_i$  estão ao longo do eixo-x e as dimensões  $W$  e  $w_i$  ao longo do eixo-y – a rotação de 90° dos itens é proibida. Em relação à decisão de corte, o 2BPP-OS lida com padrões de corte guilhotinados 2-estágios. Num padrão de corte guilhotinado, todos os itens são obtidos a partir de uma sequência de cortes de borda-a-borda [Scheithauer, 2018]. No caso particular dos padrões guilhotinados 2-estágios, tem-se uma sequência de cortes na mesma direção (resp., horizontal ou vertical), isto é, os cortes de primeiro estágio geram tiras (resp., horizontais ou verticais) a partir das *bins* e, então, cortes de segundo estágio (resp., vertical ou horizontal) geram itens a partir dessas tiras, conforme ilustrado nas Figs. 1 e 2 [Lodi et al., 2004]. Nessas figuras o cortes de primeiro estágio são horizontais, e os cortes de segundo estágio são verticais.

Em relação à decisão de sequenciamento, todos os itens de um pedido de cliente  $c \in C$  são representados pelo conjunto  $I_c = \{i \in I \mid c_i = c\}$ . Seja  $\underline{s}$  um limitante inferior ao número de *bins* usadas (p.e., obtido por meio de uma relaxação ao problema). Seja  $\bar{s}$  um limitante superior ao número de *bins* usadas (p.e., obtido por de uma heurística). Considera-se que o conjunto  $S = \{1, \dots, \bar{s}\}$  de *bins* é ordinal e, portanto, seu  $k$ -ésimo elemento representa a  $k$ -ésima *bin* a ser processada durante as operações de corte. Assume-se que o parâmetro binário  $\pi_i^s$  seja igual a 1

Tabela 1: Exemplar ilustrativo do 2BPP-OS com  $L = W = 6$  e  $n = 8$  itens associados a  $m = 4$  pedidos de clientes.

$i$	1	2	3	4	5	6	7	8
$l_i$	3	2	1	1	2	2	4	3
$w_i$	3	4	3	3	4	3	2	3
$c_i$	1	1	2	2	3	3	4	4

se o item  $i \in I$  é cortado da  $bin$   $s \in S$ , e 0 caso contrário. Dessa forma, o espalhamento do pedido do cliente  $c \in C$  é dado por  $OS^c = \max_{s \in S, i \in I_c} \{s\pi_i^s : \pi_i^s = 1\} - \min_{s \in S, i \in I_c} \{s\pi_i^s : \pi_i^s = 1\} + 1$  [Martin e Yanasse, 2021]. Logo, o espalhamento do pedido do cliente é computado em termos de  $bins$  processadas. Assume-se que o processamento de cada  $bin$  dura um intervalo de tempo constante, independentemente dos itens a serem cortados a partir dela. Assim, além de se minimizar o número de  $bins$  usadas, o 2BPP-OS considera a minimização do maior espalhamento ( $\max_{c \in C} \{OS^c\}$ ) e a soma dos espalhamentos ( $\sum_{c \in C} OS^c$ ) dos pedidos dos clientes. Destaca-se que a minimização do maior espalhamento busca indiretamente um nível de serviço mínimo de pior caso ao processo de corte, enquanto que a minimização da soma dos espalhamento um nível de serviço de caso médio.

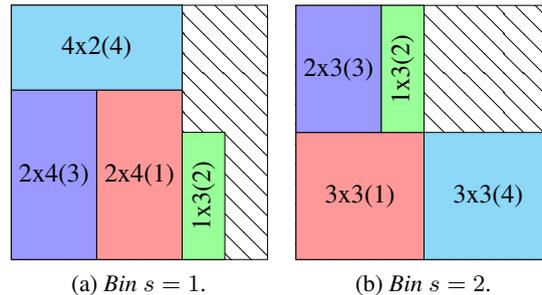


Figura 1: Uma solução ao exemplar com 2  $bins$ , maior espalhamento de 2 unidades, e soma dos espalhamentos de 8 unidades.

