

Refinamentos na heurística de George e Robinson para o problema do carregamento de caixas dentro de contêineres

F. O. Cecilio¹ e R. Morabito²

Departamento de Engenharia de Produção, Universidade Federal de São Carlos, São Paulo, Brasil.

Artigo Recebido: 02/12/2003 - Aprovado: 05/02/2004

RESUMO: Neste trabalho apresentamos refinamentos da heurística de George e Robinson (1980) para o problema de arranjar o maior volume possível de caixas de tamanhos (l_i, w_i, h_i) e quantidades $b_i, i = 1 \dots, m$, dentro de um contêiner de tamanho (L, W, H) . Esta heurística é conhecida pela sua simplicidade e facilidade de implementação em situações reais. Os refinamentos propostos também são facilmente implementados em computador, são flexíveis para incluir restrições práticas adicionais, são conservadores quanto à geração de carregamentos estáveis e obtêm boas soluções com pouco esforço computacional. Resolvemos diversos exemplos aleatórios, exemplos da literatura de corte e empacotamento e exemplos reais para ilustrar o desempenho dos refinamentos, em comparação à heurística original de George e Robinson e outros métodos.

ABSTRACT: In this paper we present refinements of George and Robinson (1980)'s heuristic for the problem of arranging the maximum possible volume of boxes of sizes (l_i, w_i, h_i) and quantities $b_i, i = 1, \dots, m$, inside a container of size (L, W, H) . The refinements proposed are easily implemented in a computer, they are flexible for inclusion of additional practical constraints, they are conservative with respect to the generation of stable cargo loadings and they obtain good solutions requiring low computational efforts. We solved different randomly generated examples, examples of the cutting and packing literature and actual examples in order to illustrate the performances of the refinements, compared to the original George and Robinson's heuristic and other methods.

1 INTRODUÇÃO

O problema do carregamento de contêineres (*container loading problem*) consiste em como carregar caixas de vários tamanhos dentro de contêineres, de maneira a otimizar um critério, por exemplo, maximizar o aproveitamento do espaço disponível dos contêineres. Tal problema pode ser visto como um caso especial dos problemas de corte e empacotamento (Dyckhoff e Finke, 1992; Morabito e Arenales, 1992). Nossa motivação em estudá-lo deve-se à sua importância nas atividades logísticas de movimentação, armazenagem e transporte de produtos, por exemplo, na redução de custos destas atividades, na redução do tempo de carregamento e descarregamento de contêineres e caminhões e na maior proteção dada à carga. Para Bischoff e Marriott (1990), na prática o problema aparece em dois casos importantes: (i) quando uma combinação de contêineres deve ser escolhida para transportar uma dada carga, e (ii) quando o maior volume de uma dada carga deve ser escolhido para ser transportado em um único contêiner (este caso é classificado como 3/B/O/. de acordo com a tipologia de Dyckhoff, 1990).

Além do arranjo geométrico das caixas dentro de cada contêiner, pode haver outras restrições, por exemplo: estabilidade do carregamento; orientação das caixas dentro do contêiner; número de caixas que podem ser empilhadas umas sobre as outras; grupos de itens que devem ficar juntos dentro do contêiner (por terem características semelhantes ou por serem descarregados juntos); grupos de itens que devem permanecer separados dentro do contêiner (secos, resfriados, produtos químicos); tecnologia dos equipamentos de manuseio (por exemplo, empilhadeira); limite de peso suportado pelo contêiner; distribuição de peso dentro do contêiner, etc.

Diversos trabalhos podem ser encontrados na literatura tratando diferentes situações de carregamento de contêineres, tais como: caixas iguais (Han *et al.*, 1989; George, 1992; Herbert e Dowsland, 1996; Liu e Hsiao, 1997; Lins *et al.*, 1999, 2002), caixas diferentes e um único contêiner (Abdou e Yang, 1994; Morabito e Arenales, 1994, 1997; Abdou e Arghavani, 1997; Gehring e Bortfeldt, 1997) e caixas diferentes e vários contêineres (Mohanty *et al.*, 1994; Bischoff e Ratcliff, 1995a; Martello *et al.*, 2000). Algumas teses de doutorado já foram defendidas no Brasil estudando o

empacotamento tridimensional, entre elas, Morabito (1992), Miyazawa (1997) e Silva (2002).

Problemas de empacotamento tridimensional são em geral NP-difíceis e frequentemente não é possível encontrar soluções exatas dentro de um limite de tempo razoável (Dowland e Dowland, 1992). Assim, devido à dificuldade com a complexidade dos métodos exatos, a maioria das abordagens propostas na literatura refere-se a métodos aproximados (heurísticas, metaheurísticas, algoritmos aproximativos, etc.), conforme as referências citadas acima e várias outras (George e Robinson, 1980; Han *et al.*, 1989; Gehring *et al.*, 1990; Haessler e Talbot, 1990; George, 1992; Correia *et al.*, 1992; Tsai *et al.*, 1993; Abdou e Yang, 1994; Mohanty *et al.*, 1994; Morabito e Arenales, 1994; Bischoff *et al.*, 1995; Chen *et al.*, 1995; Abdou e Arghavani, 1997; Gehring e Bortfeldt, 1997; Miyazawa e Wakabayashi, 1997, 2000; Xue e Lai, 1997; Chien e Wu, 1998, 1999; Davies e Bischoff, 1999; Pisinger, 1999, Lins *et al.*, 1999, 2002; Martello *et al.*, 2000; Morabito *et al.*, 2000; Beasley, 2001; Bortfeldt e Gehring, 2001; Martins, 2002; Silva *et al.*, 2003). Métodos aproximados são geralmente bem mais rápidos que métodos exatos, o que é extremamente importante na prática devido ao grande número de pedidos processados por dia e a necessidade de resposta imediata durante o processamento do pedido.

Nesse trabalho estamos interessados no caso (ii) mencionado acima, ou seja, em como arranjar o maior volume possível de caixas de dimensões (l_i, w_i, h_i) e quantidade $b_i, i = 1, \dots, m$, dentro de um único contêiner de dimensões (L, W, H) . Este problema pode ser modelado matematicamente estendendo para o caso tridimensional alguns modelos conhecidos da literatura para o caso bidimensional; por exemplo, como um modelo de programação inteira 0-1 (Beasley, 1985; Hadjiconstantinou e Christofides, 1995), como um modelo de programação inteira mista 0-1 (Tsai *et al.*, 1993; Chen *et al.*, 1995; Martins, 2002), ou como um modelo de programação não linear (Beasley, 2001). Tais modelos estão revisados em Cecilio (2003), porém, nenhum deles garante carregamentos estáveis. De fato, a maioria dos estudos tratando empacotamento tridimensional não considera a estabilidade do carregamento.

No presente estudo inicialmente propomos um simples refinamento da conhecida heurística de George e Robinson (1980) (aqui chamada simplesmente de heurística G&R), que considera as restrições geométricas e de estabilidade do carregamento. A heurística G&R é conhecida pela sua simplicidade, flexibilidade e facilidade de implantação nas situações reais, em relação a outros métodos da literatura. Mostramos que, ao contrário da heurística G&R, este refinamento é capaz de encontrar a solução ótima do exemplo real analisado em George e Robinson (1980). Propomos também duas novas versões da heurística G&R: a versão *Arranjo* (no contêiner) e a versão (arranjo na) *Camada*. A partir da heurística G&R, do

refinamento proposto e das duas novas versões, definimos 5 novos métodos de solução para o problema de carregamento de contêineres. Todos estes métodos são facilmente implementados em computador, são flexíveis para incluir restrições práticas adicionais, são conservadores quanto à geração de carregamentos estáveis, e obtêm boas soluções com pouco esforço computacional, o que viabiliza suas aplicações nas atividades diárias de carregamento de contêineres e caminhões. Por fim, testamos os métodos propostos com exemplos aleatórios, alguns exemplos da literatura e alguns exemplos reais de carregamento de contêineres, e os resultados obtidos são analisados e comparados com a heurística de G&R e outros métodos.

O artigo está organizado da seguinte maneira: na seção 2 revisamos brevemente a heurística G&R, na seção 3 estendemos a heurística G&R para tratar orientação vertical fixa das caixas, na seção 4 apresentamos o refinamento da heurística G&R e na seção 5 apresentamos as duas novas versões da heurística G&R (versão Arranjo e versão Camada) e os métodos de solução aqui propostos para o problema. Na seção 6 apresentamos os resultados computacionais obtidos nos testes com exemplos da literatura, exemplos com dados aleatórios e exemplos reais de carregamento de contêineres. Finalmente, na seção 7 fazemos considerações finais e discutimos perspectivas para pesquisa futura.

2 HEURÍSTICA G&R

George e Robinson (1980) estudaram o caso (ii) do problema do carregamento de contêiner e desenvolveram um procedimento heurístico para ordenar caixas de tamanhos diferentes em um contêiner. O procedimento preenche o contêiner construindo camadas ao longo do seu comprimento L , e combinando espaços vazios entre camadas para aumentar a utilização de espaço. Não são impostas restrições quanto ao número de caixas que podem ser empilhadas umas sobre as outras, nem sobre qual face das caixas deve ficar voltada para cima, embora a heurística possa ser adaptada para tratá-las. Na prática, é desejável que caixas de um mesmo tipo (mesmas dimensões) sejam colocadas próximas umas das outras, o que em parte também é levado em conta pela heurística.

