***Empacotamento de itens
irregulares considerando o
balanceamento da carga***

**Raquel Akemi Okuno
Kitazume da Silva**
Dissertação de Mestrado

SERVIÇO DE PÓS-GRADUAÇÃO DO ICMC-USP

Data de Depósito:

Assinatura: _____

Raquel Akemi Okuno Kitazume da Silva

Empacotamento de itens irregulares considerando balanceamento da carga

Dissertação apresentada ao Instituto de Ciências Matemáticas e de Computação – ICMC-USP, como parte dos requisitos para obtenção do título de Mestra em Ciências – Ciências de Computação e Matemática Computacional. *VERSÃO REVISADA*

Área de Concentração: Ciências de Computação e Matemática Computacional

Orientadora: Profa. Dra. Marina Andretta

USP – São Carlos
Julho de 2017

Ficha catalográfica elaborada pela Biblioteca Prof. Achille Bassi
e Seção Técnica de Informática, ICMC/USP,
com os dados fornecidos pelo(a) autor(a)

S586e Silva, Raquel Akemi Okuno Kitazume da
Empacotamento de itens irregulares considerando
balanceamento da carga / Raquel Akemi Okuno
Kitazume da Silva; orientadora Marina Andretta. --
São Carlos, 2017.
111 p.

Dissertação (Mestrado - Programa de Pós-Graduação
em Ciências de Computação e Matemática
Computacional) -- Instituto de Ciências Matemáticas
e de Computação, Universidade de São Paulo, 2017.

1. Problemas de empacotamento. 2.
empacotamentode itens irregulares. 3. recipientes
regulares eirregulares. 4. balanceamento de carga.
5. heurísticas. I. Andretta, Marina, orient. II.
Título.

Raquel Akemi Okuno Kitazume da Silva

Irregular bin packing considering load balance

Master dissertation submitted to the Instituto de Ciências Matemáticas e de Computação – ICMC-USP, in partial fulfillment of the requirements for the degree of the Master Program in Computer Science and Computational Mathematics. *FINAL VERSION*

Concentration Area: Computer Science and Computational Mathematics

Advisor: Profa. Dra. Marina Andretta

USP – São Carlos
July 2017

Ao meu marido, Marcelo, fonte de inspiração nos estudos e na vida.

AGRADECIMENTOS

A Deus, sem o qual nada do que existe teria sido feito.

A meu marido, Marcelo, por todo o amor, pelo incentivo aos estudos, pela cumplicidade em todos os momentos.

Aos meus pais, Jun e Zeni, pelas palavras de sabedoria, direção e apoio.

Aos meus irmãos, Samuel e Esther, pela amizade e companheirismo em todos os momentos.

À minha orientadora, Marina Andretta, pelas ideias, sugestões e por todo o tempo investido em mim e nesse trabalho.

Aos membros do Laboratório de Otimização por toda a ajuda e conselhos dados durante esse mestrado.

À Fundação de Amparo à Pesquisa do Estado de São Paulo (FAPESP), pelo apoio financeiro na bolsa de mestrado (processo n^o 2015/16298-4).¹

¹ As opiniões, hipóteses e conclusões ou recomendações expressas neste material são de responsabilidade do autor e não necessariamente refletem a visão da FAPESP.

*“Entrega o teu caminho ao Senhor,
confia nele e o mais ele fará.”
Salmo 37.5 .*

RESUMO

KITAZUME, R. A. O. **Empacotamento de itens irregulares considerando balanceamento da carga**. 2017. 111 p. Dissertação (Mestrado em Ciências – Ciências de Computação e Matemática Computacional) – Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, São Carlos – SP, 2017.

O problema de empacotamento de itens irregulares com balanceamento da carga é encontrado no carregamento de aviões, caminhões e navios. O objetivo é empacotar itens irregulares utilizando o menor número de recipientes possível de forma que os recipientes estejam balanceados, que os itens não se sobreponham e estejam inteiramente contidos no recipiente. Neste trabalho, propomos três heurísticas bases com três variações cada para o problema com recipientes retangulares e irregulares. As heurísticas utilizam abordagens diferentes para representar os itens e para fazer o balanceamento. Uma das heurísticas utiliza malha para representação dos itens e faz o balanceamento dividindo o recipiente em quadrantes e revezando a alocação dos itens entre eles de forma que o balanceamento é feito de forma indireta. Tal heurística resolve o problema tanto para recipientes retangulares quanto irregulares. A segunda heurística utiliza a representação dos itens por polígonos e impossibilita a sobreposição de itens utilizando a técnica do *nofit polygon*. A heurística constrói a solução item por item, sem posições fixas e a cada item alocado, os itens são deslocados em direção ao centro de gravidade desejado do recipiente. Esta heurística resolve apenas problemas com recipientes retangulares. A última heurística é uma adaptação da heurística anterior para a resolução do problema com recipientes irregulares, de forma que o problema é resolvido em duas fases. Cada heurística base possui três variações cada, totalizando nove heurísticas. As heurísticas foram comparadas com outro trabalho da literatura e conseguiram melhorar os resultados para nove das dezenove instâncias testadas.

Palavras-chave: Problemas de empacotamento, empacotamento de itens irregulares, recipientes retangulares e irregulares, balanceamento de carga, heurísticas.

ABSTRACT

KITAZUME, R. A. O. **Irregular bin packing considering load balance**. 2017. 111 p. Dissertação (Mestrado em Ciências – Ciências de Computação e Matemática Computacional) – Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, São Carlos – SP, 2017.

The irregular bin packing problem with load balancing is found in the loading of airplanes, trucks and ships. The aim is to use as few bins as possible to pack all the items so that all bins are balanced, items do not overlap and are fully contained in the bin. In this work, we propose three base heuristics with three variations each for the problem with rectangular and irregular bin. The three heuristics use different approaches to represent the items and to balance the bin. One of the heuristics uses a grid to represent the items and does the balancing by dividing the container into quadrants and alternating the allocation of items between them so that the balancing is done indirectly. Such heuristic solves the problem for both rectangular and irregular bins. The second heuristic uses the representation of items by polygons and uses the nofit polygon technique. The heuristic constructs the solution item by item, with no fixed positions and with each item allocated, the items are shifted towards the desired center of gravity of the bin. This heuristic only solves problems with rectangular bins. The last heuristic is an adaptation of the previous one to solve the problem with irregular bins, so that the problem is solved in two phases. Each base heuristic has three variations, totaling nine heuristics. The heuristics were compared with other work in the literature and managed to improve the results for nine of the nineteen instances tested.

Keywords: Packing problems, irregular bin packing, regular and irregular bins, load balance, heuristics.

SUMÁRIO

1	INTRODUÇÃO	17
2	DEFINIÇÃO DO PROBLEMA	19
2.1	Importância do balanceamento de carga	20
2.2	Geometria dos itens e recipientes	21
3	REVISÃO BIBLIOGRÁFICA	29
3.1	Corte de itens irregulares	30
3.1.1	<i>Problemas de empacotamento em faixa</i>	31
3.1.2	<i>Problemas de empacotamento em recipientes fechados</i>	35
3.2	Problemas de empacotamento considerando o peso dos itens	37
3.3	Considerações	40
4	HEURÍSTICAS PROPOSTAS	41
4.1	Heurística QuadInside e suas variações	41
4.1.1	<i>Heurística Quad-BottomLeft</i>	47
4.1.2	<i>Heurística QuadSwirl</i>	47
4.2	Heurística Desloc	48
4.2.1	<i>Heurística Desloc-Irregular</i>	50
5	EXPERIMENTOS COMPUTACIONAIS	55
5.1	Instâncias	55
5.2	Resultados numéricos	57
5.2.1	<i>Comparação entre as heurísticas discretas</i>	57
5.2.2	<i>Comparação entre as heurísticas contínuas</i>	60
5.2.3	<i>Comparação entre as heurísticas propostas</i>	62
5.2.4	<i>Comparação com a literatura</i>	63
6	CONSIDERAÇÕES E TRABALHOS FUTUROS	65
	REFERÊNCIAS	67
	APÊNDICE A DESCRIÇÃO DAS INSTÂNCIAS	73
A.1	Instância Albano	73

A.2	Instância Dagli	74
A.3	Instância Fu	75
A.4	Instância Marques	76
A.5	Instância Shapes0	77
A.6	Instância Shapes2	78
A.7	Instância Shirts	79
A.8	Instância Swim	80
A.9	Instância Trousers	81
A.10	Instância Car	82
A.11	Instância Car-dagli	83
A.12	Instância Car-fu	84
APÊNDICE B SOLUÇÕES OBTIDAS		85
B.1	Heurística SDesloc	85
B.2	Heurística ADesloc	93
B.3	Heurística RDesloc	101
B.4	Heurística SDesloc-Irregular	109
B.5	Heurística ADesloc-Irregular	110
B.6	Heurística RDesloc-Irregular	111

INTRODUÇÃO

O problema de empacotamento com distribuição de peso é um problema que busca um arranjo de objetos menores, os itens, dentro de objetos maiores, os recipientes, com o objetivo de minimizar o desbalanceamento dos recipientes de forma que se utilize o menor número de recipientes possível. Os itens não podem se sobrepor e todos os itens devem estar inteiramente contidos no recipiente. Os recipientes podem possuir formato retangular ou irregular. Esse problema tem várias aplicações, como o carregamento de aviões, caminhões e navios. O desbalanceamento em aviões e em caminhões pode contribuir para um gasto maior de combustível e para o aumento do risco de acidentes ([AMIOUNY *et al.*, 1992](#); [MONGEAU; BES, 2003](#)). Já o desbalanceamento em navios pode causar avarias na estrutura do navio ([WILSON; ROACH, 1999](#)).

Desejamos abordar o problema de empacotamento de itens irregulares distribuindo o peso da carga como um problema bidimensional. Dessa forma, vamos considerar apenas a representação dos itens nos eixos x e y , assumindo que os itens não podem ser empilhados ou que o empilhamento dos itens foi feito em uma fase de pré-processamento ([KALUZNY; SHAW, 2009](#); [EGEBLAD, 2009](#)).

Neste trabalho, discorreremos sobre as ferramentas mais utilizadas para evitar a sobreposição de itens irregulares e garantir a contenção dos itens dentro de um recipiente, e fazemos um levantamento da literatura de problemas relacionados, como o problema empacotamento de itens com balanceamento de carga e o problema de empacotamento de itens irregulares.

Também apresentamos heurísticas para a resolução do problema tanto para recipientes retangulares quanto para recipientes irregulares. As heurísticas utilizam ferramentas diferentes para empacotar os itens e para fazer o balanceamento. Apresentamos três heurísticas bases com três variações cada. Uma das heurísticas trabalha com ambos os tipos de recipientes, a outra apenas com recipientes retangulares e a última é uma adaptação da anterior para recipientes

irregulares.

Na heurística para ambos os tipos de recipiente, utilizamos a estratégia de dividir o recipiente em quadrantes e a alocação dos itens é feita revezando entre eles, com foco na distribuição de peso. A heurística para recipientes retangulares constrói a solução sem posições fixas, como um bloco de itens flutuante, e o bloco é deslocado em direção ao centro de gravidade desejado. A última heurística é uma adaptação da anterior para o empacotamento em recipientes irregulares. As heurísticas bases e suas variações, que totalizaram nove heurísticas, foram testadas com dezenove instâncias e comparadas com o trabalho de [Egeblad \(2009\)](#). Algumas heurísticas obtiveram melhores resultados para nove das dezenove instâncias testadas.

O trabalho está estruturado da seguinte forma: no [Capítulo 2](#), formalizamos o problema, explanamos sobre suas aplicações e apresentamos as ferramentas geométricas mais utilizadas na literatura para verificar a sobreposição e contenção dos itens. No [Capítulo 3](#), apresentamos uma revisão bibliográfica dos métodos para a resolução de problemas relevantes ao nosso estudo. No [Capítulo 4](#), apresentamos os métodos heurísticos propostos e, no [Capítulo 5](#), são apresentados os resultados computacionais obtidos e a comparação com a literatura. No [Capítulo 6](#), apresentamos nossas considerações e nossas propostas para trabalhos futuros. Por fim, no Apêndice [A](#) descrevemos as instâncias utilizadas e no Apêndice [B](#) apresentamos as principais soluções encontradas.

DEFINIÇÃO DO PROBLEMA

O problema de empacotamento bidimensional com balanceamento de carga é definido 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, com um centro de gravidade desejado (CGd), o objetivo é empacotar todos os itens no menor número de recipientes possível, de forma que o centro de gravidade final de cada recipiente fique o mais próximo do centro desejado. Os recipientes podem possuir formato retangular ou irregular. A localização do CGd varia de acordo com a aplicação do problema. Em um empacotamento válido não pode haver sobreposição entre os itens e todos os itens devem estar inteiramente contidos nos recipientes.

Seja R_j o conjunto de índices dos itens que foram empacotados no recipiente j e m o número de recipientes utilizados. O centro de gravidade final do recipiente j , com $j = 1, 2, \dots, m$, $(\lambda_{x,j}, \lambda_{y,j})$, é dado por (EGEBLAD, 2009):

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

Neste trabalho, assumimos que itens possuem distribuição de massa homogênea. Logo, o centro de gravidade de cada item coincide com seu centro geométrico.

O balanceamento de carga tem grande importância no empacotamento de itens em caminhões, aviões, navios e até em contêineres. Na literatura, pode-se encontrar o balanceamento de carga descrito como distribuição de peso ou estabilidade do recipiente. A importância do balanceamento de carga é discutida na Seção 2.1. Já as restrições de não sobreposição e contenção são comuns a quase todos os problemas de corte e empacotamento, mas nos problemas que envolvem formas irregulares essas restrições se mostram particularmente complicadas, exigindo ferramentas computacionais refinadas para manipular os itens. Na Seção 2.2, apresentamos as ferramentas mais comuns para verificar sobreposição e contenção dos itens.

2.1 Importância do balanceamento de carga

O balanceamento de carga, segundo [Bortfeldt e Wäscher \(2013\)](#), consiste em distribuir o peso da carga sobre a superfície do recipiente, com o objetivo de deixar o recipiente o mais estável possível. Assim, deseja-se que o centro de gravidade final do recipiente esteja o mais perto possível do CGd. Nesta seção, levantamos a importância do balanceamento nas diversas aplicações que encontramos na literatura.

Encontrado no carregamento de aviões, caminhões, navios e contêineres, o balanceamento de carga tem grande importância na prevenção de danos ao recipiente, eficiência do transporte e outros, dependendo do recipiente utilizado. No carregamento de aviões, o arranjo da carga afeta a força de arrasto exercida sobre a aeronave ([Figura 1](#)). Assim, quando a carga está melhor balanceada, o voo é mais seguro, mais rápido e utiliza menos combustível. [Mongeau e Bes \(2003\)](#) afirmam que, para aviões de longo alcance em uma viagem de 10.000 km, um deslocamento menor que 75 cm do centro de gravidade ideal gera uma economia de 4.000 kg de combustível. Em relação à segurança, a má distribuição da carga, em casos extremos, pode levar o avião a ficar dinamicamente instável e, se não corrigido a tempo, pode resultar na queda do avião.

Figura 1 – Força de arrasto exercida sobre a aeronave durante o voo.

Fonte: <http://sciencelearn.org.nz/> - Último acesso: 24/06/2016.

No caso de contêineres, quando eles são transportados por embarcações, a má distribuição da carga pode levar o contêiner a tombar quando empilhado e dificultar ou impossibilitar o guindaste de manipulá-lo.

Para caminhões, há regulamentações no Brasil e outros países que limitam o peso por eixo para prevenir danos às rodovias. Assim, balancear a carga auxilia na redução do peso máximo por eixo, contribui para um menor consumo de combustível e evita o desgaste desnecessário dos pneus ([AMIOUNY et al., 1992](#)). Outra preocupação no carregamento de caminhões é que o centro de gravidade esteja o mais abaixo possível para diminuir a chance do caminhão tombar em curvas e também reforça restrições de fragilidade, já que itens pesados devem ser alocados mais abaixo e itens mais leves acima dos pesados.

No carregamento de navios, segundo [Wilson e Roach \(1999\)](#), a carga deve ser balanceada para evitar banda (ou adernamento), quando o navio inclina para bombordo ou estibordo ([Figura 2](#)), e também assegurar que o caimento, isto é, a inclinação longitudinal, esteja próximo

Figura 2 – Ilustração de banda na embarcação.

(a) Embarcação aprumada.

(b) Embarcação com banda.

Fonte: <http://www.boatdesign.net> - Último acesso: 24/06/2016.

do zero. Além disso, a carga má distribuída pode distorcer a estrutura física do navio, causando alquebramento ou contra-alquebramento (Figura 3).

Figura 3 – Distorções na embarcação.

(a) Alquebramento na embarcação.

(b) Contra-alquebramento na embarcação.

Fonte: <http://www.boatdesign.net> - Último acesso: 24/06/2016.

2.2 Geometria dos itens e recipientes

O problema que desejamos resolver é a versão bidimensional do problema de empacotamento de itens irregulares com balanceamento de carga. A simplificação do caso tridimensional para o bidimensional é feita considerando apenas a representação dos itens nos eixos x e y , ignorando o eixo z . Assim, considera-se que os itens não podem ser empilhados ou que qualquer empilhamento dos itens foi feito em uma fase de pré-processamento.

A forma irregular dos itens é um desafio do problema, já que é preciso verificar sobreposição entre itens e se os itens estão inteiramente contidos no recipiente. Assim, precisamos de ferramentas para representar a forma irregular dos itens, verificar sobreposição e contenção dos itens. Nessa seção, apresentamos uma breve revisão sobre as ferramentas mais utilizadas na literatura.

As formas mais comuns de se representar os itens são por malha ou por aproximação por polígonos irregulares. Ao utilizar a representação por polígonos (Figura 4), a verificação de sobreposição pode ser feita utilizando *nofit polygon*, função ϕ ou trigonometria direta. Uma revisão sobre essas ferramentas pode ser vista em Bennell e Oliveira (2008).

A trigonometria direta verifica a sobreposição dos itens através de seus vértices e arestas. Assim, a ideia é verificar, item por item, se há intersecção de arestas e inclusão de pontos. A fim

Figura 4 – Item representado por aproximação poligonal.

(a) Forma do item original

(b) Representação do item por polígono.

Fonte: Baldacci *et al.* (2014).

de agilizar o processo de verificação de sobreposição, Bennell e Oliveira (2008) apresentam a seguinte hierarquia de testes:

1. Os retângulos envolventes dos itens se sobrepõem (Figura 5a)? Se não, os itens não se sobrepõem. Se sim, ir para o teste 2;
2. Para cada par de arestas dos itens, os retângulos envolventes delas se sobrepõem (Figura 5b)? Se não, os itens não se sobrepõem. Se sim, aplicar o teste 3 para as arestas que resultaram em sobreposição dos retângulos envolventes.
3. Para cada par de arestas dos itens, há sobreposição entre elas (Figura 5c)? Se sim, os itens se sobrepõem. Se não para todas as arestas, aplicar o teste 4.
4. Para cada vértice de cada item, esse vértice está dentro do outro item? Se sim, os itens se sobrepõem (Figura 5d). Se não, os polígonos não se sobrepõem (Figura 5e).

