

Uma Proposta de Formulação Matemática para o Problema de Carregamento de Múltiplos Contêineres Heterogêneos com Restrição Adicional de Separação de Itens

Cleder Marcos Schenekemberg (Universidade Federal do Paraná) cledercms@hotmail.com
Cassius Tadeu Scarpin (Universidade Federal do Paraná) cassiusts@gmail.com

Resumo:

Neste trabalho apresenta-se uma proposta de resolução para o Problema de Carregamento de Múltiplos Contêineres Heterogêneos. Este problema consiste em empacotar caixas retangulares ortogonalmente e sem sobreposição dentro de contêineres, de modo a maximizar a ocupação do espaço disponível. Com base em propostas da literatura, este trabalho visa apresentar uma formulação matemática para o problema, por meio de um modelo de Programação Linear Inteira 0-1, que permite assegurar a separação de determinados itens em contêineres distintos. Embora comum em situações reais, esta restrição prática é raramente tratada em trabalhos correlatos. Testes computacionais com instâncias da literatura foram efetuados para avaliar os modelos gerados. Os resultados obtidos mostram que, apesar da formulação proposta se limitar a resolver problemas relativamente simples, a mesma descreve adequadamente as considerações tratadas.

Palavras chave: Formulação Matemática, Problema de Carregamento de Contêiner, Restrição de Separação de Itens.

A Proposal of Mathematical Formulation for Multiple Heterogeneous Container Loading Problem with Additional Constraint of Separation of Items

Abstract

In this work we present a proposal to solve the Multiple Heterogeneous Container Loading Problem. This Problem consists of packing orthogonally and without overlap rectangular boxes inside containers, in order to maximize the occupation of available space. Based on literature proposals, this paper presents a mathematical formulation of the problem, through a model of 0-1 Integer Linear Programming, which ensures the separation of certain items in separate containers. Although common in real situations, this practice consideration is rarely treated in related work. Computational tests with instances literature were performed to evaluate the generated models. The results show that although the proposed formulation is limited to solve relatively simple problems, it adequately describes the considerations addressed.

Key-words: Mathematical Formulation, Container Loading Problem, Constraint of Separation of Items

1. Introdução

Com a globalização econômica e o fluxo constante de mercadorias sendo transportada pelos mais variados modais e dispositivos de transporte, a otimização dos recursos disponíveis mostra ser um fator decisivo e, até mesmo, essencial na sobrevivência das organizações. O planejamento estratégico do transporte de matérias-primas e produtos acabados, ou ainda, a ausência de planejamento, pode impactar diretamente no custo associado às mercadorias destinadas aos consumidores.

Neste sentido, este trabalho visa contribuir com uma proposta de resolução para um tipo clássico de Problema de Corte e Empacotamento (PCE), conhecido como Problema de Carregamento de Contêiner (*Container Loading Problem* - CLP). O CLP engloba inúmeras aplicações práticas, tendo diferentes objetivos e restrições de carregamento. Para resolvê-lo, busca-se dispor caixas retangulares ortogonalmente dentro de contêineres, de tal modo que o volume (valor) da carga seja maximizado, ou em outros casos, que a quantidade de contêineres necessários para o carregamento seja minimizado.

Segundo Bischoff e Ratcliff (1995) um método de resolução para ser verdadeiramente útil para o CLP, deve levar em consideração, além dos objetivos primários, certas considerações práticas comumente encontradas em situações reais. Estabilidade da carga, restrição de limitação de peso dentro do contêiner, prioridades de envio, restrição de manuseio e separação de itens, são algumas das doze restrições elencadas pelos autores que podem ser incorporadas no carregamento da carga.

O objetivo deste trabalho é apresentar uma proposta de formulação matemática para o CLP, com base em um modelo de Programação Linear Inteira 0 – 1 da literatura, capaz de considerar o carregamento de múltiplos contêineres e a restrição prática de separação de itens. Segundo Bortfeldt e Wäscher (2013), apesar do aumento significativo no número de publicações na literatura especializada, ainda há considerações práticas pouco ou raramente abordadas, como, por exemplo, a restrição prática tratada neste trabalho.

Algumas formulações matemáticas para o CLP podem ser obtidas na literatura. Por exemplo, pode-se citar Tsai, Malstrom e Kuo (1993), Mohanty, Mathur e Ivancic (1994), Chen, Lee, e Shen (1995), Hifi *et al* (2010) e Junqueira, Morabito e Yamashita (2012). Embora, cabe destacar que algumas destas abordagens se limitam quanto à consideração de exigências práticas. De modo alternativo aos modelos matemáticos, é comum encontrar outras metodologias de resolução para o CLP. Algoritmos heurísticos e metaheurísticos podem ser obtidos, por exemplo, em Gehring e Bortfeldt (1997), Pisinger (2002), Eley (2003) e Altarazi (2013). Da mesma forma, métodos exatos, de aproximação e de busca em árvore podem ser encontrados em Mizawa e Wakabayashi (1997), Martello, Pisinger e Vigo (2000), Oliveira e Morabito (2006) e Ren, Tian e Sawaragi (2011).

Este trabalho está organizado da seguinte forma: na seção 2, o problema de carregamento de múltiplos contêineres heterogêneos com restrição de separação de itens é brevemente explicado, juntamente com um modelo de Programação Linear Inteira 0 – 1 para descrevê-lo; na seção 3, apresenta-se um procedimento heurístico capaz de reduzir a complexidade da formulação proposta; na seção 4, os resultados computacionais obtidos na avaliação dos modelos são apresentados e analisados; finalmente, na seção 5 são apresentadas as principais conclusões alcançadas, bem como perspectivas para trabalhos futuros.

