

HEURÍSTICAS PARA O PROBLEMA BIDIMENSIONAL DE EMPACOTAMENTO IRREGULAR COM DISTRIBUIÇÃO DE PESO

Raquel Akemi Okuno Kitazume

Instituto de Ciências Matemáticas e de Computação
Avenida Trabalhador São-carlense, 13566-590, São Carlos - SP, Brasil
raquel.akemi@usp.br

Marina Andretta

Instituto de Ciências Matemáticas e de Computação
Avenida Trabalhador São-carlense, 13566-590, São Carlos - SP, Brasil
andretta@icmc.usp.br

RESUMO

O problema de empacotamento de itens irregulares com distribuição de peso é encontrado no carregamento de aviões, caminhões e navios. O objetivo é utilizar o menor número de recipientes possível, mas de forma que os recipientes utilizados fiquem com os pesos distribuídos. Neste trabalho, resolvemos a versão bidimensional do problema, propondo duas heurísticas que dividem os recipientes em quatro regiões e intercala a alocação dos itens entre elas. A diferença entre as heurísticas está na regra de posicionamento utilizada em cada uma. Os testes computacionais mostraram que as heurísticas são promissoras.

PALAVRAS CHAVE. Problema de empacotamento de itens irregulares, Distribuição de peso, Heurísticas.

Tópicos (Otimização Combinatória)

ABSTRACT

The irregular bin packing problem with weight distribution is found in the loading of aircrafts, trucks and ships. Its goal is to use the lowest possible number of containers but in a way that they remain with distributed weights. In this paper, we solve the two-dimensional version of the problem, proposing two heuristics that divide the containers into four regions and alternate the allocation among them. The difference between both heuristics is in the placement rule adopted for each one. The computational results showed that the heuristics are promising.

KEYWORDS. Irregular bin packing problem, Weight distribution, Heuristics.

Paper topics (Combinatorial Optimization)

1. Introdução

O problema de empacotamento com distribuição de peso busca um arranjo de objetos menores, os itens, dentro de objetos maiores, recipientes, com o objetivo de utilizar o menor número de recipientes possível, mas de forma que os itens não causem o desbalanceamento dos recipientes. Esse problema tem várias aplicações, como o carregamento de aviões e navios. O desbalanceamento em aviões afeta a força de arrasto exercida sobre a aeronave. Assim, balancear a carga contribui para um voo mais seguro, mais rápido e um menor consumo de combustível [Mongeau e Bes, 2003]. Já o desbalanceamento em navios pode causar danos estruturais no navio, como o alquebramento. Estamos interessados no caso em que as cargas contêm itens irregulares, ou seja, itens com formas não-circulares e não-retangulares.

Desejamos resolver a versão bidimensional do problema, considerando a representação dos itens no eixo x e y , desprezando o eixo z . Assim, consideramos que o empilhamento de itens não é permitido ou que os empilhamentos já foram feitos em uma fase de pré-processamento. Na literatura, encontramos alguns trabalhos relacionados ao nosso problema, como [Egeblad, 2009], [Kaluzny e Shaw, 2009], [Liu et al., 2011] e [Davies e Bischoff, 1999].

Dos trabalhos citados, o único que trabalha com empacotamento de itens irregulares é o trabalho de [Egeblad, 2009]. O autor resolve o problema de posicionamento de itens em um único recipiente considerando o balanceamento e momento de inércia bidimensional e tridimensional. Assim, dado um recipiente e uma lista de itens (regulares e irregulares), deseja-se empacotar todos os itens dentro do recipiente, de forma que o momento de inércia e o desbalanceamento da carga sejam minimizados e os itens não se sobreponham. A simplificação do tridimensional para o bidimensional é feita considerando os eixos x e y dos itens e recipientes, desconsiderando o eixo z . O autor prova que o problema é NP-difícil e apresenta um algoritmo baseado no trabalho de [Egeblad et al., 2007]. O algoritmo encontra bons resultados, mas não foi comparado com outros métodos da literatura.

O trabalho de [Kaluzny e Shaw, 2009] resolve o problema bidimensional de carregamento de avião com itens retangulares, no qual o objetivo é minimizar o desbalanceamento da aeronave. A simplificação do tridimensional é feita considerando a largura e comprimento dos itens. Os autores propõem modelos lineares inteiros mistos, um para o empacotamento em recipientes e outro para o problema da mochila.

Já os trabalhos de [Liu et al., 2008] e [Fernández et al., 2013] resolvem o problema multiobjetivo de empacotamento com balanceamento bidimensional retangular. A simplificação do tridimensional para o bidimensional é feita considerando o eixo x e z , ignorando o eixo y . Em seu trabalho, Liu et al. [2008] formulam dois modelos bidimensionais para o problema, um multiobjetivo e outro com apenas um objetivo (minimizar o número de recipientes), e propõem um algoritmo multiobjetivo de otimização por enxame de partículas evolucionário para resolver o modelo multiobjetivo proposto. O método foi comparado com outros da literatura e os superou na maior parte dos casos testados. Também apresentou um tempo médio menor que os outros métodos.