Figura 5 – Posições relativas identificadas pela série de testes.

Figura 6 – Fonte: Bennell e Oliveira (2008).

A função ϕ , introduzida por Stoyan, Novozhilova e Kartashov (1996), é uma expressão matemática que representa a posição relativa de dois itens. Assim, o valor da função ϕ é maior que zero se os itens estão separados; igual a zero, caso os itens estejam se tocando; e menor que

zero, caso eles se sobreponham. A Figura 7 ilustra representações de funções ϕ entre figuras geométricas. A linha mais grossa representa o local geométrico onde a função ϕ vale zero. Essa ferramenta é muito eficiente, mas não existe nenhum algoritmo capaz de gerar a função ϕ para qualquer item. Isso explica o fato dessa ferramenta não ser tão utilizada na literatura.

Figura 7 – Representação de funções ϕ .

(a) Função ϕ entre um retângulo e um círculo.

(b) Função ϕ entre dois círculos.

Fonte: Stoyan *et al.* (2001).

A técnica do *nofit polygon* define uma região de sobreposição entre dois itens. Assim, dados dois itens A e B , e um ponto qualquer (chamado ponto de referência do item B), o *nofit polygon* entre eles, denotado por $NFP_{A,B}$, é o polígono traçado pelo ponto de referência do item B quando ele desliza encostado, sem se sobrepor, ao redor do item A (veja Figura 8). O item que desliza é chamado item orbital e o outro item é chamado item fixo.

Figura 8 – *Nofit polygons* dos itens A e B .

(a) $NFP_{A,B}$.

(b) $NFP_{B,A}$.

Fonte: Elaborada pelo autor.

Dessa forma, três situações podem ocorrer (Figura 9):

- se o ponto de referência do item B está no interior do $NFP_{A,B}$, então há sobreposição entre itens A e B ;
- se o ponto de referência do item B está na fronteira do $NFP_{A,B}$, então os itens A e B estão em contato, mas não se sobrepõem;

- se o ponto de referência do item B está no exterior do $NFP_{A,B}$, então os itens A e B não se sobrepõem e não estão em contato.

Figura 9 – Posições relativas entre os itens dado o NFP entre os itens.

Fonte: Elaborada pelo autor.

Um algoritmo simples para o cálculo de *nofit polygons* de polígonos convexos foi proposto por Cuninghame-green (1989). A Figura 10 ilustra o processo. Dados um item fixo A , com orientação anti-horária, um ponto arbitrário, chamado *origem* de A , um item orbital B , com orientação horária e seu ponto de referência (Figura 10a), o algoritmo translada todas as arestas dos itens para algum ponto arbitrário (Figura 10b) e então as arestas são concatenadas no sentido anti-horário (Figura 10c).

Figura 10 – Algoritmo para o cálculo do NFP .

Fonte: Elaborada pelo autor.

A fim de posicionar o $NFP_{A,B}$ corretamente (Figura 11), é necessário colocar o item B em contato com o item A no ponto de início da aresta a_1 (Figura 11a), localizar a aresta a_1 no $NFP_{A,B}$ (Figura 11b) e posicionar seu início no ponto de referência do item B (Figura 11c). Assim, o $NFP_{A,B}$ é posicionado e o ponto de referência do $NFP_{A,B}$ é dado pelo ponto de referência do item A .

O algoritmo apresentado anteriormente funciona exclusivamente para itens com formas convexas. Para utilizá-lo com formas não convexas, é necessário um tratamento prévio. Assim, os

Figura 11 – Posicionamento do NFP .

Fonte: Elaborada pelo autor.

itens não convexos são decompostos em formas convexas, isto é, cada item é representado por um conjunto de polígonos convexas. Dessa forma, pode-se usar o algoritmo de [Cunningham-green \(1989\)](#) para calcular os NFP parciais de cada par de polígonos convexas e depois combiná-los novamente para gerar o NFP dos itens originais ([Figura 12](#)). O trabalho de [Cherri et al. \(2016\)](#) explica o processo de forma detalhada.

Figura 12 – *Nofit polygon* entre um item convexo e um não convexo.

Fonte: Elaborada pelo autor.

Verificar a sobreposição utilizando a técnica do *nofit polygon* se resume a verificar se um ponto está no interior de um polígono. Essa verificação é computacionalmente menos custosa que verificar sobreposição através da trigonometria direta, porém, é custoso o cálculo dos *nofit polygons* de todos os pares de itens, especialmente quando se trabalha com itens não convexos. Assim, os *nofit polygons* são calculados apenas uma vez, em uma fase de pré-processamento, e então utilizados durante todo o processo de empacotamento.

De forma análoga, o *inner-fit rectangle (IFR)* define a região de contenção de um item em um retângulo. Assim, dado um item A e um recipiente retangular R , o $IFR_{R,A}$ é o retângulo formado quando o item A , no interior do retângulo R , desliza encostado por todos os lados do retângulo ([Figura 13](#)). Dessa forma, o item A está inteiramente contido no recipiente R , se o seu ponto de referência está no interior ou na fronteira do $IFR_{R,A}$.

Como, nesse caso, o recipiente é retangular, calcular as dimensões do IFR é muito simples. Assim a altura do $IFR_{R,A}$ é a altura do recipiente menos a altura do item e , analogamente,

o comprimento do *IFR* é o comprimento do recipiente menos o comprimento do item.

Figura 13 – *Inner-fit rectangle* entre um retângulo R e um item A .

Fonte: Elaborada pelo autor.

Caso o recipiente não seja retangular (Figura 14), isto é, irregular, a região de contenção do item no recipiente é chamada *inner-fit polygon* (*IFP*). Dado um item A e um recipiente irregular P , uma das formas de calcular o $IFP_{A,P}$ é aproximar o recipiente pelo seu retângulo envolvente e cobrir os pedaços faltantes do recipiente com polígonos. Uma vez feito isso, basta calcular o *IFR* do item A com o retângulo envolvente de P e os *NFP* do item A com os polígonos no recipiente. O *IFP* é a região do *IFR* menos as regiões dos *NFPs* calculados.

Figura 14 – *Inner-fit polygon* entre um recipiente irregular e um item.

Fonte: Elaborada pelo autor.

Representando os itens por malha, podemos utilizar o método de rasterização para verificar a sobreposição entre itens. 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.

A codificação da matriz expressa a geometria dos itens e na literatura existem várias codificações (SEGENREICH; BRAGA, 1986; BABU; BABU, 2001; OLIVEIRA; FERREIRA, 1993). O código mais simples é o código binário, proposto por Oliveira e Ferreira (1993), que atribui zero às células não ocupadas pelo item e atribui um para as ocupadas. Na Figura 15 temos o exemplo de um item representado pela rasterização 0-1.

Figura 15 – Item representado pela rasterização binária.

Fonte: Elaborada pelo autor.

Para representar recipientes irregulares utilizando malha, basta criar uma matriz com o tamanho aproximado do retângulo envolvente do recipiente e alocar 1 nas células onde há defeitos e 0 caso contrário. A Figura 16 ilustra um recipiente representado por malha. Note que caso as dimensões do retângulo envolvente dividido pela unidade da malha não dê um número exato, então a representação é arredondada para baixo.

Ao utilizar a representação por malha, itens e recipientes devem ser rasterizados. 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 do recipiente. Caso alguma célula resulte em um valor maior do que um, então há sobreposição e, portanto, o ponto de alocação escolhido é infactível. Caso contrário, a posição é viável, o item pode ser alocado e a matriz do recipiente atualizada. A Figura 17 ilustra a verificação de sobreposição utilizando a técnica de rasterização.

Ao utilizar malha para representar os itens, é necessário escolher a unidade da malha. Uma unidade maior resulta em uma matriz de dimensões menores e uma célula representa uma área maior do item. Assim, a escolha da unidade da malha requer atenção, já que a precisão da

Figura 16 – Recipiente irregular representado por malha.

0	1	1	0	0	1	1	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	1	1	0	0	1	1	0

Fonte: Elaborada pelo autor.

Figura 17 – Verificação de sobreposição pelo método de rasterização.

Fonte: Elaborada pelo autor.

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. A Figura 18 mostra um item representado por malhas com unidades diferentes. Nas figuras 18b e 18c, as células pintadas de cinza representam células um e as células brancas representam células zero.

Figura 18 – Item representado por malhas com unidades diferentes.

- (a) Forma original do item. (b) Malha com unidade igual a 1. (c) Malha com unidade igual a 0,5.

Fonte: Elaborada pelo autor.

Neste trabalho, os itens e recipientes são representados tanto por malha quanto por polígonos. A codificação de malha utilizada é a binária 0-1 de Oliveira e Ferreira (1993). Para a representação por polígonos, utilizou-se a técnica do *nofit polygon* para verificar a sobreposição (CUNINGHAME-GREEN, 1989).

Neste capítulo apresentamos o problema estudado e definimos as ferramentas utilizadas para verificar sobreposições entre itens. Nos próximos capítulos apresentamos os métodos e abordagens para resolução de problemas relevantes para o nosso estudo e propomos três métodos heurísticos bases com três variações cada para a resolução do nosso problema tanto para recipientes retangulares quanto irregulares.

REVISÃO BIBLIOGRÁFICA

Neste capítulo, apresentamos um levantamento da literatura sobre o problema de empacotamento de itens irregulares considerando balanceamento de carga. Assim, a revisão bibliográfica foi dividida da seguinte forma: ainda nesta seção, apresentamos o trabalho de [Egeblad \(2009\)](#), que é fortemente relacionado ao nosso problema e nas próximas seções apresentamos outros trabalhos relevantes para o nosso estudo. Na Seção [3.1](#) apresentamos trabalhos que resolvem o problema de corte irregular, também chamado de problema de *nesting*, e na Seção [3.2](#) abordamos os problemas de empacotamento considerando o balanceamento de carga.

O trabalho de [Egeblad \(2009\)](#) resolve o problema bidimensional e tridimensional de posicionamento de itens em um único recipiente considerando o balanceamento de carga e o momento de inércia. Assim, dado um recipiente e uma lista de itens (regulares e irregulares), busca-se um posicionamento que minimize o momento de inércia e o desbalanceamento de carga, de forma que todos os itens estejam contidos no recipiente, sem sobreposição. O autor apresenta uma formulação para o problema e prova que a versão de decisão do problema é NP-completo. Também, apresenta uma heurística baseada no trabalho de [Egeblad, Nielsen e Odgaard \(2007\)](#). O algoritmo começa priorizando o balanceamento e a minimização do momento de inércia e, ao longo do processo, o algoritmo prioriza a factibilidade da solução. A sobreposição dos itens é minimizada utilizando o algoritmo de [Egeblad, Nielsen e Odgaard \(2007\)](#). A heurística foi testada com instâncias com itens e recipientes regulares e irregulares, algumas delas foram propostas e outras foram adaptadas do problema de empacotamento em faixa, considerando o peso dos itens igual à área de cada um multiplicado por um número aleatório entre meio e dois. Os experimentos apresentaram bons resultados dentro de minutos, mesmo com instâncias com mais de 150 retângulos. Porém, o trabalho não foi comparado com outros métodos.

3.1 Corte de itens irregulares

Problemas de corte de itens irregulares são problemas de corte e empacotamento nos quais os itens possuem formas regulares e irregulares, considerando itens irregulares aqueles que possuem formas não circulares e não retangulares. Esses problemas são NP-Difíceis (NIELSEN; ODGAARD, 2003), logo a maior parte dos trabalhos relacionados utiliza heurísticas para resolver tais problemas. É claro que os problemas de corte irregular podem ser tridimensionais, mas há pouquíssimos trabalhos sobre o tema. Esses problemas geralmente são mais complicados de se resolver e alguns ainda acrescentam várias restrições relacionadas ao peso dos itens. Assim, esses trabalhos serão abordados junto com os problemas de empacotamento que consideram peso (Seção 3.2).

Dentre os problemas bidimensionais de corte irregular temos o problema de empacotamento em faixa e o problema de empacotamento em recipientes. No problema de empacotamento em faixa, também chamado de *strip packing problem*, o recipiente possui largura fixa e comprimento considerado infinito. Esse tipo de recipiente é comumente chamado de recipiente aberto ou recipiente com uma dimensão aberta (Figura 19a). No problema de empacotamento em recipientes, os recipientes possuem largura e comprimento fixos e são chamados recipientes fechados (Figura 19b). Os recipientes também podem ter formatos irregulares. Uma classificação dos problemas de corte e empacotamento é apresentada por Wäscher, Haußner e Schumann (2007).

Figura 19 – Tipos de recipiente.

Fonte: Elaborada pelo autor.

O problema que estudamos faz parte dos problemas de empacotamento em recipientes, mas a maioria dos trabalhos na literatura relacionados aos problemas de corte irregular resolvem o problema de empacotamento em faixa. Como as estratégias utilizadas para um problema são facilmente adaptáveis para o outro, nossa revisão bibliográfica contém trabalhos que resolvem ambos os problemas. Assim, apresentamos na subseção 3.1.1 os trabalhos com recipientes com uma dimensão aberta e, em seguida, na subseção 3.1.2 os trabalhos com recipientes fechados.

3.1.1 Problemas de empacotamento em faixa

Inspirados na indústria têxtil, onde os recipientes são rolos de tecidos que possuem comprimento que pode ser considerado infinito, os problemas de empacotamento em faixa procuram cortar um conjunto de itens dentro de um único recipiente, que possui largura fixa e comprimento infinito, de forma a minimizar o tamanho da faixa utilizada, respeitando as restrições de contenção dos itens no recipiente e não sobreposição dos itens. Uma revisão de abordagens heurísticas para esse problema é apresentada por [Bennell e Oliveira \(2009\)](#).

Podemos dividir os métodos heurísticos da literatura em dois grupos: as heurísticas construtivas e de melhoramento. Heurísticas construtivas trabalham com soluções parciais até chegar na solução final, ou seja, constroem a solução final decidindo item a item a posição que será ocupada no recipiente, e uma vez posicionado, o item não é movido novamente. Já as heurísticas de melhoria trabalham com soluções completas, movendo os itens dentro do recipiente ou trocando itens na sequência de posicionamento, com o objetivo de melhorar a solução inicial.

Uma heurística construtiva bastante intuitiva é a heurística *Bottom-Left* (BL) introduzida por [Baker, Coffman Jr e Rivest \(1980\)](#) que, como o próprio nome sugere, posiciona o item o mais à esquerda e mais abaixo possível no recipiente, respeitando as restrições de contenção e não sobreposição. A BL proposta por [Baker, Coffman Jr e Rivest \(1980\)](#) funciona da seguinte forma: cada item a ser alocado é inicialmente posicionado no canto superior direito do recipiente e deslocado para baixo até que encoste em outro item ou no recipiente. Então, o item é deslocado à esquerda até encostar em outro item ou no recipiente. Repete-se o processo até que não seja mais possível movimentar o item. Essa implementação da BL não é única na literatura, já que a BL pode variar de acordo com a ferramenta utilizada para representar itens, com a orientação do recipiente e também caso a BL tenha a capacidade de preencher buracos. Quando a *Bottom-Left* tem tal capacidade, ela é chamada *Bottom-Left-Fill* (BLF). Outras formas de implementar a BL/BLF podem ser vistas em [Segenreich e Braga \(1986\)](#), [Dowsland, Vaid e Dowsland \(2002\)](#), [Gomes e Oliveira \(2002\)](#), [Burke et al. \(2006\)](#) e [Mundim \(2015\)](#). Em seu trabalho, [Mundim \(2015\)](#) propõe uma implementação da BLF com a representação dos itens por malha. Assim, buscar uma posição factível seguindo a regra BL consiste em fazer uma varredura nas células da malha do recipiente em uma determinada ordem, partindo da célula mais à esquerda e mais abaixo. Além disso, também propõe variações para a regra de posicionamento, como alocar o item o mais à direita e mais acima possível. Por ser uma heurística bastante intuitiva e por gerar soluções de forma rápida, a BL/BLF é vastamente utilizada na literatura como regra de posicionamento, isto é, uma heurística rápida, que é utilizada para gerar uma solução a partir de uma determinada sequência de itens ou gerar uma solução inicial, que será melhorada depois por outra heurística. Neste trabalho utilizamos a BLF como regra de posicionamento para algumas das heurísticas propostas, baseada na BLF apresentada no trabalho de [Mundim \(2015\)](#).

Outra heurística construtiva é a TOPOS, proposta por [Oliveira, Gomes e Ferreira \(2000\)](#).

Essa heurística constrói a solução sem posições fixas para os itens. Assim, os itens são alocados a partir de um ponto flutuante e a cada item alocado, forma-se uma solução parcial que é, então, considerada como um único item, a fim de facilitar. Essas soluções parciais são restritas a uma largura fixa, que é a largura do recipiente. Os itens são alocados segundo os pontos factíveis dado pelo NFP e segundo uma regra de posicionamento definida previamente. A heurística foi testada com cinco instâncias, sendo quatro tomadas da literatura e uma proposta por eles, e encontrou melhores resultados para duas das quatro instâncias da literatura, até então.

Ainda nas heurísticas construtivas, [Bennell e Song \(2010\)](#) introduzem uma heurística que utiliza uma adaptação do algoritmo TOPOS para construir as soluções parciais, mas a ordem dos itens a serem alocados é feita dinamicamente de acordo com o desperdício gerado pela alocação dos itens e uma estimativa do desperdício futuro. Essa seleção dinâmica é feita através de uma árvore de busca, que é percorrida por um algoritmo de *beam search* que poda a árvore segundo avaliações local e global. O algoritmo foi testado com várias instâncias clássicas da literatura e apresentou resultados bastante competitivos.

Nas heurísticas de melhoramento, temos os algoritmos que buscam nas sequências dos itens e que buscam no próprio *layout*. Das heurísticas que buscam nas sequências citamos o algoritmo *Jostle* de [Dowsland, Dowsland e Bennell \(1998\)](#), os algoritmos de busca de [Gomes e Oliveira \(2002\)](#) e [Burke et al. \(2006\)](#) e o algoritmo genético híbrido de [Wong et al. \(2009\)](#).

[Dowsland, Dowsland e Bennell \(1998\)](#) propõem um algoritmo que nivela melhor a silhueta direita do plano de corte, ou seja, que busca um arranjo de itens que se aproxime a um retângulo. Para tanto, a heurística ora empacota os itens o mais à esquerda, ora o mais à direita do recipiente e a sequência utilizada para o empacotamento é determinada pela coordenada x de cada item na iteração anterior. O processo se repete até uma quantidade fixa de iterações. A regra de posicionamento utilizada é uma adaptação da BL, que prioriza a alocação nas bordas inferiores e superiores, revezando entre elas. Há duas versões do algoritmo, uma utiliza trigonometria direta e a outra *NFP*, calculados numa fase de pré-processamento. Os algoritmos foram testados com instâncias reais e artificiais. Cada instância foi testada com a regra de posicionamento com preenchimento de buraco e a mesma regra sem tal capacidade. As variações do algoritmo capazes de preencher buracos encontraram melhores resultados que as outras variações, mas elas não foram comparadas com outros algoritmos da literatura.