2. Formulação Matemática

Nesta seção, inicialmente busca-se descrever brevemente o Problema de Carregamento de Contêiner. Em seguida, com base em um modelo da literatura, apresenta-se uma proposta de formulação matemática para o carregamento de múltiplos contêineres heterogêneos. Por fim,

a consideração prática de separação de itens é formulada como restrição do modelo matemático proposto.

2.1. Descrição do Problema

O CLP é um caso particular dos Problemas de Corte e Empacotamento. Os PCE's, como muitos dos problemas de natureza combinatória, abrangem uma variedade de problemas caracterizados por uma estrutura comum, que consistem em combinar unidades menores, denominadas *itens* (peças, caixas, entre outras) dentro de unidades maiores, chamadas de *objetos* (chapas, contêineres, entre outras), com a finalidade de otimizar certos objetivos pré-determinados. Esta combinação, dentro de suas mais variadas versões, pode determinar o padrão de corte de peças para composição de itens encomendados, como também o empacotamento de caixas em contêineres.

Genericamente, os Problemas de Carregamento de Contêiner, de acordo com a sua designação, podem ser divididos em dois tipos de subproblemas. Nos primeiros, denominados problemas de minimização das entradas, o espaço de armazenagem é suficiente para a alocação de todas as caixas, visto que o número de contêineres é irrestrito. Logo, busca-se minimizar o número de contêineres necessários para carregar todos os itens disponíveis. Já os demais, chamados problemas de maximização das saídas, o espaço de um número limitado de contêineres não é suficiente para armazenar todos os itens. O objetivo é, então, determinar um subconjunto de caixas que maximiza a utilização do volume dos contêineres ou valor associado à carga.

Uma classificação mais adequada para os CLP's pode ser obtida de acordo com a tipologia de Wäscher, Haußner e Schumann (2007) proposta para categorizar os PCE's. Os autores, por meio de combinações de critérios básicos, como dimensionalidade, variedade dos itens, variedade dos objetos, forma dos itens e tipo de designação (minimização ou maximização), classificam os mais variados tipos de PCE's, e consequentemente os CLP's, em problemas do tipo básico, intermediário e refinado.

Este trabalho aborda o carregamento de múltiplos contêineres distintos. Com base na tipologia de Wäscher, Haußner e Schumann (2007), este problema pode ser dividido em dois novos subproblemas de acordo com a variedade dos itens. O primeiro, chamado *Multiple Heterogeneous Knapsack Problem* (MHKP), visa empacotar um subconjunto de caixas fortemente heterogêneas em um conjunto de contêineres heterogêneos, isto é, contêineres não idênticos, tal que o valor (volume) total das caixas carregadas seja maximizado. Já o segundo, conhecido como *Multiple Heterogeneous Large Object Placement Problem* (MHLOPP), tem por objetivo empacotar um subconjunto de caixas fracamente heterogêneas em um conjunto de contêineres heterogêneos, tal que o valor (volume) total das caixas carregadas seja maximizado.

2.2. Modelo Matemático

Em sua abordagem ao CLP, Junqueira, Morabito e Yamashita (2012) apresentam um modelo matemático baseado em Programação Linear Inteira 0 – 1 para o carregamento de um único contêiner. Na sequência deste trabalho, por meio de adaptações, esta formulação matemática será estendida de forma a considerar o carregamento de múltiplos contêineres heterogêneos (não idênticos).

Considere c tipos de contêineres com largura L_k , comprimento W_k , altura H_k e disponibilidade d_k (onde $1 \leq k \leq c$), que devem ser carregados com m tipos de caixas distintas. Cada caixa do tipo i , $i = 1, \dots, m$, que pode ser alocada no interior de um dos contêineres do tipo k , tem largura l_i , comprimento w_i , altura h_i e volume (ou valor associado à caixa) v_i . Existe um total de b_i caixas do tipo i que podem ser arranjadas nos contêineres,

logo, o problema é restrito quanto ao número de itens disponíveis para carregamento.

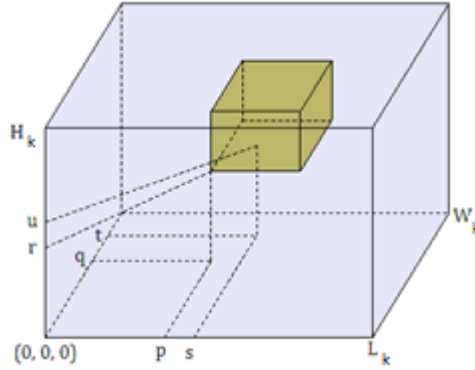


Figura 1 - Uma caixa do tipo i alocada no vértice (p, q, r) de um contêiner do tipo k

Tomando o sistema de coordenadas cartesiano, seja (p, q, r) o vértice inferior frontal esquerdo de uma caixa qualquer no interior de um contêiner do tipo k (veja Figura 1). Os conjuntos abaixo, adaptados de Junqueira, Morabito e Yamashita (2012), indicam as possíveis posições que uma caixa i (onde $i = 1, \dots, m$) pode assumir em relação às dimensões do(s) contêiner(es) do tipo k . Ressalta-se que os itens possuem orientação fixa, isto é, a largura, comprimento e altura das caixas devem ser alocadas paralelamente às respectivas dimensões do contêiner designado, impossibilitando o movimento de rotação das caixas dentro dos contêineres.

