



**Simpósio de Métodos
Numéricos em Engenharia**

25 a 27 de outubro, 2017

Minimização do espaço ocioso no interior de contêineres: Uma abordagem exata

Deidson Vitorio Kurpel ^{a, 1}
Cassius Tadeu Scarpin ^{a, b, 2}
José Eduardo Pécora Junior ^{a, b, 3}
Cleder Marcos Schenekemberg ^{a, 4}
Nathália Cristina Ortiz da Silva ^{a, 5}
Alexandre Checoli Choueiri ^{a, 6}

^a Programa de Pós-Graduação em Métodos Numéricos em Engenharia

^b Departamento de Administração Geral e Aplicada
Universidade Federal do Paraná
Curitiba, Brasil

¹ kurpeld@gmail.com, ² cassiusts@gmail.com, ³ pecora@ufpr.br, ⁴ cledercms@hotmail.com, ⁵ ncosilva2@gmail.com,
⁶ alexandrechecoli@gmail.com

Resumo — O arranjo de caixas no interior de contêineres, de modo a minimizar o espaço ocioso é um dos principais problemas abordados na Pesquisa Operacional. Este trabalho apresenta um modelo de Programação Linear Inteira Binária para problemas de carregamento de múltiplos contêineres distintos, que busca minimizar o espaço não utilizado dos contêineres selecionados para carregar um conjunto de caixas. Dados gerados aleatoriamente foram utilizados para avaliar a formulação matemática proposta, e um *software* de otimização foi empregado para resolver os modelos gerados. A técnica de resolução apresentada neste trabalho apresentou excelentes resultados para as instâncias testadas, obtendo padrões de carregamento ótimos em um tempo computacional competitivo.

Palavras-chave—carregamento de contêineres; modelagem matemática; problemas de corte e empacotamento.

I. INTRODUÇÃO

Problemas de otimização combinatória que procuram determinar o melhor arranjo possível de unidades menores, denominadas itens, no interior de unidades maiores, chamadas objetos, são ditos problemas de corte e empacotamento [1]. Os problemas de carregamento de contêineres podem ser vistos como um caso particular dos problemas de empacotamento. Para resolvê-los, procura-se dispor, ortogonalmente e sem sobreposição, caixas com formato regular no interior de um dispositivo de carga, tais como caminhões, trens, navios e contêineres.

De acordo com [2], problemas de carregamento de contêineres podem ser classificados analisando cinco critérios: dimensão, tipo de designação, diversidade de caixas, diversidade de contêineres e forma das caixas.

Este trabalho apresenta um modelo de Programação Linear Inteira Binária (PLIB) para resolução de problemas de carregamento de múltiplos contêineres distintos, classificados, de acordo com a tipologia de [2], como *Multiple Stock-Size Cutting Stock Problem* (MSSCSP),

Residual Cutting Stock Problem (RCSP), *Multiple Bin-Size Bin Packing Problem (MBSBPP)* e *Residual Bin Packing Problem (RBPP)*. De modo genérico, todos esses problemas se caracterizam pela necessidade de minimizar o total de contêineres necessários para carregar um conjunto de caixas.

O MSSCSP procura acomodar um conjunto de caixas *fracamente heterogêneas* (muitas caixas de relativamente poucos tipos) no interior de contêineres *fracamente heterogêneos* (muitos contêineres de relativamente poucos tipos). Este problema foi tratado, entre outros, por [3], [4] e [5].

O RCSP é caracterizado pela necessidade de alocar um conjunto de itens *fracamente heterogêneos* dentro de uma seleção de contêineres *fortemente heterogêneos* (existem relativamente muitos tipos de contêineres, mas cada tipo tem pouca disponibilidade). Este tipo de problema foi abordado por [6].

Em problemas MBSBPP, busca-se empacotar um conjunto de itens *fortemente heterogêneos* (existem relativamente muitos tipos de caixas, mas cada tipo tem pouca disponibilidade) no interior de uma seleção de contêineres *fracamente heterogêneos*. Este problema foi abordado por [4] e [7].

Já o RBPP é caracterizado pela necessidade de acondicionar um conjunto de itens *fortemente heterogêneos* no interior de uma seleção de contêineres *fortemente heterogêneos*. Este problema foi tratado por [8], [9] e [10].