Já Fernández et al. [2013] analisa a performance de um algoritmo memético (sequencial e paralelo) proposto para resolver problemas de empacotamento multiobjetivos com rotação. O algoritmo apresentado utiliza uma estratégia de inicialização e vários operadores de busca feitos especificamente para resolver o problema. Os experimentos computacionais obtidos resultaram em boa performance. Também o algoritmo memético paralelo melhorou a qualidade das fronteiras de pareto obtidas pela versão sequencial.

Neste trabalho, desejamos abordar o problema bidimensional de empacotamento de itens irregulares distribuindo o peso da carga. Para isso, propomos duas heurísticas que são adaptações do algoritmo *Bottom-Left*, proposto por [Baker et al., 1980]. Utilizamos também o método de rasterização para representação dos itens. Testamos os algoritmos com instâncias clássicas adaptadas dos problemas de corte de itens irregulares.

Este artigo está organizado da seguinte forma. Na Seção 2 formalizamos o problema

estudado. A Seção 3 apresenta os métodos propostos. Na Seção 4, os experimentos computacionais e por fim, a conclusão na Seção 5.

2. Definição do Problema

O problema de empacotamento bidimensional com distribuição da carga pode ser descrito da seguinte forma: dado um conjunto de n itens irregulares bidimensionais, o peso γ_i e o centro de gravidade $(\lambda_{x,i}, \lambda_{y,i})$ de cada item i ($i = 1, 2, \dots, n$), e um número ilimitado de recipientes retangulares, todos com largura L e comprimento C , com um centro de gravidade desejado, o objetivo é empacotar todos os itens no menor número de recipientes possível, de forma que o centro de gravidade final dos recipientes fique o mais próximo do centro desejado. Para um empacotamento válido, não pode haver sobreposição entre quaisquer dois itens e todos os itens devem estar inteiramente contidos nos recipientes. Neste trabalho consideramos o centro de gravidade desejado (CGd) no centro do recipiente, isto é, $CGd = (C/2, L/2)$ e que os itens possuem distribuição de massa homogênea (ou seja, o centro de massa é dado pelo baricentro do item).

Seja R_j o conjunto dos itens que foram empacotados no recipiente j e m o número de recipientes utilizados. O centro de gravidade final do recipiente j ($j = 1, 2, \dots, m$), $(\lambda_{x,j}, \lambda_{y,j})$, é dado por:

$$\lambda_{x,j} = \frac{\sum_{i \in R_j} \gamma_i \cdot \lambda_{x,i}}{\sum_{i \in R_j} \gamma_i} \quad \lambda_{y,j} = \frac{\sum_{i \in R_j} \gamma_i \cdot \lambda_{y,i}}{\sum_{i \in R_j} \gamma_i}$$

Ainda nessa seção, apresentamos o método de rasterização, utilizado para representar os itens e recipientes, e também, verificar sobreposição entre os itens. Também explanamos sobre a regra de posicionamento utilizada nos métodos propostos.

2.1. Geometria de itens e recipiente

Na literatura, as duas formas mais comuns de se representar os itens irregulares são através de polígonos ou malhas. Uma revisão sobre essas abordagens pode ser vista em [Bennell e Oliveira, 2008]. Para este trabalho, escolhemos a representação por malha, também chamada de método de rasterização.

A rasterização consiste em dividir uma região em áreas discretas. Assim, o retângulo envolvente do item é transformado em uma malha codificada, representada por uma matriz. Utilizamos uma codificação 0-1, proposta por [Oliveira e Ferreira, 1993], que atribui zero às células não ocupadas pelo item e atribui um para as ocupadas. Na Figura 1 temos o exemplo de um item representado pela rasterização 0-1.

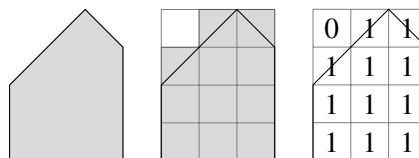


Figura 1: Item representado pela rasterização 0-1.

Note que não apenas o item, mas também o recipiente pode ser representado por esse método. Assim, para recipientes com formato retangular, a matriz é criada e inicializada inteiramente com zeros, isto é, com todas as suas posições livres de item.

Para posicionar um item em determinado ponto do recipiente, soma-se a matriz do item com as posições correspondentes da matriz recipiente. Caso alguma célula resulte em um valor maior que um, então há sobreposição e, portanto, o ponto de alocação escolhido é infactível. Caso contrário, a posição é viável, o item é alocado e a matriz recipiente é atualizada. A Figura 2 ilustra a verificação de sobreposição utilizando a técnica de rasterização.

Consideramos neste trabalho que os itens possuem distribuição de massa homogênea. Logo, todas as células ocupadas da malha de um item tem valor igual. Isso implica que, para se obter o peso de um recipiente basta somar as células ocupadas pelos itens alocados nele.