$$X_k = \{p | 0 \leq p \leq L_k - \min_i(l_i), \quad p \in \mathbb{Z}, \quad i = 1, \dots, m\} \quad (1)$$

$$Y_k = \{q | 0 \leq q \leq W_k - \min_i(w_i), \quad q \in \mathbb{Z}, \quad i = 1, \dots, m\} \quad (2)$$

$$Z_k = \{r | 0 \leq r \leq H_k - \min_i(h_i), \quad r \in \mathbb{Z}, \quad i = 1, \dots, m\} \quad (3)$$

Considere ainda os conjuntos:

$$X_{ik} = \{p \in X_k | 0 \leq p \leq L_k - l_i\} \quad (4)$$

$$Y_{ik} = \{q \in Y_k | 0 \leq q \leq W_k - w_i\} \quad (5)$$

$$Z_{ik} = \{r \in Z_k | 0 \leq r \leq H_k - h_i\} \quad (6)$$

Embora os elementos dos conjuntos acima sejam restritos a valores inteiros, essa condição pode ser facilmente relaxada. As variáveis de decisão do modelo são do tipo binário, definidas da seguinte maneira:

$$x_{ijkpqr} = \begin{cases} 1, & \text{Se uma caixa do tipo } i \text{ tem seu vértice inferior frontal} \\ & \text{esquerdo no ponto } (p, q, r) \text{ do } j - \text{ésimo contêiner do tipo } k, \\ & \text{tal que: } 0 \leq p \leq L_k - l_i, \quad 0 \leq q \leq W_k - w_i, \\ & \quad 0 \leq r \leq H_k - h_i, \quad 1 \leq j \leq d_k; \\ 0, & \text{Caso contrário} \end{cases} \quad (7)$$

A seguir apresenta-se a formulação matemática para o problema com base no modelo de carregamento de um único contêiner de Junqueira, Morabito e Yamashita (2012). Por simplicidade, será mantida a notação original definida pelos autores. O modelo matemático para o Problema de Carregamento de Múltiplos Contêineres Heterogêneos, que será chamado de modelo base no decorrer deste trabalho, é obtido da seguinte maneira:

$$\text{Maximizar } Z = \sum_{k=1}^c \sum_{j=1}^{d_k} \sum_{i=1}^m \sum_{p \in X_{ik}} \sum_{q \in Y_{ik}} \sum_{r \in Z_{ik}} v_i \cdot x_{ijkpqr} \quad (8)$$

Sujeito a

$$\sum_{i=1}^m \sum_{\{p \in X_{ik} | s-l_i+1 \leq p \leq s\}} \sum_{\{q \in Y_{ik} | t-w_i+1 \leq q \leq t\}} \sum_{\{r \in Z_{ik} | u-h_i+1 \leq r \leq t\}} x_{ijkpqr} \leq 1, \quad (9)$$

Onde $s \in X_k, t \in Y_k, u \in Z_k, \forall k = 1, \dots, c$ e $\forall j = 1, \dots, d_k$

$$\sum_{k=1}^c \sum_{j=1}^{d_k} \sum_{p \in X_{ik}} \sum_{q \in Y_{ik}} \sum_{r \in Z_{ik}} x_{ijkpqr} \leq b_i, \quad \forall i = 1, \dots, m \quad (10)$$

$$\text{Onde } x_{ijkpqr} \in \{0, 1\} \quad (11)$$

No modelo descrito por (8) – (11), a função objetivo (8) visa maximizar o volume (valor) das caixas empacotadas nos contêineres disponíveis. As restrições (9) garantem que duas ou mais caixas não se sobreponham, isto é, que não ocupem pontos internos em comum dentro de um mesmo contêiner. As expressões (10) impedem que caixas sejam carregadas além da disponibilidade máxima permitida. Por fim, em (11) define-se o domínio das variáveis de decisão.

2.3. Restrição Prática de Separação de Itens

Em muitos problemas práticos é frequentemente desejável que caixas de dois (ou mais) tipos sejam alocadas em contêineres distintos, por exemplo, produtos alimentícios e agrotóxicos. Embora com grande aplicabilidade em situações reais de carregamento de contêiner, esta restrição adicional não tem recebido muita atenção na literatura especializada. Segundo Bortfeldt e Wäscher (2013), apenas Eley (2003), por meio de um algoritmo heurístico, aborda esta exigência prática.

A proposta de formulação matemática apresentada neste trabalho visa assegurar a separação de diversos tipos de itens em contêineres distintos. Mais precisamente, garantir que para cada contêiner disponível seja empacotado no máximo um dos tipos de caixas do grupo de itens que se deseja separar.

Para isso, define-se um conjunto de tipos de caixas que devem ser separadas. Este conjunto, que será denotado por B , é formado apenas pelos tipos de itens, entre os m tipos disponíveis, que devem ser obrigatoriamente alocados em contêineres distintos. Cabe destacar que os demais tipos de caixas, isto é, que não pertençam ao conjunto B , podem ser empacotadas simultaneamente nos contêineres disponíveis.

A formulação matemática da restrição prática de separação de itens é dada por:

$$\sum_{p \in X_{ak}} \sum_{q \in Y_{ak}} \sum_{r \in Z_{ak}} x_{ajkpqr} \leq b_a \cdot y_{ajk}, \quad \forall a \in B, \forall k = 1, \dots, c \text{ e } \forall j = 1, \dots, d_k \quad (12)$$

$$\sum_{a \in B} y_{ajk} \leq 1, \forall k = 1, \dots, c \text{ e } \forall j = 1, \dots, d_k \quad (13)$$

$$\text{Onde } y_{ajk} \in \{0, 1\}, \forall a \in B, \forall k = 1, \dots, c \text{ e } \forall j = 1, \dots, d_k \quad (14)$$

De acordo com (14), novas variáveis são definidas que, combinadas com (13), asseguram,

por meio de (12), que no máximo um tipo de caixa do conjunto B pode ser carregado no j -ésimo contêiner do tipo k . Observe que, como definido em (13), para o j -ésimo contêiner do tipo k , apenas uma variável binária pode assumir o valor unitário, sendo assim, indicando o único tipo de caixa do conjunto B que pode ser alocada neste contêiner específico.

