

CONTRIBUIÇÕES PARA UM PROBLEMA DE CARREGAMENTO DE CONTÊNER COM MÚLTIPLOS COMPARTIMENTOS

Rodolfo Ranck Jr

Programa de Doutorado em Computação Aplicada – CAP, INPE
Av. dos Astronautas, 1758 – CEP 12227-010 – São José dos Campos, SP - Brasil
rodolforanck@gmail.com

Horacio Hideki Yanasse

Instituto de Ciência e Tecnologia - ICT, UNIFESP
Rua Talim, 330, CEP 122231-280 - São José dos Campos, SP - Brasil
horacio.yanasse@unifesp.br

Reinaldo Morabito

Departamento de Engenharia de Produção - DEP, UFSCAR
Rod. Washington Luís, KM 235 – CEP 13565-905 - São Carlos, São Paulo - Brasil
morabito@ufscar.br

RESUMO

Neste trabalho aborda-se um problema em que caixas de diferentes tipos têm de ser carregadas nos compartimentos de um contêiner para serem entregues em uma rota conhecida. Essas caixas podem ser empacotadas sem um padrão específico e/ou em camadas horizontais separadas por divisórias, e deve-se atender a quesitos práticos como orientação, estabilidade, empilhamento, distribuição de peso, e múltiplos destinos. Busca-se um empacotamento viável que reduza o custo com o descarregamento de itens ao longo da rota. Para resolver este problema, apresenta-se uma heurística baseada na geração de camadas horizontais auxiliada por um modelo matemático.

PALAVRAS CHAVE. Problema de Carregamento de Contêiner, Problema de Empacotamento, Contêineres Compartimentados.

Área principal: OC - Otimização Combinatória.

ABSTRACT

In this paper we focus on a problem where boxes of several types have to be loaded in the container compartments to be delivered on a known route. These boxes can be packed without a specific pattern and/or on horizontal layers separated by divisions, and practical requisites as orientation, stability, stacking, weight distribution, and multiple destinations have to be met. We seek a packing that reduces the unloading costs along the route. To solve this problem, we present a heuristic based on the generation of horizontal layers with a support of a mathematical model.

KEYWORDS. Container Loading Problem. Packing Problem. Compartmented Containers.

Main area: CO - Combinatorial Optimization.

1. Introdução

O planejamento e o gerenciamento do transporte de carga, bem como das atividades subjacentes de armazenagem e manuseio, são tarefas comuns em diversas empresas que precisam atender clientes fornecendo e/ou coletando produtos. Por estarem intimamente relacionadas à lucratividade e à satisfação desses clientes, busca-se realizá-las da melhor maneira: produtos devem ser entregues dentro do prazo, íntegros e, se possível, com os menores custos.

O transporte de produtos pode ser feito por vários modais. Geralmente, para cada um deles existe um veículo especializado com um ou mais contêineres que acondicionam a carga e facilitam sua movimentação com segurança e rapidez.

Empacotar unidades menores (itens) em unidades maiores (objetos) é um problema comum quando se deseja transportar carga e é frequentemente árduo quando se requer eficiência. Análogo a este problema, pode-se definir outro, matematicamente equivalente, em que os itens são cortados de objetos, como de chapas, ou tiras. Estes problemas são conhecidos na literatura como Problemas de Corte e Empacotamento (PCE).

Os itens podem ser dispostos em um objeto de diversos modos. Cada um deles determina um padrão de empacotamento/corte que deve atender às seguintes restrições: (a) os itens não podem ocupar o mesmo ponto no objeto; (b) os itens devem estar completamente dentro do objeto.

Em PCEs reais práticos a quantidade de padrões distintos costuma ser muito alta, o que dificulta a solução destes problemas de maneira exata. A quantidade desses padrões depende das dimensões, quantidades e variedades dos itens/objetos, além de outras restrições impostas, por exemplo, por equipamentos usados em operações de corte/empacotamento.

Os PCEs encontrados na literatura diferem entre si segundo tipos de restrições e objetivos. Uma classificação para esses problemas pode ser encontrada em Wäscher et al. (2007).

Um problema de empacotamento com relevância prática é o de carregar itens em contêineres. A literatura que trata deste problema considera geralmente que itens retangulares e tridimensionais devem ser empacotados ortogonalmente em objetos também retangulares e tridimensionais. Dois tipos deste problema são considerados: Problema de Carregamento de Contêineres (PCC) e Problema de Carregamento de Paletes (PCP). O PCC e o PCP são semelhantes e uma possível diferença entre eles é que no PCC os itens podem ser apoiados nas paredes verticais dos contêineres. Outra possível diferença é que o PCP normalmente utiliza arranjos de itens em camadas horizontais, o que pode simplificar o problema em outros de menores dimensões (Alvarez-Valdes et al., 2005).

O arranjo dos itens em camadas facilita o carregamento e descarregamento da carga na prática, pois diminui a complexidade do padrão de empacotamento. Em alguns casos, ao empacotar camadas horizontais, é possível separá-las com chapas finas de madeira ou plástico (divisórias) a fim de aumentar a estabilidade do carregamento e/ou separar a carga. Em alguns ambientes logísticos o empacotamento dos itens em camadas horizontais é uma norma.

No melhor do nosso conhecimento, trabalhos pioneiros da literatura que abordam problemas de carregar itens em contêineres são dos anos 70 e 80, dos quais podemos destacar: De Sha (1970); Peleg e Peleg (1976); Steuder (1979); e George e Robinson (1980). Segundo Bortfeldt e Wäscher (2012), a quantidade de trabalhos que trata destes problemas não é muita, mas ela vem crescendo gradativamente nos últimos anos.

O PCC e o PCP podem exigir a consideração de outros aspectos não contemplados em PCEs mais simples, como estabilidade, empilhamento da carga, e distribuição de peso no veículo. Uma discussão de vários desses aspectos pode ser encontrada em Bortfeldt e Wäscher (2012).

