



# Um algoritmo de busca tabu para o carregamento de contêineres com caixas idênticas

## *A tabu search algorithm for the container loading problem with identical boxes*

Guilherme Izidoro Poli<sup>1</sup>  
Vitória Pureza<sup>2</sup>

**Resumo:** Este trabalho aborda o problema do carregamento de contêineres, cuja forma mais geral consiste em arranjar itens (por exemplo, produtos embalados em caixas) de vários tamanhos dentro de objetos maiores (por exemplo, contêineres) com máximo aproveitamento do volume disponível. Em particular, é considerado o caso especial em que as caixas a serem arranjadas são retangulares e idênticas. Com o objetivo de resolver este problema, estendemos a heurística de busca tabu de Pureza e Morabito, originalmente projetada para o problema de carregamento de paletes do produtor. Partindo-se de uma solução inicial gerada por uma heurística de blocos, são realizados movimentos de troca de orientação e/ou expansão de blocos selecionados, resultando na geração de novos padrões de carregamento. Resultados computacionais obtidos com 22 exemplos da literatura sugerem que a abordagem proposta é promissora.

**Palavras-chave:** Transporte intermodal. Carregamento de contêineres. Caixas idênticas. Busca tabu.

**Abstract:** This paper tackles the container loading problem, which consists of determining the most effective way of arranging items (for example, products packed in boxes) in larger objects (for example, containers). In particular, we consider the special case in which the boxes to be arranged are rectangular-shaped and identical. In order to solve this problem, we extend the tabu search algorithm by Pureza and Morabito, originally designed for the manufacturer's pallet loading problem. From a starting solution generated by a block heuristic, the algorithm performs moves, which change the orientation and/or expand selected blocks, resulting in new loading patterns. Computational results obtained with 22 instances from the literature suggest that the proposed approach is promising.

**Keywords:** Intermodal transportation. Container loading. Identical boxes. Tabu search.

## 1 Introdução

As últimas décadas vêm sendo marcadas por um crescente interesse na aplicação de técnicas de pesquisa operacional com vistas à resolução de problemas da cadeia logística (BISCHOFF; DOWSLAND, 1982). Este fenômeno é em parte devido ao aumento nos custos de estocagem e distribuição, o que levou à necessidade premente do uso racional dos recursos disponíveis. Mais recentemente, muitas empresas se inseriram no mercado global e questões de custo, rapidez e preço tornaram-se primordiais para sua sobrevivência nesses ambientes altamente competitivos.

No que diz respeito a atividades logísticas, o transporte é a atividade mais importante porque absorve, em média, de um a dois terços dos custos logísticos (BALLOU, 2004). Neste contexto, a *unitização de carga* tornou-se uma estratégia eficiente para o transporte de produtos ao longo

da cadeia de suprimentos. A unitização de carga consiste no carregamento de volumes unitários da carga sobre paletes ou dentro de contêineres de maneira que a configuração final se conforme exatamente com dimensões especificadas de comprimento, largura e altura. Estas dimensões são geralmente selecionadas para maximizar o espaço utilizado nos veículos selecionados para o transporte dos produtos.

O transporte intermodal, ou seja, a integração entre os diversos modos de transporte tornou-se possível com o uso de contêineres. Contêineres são grandes caixas de dimensões padronizadas. De acordo com Morabito (1992), os volumes unitários a serem carregados dentro dos contêineres podem estar soltos ou paletizados para facilitar a estofagem (carregamento do contêiner) e a desova (descarregamento do contêiner) por meio de empilhadeiras.

<sup>1</sup> Diretoria de Sistemas de Gestão e Controle, Área de Tecnologia, Banco Itau - Unibanco S.A., Av. do Estado, 5533, São Paulo, SP, Brasil, e-mail: guilherme\_poli@yahoo.com.br

<sup>2</sup> Departamento de Engenharia de Produção, Universidade Federal de São Carlos – UFSCar, Rod. Washington Luiz, Km 235, São Carlos, SP, Brasil, e-mail: vpureza@dep.ufscar.br

Em linhas gerais, o problema de carregamento de contêineres consiste em carregar produtos embalados em caixas. Admite-se, em geral, que as caixas tenham formato retangular e que devam ser arrançadas ortogonalmente, isto é, com os seus lados paralelos aos lados do contêiner. Sob a perspectiva da pesquisa operacional, este problema pertence à classe de *problemas de empacotamento* cujo principal objetivo é a otimização do espaço utilizado pela carga (MORABITO, 1992).

As principais vantagens do uso do contêiner são a diminuição dos custos de carga, descarga e manipulação nos terminais, e diminuição de perdas por danificação e furto. Uma função importante dos contêineres é a de facilitar a transferência dos materiais de um modo de transporte a outro com um mínimo de manipulação ou pelo menos com manipulação mecanizada (MAGEE, 1977). Muitos trabalhos tratam da estabilidade estática da carga, porém são poucas as contribuições reportadas que abordam sua estabilidade dinâmica quando o contêiner é movimentado pelos diferentes modais de transporte. Situações com um grau mínimo de estabilidade e de distribuição de peso no contêiner devem ser, portanto, consideradas (DAVIES; BISCHOFF, 1999).

As principais desvantagens do contêiner são seu custo e peso, a necessidade de consolidação de carga, a programação e manipulação do seu retorno vazio ao ponto de origem, ainda que alguns contêineres possam ser desmontados em seu retorno (MORABITO, 1992). Em Choong, Cole e Kutanoglu (2002), é apresentada uma análise computacional do efeito do horizonte de planejamento para o gerenciamento de contêineres vazios em uma rede de transportes intermodal. De acordo com Araújo (2006), a organização de cargas em contêineres, seja ela paletizada ou distribuída em caixas, é uma das tarefas mais complexas em empresas que têm no transporte de carga um fator logístico importante e de alto custo.

Neste trabalho, tratamos o caso especial em que as caixas a serem arrançadas são retangulares e idênticas, visando otimizar a utilização do volume do contêiner, ou seja, maximizar o número de caixas empacotadas. No entanto, mesmo com a tendência do mercado de substituir grandes pedidos por muitos pedidos pequenos com grande variedade de itens (CECILIO, 2003), persiste a necessidade comercial de produtores que possuem cargas de tamanho considerável de um único tipo de produto. Apesar de ser possível obter um carregamento satisfatório utilizando heurísticas para o problema geral (GEORGE; ROBINSON, 1980; BISCHOFF; MARRIOTT, 1990; GEHRING; BORTFELDT, 1997; ELEY, 2002; PISINGER, 2002), é possível melhorar o carregamento endereçando as especificidades de situações em que as caixas são idênticas (GEORGE, 1992).

A heurística de busca tabu de Pureza e Morabito (2006) foi originalmente projetada para resolução do problema de carregamento de paletes do produtor. Este problema visa determinar o arranjo ortogonal de caixas idênticas que maximizem a utilização da área do palete. Um dos motivos de se estender a referida heurística para o tratamento do problema de carregamento de contêineres é o número expressivo de soluções ótimas obtidas com instâncias difíceis da literatura em tempos computacionais razoáveis. Dada a similaridade entre os dois problemas, espera-se também obter bons resultados com a extensão da heurística.

O restante deste trabalho é organizado como se segue. A Seção 2 descreve o problema geral de carregamento de contêineres e algumas de suas variações, e faz uma breve revisão da literatura associada à variante tratada nesta pesquisa. Na Seção 3, o algoritmo proposto é apresentado. Na Seção 4, são discutidos os experimentos computacionais e resultados obtidos frente a métodos da literatura, seguidos de conclusões e próximos passos da pesquisa na Seção 5.

## 2 O problema do carregamento de contêineres com caixas idênticas

O problema de carregamento de contêineres (PCC) pode ser definido como se segue: dado um conjunto de caixas retangulares (itens) disponíveis em  $n$  tipos de dimensões  $(l_i, w_i, h_i)$  ( $i = 1..n$ ) e um conjunto de contêineres retangulares (objetos) disponíveis em  $m$  tipos de dimensões  $(L_j, W_j, H_j)$  ( $j = 1..m$ ), deseja-se arranjar as caixas ortogonalmente e sem sobreposição dentro dos contêineres de forma a maximizar o valor ou o volume do carregamento. As dimensões  $l_i, w_i$  e  $h_i$  referem-se respectivamente ao comprimento, à largura e à altura de caixas do tipo  $i$ ; similarmente,  $L_j, W_j$  e  $H_j$  referem-se respectivamente ao comprimento, à largura e à altura de contêineres do tipo  $j$ . Apesar de situações práticas não estarem limitadas a itens e objetos retangulares (BIRGIN; MARTÍNEZ; RONCONI, 2005), a dificuldade no tratamento de problemas com elementos não retangulares ou irregulares tem restringido a maioria das contribuições na literatura ao empacotamento de cubos em cubos maiores.