O algoritmo de [Gomes e Oliveira \(2002\)](#), chamado *2-exchange*, faz uma busca nas sequências de empacotamento dos itens e essas sequências são transformadas em *layouts* pela regra de posicionamento BLF. Utilizando o NFP e IFR para satisfazer as restrições de contenção e não sobreposição dos itens, o algoritmo, a partir de uma solução inicial, gera soluções vizinhas mudando a sequência de itens. Essas mudanças podem ser troca de lugar entre dois itens, inserção de um item em outra posição e a rotação de um item. O algoritmo possui estratégias para escolher a solução para a próxima iteração. A meta-heurística foi testada com cinco *benchmarks* e, para quatro das cinco instâncias testadas, obteve melhores resultados que os melhores da literatura até

então.

Outro algoritmo de busca é o proposto por [Burke et al. \(2006\)](#), que é um algoritmo de busca tabu junto com um *hill-climbing* para escolher as soluções. As mudanças nas sequências são feitas a partir de um critério escolhido aleatoriamente dentre os seguintes: inserção, troca de dois, três, quatro e n itens. Para transformação de sequência em *layout*, [Burke et al. \(2006\)](#) propõem uma implementação diferente de BLF. Das 26 instâncias testadas, o algoritmo conseguiu produzir melhores resultados para 25 delas.

O trabalho de [Wong et al. \(2009\)](#) utiliza o algoritmo genético para determinar as sequências de empacotamento. Esse algoritmo genético não utiliza operadores de mutação e cruzamento específicos, o que resulta em uma eficiência computacional melhor. As sequências são transformadas em *layouts* utilizando abordagem de dois estágios baseado em malha binária. Essa abordagem foi comparada com um algoritmo que utiliza busca aleatória e regra de posicionamento BL. A abordagem proposta obteve resultados melhores para todas as oito instâncias testadas.

Dentre as heurísticas de melhoramento que fazem busca dentro do *layout*, temos o algoritmo de [Błażewicz, Hawryluk e Walkowiak \(1993\)](#), que utiliza um algoritmo de busca tabu. Para obter resultados mais rápidos, elimina a possibilidade de movimentos complicados, permitindo apenas a mudança de um elemento por vez. Assim, o algoritmo ordena os itens em ordem decrescente de área e empacota os itens segundo a regra de posicionamento BLF. Então, movimentando uma peça para um buraco ou para o fim do *layout*, obtêm-se soluções vizinhas. Os resultados encontrados foram bastante competitivos com os resultados da literatura até então.

O algoritmo de [Gomes e Oliveira \(2006\)](#) consiste em um algoritmo híbrido de *simulated annealing* e programação linear. Utilizando NFP e IFR para satisfazer as restrições do problema, o algoritmo gera uma solução inicial a partir da BLF e então, gera soluções vizinhas trocando dois itens ou movendo apenas um item de lugar. Esses movimentos podem causar sobreposição, mas estas são eliminadas utilizando um modelo de separação. Depois, utiliza-se um modelo de compactação para tentar diminuir o comprimento do empacotamento. O *simulated annealing* guia a busca sobre o espaço de soluções. O algoritmo foi testado com quinze instâncias clássicas e encontrou resultados melhores que os da literatura para todas as quinze.

[Egeblad, Nielsen e Odgaard \(2007\)](#) propõem um algoritmo para problemas bidimensionais e tridimensionais que é baseado em busca local e que também utiliza busca local guiada para fugir de ótimos locais. O algoritmo funciona da seguinte forma: a partir de uma solução inicial, o algoritmo diminui o tamanho do recipiente dessa solução. Os itens que não estão mais inteiramente contidos no recipiente são transladados para dentro do recipiente. Se essa etapa causar sobreposição entre itens, então a busca local reduz a sobreposição iterativamente. Caso a busca local resulte em uma solução factível, então o tamanho da placa é diminuído novamente e o processo se repete. Trigonometria direta é utilizada para verificar sobreposição entre os itens.

O trabalho de [Imamichi, Yagiura e Nagamochi \(2009\)](#) consiste em um algoritmo iterativo de busca local que utiliza um algoritmo de separação baseado em programação não linear e um algoritmo que troca dois itens específicos, para minimizar a sobreposição entre itens. O algoritmo foi comparado com outros algoritmos e apresentou resultados competitivos.

Baseado no algoritmo de [Imamichi, Yagiura e Nagamochi \(2009\)](#), [Leung, Lin e Zhang \(2012\)](#) apresentam o algoritmo ELS (*extended local search*) que utiliza busca local para guiar a busca pelos *layouts* gerados e busca tabu para evitar ótimos locais. Os *layouts* são gerados movendo um item de lugar ou trocando dois itens de posição. Por fim, um algoritmo de compactação é utilizado para melhorar os resultados. O algoritmo ELS além de se mostrar bastante competitivo, melhorou vários resultados de instâncias clássicas.

O algoritmo híbrido proposto por [Elkeran \(2013\)](#) é o algoritmo que apresenta os melhores resultados até o presente momento. Utilizando NFP para evitar sobreposição e uma estratégia de agrupamento de pares de itens para aumentar a qualidade da solução, a meta-heurística híbrida de *cuckoo search* com busca local guiada é basicamente composta por duas fases. A primeira diminui o comprimento do *layout* atual e a segunda minimiza qualquer sobreposição que possa ter sido causada pela primeira fase. O algoritmo foi testado com 15 instâncias clássicas do problema e obteve os melhores resultados da literatura, até o momento.

Dos métodos exatos, o trabalho de [Ribeiro, Carravilla e Oliveira \(1999\)](#) utiliza programação por restrições. A estratégia de busca constrói os domínios iniciais de cada item utilizando o IFR dos itens. Para cada par de itens, uma restrição de não sobreposição é feita com o auxílio do NFP desses dois itens e essas restrições se tornam ativas a medida que os itens são alocados. [Carravilla, Ribeiro e Oliveira \(2003\)](#) estendem o trabalho para itens com formas convexas e não convexas. Também adicionaram restrições para se evitar posicionamentos simétricos. O modelo foi testado com três instâncias, encontrando a solução ótima para uma delas.

Já [Fischetti e Luzzi \(2009\)](#) propõem um modelo de programação linear inteira mista baseado no modelo de [Daniels, Milenkovic e Li \(1994\)](#). Para garantir a não sobreposição dos itens, eles utilizam o NFP dos itens. Eles conseguiram provar a otimalidade para duas das três instâncias testadas, sendo a maior instância com otimalidade provada composta por sete itens. Essa dificuldade em resolver instâncias reais motivou os autores a proporem um modelo para o problema de *multiple containment*, problema que surge quando alguns itens considerados grandes já foram posicionados no recipiente e deseja-se utilizar o espaço não utilizado, alocando o maior número de itens pequenos possível. Testes computacionais mostraram que o modelo pode ser resolvido em um tempo computacional razoável e ainda melhorar a qualidade dos resultados dados por métodos heurísticos.

No trabalho de [Alvarez-Valdes, Martinez e Tamarit \(2013\)](#), os autores apresentam um levantamento dos modelos lineares inteiros mistos da literatura e propõem um modelo, baseado no trabalho de [Fischetti e Luzzi \(2009\)](#), que utiliza *lifting* nos limitantes das variáveis contínuas e descreve a construção das partições do exterior dos NFPs. Eles também propõem um algoritmo

Branch & Bound. O algoritmo foi testado e provou a otimalidade para instâncias com até 16 peças.

O modelo linear inteiro misto proposto por Toledo *et al.* (2013) utiliza a discretização do recipiente para alocação dos itens. Variáveis binárias expressam a alocação dos itens nos pontos do recipiente. NFP e IFR são usados para garantir que os itens não se sobreponham e não sejam alocados fora do recipiente. O modelo foi resolvido utilizando o *solver* CPLEX12.3¹. Os resultados computacionais mostram que o modelo é capaz de provar a otimalidade para instâncias bem maiores do que os outros modelos, mas as soluções são dependentes da malha adotada.

Leão *et al.* (2016) propõem um modelo diferente dos modelos anteriores, por utilizar uma abordagem semicontínua para o recipiente. Assim, o recipiente, os NFPs e IFRs são definidos por conjuntos de linhas horizontais e o ponto de referência dos itens devem ser alocados nessas linhas. O modelo foi resolvido com CPLEX 12.5 com 38 instâncias o modelo se mostrou bastante competitivo quando comparado com modelos contínuos e discretos.

O trabalho de Cherri *et al.* (2016) apresenta duas formulações, sendo que uma delas utiliza trigonometria direta para verificar a sobreposição e a outra utiliza a decomposição dos itens em polígonos convexos e NFP. Os modelos foram comparados com o melhor modelo da literatura até o momento, atribuído a Alvarez-Valdes, Martinez e Tamarit (2013), utilizando o *solver* CPLEX 12.6. Os resultados mostram que o modelo que utiliza NFP é mais eficiente que o modelo que utiliza trigonometria direta e quando comparado com a formulação de Alvarez-Valdes, Martinez e Tamarit (2013), para instâncias de até 10 itens, os resultados são bastante similares, mas para instâncias maiores o modelo apresenta melhores resultados em menor tempo.

Rodrigues e Toledo (2016) apresentam um modelo de cobertura por cliques baseado no modelo linear inteiro misto de Toledo *et al.* (2013). O modelo proposto gera um menor número de restrições que o trabalho de Toledo *et al.* (2013). Eles também desenvolveram um método iterativo para melhorar o limitante inferior. O modelo obteve solução ótima para instâncias com 25 itens, sendo 22 deles distintos.

3.1.2 Problemas de empacotamento em recipientes fechados

Dentre os problemas de empacotamento em recipientes, temos o *bin packing* ou problema de empacotamento em recipientes fechados e o *knapsack problem* ou problema da mochila. No problema de empacotamento em recipientes fechados deseja-se empacotar uma lista de itens utilizando o menor número de recipientes possível de forma que os itens devem estar inteiramente contidos e eles não podem se sobrepor. Dos trabalhos que resolvem esse problema, temos Terashima-Marín *et al.* (2010) e López-Camacho *et al.* (2013).

¹ <https://www-01.ibm.com/software/commerce/optimization/cplex-optimizer> - Último acesso: 02/04/2016

O trabalho de [Terashima-Marín et al. \(2010\)](#) consiste em uma hiper-heurística proposta para resolver o problema de empacotamento irregular e regular. O algoritmo é baseado em um algoritmo genético que gerencia heurísticas de baixo nível. As heurísticas de baixo nível são usadas para seleção de itens e recipientes e para alocação dos itens. O algoritmo encontrou melhores resultados para um grande conjunto de instâncias.

Já o algoritmo de [López-Camacho et al. \(2013\)](#) é uma heurística construtiva que utiliza trigonometria direta para verificar sobreposição dos itens. A heurística primeiramente aloca os itens em ordem decrescente até que um terço da área do recipiente esteja ocupada. Depois, o algoritmo busca um, dois ou três itens que preencham completamente o restante do recipiente. Caso não encontre, ele faz a busca novamente com o objetivo de preencher o recipiente, deixando uma área de desperdício. Caso não encontre, a área de desperdício aumenta e assim por diante. Variações do algoritmo também foram implementadas com área inicial a ser ocupada de um quarto, meio e dois terços. A heurística e suas variações foram implementadas e testadas. As variações de um quarto e um terço de área ocupada inicialmente produziram os melhores resultados, no geral.

Já no problema da mochila, deseja-se maximizar a ocupação de um único recipiente e nem todos os itens precisam ser empacotados. Novamente, os itens não podem se sobrepor e devem estar inteiramente contidos no recipiente. Dos trabalhos que resolvem o problema, temos [Halavati et al. \(2005\)](#), [Halavati, Shouraki e Zadeh \(2008\)](#) e [Dalalah, Khrais e Bataineh \(2014\)](#).

O algoritmo proposto por [Halavati et al. \(2005\)](#) é uma heurística de melhoramento que é baseada em uma abordagem *novel population*, em que o *layout* é formado a partir de agrupamento de itens como relações de amizade. Esses agrupamentos são feitos unindo itens por seus vértices. O trabalho é melhorado em [Halavati, Shouraki e Zadeh \(2008\)](#), que propõe um algoritmo evolutivo em que os cromossomos representam soluções parciais formadas por itens conectados sem sobreposição. Três operadores são usados para a evolução das soluções parciais mais adequadas até que uma solução completa aceitável apareça. O algoritmo foi testado com instâncias da literatura e encontrou melhores resultados para algumas delas.

[Dalalah, Khrais e Bataineh \(2014\)](#) propõem um algoritmo construtivo bastante similar ao algoritmo TOPOS de [Oliveira, Gomes e Ferreira \(2000\)](#). Mas, diferentemente do TOPOS, o item a ser alocado e a sua posição são decididos pelo critério de aproveitamento da envoltória convexa, ao invés de ordenar a sequência de itens por algum critério predefinido. Os resultados mostraram que o algoritmo gera resultados competitivos com um tempo computacional aceitável.

Em relação a modelos e métodos exatos, citamos os trabalhos de [Scheithauer e Terno \(1993\)](#) e [Baldacci et al. \(2014\)](#). Apesar de ambos os trabalhos apresentarem modelos para o problema de empacotamento em recipientes fechados, nenhum deles propõe métodos exatos para sua resolução. O trabalho de [Scheithauer e Terno \(1993\)](#) apresenta modelos lineares inteiros mistos para o problema da mochila para o caso unidimensional e bidimensional e propõe uma generalização para o caso tridimensional. Já [Baldacci et al. \(2014\)](#) resolvem o problema de

empacotamento de itens irregulares com recipientes irregulares, que podem conter defeitos e zonas de qualidades. Para um único recipiente, eles propõem um modelo para o problema e uma relaxação lagrangiana. Também, adaptam o modelo e o algoritmo para múltiplos recipientes.

Abordagens para empacotamento em recipientes irregulares podem ser vistas em [Martins e Tsuzuki \(2010\)](#), [Valle et al. \(2012\)](#), [Alves et al. \(2012\)](#), [Dalalah, Khrais e Bataineh \(2014\)](#) e [Baldacci et al. \(2014\)](#).

3.2 Problemas de empacotamento considerando o peso dos itens

Nesta seção vamos abordar problemas de empacotamento em recipientes nos quais o peso influencia no arranjo dos itens. Nossa revisão consiste de trabalhos relevantes, com problemas bidimensionais e tridimensionais, que buscam o balanceamento de carga com itens regulares, empacotamento tridimensional de itens irregulares que não consideram a distribuição do peso e outros trabalhos.

Dos problemas de empacotamento bidimensional com itens regulares que busca o balanceamento de carga, o trabalho de [Kaluzny e Shaw \(2009\)](#) resolve o problema de carregamento de avião. A simplificação do tridimensional para o bidimensional é feita considerando apenas a largura e comprimento dos itens, admitindo que os itens tem a mesma altura e impedindo o empilhamento dos itens. Os autores propõem modelos lineares inteiro misto, um modelo para recipientes fechados e outro para o problema da mochila. Além disso, os autores propõem um conjunto de restrições para diminuir o espaço de busca. O modelo foi resolvido com 100 instâncias utilizando o *solver* CPLEX 11.0.1. Sem utilizar os cortes especializados, o modelo foi capaz de encontrar soluções factíveis para 75 das 100 instâncias testadas em menos de um minuto. Utilizando os cortes, o modelo encontrou soluções factíveis para todas as instâncias e provou a otimalidade para 87 delas.

Outras abordagens bidimensionais são [Liu et al. \(2008\)](#), [Fernández et al. \(2013\)](#) e [Queiroz e Miyazawa \(2013\)](#). Os três trabalhos fazem a simplificação do tridimensional para o bidimensional considerando a largura e altura dos itens, e assumindo o comprimento igual para todos os itens.

[Liu et al. \(2008\)](#) formulam dois modelos bidimensionais para o problema, um multiobjetivo (minimizar o número de recipientes utilizados e minimizar o desvio do centro de gravidade) e outro com apenas um objetivo (minimizar o número de recipientes), e propõem um algoritmo multiobjetivo de otimização por enxame de partículas evolutivo para resolver os modelos mono e multiobjetivo propostos. As partículas codificam as soluções e incluem o número de recipientes e a ordem em que os itens são empacotados. O método faz uso de BLF para empacotar a sequência de itens nos recipientes. 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. \(2013\)](#) propõem três implementações de algoritmo memético para resolver problemas de empacotamento. Um para o empacotamento de itens regulares mono-objetivo e dois para o problema multiobjetivo. Para o problema mono-objetivo, o algoritmo utiliza BLF para inicializar as soluções e para os multiobjetivos utilizam uma adaptação da BLF, denominada por eles como BLL (*Bottom-Left Level*), que usa o parâmetro *level* para estabelecer a altura máxima dos recipientes onde os itens podem ser empacotados. 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.

O trabalho de [Queiroz e Miyazawa \(2013\)](#) resolve o problema de empacotamento em faixa com restrições de ordem de descarregamento, fragilidade e balanceamento de carga. A restrição de balanceamento de carga exige que apenas a coordenada x do centro de gravidade esteja em uma região definida. Os autores propõem formulações matemáticas e duas heurísticas para resolução do problema, uma de empacotamento em nível e outra baseada em pontos de canto. As heurísticas são interessantes, já que resolvem problemas em um cenário mais realista que as outras até então, mas testes computacionais mostram que os algoritmos são limitados ao resolver certas instâncias.

Para problemas de empacotamento tridimensional com itens regulares considerando o balanceamento de carga temos o trabalho de [Liu et al. \(2011\)](#), que propõe uma abordagem híbrida de busca tabu com heurísticas de carregamento para o empacotamento de caixas em um único contêiner com o objetivo de maximizar a utilização do espaço. As restrições relacionadas ao peso são de orientação, empacotamento ortogonal, estabilidade, conectividade, limite e distribuição de peso. A distribuição de peso define que o centro de gravidade do contêiner fique perto do centro geométrico da base do contêiner. A partir de uma solução inicial, é efetuada busca tabu na sequência de caixas a serem empacotadas e a heurística de carregamento empacota os itens dentro do contêiner. A heurística foi testada com instâncias clássicas da literatura e encontrou as melhores taxas de ocupação, dadas algumas condições. Também, a heurística encontrou os melhores resultados para estabilidade da carga.

Com o objetivo de resolver o problema de empacotamento regular em contêineres em duas fases, [Trivella e Pisinger \(2016\)](#) propõem um modelo para o problema de balanceamento de carga para um único contêiner multidimensional. Tal forma de resolução nem sempre garante a solução ótima para o problema geral. Logo, os autores também propõem um modelo para o problema de empacotamento regular multidimensional com balanceamento de carga. Além disso, é apresentada uma heurística de busca local em vários níveis, que representa os empacotamentos por grafos. Os experimentos computacionais mostram que um empacotamento com carga balanceada eficaz pode ser obtido.