Os PCCs englobam os PCEs unidimensionais e, portanto, são NP-árduos. Geralmente, apenas instâncias pequenas destes problemas podem ser resolvidas na prática por algoritmos exatos e, por este motivo, vários métodos de solução propostos na literatura para resolver exemplares maiores são heurísticas. Algumas heurísticas para esses problemas são baseadas na estratégia de dividir o objeto em partes menores e resolver, para cada uma delas, um problema menor. Pisinger (2002) classificou essas heurísticas em 4 tipos básicos, cada um deles está relacionado a uma maneira de empacotar os itens no objeto: a) Paredes Virtuais. Os itens são

arranjados em camadas horizontais (Morabito e Arenales, 1997; Lim e Zhang, 2005) ou verticais (George e Robinson, 1980); b) Pilhas de Itens. Os itens são arranjados em colunas verticais (Haessler e Talbot, 1990); c) Cortes Guilhotinados. Os itens são arranjados de maneira que poderiam ser separados íntegros deste arranjo por meio de cortes ortogonais (Morabito e Arenales, 1994); d) Cuboides. Os itens são arranjados em blocos (Eley, 2002).

Para obter uma revisão recente da literatura do PCC e PCP, veja, por exemplo, Bortfeldt e Wäscher (2012).

Neste trabalho investigamos um PCC motivado de uma indústria de bebidas que precisa transportar diariamente seus produtos a clientes. Um tipo de veículo, largamente utilizado pela empresa, é um caminhão que possui, em sua carroçaria, um contêiner multicompartimentado. Este tipo de contêiner é útil, por exemplo, quando se deseja aumentar a estabilidade da carga e/ou separá-la.

O caminhão deve cumprir uma rota conhecida e todas as caixas demandadas pelos clientes dessa rota devem ser empacotadas nele. Ao empacotar a carga, devem-se atender diversas restrições práticas como orientação, estabilidade e resistência das caixas ao empilhamento realizado. Deseja-se encontrar um empacotamento viável que penalize o desbalanceamento do veículo e o remanejamento de caixas ao longo da rota.

2. Definição do Problema

Um veículo dispõe de vários compartimentos destinados ao armazenamento da carga, que podem ser acessados de maneira independente do seu lado externo. Esses compartimentos possuem a mesma largura e comprimento e são dispostos em fileiras justapostas. Há um número variável de fileiras de compartimentos (depende do comprimento do veículo) no lado maior do veículo (eixo y) e duas fileiras de compartimentos definidas no lado menor do veículo (eixo x).

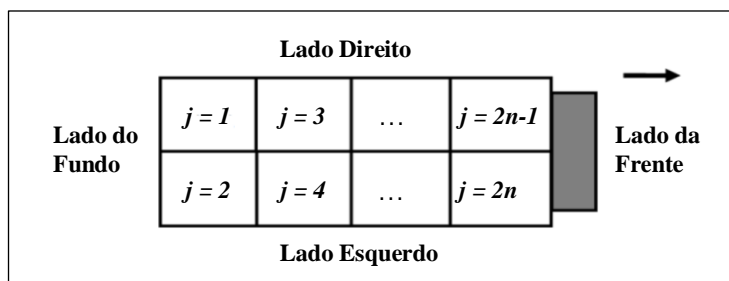


Figura 1. Visão superior e em corte do veículo com múltiplos compartimentos.

Admite-se que todo compartimento e todos os itens são hexaedros retangulares (cuboides). Além disso, os itens são rígidos, têm dimensões variadas e devem ser colocados ortogonalmente dentro dos compartimentos.

Os itens podem ser arranjados sem obedecer a um padrão específico e/ou em camadas horizontais. Admite-se que uma camada seja formada por itens de uma mesma família de produtos (mesmas dimensões, peso e resistência a empilhamento) e não contenham itens empilhados. Em uma camada os itens já possuem um padrão de montagem (layout) ótimo. As camadas são separadas de outras camadas ou itens por uma divisória rígida com comprimento e largura de um compartimento.

O veículo parte carregado de uma célula logística e sua rota é definida por uma sequência de paradas que ele deve realizar para atender todos os clientes. A rota e as demandas de todos os clientes são conhecidas.

Ao empacotar os itens, devem ser atendidas as restrições (a)-(d) e procura-se atender ao quesito (e):

- Atendimento completo da demanda.** Todos os itens demandados devem ser empacotados no veículo;
- Orientação dos itens.** Os itens devem ser empacotados com orientação vertical fixa;
- Estabilidade dos itens.** Os Itens devem ser estabilizados por outros itens e/ou pelas paredes presentes no contêiner. Uma camada oferece suporte necessário para os itens

- colocados acima, contanto que haja uma quantidade mínima de itens nesta camada;
- d) **Empilhamento dos itens.** Em qualquer empilhamento, o item abaixo deve ter resistência suficiente para suportar o peso dos itens colocados acima dele. O peso sobre uma camada é distribuído homogeneamente aos itens desta camada;
- e) **Distribuição de peso no contêiner.** Deseja-se *penalizar* o desbalanceamento do veículo ao longo da rota, no eixo x , quando este é maior que um valor de tolerância conhecido.

Nenhuma restrição deve ser violada ao longo da rota e o objetivo é *penalizar* o remanejamento de itens ao longo da rota. Esses custos incidem quando itens f' , a serem descarregados em uma parada k' , estão sobre itens f , de uma parada k , com $k' > k$, ou seja, para descarregar os itens f , os itens f' precisam ser remanejados na parada k incorrendo em custos adicionais. Admite-se que esses itens f' sejam recolocados em suas posições definidas, imediatamente após descarregar os itens f na parada k ; e que um item possa ser sempre alcançado diretamente, desde que outros não bloqueiem seu acesso pelo eixo z .

3. Uma heurística para o problema

Propomos uma heurística que simplifica o problema descrito em outros que, por sua vez, são resolvidos por métodos exatos e também por outras heurísticas.

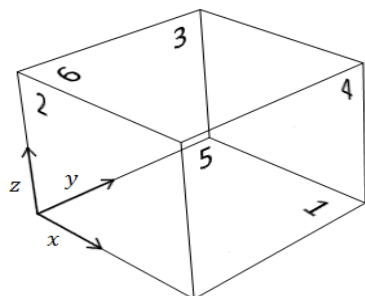
Inicialmente, busca-se gerar camadas que contenham apenas itens de um mesmo tipo e na maior quantidade possível. Estas camadas são denominadas de *completas* e já possuem um padrão de empacotamento definido. Os itens *residuais*, que não podem formar camadas completas, não obedecem a um padrão específico e, por questões de estabilidade, não devem ser colocados abaixo de alguma camada completa. Denominamos um arranjo de itens residuais empacotados em um mesmo objeto de *camada incompleta*.