A seguir revisamos resumidamente a heurística G&R; para mais detalhes o leitor pode consultar o artigo original e Cecilio (2003). O método carrega as caixas em pilhas compostas de caixas do mesmo tipo. Cada vez que um tipo é utilizado, a ele é dado prioridade sobre os demais. Desta forma, as caixas iguais tenderão a ser empilhadas próximas umas das outras. O contêiner é carregado do seu fundo para sua entrada, tentando sempre manter uma superfície plana de trabalho. Note que pode não ser possível colocar

todas as caixas no contêiner, mesmo que o volume total das caixas ($\sum_{i=1, \dots, m} b_i l_i w_i h_i$) seja menor que o volume do contêiner (LWH), pois desperdícios de espaço podem ser inevitáveis devido ao arranjo geométrico das caixas. O volume não ocupado por caixas é considerado volume perdido.

A heurística baseia-se no preenchimento do contêiner, camada por camada. Uma camada é uma seção do comprimento do contêiner na sua completa altura H e largura W , como mostra a Figura 1. O comprimento da camada é determinado pelo comprimento da primeira caixa colocada nela. Uma nova camada não deve ser iniciada até que a camada anterior esteja totalmente preenchida. Para que a camada tenha um tamanho viável (nem tão grande, nem tão pequeno), algumas restrições são impostas na escolha da primeira caixa. Outros tipos de caixas são considerados apenas para preencher espaços deixados dentro da camada (quando não é possível preenchê-la totalmente com caixas de um mesmo tipo), ou espaços combinados com espaços que sobraram em camadas anteriores.

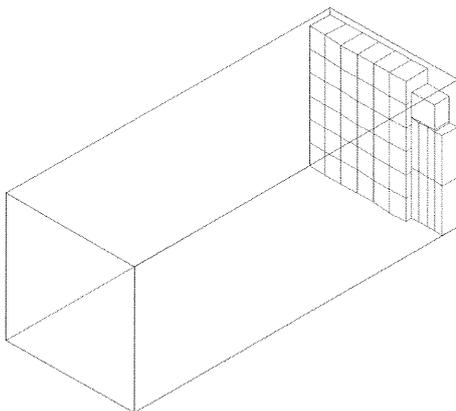


Figura 1. Primeira camada do contêiner.

Para preencher um espaço escolhemos um tipo de caixa e preenchamos com tantas colunas completas quanto possíveis. Para iniciar uma nova camada, a caixa é selecionada de acordo com o fluxograma da Figura 2, que define a dimensão de comprimento da camada. Um tipo de caixa que já tenha sido utilizado ao longo do procedimento é classificado como *open* (aberto). Considera-se que seja preferível utilizar um tipo de caixa aberto a um não aberto, para que caixas iguais sejam colocadas mais próximas umas das outras. Deste modo, se existirem tipos abertos, aquele com maior prioridade (segundo um certo critério de prioridades) é usado para iniciar a camada. Caso não existam tipos abertos, seleciona-se o tipo com maior prioridade entre todas as possíveis caixas.

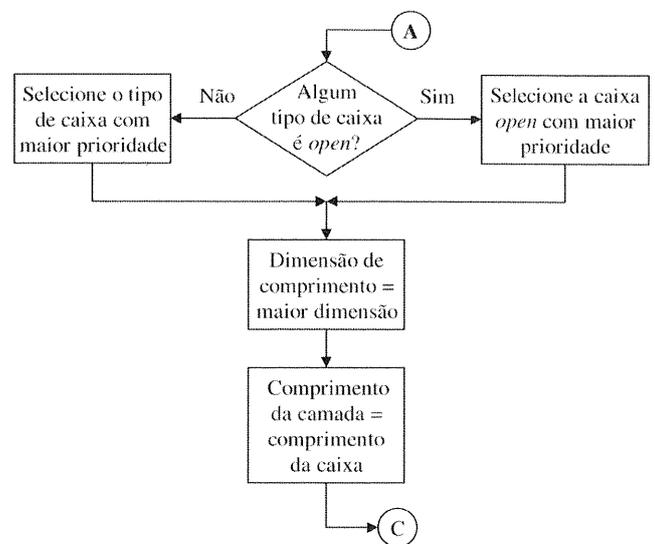


Figura 2. Escolha do tipo de caixa para uma nova camada

O critério de prioridades é resumidamente apresentado a seguir.

1. Caixa com a maior das menores dimensões (a justificativa é que pode ser mais difícil arranjá-la depois);
2. Caixa com a maior quantidade disponível (a justificativa é que uma maior quantidade pode preencher uma maior parte da camada);
3. Caixa com a maior dimensão (isto antecipa o arranjo de caixas de dimensões desfavoráveis).

No caso de empate no primeiro critério, o segundo critério é utilizado. Caso ocorra empate no segundo critério, utiliza-se o terceiro. Ainda no fluxograma da Figura 2, tendo determinado o tipo de caixa a ser utilizado, escolhemos agora qual das três dimensões da caixa será a dimensão de comprimento da camada. Para esta dimensão é escolhida a maior dimensão da caixa. Assim, determina-se o comprimento da camada, já que ele é igual ao comprimento da primeira caixa nela colocada. Após esses passos, vamos para o fluxograma da Figura 4 explicado adiante. Quando o comprimento do contêiner que ainda está vazio torna-se menor que o comprimento da caixa escolhida não se introduz uma nova camada e, então, utilizamos o fluxograma da Figura 3 ao invés do da Figura 2.

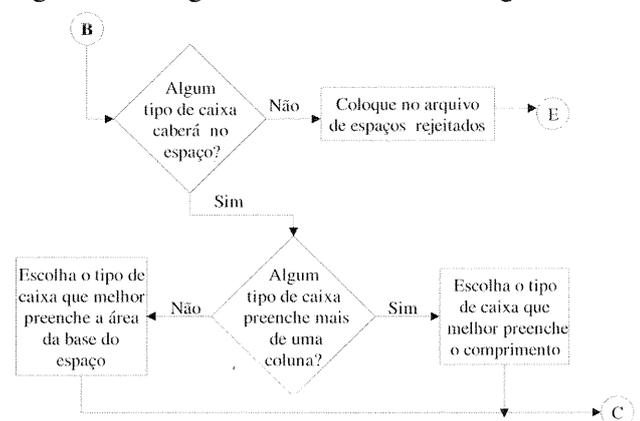


Figura 3. Escolha do tipo de caixa para espaços que sobraram na camada.

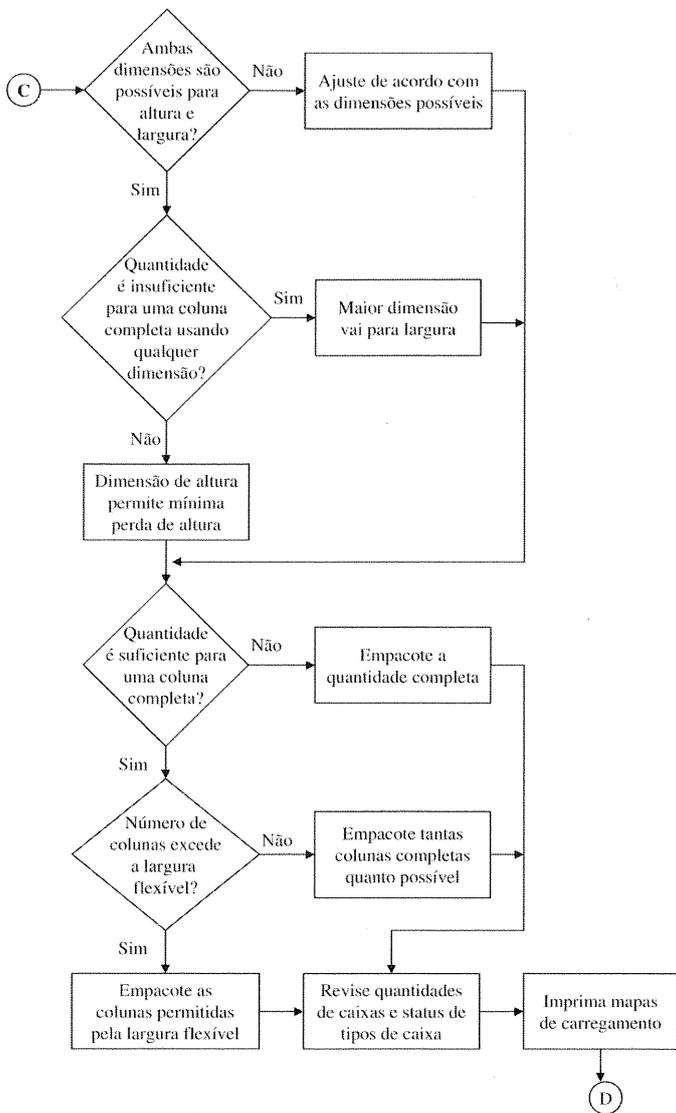


Figura 4. Escolha das dimensões de largura e altura e quantidade de caixas.

Se a camada não é preenchida completamente pelo tipo de caixa que a iniciou, os espaços vazios na camada são considerados como novos espaços a serem preenchidos. Esses espaços (que não constituem uma nova camada) são preenchidos de acordo com o fluxograma da Figura 3. Este procedimento também define a dimensão de comprimento para a camada. Ele também verifica se algum tipo de caixa caberá no espaço que sobrou na camada. Se existir, verificamos se algum tipo de caixa preenche mais de uma coluna neste espaço. Caso a resposta seja afirmativa, de acordo com a Figura 3, verificamos se as dimensões das possíveis caixas cabem na altura e largura do espaço e procuramos a dimensão de comprimento que, a partir de uma combinação de caixas iguais, preencha o melhor possível o comprimento da camada (como ilustra a Figura 5). Ou seja, a caixa que proporciona a menor perda no comprimento do espaço é a escolhida e, então, permitimos que várias colunas dessa caixa preencham o comprimento do espaço. Caso haja empate entre caixas, escolhemos a caixa com maior prioridade segundo o critério de prioridades.