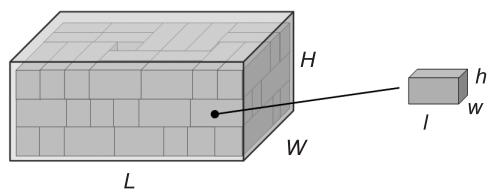
Um número expressivo de variações do PCC vem sendo explorado nas últimas décadas, em particular quando existe mais de um tipo de caixas a serem carregadas ( $n > 1$ ). Exemplos incluem a seleção e carregamento de múltiplos contêineres de diferentes tipos ( $m > 1$ ) (CHEN; LEE; SHEN, 1995), o carregamento de um único contêiner ( $m = 1$ ) com o maior volume de carga (GEORGE; ROBINSON, 1980; CECÍLIO; MORABITO, 2004) e o carregamento da carga total em um contêiner com largura e altura fixas e profundidade mínima (BISCHOFF; MARRIOTT,

1990; BISCHOFF; RATCLIFF, 1995a). Além de restrições de sobreposição das caixas, alguns trabalhos prescrevem o balanceamento de peso e localização dos itens dentro do contêiner, como limitações de orientação de caixas (MARTELLO; PISINGER; VIGO, 2000), limitações de peso bruto (DAVIES; BISCHOFF, 1999), empilhamento máximo (RATCLIFF; BISCHOFF, 1998; BISCHOFF, 2006; GENDREAU et al., 2006) e estabilidade da carga (BISCHOFF; RATCLIFF, 1995b; GEHRING; BORTFELDT, 1997; ARAÚJO; ARMENTANO, 2007; JUNQUEIRA; MORABITO; YAMASHITA, 2010).

O PCC pertence à classe de problemas combinatórios NP-*hard*, o que torna um desafio à obtenção de soluções ótimas dentro de um limite de tempo razoável. De fato, a complexidade do problema geralmente prescreve a utilização de métodos heurísticos para a resolução do problema, o que motiva a busca por algoritmos eficientes. Discussões sobre classificação de problemas, modelos matemáticos, métodos de solução e aplicações de carregamento de contêineres podem ser encontradas em Dyckhoff (1990), Bischoff e Marriott (1990), Gehring, Menschner e Meyer (1990), Dyckhoff e Finke (1992), Scheithauer (1992), Abdou e Yang (1994), Morabito e Arenales, (1994, 1997), Gehring e Bortfeldt (1997), Pisinger (2002), Eley (2002), Moura e Oliveira (2005), Araújo e Armentano (2007), e Wäscher, Haußner e Schumann (2007).

Neste trabalho, endereçamos a situação relativamente pouco explorada na literatura em que a carga consiste de caixas idênticas, ou seja, quando  $n = 1$  (Figura 1). Em carregamentos deste tipo, admite-se que as caixas estejam disponíveis em grandes quantidades de maneira que o objetivo consiste em se obter um carregamento que maximize o número de caixas em um único contêiner ( $m = 1$ ). Segundo a tipologia de Wäscher, Haußner e Schumann (2007), o problema é um exemplo de IIPP (*Identical Item Packing Problem* – Problema de Empacotamento de Itens Idênticos) tridimensional retangular. Apenas restrições de sobreposição são consideradas neste estudo.

Um dos primeiros estudos reportados para o PCC com caixas idênticas é apresentado em Han, Knott e Egbelu (1989). Naquele trabalho, é descrito um



**Figura 1.** Exemplo de carregamento com caixas idênticas em um contêiner (Adaptado de Morabito e Purity, 2008).

algoritmo de programação dinâmica que preenche o contêiner em camadas construídas por prismas com formato em “L”. George (1992), por sua vez, propõe uma abordagem, também de construção em camadas, em que cada dimensão das caixas é investigada para definir a dimensão de profundidade da camada, e um algoritmo é utilizado para construir um padrão na camada (de largura  $W$ ), a qual é repetida ao longo da direção  $H$ .

Liu e Hsiao (1997) apresentam um algoritmo de carregamento em camadas para obtenção do carregamento de maior estabilidade de carga dentre aqueles com maior utilização do volume do contêiner. O método gera várias opções de camadas, e um modelo linear sujeito a restrições de altura máxima, carga máxima sobre o chão do contêiner e limitantes superiores triviais é utilizado para selecionar o conjunto de camadas que comporão o carregamento. Lins, Lins e Morabito (1999) propõem um algoritmo recursivo que representa empacotamentos factíveis como designações factíveis de profundidade em uma tríade (3 “fios” representando as dimensões  $X$ ,  $Y$  e  $Z$ ). Cada designação corresponde a uma partição do contêiner em subcontêineres específicos. Cada subcontêiner pode ser recursivamente dividido em subcontêineres adicionais, de acordo com as especificações na tríade. Um subcontêiner que não é dividido é carregado com caixas de mesma orientação. Não há restrições de orientação das caixas, no entanto o algoritmo pode ser facilmente adaptado para o caso em que a orientação é fixada. Em Lins, Lins e Morabito (2002), o algoritmo anterior é estendido para o tratamento do problema de empacotamento de caixas com  $d$  dimensões em um  $d$ -contêiner.

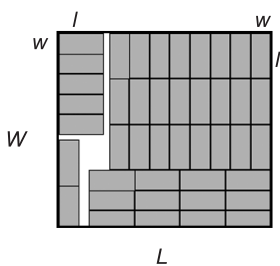
Tanto as abordagens de Han, Knott e Egbelu (1989) como de Liu e Hsiao (1997) e Lins, Lins e Morabito (1999, 2002) guardam estreita relação com o problema de se arranjar o máximo número de retângulos de dimensões  $(l, w)$  dentro do retângulo  $(L, W)$ . Esta variante bidimensional do PCC com caixas idênticas é conhecida na literatura como o *problema de carregamento de paletes do produtor*, sendo tratada com relativo sucesso por heurísticas construtivas conhecidas como heurísticas de blocos (BISCHOFF; DOWSLAND, 1982; NELISSEN, 1994; SCHEITHAUER; TERNO, 1996). Especificamente, o algoritmo de programação dinâmica de Han, Knott e Egbelu (1989) é semelhante ao proposto em Steudel (1979), e Liu e Hsiao (1997) utilizam uma modificação da heurística de blocos de Smith e De Cani (1980) para obtenção das camadas candidatas a compor o carregamento. O algoritmo de Lins, Lins e Morabito (2002), por sua vez, é uma generalização da heurística de Morabito e Morales (1998, 1999).

No empacotamento bidimensional, heurísticas de bloco pesquisam *layouts* de padrões que resultam em um bom ajuste em torno das bordas da área do

paleta e estendem estes padrões ao centro do paleta de maneira a formar um ou mais blocos retangulares em que as caixas possuem a mesma orientação pré-fixada (Figura 2). Cada *layout* resultante corresponde a uma solução e aquela com o maior número de retângulos é selecionada. Note na figura que este problema é chamado de bidimensional, pois considera duas dimensões relevantes - comprimento e largura do paleta ( $L, W$ ) e comprimento e largura das caixas ( $l, w$ ) - e duas possíveis orientações de caixas em relação ao comprimento  $L$  do paleta - dimensão  $l$  e dimensão  $w$ . Note também que estas orientações definem os retângulos  $(l, w)$  e  $(w, l)$ .

Quanto maior o número de blocos considerado, maior tende a ser a qualidade do padrão de carregamento gerado. Por exemplo, a heurística de Smith e De Cani constrói até quatro blocos, permitindo um carregamento de 43 caixas para a instância  $(L, W, l, w) = (42, 39, 9, 4)$  (Figura 2). A heurística de Morabito e Morales, por sua vez, aplica recursivamente a heurística de cinco blocos de Bischoff e Dowsland (1982) na área de cada bloco gerado, resultando em um carregamento de 44 caixas para a mesma instância.

Padrões de carregamento podem ser classificados em guilhotinados, não guilhotinados de 1ª ordem, e não guilhotinados de ordem superior (ARENALES; MORABITO, 1995). Um padrão guilhotinado é obtido por sucessivos “cortes” guilhotinados, por sua vez definidos como cortes que, quando aplicados a um retângulo, produzem 2 novos retângulos (Figura 3a). Um padrão não guilhotinado de 1ª ordem é obtido por sucessivos cortes guilhotinados e/ou cortes não



**Figura 2.** Padrão de carregamento com quatro blocos gerado pela heurística de Smith e De Cani (1980) para a instância  $(L, W, l, w) = (42, 39, 9, 4)$ .

guilhotinados de 1ª ordem. Um corte é chamado não guilhotinado de 1ª ordem se, quando aplicado a um retângulo, produz 5 novos retângulos arranjados de maneira a não formar um padrão guilhotinado (veja o corte em negrito na Figura 3b). Um padrão não guilhotinado de ordem superior é aquele que não pode ser obtido por sucessivos cortes guilhotinados e/ou cortes não guilhotinados de 1ª ordem (Figura 3c).

Uma característica de heurísticas de bloco é que os padrões de carregamento gerados estão, em geral, limitados a padrões não guilhotinados de 1ª ordem. O algoritmo recursivo de Morabito e Morales (1998, 1999), por exemplo, foi capaz de resolver otimamente mais de 99,9% dentre 20.000 exemplos. Como esta heurística tem a garantia de obter padrões ótimos de 1ª ordem, os exemplos não resolvidos correspondem a casos cujas soluções ótimas são de ordem superior.