Gera-se o maior número possível de camadas completas atribuindo os itens a elas em ordem inversa das paradas do veículo. Desta maneira, os itens residuais, que ficarão no topo em algum compartimento, tendem a ser aqueles demandados pelos clientes das paradas iniciais. Após gerar as camadas completas, as camadas incompletas são geradas com um procedimento construtivo que procura obedecer às restrições e quesitos desejáveis do problema. Em seguida, tenta-se empacotar todas as camadas nos compartimentos resolvendo um problema de programação linear inteira. Caso uma solução viável não possa ser obtida, tenta-se aplicar um procedimento recursivo, novamente em busca de uma solução viável; caso contrário, tenta-se melhorá-la buscando um novo empacotamento para os itens residuais em um procedimento iterativo de atribuição e empacotamento.

A heurística proposta é apresentada nas Tabelas 1, 2, 3 e 4. A Tabela 1 descreve o procedimento principal; a Tabela 2 descreve o procedimento para gerar as camadas completas; a Tabela 3 descreve o procedimento para gerar as camadas incompletas; a Tabela 4 descreve o procedimento para tentar melhorar a solução obtida.

Algumas definições e notações utilizadas.

Para indicar a posição dos itens residuais em um objeto, utiliza-se como referência o ponto que dá origem aos eixos x , y e z , adjacente às faces 1, 2 e 3 do cuboide da Figura 2. Este ponto é denominado de “Canto Frontal Inferior Esquerdo” (*Front Lower Left Corner - FLLC*).



Face	Nomenclatura
1	Face inferior
2	Face lateral da frente
3	Face lateral esquerda
4	Face lateral do fundo
5	Face lateral direita
6	Face superior

Figura 2. Cuboide de referência para itens e objetos.

Os domínios das coordenadas p , q e r no objeto c (respectivamente, para os eixos x , y e z) candidatas para um item residual do tipo i ser colocado com seu FLLC, quando os itens podem ser rotacionados em 90° no eixo horizontal, são dados respectivamente por \hat{X}_i , \hat{Y}_i e \hat{Z}_{ic} . Considera-se apenas a geração de padrões normais (veja sobre estes padrões em Junqueira et al., 2011):

$$\begin{aligned} \hat{X}_i &= \left\{ p \mid p = \sum_{i' \in M} \lambda_{i'}^l l_{i'} + \lambda_{i'}^w w_{i'}, 0 \leq p \leq L - \text{Min}\{l_i, w_i\}, \right. \\ &\quad \left. 0 \leq (\lambda_{i''}^l + \lambda_{i''}^w) \leq b_{i''} \mid \lambda_{i''}^l, \lambda_{i''}^w \in \mathbb{N}, i'' \in M \right\}, i \in M \\ \hat{Y}_i &= \left\{ q \mid q = \sum_{i' \in M} \lambda_{i'}^l l_{i'} + \lambda_{i'}^w w_{i'}, 0 \leq q \leq W - \text{Min}\{l_i, w_i\}, \right. \\ &\quad \left. 0 \leq (\lambda_{i''}^l + \lambda_{i''}^w) \leq b_{i''} \mid \lambda_{i''}^l, \lambda_{i''}^w \in \mathbb{N}, i'' \in M \right\}, i \in M \\ \hat{Z}_{ic} &= \left\{ r \mid r = \sum_{i' \in M} \lambda_{i'}^h h_{i'}, 0 \leq r \leq H_c - h_i, \right. \\ &\quad \left. 0 \leq \lambda_{i''}^h \leq b_{i''} \mid \lambda_{i''}^h \in \mathbb{N}, i'' \in M \right\}, i \in M \end{aligned} \quad (1)$$

em que:

b_i : quantidade disponível do item de tipo i ;

l_i, w_i e h_i : respectivamente, comprimento (eixo x), largura (eixo y) e altura (eixo z) do item de tipo i ;

L e W : respectivamente, comprimento e largura do compartimento c ;

H_c : altura disponível no compartimento c ;

M : conjunto dos tipos de itens demandados.

Tabela 1. Procedimento principal da heurística.

1.	Início main()
2.	Gere a maior quantidade de camadas completas com $gComp(L^{it})$;
3.	Repita
4.	Gere as camadas incompletas com $gInc(L^r, M', 0)$;
5.	Se algum item residual não puder ser empacotado;
6.	Fim main() ;
7.	Tente empacotar todas as camadas resolvendo a formulação (2)-(12);
8.	Se não for possível obter uma solução viável para (2)-(12)
9.	Se N' for vazio
10.	Fim main() ;
11.	Atribua ao conjunto M' os itens da camada completa j de menor R_j e exclua j de N' ;
12.	Exclua as camadas incompletas de N'' ;
13.	Enquanto uma solução viável para (2)-(12) não puder ser obtida;
14.	Armazene a solução obtida;
15.	Exclua as camadas incompletas de N'' e atualize H_c , $c \in C$;
16.	Tente gerar novamente as camadas incompletas nos compartimentos com $pRes(L^r)$;
17.	Avalie o custo da solução obtida pelo procedimento $pRes(L^r)$, se houver, com (2)-(12);
18.	Retorne a melhor solução obtida;
19.	Fim main()

em que:

C : conjunto dos compartimentos do veículo;

L^{it} : lista contendo os itens demandados do tipo $i \in M$ em ordem inversa das paradas do veículo;

L^r : lista contendo somente os itens residuais em ordem inversa das paradas do veículo. Caso haja empate, os itens de mesma parada são ordenados de maneira decrescente por altura. Caso ainda haja empate, os itens de mesma parada e altura são ordenados de maneira não crescente por peso.

M' : conjunto dos itens residuais;

N' : conjunto das camadas completas;

N'' : conjunto das camadas incompletas;

R_j : maior valor de peso que pode ser colocado sobre uma camada completa j .

Tabela 2. Procedimento para gerar as camadas completas.

20.	Início $gComp(L^{it})$
21.	$j = 1;$
22.	Para $i = 0, \dots, M - 1$
23.	Para $f = 0, \dots, M_i - 1$
24.	Se houver espaço disponível para o item $L^{it}[i, f]$ na camada j
25.	Atribua o item $L^{it}[i, f]$ à camada $j;$
26.	Caso contrário
27.	$j = j + 1;$ Atribua o item $L^{it}[i, f]$ à camada $j;$
28.	Fim Para
29.	$j = j + 1;$
30.	Fim Para
31.	Exclua as camadas que não são completas;
32.	Fim $gComp(L^{it})$

em que, M_i : conjunto dos itens demandados do tipo i .