Na Tabela 1, apresenta-se um exemplar do 2BPP-OS com  $bins$  de tamanho  $L = W = 6$  e  $n = 8$  itens associados a  $m = 4$  pedidos de clientes (i.e.,  $C = \{1, 2, 3, 4\}$ ). Para esse exemplo, o limitante de material é igual a  $\lceil \sum_{i \in I} l_i w_i / (LW) \rceil = \lceil 1.50 \rceil$ , logo o número de 2  $bins$  é um limitante inferior para o número ótimo de  $bins$  a serem cortadas [Scheithauer, 2018]. Na Fig. 1, apresenta-se uma solução de 2  $bins$ , maior espalhamento de 2 unidades, e soma dos espalhamentos de 8 unidades – entre parênteses reporta-se o número do pedido do cliente; as áreas hachuradas são desperdício. Note que o processamento de cada um dos quatro pedidos se inicia na primeira  $bin$  e se encerra na segunda  $bin$ ; por isso, o espalhamento de cada um desses pedidos é igual a 2 ( $= 2 - 1 + 1$ ) unidades. Por outro lado, na Fig. 2, apresenta-se uma outra solução ao exemplar ilustrativo com 2  $bins$ , máximo espalhamento de 2 unidades, e soma dos espalhamentos de 5 unidades. Por exemplo, para essa solução, o espalhamento do pedido:  $c = 1$  é de 2 ( $= 2 - 1 + 1$ ) unidades;  $c = 2$  e  $c = 4$  é de 1 ( $= 1 - 1 + 1$ ) unidade; e,  $c = 3$  é de 1 ( $= 2 - 2 + 1$ ) unidade.

### 3. Modelos matemáticos

Nesta seção, revisa-se a formulação de PLI proposta em Lodi et al. [2004] para o 2BPP na Seção 3.1. Em seguida, na Seção 3.2, a partir dessa formulação da literatura, propõe-se uma

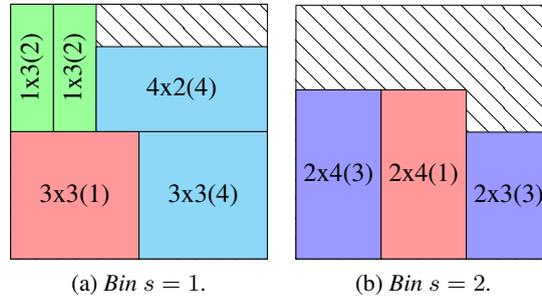


Figura 2: Uma solução ao exemplar com 2 bins, maior espalhamento de 2 unidades, e soma dos espalhamentos de 5 unidades.

formulação de PLI para o 2BPP-OS. Sem perda de generalidade, assume-se que o exemplar é constituído de inteiros positivos cujos dados dos itens (i.e.,  $l_i, w_i, c_i$ ) estão ordenados de forma não-crescente tal que  $w_1 \geq w_2 \geq \dots \geq w_n$ . Assume-se, também, que os cortes de primeiro estágio são horizontais e os cortes de segundo estágio são verticais. Para facilitar a comparação, usam-se notações similares para as variáveis de decisão dos dois modelos; no entanto, seus significados estão estritos a cada uma das subseções a seguir.

### 3.1. Modelo para o 2BPP

O modelo proposto em Lodi et al. [2004] para o 2BPP pode ser formulado por meio de duas famílias de variáveis binárias, definidas a seguir:

- $\alpha_{ji}$  variável binária cujo valor é (1), se o item  $i \in I$  é empacotado na tira  $j \in I, j \leq i$ , e (0) caso contrário;
- $\beta_j^s$  variável binária cujo valor é (1), se a tira  $j \in I$  é empacotada na bin  $s \in I, s \leq j$ , e (0) caso contrário.

Note que o número de itens  $n$  representa um limitante superior para o máximo número de tiras e o máximo número de bins em qualquer solução; por isso, todos os índices das variáveis  $\alpha_{ji}$  e  $\beta_j^s$  estão no conjunto  $I$ . Essa formulação entende que cada tira  $j \in I$  será *inicializada* pelo seu item de maior largura (i.e.,  $\alpha_{jj} = 1$ ); por isso, no domínio das variáveis  $\alpha_{ji}$ , tem-se que  $j, i \in I, j \leq i$  – note que os dados dos itens foram ordenados conforme suas larguras. Como estratégia para eliminação de simetrias, sem perda de generalidade, a formulação entende que o primeiro item será empacotado na primeira bin, o segundo item será empacotado na primeira ou segunda bin, o terceiro item na primeira, segunda ou terceira bin, e assim por diante. Nesse contexto, cada bin  $s \in I$  será *inicializada* pela tira  $s \in I$  (i.e.,  $\beta_s^s = 1$ ) que, por sua vez, foi inicializada pelo item  $s \in I$ . A formulação é dada pelo Modelo (1).