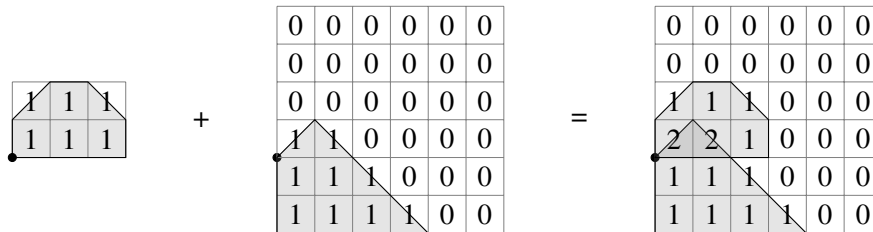


Figura 2: Verificação de sobreposição pelo método de rasterização.

A escolha da unidade da malha requer atenção, já que a precisão da representação dos itens e o tempo para verificar a sobreposição depende da unidade escolhida, de forma que unidades pequenas resultam em itens melhores representados, mas maior tempo para verificação de sobreposição e vice-versa.

2.2. Regra de Posicionamento

Dentre os métodos de resolução para o empacotamento bidimensional com itens irregulares, a regra de posicionamento *Bottom-Left* (BL), proposta por [Baker et al., 1980], é uma heurística construtiva que aloca os itens o mais à esquerda e mais abaixo possível no recipiente (para mais informações sobre outros métodos, veja [Bennell e Oliveira, 2009]). Dessa forma, a BL funciona da seguinte forma: o item a ser alocado é movimentado de baixo para cima e da esquerda para a direita na matriz recipiente, seguindo a ordem mostrada na Figura 3. A cada posição, verifica-se se há sobreposição entre o novo item e os itens já posicionados. A primeira posição em que não houver sobreposição é escolhida para o posicionamento do item e a matriz recipiente é atualizada.

Esta regra de posicionamento pode ser generalizada e aplicada a outras direções, como *Top-Right*, na qual os itens são posicionados o mais à direita e mais acima no recipiente.

4	8	12	16	20	24
3	7	11	15	19	23
2	6	10	14	18	22
1	5	9	13	17	21

Figura 3: Ordem em que os itens são movimentados na regra *Bottom-Left*.

3. Métodos Propostos

Nesta seção, apresentamos duas heurísticas para a resolução do problema de empacotamento considerando o balanceamento da carga. Ambas heurísticas procuram satisfazer a restrição de balanceamento de carga dividindo o recipiente em quatro regiões, que chamaremos de quadrantes, sendo o primeiro quadrante o superior esquerdo; o segundo, o inferior esquerdo; o terceiro quadrante, o inferior direito e, finalmente, o quarto quadrante, o superior direito (Figura 4). O objetivo é distribuir o peso dos itens no recipiente, resultando, assim, num centro de gravidade dos recipientes mais próximo do centro desejado.

Em ambas as heurísticas, para utilizar menos memória computacional, definimos que itens idênticos serão representados como um único item e a quantidade de repetições será chamada de demanda do item. Por exemplo, se temos três itens idênticos, dizemos que este item tem demanda três.

1	4
2	3

Figura 4: Recipiente dividido em quadrantes.

3.1. Heurística BL4

A heurística BL4 tem com entrada uma lista (*listItems*) de n de tipos de itens irregulares a serem empacotados com suas características (dimensões da matriz, matriz e demanda (d_i), com $i = 1, 2, \dots, n$), o comprimento (C) e largura (L) dos recipientes e a unidade (*unid*) para a matriz do recipiente (a mesma utilizada para rasterizar os itens). Como saída tem-se S que é a solução obtida formado pelas posições no recipiente e o recipiente em que cada item foi alocado. Além disso, define-se p um número suficientemente grande para o valor máximo de recipientes.

Primeiro, o algoritmo ordena a lista *listItems* em ordem decrescente em relação ao peso dos itens. Então, ele aloca um mesmo tipo de item até atender a demanda. Essa alocação é feita seguindo a regra *Bottom-Left*, mas revezando entre os quadrantes, escolhendo para alocar o quadrante com o menor peso, ou seja, o quadrante com maior número de células vazias. Em caso de empate, escolhe-se o quadrante de menor índice. Caso um item não caiba no quadrante escolhido, tenta-se colocá-lo no próximo quadrante de menor peso. Caso ele não caiba em nenhum quadrante, passa-se para o próximo recipiente. O processo se repete até encontrar uma posição factível para o item. Depois que um item é alocado, atualizam-se os pesos dos quadrantes com o peso deste item. Para atualizar o peso de um quadrante, basta acrescentar ao peso dele na iteração anterior o número de células ocupadas pelo item neste quadrante multiplicado pelo peso de uma célula ocupada pelo item. Este peso é calculado dividindo o peso do item pelo número de células ocupadas por ele. Como utilizamos malha para representar os itens, o peso de uma célula ocupada por um item é um valor aproximado, já que, utilizando essa representação, nem todas as células são preenchidas por inteiro. Caso um item ocupe dois ou mais quadrantes, seu peso é distribuído entre os quadrantes conforme a área ocupada em cada um. O algoritmo 1 formaliza a heurística BL4.