Paquay, Schyns e Limbourg (2016) propõem uma formulação para o problema de empacotamento regular em contêineres aplicado ao carregamento de avião. Além da restrição de distribuição de peso, eles incluem restrições de orientação, posicionamento ortogonal, estabilidade vertical e fragilidade. A restrição de distribuição de peso define que o centro de gravidade do contêiner deve estar horizontalmente no centro geométrico da base e, verticalmente, abaixo de um dado nível. Além disso, os contêineres podem ser retangulares ou ter formatos especiais (paralelepípedos truncados). O modelo foi testado em um conjunto de instâncias proposto por eles com o *solver* CPLEX 12.5, provando a otimalidade para instâncias com até 18 caixas

Na literatura, encontramos várias abordagens unidimensionais para o problema de empacotamento com balanceamento de carga. A justificativa para essa abordagem reside no fato de que, no carregamento de aviões, a principal preocupação é o balanceamento longitudinal, ou seja, o balanceamento ao longo do comprimento da aeronave. Utilizando essa abordagem, temos os trabalhos de Amiouny *et al.* (1992), Mongeau e Bes (2003) e Lara e Labrador (2010). Amiouny *et al.* (1992) provam que o problema de balanceamento de carga unidimensional é NP-Difícil e propõem uma heurística que aloca os itens do fim ao meio do recipiente. Utiliza-se uma estratégia de ordenar os itens em ordem crescente em relação à densidade do item e calcular, a cada item alocado, onde alocar o próximo item de forma que o centro de gravidade do recipiente fique o mais próximo do centro desejado. Mongeau e Bes (2003) propõem um modelo multiobjetivo para o problema da mochila com balanceamento de carga. Lara e Labrador (2010) propõem um algoritmo baseado em colônia de formigas para o problema multiobjetivo de empacotamento com balanceamento de carga.

Temos também trabalhos que resolvem o problema de carregamento de navios, mais conhecido como *master bay problem*, como os trabalhos de Ambrosino, Sciomachen e Tanfani (2004) e Ambrosino, Sciomachen e Tanfani (2006). Esses trabalhos lidam com o balanceamento de carga de uma forma mais simplificada, de forma que existem restrições que impõem que o peso do lado esquerdo do navio deve ser igual ao peso do lado direito, dado um valor de tolerância. Da mesma forma, o peso da popa deve ser igual ao peso da proa. Também, contêineres mais pesados são alocados abaixo de contêineres mais leves.

Existem alguns trabalhos na literatura que abordam o problema de empacotamento irregular tridimensional, como os trabalhos de Ikonen *et al.* (1997) e Egeblad *et al.* (2010). O trabalho de Ikonen *et al.* (1997) deseja encontrar uma disposição de objetos não convexos em cilindros, mas nesse problema o peso dos objetos não é considerado. O trabalho de Egeblad *et al.* (2010) resolve o problema de empacotamento de objetos irregulares em contêineres proveniente da indústria de móveis. Os autores consideram o peso dos itens, mas não há restrições relacionadas à distribuição do peso no recipiente, apenas a estabilidade da carga.

3.3 Considerações

Nesta seção apresentamos um levantamento da literatura sobre o problema de empacotamento de itens irregulares considerando o balanceamento de carga. Até onde vai nosso conhecimento, [Egeblad \(2009\)](#) é o único trabalho que resolve tal problema. Assim, procuramos estudar o problema de forma separada, estudando as ferramentas e estratégias utilizadas para empacotar itens irregulares e as abordagens propostas na literatura para resolver o balanceamento de itens regulares. Logo, este trabalho foi motivado pela escassez da literatura em relação ao problema que tem grande importância prática. Propomos neste trabalho métodos heurísticos que utilizam de ferramentas diferentes do trabalho de [Egeblad \(2009\)](#), tanto para o empacotamento, quanto para o balanceamento.

HEURÍSTICAS PROPOSTAS

Neste capítulo, apresentamos as heurísticas desenvolvidas para a resolução do problema bidimensional de empacotamento de itens irregulares considerando o balanceamento de carga. Na Seção 4.1 apresentamos a heurística QUAD-INSIDE e suas variações. Estas heurísticas procuram balancear o recipiente de forma indireta, espalhando o peso dos itens pelo recipiente, a fim de obter um centro de gravidade mais próximo do centro desejado. Na Seção 4.2 descrevemos a heurística DESLOC que aborda o balanceamento de forma direta. Assim, o centro de gravidade do recipiente e dos itens influenciam diretamente na posição escolhida para cada item.

Neste trabalho, assumimos que o centro de gravidade de um item coincide com seu centro geométrico e que cada item cabe dentro do recipiente.

4.1 Heurística QuadInside e suas variações

A heurística QUADINSIDE representa os itens e recipientes por meio de malhas e utiliza a estratégia de espalhar o peso o melhor possível pelo recipiente, para se obter um centro de gravidade mais próximo do centro geométrico do chão do recipiente. Para tanto, o algoritmo divide cada 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. A divisão do recipiente em quadrantes depende do centro de gravidade desejado (CGd) do recipiente. A Figura 20 ilustra a divisão em quadrantes para dois recipientes com CGds diferentes.

Com o recipiente dividido, a heurística aloca os itens ora em um quadrante, ora em outro, partindo do CGd para fora, imitando uma espiral. O posicionamento do item é feito utilizando variações da *Bottom-Left* (BL) e, para cada quadrante, há uma regra de alocação diferente. Por exemplo, a regra de alocação do primeiro quadrante foi implementada da seguinte forma: dado

Figura 20 – Recipientes divididos em quadrantes.

Fonte: Elaborada pelo autor.

o item a ser alocado, ele é movimentado de baixo para cima e da direita para esquerda na matriz recipiente, seguindo a ordem mostrada na Figura 21. 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. Para os outros quadrantes a regra varia apenas na ordem em que as células da matriz são verificadas. A Figura 22 ilustra as variações para cada quadrante.

Figura 21 – Ordem de busca no primeiro quadrante na heurística QUADINSIDE.

12	9	6	3	
11	8	5	2	
10	7	4	1	

Fonte: Elaborada pelo autor.

Figura 22 – Regras de posicionamento em cada quadrante na heurística QUADINSIDE.

(a) 1º quadrante.

12	9	6	3
11	8	5	2
10	7	4	1

(b) 4º quadrante.

9	10	11	12
5	6	7	8
1	2	3	4

(c) 2º quadrante.

4	3	2	1
8	7	6	5
12	11	10	9

(d) 3º quadrante.

1	4	7	10
2	5	8	11
3	6	9	12

Fonte: Elaborada pelo autor.

A entrada da heurística consiste em uma lista (*listItens*) de n itens irregulares a serem empacotados com suas características, as informações do recipiente, isto é, o número de linhas e colunas da matriz, a matriz em si e a posição do centro de gravidade desejado (CGd). As características dos itens são: a matriz que representa o item, dimensões da matriz, centro de gravidade do item e peso. Caso haja itens iguais, não é necessário guardar informações duplicadas, podendo tratar os itens como um único item e a quantidade de cópias de itens é chamada demanda do item. Para facilitar a explicação todos os itens são tratados como diferentes. Como saída tem-se S , que é a solução obtida, formada pela posição de cada item no recipiente, o recipiente em que cada item foi alocado, a taxa de ocupação de cada recipiente e o centro de gravidade final de cada recipiente.

Primeiramente, o algoritmo ordena a lista *listItens* em ordem decrescente em relação ao peso dos itens, para então, começar a alocar os itens no recipiente. Essa alocação é feita seguindo as regras de posicionamento explicadas anteriormente, mas revezando entre os quadrantes. A escolha do quadrante depende do centro de gravidade desejado (CGd). Caso o CGd seja o centro geométrico do recipiente, então o algoritmo escolhe o quadrante com menor peso no momento. Caso contrário, o algoritmo prioriza a alocação nos quadrantes de menor área. Caso não haja qualquer posição disponível para o item no quadrante escolhido, tenta-se alocá-lo no próximo quadrante de menor peso ou menor área. Caso ele não caiba em nenhum quadrante do recipiente atual, 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, somando ao peso anterior o número de células ocupadas pelo item recém-alocado vezes sua densidade. A densidade de um item é seu peso dividido por sua área. Um item pode ser alocado em dois ou mais quadrantes, caso ele não caiba inteiramente em apenas um. Quando isso acontece, seu peso é distribuído entre os quadrantes conforme a área ocupada em cada um. Note que o peso dos quadrantes é dado de forma aproximada, já que seu cálculo depende do número de células ocupadas pelo item e não pela sua área real. Após todos itens serem alocados, o algoritmo modifica o posicionamento dos itens do último recipiente utilizado, movendo o bloco de itens em direção ao CGd, respeitando a restrição de contenção dos itens.

O seguinte exemplo ilustra o funcionamento da heurística QUADINSIDE. Deseja-se empacotar 7 itens em recipientes retangulares com comprimento igual a 8, altura igual a 8 e unidade da malha igual a 1. As malhas dos itens são apresentadas na [Figura 23](#). As outras características dos itens são: o item 1 possui peso $\gamma_1 = 5$, centro de gravidade $(\lambda_{x,1}, \lambda_{y,1}) = (1,67; 1,56)$; o item 2, $\gamma_2 = 4$ e $(\lambda_{x,2}, \lambda_{y,2}) = (1,67; 1,56)$; o item 3, $\gamma_3 = 3$ e $(\lambda_{x,3}, \lambda_{y,3}) = (1,56; 0,89)$; o item 4, $\gamma_4 = 2,5$ e $(\lambda_{x,4}, \lambda_{y,4}) = (1,5; 1)$; o item 5, $\gamma_5 = 2$ e $(\lambda_{x,5}, \lambda_{y,5}) = (1,5; 1)$; o item 6, $\gamma_6 = 2$ e $(\lambda_{x,6}, \lambda_{y,6}) = (2; 1,5)$; por fim, o item 7, $\gamma_7 = 2$ e $(\lambda_{x,7}, \lambda_{y,7}) = (2; 1,5)$.

A heurística QUADINSIDE começa ordenando a lista de itens em ordem decrescente em relação ao peso dos itens (neste caso, mantém-se a ordem apresentada na [Figura 23](#)) e cria uma lista (*mRecip*) com uma matriz de dimensão 8×8 . Então, o algoritmo irá alocar o primeiro

Figura 23 – Itens a serem empacotados pelo algoritmo QUADINSIDE no exemplo.

Fonte: Elaborada pelo autor.

item da lista. Como os quadrantes são idênticos, o item é alocado no quadrante de menor peso e, como não há nenhum item alocado ainda, o item é alocado na primeira posição do primeiro quadrante. O peso do primeiro quadrante é atualizado para 5. A [Figura 24a](#) ilustra a primeira etapa do empacotamento com os pesos correspondentes de cada quadrante. O algoritmo passa para o segundo item da lista. O quadrante de menor peso e menor índice é o segundo quadrante. Como não há nenhum item alocado nele, o item é alocado na primeira posição e o peso do segundo quadrante é atualizado para 4 ([Figura 24b](#)). Da mesma forma, o item 3 é alocado na primeira posição do terceiro quadrante, e o peso do quadrante é atualizado para 3 ([Figura 24c](#)). O item 4 deve ser alocado no quadrante de menor peso, que é o quarto quadrante (peso 0). Assim, o quarto item da lista é alocado na primeira posição do quadrante e seu peso é atualizado para 2,5 ([Figura 24d](#)). Para o quinto item da lista, o algoritmo primeiro testa a primeira posição do quadrante de menor peso, o quarto quadrante (peso 2,5). Como a primeira posição está ocupada por outro item, o algoritmo verifica outras posições, seguindo a regra de posicionamento, até encontrar uma factível. Assim, o algoritmo aloca o item logo acima do item anteriormente posicionado e o peso do quadrante é atualizado para 4,5 ([Figura 24e](#)).

Figura 24 – Etapas do empacotamento do exemplo.

Fonte: Elaborada pelo autor.

Já o sexto item da lista, deve ser alocado no terceiro quadrante (peso 3), mas como não há nenhuma posição factível para ele ser alocado, então o algoritmo procura uma posição no segundo quadrante de menor peso, quadrante 2, e também não encontra posição factível. O algoritmo repete esse processo, procurando uma posição nos quadrantes 4 e 1. Uma vez que não há posição factível para o item em nenhum quadrante do recipiente, o algoritmo cria outra matriz para o segundo recipiente e aloca o item no quadrante de menor peso e menor índice, isto é, o

primeiro quadrante (Figura 25a). O peso do quadrante 1 do recipiente 2 é atualizado para 2. Para o sétimo item da lista, o algoritmo procura por todos os quadrantes do primeiro recipiente e, não achando posição factível, aloca o item no quadrante de menor peso e menor índice do segundo recipiente, o quadrante 2. O peso do segundo quadrante do segundo recipiente é atualizado para 2 (Figura 25b).

Figura 25 – Etapas do empacotamento do exemplo.

Fonte: Elaborada pelo autor.

Após todos os itens serem alocados, o algoritmo calcula o centro de gravidade final do último recipiente e desloca o bloco de itens alocados no último recipiente, recipiente de número dois, em direção ao centro de gravidade desejado. Dessa forma, o empacotamento no último recipiente, antes do deslocamento do bloco de itens, obteve um centro de gravidade igual a (2; 4) e após o deslocamento, o centro de gravidade igual a (4; 4), coincidindo com o centro de gravidade desejado. A Figura 26 ilustra o segundo recipiente antes e depois do deslocamento do bloco de itens alocados nele. O centro de gravidade do recipiente é representado pelo círculo branco.

Figura 26 – Segundo recipiente antes e depois do pós processamento.

Fonte: Elaborada pelo autor.

Para o primeiro recipiente, o algoritmo obteve centro de gravidade igual a (3,96; 4,19). A Figura 27 ilustra a solução obtida pelo algoritmo.

Note que a heurística funciona tanto para recipiente regulares, quanto irregulares. Para recipientes regulares, a matriz do recipiente é uma matriz de zeros e para recipientes irregulares, a matriz é do tipo 0-1, onde as células receberão 0 para as regiões viáveis do recipiente e 1, para as regiões fora do recipiente. Além disso, quando o bloco de itens é deslocado, é trivial

Figura 27 – Solução encontrada pela heurística QUADINSIDE para o exemplo.

Fonte: Elaborada pelo autor.

movê-lo de forma que ele continue dentro do recipiente. O Algoritmo 1 formaliza a heurística QUADINSIDE.

Algoritmo 1: QUADINSIDE.

Entrada: n , $listItems$, $recip$, CGd ;
 n : quantidade de itens;
 $listItems$: lista dos itens contendo para cada item: dimensões da matriz, matriz e peso do item;
 $recip$: matriz do recipiente;
 CGd : Centro de gravidade desejado;
Saída: S .
 S : Solução contendo as matrizes do recipientes utilizados e a lista de itens alocados em cada um.

```

1   $S \leftarrow \emptyset$ ;
2   $p \leftarrow 1$ ; //número de recipientes utilizados
3  Ordene a  $listItems$  em ordem decrescente em relação ao peso dos itens;
4  Crie uma lista ( $mRecip$ ) com uma matriz recipiente  $recip$ ;
5   $i \leftarrow 1$ ; //índice do item corrente
6  Enquanto  $i \leq n$  faça
7       $j \leftarrow 1$ ; //índice do recipiente corrente
8      Enquanto  $j \leq p$  faça
9          Se  $CGd$  é o centro geométrico então
10             Ordene a lista de quadrantes do recipiente  $j$  em ordem crescente em relação aos pesos;
11         Fim
12         Senão
13             Ordene a lista de quadrantes do recipiente  $j$  em ordem crescente em relação às áreas;
14         Fim
15          $k \leftarrow 1$ ; //índice corrente da lista de quadrantes
16         Enquanto  $k \leq 4$  faça
17             Busque um ponto factível  $(x, y)$  no  $k$ -ésimo quadrante da lista para o item  $i$  pela regra de
                posicionamento do quadrante;
18             Se Existe uma posição factível para o item  $i$  então
19                 A solução  $S$  recebe o item  $i$  alocado no recipiente  $j$  no quadrante  $k$  da lista na posição  $(x, y)$ ;
20                 Atualize a malha do recipiente  $j$  e o peso de cada quadrante;
21                  $i \leftarrow i + 1$ ;
22                 Vá para a linha 6;
23             Fim
24              $k \leftarrow k + 1$ ;
25         Fim
26          $j \leftarrow j + 1$ ;
27         Se  $j > p$  então
28             Adicione uma nova matriz recipiente à lista  $mRecip$ ;
29              $p \leftarrow p + 1$ ;
30         Fim
31     Fim
32 Fim
33 Desloque enquanto possível o bloco de itens alocados no recipiente  $p$  em direção ao  $CGd$  do recipiente, de forma
    que o bloco continue inteiramente dentro do recipiente;
34 Devolva  $S$ 

```

4.1.1 Heurística Quad-BottomLeft

A heurística QUAD-BOTTOMLEFT assemelha-se muito à heurística QUADINSIDE, diferindo na regra de posicionamento utilizada nos recipientes (linha 17 do Algoritmo 1). Nesta heurística, utiliza-se a *Bottom-Left* como regra de posicionamento para todos os quadrantes. A Figura 28 ilustra a regra de posicionamento para os quadrantes.

Outra diferença nas heurísticas é que os itens posicionados no último recipiente utilizado são desalocados e alocados novamente utilizando a heurística QUADINSIDE. Uma vez que todos os itens do último recipiente foram realocados, é feito o deslocamento do bloco de itens em direção ao CGd.

Figura 28 – Ordem em que os itens são movimentados na regra *Bottom-Left*.

3	6	9	12
2	5	8	11
1	4	7	10

Fonte: Elaborada pelo autor.

4.1.2 Heurística QuadSwirl

Assim como na heurística QUADINSIDE, a heurística QUADSWIRL também simula uma espiral para alocar os itens, mas de fora para dentro. Assim, a regra de posicionamento para cada quadrante é diferente. A Figura 29 ilustra a regra de posicionamento para os quadrantes. Nesta heurística, análoga à heurística anterior, também é feita a realocação dos itens do último recipiente utilizando a heurística QUADINSIDE, para então fazer o deslocamento do bloco de itens em direção ao CGd.

Figura 29 – Regras de posicionamento em cada quadrante na heurística QuadSwirl.

(a) 1º quadrante.

1	4	7	10
2	5	8	11
3	6	9	12

(b) 4º quadrante.

4	3	2	1
8	7	6	5
12	11	10	9

(c) 2º quadrante.

9	10	11	12
5	6	7	8
1	2	3	4

(d) 3º quadrante.

12	9	6	3
11	8	5	2
10	7	4	1

Fonte: Elaborada pelo autor.