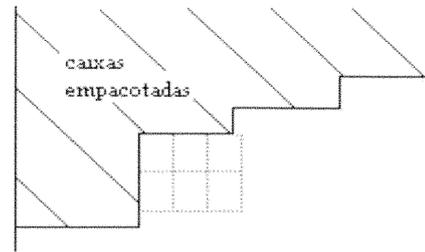


Figura 5. Espaço com três colunas na largura e duas no comprimento (visão do topo do contêiner).

Se não couber mais de uma coluna no espaço, de acordo com o fluxograma da Figura 3, escolhemos o tipo de caixa que preenche a maior área da base do espaço (como ilustra a Figura 6). Neste caso, as três dimensões da caixa são determinadas enquanto no caso anterior apenas a dimensão de comprimento é determinada. Se houver empate entre tipos de caixas, escolhemos o tipo com maior prioridade segundo o critério de prioridades. Em seguida, tendo escolhido o tipo de caixa e a melhor dimensão para o comprimento, vamos para o fluxograma da Figura 4. Caso não caiba nenhum tipo de caixa no espaço que sobrou na camada, de acordo com o fluxograma da Figura 3, colocamos o espaço no arquivo de espaços rejeitados e vamos para o fluxograma da Figura 7.

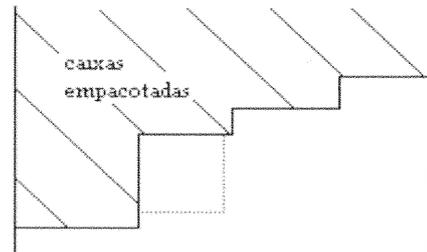


Figura 6. Espaço preenchido com caixa de maior área (visão do topo do contêiner).

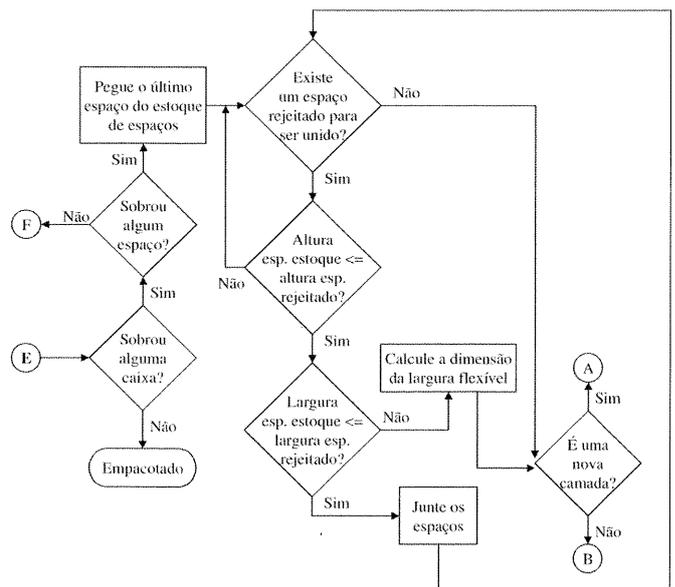


Figura 7. Escolha de um novo espaço

Tendo determinado o tipo de caixa e a dimensão de comprimento (Figuras 2 e 3), procuramos agora as dimensões de largura e altura e o número de caixas para colocar no espaço, de acordo com o fluxograma da Figura 4. Se as duas dimensões restantes da caixa podem ser utilizadas tanto como altura quanto como largura, verificamos se a quantidade de caixas é suficiente para completar uma coluna usando qualquer uma das dimensões. Caso contrário, escolhemos a maior dimensão da caixa para a dimensão de largura, porque deste modo é mais fácil encontrar outro tipo de caixa para colocar em cima dela posteriormente. Se não for insuficiente, escolhemos como dimensão de altura a dimensão que resultar em uma coluna maior (menor perda de altura).

Se as duas dimensões restantes da caixa não puderem ser utilizadas tanto como altura quanto como largura, escolhemos as dimensões de acordo com o possível. Ainda no fluxograma da Figura 4, quando a quantidade disponível da caixa é suficiente para completar uma coluna, verificamos se o número de colunas vai exceder a largura flexível (termo explicado adiante). Se exceder, empacotamos as colunas permitidas pela largura flexível. Se não exceder, empacotamos tantas colunas completas quanto possível. Caso a quantidade de caixas não seja suficiente para completar uma coluna, empacotamos todas elas. Então, de acordo com o fluxograma da Figura 4, revisamos a quantidade de caixas utilizadas e o *status* do tipo da caixa. Em seguida, vamos para o fluxograma da Figura 8.

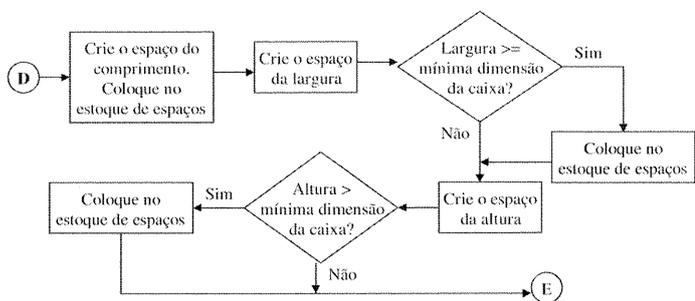


Figura 8. Criação de novos espaços.

Quando um conjunto de caixas é ajustado em um espaço, podem ser criados três sub-espaços: o espaço em cima do conjunto de caixas (espaço vazio na altura), o espaço ao lado do conjunto de caixas (espaço vazio na largura) e o espaço na frente do conjunto de caixas (espaço vazio no comprimento), conforme ilustrado na Figura 9. Estes espaços são colocados em um estoque de espaços. O espaço em cima das caixas é preenchido primeiro, depois o espaço ao lado, e finalmente o espaço na frente (esta ordem é arbitrária). Descartamos o espaço do lado e o espaço de cima das caixas se sua largura ou altura, respectivamente, são menores que a dimensão

mínima de qualquer caixa. O espaço da frente das caixas nunca é descartado porque tentamos juntá-lo com outros espaços que aparecerão posteriormente. Para isso, temos o fluxograma da Figura 8 que verifica se há criação desses espaços e se eles devem ser guardados no estoque de espaços ou não.

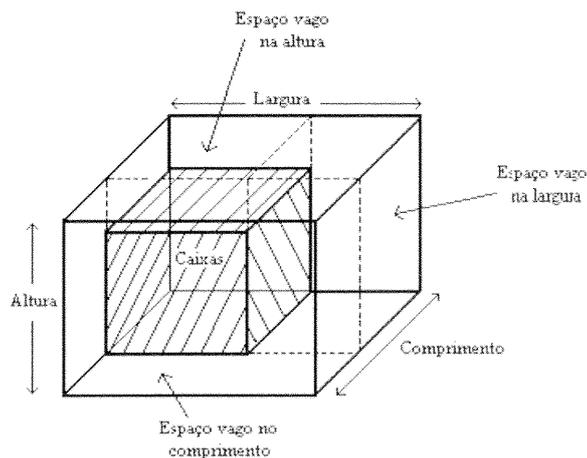


Figura 9. Criação de novos espaços.

Sempre utilizamos o último espaço colocado no estoque de espaços, mas antes de preenchê-lo, verificamos se ele pode ser unido com outros espaços rejeitados anteriormente ou se o parâmetro largura flexível (definido abaixo) é requerido. A Figura 10 ilustra a união de espaços entre a camada atual e a camada anterior. Após as caixas do bloco A terem sido empacotadas (admitindo que não há espaço sobrando em cima delas), o espaço criado ao lado é o espaço *bcgf* (Figura 10a). Entretanto, o espaço rejeitado na camada anterior (*hife*) permite que este espaço seja aumentado no comprimento. O espaço aumentado é agora *acge* e o espaço *hiba* é um espaço perdido (Figura 10a). Mas se preenchermos este espaço todo, ainda deixaremos um espaço vazio por trás (da camada anterior). Então, voltamos a verificar se esse novo espaço pode ser unido com outros espaços rejeitados (*jdek* na Figura 10b). Se a largura do espaço (*cg* na Figura 10b) for maior que a largura do espaço rejeitado (*de* na Figura 10b), calculamos a dimensão de largura flexível (*ad* na Figura 10b) e preenchemos apenas o espaço delimitado por essa largura. Então, quando não existir espaço rejeitado para juntar com o espaço atual, simplesmente vamos para o fluxograma da Figura 2 se este espaço corresponder a uma nova camada (espaço com altura e largura iguais às do contêiner), ou para o da Figura 3 se não corresponder a uma nova camada. Utilizando a definição de largura flexível, mantemos o contorno da camada anterior e evitamos que espaços vazios não sejam preenchidos. O fluxograma da Figura 7 descreve essa lógica.

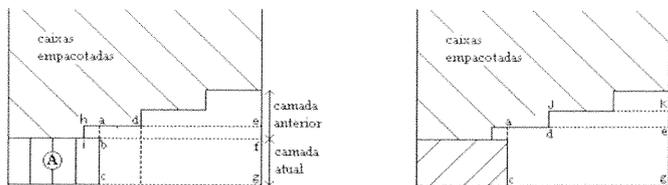


Figura 10. União de espaços (visão de topo do contêiner).