$$\text{Minimize } \sum_{s \in I} \beta_s^s, \tag{1a}$$

$$\text{s.a. } \sum_{j=1}^i \alpha_{ji} = 1, \tag{1b} \quad i \in I,$$

$$\sum_{i=j+1}^n l_i \alpha_{ji} \leq (L - l_j) \alpha_{jj}, \tag{1c} \quad j \in I,$$

$$\sum_{s=1}^j \beta_j^s = \alpha_{jj}, \quad j \in I, \quad (1d)$$

$$\sum_{j=s+1}^n w_j \beta_j^s \leq (W - w_s) \beta_s^s, \quad s \in S, \quad (1e)$$

$$\alpha_{ji} \in \{0, 1\}, \quad j, i \in I, j \leq i, \quad (1f)$$

$$\beta_j^s \in \{0, 1\}, \quad s, j \in I, s \leq j. \quad (1g)$$

A função objetivo (1) minimiza o número de *bins* usadas. As restrições (1b) garantem que cada item será empacotado em uma única tira. As restrições (1c) garantem que a soma do comprimento dos itens  $i$  empacotados na tira  $j$  não excede o comprimento  $L$  de cada *bin*. Além disso, se  $\alpha_{jj} = 0$ ,  $j \in I$ , tem-se que  $\alpha_{ji} = 0$ ,  $i \in I$ ,  $j \leq i$ , isto é, nenhum item é empacotado em tira não usada. As restrições (1d) garantem que uma tira usada (i.e.,  $\alpha_{jj} = 1$ ,  $j \in I$ ) deve ser empacotada em alguma *bin*  $s \in I$ ,  $s \leq j$ . As restrições (1e) garantem que a soma da largura das tiras empacotadas em certa *bin* não exceda a largura  $W$  da *bin*. Além disso, se  $\beta_s^s = 0$ ,  $s \in I$ , tem-se que  $\beta_j^s = 0$ ,  $j \in I$ ,  $s \leq j$ , isto é, nenhuma tira é empacotada em *bin* não usada. As restrições (1f) e (1g) definem o domínio das variáveis de decisão.

### 3.2. Modelo para o 2BPP-OS

O modelo proposto para o 2BPP-OS tem sete famílias de variáveis de decisão, das quais quatro são variáveis binárias referentes às decisões do 2BPP e as restantes são variáveis contínuas referentes ao MORP. Destaca-se que a estratégia de eliminação de simetria da seção anterior poderia conduzir a perda de otimalidade no contexto do 2BPP-OS, dado que ela representa uma restrição de sequenciamento indireta. Portanto, já não se pode assumir, sem perda de generalidade, que o primeiro item será empacotado na primeira *bin*, o segundo item na primeira ou segunda *bins* e assim por diante. A fim de manter a otimalidade ao 2BPP-OS, o conjunto  $S$  das *bins* é redefinido para  $S = \{1, \dots, \bar{s}\}$ . Lembre que  $\underline{s}$  e  $\bar{s}$  são limitantes inferiores e superiores ao número de *bins* usadas, respectivamente. As variáveis de decisão do modelo proposto são definidas a seguir:

- $y^s$  variável binária cujo valor é (1), se a *bin*  $s \in S$  é usada, e (0) caso contrário;
- $\alpha_{ji}$  variável binária cujo valor é (1), se o item  $i \in I$  é empacotado na tira  $j \in I$ ,  $j \leq i$ , e (0) caso contrário;
- $\beta_j^s$  variável binária cujo valor é (1), se a tira  $j \in I$  é empacotada na *bin*  $s \in S$  e (0) caso contrário.
- $\lambda_{ji}^s$  variável binária cujo valor é (1), se e somente se,  $\beta_j^s = 1$  e  $\alpha_{ji} = 1$ ,  $s \in S$ ,  $j, i \in I$ ,  $j \leq i$ , e (0) caso contrário;
- $b^c$  início do processamento do pedido do cliente  $c \in C$  em termos de *bins*.
- $e^c$  fim do processamento do pedido do cliente  $c \in C$  em termos de *bins*.
- $s^c$  comprimento do intervalo de processamento do pedido do cliente  $c \in C$  em termos de *bins*.