3. Algoritmo Gerador de Números Úteis

A modelagem matemática apresentada na seção anterior, que descreve o modelo base e a restrição de separação de itens, foi formulada de acordo com os conjuntos de pontos candidatos a vértice das caixas, isto é, os conjuntos (1) – (6). No entanto, uma vez que estes conjuntos possuam todas as possíveis posições que uma caixa pode assumir ao longo dos eixos de um contêiner, apenas problemas meramente realísticos podem ser resolvidos dentro de um limite de tempo aceitável. Junqueira, Morabito e Yamashita (2012) descrevem esta limitação da modelagem do problema, onde buscam minimizar a cardinalidade destes conjuntos por meio dos padrões normais e seções cônicas de Herz (1972) e Christofides e Whitlock (1977).

O Algoritmo Gerador de Números Úteis (AGNU) foi elaborado com base na heurística proposta por Carnieri, Mendoza e Gavinho (1994) para abordar o problema de corte bidimensional guilhotinado de chapas de madeira. Os autores apresentam um procedimento que permite a geração de apenas um grupo potencialmente útil de valores que devem ser avaliados como possíveis pontos de corte, eliminando pontos desnecessários na busca pela solução ótima. Este procedimento, em termos dos pontos gerados, é equivalente a metodologia utilizada por Junqueira, Morabito e Yamashita (2012), diferindo apenas no processo de geração dos valores.

Para descrever o algoritmo, considere um contêiner genérico do tipo g com dimensões (L_g, W_g, H_g) , que deve ser carregado com m tipos de caixas de diferentes tamanhos, ou seja, (l_i, w_i, h_i) , para $i = 1, \dots, m$. Em sua formulação original, o AGNU comporta apenas o caso unidimensional, sendo assim, o algoritmo deve ser reaplicado para cada dimensão específica de cada tipo de contêiner. Para efeito de apresentação, optou-se pelo eixo X do contêiner do tipo g , entretanto, os procedimentos descritos no algoritmo podem ser diretamente estendidos aos demais eixos do contêiner.

Passo 1.	Calcule $\beta = \min \{l_i, i = 1, 2, \dots, m\}$. Defina $X_g^{(m+1)} = F^{(m+1)} = \{0\}$, e faça $k = m$.
Passo 2.	Calcule $F^{(k)} = \{x + \varepsilon \cdot l_k \mid x \in X_g^{(k+1)}, \varepsilon = 1, 2, \dots, \text{e } L_g - (x + \varepsilon \cdot l_k) \geq \beta\}$.
Passo 3.	Faça $X_g^{(k)} = F^{(k)} \cup X_g^{(k+1)}$. Remova os números equivalentes de $X_g^{(k)}$.
Passo 4.	Se $k > 1$, faça $k \leftarrow k - 1$ e volte ao Passo 2 . Caso contrário, pare. O conjunto $X_g^{(1)}$, ou simplesmente X_g , contém a lista dos números úteis do contêiner do tipo g gerados a partir do eixo X .

Figura 2 - Pseudocódigo do algoritmo gerador de números úteis

No pseudocódigo do algoritmo, descrito na Figura 2, o **Passo 1** determina o menor comprimento entre todas as caixas disponíveis (β), além disso, o primeiro número útil é definido como a coordenada nula do eixo X . No **Passo 2**, a partir da caixa do tipo k e dos números úteis determinados, devem ser obtidos os pontos de vértices ao longo de L_g , adicionando ε comprimentos de caixa ($\varepsilon = 1, 2, \dots$), desde que o tamanho remanescente do contêiner seja suficiente para posicionar uma caixa de menor comprimento (β , obtido no

Passo 1). Os números úteis obtidos são adicionados ao conjunto $X_g^{(k)}$, removendo os elementos repetidos (**Passo 3**). O critério de parada do algoritmo é satisfeito quando $k = 1$, onde o conjunto $X_g^{(1)}$ (X_g) é composto pelos números úteis do comprimento do contêiner.

Para exemplificar a aplicação do AGNU, considere um contêiner com dimensões $(L_g, W_g, H_g) = (20, 15, 15)$, que deve ser carregado com três tipos de caixas: $(l_1, w_1, h_1) = (7, 5, 3)$, $(l_2, w_2, h_2) = (5, 6, 5)$ e $(l_3, w_3, h_3) = (8, 6, 3)$.

Após a execução do algoritmo, os seguintes conjuntos de números úteis são gerados:

$$X_g = \{0; 5; 7; 8; 10; 12; 13; 14; 15\} \quad (15)$$

$$Y_g = \{0; 5; 6; 10\} \quad (16)$$

$$Z_g = \{0; 3; 5; 6; 8; 10; 11; 12\} \quad (17)$$

Em contrapartida, abaixo são apresentados os conjuntos gerados sem o uso do AGNU, isto é, pela definição (1) – (3):

$$X_g = \{0; 1; 2; 3; 4; 5; 6; 7; 8; 9; 10; 11; 12; 13; 14; 15\} \quad (18)$$

$$Y_g = \{0; 1; 2; 3; 4; 5; 6; 7; 8; 9; 10\} \quad (19)$$

$$Z_g = \{0; 1; 2; 3; 4; 5; 6; 7; 8; 9; 10; 11; 12\} \quad (20)$$

Com base na cardinalidade dos conjuntos (15) – (17) e (18) – (20), fica evidente que um modelo gerado com o uso do AGNU, deve apresentar um número consideravelmente menor de restrições e variáveis binárias. No presente exemplo, a primeira formulação seria composta por 603 variáveis binárias e 324 restrições de não sobreposição, enquanto que, sem o uso do AGNU, este mesmo modelo seria formado por 5.452 e 2.288, respectivamente. Desta forma, a utilização do AGNU na geração dos modelos pode levar a significativos ganhos de desempenho do método de solução empregado, visto que permite gerar modelos substancialmente menores.