Para avaliar o modelo de PLIB apresentado nesse trabalho, um conjunto de instâncias foi gerado aleatoriamente, de modo a criar conjuntos de caixas tanto fraca quanto fortemente heterogêneas. Testes computacionais utilizando o *software* de otimização CPLEX® *Interactive Optimizer* foram efetuados para avaliar a qualidade das soluções obtidas pela abordagem proposta para as instâncias geradas.

Este trabalho está organizado da seguinte maneira: A presente Seção apresenta o problema de carregamento de múltiplos contêineres distintos. A Seção II detalha a abordagem proposta neste artigo, enquanto a Seção III explica como foram geradas as instâncias utilizadas para realização dos testes computacionais. Os resultados destes testes são apresentados na Seção IV. Por fim, na Seção V são feitas as considerações finais e sugestões para trabalhos futuros.

II. MODELO DE PROGRAMAÇÃO LINEAR INTEIRA BINÁRIA

A abordagem proposta para resolução do problema de carregamento de múltiplos contêineres baseia-se nos modelos apresentados nos trabalhos de [11] e [12].

Sejam m caixas do tipo i , com comprimento l_i , largura w_i , altura h_i , volume (ou valor associado) v_i e disponibilidade b_i . C tipos de contêineres, com um total de T_k contêineres do tipo k , com comprimento L_k , largura W_k e altura H_k . Tomando o sistema de coordenadas cartesiano, seja (p, q, r) o vértice inferior frontal esquerdo de uma caixa

do tipo i , posicionada dentro de um contêiner do tipo k , conforme mostrado na Fig. 1.

Em sua abordagem para o problema de carregamento de um único contêiner, [12] apresentam conjuntos que indicam as possíveis posições que uma caixa pode ocupar dentro de um contêiner. Esses conjuntos são construídos utilizando os chamados padrões normais (ou combinações cônicas) definidos por [13] e [14]. Embora não possuam todas as possíveis coordenadas que uma caixa pode assumir em um contêiner, os conjuntos definidos por [12] possuem, em geral, menor cardinalidade, podendo ser utilizados sem perda de generalidade.

Para o problema de carregamento de múltiplos contêineres distintos apresentado neste trabalho, os conjuntos definidos por [12] podem ser adaptados da seguinte maneira:

$$X_k = \{p \in \mathbb{Z} \mid p = \sum_{i=1}^m \beta_i \cdot l_i, 0 \leq p \leq L_k - \min_i(l_i), 0 \leq \beta_i \leq b_i, \forall i\} \quad (1)$$

$$Y_k = \{q \in \mathbb{Z} \mid q = \sum_{i=1}^m \beta_i \cdot w_i, 0 \leq q \leq W_k - \min_i(w_i), 0 \leq \beta_i \leq b_i, \forall i\} \quad (2)$$

$$Z_k = \{r \in \mathbb{Z} \mid r = \sum_{i=1}^m \beta_i \cdot h_i, 0 \leq r \leq H_k - \min_i(h_i), 0 \leq \beta_i \leq b_i, \forall i\} \quad (3)$$

$$X_{ik} = \{p \in X_k \mid 0 \leq p \leq L_k - l_i, i = 1, \dots, m\} \quad (4)$$

$$Y_{ik} = \{q \in Y_k \mid 0 \leq q \leq W_k - w_i, i = 1, \dots, m\} \quad (5)$$

$$Z_{ik} = \{r \in Z_k \mid 0 \leq r \leq H_k - h_i, i = 1, \dots, m\} \quad (6)$$

Os conjuntos (1)-(3) definem as coordenadas que o vértice inferior frontal esquerdo de uma caixa pode ocupar em relação às dimensões de um contêiner do tipo k , enquanto (4)-(6) indicam as posições que uma caixa do tipo i pode assumir dentro de um contêiner do tipo k .