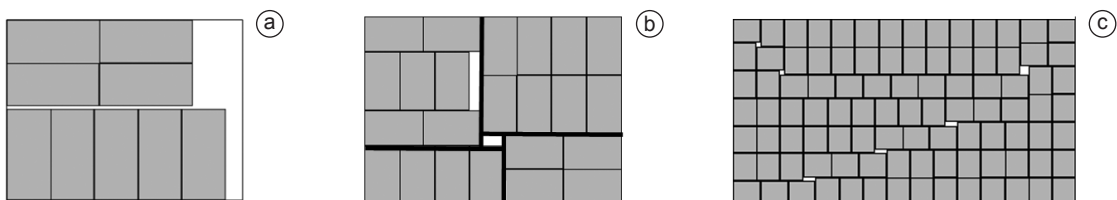
Em vista desta limitação, Pureza e Morabito (2006) propõem um algoritmo de busca tabu (aqui referido como *PM2*) cuja principal característica consiste na aplicação de movimentos de expansão de blocos a partir de um padrão de carregamento inicial gerado por uma heurística de blocos. O algoritmo *PM2* foi aplicado a 21 instâncias clássicas da literatura e, das 17 instâncias resolvidas otimamente, sete correspondem a exemplos cujas soluções ótimas são padrões não guilhotinados de ordem superior. A próxima seção descreve como o algoritmo *PM2* pode ser estendido para o tratamento do problema de carregamento de contêineres com caixas idênticas.

### 3 O algoritmo proposto (PM3)

Assim como *PM2*, o algoritmo *PM3* é um procedimento heurístico iterativo no qual a busca por melhores padrões de carregamento ocorre no espaço factível. Isso permite que uma solução esteja disponível em qualquer iteração. Os passos gerais do algoritmo são descritos na Figura 4 e discutidos nas seções seguintes.

#### 3.1 Padrão de carregamento inicial

A geração do padrão inicial (passo 1 na Figura 4) é realizada por uma extensão da heurística de Smith e De Cani (1980), cuja forma original é utilizada em *PM2*. Conforme visto na seção anterior, esta heurística



**Figura 3.** Padrões de carregamento bidimensionais: (a) padrão guilhotinado; (b) corte (em negrito) e padrão não guilhotinado de 1ª ordem; (c) padrão não guilhotinado de ordem superior.

gera até quatro blocos de caixas arranjadas sobre a superfície do palete (definido pelas dimensões  $L$  e  $W$ ) conforme uma orientação pré-fixada das caixas ( $lw$  ou  $wl$ ). O procedimento varia as dimensões dos blocos (inclusive com a possibilidade de serem nulas) examinando todas as possíveis modificações nestes blocos e retornando o padrão que resulta no maior número de caixas dentro dos blocos.

No caso do problema tridimensional, esta ideia é estendida considerando que as caixas podem ser posicionadas no chão do contêiner (definido pelas dimensões  $L$  e  $W$ ) sobre três faces distintas ( $lw$ ,  $lh$  ou  $wh$ ). Para cada uma dessas três opções é aplicada a heurística de Smith e De Cani para o preenchimento do chão do contêiner. Em seguida, o padrão resultante é repetido na direção da altura  $H$  do contêiner (adicionando camadas idênticas àquela preenchida no chão) enquanto não se ultrapasse o teto do contêiner. Dentre as três possibilidades, o padrão de carregamento que resulta no maior número de caixas é selecionado para compor a solução inicial. A Figura 5 ilustra um exemplo de três padrões de preenchimento do chão do contêiner (cada qual com dois blocos) e a solução inicial obtida com este procedimento.

Na Figura 5a, o padrão de carregamento do chão do contêiner com as caixas arranjadas sobre a face  $(l, w)$  apresenta uma camada com 38 caixas. Repetindo este padrão até o teto do contêiner (total de 10 camadas), é possível empacotar 380 caixas. Na Figura 5b (caixas arranjadas sobre a face  $(w, h)$ ), obtêm-se um total de 355 caixas divididas em cinco camadas. Finalmente,

na Figura 5c, o arranjo das caixas arranjadas sobre a face  $(l, h)$  resulta no empacotamento de 385 caixas divididas em sete camadas. O arranjo da Figura 4c seria, portanto, selecionado para a construção do carregamento inicial (Figura 5d).

Uma situação especial ocorre quando o padrão inicial consiste de um único bloco (padrão homogêneo). Como a aplicação da busca tabu (passo 3 na Figura 4) envolve movimentos que requerem pelo menos dois blocos no padrão, o bloco é dividido o mais próximo possível de suas coordenadas centrais, resultando em até quatro blocos. Cada bloco resultante é então deslizado para as extremidades mais próximas dos limites do contêiner. Cabe ressaltar que nos experimentos realizados (Seção 4), nenhum padrão homogêneo inicial foi obtido.

### 3.2 Estrutura de vizinhança

Seguindo-se à geração do padrão inicial, verifica-se se este satisfaz algum critério de otimalidade (passo 2 na Figura 4). O critério utilizado no algoritmo consiste na satisfação do limitante superior trivial

$$\left\lfloor \frac{L \times W \times H}{l \times w \times h} \right\rfloor.$$

Se o número de caixas do padrão é igual a este limitante, o padrão é ótimo e o algoritmo é finalizado. Caso este critério não seja satisfeito, inicia-se a fase de busca em vizinhança guiada por um algoritmo de busca tabu. Padrões candidatos à seleção resultam de *movimentos* aplicados aos blocos do padrão corrente (passo 3.1 na Figura 4). Além dos movimentos de expansão de blocos adotados no

**Passo 1.** Gere um padrão inicial  $S_0$ . Se  $S_0$  consistir de um único bloco, divida-o em suas coordenadas centrais, e deslize os blocos resultantes para as extremidades mais próximas do contêiner. Faça  $f(S_0)$ =número de caixas em  $S_0$ ,  $blk$ =número de blocos em  $S_0$  e  $ótimo$ =falso.

**Passo 2.** Se  $S_0$  atender algum critério de otimalidade, faça  $ótimo$ =verdadeiro e vá para o passo 4. Caso contrário, seja  $t$  o tamanho da lista tabu  $L$ , e  $maxit$  o número máximo de iterações sem melhoria da fase tabu. Faça  $S=S_0$ ,  $f(S)=f(S_0)$ ,  $S^*$  (solução incumbente)= $S$ ,  $f(S^*)=f(S)$ ,  $it=0$  (número da iteração corrente),  $it^*=it$  (número da iteração em que foi obtida a solução incumbente) e  $L=\emptyset$ .

**Passo 3.** Repita

**3.1** Para cada bloco  $i=1, \dots, blk$ , calcule a vizinhança do padrão corrente (descrita na Seção 3.2). Seja  $mv$  o número total de movimentos computados.

**3.2** Se  $mv = 0$ , vá para o passo 4. Caso contrário, faça  $it=it + 1$  e selecione o movimento não tabu com a maior avaliação segundo a função  $G$  (descrita na Seção 3.2.3). Se todos os movimentos estão proibidos, elimine o elemento mais antigo de  $L$  ( $t$ -ésimo elemento) e repita o passo. Caso contrário, efetive o movimento gerando um novo padrão  $S'$ , faça  $S=S'$  e inclua os atributos do movimento em  $L$ .

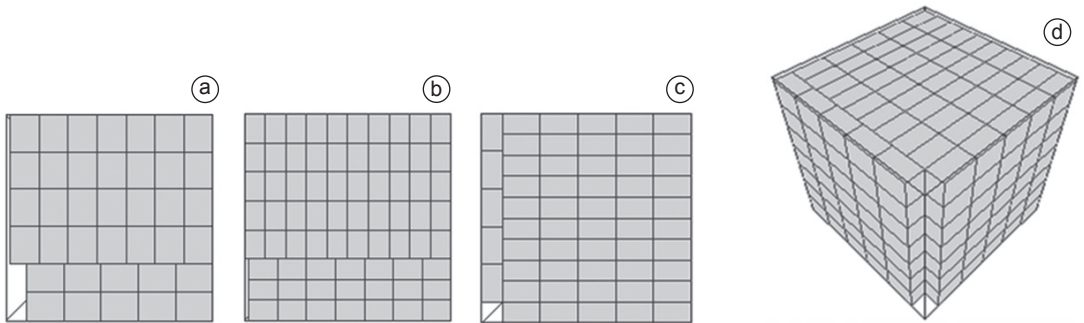
**3.3** Se  $f(S) > f(S^*)$ , faça  $S^*=S$ ,  $f(S^*)=f(S)$  e  $it^*=it$ . Se  $S^*$  atende algum critério de otimalidade, faça  $ótimo$ =verdadeiro e vá para o passo 4.

**3.4** Divida com probabilidade  $divip$  cada bloco do padrão corrente  $S$  nas coordenadas mais próximas do centro do contêiner, e deslize-os o máximo possível para as extremidades mais próximas do contêiner. Repita o passo até que nenhum deslizamento seja possível.

**3.5** Una com probabilidade  $unip$  os blocos do padrão corrente  $S$ .

Até que  $mv = 0$  ou  $(it - it^*) = maxit$  ou  $ótimo = verdadeiro$ .

Figura 4. Passos do algoritmo PM3.