O modelo proposto ao 2BPP-OS é dado pelo Modelo (2).

$$\text{Minimize} \quad r_1 \sum_{s \in S} y^s + r_2 \max_{c \in C} \{s^c\} + r_3 \sum_{c \in C} s^c, \quad (2a)$$

$$\text{s.t.} \quad \sum_{j=1}^i \alpha_{ji} = 1, \quad i \in I, \quad (2b)$$

$$\sum_{s \in S} \beta_j^s = \alpha_{jj}, \quad j \in I, \quad (2c)$$

$$\sum_{i=j+1}^n l_i \alpha_{ji} \leq (L - l_j) \alpha_{jj}, \quad j \in I, \quad (2d)$$

$$\sum_{j \in I} w_j \beta_j^s \leq W y^s, \quad s \in S, \quad (2e)$$

$$\beta_j^s + \alpha_{ji} \leq 1 + \lambda_{ji}^s, \quad s \in S, j, i \in I, j \leq i, \quad (2f)$$

$$\lambda_{ji}^s \leq (\beta_j^s + \alpha_{ji})/2, \quad s \in S, j, i \in I, j \leq i, \quad (2g)$$

$$s^c = e^c - b^c + 1, \quad c \in C, \quad (2h)$$

$$b^c \leq \sum_{s \in S} \sum_{j=1}^i s \lambda_{ji}^s, \quad c \in C, i \in I_c, \quad (2i)$$

$$e^c \geq \sum_{s \in S} \sum_{j=1}^i s \lambda_{ji}^s, \quad c \in C, i \in I_c, \quad (2j)$$

$$y^s \in \{0, 1\}, \quad s \in S, \quad (2k)$$

$$\alpha_{ji} \in \{0, 1\}, \quad j, i \in I, j \leq i, \quad (2l)$$

$$\beta_j^s \in \{0, 1\}, \quad s \in S, j \in I, \quad (2m)$$

$$\lambda_{ji}^s \in \{0, 1\}, \quad s \in S, j, i \in I, j \leq i. \quad (2n)$$

$$1 \leq b^c, e^c, s^c \leq \bar{s}, \quad c \in C. \quad (2o)$$

A função objetivo (2b) minimiza o número de *bins* usadas, o maior espalhamento dentre os pedidos, e a soma do espalhamento dos pedidos de acordo com os parâmetros  $r_1$ ,  $r_2$ , and  $r_3$  que funcionam como pesos a esses três objetivos. Nos experimentos computacionais da Seção 4 foram usados  $r_1 = m\bar{s}^2$ ,  $r_2 = m\bar{s}$ , e  $r_3 = 1$  a fim de estabelecer um objetivo primário ao número de *bins* usadas, um objetivo secundário ao maior espalhamento, e um objetivo terciário para a soma dos espalhamentos. Note que o termo  $\max_{c \in C} \{s^c\}$  pode ser escrito numa forma linear se substituído por uma variável auxiliar  $K$  e incluídas as restrições  $K \geq s^c$ ,  $c \in C$ .

As restrições (2b) garantem que cada item  $i \in I$  é empacotado em uma única tira  $j \in I$ ,  $j \leq i$ . As restrições (2c) asseguram que cada tira  $j \in I$ , se usada, é empacotada em uma única *bin*  $s \in S$ . As restrições (2d) asseguram que a soma dos comprimentos dos itens empacotados na tira  $j \in I$  não exceda o comprimento  $L$  da *bin*. As restrições (2e) garantem que a soma das larguras das tiras empacotadas na *bin*  $s \in S$  não exceda o comprimento  $W$  da *bin*. As duas restrições anteriores também forçam que:  $\alpha_{ji} = 0$  se  $\alpha_{jj} = 0$ ,  $j, i \in I, j \leq i$ ; e,  $\beta_j^s = 0$  se  $y^s = 0$ ,  $s \in S, j \in I$ . As restrições (2f) e (2g) são responsáveis por gerar o resultado  $\lambda_{ji}^s = \beta_j^s \alpha_{ji}$ ,  $s \in S, j, i \in I, j \leq i$ . A partir das variáveis de corte  $\lambda_{ji}^s$ , as restrições de acoplamento (2i) e (2j) geram a definição das variáveis  $b^c$  e  $e^c$  ao considerar que todos os itens  $i \in I_c$  do pedido do cliente  $c \in C$ . As restrições (2k) até (2o) definem o domínio das variáveis.