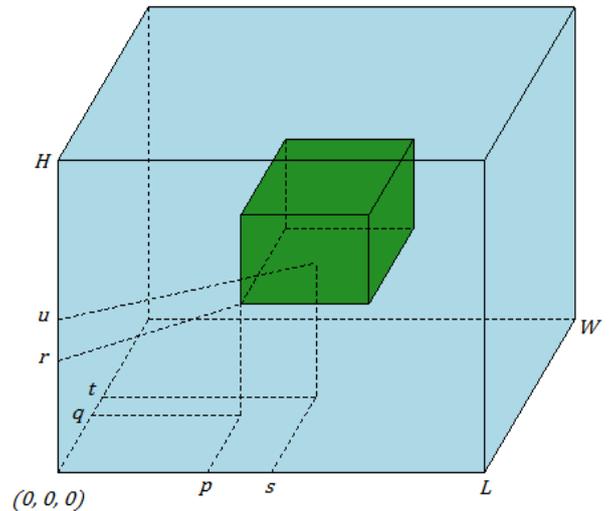


Figura 1. Caixa do tipo i no interior de um contêiner do tipo k , com seu vértice inferior frontal esquerdo no ponto (p, q, r) e um ponto (s, t, u) ocupado pelo interior da caixa. FONTE: Adaptado de [12].

O modelo de PLIB proposto neste trabalho é definido da seguinte maneira:

$$\text{Minimizar } \sum_{k=1}^C \sum_{j=1}^{T_k} L_k \cdot W_k \cdot H_k \cdot e_{jk} - \sum_{i=1}^m v_i \cdot b_i \quad (7)$$

Sujeito a:

$$\sum_{i=1}^m \sum_{\{p \in X_{ik} \mid s-l_i+1 \leq p \leq s\}} \sum_{\{q \in Y_{ik} \mid t-w_i+1 \leq q \leq t\}} \quad (8)$$

$$\sum_{\{r \in Z_{ik} \mid u-h_i+1 \leq r \leq u\}} x_{ijkpqr} \leq e_{jk}$$

$$\forall k = 1, \dots, C, e j = 1, \dots, T_k$$

$$\sum_{k=1}^C \sum_{j=1}^{T_k} \sum_{p \in X_{ik}} \sum_{q \in Y_{ik}} \sum_{r \in Z_{ik}} x_{ijkpqr} = b_i \quad (9)$$

$$i = 1, \dots, m$$

$$x_{ijkpqr} \in \{0,1\}, \quad i = 1, \dots, m, \quad k = 1, \dots, C$$

$$j = 1, \dots, T_k, \quad p \in X_{ik}, \quad q \in Y_{ik}, \quad r \in Z_{ik} \quad (10)$$

$$s \in X_k, \quad t \in Y_k, \quad u \in Z_k, \quad e_{jk} \in \{0,1\}$$

As variáveis x_{ijkpqr} indicam se um caixa do tipo i tem seu vértice inferior frontal esquerdo no ponto (p, q, r) do j -ésimo contêiner do tipo k , enquanto as variáveis e_{jk} indicam se o j -ésimo contêiner do tipo k é utilizado. A função objetivo (7) procura minimizar o volume não utilizado do conjunto de contêineres selecionados para acomodar as caixas. O somatório em (8) garante que não há sobreposição entre caixas no interior do j -ésimo contêiner do tipo k , caso o mesmo tenha sido selecionado, impedindo que qualquer ponto (s, t, u) no interior de uma caixa com vértice inferior frontal esquerdo no ponto (p, q, r) seja usado como vértice de outra caixa. A restrição (9) exige que todas as caixas do tipo i façam parte do padrão de carregamento, enquanto (10) apresenta o domínio das variáveis de decisão.

III. GERAÇÃO DE INSTÂNCIAS

O processo empregado para geração de conjuntos de teste utilizado nesse trabalho é baseado na metodologia proposta por [12], consistindo em determinar valores inteiros que irão compor as dimensões, a quantidade e os tipos de caixas e contêineres de cada instância a ser avaliada.

Um total de 40 instâncias foram criadas. A quantidade de tipos de contêineres disponíveis para carregamento corresponde a um valor inteiro escolhido por meio da distribuição uniforme no intervalo de um a três, ou seja, $C \in [1, 3]$, $C \in \mathbb{Z}$. As dimensões (L_k, W_k, H_k) de um contêiner do

tipo k , $k = 1, \dots, C$, são valores inteiros uniformemente escolhidos no intervalo [10, 25].