**Figura 5.** Opções de carregamento do chão do contêiner para a instância  $(L, W, H, l, w, h) = (50, 50, 50, 9, 7, 5)$  e solução inicial resultante: (a) caixas sobre a face  $(l, w)$ ; (b) caixas sobre a face  $(w, h)$ ; (c) caixas sobre a face  $(l, h)$ ; (d) padrão de carregamento correspondente à adição de camadas com o padrão (c) na direção da altura  $H$  do contêiner.

algoritmo *PM2*, foram considerados outros dois tipos de movimentos: troca de orientação das caixas de cada bloco, e troca de orientação das caixas de cada bloco seguida de expansão. Cada movimento é caracterizado por até seis parâmetros, discutidos a seguir.

**(1) Tipo de movimento:** Denotados por “e” (expansão), “t” (troca de orientação), e “te” (troca de orientação seguida de expansão).

**(2) Bloco a iniciar um movimento:** Denotado por *bloco ativo*, é identificado por um número  $i \in \{1, 2, \dots, blk\}$  em que *blk* é o número de blocos no padrão corrente.

**(3) Direção de expansão (movimentos do tipo “e” e “te”):** Consideram-se as seis direções de expansão aplicadas a cada face do bloco, a saber, “para a direita”, “para a esquerda”, “acima”, “abaixo”, “para frente”, e “para trás” (note que em *PM2*, apenas as quatro primeiras são consideradas). A Figura 6 ilustra exemplos de possíveis expansões “para trás”.

**(4) Camadas a serem expandidas (movimentos do tipo “e” e “te”):** A expansão de um dado bloco ativo é feita tomando-se quaisquer camadas da face de expansão, desde que adjacentes. Quando a expansão contempla todas as camadas da face do bloco (Figura 6a), o resultado é o mesmo bloco aumentado. Caso apenas algumas camadas da face se expandam, o bloco ativo é dividido em dois (separados pela linha pontilhada na Figura 6b) ou três blocos (separados pelas linhas pontilhadas na Figura 6c).

**(5) Número de caixas adicionadas a cada camada de expansão (movimentos do tipo “e” e “te”):** Este número pode ser qualquer desde que os limites do contêiner não sejam ultrapassados. Na Figura 6a, três caixas foram adicionadas a todas as camadas da face de expansão do bloco, enquanto na Figura 6b, duas caixas foram adicionadas às três primeiras camadas da mesma face.

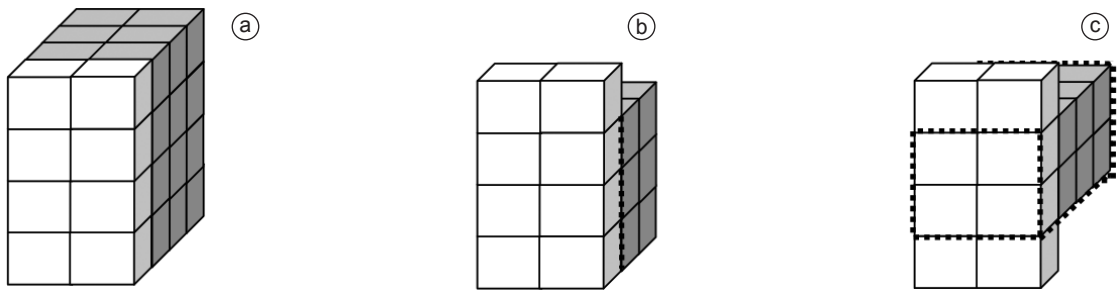
**(6) Configuração dos volumes remanescentes dos blocos passivos (movimentos do tipo “e” e “te”):** Movimentos de expansão geralmente resultam na sobreposição de blocos nas adjacências, denotados de *blocos passivos*. Quando há sobreposição,

blocos passivos são eliminados e novos blocos são criados nas áreas não superpostas. São consideradas 4 configurações candidatas para geração destes novos blocos. As configurações envolvem a divisão da área em até cinco volumes retangulares dentre 10 possíveis. A Figura 7 ilustra estas configurações (com seus volumes identificados por A, B, ..., J) para movimentos “para trás” (para as demais direções basta rotacionar consecutivamente os desenhos em  $90^\circ$ ). Se a expansão do bloco ativo sobrepuser mais de um bloco passivo, volumes remanescentes de cada um destes blocos são tratados separadamente. Note que no problema bidimensional as projeções de A, B, C, F, G, H, I, e J na face superior do bloco (em negrito) correspondem exatamente às áreas remanescentes utilizadas em *PM2*, enquanto a projeção de E consiste da área de expansão do bloco ativo.

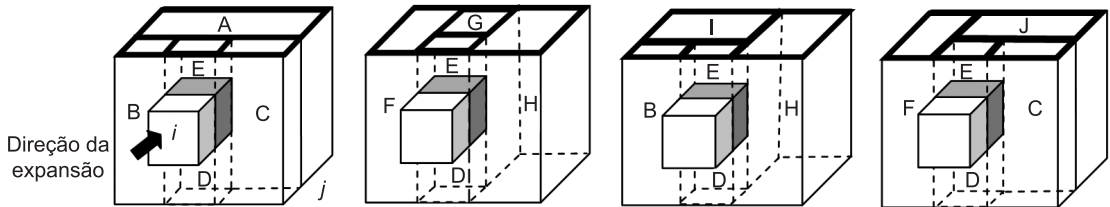
Em cada volume de uma dada configuração, é analisado o posicionamento de caixas com cada uma das seis orientações (ou seja, com a face  $lw, wl, lh, hl, wh$  ou  $hw$  sobre o plano paralelo ao chão do contêiner) e selecionada aquela que maximiza o número de caixas carregadas. Uma vez calculadas as quatro possíveis configurações, a opção selecionada para o movimento é aquela que maximiza o número total de caixas. Empates são resolvidos aleatoriamente.

Em movimentos do tipo “t” (troca de orientação), são analisadas todas as demais orientações de caixas para o bloco ativo desde que não provoquem sobreposição em outros blocos. Assim sendo, este tipo de movimento não resulta em blocos passivos (ou o bloco ativo é também passivo), podendo ser visto como um operador unário. Caso haja mais de uma orientação factível, é selecionada aquela que resulta no maior ganho em número de caixas, com empates resolvidos aleatoriamente.

Note que a complexidade da estrutura de vizinhança faz da análise de seu tamanho uma tarefa desafiadora. O número de movimentos (e, portanto, de soluções vizinhas) depende das dimensões do contêiner e das caixas, assim como do padrão de carregamento corrente (número de blocos, número de caixas em cada



**Figura 6.** Expansões “para trás” (em cinza) de um bloco com oito caixas (em branco): (a) expansão da face inteira do bloco; (b) expansão das três primeiras camadas do bloco; e (c) expansão da segunda e da terceira camada do bloco.



**Figura 7.** Quatro possíveis configurações dos volumes remanescentes do bloco passivo  $j$  com a expansão (em cinza) “para trás” do bloco ativo  $i$ .

camada dos blocos, posicionamento de cada bloco no interior do contêiner). Cabe mencionar que, nos exemplos tratados neste trabalho (Seção 4), o tamanho da vizinhança observado variou entre poucas centenas e algumas centenas de milhares de movimentos.

### 3.2.1 Divisão e deslizamento de blocos

Sempre que um movimento é realizado, é comum o surgimento de espaços não aproveitados entre os blocos do padrão corrente. Antes do cálculo da nova vizinhança, estes espaços são aproveitados de duas maneiras: (i) dividindo cada bloco por vez o mais próximo possível das coordenadas centrais do contêiner ( $\frac{1}{2}L$ ,  $\frac{1}{2}W$ ,  $\frac{1}{2}H$ ) seguido de seu deslizamento para as extremidades mais próximas dos limites do contêiner; e (ii) adicionando-se tiras de espaços vazios ao volume remanescente de blocos passivos antes de sua reconfiguração. A divisão e deslizamento dos blocos (passo 3.4 na Figura 4) permitem que espaços não utilizados se concentrem no centro do contêiner, favorecendo a inclusão de mais caixas no carregamento em movimentos posteriores (Figura 8). A incorporação de tiras de espaços vazios ao volume remanescente de blocos passivos amplia também a chance de inclusão de um número maior de caixas durante a reconfiguração destes volumes.

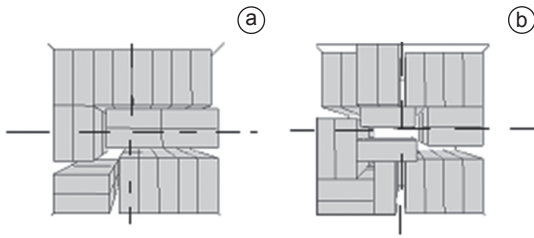
Note, entretanto, que a aplicação da divisão de bloco seguida de deslizamento não pode ser feita a cada iteração. Dado que são proibidos movimentos de blocos ativos cujas coordenadas estejam armazenadas na lista  $L$  (passo 3.2 na Figura 4), alterações constantes nas coordenadas dos blocos tornam ineficaz o efeito restritivo da lista. Assim, este procedimento

é aplicado com probabilidade dada por um parâmetro exógeno *divip*. Cabe ressaltar que procedimentos de deslizamento de blocos e incorporação de tiras de espaços vazios estão presentes em *PM2*, porém o algoritmo não prescreve a divisão de blocos.