As expressões (3a) até (3d), a seguir, buscam eliminar algumas simetrias do modelo proposto. A restrição (3a) garante que as primeiras  $\underline{s}$  *bins* serão usadas. As desigualdades válidas (3b) garantem que a *bin*  $s + 1$  é usada apenas se a *bin*  $s$  (i.e., anterior) também é usada. As restrições (3c) e (3d) eliminam o caso quando dois itens  $i, j \in I, j < i$  não cabem em uma tira horizontal, dado que  $l_i + l_j > L$ . Assim, algumas dessas variáveis nem precisam ser geradas para o modelo.

$$y^s = 1, \quad s = 1, \dots, \underline{s}, \quad (3a)$$

$$y^s \geq y^{s+1},$$

$$\alpha_{ji} = 0,$$

$$\lambda_{ji}^s = 0,$$

$$s = \underline{s}, \dots, \bar{s} - 1, \quad (3b)$$

$$j, i \in I, j < i, l_i + l_j > L, \quad (3c)$$

$$s \in S, j, i \in I, j < i, l_i + l_j > L. \quad (3d)$$

#### 4. Experimentos computacionais

Nesta seção, reportam-se os experimentos realizados para avaliar o desempenho computacional do Modelo (2). Como se desconhece de outros modelos de PLI ou mesmo abordagens algorítmicas para o 2BPP-OS, toma-se o modelo de Lodi et al. [2004], descrito na Seção 3.1, como abordagem de *benchmark*. A seguir, refere-se ao Modelo (1) como Modelo-A, e ao Modelo (2) com as expressões (3) como Modelo-B. Como essas abordagens não são diretamente comparáveis, a seguir, o foco estará em destacar o potencial prático do 2BPP-OS pela comparação das métricas de espalhamento de pedidos de clientes de uma abordagem que negligencia (i.e., Modelo-A) e de uma que considera (i.e., Modelo-B) as decisões de sequenciamento. Os modelos foram implementados em C++ e usam o GUROBI v.10.0.0 como *solver* de propósito geral. Todos os experimentos foram realizados em um PC com processador Intel Xeon E5-2680v2 (2.8 GHz), usando 1 thread, 16 GB RAM, sob sistema operacional CentOS Linux 7.2.1511. Cada execução do *solver* foi limitada a 3600 segundos. A seguir, usam-se as letras “t” nas tabelas para indicar quando esse tempo limite foi atingido. Referente ao Modelo-B: o limitante inferior  $\underline{s}$  foi obtido ao se resolver uma relaxação do problema, dada pelo BPP unidimensional em que o peso  $p_i$  de cada item é do tipo  $p_i = l_i w_i$ , e a capacidade de cada *bin* é  $LW$ ; e, o limitante superior  $\bar{s}$  foi obtido ao se resolver uma heurística de prateleira do tipo *first fit decreasing height*. Os tempos de processamento desses cálculos foram incorporados no tempo total reportado para o Modelo-B.

Nos experimentos foram considerados dois conjuntos de exemplares, que foram adaptados da literatura de 2BPP pela inclusão de informações relativas aos pedidos dos clientes. O conjunto de exemplares #A tem um total de 18 instâncias. Esses exemplares foram criados a partir dos doze exemplares gcut1-12 propostos em Beasley [1985]. A principal característica das instâncias gcut é que o tamanho dos itens é relativamente grande em relação ao tamanho das *bins* – para informações mais detalhadas, ver Beasley [1985]. A fim de gerar as informações dos pedidos de clientes, agregou-se arbitrariamente os itens em pedidos de clientes. Para cada instância, estabeleceu-se um número de pedidos de clientes e um número mínimo de itens por pedido. Assim, nas instâncias adaptadas com:  $n = 10, 20$  itens existem  $m = 3$  pedidos de clientes (mínimo de 2 itens por pedido); e, com  $n = 30$  itens existem  $m = 3, 5$  pedidos de clientes (mínimo de 3 itens por pedido); e, com  $n = 50$  itens existem  $m = 5, 7$  pedidos de clientes (mínimo de 4 itens por pedido). Refere-se a cada exemplar como “nome-#n-#m”; por exemplo, o exemplar gcut12-50-07 foi adaptado do exemplar gcut12, e tem  $n = 50$  itens e  $m = 7$  pedidos de clientes.

Na Tabela 2 reportam-se os resultados dos exemplares do conjunto #A. No cabeçalho da tabela, tem-se o nome do exemplar, o número de pedidos de clientes  $m$  e o número de itens  $n$  para cada exemplar. Em seguida, para cada um dos modelos, tem-se o número de *bins* usadas ( $\sum y^s$ ), o maior espalhamento ( $\max\{OS^c\}$ ), a soma dos espalhamentos ( $\sum OS^c$ ), o *gap* de otimalidade em porcentagem ( $\text{gap}[\%]$ ), e o tempo de processamento em segundos ( $\text{tempo}[s]$ ). Os resultados reportados na Tabela 2 mostram que o *gap* médio obtido pelo *solver* com o Modelo-A foi de 0,00% com tempo médio de 0,53 segundos, enquanto que com o Modelo-B foi de 0,39% com tempo médio de 2.084,41 segundos. Nota-se que em ambos os modelos o valor de 9,94 foi alcançado como valor médio de número de *bins* usadas. No entanto, as duas métricas de espalhamento de pedidos são muito diferentes quando se considera e quando não se considera o espalhamento dos pedidos. Por exemplo, na instância gcut12-50-07, para o mesmo número de 16 *bins* usadas, o *solver* com o

Tabela 2: Resultados para o conjunto de exemplares #A.