O seguinte exemplo ilustra o funcionamento da BL4. Deseja-se empacotar cinco itens em recipientes ($C = 5, L = 4$) e *unid* = 0.5. As malhas dos itens são apresentados na Figura 5. As outras características dos itens são: o item 1 possui demanda $d_1 = 2$, peso $\gamma_1 = 11$, centro de gravidade $(\lambda_{x,1}, \lambda_{y,1}) = (1.6667, 1.8867)$; o item 2, $d_2 = 1$, $\gamma_2 = 7$ e $(\lambda_{x,2}, \lambda_{y,2}) = (1.5533, 0.8866)$; o item 3, $d_3 = 2$, $\gamma_3 = 6$ e $(\lambda_{x,3}, \lambda_{y,3}) = (1.5, 1)$; o item 4, $d_4 = 3$, $\gamma_4 = 6$ e $(\lambda_{x,4}, \lambda_{y,4}) = (1, 1.5)$; por fim, o item 5, $d_5 = 1$, $\gamma_5 = 4$ e $(\lambda_{x,5}, \lambda_{y,5}) = (2, 0.5)$.

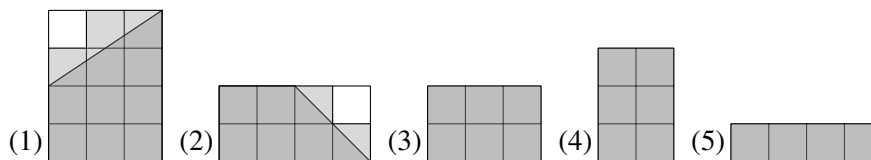


Figura 5: Itens a serem empacotados pelo BL4 no exemplo.

A heurística BL4 começa ordenando a lista de itens em ordem decrescente em relação ao peso dos itens (neste caso, mantém-se a ordem apresentada na Figura 5) e cria o vetor (*mRecip*) da matriz de tamanho 8×10 . Então, o BL4 irá alocar o primeiro item da lista: a primeira unidade do item 1. Como nenhum quadrante tem qualquer item alocado, então o item é alocado na primeira

Algoritmo 1: BL4.

Entrada: $n, listItens, C, L, unid$

n : quantidade de tipos de itens;

$listItens$: lista dos itens contendo para cada tipo de item i ($i = 1, 2, \dots, n$):

dimensões da matriz, matriz e demanda (d_i); C : comprimento dos recipientes;

L : largura dos recipientes;

$unid$: unidade da malha

Saída: S .

S : Solução obtida pelo algoritmo.

```

1   $S \leftarrow \emptyset$ ;
2   $i \leftarrow 0$ ;
3   $p \leftarrow$  número suficiente paragrafo rosa da introdmente grande de recipientes;
4   $listItens$  é ordenado em ordem decrescente em relação ao peso dos itens;
5  Crie uma lista ( $mRecip$ ) com  $p$  matrizes recipiente com as dimensões
   ( $C/unid, L/unid$ );
6  para  $i \leq n$  faça
7      enquanto  $j$  de 1 a  $d_i$  faça
8          para  $k$  de 1 a  $p$  faça
9              Ordene os quadrantes do recipiente  $k$  em relação ao peso;
10             para  $l$  de 1 a 4 faça
11                 Busque um ponto factível  $(x, y)$  no  $l$ -ésimo quadrante de menor
                    peso do recipiente  $mRecip[k]$  para o item  $i$  pela regra de
                    posicionamento Bottom-Left.
12                 se Existe uma posição factível para o item  $i$  então
13                     A solução  $S$  recebe o item  $i$  do tipo  $i$  alocado no recipiente  $k$  no
                         $l$ -ésimo quadrante de menor peso na posição  $(x, y)$ 
14                     Atualize a malha do recipiente  $k$  e o peso de cada quadrante;
15                      $j \leftarrow j + 1$ ;
16                     Vá para a linha 7;
17                 fim
18             fim
19         fim
20          $j \leftarrow j + 1$ ;
21     fim
22 fim
23 devolva  $S$ 

```

posição do primeiro quadrante. O peso do primeiro quadrante é atualizado para 11 (o peso do item, que foi inteiramente inserido no primeiro quadrante). O próximo item da lista é a segunda unidade do item 1. O quadrante de menor peso (e menor índice) é o segundo quadrante. Como não há nenhum nele item alocado, o item é alocado na primeira posição e o peso do segundo quadrante é atualizado para 11. Da mesma forma, o terceiro item da lista, a primeira unidade do item 2, é alocado na primeira posição do terceiro quadrante, e o peso do quadrante é atualizado para 7. E o quarto item da lista, a primeira unidade do item 3, deve ser alocado no quadrante de menor peso, que é o quarto quadrante (peso zero). Assim, o quarto item da lista é alocado na primeira posição do quadrante e seu peso é atualizado para 6.