4. Testes Computacionais

Nesta seção são apresentados os resultados obtidos com os experimentos computacionais realizados. Os modelos foram gerados com base em um algoritmo em linguagem de programação *Microsoft Visual Studio 2012*© (versão 11.0.50727.1) e o *solver CPLEX*© *Interactive Optimizer* (versão 12.6.0.0), com parâmetros *default*, foi utilizado para resolvê-los. Os testes foram efetuados em um computador com sistema operacional *Microsoft Windows 8* (64 bits), processador *Intel*© *Core*TM *i5-4200U CPU @ 1,60HZ* e memória *RAM* de 8GB. Um limite de tempo de 86.400 segundos (24 horas) foi estabelecido para a execução dos testes.

Para avaliar os modelos os seguintes cenários foram considerados:

- a) Cenário 1 (C-1): Modelo definido pelas expressões (8) – (11), gerado sem o uso do AGNU e sem restrição adicional;
- b) Cenário 2 (C-2): Modelo definido pelas expressões (8) – (11), gerado com o uso do AGNU e sem restrição adicional;
- c) Cenário 3 (C-3): Modelo definido pelas expressões (8) – (14), gerado com o uso do AGNU e com restrição adicional de separação de itens.

Os testes foram efetuados com os conjuntos de dados de Mohanty, Mathur e Ivancic (1994) conforme Tabela 1.

Conj. de testes	Caixas						Contêineres				
	i	b_i	l_i	w_i	h_i	c_i	k	d_k	L_k	W_k	H_k
1	1	20	2	6	8	1,0	1	2	10	6	16
	2	50	8	4	10	1,5	2	2	14	14	14
2	1	45	8	16	4	1,4	1	5	20	16	20
	2	60	8	8	8	1,0	2	5	24	12	16
	3	50	8	12	12	1,2	3	5	20	8	16
	4	25	12	4	4	1,0					
3	1	35	10	5	5	1,7	1	1	20	15	25
	2	40	5	15	10	1,0	2	2	15	15	20
	3	28	5	15	15	1,5	3	1	45	20	20
4	1	50	36	28	24	1,4	1	5	60	40	72
	2	60	40	32	20	1,0	2	5	40	36	52
5	1	25	15	20	25	2,5	1	3	35	30	20
	2	30	10	15	10	1,0	2	3	35	35	40
	3	40	20	25	35	2,0	3	2	35	25	40
6	1	28	18	10	8	2,0	1	2	24	20	24
	2	37	12	14	10	1,2	2	5	18	16	20
	3	30	6	10	8	1,6	3	2	30	20	30
7	1	15	7	4	10	1,2	1	1	20	15	29
	2	12	3	5	11	1,0	2	1	17	12	24
	3	20	6	9	12	1,5					
8	1	20	6	8	11	1,2	1	2	17	16	20
	2	17	5	15	9	1,0	2	2	23	18	30
	3	13	4	13	7	1,7	3	1	15	12	30
	4	24	12	13	8	1,3					
	5	21	10	14	11	2,0					
	6	80	8	2	4	1,0					
9	1	18	10	13	11	1,8	1	1	40	22	26
	2	20	8	9	10	1,6	2	1	30	30	30
	3	12	7	12	15	1,5	3	1	35	32	40
	4	22	14	15	11	1,2					
10	1	50	2	4	16	1,2	1	2	24	16	8
	2	70	2	8	16	1,5	2	1	16	16	16
	3	80	8	2	4	1,0					
11	1	30	5	11	14	1,2	1	2	21	18	24
	2	28	9	12	10	1,8	2	1	19	15	26
	3	37	7	13	17	1,3	3	1	23	20	27
	4	23	10	10	8	1,1					
12	1	35	3	6	8	1,1	1	1	30	15	18
	2	40	5	9	4	1,8	2	1	28	25	10
	3	15	8	9	12	1,0	3	1	20	18	10
13	1	56	7	8	3	2,0	1	2	25	21	25
	2	28	12	11	4	1,2	2	2	18	15	9
14	1	25	4	8	6	1,8	1	2	30	24	12
	2	32	10	12	6	1,9	2	2	20	20	10
	3	45	14	14	9	1,3	3	2	18	16	30
15	1	25	6	6	9	2,2	1	2	28	14	18
	2	25	12	10	8	1,7	2	2	15	12	20
	3	15	4	7	5	1,0					
	4	20	7	9	11	1,9					
16	1	35	15	18	17	1,6	1	1	48	70	32
	2	30	11	14	11	2,5	2	1	57	69	44
	3	25	24	26	15	1,3	3	1	38	71	55

Fonte: Mohanty, Mathur e Ivancic (1994)

Tabela 1 - Conjuntos de testes da literatura

Em todos os testes efetuados, a qualidade da solução obtida pode ser considerada: solução ótima, com *gap* nulo; solução não ótima, com *gap* não nulo e insuficiência de memória do computador para resolver o problema e, por fim, sem solução, sem *gap* e com insuficiência de memória do computador para compilar o modelo pelo *solver* CPLEX.