3 EXTENSÃO DA HEURÍSTICA G&R PARA RESTRIÇÃO DE ORIENTAÇÃO VERTICAL FIXA

Ao apresentar a heurística, George e Robinson (1980) não consideraram situações em que uma orientação vertical das caixas é fixa, o que é comum na prática. Quando um problema envolve fixar uma das dimensões das caixas, geralmente a dimensão de altura, a heurística G&R pode ser adaptada para tratar tal caso alterando-se os fluxogramas das Figuras 2-4. No fluxograma da Figura 2, após a caixa ser escolhida de acordo com os critérios de prioridades, a dimensão de altura é determinada e, portanto, a dimensão de comprimento deve ser a maior dimensão entre as duas dimensões restantes. Conseqüentemente, a última dimensão é a dimensão de largura, conforme mostrado no fluxograma da Figura 11.

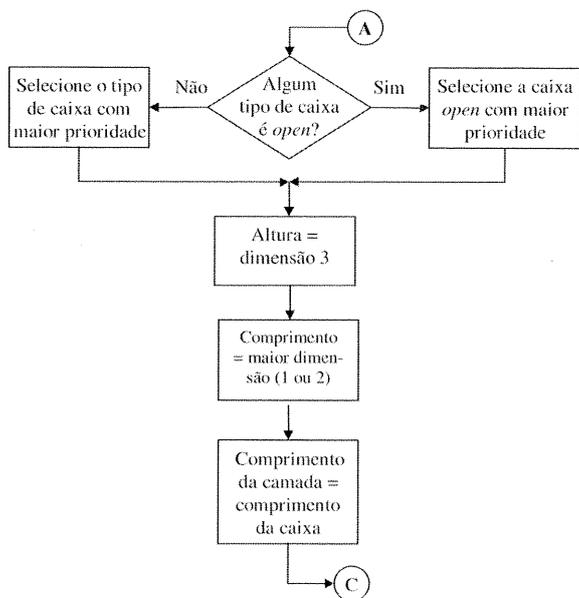


Figura 11. Fluxograma da Figura 2 para problemas com orientação vertical fixada.

No fluxograma da Figura 3, quando verificamos se é possível colocar mais de uma coluna no espaço em questão ou quando escolhemos a caixa que preenche a maior área da base do espaço, somente duas dimensões das caixas devem ser consideradas para a base do espaço, pois a dimensão de altura é fixa. Quando iniciamos o fluxograma da Figura 4, seja vindo do da Figura 2 ou 3, as três dimensões da caixa já foram determinadas. Portanto, a primeira parte do fluxograma da Figura 4 não deve ser realizada e ela deve iniciar

como mostrado no fluxograma da Figura 12. Os demais fluxogramas e procedimentos continuam iguais aos da heurística original.

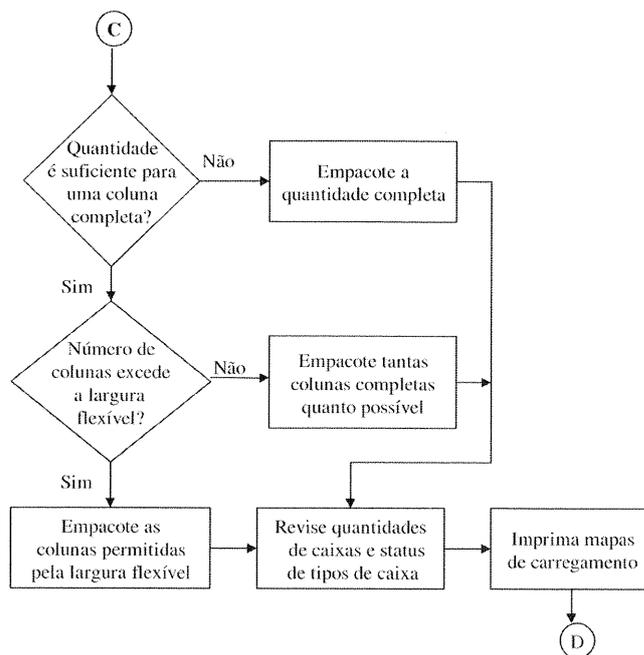


Figura 12. Fluxograma da Figura 4 para problemas com orientação vertical fixada.

4 REFINAMENTO DA HEURÍSTICA G&R

Um simples refinamento da heurística G&R pode ser obtido alterando-se um único procedimento do fluxograma original do algoritmo. Esse procedimento é o que escolhe um tipo de caixa para espaços que sobram na camada. Quando verificamos que nenhum tipo de caixa consegue preencher mais de uma coluna no espaço em questão, ao invés de simplesmente escolhermos o tipo de caixa que preenche a maior área da base do espaço, como na heurística original (fluxograma da Figura 3, verificamos se pode ser feita uma combinação de caixas de mesmo tamanho no comprimento deste espaço. A partir desta combinação de caixas, escolhemos aquela que resultar em uma maior utilização da área da base do espaço, conforme ilustrado na Figura 13. Apesar de simples, esta alteração pode melhorar bastante a solução gerada pela heurística. Os demais procedimentos e os critérios de prioridades continuam definidos conforme o algoritmo original.

Para ilustrar o benefício do refinamento proposto, ele foi aplicado no mesmo exemplo analisado em George e Robinson (1980). Este exemplo também foi analisado por outros autores, como Morabito e Arenales (1994) e Bischoff e Ratcliff (1995). Trata-se de um exemplo de uma situação real, envolvendo um contêiner padrão ISO série 2, com dimensões: $L = 5793$ mm, $W = 2236$ mm, e $H = 2261$ mm. A carga a ser carregada é composta de 8 tipos de caixas, em um total de 784 caixas. Apesar do volume total da carga ser

menor do que o volume útil do contêiner, a melhor solução publicada em George e Robinson (1980) carregou 783 caixas, deixando de fora apenas uma caixa do tipo 7. Isto se refere a um aproveitamento de aproximadamente 89,7% do volume do contêiner. A Figura 14a ilustra o contêiner empacotado com 783 caixas, conforme solução de George e Robinson (1980).

5. Caixa com a maior razão dada por: maior dimensão / menor dimensão.

O quarto critério aposta que caixas mais volumosas podem preencher maior parte da camada e o quinto critério, que quanto maior a razão acima apresentada, mais as caixas são desajeitadas e precisam ser arranjadas antes.

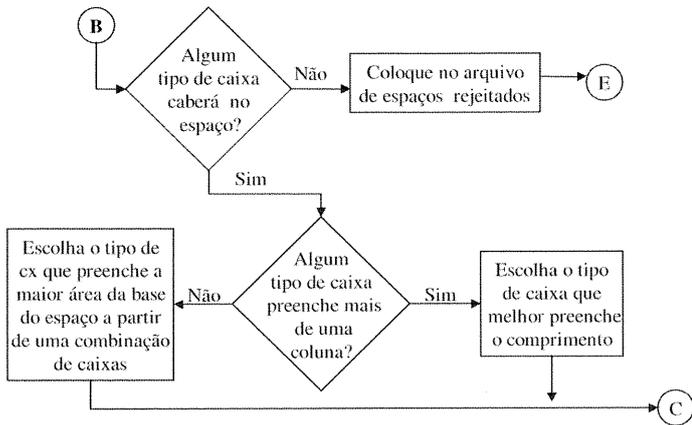


Figura 13. Escolha do tipo de caixa para espaços que sobraram na camada (refinamento).

Por outro lado, com o refinamento da heurística conseguimos carregar todas as 784 caixas, ou seja, obter a solução ótima do problema. Este resultado refere-se a um aproveitamento de aproximadamente 89,9 % do volume do contêiner. Assim como o anterior, o carregamento utilizou 10 camadas. Note que, apesar desta solução carregar apenas uma caixa a mais do que a anterior (cerca de 0,2% do volume do contêiner), na prática, isto pode ser uma melhoria significativa, uma vez que toda a carga poderia ter sido carregada e transportada no contêiner. A Figura 14b refere-se ao contêiner empacotado com todas as 784 caixas. Na seção 6 analisamos o desempenho deste refinamento em outros exemplos.

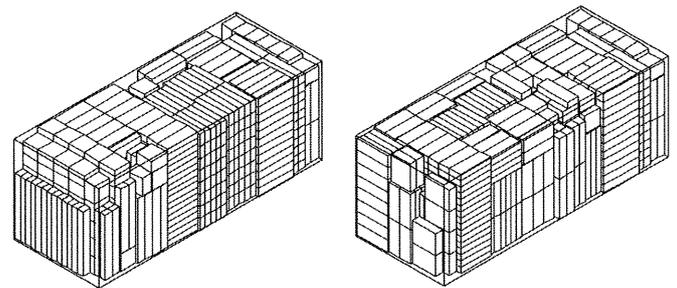


Figura 14. (a) Contêiner com 783 caixas; (b) Contêiner com 784 caixas.

5.1 Versão arranjo da heurística G&R

Na primeira versão da heurística, chamada versão Arranjo, utilizamos um arranjo de três dos cinco critérios de classificação ($A_{5,3} = 5!/2! = 60$). A justificativa para a escolha de apenas três entre cinco critérios de classificação é que dificilmente o desempate entre caixas passa do terceiro critério. A heurística é então executada 60 vezes, esgotando todas as possibilidades de arranjos, ou seja, em cada uma destas iterações um arranjo diferente de três critérios de classificação é utilizado. O melhor resultado entre eles, em termos de volume empacotado, é selecionado. Vale a pena lembrar que os critérios de classificação são utilizados na escolha da primeira caixa colocada na camada. Portanto, cada iteração produz um padrão de empacotamento distinto para o contêiner. Essa versão pode ser visualizada na Figura 15.