Tabela 3. Procedimento para empacotar os itens residuais.

33.	Início $gInc(\bar{L}^r, \bar{M}^r, aux)$
34.	$c = 1; nC' = C ;$
35.	Se $aux \neq 0$
36.	$c = aux; nC' = -1;$
37.	Exclua os elementos de \bar{L}^r que não estão em $\bar{M}^r;$
38.	Repita
39.	Para $count = 0, \dots, \bar{L}^r - 1$
40.	$f = \bar{L}^r[count]; i = \varphi(f);$
41.	Para $rT = 0$ até 1
42.	Para $r = 0, \dots, nH_c r \in \hat{Z}_{ic}$
43.	Para $q = 0, \dots, nW q \in \hat{Y}_i$
44.	Para $p = 0, \dots, nL p \in \hat{X}_i$
45.	Se o ponto (p, q, r) do objeto c não estiver ocupado e for adjacente às faces de um item ou deste objeto c
46.	Rotacione o item f em $(rt \cdot 90)$ graus no plano horizontal;
47.	Verifique se o item f pode ser colocado no ponto (p, q, r) do objeto c com $vOver(f, p, q, r, c), vStack(f, p, q, r, c)$ e $vStab(f, p, q, r, c);$
48.	Se o item f puder ser colocado neste ponto
49.	Empacote o item f com sua orientação definida, no ponto (p, q, r) do objeto c , e vá para a linha 53 (próximo item);
50.	Fim Para
51.	Fim Para
52.	Fim Para
53.	Fim Para
54.	Fim Para
55.	$c = c + 1;$ Exclua de \bar{L}^r todos os itens residuais que puderam ser empacotados;
56.	Enquanto houver itens não empacotados em \bar{L}^r e $c \leq nC';$
57.	Fim $gInc(\bar{L}^r, \bar{M}^r, aux)$

em que:

$\varphi(f)$: tipo do item f ;

nL, nW e nH_c : respectivamente, o maior valor nos conjuntos \hat{X}_i, \hat{Y}_i e \hat{Z}_{ic} ;

$vOver(f, p, q, r, c)$: procedimento que verifica se o item f colocado no ponto (p, q, r) do objeto c sobrepõe outro item f' já empacotado;

$vStack(f, p, q, r, c)$: procedimento que verifica se, ao colocar o item f no ponto (p, q, r) do objeto c , a pressão máxima suportada por qualquer ponto (s, t, u) da face superior de itens f' já empacotados não é excedida pela pressão exercida pelos itens colocados acima de (s, t, u) ;

$vStab(f, p, q, r, c)$: procedimento que verifica se, no objeto c : o total de pontos da face inferior do item f em contato com a face superior de itens f' de paradas $k' \geq k$, ou com a face inferior do

objeto, é maior do que $\alpha(l_i w_i)$; o total de pontos da face lateral esquerda do item f em contato com a face lateral direita de itens f' de paradas $k' \geq k$, ou com a face lateral esquerda do objeto, é maior do que $\beta(h_i w_i)$; o total de pontos da face lateral da frente do item f em contato com a face lateral do fundo de itens f' de paradas $k' \geq k$, ou com a face lateral da frente do objeto, é maior do que $\gamma(l_i h_i)$. Em que, $i = \varphi(f)$; k é parada do item f ; α, β e γ são parâmetros.

Tabela 4. Procedimento para tentar empacotar novamente os itens residuais.

58.	Início $pRes(\bar{L}^r)$
59.	Enquanto houver itens em \bar{L}^r
60.	Atribua os itens de \bar{L}^r aos compartimentos resolvendo (13)-(21);
61.	Se uma solução viável para (13)-(21) não puder ser obtida
62.	Fim $pRes(\bar{L}^r)$;
63.	Para $c = 1, \dots, C $
64.	Tente empacotar os itens de M'_c no compartimento c com $gInc(\bar{L}^r, M'_c, c)$;
65.	Exclua de \bar{L}^r os itens residuais que puderam ser empacotados em c ;
66.	Fixe $x_{fc} = 1$, se o item residual $f \in M'$ foi empacotado em c ;
67.	Fixe $x_{fc} = 0, f \in M_i$, se um item residual do tipo $i \in M$ não pôde ser empacotado em c ;
68.	Fim Para
69.	Fim Enquanto
70.	Fim $pRes(\bar{L}^r)$

em que, M'_c : conjunto dos itens residuais atribuídos ao compartimento c e que ainda não foram empacotados.

Formulações matemáticas.

$$\text{Min } \rho_1 = \omega^{dev} \sum_{k \in S} dev_k^x + \omega^z \sum_{k \in S/\{S\}} \sum_{j \in N} z_{jk} (\omega^q Q_{jk+1} + \omega^p P_{jk+1}) \quad (2)$$

$$\sum_{c \in C} \sum_{r \in \bar{Z}_{jc}} y_{jrc} = 1, \quad j \in N \quad (3)$$

$$\sum_{j \in N} \sum_{r \in \bar{Z}_{jc}} \bar{h}_j y_{jrc} \leq H_c, \quad c \in C \quad (4)$$

$$\sum_{j \in N} y_{jrc} \leq 1, \quad c \in C; r \in \bar{Z}_c \quad (5)$$

$$cG_k^{l,x} = \left(\sum_{j \in N} \sum_{c \in C} \sum_{r \in \bar{Z}_{jc}} y_{jrc} P_{jk} d_{0,p^c + r}^{x_{jk}} \right) / P^k, \quad k \in S \quad (6)$$

$$\begin{aligned} dev_k^x &\geq P^k (cG_k^{l,x} - cG^{v,x}) - \varepsilon^x \\ &\geq P^k (cG^{v,x} - cG_k^{l,x}) - \varepsilon^x, \end{aligned} \quad k \in S \quad (7)$$

$$y_{j',r+1,c} \leq \sum_{j \in N'} y_{jrc} stab_{jj'}, \quad j' \in N'; c \in C; r \in \bar{Z}_{j',c}; r < ubL_c \quad (8)$$

$$y_{j',r',c} \leq \sum_{j \in N'} y_{jrc} stab_{jj'} + (1 - \sum_{j \in N'} y_{jrc}), \quad j' \in N''; c \in C; r' \in \bar{Z}_{j',c}; r \in \bar{Z}_c, r < r' \quad (9)$$