### 3.2.2 União de blocos

Este recurso é proposto visando instâncias cujo carregamento contempla um grande número de caixas, em que é comum a formação de poucos blocos com muitas caixas. Ao serem divididos, estes blocos resultam em um conjunto de blocos menores com a mesma orientação e, conseqüentemente, é gerado um número razoável de movimentos que consistem de uma mera transferência de caixas entre blocos adjacentes de mesma orientação.

A união de blocos (passo 3.5 na Figura 4) visa minimizar este efeito danoso à exploração do espaço de soluções. Ela também potencializa o papel do movimento de mudança de orientação, uma vez que blocos maiores permitem um maior número de orientações factíveis e, portanto, maiores possibilidades de ganhos em número de caixas. Este procedimento é aplicado após a divisão de blocos e deslizamento, e consiste em agrupar cada par de blocos que exibam certas características em cada dimensão do problema. Como exemplo, considere os blocos  $i$  e  $j$  na Figura 9. Os blocos são agrupados na dimensão  $H$  por terem a mesma orientação de caixas e satisfazerem as seguintes restrições: (i) as coordenadas inicial e final do comprimento de  $i$  são respectivamente iguais às coordenadas inicial



**Figura 8.** Vista superior de um carregamento, antes e durante a aplicação dos procedimentos de divisão e deslizamento de blocos.

e final do comprimento de  $j$  ( $x_1^i = x_1^j, x_2^i = x_2^j$ ); (ii) as coordenadas inicial e final da largura de  $i$  são respectivamente iguais às coordenadas inicial e final da largura de  $j$  ( $y_1^i = y_1^j, y_2^i = y_2^j$ ); e (iii) a coordenada final da altura de  $i$  é igual à coordenada inicial da altura de  $j$  ( $z_2^i = z_1^j$ ).

A aplicação do procedimento de união de blocos é repetida para cada dois blocos distintos e para cada dimensão até que nenhuma alteração no número de blocos seja verificada. Note, entretanto, que a união de blocos pode gerar um número tão pequeno de blocos a ponto das restrições tabu rapidamente proibirem qualquer movimento disponível. O critério de aspiração que elimina o elemento mais antigo da lista tabu nestes casos (passo 3.2 da Figura 4) não é suficiente para garantir um fluxo suave das explorações caso a união de blocos seja realizada com muita frequência. Por esta razão, este último procedimento também é aplicado com uma probabilidade dada por um parâmetro exógeno *unip*.

### 3.2.3 Avaliação de movimentos candidatos

Similarmente ao algoritmo *PM2*, cada movimento é avaliado no passo 3.2 (Figura 4) segundo a seguinte função (Equação 1):

$$G = \begin{cases} G_0 + \alpha \frac{volume_h}{volume_n} + (1 - \alpha) pen \\ se \ G_0 = 0 \ e \ volume_n = 0 \\ G_0 + \alpha \frac{volume_h}{volume_n} + (1 - \alpha) \frac{volume_e - \sum_{j \in P} volume_j}{volume_e}, \\ caso \ contrário \end{cases} \quad (1)$$

Note que  $G$  é composta por três termos associados a diferentes critérios. O 1º critério ( $G_0$ ) corresponde ao ganho real decorrente do movimento, ou seja, à diferença entre o número de caixas no padrão candidato e no padrão corrente.  $G_0$  é inteiro, podendo assumir valores positivos (aumento no número de caixas),

negativos ou zero. No 2º critério  $\left( \frac{volume_h}{volume_n} \right)$ ,  $volume_n$  é o volume total dos novos blocos criados, e  $volume_h$  é a porção de  $volume_n$ , cujos blocos têm orientação diferente do bloco ativo. Este critério estimula a geração de padrões formados por blocos de diferentes orientações, intercalados entre si.

Finalmente, no 3º critério  $\left( \frac{volume_e - \sum_{j \in P} volume_j}{volume_e} \right)$ ,

$volume_e$  é volume total de expansão do bloco ativo e  $volume_s(j)$  é a porção de  $volume_e$  que sobrepõe um dado bloco passivo  $j$ . Este critério privilegia movimentos que modificam minimamente a estrutura do padrão corrente. Caso o movimento tenha  $G_0 = 0$  e  $volume_h = 0$ , substitui-se o 3º critério por um termo de penalização (*pen*). No caso de movimentos do tipo “te” (mudança de orientação de bloco seguido a um movimento de expansão), as contribuições do primeiro e segundo movimentos são somadas para cada um dos termos de  $G$ .

Deve-se ressaltar que  $G_0$  é geralmente preponderante na escolha do movimento, enquanto os outros critérios são relevantes em caso de empates no ganho real. Os critérios 2 e 3 são utilizados por incorporarem uma medida de “qualidade estratégica” do movimento. Note que ambos são ponderados por um fator  $\alpha \in [0, 1]$ , permitindo o controle da dominância de um sobre o outro.

### 3.2.4 Elementos de restrição tabu

Devido à boa qualidade da solução inicial, o padrão obtido é invariavelmente um ótimo local com relação à estrutura da vizinhança proposta. Diante disto torna-se necessária a utilização de mecanismos que restrinjam o retorno a padrões previamente obtidos para que a busca heurística se desenvolva além desse ponto. Para este fim, foram incorporados elementos restritivos de busca tabu simples (passo 3.2 na Figura 4), cujo papel é o de armazenar e restringir por um determinado número de iterações certas características de soluções já visitadas.

Assim como no algoritmo *PM2*, foi utilizada uma lista tabu de tamanho fixo  $t$  em que, após cada movimento, são armazenadas: (i) as coordenadas originais do bloco ativo; e (ii) as coordenadas dos novos blocos gerados. Proíbe-se, então, que blocos com estas coordenadas sejam ativos nas próximas  $t$  iterações.

## 4 Experimentos computacionais

O algoritmo *PM3* foi implementado em Borland Delphi 7, o que permitiu a elaboração de uma interface gráfica, na qual os padrões gerados podiam ser observados. Os experimentos foram conduzidos em



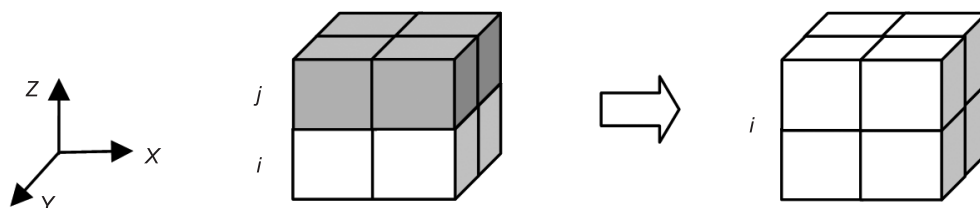


Figura 9. União dos blocos  $i$  (em branco) e  $j$  (em cinza).

um microcomputador Intel 2 Quad, 2.4 GHz, e 2.0 GB RAM, contemplando um total de 22 exemplos com até 396 caixas, extraídos dos trabalhos Han, Knott e Egbelu (1989) (conjunto 1), Liu e Hisao (1997) (conjunto 2) e Lins, Lins e Morabito (2002) (conjunto 3).

O tamanho de lista tabu (parâmetro  $t$ ) foi definido por meio de testes exaustivos com exemplos do conjunto 3. A partir destes resultados, e assim como no caso bidimensional, buscou-se obter uma função que fornecesse o tamanho  $t$  da lista tabu em função do número de caixas do padrão inicial ( $n_0$ ). Como as melhores soluções de cada instância foram obtidas com diferentes tamanhos de lista, a função foi obtida aplicando-se o Método de Diferenças Divididas de Newton (BURDEN; FAIRES, 2003). Este método foi utilizado por fornecer polinômios interpoladores de uma amostra de pontos em que parte destes pontos é previamente fixada. Dentre os pontos ( $n_0, t$ ) que forneceram as melhores soluções para o conjunto de instâncias tratado, foram testadas fixações de pontos associados a instâncias para as quais o número de soluções com os maiores números de caixas era pequeno. O polinômio que forneceu o maior número de valores de  $t$  que resultaram nestas soluções foi adotado nos experimentos, sendo dado por (Equação 2):

$$t = 7 + 0.5 (n_0 - 20) - 0,001602564 (n_0 - 20)(n_0 - 48) + 0,000002536 (n_0 - 20)(n_0 - 48)(n_0 - 72) \quad (2)$$

Os valores dos parâmetros  $\alpha$  e  $pen$  utilizados no algoritmo  $PM2$  foram mantidos em  $PM3$  ( $\alpha = 0,5$  e  $pen = -1$ ). Depois de experimentos preliminares com um conjunto selecionado de instâncias teste, os parâmetros  $divip$  e  $unip$  foram fixados em 0,3, o que significa que tanto a divisão como a união de blocos são aplicadas em 30% das iterações. O algoritmo é finalizado caso o padrão corrente satisfaça o critério de otimalidade do limitante superior trivial ou tenham sido realizadas 10.000 iterações sem melhoria (instâncias com  $n_0 \leq 50$ ) ou 50.000 iterações sem melhoria (demais instâncias). Em função dos elementos probabilísticos introduzidos com os procedimentos de divisão e união de blocos, foram realizadas cinco execuções para cada instância a partir do valor de  $t$  fornecido por (2).