Quatro tipos distintos de caixas foram considerados para construção dos conjuntos de teste, ou seja, $m = 1, 2, 3, 4$. Visando criar conjuntos tanto com caixas fracamente heterogêneas quanto fortemente heterogêneas, duas estratégias distintas foram empregadas para determinar a dimensão das caixas.

Em vinte conjuntos de testes, as dimensões das caixas foram determinadas por meio da distribuição uniforme nos seguintes conjuntos: $[l_i] \in [0,25 \cdot \mathbb{L}; 0,75 \cdot \mathbb{L}]$, $[w_i] \in [0,25 \cdot \mathbb{W}; 0,75 \cdot \mathbb{W}]$, $[h_i] \in [0,25 \cdot \mathbb{H}; 0,75 \cdot \mathbb{H}]$, onde $\mathbb{L} \leq L_k$, $\mathbb{W} \leq W_k$ e $\mathbb{H} \leq H_k$, $\forall k$. Para cada valor de m , cinco instâncias foram construídas. As caixas geradas por meio desse procedimento constituem o Grupo 1 de conjuntos de teste deste trabalho (instâncias 1 a 20).

As caixas das demais instâncias tiveram suas dimensões uniformemente escolhidas nos seguintes conjuntos: $[l_i] \in [0,10 \cdot \mathbb{L}; 0,50 \cdot \mathbb{L}]$, $[w_i] \in [0,10 \cdot \mathbb{W}; 0,50 \cdot \mathbb{W}]$, $[h_i] \in [0,10 \cdot \mathbb{H}; 0,50 \cdot \mathbb{H}]$, onde $\mathbb{L} \leq L_k$, $\mathbb{W} \leq W_k$ e $\mathbb{H} \leq H_k$, $\forall k$. Para cada valor de m , foram criados cinco conjuntos de teste. As caixas geradas por meio desse procedimento constituem o Grupo 2 de instâncias (conjuntos de teste 21 a 40).

A disponibilidade b_i de cada tipo de caixa, $i = 1, \dots, m$, foi obtida escolhendo uniformemente um valor inteiro no intervalo [10, 200] tanto para as instâncias do Grupo 1 quanto do Grupo 2. Por fim, para determinar a quantidade de contêineres disponíveis, calculou-se o volume dos C tipos de contêineres e, destes, foi escolhido o contêiner de menor volume (V_{menor}). A quantidade de cada tipo de contêiner para carregamento é dada pela expressão

$$T_k = 2 \cdot \frac{\sum_{i=1}^m v_i}{V_{menor}}, \quad \forall k \quad (11)$$

IV. RESULTADOS COMPUTACIONAIS

O modelo de PLIB apresentado neste trabalho foi implementado em VB.NET, e cada instância foi resolvida pelo *software* de otimização CPLEX© *Interactive Optimizer*, versão 12.6.0.0, com parâmetros padrão. Um limite de tempo de 3600 segundos (1 hora) foi estabelecido para a execução dos testes.

A Tabela I mostra a quantidade de contêineres utilizados, a ocupação mínima, máxima e média dos contêineres e o tempo computacional obtidos na execução dos testes para cada cenário considerado na avaliação do modelo.

A abordagem proposta neste trabalho apresenta bons resultados para os problemas abordados. Os quarenta conjuntos de testes foram resolvidos otimamente, com tempo computacional médio de 19,05 segundos. Em virtude da maneira com que as dimensões das caixas foram geradas, as instâncias do Grupo 2 apresentam, de modo geral, maior utilização do volume interno dos contêineres, uma vez que as caixas que compõem estes conjuntos de teste possuem dimensões pequenas em relação aos contêineres. Entretanto,

como apresentam maior número de variáveis e restrições, o tempo computacional é mais elevado para resolvê-los quando comparado com o tempo gasto para obter a solução das instâncias do Grupo 1.