Exemplar	m	n	Modelo-A					Modelo-B				
			$\sum y^s$	$\max\{OS^c\}$	$\sum OS^c$	gap[%]	tempo[s]	$\sum y^s$	$\max\{OS^c\}$	$\sum OS^c$	gap[%]	tempo[s]
gcut01-10-03	3	10	5	3	8	0.00	<0.1	5	2	6	0.00	0.05
gcut02-20-03	3	20	6	6	15	0.00	0.09	6	5	13	0.00	7.89
gcut03-30-03	3	30	8	8	21	0.00	0.32	8	6	14	0.00	1287.18
gcut03-30-05	5	30	8	8	29	0.00	0.33	8	5	16	1.55	tl
gcut04-50-05	5	50	14	14	60	0.00	0.26	14	10	27	1.02	tl
gcut04-50-07	7	50	14	14	84	0.00	0.27	14	7	30	1.04	tl
gcut05-10-03	3	10	4	3	7	0.00	<0.1	4	2	5	0.00	0.04
gcut06-20-03	3	20	7	7	18	0.00	0.02	7	4	10	0.00	46.69
gcut07-30-03	3	30	12	11	31	0.00	0.11	12	7	14	0.00	94.49
gcut07-30-05	5	30	12	11	41	0.00	0.10	12	6	18	0.69	tl
gcut08-50-05	5	50	14	14	55	0.00	0.76	14	8	24	0.53	tl
gcut08-50-07	7	50	14	14	76	0.00	0.46	14	5	26	0.03	tl
gcut09-10-03	3	10	3	3	6	0.00	<0.1	3	2	5	0.00	0.26
gcut10-20-03	3	20	8	8	22	0.00	0.02	8	4	11	0.00	82.76
gcut11-30-03	3	30	9	9	25	0.00	0.06	9	6	14	0.13	tl
gcut11-30-05	5	30	9	9	38	0.00	0.06	9	5	15	1.24	tl
gcut12-50-05	5	50	16	16	66	0.00	3.36	16	8	25	0.01	tl
gcut12-50-07	7	50	16	16	87	0.00	3.33	16	6	37	0.81	tl
<b>Média</b>	<b>4.33</b>	<b>31.67</b>	<b>9.94</b>	<b>9.67</b>	<b>38.28</b>	<b>0.00</b>	<b>0.53</b>	<b>9.94</b>	<b>5.44</b>	<b>17.22</b>	<b>0.39</b>	<b>2084.41</b>

Modelo-A encontrou uma solução que levou ao máximo espalhamento de 16 unidades e soma de espalhamento de 87 unidades, enquanto que com o Modelo-B o máximo espalhamento foi de 6 unidades e a soma do espalhamento de 37 unidades. Note que o certificado de otimalidade foi obtido em apenas 8 dos 18 exemplares para o Modelo-B.

O conjunto de exemplares #B tem um total de 35 instâncias, que foram baseadas nos exemplares propostos em Berkey e Wang [1987] e Lodi et al. [1999]. Essas instâncias foram geradas aleatoriamente por esses autores e possuem características distintas, como itens com formas diferentes e itens com tamanhos pequenos em relação ao tamanho das *bins* – ver Lodi et al. [1999] para uma descrição detalhada. Mais uma vez, agregou-se arbitrariamente os itens para gerar pedidos de clientes. Assim, as instâncias adaptadas: com  $n = 20$  itens possuem  $m = 3$  pedidos de clientes (mínimo de 2 itens por pedido); com  $n = 40$  itens tem  $n = 3, 5$  pedidos de clientes (mínimo de 3 itens por pedido); e, com  $n = 60$  itens tem  $n = 5, 7$  pedidos de clientes (mínimo de 4 itens por pedido). As entradas da Tabela 3 são uma média de 7 exemplares, isto é, elas estão agregadas por número de itens  $n$  e número de pedidos de cliente  $m$ . Adicionalmente, reporta-se na coluna OPT o número de certificados de otimalidade obtidos pelo *solver* diante cada modelo.