5 NOVAS VERSÕES DA HEURÍSTICA G&R

Duas novas versões da heurística G&R foram desenvolvidas, a versão Arranjo (no contêiner) e a versão (arranjo na) Camada. Para isso, cinco critérios de classificação das caixas foram determinados. Note que os três primeiros critérios são idênticos aos definidos em George e Robinson (1980). São eles:

1. Caixa com a maior das menores dimensões (a justificativa é que é mais difícil ajustá-la depois);
2. Caixa com a maior quantidade disponível (a justificativa é que uma maior quantidade pode preencher maior parte da camada);
3. Caixa com a maior dimensão (isto antecipa o arranjo de caixas de dimensões desfavoráveis);
4. Caixa com o maior volume;

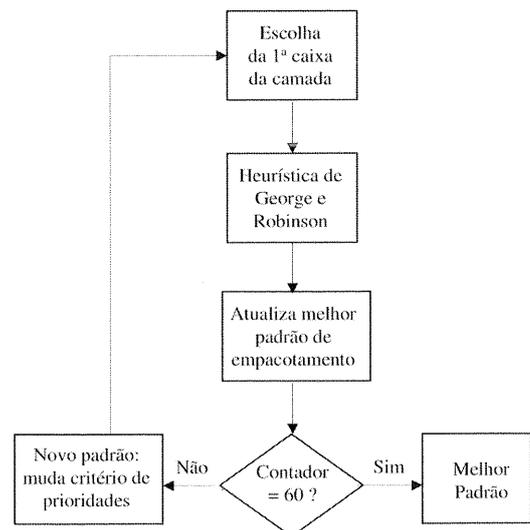


Figura 15. Versão Arranjo da heurística G&R.

5.2 Versão camada da heurística G&R

Na segunda versão da heurística, chamada versão Camada, também utilizamos o mesmo arranjo anterior de três dos cinco critérios de classificação ($A_{5,3}=5!/2!=60$), mas agora para determinar o padrão de empacotamento de cada camada do contêiner. Ou seja, para cada camada do contêiner a heurística é executada 60 vezes, esgotando todas as possibilidades de arranjos, e o melhor padrão de empacotamento encontrado para a camada é escolhido. Assim, a heurística começa a empacotar a próxima camada somente após executar 60 vezes a camada anterior, cada vez com um arranjo de critérios de classificação diferente. Então a próxima camada é empacotada levando-se em conta o melhor resultado da camada anterior e assim por diante, ou seja, as camadas que já foram empacotadas previamente não podem ser alteradas. Esse procedimento é repetido para todas as camadas. A melhor iteração de cada camada é determinada pela razão: volume das caixas empacotadas / volume parcial do contêiner, ou seja, a camada com menos espaço vazio é a melhor. Essa versão pode ser visualizada na Figura 16.

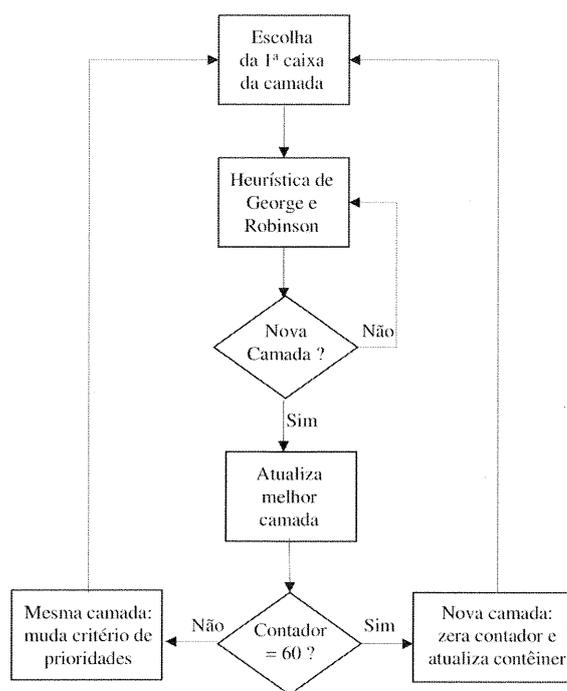


Figura 16. Versão Camada da heurística G&R.

6 MÉTODOS DE SOLUÇÃO PROPOSTOS

A partir da heurística G&R, do refinamento proposto para a heurística e das duas novas versões da heurística (versão Arranjo e versão Camada), definimos 6 métodos de solução (original mais 5 novas heurísticas) para o problema de carregamento de contêineres. São eles:

- Método 1: heurística G&R original;
- Método 2: método 1 + refinamento proposto (seção 4);
- Método 3: método 1 + versão Arranjo (seção 5.1);
- Método 4: método 1 + versão Arranjo + refinamento;
- Método 5: método 1 + versão Camada (seção 5.2);
- Método 6: método 1 + versão Camada + refinamento;

7 RESULTADOS COMPUTACIONAIS

Os 6 métodos apresentados na seção anterior (a heurística G&R original – método 1, e os 5 refinamentos propostos – métodos 2 a 6) foram implementados em Turbo-Pascal 7.0 em um microcomputador Pentium III (256 MB, 800MHz) e foram testados com exemplos aleatórios, alguns exemplos da literatura de corte e empacotamento e alguns exemplos reais de carregamento de contêineres. Em todos os exemplos duas orientações foram consideradas para o contêiner, (L, W, H) e (W, L, H) . A primeira considera que as caixas são arranjadas em camadas ao longo do comprimento L (i.e., no plano (W, H) , conforme ilustrado na Figura 1), e a segunda, em camadas ao longo da largura W (plano (L, H)). Por simplicidade, nos resultados apresentados reportamos apenas a melhor solução obtida para cada exemplo.

7.1 Exemplos com dados aleatórios

Para comparar os métodos 1-6, geramos exemplos aleatórios com as seguintes características:

- Contêiner de tamanho $L = 5793$ mm, $W = 2236$ mm e $H = 2261$ mm;
- Duas quantidades diferentes de tipos de caixa, $m = 10$ tipos e $m = 50$ tipos;
- Dimensões das caixas variando de 5% a 25% das dimensões do contêiner, ou seja, $l_i \in [0,05L, 0,25L]$, $w_i \in [0,05W, 0,25W]$, $h_i \in [0,05H, 0,25H]$, e de 5% a 50% das dimensões do contêiner, ou seja, $l_i \in [0,05L, 0,50L]$, $w_i \in [0,05W, 0,50W]$, $h_i \in [0,05H, 0,50H]$;
- Duas formas diferentes de gerar as quantidades b_i de cada tipo de caixa:

(i) Na primeira forma de gerar b_i , a parte do volume do contêiner que atribuímos para cada tipo de caixa varia, para $m=10$, de $1/10$ do volume inicial para a caixa

tipo 1, 1/9 do volume restante para a caixa tipo 2, e assim por diante, até 1/1 do volume restante para a caixa tipo 10 (similarmente para $m=50$). A seguir apresentamos esse procedimento para $m=10$ (a notação $\lfloor x \rfloor$ indica o maior inteiro menor ou igual a x);

$$V = L \times W \times H \quad v_i = l \times w \times h \quad \sum_i v_i b_i = V$$

$$V' \leftarrow V$$

$$\left\lfloor \frac{(\lfloor \frac{1}{10} \rfloor) \cdot V'}{v_1} \right\rfloor = b_1' \quad b_1 = [1, b_1'] \quad V' \leftarrow V - v_1 b_1$$

$$\left\lfloor \frac{(\lfloor \frac{1}{9} \rfloor) \cdot V'}{v_2} \right\rfloor = b_2' \quad b_2 = [1, b_2'] \quad V' \leftarrow V - v_1 b_1 - v_2 b_2$$

...

$$\left\lfloor \frac{(\lfloor \frac{1}{2} \rfloor) \cdot V'}{v_9} \right\rfloor = b_9' \quad b_9 = [1, b_9'] \quad V' \leftarrow V - v_1 b_1 - v_2 b_2 - \dots - v_9 b_9$$

$$\left\lfloor \frac{(\lfloor \frac{1}{1} \rfloor) \cdot V'}{v_{10}} \right\rfloor = b_{10}' \quad b_{10} = b_{10}'$$

Cada quantidade b_i é sorteada uniformemente no intervalo $[1, b_i']$. Assim, todos os tipos têm uma quantidade aproximadamente igual de caixas.

(ii) Na segunda forma de gerar b_i , maiores partes do volume do contêiner são atribuídas para os primeiros tipos de caixa, e menores partes do volume são atribuídas para os últimos tipos, de maneira que os primeiros tipos tenham muitas caixas e os últimos tipos tenham poucas caixas. A variação para cada tipo de caixa é de 1/2 do volume inicial do contêiner para a caixa tipo 1, 1/3 do volume restante para a caixa tipo 2, 1/4 do volume restante para a caixa tipo 3, e assim por diante, para ambos $m=10$ e $m=50$. O último tipo de caixa recebe o volume restante do contêiner, ou seja, no final temos 1/1 do volume restante do contêiner. A seguir apresentamos esse procedimento para $m=10$.

$$V = L \times W \times H \quad v_i = l \times w \times h \quad \sum_i v_i b_i = V$$

$$V' \leftarrow V$$

$$\left\lfloor \frac{(\lfloor \frac{1}{2} \rfloor) \cdot V'}{v_1} \right\rfloor = b_1' \quad b_1 = [1, b_1'] \quad V' \leftarrow V - v_1 b_1$$

$$\left\lfloor \frac{(\lfloor \frac{1}{3} \rfloor) \cdot V'}{v_2} \right\rfloor = b_2' \quad b_2 = [1, b_2'] \quad V' \leftarrow V - v_1 b_1 - v_2 b_2$$

...

$$\left\lfloor \frac{(\lfloor \frac{1}{9} \rfloor) \cdot V'}{v_9} \right\rfloor = b_9' \quad b_9 = [1, b_9'] \quad V' \leftarrow V - v_1 b_1 - v_2 b_2 - \dots - v_9 b_9$$

$$\left\lfloor \frac{(\lfloor \frac{1}{1} \rfloor) \cdot V'}{v_{10}} \right\rfloor = b_{10}' \quad b_{10} = b_{10}'$$

Assim, obtemos os oito conjuntos de dados C1-C8 mostrados na Tabela 1, cada conjunto contendo 10 exemplos gerados aleatoriamente, ou seja, um total de 80 exemplos.