4.2 Heurística Desloc

A heurística DESLOC é inspirada na heurística TOPOS proposta por (OLIVEIRA; GOMES; FERREIRA, 2000), que constrói a solução item por item, sem posições fixas. Essa abordagem de empacotamento permite que se desloque o bloco de itens, a cada iteração, para o mais perto do CGd possível. Dessa forma, a heurística DESLOC pode ser resumida da seguinte forma: o primeiro item é alocado num ponto arbitrário, os demais itens são alocados encostados em itens previamente alocados, segundo um critério definido anteriormente. Calcula-se o centro de gravidade do bloco de itens atual e move-se o bloco de itens em direção ao CGd do recipiente, de forma que os centros de gravidades fiquem o mais próximo possível. Note que mover o bloco de itens em direção ao CGd ou mover o recipiente em direção ao centro de gravidade do bloco de itens é equivalente. Abaixo, descrevemos com maior detalhe a heurística.

A representação dos itens e recipientes é feita utilizando polígonos e a técnica para verificar sobreposição é o *NFP*. A entrada da heurística consiste em uma lista (*listItens*) de *n* itens irregulares a serem empacotados com suas características, os *NFPs* de cada par de itens, o comprimento (*C*), altura (*A*) e o centro de gravidade desejado (CGd) dos recipientes. As características dos itens são: centro de gravidade do item, área, peso e o conjunto de polígonos convexos que compõe o item. Como saída tem-se *S*, que é a solução obtida, formada pelas posições dos itens nos recipientes e o recipiente em que cada item foi alocado.

A heurística começa ordenando a lista de itens (*listItens*) em ordem decrescente em relação ao peso. Para cada item, a heurística mantém uma lista de pontos factíveis que é atualizada cada vez que um item é posicionado. Essa atualização é feita da seguinte forma: para cada item, a heurística acrescenta à lista de pontos factíveis do item os vértices dos *NFPs* do item com os itens anteriormente alocados, considerando os *NFPs* transladados para sua posição correta em relação ao item. Além dos vértices dos *NFPs*, a heurística acrescenta à lista os pontos de intersecção entre os *NFPs* do item com os itens alocados anteriormente. A Figura 30 ilustra o processo de obtenção dos pontos factíveis para um item. Na Figura 30c, os pontos em branco são pontos obtidos dos vértices dos *NFPs* dos itens já alocados e o item a ser alocado. O ponto em preto é o ponto obtido a partir da intersecção dos *NFPs*. O ponto em cinza é obtido tanto do vértice dos *NFPs*, quanto da intersecção deles (JORGE *et al.*, 2015; AURELIANO; JORGE; ANDRETTA, 2016).

O primeiro item é sempre alocado em um ponto arbitrário, no caso o ponto (0,0). Alocado o primeiro item, a heurística atualiza a lista de pontos factíveis de cada item. Para alocar o segundo item, a heurística faz uma limpeza dos pontos factíveis do item corrente, excluindo pontos duplicados, pontos que estão dentro de *NFPs* do item corrente com itens já alocados (Figura 31a) e pontos que não respeitam os limites do recipiente, isto é, pontos em que a alocação do item gera um bloco de itens com altura ou comprimento maior que o recipiente (Figura 31b).

Uma vez que a lista de pontos para o segundo item já foi limpa, o algoritmo simula, para

Figura 30 – Processo para obtenção da lista de pontos factíveis.

Figura 31 – Processo de limpeza lista de pontos factíveis.

cada ponto, a alocação do item neste ponto, calcula o centro de gravidade do bloco de itens incluindo o item corrente, calcula o deslocamento do bloco de itens em direção ao CGd, simula o deslocamento do bloco de itens e calcula o novo centro de gravidade do bloco. Logo, o algoritmo tem, para cada ponto da lista, um centro de gravidade previsto caso o item fosse alocado neste ponto. O algoritmo escolhe o ponto que gera o centro de gravidade mais próximo do desejado. Caso haja empate, o algoritmo utiliza subcritérios de desempate.

Os subcritérios utilizados são:

- maximizar a sobreposição entre o retângulo envolvente do item a ser alocado e o retângulo envolvente dos bloco de itens alocados anteriormente, sem que haja sobreposição entre os itens (Figura 32a);
- minimizar a área do retângulo envolvente do bloco de itens com o item atual alocado (Figura 32b);
- minimizar a diferença entre a razão das dimensões do recipiente e a razão das dimensões do retângulo envolvente (Figura 32c).

Os subcritérios foram adaptados do algoritmo TOPOS de Oliveira, Gomes e Ferreira (2000). O primeiro subcritério busca um encaixe melhor de itens que são bastante irregulares, sem que os itens se sobreponham de fato. O segundo subcritério busca o ponto que gere a menor

Figura 32 – Processo de limpeza lista de pontos factíveis.

(a) Subcritério da sobreposição.

(b) Subcritério da área.

(c) Subcritério da razão.

Fonte: Elaborada pelo autor.

área depois que o item for alocado. E o terceiro subcritério favorece pontos que farão o bloco de itens mais proporcional as dimensões do recipiente.

Uma vez que o ponto para alocar o item é escolhido, seguindo os critérios descritos, o algoritmo aloca o item, atualiza os pontos factíveis para cada um dos itens restantes e repete o processo até que todos os itens sejam alocados. Caso não haja qualquer ponto factível para alocar um determinado item dentro de um recipiente, o algoritmo aloca o item no próximo recipiente. Caso esse item seja o primeiro a ser alocado no recipiente, o item é alocado no ponto (0,0).

Uma vez que todos os itens foram alocados, o algoritmo calcula o centro de gravidade do bloco dos itens e move o recipiente em direção a esse centro, de forma que o centro de gravidade do bloco fique o mais próximo possível do CGd do recipiente. A heurística é formalizada nos algoritmos 2 e 3.

A heurística DESLOC foi implementada utilizando os três subcritérios explicados acima, gerando três variações chamadas de SDESLOC, para a variação que utiliza o critério de sobreposição entre os retângulos envolventes do item e do bloco de itens do recipiente, ADESLOC, para a variação que utiliza o critério de menor área e RDESLOC, para a variação que utiliza o critério da razão das dimensões do recipiente.

4.2.1 Heurística Desloc-Irregular

Note que ao contrário das heurísticas QUAD, a heurística DESLOC resolve apenas problemas com recipientes retangulares, já que a verificação da contenção dos itens nesta heurística é feita limitando as dimensões do bloco às dimensões do recipiente. Para recipientes irregulares, verificar a contenção dos itens a cada iteração é computacionalmente custoso. Assim, propomos uma adaptação da heurística DESLOC para recipientes irregulares de forma que o recipiente é resolvido em duas fases. Na primeira fase, considera-se apenas o retângulo inscrito do recipiente e empacota-se o maior número de itens, considerando o balanceamento de forma direta. Caso ainda haja itens para empacotar, o algoritmo chama a segunda fase que considera o recipiente inteiro, empacota o maior número possível de itens restantes e o balanceamento é feito de forma indireta. Caso ainda fique itens sem empacotar, aplica-se a primeira e a segunda fase sucessivamente, até que todos os itens tenham sido empacotados.

Algoritmo 2: DESLOC.

Entrada: n , $listItens$, C , L e CGd ;
 n : quantidade de itens;
 $listItens$: lista dos itens contendo para cada item: centro de gravidade do item, área, peso e conjunto de polígonos que compõem o item;
 C : comprimento do recipiente;
 A : altura do recipiente;
 CGd : centro de gravidade desejado;
Saída: S .
 S : Solução contendo a lista de recipientes usados e, para cada um deles, a lista de itens nele alocados e a posição de cada item no recipiente.

```

1  $S \leftarrow \emptyset$ ;
2  $p \leftarrow 1$ ; //número de recipientes utilizados
3 Ordene a  $listItens$  em ordem decrescente em relação ao peso dos itens;
4 Crie uma lista de pontos factíveis ( $feas\_dots$ ) para cada item para cada recipiente;
5  $i \leftarrow 1$ ; //índice do item corrente
6 Enquanto  $i \leq n$  Faça
7    $j \leftarrow 1$ ; //índice do recipiente corrente
8   Enquanto  $j \leq p$  Faça
9     Se recipiente  $j$  está vazio então
10      Aloque o item  $i$  no ponto (0,0) do recipiente  $j$ ;
11       $S$  recebe o item  $i$  alocado no recipiente  $j$  no ponto (0,0);
12      Atualize a  $feas\_dots$  de todos os itens restantes para o recipiente  $j$ ;
13       $i \leftarrow i + 1$ ;
14      Vá para linha 6;
15     Fim
16     Senão
17       Exclua os pontos inviáveis da lista  $feas\_dots$  do item  $i$  para o recipiente  $j$ ;
18       Se  $feas\_dots \neq \emptyset$  então
19          $insertion\_point = choose\_point(S, item[i], j, feas\_dots[i][j], CGd)$ ;
20         Aloque o item  $i$  no ponto  $insertion\_point$ ;
21          $S$  recebe o item  $i$  alocado no recipiente  $j$  no ponto  $insertion\_point$ ;
22          $i \leftarrow i + 1$ ;
23         Vá para linha 6;
24       Fim
25       Senão
26          $j \leftarrow j + 1$ ;
27       Fim
28     Fim
29     Se  $j > p$  então
30        $p \leftarrow p + 1$ ;
31       Aloque o item  $i$  no recipiente  $p$  na posição (0,0);
32        $S$  recebe o item  $i$  alocado no recipiente  $p$  no ponto (0,0);
33        $i \leftarrow i + 1$ ;
34       Vá para linha 6;
35     Fim
36   Fim
37 Fim
38 Para  $j$  de 1 a  $p$  faça
39   Em  $S$ , desloque enquanto possível o bloco de itens alocados no recipiente  $j$  em direção ao  $CGd$  do recipiente;
40 Fim
41 Devolva  $S$ 

```

A heurística DESLOC-IRREGULAR representa os itens e recipientes por polígonos e utiliza-se da técnica de *NFP* e *IFP* para verificar a sobreposição e a contenção dos itens. O recipiente é representado por seu retângulo envolvente e por polígonos que sobrepõem as regiões inviáveis, de forma que o recipiente passa a ser retangular com polígonos representando os buracos do recipiente. Assim, a entrada da heurística DESLOC-IRREGULAR consiste no comprimento (C) e altura (A) do retângulo envolvente do recipiente, os polígonos que representam os defeitos no retângulo e suas respectivas posições, o centro de gravidade desejado (CGd) do recipiente, a lista ($listItens$) de n itens irregulares a serem empacotados com suas características, os *NFPs* de cada par de itens e os *IFPs* dos itens com o recipiente. As características dos itens são: centro de

Algoritmo 3: Função *choose_point*.

Entrada: S , $item$, k , $feas_dots$, CGd ;
 S : solução corrente;
 k : índice do recipiente;
 $item$: item corrente;
 $feas_dots$: lista dos pontos factíveis para alocar o $item$ no recipiente k ;
Saída: *insertion_point*.
insertion_point: ponto escolhido dentre a lista $feas_dots$ para alocar o $item$;

- 1 Crie uma lista de pontos ($list_cg$) do tamanho da lista $feas_dots$;
- 2 $valor \leftarrow 0$;
- 3 $indice \leftarrow 0$;
- 4 $desloc \leftarrow 0$;
- 5 $cg \leftarrow 0$;
- 6 **Para** j de 1 ao tamanho de $list_cg$ **faça**
- 7 $cg \leftarrow$ centro de gravidade do bloco de itens, incluindo o $item$ alocado no ponto $feas_dots[j]$;
- 8 $desloc \leftarrow$ o deslocamento possível dos itens alocados, incluindo o $item$, em direção ao CGd ;
- 9 $list_cg[j] \leftarrow cg + desloc$;
- 10 **Fim**
- 11 $list_points$ recebe o ponto que produz centro de gravidade mais próximo do CGd e todos os outros pontos que possuem a mesma distância do CGd ;
- 12 $indice \leftarrow 1$;
- 13 **Se** tamanho de $list_points > 1$ **então**
- 14 $criterio \leftarrow$ valor produzido pelo critério de desempate escolhido, caso o item seja alocado no ponto $list_points[1]$;
- 15 **Para** m de 2 até tamanho de $list_points$ **faça**
- 16 **Se** $criterio <$ valor produzido pelo critério de desempate escolhido, caso o item seja alocado no ponto $list_points[m]$ **então**
- 17 $indice \leftarrow m$;
- 18 $criterio \leftarrow$ valor produzido pelo critério de desempate escolhido, caso o item seja alocado no ponto $list_points[m]$;
- 19 **Fim**
- 20 **Fim**
- 21 **Fim**
- 22 $insertion_point \leftarrow list_points[indice]$;
- 23 **Devolva** $insertion_point$

gravidade do item, área, peso e o conjunto de polígonos convexos de compõe o item.

A heurística funciona basicamente em duas fases para cada recipiente necessário. Na primeira fase, o algoritmo considera apenas o maior retângulo de região viável do recipiente e empacota os itens, utilizando-se da heurística DESLOC. Uma vez que o algoritmo não consegue empacotar mais nenhum item no retângulo inscrito de um único recipiente, usa-se a segunda fase da heurística.

O algoritmo FASE2, a segunda fase da heurística DESLOC-IRREGULAR, é uma heurística baseada nos trabalhos de Aureliano, Jorge e Andretta (2016) e Jorge *et al.* (2015), diferenciando-se apenas na regra de alocação. O algoritmo considera o recipiente inteiro com os itens alocados na fase anterior. Dessa forma, o algoritmo considera a alocação dos itens de forma fixa e não flutuante, como na primeira fase. Assim, para cada item a ser alocado ainda, o algoritmo mantém uma lista de pontos factíveis em cada recipiente. Essa lista é atualizada, com os vértices dos *NFPs* dos itens já alocados e do item a ser alocado, os pontos de intersecção entre *NFPs* e

também, pontos provenientes da intersecção dos *NFPs* e do *IFP* do item. Assim, para o item corrente, o algoritmo limpa a lista de pontos, como na heurística DESLOC. Mas a escolha do ponto é feita alternando entre quatro critérios, que são: o ponto mais à esquerda e mais abaixo, o ponto mais à direita e mais abaixo, o ponto mais à direita e mais acima e o ponto mais à esquerda e mais acima. Ou seja, um item é alocado no canto inferior esquerdo, o outro, no canto inferior direito, o próximo no canto superior direito, o próximo no canto superior esquerdo, e assim sucessivamente. O objetivo é minimizar o desbalanceamento de forma indireta, como na heurística QUADSWIRL. Assim, a cada iteração o algoritmo ordena a lista de pontos factíveis do item corrente no recipiente atual segundo o critério da vez e aloca o item no primeiro ponto da lista de pontos. Uma vez que um item é alocado, o algoritmo atualiza a lista de pontos factíveis dos itens restantes. Caso a lista de pontos factíveis para o item corrente seja nula, o algoritmo cria um novo recipiente e usa a primeira fase para alocar os itens restantes nesse novo recipiente. Não havendo qualquer ponto factível, o algoritmo usa novamente a segunda fase e assim sucessivamente, até que todos os itens sejam alocados.

A heurística DESLOC-IRREGULAR e FASE2 são formalizados nos algoritmos 4 e 5.

Algoritmo 4: DESLOC-IRREGULAR.

Entrada: *caractRecipiente*, *CGd*, *n*, *listItens*, *listNFP*, *listIFP*;

caractRecipiente: características do recipiente - comprimento, altura e lista de polígonos que representam os defeitos;

CGd: Centro de gravidade desejado;

n: quantidade de itens;

listItens: lista dos itens contendo para cada item *i* ($i = 1, 2, \dots, n$): centro de gravidade do item, área, peso e conjunto de polígonos que compõem o item;

listNFP: lista de *NFPs* de cada par de itens;

listIFP: lista de *IFPs* dos itens com o recipiente;

Saída: *S*.

S: Solução contendo a lista de recipientes usados e, para cada um deles, a lista de itens nele alocados e a posição de cada item no recipiente.

```

1   $S \leftarrow \emptyset$ ;
2   $ret \leftarrow$  retângulo inscrito do recipiente irregular com as características em caractRecipiente;
3  listRet: Cria uma lista com um retângulo com as dimensões de ret;
4   $itensNaoAlocados \leftarrow listItens$ ;
5  Ordene itensNaoAlocados em ordem decrescente em relação ao peso dos itens;
6  Crie uma lista de pontos factíveis (feas_dots) para cada item para cada recipiente;
7   $p \leftarrow 0$ ;
8  Enquanto itensNaoAlocados  $\neq \emptyset$  Faça
9       $p \leftarrow p + 1$ ;
10      $S \leftarrow S \cup \text{DESLOC}(\text{tamanho de } n, itensNaoAlocados, \text{comprimento de } ret, \text{altura de } ret, CGd)$ ;
11     Remove os itens que foram alocados no recipiente p da lista itensNaoAlocados;
12     Se itensNaoAlocados  $\neq \emptyset$  então
13          $S \leftarrow S \cup \text{Fase2}(\text{tamanho de } itensNaoAlocados,$ 
14              $itensNaoAlocados, CGd, listNFP, listIFP, S, \text{recipiente } p)$ ;
14         Remove os itens que foram alocados no recipiente p da lista itensNaoAlocados;
15     Fim
16 Fim
17 Devolva S

```

A implementação da heurística DESLOC-IRREGULAR foi feita utilizando as três variações

da heurística DESLOC, as heurísticas SDESLOC, ADESLOC e RDESLOC. Assim, para cada variação da primeira fase, temos uma variação da DESLOC-IRREGULAR diferente. Quando a heurística DESLOC-IRREGULAR faz uso do subcritério da maior sobreposição dos retângulos envolventes, a heurística é chamada SDESLOC-IRREGULAR. Para o subcritério de menor área, a heurística se chama ADESLOC-IRREGULAR. Por fim, para o subcritério da razão das dimensões do recipiente, a heurística se chama RDESLOC-IRREGULAR.

Algoritmo 5: FASE2 - segunda fase do algoritmo DESLOC-IRREGULAR.

Entrada: n , $listItens$, CGd , $listNFP$, $listIFP$, S , $recip$;

n : quantidade de itens que falta empacotar;

$listItens$: lista dos itens que falta empacotar, contendo para cada item i ($i = 1, 2, \dots, n$) : centro de gravidade do item, área, peso e conjunto de polígonos que compõem o item;

$listNFP$: lista de $NFPs$ de cada par de itens;

$listIFP$: lista de $IFPs$ dos itens com o recipiente;

S : solução até o momento;

$recip$: recipiente a ser atualizado;

Saída: S .