$$\mathcal{M}(1 - y_{j',rc}) + y_{j',rc} \sigma_j \geq \sum_{j \in N} \sum_{\substack{r' \in \bar{Z}_{jc} \\ r' > r}} (P_{jk} y_{jr',c}) / \pi_{j',k}, \quad k \in S; c \in C; j' \in N' \\ r \in \bar{Z}_c, r \leq ubL_c; \pi_{j',k} \neq 0 \quad (10)$$

$$y_{j',r',c} a_{j',k'} - z_{j',k} \leq 1 - \sum_{j \in N'} y_{jrc} a_{jk}, \quad j' \in N; c \in C; r' \in \bar{Z}_{j',c}/\{1\}; k' \in S/\{1\} \\ r \in \bar{Z}_c, r < r'; k \in S, k < k' \quad (11)$$

$$\begin{aligned} dev_k^x &\geq 0, k \in S \\ y_{jrc} &\in \{0,1\}, j \in N; c \in C; r \in \bar{Z}_{jc} \\ z_{jk} &\in \{0,1\}, j \in N; k \in S/\{S\} \end{aligned} \quad (12)$$

As variáveis deste modelo são: $cG_k^{l,x}$: centro de gravidade da carga imediatamente antes da parada k (quando há somente itens de paradas $k' \geq k$ no veículo); dev_k^x : desbalanceamento, no eixo x , imediatamente antes da parada k incluindo um parâmetro de tolerância ε^x ; y_{jrc} : é igual a 1 se a camada j é colocada na posição r do compartimento c e, igual a 0, caso contrário; z_{jk} : é igual a 1 se a camada j contém itens de paradas $k' > k$ e é colocada sobre outra camada que contém itens da parada k . Neste caso, estes itens de paradas k' precisam ser remanejados na parada k , caso contrário, esta variável é igual a 0. Os parâmetros deste modelo

são: ε^x : tolerância para as restrições de balanceamento, $\varepsilon^x \geq 0$; π_{jk} : é igual à soma total das áreas das faces superiores dos itens empacotados na camada j imediatamente antes da parada k ; ω^{dev} , ω^z : respectivamente, pesos relativos para as variáveis dev_k^x e z_{jk} ; ω^p, ω^q : respectivamente, fatores relativos para o peso e para a quantidade de itens remanejados; σ_j : pressão máxima suportada por qualquer ponto da face superior de um item da camada j ; τ_{jk}^x : posição do centro de gravidade, no eixo x , da camada j , imediatamente antes da parada k ; a_{jk} : é igual a 1 se a camada j possui algum item da parada k , e igual a 0, caso contrário; $cG^{v,x}$: centro de gravidade desejável para o veículo no eixo x ; $d_{p,p'}^x$: distância entre os pontos p e p' no eixo x ; \bar{h}_j : altura da camada j ; H_c : altura do compartimento c ; \mathcal{M} : é um número suficientemente grande; N : conjunto de todas as camadas; p^c : coordenada do compartimento c no eixo x ; P_{jk} : peso da camada j imediatamente antes da parada k ; P^k : peso total das camadas $j \in N$ imediatamente antes da parada k ; Q_{jk} : quantidade de itens na camada j imediatamente antes da parada k ; S : conjunto das paradas do veículo; $stab_{j,j'}$: é igual a 1 se a camada j contém pelo menos $minL_j$ itens de parada $k \geq k'$, em que k' é o maior valor de parada em que os itens da camada j' são descarregados. Neste caso, diz-se que j estabiliza j' em toda a rota, caso contrário, este parâmetro é igual a 0; ubL_c : limitante superior para o número de camadas completas em um compartimento c ; \bar{Z}_c : conjunto de posições em que alguma camada pode ser colocada no compartimento c :

$$\bar{Z}_c = \{r | r \in \mathbb{N}, 0 < r \leq ubL_c + 1, \}, c \in C$$

\bar{Z}_{jc} : conjunto de posições em que a camada j pode ser colocada no compartimento c . Como as camadas incompletas devem ficar sobre as camadas completas, define-se:

$$\bar{Z}_{jc} = \left\{ r \mid r \in \mathbb{N}, \begin{array}{l} 0 < r \leq ubL_c, \text{ se } j \in N' \\ r = ubL_c + 1, \text{ se } j \in N'' \end{array} \right\}, j \in N; c \in C$$

A função objetivo em (2) penaliza desvios de balanceamento e remanejamento de itens ao longo da rota. As restrições: (3) impõem que todas as camadas sejam empacotadas; (4) impõem que a altura da pilha de camadas colocada em um compartimento não seja maior que a altura deste compartimento; (5) impõem que as camadas não se sobreponham; (6)-(7) impõem que, imediatamente antes da parada k , o módulo da distância entre os centros de gravidade da carga e do veículo, no eixo x , multiplicado pelo peso da carga no veículo seja penalizado se maior do que ε^x ; (8) impõem que na posição imediatamente abaixo de uma camada completa j' haja uma camada j que estabilize a camada j' em toda a rota; (9) impõem que em qualquer posição abaixo de uma camada incompleta j' , a camada completa j , se houver, deve estabilizar a camada j' em toda a rota; (10) impõem, em toda a rota, que a pressão total exercida sobre qualquer ponto da face superior de um item da camada j' não seja maior do que a pressão máxima suportada por ele. Observe que se a camada j' não estiver na posição avaliada, as restrições correspondentes são nulas; (11) identifica os casos em que uma camada é colocada sobre outra e os itens da camada acima devem ser entregues depois dos da camada abaixo; (12) definem o domínio das variáveis de decisão.