TABELA I: RESULTADOS DOS TESTES COMPUTACIONAIS

#	Quantidade de Contêineres	Ocupação			Tempo (s)
		Mínima (%)	Média (%)	Máxima (%)	
1	4	13,85	35,73	57,60	0,31
2	4	63,92	66,05	68,18	0,30
3	12	13,66	20,49	27,31	0,02
4	18	35,03	40,87	46,70	0,05
5	3	65,28	78,72	92,16	0,50
6	11	51,32	57,49	63,65	0,08
7	55	35,79	40,38	44,97	0,03
8	5	34,36	55,43	76,50	0,80
9	3	79,80	86,57	93,33	8,48
10	5	78,11	87,54	96,97	0,53
11	7	60,22	72,03	83,84	21,08
12	9	63,21	71,75	80,28	1,27
13	7	68,90	79,64	90,38	0,42
14	60	19,24	39,23	59,21	0,05
15	14	53,86	73,99	94,12	0,08
16	9	38,29	63,69	89,09	4,42
17	6	57,97	61,72	65,47	18,27
18	72	18,55	28,64	38,73	0,08
19	52	31,58	42,44	53,29	0,14
20	10	25,67	59,42	93,16	0,56
21	2	40,00	68,00	96,00	0,17
22	1	80,77	80,77	80,77	0,39
23	4	75,63	75,63	75,63	0,03
24	9	23,15	53,24	83,33	0,23
25	2	70,40	83,20	96,00	0,31
26	4	75,00	78,95	82,89	38,88
27	3	59,88	66,01	72,14	0,11
28	4	73,59	77,69	81,79	0,77
29	1	83,80	83,80	83,80	0,80
30	13	55,62	76,30	96,97	0,05
31	7	51,13	67,24	83,34	5,63
32	7	82,42	87,77	93,12	22,01
33	7	57,76	78,88	100,00	0,47
34	7	52,51	66,64	80,77	2,56
35	4	38,19	69,10	100,00	1,26
36	16	29,76	54,07	78,38	10,59
37	14	64,10	72,05	80,00	5,91
38	9	58,19	71,52	84,85	611,13
39	7	68,61	74,53	80,45	1,41
40	15	41,55	58,30	75,04	1,64
Grupo 1	384	45,43	58,09	70,75	
Grupo 2	136	59,10	72,18	85,26	
Geral	520	52,27*	65,14*	78,01*	19,05**

* Ocupação média

** Tempo médio

FONTE: O autor.

A Fig. 2 apresenta o padrão de carregamento obtido para o conjunto de teste 28. Todas as caixas estão posicionadas

inteiramente dentro dos contêineres, de modo ortogonal e sem sobreposição. O modelo PLIB apresentado neste trabalho adaptado para considerar algumas exigências práticas comuns em situações reais, descritas por [15], como a estabilidade da carga, visto que, embora a solução para este conjunto de teste seja factível, há caixas parcialmente apoiadas ou mesmo “flutuando” no interior dos contêineres. Ainda assim, a abordagem proposta determinou um padrão de carregamento ótimo para o conjunto de teste 28, com ocupação média dos contêineres de 77,69%.

V. CONSIDERAÇÕES FINAIS

Neste trabalho foi apresentado um modelo de Programação Linear Inteira Binária, com base nas formulações de [11] e [12], para abordar o problema de carregamento de múltiplos contêineres distintos. Este problema visa selecionar um subconjunto de contêineres para alocar todas as caixas disponíveis, de modo que o espaço não utilizado dos contêineres seja minimizado. O *software CPLEX© Interactive Optimizer* foi utilizado para efetuar testes computacionais com dados gerados aleatoriamente, de modo a avaliar a abordagem apresentada.

A formulação proposta pode ser estendida de modo a considerar as exigências práticas listadas por [15]. No entanto, vale a pena ressaltar que a qualidade da solução da abordagem proposta depende da quantidade de tipos de caixas disponíveis e do tamanho relativo das caixas em comparação com as dimensões dos contêineres, visto que conjuntos de caixas fortemente heterogêneos, em geral, geram problemas mais difíceis de serem resolvidos, devido ao grande número de variáveis e restrições.

Como sugestão para trabalhos futuros, podem ser explorados meios alternativos de geração dos conjuntos de possíveis posições das caixas, além de estudar a possibilidade de estender o modelo proposto para os diferentes tipos de problemas de carregamento de contêiner definidos por [2].