S : solução atualizada;

1 Crie uma lista de pontos factíveis para cada item i ($i = 1, 2, \dots, n$) para $recip$ ($feas_dots[i]$);

2 **Enquanto** i de 1 a n **Faça**

3 Atualize a $feas_dots[i]$ usando $listNFP$ e $listIFP$;

4 Exclua os pontos inviáveis da lista $feas_dots[i]$;

5 **Se** $feas_dots[i] \neq \emptyset$ **então**

6 **Se** resto da divisão de i por 4 é igual a 0 **então**

7 Ordene $listItens$ de acordo com a proximidade com o canto inferior esquerdo;

8 **Fim**

9 **Se** resto da divisão de i por 4 é igual a 1 **então**

10 Ordene $listItens$ de acordo com a proximidade com o canto inferior direito;

11 **Fim**

12 **Se** resto da divisão de i por 4 é igual a 2 **então**

13 Ordene $listItens$ de acordo com a proximidade com o canto superior direito;

14 **Fim**

15 **Se** resto da divisão de i por 4 é igual a 3 **então**

16 Ordene $listItens$ de acordo com a proximidade com o canto superior esquerdo;

17 **Fim**

18 S recebe o item i alocado em $recip$ no ponto $feas_dots[i][0]$;

19 **Fim**

20 $i \leftarrow i + 1$;

21 Vá para linha 2;

22 **Fim**

23 **Devolva** S

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 5.1 apresentamos as instâncias utilizadas e, na Seção 5.2, os resultados obtidos.

5.1 Instâncias

Para avaliar e comparar as heurísticas propostas, foram utilizadas vinte e uma instâncias, sendo dezenove delas baseadas nas instâncias do trabalho de Egeblad (2009) e as outras duas propostas neste trabalho.

As instâncias retiradas do trabalho de Egeblad (2009) são nove instâncias clássicas do problema de corte de itens irregulares em faixas, para as quais o autor define dois tamanhos de recipientes diferentes, tamanho grande e pequeno, e para uma das instâncias (*Shapes0*), além dos dois recipientes, ele define um terceiro recipiente (instância *car*) que, diferentemente dos recipientes anteriores, é irregular. Apesar de que originalmente, para algumas instâncias, é permitido algumas rotações para os itens, como Egeblad (2009) não faz uso das rotações em seus testes computacionais, da mesma forma no nosso trabalho não utilizamos as rotações possíveis para as instâncias. Adicionalmente, o autor define para cada item i de cada instância, o peso $g_i = area_i \times r_i$, sendo r_i um valor aleatório entre $[\frac{1}{2}, 2]$. Portanto, itens com formatos iguais podem possuir pesos diferentes. Porém, o autor não disponibilizou os pesos dos itens das instâncias para o público, tampouco as medidas exatas do recipiente irregular (instância *car*). Assim, os pesos dos itens foram gerados conforme a descrição e o recipiente da instância *car* foi gerado a partir de aproximação, utilizando-se do desenho apresentado em Egeblad (2009). Assim, apesar das instâncias serem baseadas no trabalho dele, não podemos fazer comparações

Figura 33 – Empacotamento da instância *car*.

Fonte: Egeblad (2009).

tão rigorosas, já que os pesos dos itens podem não ser iguais. A Figura 33 apresenta uma imagem da instância *car*.

Na Tabela 1, apresentamos as características das instâncias: número total de itens, formato do recipiente ("retang" para recipientes retangulares e "irreg" para recipientes irregulares), a altura (*A*), o comprimento (*C*), o centro de gravidade desejado (CGd) dos recipientes para cada instância, o retângulo de tolerância (*RetT*), dado em porcentagem, e a unidade para malha. O retângulo de tolerância é a região na qual se deseja que o centro de gravidade final do recipiente esteja. Este retângulo é proporcional às dimensões do recipiente e é centrado em seu CGd. Na Figura 34, o retângulo de tolerância é representado pelo retângulo hachurado, o centro de gravidade desejado do recipiente é o círculo preto e o centro de gravidade final do recipiente é o círculo branco. Na figura, apesar do centro de gravidade final não coincidir com o centro de gravidade desejado, como o centro final está dentro do retângulo de tolerância, então o balanceamento do recipiente é considerado bom o suficiente.

Figura 34 – Representação do retângulo de tolerância.

Fonte: Elaborada pelo autor.

As unidades para as malhas foram definidas em testes preliminares onde definiu-se um valor proporcional aos itens de cada instância e foram feitos testes para verificar a melhora do empacotamento ao se refinar a unidade. A partir do momento que diminuir a unidade não impactava significativamente no empacotamento, a unidade da malha era escolhida.

As instâncias *car-dagli* e *car-fu* foram propostas nesse trabalho para melhor avaliar as heurísticas para recipientes irregulares. Estas instâncias foram construídas utilizando o mesmo formato da instância *car*, respeitando a proporção entre os itens e utilizando-se da mesma proporção usada para aumentar o recipiente pequeno da *shapes0* para *car*.

Outros detalhes das instâncias utilizadas são descritos no Apêndice A.

Tabela 1 – Descrição das instâncias.

Instância	Nº total de itens	Formato	CGd (x/y)	Unid.	RetT (%)	Pequeno		Grande	
						C	A	C	A
Albano	24	Retang.	$\frac{1}{2}/\frac{1}{2}$	25	20	10453,00	5071,50	14934,00	7350,00
Dagli	30	Retang.	$\frac{2}{3}/\frac{1}{2}$	0,1	30	63,00	62,10	90,00	90,00
Fu	12	Retang.	$\frac{1}{2}/\frac{1}{2}$	0,5	40	33,60	39,33	48,00	57,00
Marques	24	Retang.	$\frac{1}{2}/\frac{1}{2}$	0,5	30	81,90	107,64	117,00	156,00
Shapes0	43	Retang.	$\frac{2}{3}/\frac{1}{2}$	0,5	30	63,00	41,40	90,00	60,00
Shapes2	28	Retang.	$\frac{1}{2}/\frac{1}{2}$	0,1	20	28,65	15,52	22,50	40,50
Shirts	99	Retang.	$\frac{2}{3}/\frac{2}{3}$	0,1	20	65,20	41,40	60,00	94,50
Swim	48	Retang.	$\frac{1}{2}/\frac{1}{2}$	10	20	6494,30	5953,32	8628,00	9277,50
Trousers	83	Retang.	$\frac{2}{3}/\frac{1}{2}$	0,5	30	251,50	81,76	364,50	118,50
Car	43	Irreg.	$\frac{2}{3}/\frac{1}{2}$	0,5	20	52,00	80,00	-	-
Car-dagli	30	Irreg.	$\frac{2}{3}/\frac{1}{2}$	0,1	20	95,90	65,80	-	-
Car-fu	12	Irreg.	$\frac{2}{3}/\frac{1}{2}$	0,5	20	57,00	37,00	-	-

5.2 Resultados numéricos

Nesta seção apresentamos os resultados obtidos pelas heurísticas propostas. Para avaliar o balanceamento, utilizamos duas medidas que são utilizadas no trabalho de Egeblad (2009). A primeira é obtida utilizando o retângulo de tolerância, explicado anteriormente, assim avaliamos se o recipiente está balanceado se o centro de gravidade final dele está dentro do retângulo de tolerância. A segunda medida é dada por :

$$\eta^1 = \frac{\|CGd - (\lambda_{x,j}, \lambda_{y,j})\|}{\|(C, L)\|}.$$

Assim, o η é um valor que expressa quão longe ou quão perto o centro de gravidade final de um recipiente está de seu CGd. De forma que quanto menor for o η , mais perto o centro de gravidade final está do CGd. Caso o η de um recipiente seja 0, então o recipiente está perfeitamente balanceado, com o centro de gravidade final coincidindo com o CGd.

5.2.1 Comparação entre as heurísticas discretas

Na Tabela 2 apresentamos os resultados das heurísticas QUADINSIDE, QUADSWIRL e QUAD-BOTTOMLEFT, respectivamente. Para cada heurísticas temos o nome da instância, em seguida temos a quantidade de recipientes utilizados m , o tempo t para execução da heurística em segundos, o valor de η para cada recipiente, a ocupação de cada recipiente dada em porcentagem (ocp), quantidade de itens empacotado em cada recipiente n e, por fim, se o centro de gravidade

¹ No trabalho de Egeblad (2009), η é chamado de COG.

Tabela 2 – Resultados numéricos - heurísticas discretas.

Instância	QUADINSIDE										QUADSWIRL										QUAD-BOTTOMLEFT									
	Recipiente 1					Recipiente 2					Recipiente 1					Recipiente 2					Recipiente 1					Recipiente 2				
	m	t(s)	η	ocp	n r	η	ocp	n r	m	t(s)	η	ocp	n r	η	ocp	n r	m	t(s)	η	ocp	n r	η	ocp	n r	η	ocp	n r			
Recipientes pequenos																														
Albano	2	0,52	0,94	69,07	22 s	0,14	11,39	2 s	2	0,24	0,45	78,10	20 s	0,10	2,36	4 s	2	0,35	4,03	71,70	19 s	0,01	8,77	5 s						
Dagli	2	12,42	11,72	62,88	26 n	4,55	11,63	4 s	2	14,06	12,32	72,32	27 n	0,15	5,46	3 s	2	10,03	10,62	72,47	27 s	0,11	5,31	3 s						
Fu	2	0,00	1,92	56,91	8 s	0,53	25,05	4 s	2	0,00	2,24	70,68	10 s	0,93	11,28	2 s	2	0,00	4,26	56,91	8 s	0,53	25,05	4 s						
Marques	2	0,08	0,98	67,63	19 s	0,30	13,98	5 s	2	0,05	1,75	75,21	22 s	0,44	6,40	2 s	2	0,06	3,03	69,31	21 s	0,31	12,30	3 s						
Shapes0	2	0,03	10,55	50,76	33 s	0,81	10,43	10 s	2	0,03	10,87	58,28	40 s	0,14	2,91	3 s	2	0,02	12,02	55,98	39 s	0,85	5,21	4 s						
Shapes2	2	0,12	0,49	58,02	23 s	0,07	14,84	5 s	2	0,07	1,06	67,24	26 s	0,26	5,62	2 s	2	0,11	5,38	58,02	23 s	0,07	14,84	5 s						
Shirts	2	2,83	10,81	77,89	94 n	0,09	2,13	5 s	1	6,62	11,07	80,02	99 n	-	-	-	2	2,47	13,72	77,61	91 n	0,07	2,41	8 s						
Swim	2	9,17	0,51	57,84	42 s	0,02	7,98	6 s	2	8,01	0,44	61,54	43 s	0,12	4,27	5 s	2	5,36	2,36	57,51	41 s	0,06	16,41	4 s						
Trousers	2	0,76	6,31	66,05	60 s	0,10	17,62	4 s	2	0,60	9,96	80,17	63 s	0,16	3,50	1 s	2	0,72	12,15	67,26	60 s	0,06	16,41	4 s						
Recipientes grandes																														
Albano	1	0,18	0,38	38,86	24 s	-	-	-	1	0,15	0,38	38,86	24 s	-	-	-	1	0,22	0,38	38,86	24 s	-	-	-						
Dagli	1	10,22	6,72	37,57	30 s	-	-	-	1	21,52	6,72	37,57	30 s	-	-	-	1	8,46	6,72	37,57	30 s	-	-	-						
Fu	1	0,00	1,48	39,58	12 s	-	-	-	1	0,00	1,48	39,58	12 s	-	-	-	1	0,00	1,48	39,58	12 s	-	-	-						
Marques	1	0,03	1,93	39,41	24 s	-	-	-	1	0,03	1,93	39,41	24 s	-	-	-	1	0,02	1,93	39,41	24 s	-	-	-						
Shapes0	1	0,03	9,59	29,56	43 s	-	-	-	1	0,04	9,59	29,56	43 s	-	-	-	1	0,03	9,59	29,56	43 s	-	-	-						
Shapes2	1	0,11	0,20	35,56	28 s	-	-	-	1	0,09	0,20	35,56	28 s	-	-	-	1	0,06	0,20	35,56	28 s	-	-	-						
Shirts	1	4,72	9,64	38,10	99 n	-	-	-	1	17,75	9,64	38,10	99 n	-	-	-	1	3,88	9,64	38,10	99 n	-	-	-						
Swim	1	11,61	0,99	31,79	48 s	-	-	-	1	10,06	0,99	31,79	48 s	-	-	-	1	8,67	0,99	31,79	48 s	-	-	-						
Trousers	1	0,54	8,08	39,84	64 s	-	-	-	1	0,86	8,08	39,84	64 s	-	-	-	1	0,50	8,08	39,84	64 s	-	-	-						
Recipientes irregulares																														
Car	2	0,03	9,58	48,66	39 n	0,47	4,53	4 s	1	0,03	11,54	53,19	43 n	-	-	-	1	0,04	12,55	53,19	43 n	-	-	-						
Car-dagli	2	11,02	9,89	65,92	29 s	8,05	1,66	1 s	1	16,62	11,75	67,59	30 s	-	-	-	1	9,10	13,88	67,59	30 s	-	-	-						
Car-fu	2	0,01	1,92	60,26	9 s	8,62	10,83	3 s	2	0,00	4,05	58,09	10 s	11,20	13,00	2 s	2	0,01	4,86	7,32	9 s	1,59	13,79	3 s						

final do recipiente está dentro do retângulo de tolerância ou não (r) (s caso esteja dentro, n , caso contrário).

Sobre o tempo de execução das heurísticas, os valores encontrados são bastante baixos, sendo o maior de 21s para instância *dagli* para a heurística QUADSWIRL.

É fácil perceber que os resultados para os recipientes grandes das três heurísticas foram idênticos, já que as três heurísticas empacotam novamente o último recipiente utilizando a heurística QUADINSIDE e movem o bloco de itens em direção ao CGd.

Para os recipientes de menores dimensões, como a heurística QUADSWIRL conseguiu empacotar a instância *shirts* em apenas um recipiente, assim não é possível fazer comparações em relação ao balanceamento com as outras duas heurísticas e a QUADSWIRL. Para as demais instâncias, temos que as heurísticas QUADINSIDE e QUADSWIRL obtiveram recipientes balanceados, isto é, com o centro de gravidade final dentro do retângulo de tolerância, para sete das oito instâncias, apenas não foi possível para a *dagli*. Já a heurística QUAD-BOTTOMLEFT obteve recipientes balanceados para as oito instâncias. Em relação aos valores de η , os melhores balanceamentos para o primeiro recipiente foram produzidos pela QUADINSIDE, perdendo para apenas três das oito instâncias. Uma destas, a instância *dagli*, teve o melhor balanceamento produzido pela heurística QUAD-BOTTOMLEFT e duas delas, as instâncias *albano* e *swim*, foram melhor balanceadas pela heurística QUADSWIRL.

Já em relação ao empacotamento, a heurística QUADSWIRL realizou os empacotamentos utilizando apenas um recipiente para as instâncias *shirts* e para todas as instâncias com recipientes grandes. Já as heurísticas QUADINSIDE e QUAD-BOTTOMLEFT utilizaram um recipiente apenas para as instâncias com recipientes grandes. Em relação à taxa de ocupação, a heurística QUADSWIRL obteve as melhores taxas para oito das nove instâncias com recipientes pequenos. Coincidentemente, a QUAD-BOTTOMLEFT obteve melhor empacotamento para a instância *dagli*, na qual também obteve melhor balanceamento.

Em relação às instâncias com recipientes irregulares, as heurísticas QUADSWIRL e QUAD-BOTTOMLEFT conseguiram empacotar os itens das instâncias *car* e *car-dagli* em apenas um recipiente e utilizou dois recipientes para a instância *car-fu*. A heurística QUADINSIDE utilizou dois recipientes para as três instâncias testadas. Da mesma forma, a comparação de balanceamento só pode ser feita quando as heurísticas utilizam o mesmo número de recipientes. Assim, para as instâncias *car* e *car-dagli*, a heurística QUADSWIRL obteve melhores valores para η nas duas instâncias que a QUAD-BOTTOMLEFT. Já para a instância *car-fu*, a melhor taxa de ocupação e o melhor valor de η foram obtidos pela heurística QUADINSIDE.

Dessa forma, percebemos que caso se deseje recipientes com melhor ocupação que balanceamento, deve-se optar pela heurística QUADSWIRL. Caso contrário, opta-se pela heurística QUADINSIDE.

5.2.2 Comparação entre as heurísticas contínuas

Os resultados produzidos pelas heurísticas contínuas SDESLOC, ADESLOC e RDESLOC são apresentados na Tabela 3. Já os resultados para recipientes irregulares produzidos pelas adaptações destas heurísticas (SDESLOC-IRREGULAR, ADESLOC-IRREGULAR e RDESLOC-IRREGULAR) são apresentados na Tabela 4. As tabelas possuem a mesma estrutura da tabela anterior.

Em relação ao tempo, vemos que com exceção da instância *swim*, as heurísticas obtiveram um tempo computacional baixo. Já em relação a instância *swim*, os tempos obtidos são bastante elevados, já que a instância possui itens fortemente não convexos, isto é, é necessário dividir o item em muitos polígonos convexos, e as heurísticas DESLOC e DESLOC-IRREGULAR calculam o centro de gravidade das posições factíveis que são geradas a partir dos vértices dos *NFPs* de todos os polígonos convexos de todos os itens. Assim, consideramos isso uma exceção, já que as heurísticas foram testadas com outras instâncias que possuem itens não convexos e os tempos para tais não foram elevados.

As heurísticas contínuas possuem desempenho bastante similar entre si. Para os recipientes grandes, as três heurísticas empacotaram todos os itens de todas as instâncias em apenas um recipiente e obtiveram $\eta = 0$. Logo, todos os recipientes possuem centro de gravidade final coincidente com o centro de gravidade desejado.

Para as instâncias com recipientes pequenos, vemos que as heurísticas ADESLOC e RDESLOC utilizaram dois recipientes para todas as instâncias com exceção da instância *shirts*. Já a heurística SDESLOC utilizou dois recipientes para todas, sem exceção. Ainda sobre o empacotamento vemos que a heurística SDESLOC obteve as melhores taxas de ocupação para seis instâncias, a heurística ADESLOC obteve melhores valores para cinco instâncias e a RDESLOC apenas para três instâncias. Note que houve empate nas melhores taxas de ocupação. Em contrapartida, a RDESLOC obteve melhores valores para η para seis das nove instâncias.

Analisando os resultados para os recipientes irregulares, temos que as heurísticas utilizaram dois recipientes para todas as instâncias, com exceção das heurísticas SDESLOC-IRREGULAR e ADESLOC-IRREGULAR para a instância *car* que utilizaram apenas um recipiente. Para essa instância, o melhor balanceamento foi obtido pela SDESLOC-IRREGULAR. Para as outras duas instâncias *car-dagli* e *car-fu*, temos que a SDESLOC-IRREGULAR obteve melhor taxa de ocupação para a instância *car-dagli* e o melhor valor de η foi obtido pela ADESLOC-IRREGULAR. Para a instância *car-fu*, as três heurísticas obtiveram mesma taxa de ocupação, mas a heurística SDESLOC-IRREGULAR obteve um balanceamento levemente melhor que as outras.

Concluimos que os critérios utilizados nas heurísticas são bastante similares, mas as heurísticas SDESLOC e ADESLOC obtêm melhores empacotamentos e a RDESLOC melhores balanceamentos.

Tabela 3 – Resultados numéricos - heurísticas contínuas.