Tabela 1. Conjuntos de dados C1-C8 gerados para os exemplos aleatórios.

Conjuntos	m	(l, w, h)	Geração de b_i
C1	10	5% a 25%	(i)
C2	10	5% a 25%	(ii)
C3	10	5% a 50%	(i)
C4	10	5% a 50%	(ii)
C5	50	5% a 25%	(i)
C6	50	5% a 25%	(ii)
C7	50	5% a 50%	(i)
C8	50	5% a 50%	(ii)

A Tabela 2 apresenta a média das soluções dos 8 conjuntos C1-C8 para cada método, e o número de vezes que cada método obteve a primeira, segunda, ..., sexta colocações, em termos do valor da solução nestes 8 conjuntos. Note que o método que obteve melhor solução média (87,04% de aproveitamento do volume do contêiner) foi o método 3 (Arranjo), seguido do método 4 (Arranjo + refinamento). Os métodos 2 (refinamento) e 6 (Camada + refinamento) tiveram melhor desempenho que seus métodos correspondentes sem o refinamento (métodos 1 e 5). Além disso, verificamos na Tabela 2 que o método 3 obteve o melhor desempenho quanto ao número de 1^{as} colocações (todas as 8 vezes!), seguido do método 4. Concluimos que, para este conjunto de exemplos C1-C8, os melhores métodos foram os métodos 3 e 4. Note que todos os métodos 2-6 tiveram desempenho superior ao método 1 (heurística G&R). Os tempos computacionais para resolver cada exemplo foram da ordem de centésimos de segundos.

Tabela 2. Resultados obtidos para os exemplos aleatórios dos conjuntos C1-C8.

Método	Solução	#1 ^a	#2 ^a	#3 ^a	#4 ^a	#5 ^a	#6 ^a
1	0,7879					2	6
2	0,7923				1	5	2
3	0,8704	8					
4	0,8670		7	1			
5	0,8104			4	4		
6	0,8117		1	3	3	1	

7.2 Exemplos da literatura

Para ilustrar o desempenho dos métodos 1-6 com relação a outros métodos da literatura, utilizamos os exemplos dos seguintes artigos (que consideram

estabilidade do carregamento): George e Robinson (1980), Abdou e Arghavani (1997), Liu e Hsiao (1997) e Morabito e Arenales (1994). O exemplo real apresentado em George e Robinson (1980) tem $m=8$ tipos de caixas (total de 784 caixas). Conforme mencionado na seção 4, a solução publicada em George e Robinson (1980) carrega 783 caixas (89,7% do volume do contêiner), deixando de fora apenas uma caixa do tipo 7 (método 1). Todos os outros métodos 2-6 encontraram uma solução ótima deste exemplo (Figura 14). Os resultados obtidos pelos métodos estão descritos na Tabela 3 (a coluna Tempo apresenta os tempos de execução em segundos).

Tabela 3. Resultados obtidos para o exemplo de George e Robinson (1980).

Método	Solução	#Caixas	Tempo (s)
1	0,8974	783	< 0,01
2	0,8988*	784	< 0,01
3	0,8988*	784	< 0,01
4	0,8988*	784	< 0,01
5	0,8988*	784	< 0,01
6	0,8988*	784	< 0,01

Abdou e Arghavani (1997) estudaram um exemplo com $m=7$ tipos de caixas (total de 11 caixas), que podem ser arranjadas nas duas orientações (l, w, h) ou (w, l, h). A solução publicada em Abdou e Arghavani (1997) carrega todas as 11 caixas. Os resultados obtidos pelos métodos 1-6, agora adaptados para tratar orientação vertical fixa (conforme descrito na seção 3), estão apresentados na Tabela 4. Apenas a heurística G&R (método 1) deixou de colocar uma caixa do tipo 2 no contêiner. Todos os outros métodos 2-6 encontraram uma solução ótima do exemplo, carregando todas as caixas no contêiner. Esse exemplo reforça que o refinamento proposto pode melhorar a solução gerada pela heurística G&R.

Tabela 4. Resultados obtidos para o exemplo de Abdou e Arghavani (1997).

Método	Solução	#Caixas	Tempo (s)
1	0,9286	10	< 0,01
2	1,0000*	11	< 0,01
3	1,0000*	11	< 0,01
4	1,0000*	11	< 0,01
5	1,0000*	11	< 0,01
6	1,0000*	11	< 0,01

Também testamos os métodos 1-6 em exemplos com um único tipo de caixa. Liu e Hsiao (1997) propuseram um método que permite o empilhamento das caixas sobre qualquer uma de suas faces. Todos os métodos 1-6 obtiveram os mesmos resultados para os exemplos

apresentados em Liu e Hsiao (1997); entretanto, os resultados obtidos por Liu e Hsiao (1997) foram melhores, conforme mostra a Tabela 5 (H é a altura do palete ou contêiner). Convém notar que, diferente do método de Liu e Hsiao (1997), os métodos 1-6 tratam do problema mais geral com diferentes tipos de caixas, sem explorar conhecimento específico do problema com caixas iguais. Aparentemente estes métodos não têm bom desempenho nestes problemas. Um método com bom desempenho para o problema com caixas iguais foi apresentado em Lins *et al.* (2002).

Tabela 5. Resultados obtidos para os exemplos de Liu e Hsiao (1997).

Altura H	Liu e Hsiao		Métodos 1-6	
	Solução	#Caixas	Solução	#Caixas
1200	0,872	44	0,833	42
1250	0,856	45	0,799	42
1300	0,933	51	0,805	44
1350	0,898	51	0,846	48
1400	0,883	52	0,816	48
1450	0,886	54	0,787	48
1500	0,856	54	0,825	52

Também testamos os métodos 1-6 para os conjuntos de exemplos aleatórios de Morabito e Arenales (1994), para analisá-los em problemas irrestritos (sem limitações nas quantidades de caixas b_i de cada tipo). Morabito e Arenales (1994) apresentaram uma abordagem de busca baseada na representação do espaço de soluções em um grafo-e/ou. Foram gerados 80 exemplos aleatórios (com $m = 5, 10, 20$ e 30 tipos de caixas, com dimensões variando de 5% a 85% das dimensões do contêiner), que foram agrupados em 8 conjuntos de exemplos S1-S8. Como estes exemplos são irrestritos, para cada tipo de caixa, foram consideradas quantidades b_i suficientemente grandes para preencher todo contêiner. A Tabela 6 mostra a média das soluções dos 8 conjuntos de dados para cada método e o número de vezes que cada método obteve a primeira, segunda, ..., sexta colocações, em termos do valor da solução nestes 8 conjuntos. Os tempos computacionais para resolver cada exemplo foram da ordem de poucos centésimos de segundos.

Tabela 6. Resultados obtidos para os exemplos dos conjuntos S1-S8 de Morabito e Arenales (1994).

Método	Solução	#1 ^a	#2 ^a	#3 ^a	#4 ^a	#5 ^a	#6 ^a
1	0,8562			1	1	1	5
2	0,8579			1	1	6	
3	0,8865	1	2	3	2		
4	0,8877	4	1	2	1		
5	0,8935	2	2	2	2		
6	0,8943	2	4	1	1		

Na Tabela 6 podemos observar que todos os métodos que consideram o refinamento proposto à heurística G&R (métodos 2, 4 e 6) tiveram desempenho um pouco superior do que seus métodos correspondentes sem o refinamento (métodos 1, 3 e 5). O método que obteve a melhor solução média (89,43% de aproveitamento do volume do contêiner) foi o método 6 (Camada + refinamento). Verificamos também que o método 4 (Arranjo + refinamento) obteve o melhor desempenho quanto ao número de 1^{as} colocações. Assim, para este conjunto de exemplos irrestritos S1-S8, os métodos que obtiveram melhor desempenho foram os métodos 4 e 6, seguidos dos métodos 3 e 5.

Na Tabela 7 comparamos a melhor solução encontrada pelos métodos 1-6 (para cada conjunto S1-S8) com a solução da abordagem em grafo e/ou (adaptada para permitir a rotação das caixas no contêiner). Para todos os conjuntos S1-S8, esta abordagem obteve soluções melhores que os métodos propostos, porém consumindo um tempo computacional (em segundos) muito maior.

Tabela 7. Comparação dos resultados obtidos para os conjuntos S1-S8.