#### 4.1 Conjunto 1

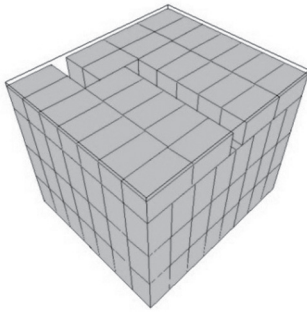
Este conjunto se limita a um único exemplo tratado em Han, Knott e Egbelu (1989), extraído de uma

situação real em *General Services Administration* (UNITED STATES, 1966). O contêiner tem dimensões  $(L, W, H) = (48, 42, 40)$ , enquanto cada caixa tem dimensões  $(l, w, h) = (11, 6, 6)$ . De acordo com os autores, a solução apresentada em *General Services Administration* fornece um empacotamento de 168 caixas, enquanto o método por eles proposto arranja 195 caixas. O algoritmo  $PM3$  forneceu uma solução de 196 caixas, requerendo para isso 0,2 segundos de tempo de processamento. A Figura 10 apresenta o padrão de carregamento obtido por  $PM3$ . Nas cinco execuções, todas as soluções apresentaram 196 caixas em um tempo médio de 0,4 segundos.

Nas seções que se seguem, os resultados com os conjuntos 2 e 3 são apresentados em tabelas. Para cada exemplo, as primeiras colunas descrevem as dimensões relevantes do contêiner e/ou caixas, e as demais colunas apresentam, para cada algoritmo da literatura considerado, o número de caixas das soluções incumbentes e os tempos computacionais (em segundos) requeridos para sua obtenção, quando reportados. Para o algoritmo  $PM3$ , os resultados com cada instância são apresentados em duas linhas. Na primeira linha de cada célula, é mostrado o número de caixas da melhor solução de todas as execuções, o número de caixas da solução inicial (entre parêntesis), e o tempo computacional (em segundos) requerido para obtenção da melhor solução. A segunda linha da célula apresenta o número médio de caixas das melhores soluções das cinco execuções e o tempo médio de obtenção destas soluções, oferecendo, portanto, informações sobre a robustez do método. Para cada algoritmo, a penúltima linha da tabela (TOTAL) mostra a soma do número de caixas nas soluções incumbentes de cada exemplo, enquanto a última linha apresenta o desvio percentual (DP) do total de caixas em relação ao maior total de caixas aqui reportado.

#### 4.2 Conjunto 2

O desempenho de  $PM3$  é comparado ao algoritmo de Liu e Hsiao (LH97) com base nos resultados de sete exemplos com restrições de peso sobre o chão do contêiner. Nestas instâncias as dimensões das caixas são fixas e iguais a  $(l, w, h) = (430, 295, 225)$  enquanto o contêiner tem dimensões  $L = 1200, W = 1000$  e dimensão  $H$  variável. Ou seja, as instâncias se



**Figura 10.** Vista isométrica do carregamento com 196 caixas para a instância  $(L, W, H, l, w, h) = (48, 42, 40, 11, 6, 6)$ .

diferenciam pela altura do contêiner. Cada caixa pesa 10 kg e o peso imposto sobre o chão do contêiner é de no máximo 1500 kg. Esta restrição não foi, entretanto, violada pelas soluções de *PM3* neste conjunto de exemplos, o que permitiu a comparação dos resultados dos dois algoritmos.

### 4.3 Conjunto 3

O desempenho de *PM3* foi inferido neste conjunto de exemplos comparando-se seus resultados com os dos quatro algoritmos (*2-D*, *lk*, *gui*, e *3-D*) considerados em Lins, Lins e Morabito (2002). O algoritmo *3-D* é o algoritmo recursivo proposto em Lins, Lins e Morabito (1999), com nível de recursividade igual a 2 ou 3 e sem fixação da orientação vertical das caixas. Sendo uma extensão da heurística de Morabito e Morales (1998, 1999), suas soluções correspondem a padrões de carregamento não guilhotinados de 1ª ordem ótimos. O algoritmo *gui* é o algoritmo recursivo com nível de recursividade igual a 2 ou 3, sem fixação da orientação vertical das caixas e limitado a padrões guilhotinados. O algoritmo *lk* admite que a orientação vertical das caixas não é fixada e resolve o problema dividindo o padrão de carregamento em camadas, em que cada camada é um padrão bidimensional com caixas de mesma altura. O algoritmo *2-D* admite que a orientação vertical das caixas seja fixada e consiste em resolver o problema bidimensional e adicionar camadas idênticas enquanto não se ultrapasse o teto do contêiner. Para três opções de empilhamento das camadas no contêiner, *2-D* resolve problemas da mochila para obter o número de camadas de cada tipo e seleciona o melhor padrão tridimensional resultante.

## 5 Análise dos resultados

Os resultados com os conjuntos 1 (Seção 4.1) e 2 (Seção 4.2 e Tabela 1) revelam uma clara dominância do algoritmo *PM3* em relação ao método da literatura considerado em cada um destes conjuntos. Ganhos em número de caixas variam entre 0,5 e 4,4%, com pouca variabilidade na qualidade das soluções incumbentes das cinco execuções e requerendo tempos

computacionais reduzidos. O papel da busca tabu em ambos os conjuntos é relevante, apresentando incrementos ao número de caixas do padrão inicial entre 3,9 e 17,5% em cinco das oito instâncias tratadas.

No que diz respeito ao conjunto 3 (Seção 4.3), a análise do desempenho de *PM3* pode ser considerada mais representativa do que a dos conjuntos anteriores, uma vez que o melhor algoritmo da literatura considerado (*3-D*) tem a garantia de prover padrões ótimos não guilhotinados de 1ª ordem sendo, portanto, altamente eficiente. Além disso, o conjunto apresenta instâncias com um largo espectro de tamanhos de contêineres e caixas, ou seja, bastante diversificado.

Dentre os cinco algoritmos, *3-D* e *PM3* apresentam os melhores resultados em número de caixas em todos os 14 exemplos deste último conjunto (veja Tabela 2). O desempenho do algoritmo *PM3* é inferior no caso da instância 10 (uma caixa a menos), porém superior no caso da instância 11 (uma caixa a mais). A aplicação de *PM3* permitiu, portanto, demonstrar que a instância 11 possui um carregamento com padrão não guilhotinado de ordem superior de melhor qualidade (possivelmente ótimo) que o carregamento com padrão ótimo não guilhotinado de 1ª ordem (veja Figura 11c).

Apesar desse resultado interessante, no trabalho de Junqueira (2009) em carregamento de contêineres com restrições de estabilidade e peso máximo, foram realizados experimentos desconsiderando tais restrições e reportada a resolução ótima (após várias horas de execução) das instâncias 1-3 com o pacote computacional GAMS/CPLEX 11. Enquanto o número de caixas ótimo da instância 3 é igual ao obtido por *3-D* e *PM3* (29 caixas), as instâncias 1 e 2 apresentam um número ótimo de 27 caixas e, portanto, correspondentes a padrões não guilhotinados de ordem superior. Estes resultados indicam a necessidade de melhorias adicionais ao método aqui proposto.

Em relação à robustez do método, para algumas instâncias, observa-se grande dificuldade em se obter o maior número de caixas nas cinco execuções realizadas. Em particular, para as instâncias 4-6 e 12, o maior número de caixas foi obtido em apenas uma execução, com diferenças entre a melhor e a pior solução de até duas caixas. As instâncias 1, 8 e 11, por outro lado, obtiveram soluções com o maior número de caixas em todas as execuções.

Enquanto a frequência de convergência a soluções de alta qualidade parece depender da instância considerada, observou-se que o número de caixas da solução incumbente é praticamente insensível ao padrão de carregamento inicial. Em experimentos adicionais com os mesmos valores de parâmetros, porém partindo-se de um padrão homogêneo e de pior qualidade que o fornecido pela heurística de blocos, os valores das melhores soluções foram mantidos em 13 dos 14 exemplos (a única exceção foi

**Tabela 1.** Resultados computacionais com exemplos do conjunto 2.

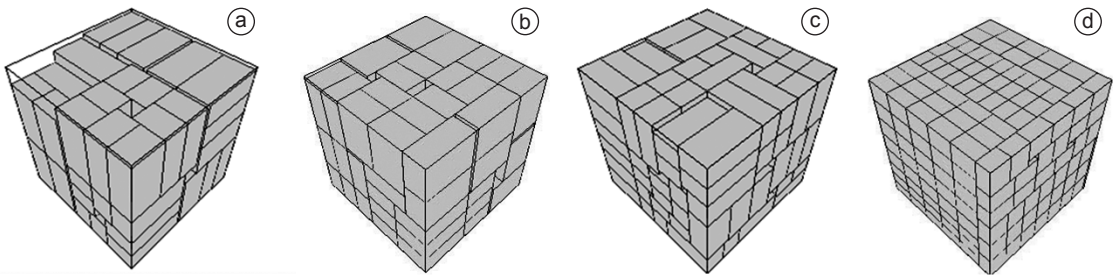
Instância	Dimensão <i>H</i> do contêiner	Algoritmo		
		LH97*		PM3+
1	1200	<b>44</b>	<b>44</b> (40) 44	0 0,2
2	1250	45	<b>47</b> (40) 46,4	13,9 25,3
3	1300	<b>51</b>	<b>51</b> (51) 51,0	0 0
4	1350	<b>51</b>	<b>51</b> (51) 51,0	0 0
5	1400	52	<b>53</b> (51) 53,0	0,6 8,9
6	1450	54	<b>55</b> (51) 55,0	0,6 9,1
7	1500	54	<b>56</b> (51) 55,8	3,9 64,3
<b>Total</b>		351		357
<b>DP</b>		1,7%		0,0%

Ambiente computacional: \*Não informado. \*Intel 2 Quad, 2,4 GHz, e 2.0 GB RAM.