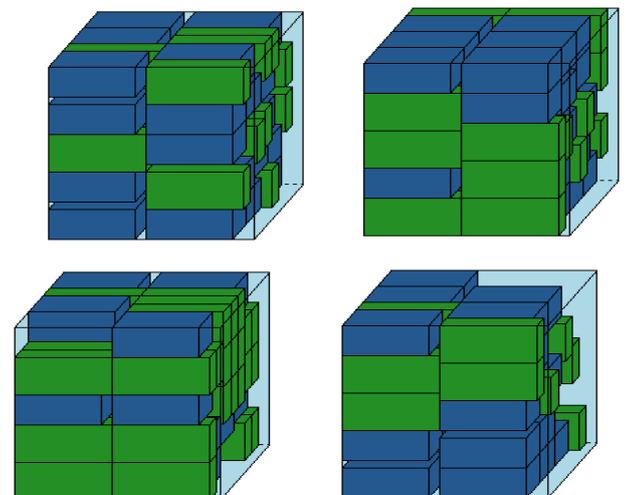


Figura 2. Padrão de carregamento obtido para o conjunto de teste 28. FONTE: O autor.

REFERÊNCIAS

- [1] H. Dyckhoff, "A typology of cutting and packing problems," *Eur. J. Oper. Res.*, vol. 44, no. 2, pp. 145–159, 1990.
- [2] G. Wäscher, H. Haußner, and H. Schumann, "An improved typology of cutting and packing problems," *Eur. J. Oper. Res.*, vol. 183, no. 3, pp. 1109–1130, 2007.
- [3] M. Eley, "A bottleneck assignment approach to the multiple container loading problem," *OR Spectr.*, vol. 25, no. 1, pp. 45–60, 2003.
- [4] L. Brunetta and P. Grégoire, "A General Purpose Algorithm for Three-Dimensional Packing," *INFORMS J. Comput.*, vol. 17, no. 3, pp. 328–338, Jul. 2005.
- [5] C. H. Che, W. Huang, A. Lim, and W. Zhu, "The multiple container loading cost minimization problem," *Eur. J. Oper. Res.*, vol. 214, no. 3, pp. 501–511, 2011.
- [6] R. Andrade, E. G. Birgin, and R. Morabito, "Two-stage two-dimensional guillotine cutting stock problems with usable leftover," *Int. Trans. Oper. Res.*, vol. 23, no. 1–2, pp. 121–145, 2016.
- [7] S. Ceschia and A. Schaerf, "Local search for a multi-drop multi-container loading problem," *J. Heuristics*, vol. 19, no. 2, pp. 275–294, 2013.
- [8] Z. H. Jin, K. Ohno, and J. L. Du, "An efficient approach for the three-dimensional container packing problem with practical constraints," *Asia-Pacific J. Oper. Res.*, vol. 21, no. 3, pp. 279–295, 2004.
- [9] A. Lim and X. Zhang, "The Container Loading Problem," in *Proceedings of the 2005 ACM Symposium on Applied Computing*, 2005, pp. 913–917.
- [10] A. de Almeida and M. B. Figueiredo, "A particular approach for the Three-dimensional Packing Problem with additional constraints," *Comput. Oper. Res.*, vol. 37, no. 11, pp. 1968–1976, 2010.
- [11] C. S. Chen, S. M. Lee, and Q. S. Shen, "An analytical model for the container loading problem," *Eur. J. Oper. Res.*, vol. 80, no. 1, pp. 68–76, 1995.
- [12] L. Junqueira, R. Morabito, and D. Sato Yamashita, "Three-dimensional container loading models with cargo stability and load bearing constraints," *Comput. Oper. Res.*, vol. 39, no. 1, pp. 74–85, Jan. 2012.
- [13] J. C. Herz, "Recursive Computational Procedure for Two-dimensional Stock Cutting," *IBM J. Res. Dev.*, vol. 16, no. 5, pp. 462–469, Sep. 1972.
- [14] N. Christofides and C. Whitlock, "An Algorithm for Two-Dimensional Cutting Problems," *Oper. Res.*, vol. 25, no. 1, pp. 30–44, Feb. 1977.
- [15] E. E. Bischoff and M. S. W. Ratcliff, "Issues in the development of approaches to container loading," *Omega*, vol. 23, no. 4, pp. 377–390, 1995.