Conj.	Abordagem em grafo-e/ou		Melhor solução dos métodos 1-6	
	Solução	Tempo	Solução	Tempo
S1	0,8802	10,63	0,8150	0,01
S2	0,9340	12,81	0,8386	0,01
S3	0,8354	3,87	0,8252	0,01
S4	0,9871	40,78	0,9473	0,03
S5	0,9521	24,49	0,8892	0,04
S6	0,9717	66,34	0,9451	0,01
S7	0,9812	127,05	0,9561	0,05
S8	0,9844	147,33	0,9670	0,02

7.3 Exemplos reais

Para testar os métodos 1-6 em exemplos reais, utilizamos dados de 7 empresas (empresas A-G) fornecidos por uma empresa de consultoria. Os resultados obtidos pelos métodos foram comparados com os resultados obtidos pelo software de otimização de empacotamento de caixas desta empresa de consultoria. O tempo computacional utilizado pelo software não foi fornecido. As Tabelas 8 e 9 comparam os resultados (B-1 e B-2 correspondem aos exemplos 1 e 2 da empresa B; similarmente para F-1, F-2, F-3 e G-1, G-2, G-3). Note que há exemplos com milhares de caixas. Mais detalhes destes exemplos são encontrados em Cecilio (2003).

Observe na Tabela 8 que todos os métodos 1-6 encontram uma solução ótima do exemplo A (com $m=47$ tipos de caixas e um total de 159 caixas), ao contrário do software de otimização. No caso do exemplo B-1 (com 262 caixas), o método 5 (Camada)

obteve a melhor solução e somente o método 2 teve um desempenho pior do que o software. Apenas o método 3 encontrou uma solução ótima para o exemplo B-2, e o método 2 teve o pior desempenho. Todos os métodos encontram soluções ótimas para os exemplos C, D e E.

Tabela 8. Resultados obtidos pelos métodos 1-6 para os exemplos das empresas A, B, C, D e E.

Exemplo	Método	Solução	#Caixas	Tempo (s)
A ($m=47$)	1	0,8622*	159	< 0,01
	2	0,8622*	159	< 0,01
	3	0,8622*	159	< 0,01
	4	0,8622*	159	< 0,01
	5	0,8622*	159	0,33
	6	0,8622*	159	0,33
	Software	0,8318	157	---
B-1 ($m=26$)	1	0,8196	241	< 0,01
	2	0,8081	237	< 0,01
	3	0,8271	242	0,22
	4	0,8252	243	0,22
	5	0,8556	250	0,39
	6	0,8271	243	0,44
	Software	0,8174	248	---
B-2 ($m=22$)	1	0,8070	238	< 0,01
	2	0,7805	231	< 0,01
	3	0,8367*	246	0,05
	4	0,8195	241	0,17
	5	0,8163	243	0,27
	6	0,8163	243	0,33
	Software	0,8209	241	---
C ($m=4$)	1	0,7999*	2800	< 0,01
	2	0,7999*	2800	< 0,01
	3	0,7999*	2800	< 0,01
	4	0,7999*	2800	< 0,01
	5	0,7999*	2800	< 0,01
	6	0,7999*	2800	< 0,01
	Software	0,7999*	2800	---
D ($m=10$)	1	0,7502*	3711	< 0,01
	2	0,7502*	3711	< 0,01
	3	0,7502*	3711	< 0,01
	4	0,7502*	3711	< 0,01
	5	0,7502*	3711	< 0,01
	6	0,7502*	3711	< 0,01
	Software	0,7502*	3711	---
E ($m=5$)	1	0,8073*	3590	< 0,01
	2	0,8073*	3590	< 0,01
	3	0,8073*	3590	< 0,01
	4	0,8073*	3590	< 0,01
	5	0,8073*	3590	< 0,01
	6	0,8073*	3590	< 0,01
	Software	0,8073*	3590	---

* solução ótima

Podemos observar na Tabela 9 que apenas o método 3 encontrou uma solução ótima para o exemplo F-1. Nos exemplos F-2 e F-3, apenas os métodos 3 e 4 e o software de otimização encontraram soluções ótimas.

No caso da empresa G, os métodos 3-6 encontram soluções melhores do que o software para o exemplo G-1, e todos os métodos 1-6 encontram soluções melhores do que o software para os exemplos G-2 e G-3. As soluções obtidas pelos métodos 3-6 para o exemplo G-1 são ótimas, mas nenhum método encontrou soluções carregando todas as caixas para os exemplos G-2 e G-3 (492 e 227 caixas, respectivamente).

Tabela 9. Resultados obtidos pelos métodos 1-6 para os exemplos das empresas F e G.

Exemplo	Método	Solução	#Caixas	Tempo (s)
F-1 (m=5)	1	0,8869	1903	< 0,01
	2	0,8869	1903	< 0,01
	3	0,9423*	1950	0,05
	4	0,9310	1941	0,06
	5	0,9105	1923	0,11
	6	0,9105	1923	0,11
	Software	0,9171	1834	---
F-2 (m=5)	1	0,9247	1859	< 0,01
	2	0,8988	1837	< 0,01
	3	0,9259*	1860	0,05
	4	0,9259*	1860	0,06
	5	0,9047	1842	0,11
	6	0,9047	1842	0,11
	Software	0,9259*	1860	---
F-3 (m=5)	1	0,9248	1895	< 0,01
	2	0,9181	1850	< 0,01
	3	0,9354*	1904	0,05
	4	0,9354*	1904	0,06
	5	0,9290	1884	0,11
	6	0,9290	1884	0,11
	Software	0,9354*	1904	---
G-1 (m=5)	1	0,9162	505	< 0,01
	2	0,9276	514	< 0,01
	3	0,9929*	592	0,06
	4	0,9929*	592	0,06
	5	0,9929*	592	0,06
	6	0,9929*	592	0,06
	Software	0,9426	543	---
G-2 (m=4)	1	0,9830	312	< 0,01
	2	0,9838	317	< 0,01
	3	0,9830	312	0,05
	4	0,9838	317	0,00
	5	0,9693	334	0,11
	6	0,9693	334	0,05
	Software	0,9330	451	---
G-3 (m=9)	1	0,8902	226	< 0,01
	2	0,8902	226	< 0,01
	3	0,8902	226	0,06
	4	0,8902	226	0,05
	5	0,8902	226	0,11
	6	0,8902	226	0,05
	Software	0,8826	221	---

* solução ótima

Os experimentos computacionais realizados neste estudo são limitados e não permitem concluir qual o

melhor dentre os métodos 2-6. Apesar de nenhum método dominar os outros, conforme ilustrado nos exemplos, observa-se em geral um bom desempenho dos métodos 3 e 4 em relação aos demais. Como os tempos computacionais são bem pequenos (bem menos de 1 segundo), recomenda-se aplicar todos os métodos (ou pelo menos os métodos 3 e 4), e simplesmente adotar a melhor solução encontrada.

8 CONCLUSÕES

Este estudo abordou o problema do carregamento de caixas dentro de contêineres. Heurísticas foram propostas para resolvê-lo (métodos 2-6), baseadas em extensões e refinamentos da conhecida heurística de George e Robinson (método 1). O método 2 considera um simples refinamento nesta heurística, o método 3 (Arranjo) produz diversos padrões de empacotamento para o contêiner e escolhe o melhor, o método 4 (Arranjo + refinamento) considera os métodos 2 e 3 ao mesmo tempo, o método 5 (Camada) produz diversos padrões de empacotamento para cada camada do contêiner e escolhe o melhor, e o método 6 (Camada + refinamento) considera os métodos 2 e 5 ao mesmo tempo. Todos os métodos 2-5 foram capazes de obter a solução ótima do exemplo real apresentado em George e Robinson (1980), ao contrário da heurística original (método 1).

Outros experimentos computacionais com exemplos da literatura, aleatórios e de situações reais de carregamento de contêineres foram realizados e os resultados obtidos foram comparados com os resultados de outros métodos e de um software de otimização fornecido por uma empresa de consultoria. Casos com a restrição de orientação vertical fixa das caixas também foram tratados. Em todos os exemplos, os tempos computacionais foram bem pequenos (bem menos de 1 segundo em um microcomputador Pentium III e compilador Turbo-Pascal 7.0), o que torna as heurísticas propostas bem convenientes para aplicação na prática.

Em geral os métodos obtiveram bons resultados. No caso dos 80 exemplos gerados aleatoriamente, o método 3 obteve o melhor desempenho em todos os casos, seguido do método 4. Nos 80 exemplos irrestritos da literatura, os métodos 4 a 6 obtiveram os melhores desempenhos, seguidos dos métodos 3 e 5. Observa-se que os métodos propostos 2-6 têm desempenho superior ao da heurística de George e Robinson (1980) (método 1), tanto nos exemplos restritos como nos irrestritos. O pior desempenho dos métodos ocorreu nos problemas com um único tipo de caixa, mostrando que eles não parecem ser adequados para esse tipo de problema. Um método eficaz para tal caso pode ser encontrado em Lins *et al.* (2002).

Com relação aos exemplos reais de carregamento de contêineres, alguns métodos demonstraram um desempenho um pouco superior aos demais, mas, de

modo geral, eles alcançaram ou superaram os resultados obtidos pelo software de otimização. Os métodos que obtiveram os melhores resultados para os exemplos reais foram os métodos 3 e 4. Na maior parte destes exemplos, eles encontraram uma solução carregando todas as caixas disponíveis. Apesar dos experimentos realizados neste estudo serem limitados, observou-se que nenhum dos métodos 2-6 domina os outros, e em geral os métodos 3 e 4 tiveram bom desempenho em relação aos demais.