**Tabela 2.** Resultados computacionais com exemplos do conjunto 3.

Instância	Dimensões das caixas			Algoritmo						
	<i>l</i>	<i>w</i>	<i>h</i>	2D*	lk*	gui*	3-d*		PM3+	
1	13	14	23	18	24	22	<b>26</b>	2	<b>26</b> (24) 26,0	1,3 8,9
2	17	20	12	16	22	22	<b>26</b>	1	<b>26</b> (20) 25,4	11,8 9,6
3	11	22	15	24	27	27	<b>29</b>	1	<b>29</b> (26) 28,6	44,0 32,6
4	17	21	6	32	47	46	<b>54</b>	148	<b>54</b> (42) 53,2	95,7 59,9
5	22	8	11	48	54	54	<b>58</b>	17	<b>58</b> (52) 57,2	84,1 63,1
6	18	21	5	40	54	54	<b>57</b>	523	<b>57</b> (48) 56,2	12,0 4,2
7	9	11	19	48	60	57	<b>62</b>	81	<b>62</b> (50) 61,4	4,7 32,2
8	13	8	18	48	<b>64</b>	60	<b>64</b>	19	<b>64</b> (48) 64,0	0,2 9,1
9	9	16	11	64	74	72	<b>75</b>	40	<b>75</b> (72) 74,6	25,7 38,8
10	13	15	7	63	79	79	<b>84</b>	347	83 (72) 82,8	144,6 240,5
11	7	9	18	78	102	100	106	163	<b>107</b> (90) 107,0	52,4 295,9
12	7	6	18	118	158	156	<b>161</b>	1977	<b>161</b> (144) 159,8	4439,8 1549,0
13	7	9	11	156	<b>176</b>	170	<b>176</b>	1419	<b>176</b> (168) 174,6	1644,5 1304,9
14	9	7	5	390	394	394	<b>396</b>	377	<b>396</b> (385) 395,6	223,2 8171,6
<b>Total</b>				1143	1335	1313		1374		1374
<b>DP</b>				16,8%	2,8%	4,4%		0,0%		0,0%

Ambiente computacional: \*Pentium II, 400 MHz. \*Intel 2 Quad, 2,4 GHz, e 2.0 GB RAM.



**Figura 11.** Vistas isométricas de carregamentos obtidos pelo algoritmo *PM3* em exemplos do conjunto 3: (a) Instância 5; (b) Instância 8; (c) Instância 11; e (d) Instância 14.

verificada para a instância 4, com uma caixa a menos), sem variações relevantes do tempo computacional requerido. Esta aparente independência é facilmente justificada pelo fato da estrutura de vizinhança ser capaz de alterar de forma significativa o padrão inicial em poucas iterações.

A busca tabu, por sua vez, mostrou-se fundamental na obtenção dos resultados aqui reportados, sendo responsável por incrementos entre 2,9 e 33,3% em relação ao número de caixas da solução inicial. Cabe ressaltar que o número de caixas obtido com a instância 14 (396 caixas) é ótimo, uma vez que satisfaz o critério de otimalidade do limitante superior trivial.

Devido aos diferentes ambientes computacionais, os tempos de processamento de *3-D* e *PM3* (Tabela 2) não são diretamente comparáveis, apesar das diferenças tecnológicas entre estes ambientes indicarem que *PM3* demanda maior tempo por execução que *3-D*. Em média, requereu-se 485 segundos para obtenção das melhores soluções, e 844,3 segundos para obtenção da incumbente de cada execução.

A Figura 11 ilustra vistas isométricas das melhores soluções obtidas com quatro exemplos do conjunto 3.

## 6 Conclusões e próximos passos

Este trabalho apresentou uma extensão do algoritmo de busca tabu de Pureza e Morabito, originalmente proposto para o problema do carregamento do paletes, com vistas ao tratamento do problema de carregamento de contêineres com caixas idênticas. Modificações foram realizadas para adequação da diferença de dimensionalidade entre os dois problemas, e novos tipos de movimentos e procedimentos foram propostos. Resultados com 22 exemplos da literatura mostraram que a abordagem é promissora, o que motiva a continuidade de seu desenvolvimento.

Os próximos passos desta pesquisa consistem na investigação de novas vizinhanças de busca e de um melhor aproveitamento dos espaços vazios. Em particular, serão consideradas configurações alternativas do volume remanescente de blocos passivos, e procedimentos mais sofisticados de divisão e deslizamento de blocos que concentrem o

maior volume possível de espaços vazios, o que não necessariamente ocorre na região central do contêiner. Acredita-se que melhorias nestes aspectos tragam ganhos significativos na qualidade das soluções geradas e na robustez do método. Paralelamente, já vem sendo implementada uma extensão do algoritmo *PM3* para o caso mais geral em que a carga é heterogênea. Finalmente, pretende-se endereçar questões de estabilidade da carga na geração das soluções, a fim de se promover uma maior praticabilidade dos padrões.

## Agradecimentos

Os autores agradecem à CAPES (DS) e ao CNPq (processo 303001/2009-7) o apoio na realização desta pesquisa, assim como aos valiosos comentários e sugestões dos dois revisores anônimos.

## Referências

- ABDOU, G.; YANG, M. A systematic approach for the three dimensional palletization problem. **International Journal of Production Research**, v. 32, n. 10, p. 2381-2394, 1994. <http://dx.doi.org/10.1080/00207549408957074>
- ARAÚJO, O. C. B. **Problemas de corte e empacotamento tridimensional e integração com roteamento de veículos**. 2006. Tese (Doutorado em Engenharia Elétrica)-Faculdade de Engenharia e de Computação, Universidade Estadual de Campinas, Campinas, 2006.
- ARAÚJO, O. C. B.; ARMENTANO, V. A. A multi start random constructive heuristic for the container loading problem. **Pesquisa Operacional**, v. 27, n. 2, p. 311-331, 2007.
- ARENALES, M.; MORABITO, R. An and/or-graph approach to the solution of two-dimensional non-guillotine cutting Problems. **European Journal of Operational Research**, v. 84, p. 599-617, 1995. [http://dx.doi.org/10.1016/0377-2217\(95\)00026-M](http://dx.doi.org/10.1016/0377-2217(95)00026-M)
- BALLOU, R. H. **Business Logistics/Supply Chain Management**. 5th ed. Prentice Hall, 2004.
- BIRGIN, E. G.; MARTÍNEZ, J. M.; RONCONI, D. P. Optimizing the packing of cylinders into a rectangular container: A nonlinear approach. **European Journal of Operational Research**, v. 160, p. 19-33, 2005. <http://dx.doi.org/10.1016/j.ejor.2003.06.018>