Onde o *gap* de otimalidade é obtido por:

$$gap = \frac{(\text{melhor limitante obtido}) - (\text{melhor valor obtido})}{(\text{melhor limitante obtido})} \cdot 100\% \quad (21)$$

Algumas suposições foram adotadas para a realização dos testes. O termo v_i da função objetivo indica o valor associado à caixa do tipo i ($i = 1, \dots, m$), que pode ser obtido por $v_i = c_i \cdot (l_i \cdot w_i \cdot h_i)$, onde c_i representa o valor por unidade de volume. Para testar o cenário C-3, optou-se por separar as caixas do tipo 1 e 2 em contêineres distintos.

A Tabela 2 apresenta o número de variáveis binárias, restrições de não sobreposição e de separação de itens presentes em cada um dos modelos gerados para teste. A quantidade de restrições de limitação de caixas é equivalente ao número de tipos de itens disponíveis em cada conjunto de teste considerado. Desta tabela, cabe destacar que os modelos gerados sem o uso do AGNU apresentam valores mais elevados. Por comparação, percebe-se que o conjunto de teste 4 gerado de acordo com o cenário C-1 é composto por 137.060 variáveis binárias, enquanto que este mesmo conjunto apresenta apenas 90 variáveis no cenário equivalente, isto é, em C-2.

Conj. De Testes	Cenário C-1		Cenário C-2		Cenário C-3		
	Variáveis Binárias	Restrição Não Sobreposição	Variáveis Binárias	Restrição Não Sobreposição	Variáveis Binárias	Restrição Não Sobreposição	Restrição de Sep. de Itens
1	2.696	2.488	120	90	128	90	12
2	36.945	28.535	1.010	660	1.040	660	45
3	18.857	18.064	369	276	377	276	12
4	137.060	93.550	90	45	110	45	30
5	105.758	82.238	414	335	430	335	24
6	44.793	25.671	444	229	462	229	27
7	12.801	6.345	816	432	820	432	6
8	51.756	29.064	15.176	10.380	15.186	10.380	15
9	115.826	40.756	5.315	2.263	5.321	2.263	9
10	4.635	6.375	544	640	550	640	9
11	23.753	11.092	438	293	446	293	12
12	13.175	9.478	928	664	934	664	9
13	20.776	13.580	730	496	738	496	12
14	24.800	15.386	706	474	718	474	18
15	18.738	8.988	1.758	1.092	1.766	1.092	12
16	445.624	210.220	6.904	4.140	6.910	4.140	9
$\bar{x}$	67.374,56	37.614,38	2.235,13	1.406,81	2.246,00	1.406,81	16,31
δ	108.955,99	53.026,33	3.949,48	2.605,27	3.948,32	2.605,27	10,33

Fonte: O autor

Tabela 2 - Número de variáveis e restrições em cada modelo, onde $\bar{x}$ representa média e σ o desvio padrão

A Tabela 3 apresenta um sumário dos resultados obtidos após a realização dos testes. Nenhuma informação foi obtida para o conjunto de teste 16 do cenário C-1, visto que o modelo excedeu a memória do computador logo na leitura dos dados pelo *solver* CPLEX. Esta situação encontra-se identificada por “*” na tabela.

Uma breve análise mostra que dos 48 testes efetuados, sendo 16 exemplos para cada cenário considerado, apenas em dois casos a solução ótima (*gap* nulo) não foi alcançada dentro do limite de tempo especificado. Estas situações indicam um *gap* de 0,8% e ocorrem no conjunto de teste 9 dos cenários C-1 e C-2. Embora os modelos do cenário C-3 indiquem complexidade mais acentuada de resolução, visto que são compostos pelas mesmas restrições dos cenários C-1 e C-2 com o acréscimo da restrição de separação de itens, em todos estes exemplos a solução ótima foi alcançada.

Conj. de testes	Cenário C-1			Cenário C-2			Cenário C-3		
	Z	gap	Tempo (seg.)	Z	gap	Tempo (seg.)	Z	gap	Tempo (seg.)
1	5.376	-	0,48	5.376	-	0,19	4.800	-	0,08
2	70.438,4	-	239,06	70.438,4	-	2,09	70.080	-	2,48
3	49.512,5	-	406,83	49.512,5	-	0,16	49.512,5	-	0,16
4	846.720	-	6.601,66	846.720	-	0,05	846.720	-	0,11
5	313.750	-	9.693,23	313.750	-	0,08	313.750	-	0,09
6	121.824	-	499,47	121.824	-	0,08	115.776	-	0,14
7	12.522	-	32,30	12.522	-	0,13	12.396	-	0,16
8	53.009,2	-	1.765,2	53.009,2	-	111,55	52.662,8	-	316,33
9	112.338	0,8%	81.167	112.338	0,8%	11.884	107.532	-	1.852,42
10	11.264	-	0,56	11.264	-	0,03	11.264	-	0,13
11	36.870,60	-	190,56	36.870,6	-	0,06	36.870,6	-	0,22
12	21.096	-	4,13	21.096	-	0,73	18.504	-	1,97
13	34.656	-	22,88	34.656	-	2,80	27.014,4	-	0,42
14	61.588,8	-	143,08	61.588,8	-	0,11	59.472	-	1,56
15	31.002	-	27,33	31.002	-	2,55	30.862	-	2,48
16	*	*	*	547.098	-	77,73	497.994	-	45,88
$\bar{x}$	118.797,8		6.719,5	145.566,5		755,1	140.950,6		139,04
δ	208.076,3		20.086,4	226.579,4		2.873,6	221.898,9		448,9