Já para o quinto item da lista, a segunda unidade do item 3, o algoritmo primeiro testa a primeira posição do quadrante de menor peso, o quarto quadrante (peso 6). Como a primeira

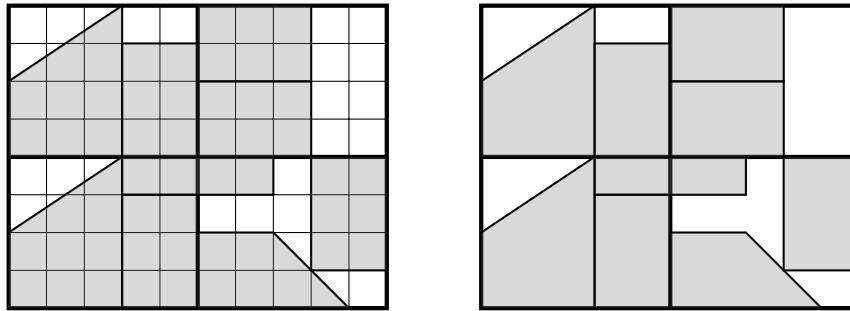


Figura 6: Solução encontrada pela heurística BL4 para o exemplo.

posição está ocupada por outro item, o algoritmo verifica outras posições, até encontrar uma factível. Assim, o algoritmo aloca o item logo acima do item anteriormente posicionado e o peso do quadrante é atualizado para 12. O sexto item da lista, a primeira unidade do item 4, deve ser alocado no quadrante 3 (peso 7) e o item é alocado à direita do item que já estava alocado neste quadrante. O peso do quadrante é atualizado para 13. O sétimo item da lista, a segunda unidade do item 4, é alocado no primeiro quadrante (peso 11) à direita do item já nele alocado. O peso do primeiro quadrante é atualizado para 17. O oitavo item da lista, a terceira unidade do item 4, é alocado no segundo quadrante, de peso 9, à direita já nele alocado. Por fim, o último item da lista, a primeira unidade do item 5, deve ser alocado no quadrante de menor peso, o quadrante de número quatro. Não havendo nenhuma posição factível neste quadrante, o algoritmo procura alguma posição factível no segundo quadrante de menor peso, o terceiro quadrante. Não havendo nenhuma posição, o algoritmo passa a procurar alguma posição factível no próximo quadrante de menor peso, que pelo critério de desempate é o primeiro quadrante. Novamente, o algoritmo não encontra nenhuma posição factível. Assim, o algoritmo busca alguma posição no segundo quadrante. Finalmente, o último item é alocado em parte do segundo quadrante e a outra parte no terceiro quadrante. E os pesos do segundo e terceiro quadrante são atualizados para um e cinco, respectivamente. Caso houvesse a demanda de uma segunda unidade do quinto item, esta seria alocada em outro recipiente, já que não haveria qualquer posição factível para o item no recipiente atual.

A Figura 6 ilustra a solução obtida pelo algoritmo. O centro de gravidade do recipiente após a alocação dos itens passa a ser $(\lambda_x, \lambda_y) = (4.4848, 3.6541)$.

3.2. Heurística BL4SWIRL

A heurística BL4SWIRL assemelha-se muito à heurística BL4, diferindo apenas na regra de posicionamento utilizada (linha 11 do Algoritmo 1). Nesta heurística, para cada quadrante há uma regra de posicionamento diferente. A BL4SWIRL imita o comportamento de uma espiral, com o objetivo de distribuir melhor os itens no recipiente. Assim, no primeiro quadrante, aloca-se o item o mais à esquerda e acima possível; no segundo quadrante, o mais abaixo e à esquerda; no terceiro, o mais à direita e mais abaixo e, por fim, no quarto quadrante, o mais acima e mais à direita. A Figura 7 ilustra as regras de posicionamento em cada quadrante.

4. Experimentos computacionais

Nesta seção, apresentamos os resultados obtidos pelos métodos propostos. Implementamos os métodos em linguagem C e os experimentos computacionais foram realizados em um computador Intel Core i7-5500U com 16GB de RAM usando o sistema operacional Ubuntu 14.04 LTS. Na Seção 4.1 apresentamos as instâncias utilizadas e, na Seção 4.2, os resultados obtidos.