Tabela 3: Resultados com o conjunto de exemplares #B.

n	m	Modelo-A						Modelo-B					
		$\sum y^s$	$\max\{OS^c\}$	$\sum OS^c$	gap[%]	tempo[s]	OPT	$\sum y^s$	$\max\{OS^c\}$	$\sum OS^c$	gap[%]	tempo[s]	OPT
20	3	8.43	8.14	20.57	0.00	0.02	7	8.43	4.71	11.14	0.00	53.69	7
40	3	11.71	11.43	32.71	0.00	5.17	7	11.71	5.57	14.57	1.89	1647.64	4
	5	11.71	11.29	47.86	0.00	5.19	7	11.71	5.00	16.57	1.84	tl	0
60	5	21.86	21.43	97.43	0.00	18.16	7	21.86	10.43	38.43	1.67	tl	0
	7	21.86	21.43	123.71	0.00	18.40	7	21.86	8.43	44.43	1.67	tl	0

Os resultados reportados na Tabela 3 mostram que o *solver* foi capaz de encontrar uma solução ótima e provar sua otimalidade em 35 exemplares (de 35) com o Modelo-A e 11 exemplares com o Modelo-B. Apesar disso, o gap médio de otimalidade de todas as classes é inferior a 2% para

o Modelo-B. Por outro lado, novamente, nota-se um ganho significativo nas métricas de pedidos de cliente ao se considerar as decisões de sequenciamento. Por exemplo, para as instâncias da classe  $n = 20$  itens e  $m = 3$  pedidos de clientes, houve uma melhora de mais de 40%, isto é, maior espalhamento de 8,14 para 4,71 unidades, e soma dos espalhamentos de 20,57 para 11,14 unidades.

Por fim, destaca-se que a fim de minimizar as métricas de espalhamento de pedidos, as soluções do Modelo-B, em geral, abrem e fecham rapidamente os pedidos de cliente, além de incluir poucos tipos de pedidos por *bin*. Na prática, nesse quesito de alocação de itens às *bins*, as soluções do Modelo-B são muito diferentes das soluções do Modelo-A. Note que não é trivial determinar uma ordem na qual os pedidos deverão ser abertos e fechados, e nem quais pedidos deverão ser obtidos de quais *bins*. Além disso, nota-se que a obtenção de limitantes duais para o Modelo-B se mostrou muito mais difícil do que para o Modelo-A, o que explica o número de exemplares com gap positivo. Esse resultado para em consonância ao reportado em Martin e Yanasse [2021] destacaram a mesma dificuldade no contexto de formulações de PLI que em relação ao MORP.

## 5. Conclusões

Abordou-se o problema de empacotamento de *bins* bidimensional guilhotinado 2-estágio com espalhamento de pedido de clientes. Esse problema surge em indústrias de manufatura que buscam por soluções de desperdício mínimo que são responsivas em termos de processar os pedidos de clientes rapidamente. Foi proposta uma formulação de PLI ao problema. Os resultados dos experimentos computacionais mostraram que é possível obter soluções satisfatórias em termos das métricas de espalhamento dos pedidos dos clientes, e que são ótimas em termos da utilização das *bins*. As soluções obtidas podem parecer similares em termos da utilização das *bins*, mas elas são completamente diferentes de soluções advindas de uma abordagem que não considera o espalhamento dos pedidos de clientes.

Para pesquisa futura, pode-se buscar estender os modelos pseudo-polinomiais de Silva et al. [2010] para o 2BPP a fim de lidar com o espalhamento dos pedidos de clientes. Note que essa extensão não parece trivial, dado que esses modelos não consideram a alocação de itens a *bins*, o que é necessário para o cômputo do espalhamento de cada pedido. Alternativamente, poderia-se investigar outros requerimentos práticos de operações de corte ou operações de sequenciamento como, por exemplo, pilhas abertas, datas de entrega ou perfis de corte como padrões não-guilhotinados, guilhotinado 3-estágios ou guilhotinado p-grupo.