É importante ressaltar que os métodos aqui propostos são fáceis de serem implementados em computador, são flexíveis para incluir restrições práticas adicionais, são conservadores quanto à geração de carregamentos estáveis e obtêm boas soluções com pouco esforço computacional, o que viabiliza suas aplicações nas atividades diárias de empacotamento das empresas. Devido à característica de rapidez na execução dos métodos, uma sugestão de utilização na prática é a criação de um pacote com todos os métodos propostos 2-6 (ou pelo menos os métodos 3 e 4), que forneça as melhores soluções encontradas. Algumas perspectivas para pesquisa futura são (i) aplicar os métodos em situações reais de estudos de caso, para analisar melhor seus desempenhos em relação aos procedimentos das empresas ao longo de um horizonte de tempo, e (ii) combinar os métodos com metaheurísticas do tipo Algoritmo Genético, *Simulated Annealing*, Busca Tabu e Grasp, para melhorar seus desempenhos.

AGRADECIMENTOS

Os autores agradecem aos revisores anônimos pelos úteis comentários e sugestões. Esta pesquisa contou com apoio da FAPESP (processo 01/2972-2) e CNPq (processo 522973/95-7).

REFERÊNCIAS BIBLIOGRÁFICAS

Abdou, G. e Arghavani, J. (1997) Interactive ILP Procedures for Stacking Optimization for the 3D Palletization Problem. *International Journal of Production Research*, v.35, n.5, p.1287-1304.

Abdou, G. e Yang, M. (1994) A Systematic Approach for the Three Dimensional Palletization Problem. *International Journal of Production Research*, v.32, n.10, p.2381-2394.

Beasley, J.E. (1985) An Exact Two-Dimensional Non-Guillotine Cutting Tree Search Procedure. *Operations Research*, v.33, p.49-64.

Beasley, J.E. (2001) A Population Heuristic for Constrained Two-Dimensional Non-Guillotine Cutting. Aceito para publicação no *European Journal of Operational Research*.

Bischoff, E.E., Janetz, F., Ratcliff, M.S.W. (1995) Loading Pallets with Non-Identical Items. *European Journal of Operational Research*, v.84, p.681-692.

Bischoff, E.E. e Dowsland, W.B. (1982) An Application of the Micro to Product Design and Distribution. *Journal of the*

Operational Research Society, v.33, p.271-280.

Bischoff, E.E. e Marriott, M.D. (1990) A Comparative Evaluation of Heuristics for Container Loading. *European Journal of Operational Research*, v.44, p.267-276.

Bischoff, E.E. e Ratcliff, M.S.W. (1995) Loading Multiple Pallets. *Journal of the Operational Research Society*, v.46, p.1322-1336.

Bischoff, E.E. e Ratcliff, M.S.W. (1995a) Issues in the Development of Approaches to Container Loading. *Omega*, v.23, n.4, p.377-390.

Bortfeldt, A. e Gehring, H. (2001) A Hybrid Genetic Algorithm for the Container Loading Problem. *European Journal of Operational Research*, v.131, p.143-161.

Brown, D.J. (1980) An Improved BL Lower Bound. *Information Processing Letters*, v.11, n.1, p.37-39.

Cecilio, F.O. (2003) *Heurísticas para o problema de carregamento de carga dentro de contêineres*. São Carlos. Dissertação de Mestrado, UFSCar.

Chen, C.S., Lee, S.M., Shen, Q.S. (1995) An Analytical Model for the Container Loading Problem. *European Journal of Operational Research*, v.80, p.68-76.

Chien, C.F. e Wu, W.T. (1998) A Recursive Computational Procedure for Container Loading. *Computers & Industrial Engineering*, v.35, p.319-322.

Chien, C.F. e Wu, W.T. (1999) A Framework of Modularized Heuristics for Determining the Container Loading Patterns. *Computers & Industrial Engineering*, v.37, p.339-342.

Correia, M.H., Gomes, A.M., Oliveira, J.F., Ferreira, J.S. (1992) Problemas de Empacotamento Tridimensional. *Investigação Operacional*, v.12, n.2, p.169-180.

Davies, A.P. e Bischoff, E.E. (1999) Weight Distribution Considerations in Container Loading. *European Journal of Operational Research*, v.114, p.509-527.

Dowsland, K.A. e Dowsland, W.B. (1992) Packing Problems. *European Journal of Operational Research*, v.56, p.2-14.

Dyckhoff, H. (1990) A Typology of Cutting and Packing Problems. *European Journal of Operational Research*, v.44, p.145-159.

Dyckhoff, H. e Finke, U. (1992) *Cutting and Packing in Production and Distribution: Typology and Bibliography*. Heidelberg: Springer-Verlag.

Gehring, H. e Bortfeldt, A. (1997) A Genetic Algorithm for Solving the Container Loading Problem. *International Transactions in Operations Research*, v.4, n.5/6, p.401-418.

Gehring, H., Menschner, K., Meyer, M. (1990) A Computer-Based Heuristic for Packing Pooled Shipment Containers. *European Journal of Operational Research*, v.44, p.277-288.

George, J.A. (1992) A Method for Solving Container Packing for a Single Size of Box. *Journal of the Operational Research Society*, v.43, n.4, p.307-312.

George, J.A. e Robinson, D.F. (1980) A Heuristic for Packing Boxes into a Container. *Computers & Operations Research*, v.7, p.147-156.

Haessler, R.W. e Talbot, F.B. (1990) Load Planning for Shipments of Low Density Products. *European Journal of Operational Research*, v.44, p.289-299.

Hadjiconstantinou, E. e Christofides, N. (1995) An Exact Algorithm for General, Orthogonal, Two-Dimensional

- Knapsack Problems. *European Journal of Operational Research*, v.83, p.39-56.
- Han, C.P., Knott, K., Egbelu, P.J. (1989) A Heuristic Approach to the Three-Dimensional Cargo Loading Problem. *International Journal of Production Research*, v.27, n.5, p.757-774.
- Herbert, E.A. e Dowsland, K.A. (1996) A Family of Genetic Algorithms for the Pallet Loading Problem. *Annals of Operations Research*, v.63, p.415-436.
- Lins, L., Lins, S., Morabito, R. (1999) A 9-Fold Partition Heuristic for Packing Boxes into a Container. *Investigación Operativa*, v.7, n.3, p.69-82.
- Lins, L., Lins, S., Morabito, R. (2002) 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.
- Liu, F.F. e Hsiao, C. (1997) A Three-Dimensional Pallet Loading Method for Single-Size Boxes. *Journal of the Operational Research Society*, v.48, p.726-735.
- Martello, S., Pisinger, D., Vigo, D. (2000) The Three Dimensional Bin Packing Problem. *Operations Research*. Março-Abril.
- Martins, G.H. (2002) *Packing in two and three dimensions*. Monterey, CA, EUA. PhD Dissertation. Naval Postgraduate School.
- Miyazawa, F.K. (1997) *Algoritmos de Aproximação para Problemas de Empacotamento*. São Paulo. Tese de Doutorado, USP.
- Miyazawa, F.K. e Wakabayashi, Y. (1997) An Algorithm for the Three-Dimensional Packing Problem. *Algorithmica*, v.18, p.122-144.
- Miyazawa, F. K. e Wakabayashi, Y. (2000) Approximation Algorithms for the Orthogonal Z-Oriented 3-D Packing Problem. *SIAM Journal on Computing*, v.29, p.1008-1029.
- Mohanty, B.B., Mathur, K., IVANCIC, N.J. (1994) Value Considerations in Three Dimensional Packing – A Heuristic Procedure Using the Fractional Knapsack Problem. *European Journal of Operational Research*, v.74, p.143-151.
- Morabito, R. (1992) *Uma Abordagem em Grafo-E/OU para o Problema do Empacotamento: Aplicação ao Problema do Carregamento de Paletes e Contêineres*. São Carlos. Tese de Doutorado, Escola de Engenharia de São Carlos, USP.
- Morabito, R. e Arenales, M. (1992) Um exame dos Problemas de Corte e Empacotamento. *Pesquisa Operacional*, v.12, n.1, p.1-20.
- Morabito, R. e Arenales, M. (1994) An And/Or-graph Approach to the Container Loading Problem. *International Transactions in Operations Research*, v.1, n.1, p.59-73.
- Morabito, R. e Arenales, M. (1997) Abordagens para o Problema do Carregamento de Contêineres. *Pesquisa Operacional*, v.17, n.1, p.29-56.
- Morabito, R., Morales, S., Widmer, J. (2000) Loading Optimization of Palletized Products on Trucks. *Transportation Research*, Part E v.36, p.285-296.
- Pisinger, D. (1999) A Tree Search Heuristic for the Container Loading Problem. *Ricerca Operativa*.
- Silva, J.L.C. (2002) *Algoritmos para o Empacotamento de Bins Tridimensionais: Uma Abordagem Distribuída*. São José dos Campos. Tese de Doutorado, ITA.
- Silva, J.L.C., Soma, N.Y., Maculan, N. (2003) A Greedy Search for the Three-Dimensional Bin Packing Problem: the Packing Static Stability Case. *International Transactions in Operational Research*, v.10, p.1-13.
- Tsai, R.D., Malstrom, E.M., Kuo, W. (1993) Three-Dimensional Palletization of Mixed Box Sizes. *IIE Transactions*, v.25, n.4, p.64-75.
- Xue, J., Lai, K.K. (1997) Effective Methods for a Container Packing Operation. *Mathematical and Computer Modelling*, v.25, n.2, p.75-84.

CONTATOS

¹Nome: Fabiana Oliveira Cecilio

E-mail: pcecilio@iris.ufscar.br

²Nome: Reinaldo Morabito

E-mail: morabito@power.ufscar.br