Fonte: O autor

Tabela 3 - Resultados obtidos com os testes efetuados, onde $\bar{x}$ representa média e σ o desvio padrão

Quanto ao tempo necessário para a resolução dos modelos temos que, no cenário C-1 o tempo médio decorrido foi de 6.719,50 segundos, com desvio padrão de 20.086,40 segundos. Já para os cenários C-2 e C-3, o tempo médio de resolução foi de 755,10 segundos (com desvio padrão de 2.873,60) e 139,04 segundos (com desvio padrão de 448,90), respectivamente. Destes valores cabe destacar que a restrição de separação de itens não impactou no tempo computacional de resolução, dado que o cenário C-3 obteve a menor média de tempo de resolução e, mesmo assim, a solução ótima foi encontrada em todos os modelos avaliados.

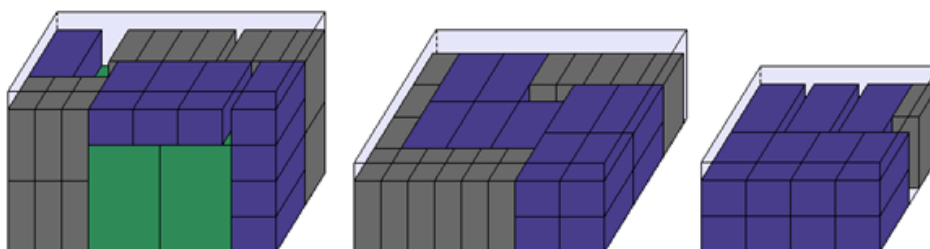


Figura 3 - Padrão de carga do conjunto de teste 12 gerado de acordo com o cenário C-2

Outro ponto em destaque cabe ao uso do AGNU na geração dos modelos. Com base na Tabela 3, pode-se perceber acentuada diferença no tempo de resolução dos exemplos. Tomando os sete primeiros conjuntos de testes, percebe-se que nenhum modelo dos cenários C-2 e C-3 ultrapassa os três segundos de execução, enquanto que em C-1, apenas um exemplo

foi resolvido nesta grandeza de tempo. Com destaque especial para o exemplo 5 deste cenário, que consumiu acima de 9.600 segundos de execução.

As Figuras 3 e 4 ilustram, respectivamente, o padrão de carga obtido para o conjunto de teste 12 gerado de acordo com os cenários C-2 e C-3. As caixas foram coloridas do seguinte modo: tipo 1 - cinza; tipo 2 - azul e tipo 3 - verde. Cabe lembrar que, para o cenário C-3, as caixas do tipo 1 e 2 devem ficar obrigatoriamente separadas em contêineres distintos.

O padrão de carga apresentado na Figura 3 foi obtido com alocação de 35 caixas do tipo 1, 40 caixas do tipo 2 e 03 caixas do tipo 3, totalizando em $Z = 21.096$. O primeiro contêiner da esquerda teve ocupação de 85,33% do seu volume disponível, o segundo com 75,09% e o terceiro com 74%, respectivamente. Já o padrão da Figura 4 é composto por 35 caixas do tipo 1, 40 caixas do tipo 2 e nenhum item do tipo 3, totalizando em $Z = 18.504$. Os contêineres obtiveram ocupação de 53,33%, 72% e 80%, respectivamente. Estes valores só reforçam o esperado, pois em modelos que consideram a restrição de separação de itens, em geral, devido ao acréscimo de exigência prática, tendem a obter um valor inferior de solução.

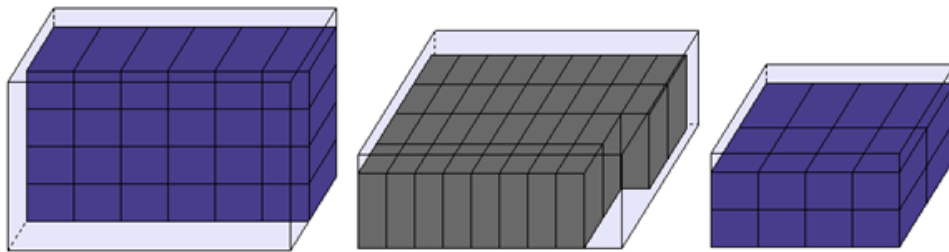


Figura 4 - Padrão de carga do conjunto de teste 12 gerado de acordo com o cenário C-3

5. Considerações Finais

Neste trabalho foi apresentada uma proposta de formulação matemática para o problema de carregamento de contêiner, em especial, carregamento de múltiplos contêineres heterogêneos. Esta abordagem foi testada por meio do *solver* CPLEX, e embora limitada a resolver problemas de complexidade moderada, representou satisfatoriamente às condições tratadas.

A formulação matemática proposta, oriunda de uma adaptação do modelo apresentado por Junqueira, Morabito e Yamashita (2012), demonstra apresentar flexibilidade para a consideração de restrições práticas listadas por Bischoff e Ratcliff (1995). Apesar da modelagem apresentada neste trabalho abordar unicamente a restrição de separação de itens, podem-se citar os trabalhos de Kurpel, Schenekemberg e Scarpin (2015), Junqueira, Morabito e Yamashita (2012) e trabalhos correlatos dos autores, como fonte de outras exigências práticas que podem ser diretamente acrescentadas ao modelo proposto.

Como indicado nos testes efetuados, o Algoritmo de Geração dos Números Úteis colaborou significativamente com a redução da cardinalidade dos conjuntos dos possíveis pontos de vértices dos itens, de forma a impactar diretamente na dimensão dos modelos avaliados. Este fator, como esperado, produziu significativos ganhos em termos de desempenho de resolução.