## Referências

- Alvelos, F., Chan, T., Vilaça, P., Gomes, T., Silva, E., e Valério de Carvalho, J. (2009). Sequence based heuristics for two-dimensional bin packing problems. *Engineering Optimization*, 41(8): 773–791.
- Beasley, J. E. (1985). Algorithms for Unconstrained Two-Dimensional Guillotine Cutting. *The Journal of the Operational Research Society*, 36(4):297. ISSN 01605682.
- Berkey, J. O. e Wang, P. Y. (1987). Two-dimensional finite bin-packing algorithms. *Journal of the Operational Research Society*, 38(5):423–429.
- Coffman, E. G., Jr., Garey, M. R., Johnson, D. S., e Tarjan, R. E. (1980). Performance bounds for level-oriented two-dimensional packing algorithms. *SIAM Journal on Computing*, 9(4):808–826.
- Cui, Y.-P., Cui, Y., e Tang, T. (2015). Sequential heuristic for the two-dimensional bin-packing problem. *European Journal of Operational Research*, 240(1):43–53. ISSN 0377-2217.

- Dyson, R. G. e Gregory, A. S. (1974). The Cutting Stock Problem in the Flat Glass Industry. *Journal of the Operational Research Society*, 25(1):41–53. ISSN 0160-5682.
- Garey, M. R. e Johnson, D. S. (1979). *Computers and Intractability; A Guide to the Theory of NP-Completeness*. W. H. Freeman & Co., USA. ISBN 0716710455.
- Gilmore, P. C. e Gomory, R. E. (1965). Multistage Cutting Stock Problems of Two and More Dimensions. *Operations Research*, 13(1):94–120. ISSN 0030-364X.
- Iori, M., De Lima, V. L., Martello, S., Miyazawa, F. K., e Monaci, M. (2021). Exact solution techniques for two-dimensional cutting and packing. *European Journal of Operational Research*, 289(2):399–415. ISSN 0377-2217.
- Linhares, A. e Yanasse, H. H. (2002). Connections between cutting-pattern sequencing, VLSI design, and flexible machines. *Computers and Operations Research*, 29(12):1759–1772. ISSN 03050548.
- Lodi, A., Martello, S., e Monaci, M. (2002a). Two-dimensional packing problems: A survey. *European Journal of Operational Research*, 141(2):241–252. ISSN 0377-2217.
- Lodi, A., Martello, S., e Monaci, M. (2002b). Two-dimensional packing problems: A survey. *European Journal of Operational Research*, 141(2):241–252. ISSN 03772217.
- Lodi, A., Martello, S., e Vigo, D. (1999). Heuristic and metaheuristic approaches for a class of two-dimensional bin packing problems. *INFORMS Journal on Computing*, 11(4):345–357.
- Lodi, A., Martello, S., e Vigo, D. (2002c). Recent advances on two-dimensional bin packing problems. *Discrete Applied Mathematics*, 123(1):379–396. ISSN 0166-218X.
- Lodi, A., Martello, S., e Vigo, D. (2004). Models and Bounds for Two-Dimensional Level Packing Problems. *Journal of Combinatorial Optimization*, 8(3):363–379. ISSN 1382-6905.
- Macedo, R., Alves, C., e Valério de Carvalho, J. M. (2010). Arc-flow model for the two-dimensional guillotine cutting stock problem. *Computers and Operations Research*, 37(6):991–1001. ISSN 03050548.
- Madsen, O. B. G. (1979). Glass cutting in a small firm. *Mathematical Programming*, 17(1):85–90. ISSN 0025-5610.
- Madsen, O. B. G. (1988). An application of travelling-salesman routines to solve pattern-allocation problems in the glass industry. *Journal of the Operational Research Society*, 39(3):249–256. ISSN 14769360.
- Martin, M. e Yanasse, H. H. (2021). O problema de minimização de espalhamento de ordens: um estudo de formulações matemáticas. In *Anais do Simpósio Brasileiro de Pesquisa Operacional de 2021, João Pessoa, Brasil*. Disponível em: <https://proceedings.science/sbpo/sbpo-2021/trabalhos/o-problema-de-minimizacao-de-espalhamento-de-ordens-um-estudo-de-formulacoes-mat?lang=pt-br>.
- Martin, M., Yanasse, H. H., Santos, M. O., e Morabito, R. (2023). Models for two- and three-stage two-dimensional cutting stock problems with a limited number of open stacks. *International Journal of Production Research*, 61(9):2895–2916.

Puchinger, J. e Raidl, G. R. (2007). Models and algorithms for three-stage two-dimensional bin packing. *European Journal of Operational Research*, 183(3):1304–1327.

Scheithauer, G. (2018). *Introduction to cutting and packing optimization: problems, modeling approaches, solution methods*. International Series in Operations Research and Management Science. Springer.

Silva, E., Alvelos, F., e Valério de Carvalho, J. M. (2010). An integer programming model for two- and three-stage two-dimensional cutting stock problems. *European Journal of Operational Research*, 205(3):699–708. ISSN 03772217.