4.1. Instâncias

Até onde vai nosso conhecimento, não existem instâncias disponíveis com as características do problema tratado neste trabalho. Assim, optamos por adaptar algumas instâncias

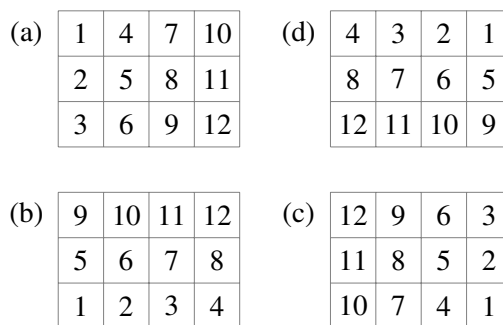


Figura 7: Regras de posicionamento em cada quadrante na heurística BL4SWIRL: (a) 1º quadrante, (b) 2º quadrante, (c) 3º quadrante, (d) 4º quadrante.

clássicas do problema de corte irregular bidimensional. As instâncias estão disponíveis *online* na página da *EURO Special Interest Group on Cutting and Packing*¹. As instâncias utilizadas foram originalmente criadas para o problema de corte com uma dimensão aberta, logo, para cada instância, apenas a largura (L) dos recipientes é definida. Para definir o comprimento, utilizamos como referência o menor comprimento (c) encontrado na literatura até o momento [Elkeran, 2013], arredondado para o inteiro mais próximo. Algumas características das instâncias estão descritas na Tabela 1, bem como a largura (L) e o comprimento de referência (c) para cada instância. As unidades escolhidas para as malhas se mostraram as mais adequadas em testes preliminares.

Tabela 1: Descrição das instâncias.

Instância	Nº de polígonos diferentes	Nº total de polígonos	L	c	Unidade da malha
Albano	8	24	4900	9718	25
Dagli	10	30	60	46	0.1
Dighe1	16	16	100	100	0.5
Dighe2	10	10	100	100	0.5
Fu	12	12	38	31	0.5
Jakobs1	25	25	40	11	0.1
Jakobs2	25	25	70	22	0.1
Shapes0	4	43	40	58	0.5

Para cada uma das instâncias apresentadas na Tabela 1, definimos três instâncias diferentes, que variam no comprimento (C) dos recipientes utilizados. Estes comprimentos foram: c , $1,5c$ e $3c$. Definimos o peso de cada item como o valor de sua área.

4.2. Resultados

Nesta seção apresentamos os resultados obtidos pelas heurísticas propostas. Utilizamos a notação $instância_\alpha$ para representar o comprimento do recipiente para a $instância$, sendo $C = \alpha \times c$, com $\alpha = 1; 1,5; 3$.

Na Tabela 2, temos na primeira coluna o nome da instância utilizada; das colunas dois a seis temos os resultados da heurística BL4; das colunas

sete a dez os resultados da heurística BL4SWIRL, sendo m o número de recipientes utilizados para empacotar os itens, t o tempo computacional, em segundos,

$$\eta = \frac{\|CGd - (\lambda_{x,j}, \lambda_{y,j})\|}{\|CGd\|},$$

utilizando a norma euclidiana, e ocp é taxa de ocupação, que é calculada como a área dos itens alocados no recipiente dividida pela área do recipiente, em porcentagem. É fácil ver que quanto

¹<http://paginas.fe.up.pt/~esicup/datasets>, último acesso em 13 de abril de 2016.

menor o valor do η , mais perto o centro de gravidade final $(\lambda_{x,j}, \lambda_{y,j})$ está do centro de gravidade desejado (CGd). Na tabela, destacamos o menor valor para η dentre os maiores valores produzidos pelas heurísticas BL4 e BL4SWIRL.

Tabela 2: Resultados numéricos.

Instâncias	BL4					BL4SWIRL				
	m	$t(s)$	η - recip1	η - recip2	$ocp(\%)$	m	$t(s)$	η - recip1	η - recip2	$ocp(\%)$
Albano ₃	1	0.03	0.3113	-	29.86	1	0.09	0.1035	-	29.86
Albano _{1.5}	1	0.02	0.2376	-	59.72	1	0.11	0.2070	-	59.72
Albano ₁	2	0.39	0.2923	0.4428	44.79	2	0.33	0.2869	0.4996	44.79
Dagli ₃	1	0.45	0.3571	-	29.86	1	2.90	0.2144	-	29.86
Dagli _{1.5}	1	0.45	0.3600	-	59.71	1	2.99	0.3296	-	59.71
Dagli ₁	2	6.25	0.4175	0.4211	44.78	2	4.77	0.4232	0.2926	44.78
Dighe _{1.3}	1	0.05	0.3374	-	33.33	1	0.16	0.2466	-	33.33
Dighe _{1.5}	2	0.12	0.4638	0.7128	33.33	2	0.17	0.4444	0.6854	33.33
Dighe _{1.1}	2	0.08	0.6051	0.5365	50.00	2	0.16	0.6281	0.4835	50.00
Dighe _{2.3}	1	0.02	0.3717	-	33.33	1	0.06	0.2988	-	33.33
Dighe _{2.1.5}	2	0.06	0.6091	0.6717	33.33	2	0.09	0.5215	0.5821	33.33
Dighe _{2.1}	2	0.05	0.7871	0.5536	50.00	2	0.04	0.7252	0.6159	50.00
Fu ₃	1	0.00	0.3630	-	30.64	1	0.00	0.2175	-	30.64
Fu _{1.5}	1	0.00	0.3888	-	61.29	1	0.00	0.3539	-	61.29
Fu ₁	2	0.00	0.5031	0.5671	45.97	2	0.01	0.4442	0.5145	45.97
Jakobs _{1.3}	1	0.01	0.2867	-	29.39	1	0.05	0.1989	-	29.39
Jakobs _{1.1.5}	1	0.02	0.2717	-	58.79	1	0.08	0.2354	-	58.79
Jakobs _{1.1}	2	0.06	0.2552	0.4544	44.09	2	0.00	0.2679	0.0720	44.09
Jakobs _{2.3}	1	0.35	0.2899	-	29.24	1	0.56	0.2091	-	29.24
Jakobs _{2.1.5}	1	0.65	0.2774	-	58.48	1	0.72	0.2662	-	58.48
Jakobs _{2.1}	2	0.71	0.3507	0.3911	43.86	2	1.01	0.2747	0.1833	43.86
Shapes _{0.3}	1	0.01	0.2995	-	22.93	1	0.01	0.0910	-	22.93
Shapes _{0.1.5}	1	0.00	0.1839	-	45.86	1	0.01	0.1574	-	45.86
Shapes _{0.1}	2	0.02	0.2188	0.3270	34.40	2	0.01	0.2203	0.1358	34.40