SDESLOC										ADESLOC										RDESLOC																			
Recipiente 1										Recipiente 2										Recipiente 1										Recipiente 2									
Instância	m	t(s)	η	ocp	n	r	η	ocp	n	r	t(s)	η	ocp	n	r	η	ocp	n	r	t(s)	η	ocp	n	r	η	ocp	n	r											
Recipientes pequenos																																							
Albano	2	2,45	0,00	74,78	23	s	0,00	5,74	1	s	2	3,31	2,86	73,34	20	s	0,00	5,46	4	s	2	3,10	0,00	69,47	22	s	0,00	11,05	2	s									
Dagli	2	3,51	7,03	71,94	27	n	0,00	5,62	3	s	2	3,68	6,70	72,11	27	s	0,00	5,46	3	s	2	3,09	4,38	72,11	27	s	0,00	5,46	3	s									
Fu	2	0,03	1,82	60,84	8	s	0,00	21,11	4	s	2	0,03	2,00	66,06	9	s	0,00	15,89	3	s	2	0,04	0,54	69,77	10	s	0,00	12,18	2	s									
Marques	2	2,02	0,82	76,05	23	s	0,00	5,56	1	s	2	2,20	0,00	76,05	23	s	0,00	5,56	1	s	2	1,98	0,00	70,15	22	s	0,00	11,46	2	s									
Shapes0	2	11,53	8,07	57,51	40	s	0,00	3,68	3	s	2	11,94	9,83	54,44	37	s	0,00	6,75	6	s	2	11,72	9,34	55,98	39	s	0,00	5,21	4	s									
Shapes2	2	2,33	2,29	71,52	27	s	0,00	1,35	1	s	2	2,33	0,42	66,46	26	s	0,00	6,41	2	s	2	2,27	0,00	62,63	24	s	0,00	10,23	4	s									
Shirts	2	27,91	9,97	79,50	97	n	0,00	0,52	2	s	1	27,75	11,10	80,02	99	n	-	-	-	-	1	26,27	11,36	80,02	99	n	-	-	-	-									
Swim	2	6730,09	0,87	61,70	44	s	0,00	4,11	4	s	2	5045,18	0,00	62,79	45	s	0,00	3,02	3	s	2	4294,44	0,00	60,90	43	s	0,00	4,91	5	s									
Trousers	2	17,76	6,53	76,68	62	s	0,00	7,00	2	s	2	20,84	6,50	76,68	62	s	0,00	7,00	2	s	2	0,72	12,15	67,26	60	s	0,06	16,41	4	s									
Recipientes grandes																																							
Albano	1	3,19	0,00	38,86	24	s	-	-	-	-	1	3,63	0,00	38,86	24	s	-	-	-	-	1	4,97	0,00	38,86	24	s	-	-	-	-									
Dagli	1	5,45	0,00	37,57	30	s	-	-	-	-	1	5,31	0,00	37,57	30	s	-	-	-	-	1	5,66	0,00	37,57	30	s	-	-	-	-									
Fu	1	0,05	0,00	39,58	12	s	-	-	-	-	1	0,06	0,00	39,58	12	s	-	-	-	-	1	0,05	0,00	39,58	12	s	-	-	-	-									
Marques	1	3,05	0,00	39,41	24	s	-	-	-	-	1	3,65	0,00	39,41	24	s	-	-	-	-	1	3,28	0,00	39,41	24	s	-	-	-	-									
Shapes0	1	3,05	0,00	29,56	43	s	-	-	-	-	1	3,65	0,00	29,56	43	s	-	-	-	-	1	22,10	0,00	29,56	43	s	-	-	-	-									
Shapes2	1	2,87	0,00	35,56	28	s	-	-	-	-	1	2,84	0,00	35,56	28	s	-	-	-	-	1	2,98	0,00	35,56	28	s	-	-	-	-									
Shirts	1	43,85	0,00	38,10	99	n	-	-	-	-	1	47,15	0,00	38,10	99	n	-	-	-	-	1	44,17	0,00	38,10	99	n	-	-	-	-									
Swim	1	8492,39	0,00	31,79	48	s	-	-	-	-	1	5860,14	0,00	31,79	48	s	-	-	-	-	1	4759,81	0,00	31,79	48	s	-	-	-	-									
Trousers	1	23,67	0,00	39,84	64	s	-	-	-	-	1	27,19	0,00	39,84	64	s	-	-	-	-	1	26,52	0,00	39,84	64	s	-	-	-	-									

Tabela 4 – Resultados numéricos - heurísticas contínuas irregulares.

Instância	m	t(s)	Recipiente 1				Recipiente 2			
			$\eta(\%)$	ocp(%)	n	r	$\eta(\%)$	ocp(%)	n	r
Heurística SDESLOC-IRREGULAR										
Car	1	21,36	11,13	53,19	43	n	-	-	-	-
Car-dagli	2	4,53	13,12	63,37	27	n	0,00	4,20	3	s
Car-fu	2	0,06	8,26	58,09	10	s	0,00	13,00	2	s
Heurística ADESLOC-IRREGULAR										
Car	1	24,95	13,71	53,19	43	n	-	-	-	-
Car-dagli	2	7,09	9,79	62,37	29	n	0,00	5,22	1	s
Car-fu	2	0,06	8,35	58,09	10	n	0,00	13,00	2	s
Heurística RDESLOC-IRREGULAR										
Car	2	17,20	13,10	52,43	42	n	0,00	0,94	1	s
Car-dagli	2	6,04	12,38	61,10	28	n	0,00	6,49	2	s
Car-fu	2	0,06	8,35	58,09	10	n	0,00	13,00	2	s

5.2.3 Comparação entre as heurísticas propostas

Ao compararmos todas as heurísticas propostas para recipientes regulares podemos ver que as heurísticas contínuas produzem melhores balanceamentos para os recipientes. Para os recipientes grandes, as heurísticas discretas conseguiram $\eta = 0$ apenas para a instância *fu*. Já as heurísticas contínuas obtiveram $\eta = 0$ para todas as instâncias com recipientes grandes. Para os recipientes pequenos, as heurísticas contínuas, no geral, obtêm melhores resultados, perdendo apenas para uma instância (*trousers*) para a heurística QUADINSIDE.

Em relação à ocupação dos recipientes, as heurísticas obtêm os mesmos resultados para os recipientes grandes. Logo, a comparação é feita para os recipientes pequenos. A heurística QUADINSIDE obtém melhores taxas de ocupação que qualquer outro método, ganhando em sete das nove instâncias e empatando na instância *shirts* com as heurísticas DESLOC.

Para as heurísticas com recipientes irregulares, as heurísticas discretas produzem empacotamentos melhores que as heurísticas contínuas, utilizando menor número de recipientes para duas das três instâncias testadas, contra apenas uma das heurísticas contínuas. Em relação ao balanceamento, para a instância *car*, a heurística SDESLOC-IRREGULAR obteve menor η . Para a instância *car-fu*, a heurística QUADINSIDE obteve melhor balanceamento.

Pode-se concluir que as heurísticas contínuas produzem melhores balanceamentos que as heurísticas discretas, por causa da estratégia de balancear os recipientes de forma direta, contra a forma indireta das heurísticas discretas. Em relação ao empacotamento, a heurística QUADINSIDE produz melhores resultados, mas como os resultados no empacotamento não são

acompanhados de bons balanceamentos, a heurística não é de grande vantagem. Já as heurísticas contínuas possuem um empacotamento não tão bom quanto a QUADINSIDE, mas com bons resultados para o balanceamento. Assim, as heurísticas DESLOCS foram consideradas melhores, no geral, que as heurísticas QUADS. Logo, a comparação com a literatura foi feita apenas com as heurísticas consideradas melhores, as heurísticas contínuas.

As soluções obtidos pelas heurísticas DESLOC e DESLOC-IRREGULAR, consideradas melhores, são apresentadas no Apêndice B.

5.2.4 Comparação com a literatura

Nesta seção comparamos os resultados produzidos pelas três heurísticas contínuas, SDESLOC, ADESLOC e RDESLOC, com o trabalho de Egeblad (2009). As dezenove instâncias utilizadas no trabalho de Egeblad (2009) não são exatamente iguais às utilizadas neste trabalho, já que o autor não disponibilizou todas as informações das instâncias, como o peso dos itens e o formato exato da instância irregular. Os dados foram reproduzidos conforme descrito no trabalho de Egeblad (2009), mas não temos garantias de que os dados são os mesmos.

O empacotamento produzido pelas heurística de Egeblad (2009) é superior aos empacotamentos produzidos pelas heurísticas propostas, já que o algoritmo de Egeblad foi capaz de empacotar todos os itens de todas as instâncias utilizando apenas um recipiente. As heurísticas propostas conseguiram a mesma taxa de ocupação apenas para as instâncias com recipientes grandes e para as instâncias *shirts* e *car*.

A comparação entre o balanceamento produzido pelas heurísticas só é possível quando elas se utilizam do mesmo número de recipientes. Para a instância *car*, o algoritmo de Egeblad (2009) conseguiu melhor balanceamento. Já para a instância *shirts*, o menor η foi obtido pela heurística ADESLOC com valor igual a 11,1, mesmo valor que Egeblad (2009). Para todas as instâncias com recipientes grandes, as heurísticas DESLOCS tiveram melhores resultados no balanceamento, obtendo $\eta = 0$ para todas as instâncias, com resultados superiores ao algoritmo de Egeblad (2009), que só obteve $\eta = 0$ para a instância *Marques*.

Assim, é possível observar que as heurísticas DESLOC produzem balanceamentos bons e seria interessante trabalhar com outras ordenações dos itens ou outros critérios para obtenção e escolha dos pontos para alocação dos itens de forma que as heurísticas consigam maiores taxas de ocupação.

Tabela 5 – Resultados numéricos de Egeblad (2009) e da heurística RDESLOC.

Instância	EGEBLAD (2009)				RDESLOC							
	Recipiente 1				Recipiente 1					Recipiente 2		
	<i>m</i>	<i>t</i>	η	<i>ocp</i>	<i>m</i>	<i>t</i>	η	<i>ocp</i>	<i>n</i>	η	<i>ocp</i>	<i>n</i>
Recipientes pequenos												
Albano	1	300	1,56	80,5	2	3,10	0,00	69,47	22	0,00	11,05	2
Dagli	1	300	6,93	77,8	2	3,09	4,38	72,11	27	0,00	5,46	3
Fu	1	300	3,09	82,0	2	0,04	0,54	69,77	10	0,00	12,18	2
Marques	1	300	1,54	81,6	2	1,98	0,00	70,15	22	0,00	11,46	2
Shapes0	1	300	8,96	61,2	2	11,72	9,34	55,98	39	0,00	5,21	4
Shapes2	1	300	1,89	73,6	2	2,27	0,00	62,63	24	0,00	10,23	4
Shirts	1	300	11,1	80,0	1	26,27	11,36	80,02	99	-	-	-
Swim	1	300	1,12	65,8	2	4294,44	0,00	60,90	43	0,00	4,91	5
Trousers	1	300	10,7	82,5	2	16,61	7,04	75,47	62	0,00	8,21	2
Recipientes grandes												
Albano	1	300	0,67	38,86	1	4,97	0,00	38,89	24	-	-	-
Dagli	1	300	1,95	37,57	1	5,66	0,00	37,57	30	-	-	-
Fu	1	300	0,77	39,58	1	0,05	0,00	39,58	12	-	-	-
Marques	1	300	0,00	39,41	1	3,28	0,00	39,41	24	-	-	-
Shapes0	1	300	0,79	29,56	1	22,10	0,00	29,56	43	-	-	-
Shapes2	1	300	0,57	35,56	1	2,98	0,00	35,56	28	-	-	-
Shirts	1	300	1,22	38,10	1	44,17	0,00	38,10	99	-	-	-
Swim	1	300	0,52	31,79	1	4759,81	0,00	31,79	48	-	-	-
Trousers	1	300	3,90	39,84	1	26,52	0,00	39,84	64	-	-	-
Recipiente irregular												
Car	1	300	6,42	66,66	2	17,20	13,10	52,43	42	0,00	0,94	1

CONSIDERAÇÕES E TRABALHOS FUTUROS

Neste trabalho, estudamos o problema bidimensional de empacotamento de itens irregulares em recipientes retangulares e irregulares considerando o balanceamento de carga. O objetivo é empacotar todos os itens utilizando o menor número de recipientes possível, de forma que o centro de gravidade final dos recipientes fique o mais próximo do centro desejado. Além disso, não pode haver sobreposição entre itens e eles devem estar inteiramente contidos no recipiente.

O problema ainda é pouco estudado na literatura, sendo o trabalho de [Egeblad \(2009\)](#), até onde vai o nosso conhecimento, o único a resolver tal problema. Assim, o levantamento da literatura realizado consistiu em estudar artigos que resolvem o problema de corte e empacotamento de itens irregulares e outros que resolvem o problema de empacotamento de itens regulares considerando o balanceamento de carga.

Apresentamos três heurísticas bases, com três variações cada, para a resolução do problema. A heurística QUAD e suas variações, as heurísticas QUAD-BOTTOMLEFT, QUADINSIDE e QUADSWIRL, utilizam malha para representar os itens e recipientes e para verificar sobreposição entre os itens. O balanceamento da carga é feito de forma indireta, dividindo o recipiente em quatro quadrantes e os itens são alocados revezando entre eles. As três heurísticas são capazes de resolver tanto o problema com recipientes retangulares quanto irregulares.

A heurística DESLOC e suas variações, SDESLOC, ADESLOC e RDESLOC, representam os itens por meio de polígonos e utilizam *NFP* para evitar a sobreposição entre eles. As heurísticas constroem a solução item a item sem posições fixas, como um bloco de itens flutuante, que não deve ultrapassar uma largura e um comprimento predefinidos. As heurísticas deslocam o bloco de itens em direção ao centro de gravidade desejado para que se obtenha melhores balanceamentos. Também, a escolha do ponto para se alocar um item considera o centro de gravidade que o bloco de itens terá, caso o item seja alocado ali. As heurísticas se diferenciam na escolha do ponto em que cada item será alocado. Essas três heurísticas foram desenvolvidas para a resolução do problema com recipiente regular.

Para recipientes irregulares, as três heurísticas anteriores foram adaptadas gerando a heurística DESLOC-IRREGULAR e suas variações, SDESLOC-IRREGULAR, ADESLOC-IRREGULAR e RDESLOC-IRREGULAR, que empacotam os itens em duas fases. A primeira fase utiliza as heurísticas DESLOCS e considera apenas o retângulo inscrito do recipiente e a segunda fase considera o recipiente inteiro e o balanceamento é feito de forma indireta.

As nove heurísticas foram comparadas com o trabalho de [Egeblad \(2009\)](#) utilizando dezenove instâncias apresentadas em seu trabalho. Propusemos também duas instâncias com recipientes irregulares. Os resultados mostram que a heurística QUADINSIDE é melhor que as outras heurísticas discretas no balanceamento e que a heurística QUADSWIRL tem melhores taxas de ocupação. No geral, nenhuma delas produz resultados muito bons para o balanceamento dos recipientes e atribuímos isso ao fato do balanceamento ser feito de forma indireta.

As heurísticas SDESLOC, RDESLOC e ADESLOC, para recipientes regulares, produzem resultados bastante similares entre si e, no geral, os empacotamentos são inferiores aos da literatura, mas possui desempenho competitivo em relação ao balanceamento. As três obtiveram melhores resultados para nove das dezenove instâncias testadas (todas as instâncias com recipientes grandes) conseguindo melhorar os resultados da literatura, de forma que o centro de gravidade final dos recipientes coincidiram com os centros de gravidade desejados para nove delas, que são as instâncias com recipientes grandes.

As heurísticas para recipientes irregulares, tanto as heurísticas discretas como as heurísticas contínuas adaptadas, foram comparadas com a literatura utilizando apenas uma instância, a instância *car*. Nenhuma das heurísticas propostas obteve resultados melhores que o trabalho de [Egeblad \(2009\)](#) para esta instância.

Por fim, como trabalhos futuros, sugerimos a implementação de uma meta-heurística baseado na heurística DESLOC, explorando novas ordenações para os itens, outras ferramentas para obtenção de pontos para alocação dos itens e outros critérios para a escolha dos pontos, a fim de melhorar o desempenho do empacotamento. Também, um método exato para resolver o problema, já que ainda não existe na literatura, até onde sabemos.

REFERÊNCIAS

- ALVAREZ-VALDES, R.; MARTINEZ, A.; TAMARIT, J. A branch & bound algorithm for cutting and packing irregularly shaped pieces. **International Journal of Production Economics**, Elsevier, v. 145, n. 2, p. 463–477, 2013. Citado nas páginas 34 e 35.
- ALVES, C.; BRÁS, P.; CARVALHO, J. V. de; PINTO, T. New constructive algorithms for leather nesting in the automotive industry. **Computers & Operations Research**, Elsevier, v. 39, n. 7, p. 1487–1505, 2012. Citado na página 37.
- AMBROSINO, D.; SCIOMACHEN, A.; TANFANI, E. Stowing a containership: the master bay plan problem. **Transportation Research Part A: Policy and Practice**, Elsevier, v. 38, n. 2, p. 81–99, 2004. Citado na página 39.
- _____. A decomposition heuristics for the container ship stowage problem. **Journal of Heuristics**, Springer, v. 12, n. 3, p. 211–233, 2006. Citado na página 39.
- AMIOUNY, S. V.; III, J. J. B.; VATE, J. H. V.; ZHANG, J. Balanced loading. **Operations Research**, INFORMS, v. 40, n. 2, p. 238–246, 1992. Citado nas páginas 17, 20 e 39.
- AURELIANO, F.; JORGE, A.; ANDRETTA, M. Métodos heurísticos para o problema de corte de luvas de couro. **XLVIII SBPO - Simpósio Brasileiro de Pesquisa Operacional, 2016, Vitória/ES**, XLVIII Anais do SBPO, p. 3952–3963, 2016. Citado nas páginas 48 e 52.
- BABU, A. R.; BABU, N. R. A generic approach for nesting of 2-d parts in 2-d sheets using genetic and heuristic algorithms. **Computer-Aided Design**, Elsevier, v. 33, n. 12, p. 879–891, 2001. Citado na página 27.
- BAKER, B. S.; COFFMAN JR, E. G.; RIVEST, R. L. Orthogonal packings in two dimensions. **SIAM Journal on Computing**, SIAM, v. 9, n. 4, p. 846–855, 1980. Citado na página 31.
- BALDACCI, R.; BOSCHETTI, M. A.; GANOVELLI, M.; MANIEZZO, V. Algorithms for nesting with defects. **Discrete Applied Mathematics**, Elsevier, v. 163, p. 17–33, 2014. Citado nas páginas 22, 36 e 37.
- BENNELL, J. A.; OLIVEIRA, J. F. The geometry of nesting problems: A tutorial. **European Journal of Operational Research**, Elsevier, v. 184, n. 2, p. 397–415, 2008. Citado nas páginas 21 e 22.
- _____. A tutorial in irregular shape packing problems. **Journal of the Operational Research Society**, Nature Publishing Group, v. 60, n. 1, p. S93–S105, 2009. Citado na página 31.
- BENNELL, J. A.; SONG, X. A beam search implementation for the irregular shape packing problem. **Journal of Heuristics**, Springer, v. 16, n. 2, p. 167–188, 2010. Citado na página 32.
- BŁAŻEWICZ, J.; HAWRYLUK, P.; WALKOWIAK, R. Using a tabu search approach for solving the two-dimensional irregular cutting problem. **Annals of Operations Research**, Springer, v. 41, n. 4, p. 313–325, 1993. Citado na página 33.