$$\text{Min } \rho_2 = \omega^{dev} \sum_{k \in S} dev_k^x + \omega^z \sum_{f \in M'} \sum_{c \in C} Q_{fc}^< x_{fc} (\omega^q + \omega^p P'_{\varphi(f)}) \quad (13)$$

$$\sum_{f \in M'} x_{fc} (l_{\varphi(f)} w_{\varphi(f)} h_{\varphi(f)}) \leq fV(L \cdot W \cdot H'_c), \quad c \in C \quad (14)$$

$$x_{fc} h_{\varphi(f)} \leq H'_c, \quad f \in M'; c \in C \quad (15)$$

$$\sum_{f \in M'} x_{fc} P'_{\varphi(f)} a_{fk}^z \leq \max P_{ck}, \quad c \in C; k \in S \quad (16)$$

$$x_{fc} \leq stab'_{fc}, \quad f \in M'; c \in C \quad (17)$$

$$\sum_{c \in C} x_{fc} = 1, \quad f \in M' \quad (18)$$

$$cG_k^{l,x} = \left(\frac{\sum_{f \in M'} \sum_{c \in C} x_{fc} a_{fk}^z P'_{\varphi(f)} d_{0,p^c+L/2}^x}{\sum_{j \in N'} \sum_{c \in C} \sum_{r \in \bar{Z}_{jc}} \bar{y}_{jrc} P_{jk} d_{0,p^c+\tau_{jk}^x}^x} \right) / P^k, \quad k \in S \quad (19)$$

$$\begin{aligned} dev_k^x &\geq P^k(cG_k^{l,x} - cG_k^{v,x}) - \varepsilon^x \\ &\geq P^k(cG_k^{v,x} - cG_k^{l,x}) - \varepsilon^x \end{aligned} \quad k \in S \quad (20)$$

$$\begin{aligned} dev_k^x &\geq 0, k \in S \\ x_{fc} &\in \{0,1\}, f \in M'; c \in C \end{aligned} \quad (21)$$

A variável x_{fc} é igual a 1 se o item f é atribuído ao compartimento c , e igual a 0, caso contrário. Parâmetros adicionais: a_{fk}^{\geq} : é igual a 1 se o item f é de uma parada $k' \geq k$, e igual a 0, caso contrário; fV : fração do volume disponível no compartimento c para colocar os itens residuais; $maxP_{ck}$: é o maior peso que a pilha de camadas completas, se houver, colocada no compartimento c , pode suportar sobre ela imediatamente antes da parada k . Caso não haja itens de camadas completas no compartimento c , $maxP_{ck} = \mathcal{M}$; P_i' : peso de um item do tipo i ; $Q_{fc}^<$: quantidade de paradas anteriores à parada do item f em que os itens de camadas completas são descarregados do compartimento c . Portanto, define a quantidade de vezes em que o item f precisará ser remanejado se empacotado no compartimento c ; $stab'_{fc}$: é igual a 1 se não houver camadas completas no compartimento c , ou se qualquer camada completa j empacotada no compartimento c contém pelo menos $minL_j$ itens de parada $k \geq k'$, em que k' é a parada do item f . Nestes casos, diz-se que o compartimento c fornece estabilidade vertical ao item f em toda a rota, caso contrário, este parâmetro é igual a 0; \bar{y}_{jrc} : respectivamente, os valores de y_{jrc} , $j \in N$; $c \in C$; $r \in \bar{Z}_c$ obtidos com a resolução da formulação (2)-(12).

A função objetivo (13) penaliza desvios de balanceamento e o remanejamento dos itens residuais, ao longo da rota. As restrições: (14) impõem que o volume dos itens residuais atribuídos ao compartimento c não seja maior do que o volume disponível nele ajustado por um fator; (15) impõem que um item residual não seja atribuído a um compartimento com altura disponível menor do que a altura deste item; (16) limita, para qualquer parada, o peso total dos itens residuais atribuídos a um compartimento observando $maxP_{ck}$; (17) impõem que um item residual somente seja atribuído a um compartimento que forneça suporte vertical a ele em toda a rota; (18) impõem que todos os itens residuais sejam atribuídos; (19)-(20) possuem a mesma função das restrições (6)-(7). O centro de gravidade, no eixo x , do conjunto de itens residuais atribuídos a um compartimento é estimado no centro geométrico deste compartimento; (21) definem o domínio das variáveis de decisão.

4. Testes Computacionais

Para avaliar a heurística proposta foram utilizadas 12 classes de instâncias (I1,...,I12), e para cada uma delas geraram-se aleatoriamente 15 exemplares com o procedimento da Tabela 7, totalizando 180 problemas teste. A Tabela 5 apresenta os dados para a geração dos exemplares de cada classe. A Tabela 6 apresenta os dados para a geração de qualquer exemplar.

Seja, b_{ik} : demanda por itens do tipo i na parada k ; b_i^{max} e b_i^{min} : respectivamente, a maior e menor quantidade demandada do item de tipo i em uma iteração; fL : fator que define o volume máximo dos itens demandados; L^S : lista contendo as paradas $1, \dots, nS^{max}$; \bar{L}^S : lista auxiliar inicialmente vazia; nI^{max} e nS^{max} : respectivamente, a quantidade máxima de tipos de itens e de paradas geradas; v_i : volume do item do tipo i ; V_c : volume do compartimento c .

Tabela 5. Dados utilizados para a geração dos exemplares.

	I1	I2	I3	I4	I5	I6	I7	I8	I9	I10	I11	I12
$ C $	2	2	2	2	4	4	4	4	6	6	6	6
nS^{max}	3	3	4	4	6	6	8	8	9	9	12	12
fL	0.6	0.7	0.6	0.7	0.6	0.7	0.6	0.7	0.6	0.7	0.6	0.7

Tabela 6. Dados complementares utilizados para a geração dos exemplares.

nI^{max}	b_i^{min}	b_i^{max}	Semente para geração de números aleatórios
$2 \cdot C $	10	$Max(\lceil 0.11 \cdot \sum_{c \in C} V_c / v_i \rceil, b_i^{min})$	2013

Tabela 7. Procedimento para gerar os exemplares do problema.

71.	Início gInst()
72.	$tV = 0; aux = 1; b_{ik} = 0, i \in M, k \in S;$
73.	Selecione aleatoriamente nI^{max} tipos de itens distintos da lista de itens disponíveis e armazene-os em outra lista (L^t);
74.	Enquanto $aux = 1$
75.	Se a lista L^s estiver vazia
76.	$\bar{L}^s = L^s;$
77.	Selecione aleatoriamente uma parada k da lista \bar{L}^s e remova k de \bar{L}^s ;
78.	Selecione aleatoriamente um tipo de item i da lista L^t ;
79.	Selecione aleatoriamente entre $[b^{min}, \dots, b_i^{max}]$ um valor inteiro q ;
80.	Se $tV + (q * v_i) \leq fL * \sum_{c \in C} V_c$
81.	$tV = tV + (q * v_i); b_{ik} = b_{ik} + q;$
82.	Caso contrário
83.	$b_{ik} = b_{ik} + [(fL * \sum_{c \in C} V_c - tV)/v_i]; aux = 0;$
84.	Fim Enquanto
85.	Fim gInst()

Dados dos itens são fornecidos pela empresa que motivou o problema aqui abordado. Consideraram-se 28 tipos de itens para a geração dos exemplares. A tabela a seguir apresenta dados das dimensões destes itens:

Tabela 8. Dados estatísticos das dimensões dos tipos de itens utilizados.