Ambas heurísticas utilizaram a mesma quantidade de recipientes em todas as instâncias, então a taxa de ocupação também é a mesma. As heurísticas utilizam comprimento maior (mais recipientes) que o algoritmo de [Elkeran, 2013], até então, o trabalho com os melhores resultados da literatura.

Em relação ao centro de gravidade final, a heurística BL4SWIRL obteve melhores valores para quase todas as instâncias. As instâncias em que a BL4 obteve melhores resultados são instâncias com o menor comprimento ($\alpha = 1$). Já a BL4SWIRL encontra melhores resultados para recipientes maiores ($\alpha = 3$), como na instância Shapes_{0.3} em que a BL4SWIRL obteve o menor η de todas as instâncias (Figura 8). Apesar disso, os valores de η encontrados pelas heurísticas são bastante altos. A média dos valores de η encontrado pela heurística BL4 é 0.4093 e pela heurística BL4SWIRL é 0.3362. Assim, o balanceamento da carga executada pelas heurísticas é bastante aquém do desejado.

Sobre o tempo de execução, ambas heurísticas são muito rápidas e têm tempo de execução comparáveis.

Na Figura 9, temos o resultado das duas heurísticas para a instância Fu_{1.5}, que é a instância com a maior taxa de ocupação. À esquerda, a solução encontrada pela heurística BL4 e a direita, pela heurística BL4SWIRL.

5. Conclusão

Neste trabalho apresentamos duas heurísticas para resolver o problema de empacotamento irregular bidimensional com distribuição de carga. Os objetivos foram minimizar o número de recipientes utilizados para empacotar os itens e minimizar a distância do centro de gravidade final ao centro de gravidade desejado dos recipientes. Utilizamos o método de rasterização para representar

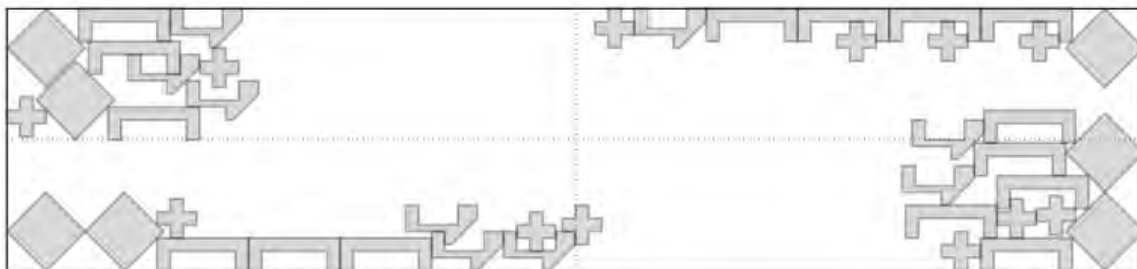


Figura 8: Instância que apresentou menor η - Shapes0₃.