- BISCHOFF, E. E. Three-dimensional packing of items with limited load bearing strength. **European Journal of Operational Research**, v. 168, p. 952-966, 2006. <http://dx.doi.org/10.1016/j.ejor.2004.04.037>
- BISCHOFF, E. E.; DOWSLAND, W. An application of the micro to product design and distribution. **Journal of the Operational Research Society**, v. 33, p. 271-280, 1982.
- BISCHOFF, E. E.; MARRIOTT, M. D. A comparative evaluation of heuristics for container loading. **European Journal of Operational Research**, v. 44, p. 267-276, 1990. [http://dx.doi.org/10.1016/0377-2217\(90\)90362-F](http://dx.doi.org/10.1016/0377-2217(90)90362-F)
- BISCHOFF, E. E.; RATCLIFF, M. S. W. Loading multiple pallets. **Journal of the Operational Research Society**, v. 46, p. 1322-1336, 1995a.
- BISCHOFF, E. E.; RATCLIFF, M. S. W. Issues in the development of approaches to container loading. **Omega**, v. 23, n. 4, p. 377-390, 1995b. [http://dx.doi.org/10.1016/0305-0483\(95\)00015-G](http://dx.doi.org/10.1016/0305-0483(95)00015-G)
- BURDEN, R. L.; FAIRES, J. D. **Análise Numérica**. São Paulo: Thomson Learning (Pioneira), 2003.
- CECILIO, F. O. **Heurísticas para o problema de carregamento de carga dentro de contêineres**. 2003. Dissertação (Mestrado em Engenharia de Produção)-Universidade Federal de São Carlos, São Carlos, 2003.
- CECILIO, F. O.; MORABITO, R. Refinamentos na heurística de George e Robinson para o problema de carregamento de caixas dentro de contêineres. **Transportes**, v. 12, p. 32-45, 2004.
- CHEN, C. S.; LEE, S. M.; SHEN, Q. S. An analytical model for the container loading problem. **European Journal of Operational Research**, v. 80, p. 68-76, 1995. [http://dx.doi.org/10.1016/0377-2217\(94\)00002-T](http://dx.doi.org/10.1016/0377-2217(94)00002-T)
- CHOONG, S. T.; COLE, M. H.; KUTANOGLU, E. Empty container management for intermodal transportation networks. **Transportation Research Part E**, v. 38, p. 423-438, 2002. [http://dx.doi.org/10.1016/S1366-5545\(02\)00018-2](http://dx.doi.org/10.1016/S1366-5545(02)00018-2)
- DAVIES, A. P.; BISCHOFF, E. E. Weight distribution considerations in container loading. **European Journal of Operational Research**, v. 114, p. 509-527, 1999. [http://dx.doi.org/10.1016/S0377-2217\(98\)00139-8](http://dx.doi.org/10.1016/S0377-2217(98)00139-8)
- DYCKHOFF, H. A typology of cutting and packing problems. **European Journal of Operational Research**, v. 44, p. 145-159, 1990. [http://dx.doi.org/10.1016/0377-2217\(90\)90350-K](http://dx.doi.org/10.1016/0377-2217(90)90350-K)
- DYCKHOFF, H.; FINKE, U. **Cutting and packing in production and distribution: typology and bibliography**. Heidelberg: Springer-Verlag, 1992. <http://dx.doi.org/10.1007/978-3-642-58165-6>
- ELEY, M. Solving container loading problems by block arrangement. **European Journal of Operational Research**, v. 141, n. 2, p. 393-409, 2002. [http://dx.doi.org/10.1016/S0377-2217\(02\)00133-9](http://dx.doi.org/10.1016/S0377-2217(02)00133-9)
- GEHRING, H.; BORTFELDT, A. A genetic algorithm for solving the container loading problem. **International Transactions in Operations Research**, v. 4, n. 5-6, p. 401-418, 1997. <http://dx.doi.org/10.1111/j.1475-3995.1997.tb00095.x>
- GEHRING, H.; MENSCHNER, K.; MEYER, M. A computer-based heuristic for packing pooled shipment containers. **European Journal of Operational Research**, v. 44, p. 277-288, 1990. [http://dx.doi.org/10.1016/0377-2217\(90\)90363-G](http://dx.doi.org/10.1016/0377-2217(90)90363-G)
- GENDREAU, M. et al. A tabu search algorithm for a routing and container loading problem. **Transportation Science**, v. 40, n. 3, p. 342-350, 2006. <http://dx.doi.org/10.1287/trsc.1050.0145>
- GEORGE, J. A. A method for solving container packing for a single size of box. **Journal of Operational Research Society**, v. 43, n. 4, p.307-312, 1992.
- GEORGE, J. A.; ROBINSON, D. F. A heuristic for packing boxes into a container. **Computers and Operations Research**, v. 7, p. 147-156, 1980. [http://dx.doi.org/10.1016/0305-0548\(80\)90001-5](http://dx.doi.org/10.1016/0305-0548(80)90001-5)
- HAN, C. P.; KNOTT, K.; EGBELU, P. J. A heuristic approach to the three-dimensional cargo loading problem. **International Journal of Production Research**, v. 27, n. 5, p. 757-774, 1989. <http://dx.doi.org/10.1080/00207548908942585>
- JUNQUEIRA, L. **Modelos de programação matemática para problemas de carregamento de caixas dentro de contêineres**. 2009. Dissertação (Mestrado em Engenharia de Produção)-Universidade Federal de São Carlos, São Carlos, 2009.
- JUNQUEIRA, L.; MORABITO, R.; YAMASHITA, D. S. Three-dimensional container loading models with cargo stability and load bearing constraints. **Computers and Operations Research**, v. 39, n. 1, p. 74-85, 2010. <http://dx.doi.org/10.1016/j.cor.2010.07.017>
- LINS, L.; LINS, S.; MORABITO, R. A 9-fold partition heuristic for packing boxes into a container. **Investigación Operativa**, v. 7, n. 3, p. 69-82, 1999.
- LINS, L.; LINS, S.; MORABITO, R. An  $n$ -tet graph approach for non-guillotine packing of  $n$ -dimensional boxes into an  $n$ -container. **European Journal of Operational Research**, v. 141, p. 421-439, 2002. [http://dx.doi.org/10.1016/S0377-2217\(02\)00135-2](http://dx.doi.org/10.1016/S0377-2217(02)00135-2)
- LIU, F. F.; HSIAO, C. A three-dimensional pallet loading method for single-size boxes. **Journal of the Operational Research Society**, v. 48, p. 726-735, 1997.
- MAGEE, J. F. **Logística industrial: análise e administração dos sistemas de suprimento e distribuição**. São Paulo: Pioneira, 1977.
- MARTELLO, S.; PISINGER, D.; VIGO, D. The three dimensional bin packing problem. **Operations Research**, v. 48, p. 256-267, 2000. <http://dx.doi.org/10.1287/opre.48.2.256.12386>
- MORABITO, R. **Uma abordagem em grafo-E/OU para o problema do empacotamento: aplicação ao problema do carregamento de paletes e contêineres**. 1992. Tese (Doutorado em Engenharia de Transportes)-Escola de Engenharia de São Carlos, Universidade de São Paulo, São Carlos, 1992.
- MORABITO, R.; ARENALES, M. An And/Or-graph approach to the container loading problem. **International Transactions in Operations Research**, v. 1, n. 1, p. 59-73, 1994. [http://dx.doi.org/10.1016/0969-6016\(94\)90046-9](http://dx.doi.org/10.1016/0969-6016(94)90046-9)
- MORABITO, R.; ARENALES, M. Abordagens para o problema do carregamento de contêineres. **Pesquisa Operacional**, v. 17, n. 1, p. 29-56, 1997.

- MORABITO, R.; MORALES, S. A simple and effective recursive procedure for the manufacturer's pallet loading problem. **Journal of the Operational Research Society**, v. 49, p. 819-828, 1998. <http://dx.doi.org/10.1038/sj.jors.2600588>
- MORABITO, R.; MORALES, S. Errata: A simple and effective recursive procedure for the manufacturers' pallet loading problem. **Journal of the Operational Research Society**, v. 50, p. 876, 1999.
- MORABITO, R.; PUREZA, V. **Loading optimization**. In: ERDOGDU, F. (Ed.). **Optimization in Food Engineering**. Taylor & Francis CRC Press, 2008. <http://dx.doi.org/10.1201/9781420061420.ch29>
- MOURA, A.; OLIVEIRA, J. F. A GRASP approach to the container-loading problem. **IEEE Intelligent Systems**, v. 4, p. 50-57, 2005. <http://dx.doi.org/10.1109/MIS.2005.57>
- NELISSEN, J. **Solving the pallet loading problem more efficiently**. Aachen: Graduiertenkolleg Informatik und Technik, 1994. Working Paper.
- PUREZA, V.; MORABITO, R. Some experiments with a simple tabu search approach for the manufacturer's pallet loading problem. **Computers & Operations Research**, v. 33, n. 3, p. 804-819, 2006. <http://dx.doi.org/10.1016/j.cor.2004.08.009>
- PISINGER, D. Heuristics for the container loading problem. **European Journal of Operational Research**, v. 141, n. 2, p. 382-392, 2002. [http://dx.doi.org/10.1016/S0377-2217\(02\)00132-7](http://dx.doi.org/10.1016/S0377-2217(02)00132-7)
- RATCLIFF, M. S. W.; BISCHOFF, E. E. Allowing for weight considerations in container loading. **OR Spektrum**, v. 20, n. 1, p. 65-71, 1998. <http://dx.doi.org/10.1007/BF01545534>
- SCHEITHAUER, G. Algorithms for the container loading problem. In: OPERATIONS RESEARCH, 1991, Heidelberg. **Proceedings...** Heidelberg: Springer-Verlag Berlin, 1992.
- SCHEITHAUER, G.; TERNO, J. The G4-heuristic for the pallet loading problem. **Journal of the Operational Research**, v. 84, p. 662-680, 1996.
- SMITH, A.; DE CANI, P. An algorithm to optimize the layout of boxes in pallets. **Journal of the Operational Research Society**, v. 31, p. 573-578, 1980.
- STEUDEL, H. J. Generating pallet loading patterns: a special case of the two dimensional cutting stock problem. **Management Science**, v. 10, p. 997-1004, 1979. <http://dx.doi.org/10.1287/mnsc.25.10.997>
- UNITED STATES. General Services Administration. **Warehouse operations**. Washington: U.S. Government Printing Office, 1966.
- WÄSCHER, G.; HAUBNER, H.; SCHUMANN, H. An improved typology of cutting and packing problems. **European Journal of Operational Research**, v. 183, p. 1109-1130, 2007. <http://dx.doi.org/10.1016/j.ejor.2005.12.047>