	Mínimo	Máximo	Média	Desvio Padrão
Volume (cm³)	976	57.120	11.524	13.723

Definiram-se: $fV = 0,75; \min L_j = 4, j \in N; \varepsilon^x = 0,03 \cdot L \cdot P^{k=1}/|C|; cG^{v,x} = L;$
 $L = 100; W = 120; H_c = 143, c \in C; \alpha = 1; \beta = \gamma = 0; \omega^p = 0,2 \cdot \omega^q; \omega^z = \omega^q = 1$
e $ubL_c = [|N'|/|C|] + \delta \cdot [\min(ubL_c^1, ubL_c^2) - [|N'|/|C|]], c \in C$, em que ubL_c^1 é a maior quantidade de camadas completas cuja soma de suas alturas não é maior que H_c ; ubL_c^2 é a maior quantidade de camadas completas que podem ser empilhadas sobre a camada completa de maior resistência ao empilhamento mais uma unidade; δ : parâmetro que se $\delta = 1$, ubL_c é um limitante superior para a quantidade de camadas completas no compartimento c ; se $0 < \delta < 1$, não há garantias de que ubL_c ainda é o limitante superior. Utilizou-se $\delta = 0,7$.

Considera-se prioritário o remanejamento de itens e, desta forma, a penalização com o desbalanceamento é ponderada de modo que seja sempre menor do que a menor unidade de penalização com o descarregamento, $\omega^z[\omega^q + \min_{i \in M}(\omega^p P_i)]$, que neste caso não é menor do que 1. Observe que um limitante superior para a variável dev_k^x é: $P^k \cdot 2L - \varepsilon^x$. Portanto, define-se $\omega^{dev} = 1/(1 + \sum_{k \in S} P^k \cdot 2L - \varepsilon^x)$.

Todos os problemas de programação linear são resolvidos com auxílio do *software* IBM ILOG CPLEX versão 12.4. O tempo máximo permitido para a resolução de qualquer um desses problemas foi de 1200 segundos. O código das implementações foi escrito em C# e executado em um computador com processador Intel Core i7-2860QM, 3.6 GHz, 8GB de memória DDR3 SDRAM. O gerador de números aleatórios utiliza o método *System.Random* da biblioteca de classes Microsoft .NET *Framework* 4 e gera números uniformemente distribuídos.

Tentou-se resolver cada um dos exemplares com a heurística proposta e a tabela seguinte apresenta dados referentes a estes testes. Os valores desta tabela são médios e contabilizam apenas exemplares para as quais se encontrou ao menos uma solução factível. Os valores para cB , cS e $|N''|$ são referentes à melhor solução encontrada para cada exemplar. Para padronizar os valores de cB nesta tabela, eles foram recalculados com $\omega^{dev} = 0,01$ para todas as soluções já geradas.

Tabela 9. Informações referentes aos testes computacionais.

	I1	I2	I3	I4	I5	I6	I7	I8	I9	I10	I11	I12
cB	129,4	215,7	238,5	248,9	167,6	302,1	471	522,4	207,6	336,2	286,5	775,8

cS	21,8	35,2	28	35,4	0	20,1	13,7	26,2	0	6,7	3,6	8
nD	0,9	0,6	0,2	0,3	0,3	0,1	0,2	0,1	0,4	0,7	0,6	0,8
nA	1,5	1,5	1,3	1,6	1,9	2	1,9	2,3	2,4	2,3	2,4	2,4
$ M' $	144,2	90	64,2	70,4	113,3	110	91,6	105,8	122,8	164,2	136,3	158,5
$ N' $	4,8	6,6	5,5	6,9	13,4	15,2	13,8	15,3	19,1	22,2	18,9	21,9
$ N'' $	1,6	1,4	1,5	1,4	2,3	2	2,1	1,9	2,4	2,6	2,6	2,5
tC	0,4	0,5	0,5	0,7	406,3	408,4	203,2	324,4	801,8	1106,0	1270,5	1287,9
tI	1,5	0,7	0,2	0,4	0,8	0,7	0,7	0,9	1,2	1,6	1,7	1,7
tG	11,3	3,6	0,7	1,1	2,2	1,9	1,7	1,7	3,1	5,4	7,1	8
ubL_c	3,5	4,5	3,9	4,7	6,3	6,2	6,3	6,1	6,4	6,2	6,4	6,2
$bS(\%)$	30,8	9,1	27,3	11,1	21,4	7,7	23,1	0	0	16,7	6,7	7,7
$ M $	3,6	3,6	3,6	3,6	5,9	6,1	5,9	5,9	6,9	7,6	6,9	7,5
$\sum_{i \in M} M_i$	319,3	360	302,4	371,2	734,4	806,8	713,7	823,8	891,4	1115	869,9	1090,2
$p^{k=1}$	1141,4	1346,8	1137,9	1353,4	2553,2	2863,4	2562,4	2845,2	3583,9	4364,2	3639,6	4339,5
$ S $	3	3	4	4	6	6	8	8	9	9	12	12

em que, cB e cS : respectivamente, os custos com o balanceamento e com o descarregamento dos itens; nD e nA : respectivamente, o número de vezes em que as linhas 11 da Tabela 1 e 60 da Tabela 4 foram executadas; tC , tI e tG : respectivamente, os tempos, em segundos, gastos com a resolução dos problemas (2)-(12) e (13)-(21), e com o procedimento da Tabela 3; $bS(\%)$ é o valor percentual de soluções que puderam ser melhoradas com o procedimento da Tabela 4.

A Tabela 10 apresenta o percentual de exemplares que puderam ser resolvidos.

Tabela 10. Informações complementares dos testes computacionais.