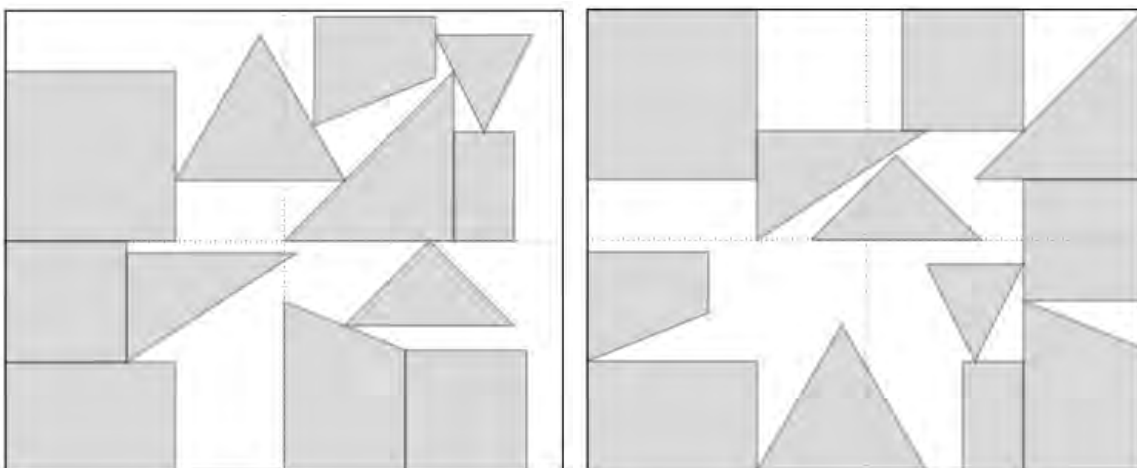


Figura 9: Instância com maior taxa de ocupação - Fu_{1,5}.

os itens e os recipientes, e para verificar a sobreposição entre os itens. As heurísticas propostas utilizaram a estratégia de dividir o recipiente em quadrantes e revezar a alocação dos itens entre eles com o objetivo de distribuir o peso no recipiente.

Utilizamos oito instâncias clássicas do problema de corte irregular bidimensional adaptadas para o nosso problema e cada instância foi testada com três tipos de recipientes diferentes. Ambas heurísticas utilizaram a mesma quantidade de recipientes para todas as instâncias, mas a BL4SWIRL apresentou melhores resultados em relação à distribuição de peso na maior parte das instâncias, apesar disso, o balanceamento produzido pelas heurísticas está aquém do desejado. O tempo computacional de ambas as heurísticas foi baixo.

No futuro, pretendemos obter dados reais para testar e analisar o desempenho das heurísticas. Também pretendemos desenvolver uma heurística para melhorar o balanceamento produzido pelas heurísticas e uma meta-heurística que utiliza as heurísticas propostas, a fim de obter melhores resultados.

Agradecimentos

Este trabalho contou com o apoio da CAPES, FAPESP (bolsa de Mestrado: 2015/16298-4 e CEPID: 2013/07375-0) e Universal do CNPq (processo 476792/2013-4). Agradecemos aos revisores pelos comentários importantes, que tornaram este trabalho melhor.

Referências

- Baker, B. S., Coffman, E. G., Jr, e Rivest, R. L. (1980). Orthogonal packings in two dimensions. *SIAM Journal on Computing*, 9(4):846–855.
- Bennell, J. A. e Oliveira, J. F. (2008). The geometry of nesting problems: A tutorial. *European Journal of Operational Research*, 184(2):397–415.

- Bennell, J. A. e Oliveira, J. F. (2009). A tutorial in irregular shape packing problems. *Journal of the Operational Research Society*, p. S93–S105.
- Davies, A. P. e Bischoff, E. E. (1999). Weight distribution considerations in container loading. *European Journal of Operational Research*, 114(3):509–527.
- Egeblad, J. (2009). Placement of two-and three-dimensional irregular shapes for inertia moment and balance. *International Transactions in Operational Research*, 16(6):789–807.
- Egeblad, J., Nielsen, B. K., e Odgaard, A. (2007). Fast neighborhood search for two-and three-dimensional nesting problems. *European Journal of Operational Research*, 183(3):1249–1266.
- Elkeran, A. (2013). A new approach for sheet nesting problem using guided cuckoo search and pairwise clustering. *European Journal of Operational Research*, 231(3):757–769.
- Fernández, A., Gil, C., Baños, R., e Montoya, M. G. (2013). A parallel multi-objective algorithm for two-dimensional bin packing with rotations and load balancing. *Expert Systems with Applications*, 40(13):5169–5180.
- Kaluzny, B. L. e Shaw, R. (2009). Optimal aircraft load balancing. *International Transactions in Operational Research*, 16(6):767–787.
- Liu, D., Tan, K. C., Huang, S., Goh, C. K., e Ho, W. K. (2008). On solving multiobjective bin packing problems using evolutionary particle swarm optimization. *European Journal of Operational Research*, 190(2):357–382.
- Liu, J., Yue, Y., Dong, Z., Maple, C., e Keech, M. (2011). A novel hybrid tabu search approach to container loading. *Computers & Operations Research*, 38(4):797–807.
- Mongeau, M. e Bes, C. (2003). Optimization of aircraft container loading. *Aerospace and Electronic Systems, IEEE Transactions on*, 39(1):140–150.
- Oliveira, J. F. C. e Ferreira, J. A. S. (1993). Algorithms for nesting problems. In *Applied simulated annealing*, p. 255–273. Springer.