Os resultados computacionais obtidos mostram que a abordagem proposta é capaz de resolver, em tempo computacional aceitável, apenas problemas de complexidade moderada. No entanto, esta formulação pode servir de motivação para futuras pesquisas que explorem outros métodos de solução, como procedimentos heurísticos, metaheurísticos, métodos de relaxação, entre outros.

Este trabalho pode ser utilizado, futuramente, como motivação para as seguintes pesquisas: explorar formas alternativas aos conjuntos (1) – (6), de modo a diminuir a cardinalidade e

consequentemente a complexidade dos modelos; estender a formulação proposta para considerar os mais variados tipos de CLP's; expandir a modelagem para abordar restrições práticas não tratadas neste trabalho, como restrição adicional de manuseio, prioridades de envio e distribuição de peso dentro do contêiner e, por fim, testar combinações distintas de parâmetros para o *solver CPLEX*, visto que neste trabalho optou-se por parâmetros *default*.

Referências

- ALTARAZI, S.** *A New Prioritizing-Stacking Heuristic Algorithm for the Inner-City Truck Loading Problem.* International Journal of Business and Management. Vol. 8, n. 3, p. 1-7, 2013.
- BISCHOFF, E.E.; RATCLIFF, M.S.W.** *Issues in the Development of Approaches to Container Loading.* Omega. Vol. 23, n. 4, p.377-390, 1995.
- BORTFELDT, A.; WÄSCHER, G.** *Constraints in Container Loading – A State-of-the-Review.* European Journal of Operational Research. Vol. 229, n. 1, p.1-20, 2013.
- CARNIERI, C.; MENDOZA, G.A.; GAVINHO, L.G.** *Solution Procedures for Cutting Lumber into Furniture Parts.* European of Operations Research. Vol. 73, p.495-501, 1994.
- CHEN, C.S.; LEE, S.M.; SHEN, Q.S.** *An analytical model for the Container Loading Problem.* European Journal of Operational Research. Vol. 80, n.1, p.68-76, 1995.
- CHRISTOFIDES, N.; WHITLOCK, C.** *An Algorithm for Two-Dimensional Cutting Problems.* Operations Research. Vol. 25, n.1, p. 30-44, 1977.
- ELEY, M.** *A Bottleneck Assignment Approach to the Multiple Container Loading Problem.* OR Spectrum. Vol. 25, n.1, p.45-60, 2003.
- GEHRING, H.; BORTFELDT, A.** *A Genetic Algorithm for Solving the Container Loading Problem.* International Transactions in Operational Research. Vol. 4, n.5-6, p.401-418, 1997.
- HERZ, J.C.** *Recursive Computational Procedure for Two-Dimensional Stock Cutting.* IBM Journal of Research and Development. Vol. 16, n. 5, p. 462-469, 1972.
- HIFI, M.; KACEM, I.; NEGRE, S.; WU, L.** *A Linear Programming Approach for the Three-Dimensional Bin-Packing Problem.* Electronic Notes in Discrete Mathematics. Vol. 33, p. 993-1000, 2010.
- JUNQUEIRA, L.; MORABITO, R.; YAMASHITA, D.S.** *Three-Dimensional Container Loading Models with Cargo Stability and Load Bearing Constraints.* Computer & Operations Research. Vol. 39, n.1, p.74-85, 2012.
- KURPEL, V.K.; SCHENEKEMBERG, C.M.; SCARPIN, C.T.** *Um Modelo Matemático Exato para o Problema de Carregamento de Contêineres com Restrição de Carregamento Completo de Grupo de Itens.* Revista Eletrônica Gestão e Saúde. Vol.6 (Supl.2), p.1014-1026, 2015.
- MARTELLO, S.; PISINGER, D.; VIGO, D.** *The Three-Dimensional Bin Packing Problem.* Operations Research. Vol. 48, n.2, p.256-267, 2000.
- MIZAWA, F.K.; WAKABAYASHI, Y.** *An Algorithm for the Three-Dimensional Packing Problem with Asymptotic Performance Analysis.* Algorithmica. Vol. 18, p. 122-144, 1997.
- MOHANTY, B.B.; MATHUR, K.; IVANCIC, N.J.** *Value Considerations in Three-Dimensional Packing – A Heuristic Procedure using the Fractional Knapsack Problem.* European Journal of Operational Research. Vol. 74, n.1, p.143-151, 1994.
- OLIVEIRA, L.K.; MORABITO, R.** *Métodos Exatos baseados em Relaxações Lagrangeana e Surrogate para o Problema de Carregamento de Paletes do Produtor.* Pesquisa Operacional. Vol. 26, n.2, p.403-432, 2006.
- PISINGER, D.** *Heuristics for the Container Loading Problem.* European Journal of Operational Research. Vol. 141, n. 2, p.382-392, 2002.
- REN, J.; TIAN, Y.; SAWARAGI, T.** *A Tree Search Method for the Container Loading Problem with Shipment Priority.* European Journal of Operational Research. Vol. 214, n. 3, p.526-535, 2011.
- TSAI, R.D.; MALSTROM, E.L.; KUO, W.** *Three Dimensional Palletization of Mixed Box Sizes.* IIE Transactions. Vol.25, n.4, p.64-75, 1993.
- WÄSCHER, G.; HAUßNER, H.; SCHUMANN, H.** *An Improved Typology of Cutting and Packing Problems.* European Journal of Operational Research. Vol 183, n. 3, p. 1109-1130, 2007.