	I1	I2	I3	I4	I5	I6	I7	I8	I9	I10	I11	I12
(%)	86,7	73,3	73,3	60	93,3	86,7	86,7	80,0	93,3	80	100	86,7

Ao menos 1 solução foi encontrada pela heurística em 150 dos 180 casos avaliados nesta seção. Os casos restantes não necessariamente podem ser resolvidos, pois o gerador de instâncias não garante a viabilidade deles. Em 76 destes 150 casos, uma nova solução pôde ser obtida pelo procedimento da Tabela 4; nos casos restantes, alguns itens residuais ficaram para fora do veículo. Em somente 21 destes 76 casos, as novas soluções são melhores do que as soluções iniciais para o objetivo avaliado e em 20 deles as novas soluções dominam (são melhores para um dos objetivos enquanto não são piores para o outro) as soluções iniciais. Em 19 destes 20 casos, as novas soluções foram melhores para os custos com o balanceamento e iguais para os custos com descarregamento. A menor frequência de casos (2 no total) em que custos com descarregamento são reduzidos pelo procedimento da Tabela 4, pode ser explicada pelo fato de os itens que impõem custos com descarregamento serem, na maioria dos casos avaliados, de camadas completas, que já têm suas posições definidas na solução inicial.

Os custos com balanceamento e com descarregamento geralmente crescem com o aumento do volume e peso dos itens, e com o aumento da quantidade de paradas, para as classes com um mesmo número de compartimentos. Em somente um exemplar avaliado, o desbalanceamento do veículo foi menor que a tolerância ε^x em todas as paradas avaliadas (o custo com desbalanceamento é nulo). Os custos de descarregamento foram maiores nos casos em que há maior quantidade de tipos de itens por compartimento (classes I1-I4), o que dificulta um melhor sequenciamento dos itens no veículo. Também devido a esta dificuldade, mais exemplares destas classes não puderam ser resolvidas em comparação com as demais classes.

O procedimento de “desmanchar” as camadas, vide linha 11 da Tabela 1, foi aplicado em 35% dos casos avaliados para tentar obter alguma solução factível e em 52,38% das vezes ele foi eficaz (levou a uma solução factível). Este procedimento foi aplicado em mais casos das classes I9-I12, as maiores avaliadas, do que em casos das classes I1-I4 ou I5-I8.

Os tempos: tC : crescem bastante com o aumento da quantidade de compartimentos. Nestes casos aumentam também o valor dos limitantes ubL_c , $c \in C$, e a quantidade de camadas completas (que dependem do tamanho e quantidades dos itens). Estes valores, juntamente com a quantidade de paradas, influenciam no tamanho das instâncias da formulação (2)-(12). Estes

tempos tendem a crescer também com o aumento do volume de itens e de paradas; tI : são sempre baixos (menores do que 2 segundos para todos os casos); tG : considerando todos os casos resolvidos, em média, não são maiores do que 4 segundos. Os tempos mais altos observados são referentes aos casos em que o procedimento de “desmanchar” as camadas é aplicado mais vezes. Nestes casos, o procedimento da Tabela 3 também é aplicado mais vezes, e cada problema construtivo resolvido conta com mais itens residuais (maior custo computacional).

5. Conclusão e trabalhos futuros

Apresentou-se uma heurística para a resolução de um PCC prático baseada na geração de camadas horizontais e na resolução de problemas menores de empacotamento e de atribuição. As formulações utilizadas contemplam as diversas restrições e objetivos práticos e permitiram que instâncias de tamanho pequeno/moderado fossem resolvidas com dados realistas. As diversas restrições são contempladas simultaneamente e devem ser satisfeitas ao longo de toda a rota, o que dificulta bastante a resolução de problemas maiores.

Um procedimento frequentemente utilizado na prática, e que não foi explorado neste trabalho, é o rearranjo dos itens, em cada parada, para outras posições no veículo. Este rearranjo pode ser vantajoso para evitar que custos com descarregamento sejam reincidentes, para a manutenção dos desvios no balanceamento e para obter soluções para exemplares que antes seriam infactíveis. Propomos considerar este tipo de rearranjo em trabalhos futuros.

Outro caminho, possivelmente a seguir, é buscar empacotar os itens também em colunas/paredes verticais. Estes tipos de arranjo podem ser vantajosos para o sequenciamento correto da carga, pois mais itens podem ser acessados de maneira independente, embora também possam ter desvantagens, por exemplo, no quesito estabilidade horizontal.

Agradecimentos: ao CNPq e à FAPESP pelo auxílio financeiro.

Referências

- Alvarez-Valdes, R.; Parreño, F.; Tamarit, j.M.** (2005), A branch-and-cut algorithm for the pallet loading problem. *Computers & Operations Research*, v.32, n.11, p.3007–3029.
- Bortfeldt, A., Wäscher, G.**, Container Loading Problems - A State-of-the-Art Review, No 120007, *FEMM Working Papers*, Otto-von-Guericke University Magdeburg, FEMM, 2012.
- De Sha E.** *Area efficiency and volume efficiency for loading cargo*. Masters Thesis, U.S. Navy Postgraduate School, 1970.
- Eley, M.** (2002), Solving container loading problems by block arrangement. *European Journal of Operational Research*, v. 141, n. 2, p. 393-409.
- George J.A., Robinson D.** (1980), A heuristic for packing boxes into a container. *Computers & Operations Research* 7, 147-156.
- Haessler, R. W., Talbot, F. B.** (1990), Load planning for shipments of low density products. *European Journal of Operational Research*, v. 44, n. 2, p. 289-299.
- Junqueira, L., Morabito, R., Yamashita, D. S.** (2011), Abordagens para problemas de carregamento de contêineres com considerações de múltiplos destinos. *Gestão & Produção*, S. Carlos, v. 18, n. 2, p. 265-284.
- Lim, A. & Zhang, X.** (2005) The container loading problem. In: *ACM Symposium on Applied C.*, 913-917.
- Morabito, R., Arenales, M.** (1994), An And/Or-graph approach to the container loading problem. *International Transactions in Operational Research*, v. 1, n. 1, p. 59-73.
- Morabito, R., Arenales, M.** (1997), Abordagens para o problema do carregamento de contêineres. *Pesquisa Operacional*, v. 17, n. 1, p. 29-56.
- Peleg K., Peleg E.** (1976) Container dimensions for optimal utilization of storage and transportation space. *Computer Aided Design* v.8, n.3, 775-781.
- Pisinger, D.** (2002), Heuristics for the container loading problem. *European Journal of Operational Research*, v. 141, n. 2, p. 382–92.
- Steudel H.** (1979), Generating pallet loading patterns - a special case of the two dimensional cutting stock problem. *Management Science* v.25,n.10, 997-1004.
- Wäscher, G.; Haussner, H.; Schumann, H.** (2007), An improved typology of cutting and packing problems. *European Journal of Operational Research*, v. 183, n. 3, p. 1109-1130.