- BORTFELDT, A.; WÄSCHER, G. Constraints in container loading—a state-of-the-art review. **European Journal of Operational Research**, Elsevier, v. 229, n. 1, p. 1–20, 2013. Citado na página [20](#).
- BURKE, E.; HELLIER, R.; KENDALL, G.; WHITWELL, G. A new bottom-left-fill heuristic algorithm for the two-dimensional irregular packing problem. **Operations Research**, INFORMS, v. 54, n. 3, p. 587–601, 2006. Citado nas páginas [31](#), [32](#) e [33](#).
- CARRAVILLA, M. A.; RIBEIRO, C.; OLIVEIRA, J. F. Solving nesting problems with non-convex polygons by constraint logic programming. **International Transactions in Operational Research**, Wiley Online Library, v. 10, n. 6, p. 651–663, 2003. Citado na página [34](#).
- CHERRI, L. H.; MUNDIM, L. R.; ANDRETTA, M.; TOLEDO, F. M.; OLIVEIRA, J. F.; CARRAVILLA, M. A. Robust mixed-integer linear programming models for the irregular strip packing problem. **European Journal of Operational Research**, Elsevier, v. 253, n. 3, p. 570–583, 2016. Citado nas páginas [25](#) e [35](#).
- CUNINGHAME-GREEN, R. Geometry, shoemaking and the milk tray problem. **New Scientist**, v. 123, n. 1677, p. 50–53, 1989. Citado nas páginas [24](#), [25](#) e [28](#).
- DALALAH, D.; KHRAIS, S.; BATAINEH, K. Waste minimization in irregular stock cutting. **Journal of Manufacturing Systems**, Elsevier, v. 33, n. 1, p. 27–40, 2014. Citado nas páginas [36](#) e [37](#).
- DANIELS, K.; MILENKOVIC, V.; LI, Z. Multiple containment methods. **Center for Research in Computing Technology, Harvard University, Cambridge, MA**, Citeseer, p. 12–94, 1994. Citado na página [34](#).
- DOWSLAND, K.; DOWSLAND, W.; BENNELL, J. Jostling for position: local improvement for irregular cutting patterns. **Journal of the Operational Research Society**, p. 647–658, 1998. Citado na página [32](#).
- DOWSLAND, K. A.; VAID, S.; DOWSLAND, W. B. An algorithm for polygon placement using a bottom-left strategy. **European Journal of Operational Research**, Elsevier, v. 141, n. 2, p. 371–381, 2002. Citado na página [31](#).
- EGEBLAD, J. Placement of two-and three-dimensional irregular shapes for inertia moment and balance. **International Transactions in Operational Research**, Wiley Online Library, v. 16, n. 6, p. 789–807, 2009. Citado nas páginas [17](#), [18](#), [19](#), [29](#), [40](#), [55](#), [56](#), [57](#), [63](#), [64](#), [65](#) e [66](#).
- EGEBLAD, J.; GARAVELLI, C.; LISI, S.; PISINGER, D. Heuristics for container loading of furniture. **European Journal of Operational Research**, Elsevier, v. 200, n. 3, p. 881–892, 2010. Citado na página [39](#).
- EGEBLAD, J.; NIELSEN, B. K.; ODGAARD, A. Fast neighborhood search for two-and three-dimensional nesting problems. **European Journal of Operational Research**, Elsevier, v. 183, n. 3, p. 1249–1266, 2007. Citado nas páginas [29](#) e [33](#).
- ELKERAN, A. A new approach for sheet nesting problem using guided cuckoo search and pairwise clustering. **European Journal of Operational Research**, Elsevier, v. 231, n. 3, p. 757–769, 2013. Citado na página [34](#).

FERNÁNDEZ, A.; GIL, C.; BAÑOS, R.; MONTOYA, M. G. A parallel multi-objective algorithm for two-dimensional bin packing with rotations and load balancing. **Expert Systems with Applications**, Elsevier, v. 40, n. 13, p. 5169–5180, 2013. Citado nas páginas 37 e 38.

FISCHETTI, M.; LUZZI, I. Mixed-integer programming models for nesting problems. **Journal of Heuristics**, Springer, v. 15, n. 3, p. 201–226, 2009. Citado na página 34.

GOMES, A. M.; OLIVEIRA, J. F. A 2-exchange heuristic for nesting problems. **European Journal of Operational Research**, Elsevier, v. 141, n. 2, p. 359–370, 2002. Citado nas páginas 31 e 32.

_____. Solving irregular strip packing problems by hybridising simulated annealing and linear programming. **European Journal of Operational Research**, Elsevier, v. 171, n. 3, p. 811–829, 2006. Citado na página 33.

HALAVATI, R.; SHOURAKI, S. B.; NOROOZIAN, M.; ZADEH, S. H. Optimizing allocation of two dimensional irregular shapes using an agent based approach. **World Academy of Science, Engineering and Technology**, Citeseer, v. 11, 2005. Citado na página 36.

HALAVATI, R.; SHOURAKI, S. B.; ZADEH, S. H. A novel evolutionary approach for two dimensional bin packing. **Computer Science and Engineering**, p. 58–67, 2008. Citado na página 36.

IKONEN, I.; BILES, W. E.; KUMAR, A.; WISSEL, J. C.; RAGADE, R. K. A genetic algorithm for packing three-dimensional non-convex objects having cavities and holes. In: **ICGA**. [S.l.: s.n.], 1997. p. 591–598. Citado na página 39.

IMAMICHI, T.; YAGIURA, M.; NAGAMOCCHI, H. An iterated local search algorithm based on nonlinear programming for the irregular strip packing problem. **Discrete Optimization**, Elsevier, v. 6, n. 4, p. 345–361, 2009. Citado na página 34.

JORGE, A.; MUNDIM, L.; CHERRI, L. H.; ANDRETTA, M. O problema de corte de itens irregulares: aplicação na indústria de aventais e forros de luva. **XLVII SBPO - Simpósio Brasileiro de Pesquisa Operacional, 2015, Porto de galinhas/PE**, Anais do XLVII SBPO, 2015. Citado nas páginas 48 e 52.

KALUZNY, B. L.; SHAW, R. Optimal aircraft load balancing. **International Transactions in Operational Research**, Wiley Online Library, v. 16, n. 6, p. 767–787, 2009. Citado nas páginas 17 e 37.

LARA, O. D.; LABRADOR, M. A. A multiobjective ant colony-based optimization algorithm for the bin packing problem with load balancing. In: **IEEE. Evolutionary Computation (CEC), 2010 IEEE Congress on**. [S.l.], 2010. p. 1–8. Citado na página 39.

LEÃO, A. A.; TOLEDO, F. M.; OLIVEIRA, J. F.; CARRAVILLA, M. A. A semi-continuous mip model for the irregular strip packing problem. **International Journal of Production Research**, Taylor & Francis, v. 54, n. 3, p. 712–721, 2016. Citado na página 35.

LEUNG, S. C.; LIN, Y.; ZHANG, D. Extended local search algorithm based on nonlinear programming for two-dimensional irregular strip packing problem. **Computers & Operations Research**, Elsevier, v. 39, n. 3, p. 678–686, 2012. Citado na página 34.

- LIU, D.; TAN, K. C.; HUANG, S.; GOH, C. K.; HO, W. K. On solving multiobjective bin packing problems using evolutionary particle swarm optimization. **European Journal of Operational Research**, Elsevier, v. 190, n. 2, p. 357–382, 2008. Citado na página 37.
- LIU, J.; YUE, Y.; DONG, Z.; MAPLE, C.; KEECH, M. A novel hybrid tabu search approach to container loading. **Computers & Operations Research**, Elsevier, v. 38, n. 4, p. 797–807, 2011. Citado na página 38.
- LÓPEZ-CAMACHO, E.; OCHOA, G.; TERASHIMA-MARÍN, H.; BURKE, E. K. An effective heuristic for the two-dimensional irregular bin packing problem. **Annals of Operations Research**, Springer, v. 206, n. 1, p. 241–264, 2013. Citado nas páginas 35 e 36.
- MARTINS, T.; TSUZUKI, M. d. S. G. Simulated annealing applied to the irregular rotational placement of shapes over containers with fixed dimensions. **Expert Systems with Applications**, Elsevier, v. 37, n. 3, p. 1955–1972, 2010. Citado na página 37.
- MONGEAU, M.; BES, C. Optimization of aircraft container loading. **Aerospace and Electronic Systems, IEEE Transactions on**, IEEE, v. 39, n. 1, p. 140–150, 2003. Citado nas páginas 17, 20 e 39.
- MUNDIM, L. R. **Uma abordagem heurística para o corte de itens irregulares em múltiplos recipientes**. Tese (Doutorado) — Universidade de São Paulo, 2015. Citado na página 31.
- NIELSEN, B. K.; ODGAARD, A. Fast neighborhood search for the nesting problem. **Sci. Report, University of Copenhagen**, Citeseer, 2003. Citado na página 30.
- OLIVEIRA, J. F.; GOMES, A. M.; FERREIRA, J. S. Topos—a new constructive algorithm for nesting problems. **OR-Spektrum**, Springer, v. 22, n. 2, p. 263–284, 2000. Citado nas páginas 31, 36, 48 e 49.
- OLIVEIRA, J. F. C.; FERREIRA, J. A. S. Algorithms for nesting problems. In: **Applied simulated annealing**. [S.l.]: Springer, 1993. p. 255–273. Citado nas páginas 27 e 28.
- PAQUAY, C.; SCHYNS, M.; LIMBOURG, S. A mixed integer programming formulation for the three-dimensional bin packing problem deriving from an air cargo application. **International Transactions in Operational Research**, Wiley Online Library, v. 23, n. 1-2, p. 187–213, 2016. Citado na página 39.
- QUEIROZ, T. A. de; MIYAZAWA, F. K. Two-dimensional strip packing problem with load balancing, load bearing and multi-drop constraints. **International Journal of Production Economics**, Elsevier, v. 145, n. 2, p. 511–530, 2013. Citado nas páginas 37 e 38.
- RIBEIRO, C.; CARRAVILLA, M. A.; OLIVEIRA, J. F. Applying constraint logic programming to the resolution of nesting problems. **Pesquisa Operacional**, v. 19, n. 2, p. 239–247, 1999. Citado na página 34.
- RODRIGUES, M. O.; TOLEDO, F. M. A clique covering mip model for the irregular strip packing problem. **Computers & Operations Research**, Elsevier, 2016. Citado na página 35.
- SCHEITHAUER, G.; TERNO, J. Modeling of packing problems. **Optimization**, Taylor & Francis, v. 28, n. 1, p. 63–84, 1993. Citado na página 36.

- SEGENREICH, S. A.; BRAGA, L. M. P. F. Optimal nesting of general plane figures: A monte carlo heuristical approach. **Computers & graphics**, Elsevier, v. 10, n. 3, p. 229–237, 1986. Citado nas páginas 27 e 31.
- STOYAN, Y.; TERNO, J.; SCHEITHAUER, G.; GIL, N.; ROMANOVA, T. ϕ -functions for primary 2D-objects. **Studia Informatica Universalis**, v. 2, p. 1–32, 2001. Citado na página 23.
- STOYAN, Y. G.; NOVOZHILOVA, M.; KARTASHOV, A. Mathematical model and method of searching for a local extremum for the non-convex oriented polygons allocation problem. **European Journal of Operational Research**, Elsevier, v. 92, n. 1, p. 193–210, 1996. Citado na página 22.
- TERASHIMA-MARÍN, H.; ROSS, P.; FARÍAS-ZÁRATE, C.; LÓPEZ-CAMACHO, E.; VALENZUELA-RENDÓN, M. Generalized hyper-heuristics for solving 2d regular and irregular packing problems. **Annals of Operations Research**, Springer, v. 179, n. 1, p. 369–392, 2010. Citado nas páginas 35 e 36.
- TOLEDO, F. M.; CARRAVILLA, M. A.; RIBEIRO, C.; OLIVEIRA, J. F.; GOMES, A. M. The dotted-board model: a new mip model for nesting irregular shapes. **International Journal of Production Economics**, Elsevier, v. 145, n. 2, p. 478–487, 2013. Citado na página 35.
- TRIVELLA, A.; PISINGER, D. The load-balanced multi-dimensional bin-packing problem. **Computers & Operations Research**, Elsevier, v. 74, p. 152–164, 2016. Citado na página 38.
- VALLE, A. M. D.; QUEIROZ, T. A. D.; MIYAZAWA, F. K.; XAVIER, E. C. Heuristics for two-dimensional knapsack and cutting stock problems with items of irregular shape. **Expert Systems with Applications**, Elsevier, v. 39, n. 16, p. 12589–12598, 2012. Citado na página 37.
- WÄSCHER, G.; HAUSSNER, H.; SCHUMANN, H. An improved typology of cutting and packing problems. **European journal of operational research**, Elsevier, v. 183, n. 3, p. 1109–1130, 2007. Citado na página 30.
- WILSON, I. D.; ROACH, P. A. Principles of combinatorial optimization applied to container-ship stowage planning. **Journal of Heuristics**, Springer, v. 5, n. 4, p. 403–418, 1999. Citado nas páginas 17 e 20.
- WONG, W. K.; WANG, X.; MOK, P.; LEUNG, S.; KWONG, C. Solving the two-dimensional irregular objects allocation problems by using a two-stage packing approach. **Expert Systems with Applications**, Elsevier, v. 36, n. 2, p. 3489–3496, 2009. Citado nas páginas 32 e 33.

DESCRIÇÃO DAS INSTÂNCIAS

A seguir, apresentamos as informações das instâncias utilizadas em nossos experimentos computacionais. As informações sobre a forma dos itens das instâncias estão disponíveis *online* na página da *EURO Special Interest Group on Cutting and Packing - ESICUP*.¹ Para os itens não convexos, o desenho do item é dado pelo conjunto de polígonos convexos que o compõe.

A.1 Instância Albano

Tabela 6 – Dados gerais da instância *Albano*.

Nome	Forma do recipiente	Quantidade de itens	Quantidade de tipos de itens
Albano	Regular	24	8

Figura 35 – Tipos de itens da instância *Albano*.

Fonte: Elaborada pelo autor.

¹ <https://paginas.fe.up.pt/~esicup/datasets>, último acesso em 28 de maio de 2017.

Tabela 7 – Dados dos itens da instância *Albano*.

ID	Cópias	Pesos			
1	2	7357939,20	11094392,70		
2	2	1116542,34	855223,92		
3	4	3349309,70	5054412,82	5511136,87	5998309,19
4	4	172880,64	332254,98	202594,50	202594,50
5	4	495712,00	318672,00	570954,00	433748,00
6	4	437945,04	801871,20	1067105,52	863553,60
7	2	4578185,79	4223537,59		
8	2	2027016,00	3969573,00		

A.2 Instância Dagli

Tabela 8 – Dados gerais da instância *Dagli*.

Nome	Forma do recipiente	Quantidade de itens	Quantidade de tipos de itens
Dagli	Regular	30	3

Tabela 9 – Dados dos itens da instância *Dagli*.

ID	Cópias	Pesos			
1	3	89,49	38,19	105,45	
2	3	63,00	44,25	144,00	
3	3	331,20	192,00	456,00	
4	3	409,50	386,40	252,00	
5	3	38,50	45,50	66,50	
6	3	305,50	350,15	420,65	
7	3	53,79	146,70	79,87	
8	3	32,68	34,96	30,97	
9	3	26,01	11,73	17,85	
10	3	72,13	77,06	59,58	

Figura 36 – Tipos de itens da instância *Dagli*.

Fonte: Elaborada pelo autor.

A.3 Instância *Fu*

Tabela 10 – Dados gerais da instância *Fu*.

Nome	Forma do recipiente	Quantidade de itens	Quantidade de tipos de itens
Shapes0	Regular	12	11

Figura 37 – Tipos de itens da instância *Fu*.

Fonte: Elaborada pelo autor.

Tabela 11 – Dados dos itens da instância *Fu*.

ID	Cópias	Pesos
1	2	139,00 69,00
2	1	185,22
3	1	33,81
4	1	71,82
5	1	188,16
6	1	125,30
7	1	60,75
8	1	68,60
9	1	139,20
10	1	17,60
11	1	76,44

A.4 Instância Marques

Tabela 12 – Dados gerais da instância *Marques*.

Nome	Forma do recipiente	Quantidade de itens	Quantidade de tipos de itens
Marques	Regular	24	8

Tabela 13 – Dados dos itens da instância *Marques*.

ID	Cópias	Pesos
1	4	802,62 709,80 1026,48 671,58
2	4	146,52 122,10 95,46 100,64
3	2	358,80 483,60
4	2	187,20 235,56
5	4	180,40 155,10 121,00 157,30
6	4	36,80 79,20 71,20 67,20
7	2	810,81 1470,15
8	2	254,80 686,00

Figura 38 – Tipos de itens da instância *Marques*.

Fonte: Elaborada pelo autor.

A.5 Instância *Shapes0*

Tabela 14 – Dados gerais da instância *Shapes0*.

Nome	Forma do recipiente	Quantidade de itens	Quantidade de tipos de itens
Shapes0	Regular	43	4

Tabela 15 – Dados dos itens da instância *Shapes0*.

ID	Cópias	Pesos														
1	15	46,00	78,00	58,00	74,80	72,80	60,00	34,00	70,00	58,00	31,20	20,80	72,80	65,20	76,00	24,40
2	7	89,28	114,48	91,44	95,04	139,68	143,28	67,68								
3	9	44,52	29,96	17,08	41,44	44,24	36,40	14,84	45,08	28,56						
4	12	23,80	31,20	19,80	21,20	27,60	39,80	38,80	22,60	29,00	14,40	33,60	35,80			

Figura 39 – Tipos de itens da instância *Shapes0*.

Fonte: Elaborada pelo autor.

A.6 Instância Shapes2

Tabela 16 – Dados gerais da instância *Shapes2*.

Nome	Forma do recipiente	Quantidade de itens	Quantidade de tipos de itens
Shapes2	Regular	28	7

Tabela 17 – Dados dos itens - instância *Shapes2*

ID	Cópias	Pesos				
1	4	25,44	13,76	13,28	23,84	
2	4	13,18	9,30	24,02	14,57	
3	4	7,42	18,90	27,44	10,50	
4	4	14,41	12,10	6,05	15,29	
5	4	20,16	12,76	8,26	19,43	
6	4	5,10	7,26	6,30	8,28	
7	4	3,68	7,08	2,68	7,24	

Figura 40 – Tipo de itens da instância *Shapes2*.

Fonte: Elaborada pelo autor.

A.7 Instância Shirts

Tabela 18 – Dados gerais da instância *Shirts*.

Nome	Forma do recipiente	Quantidade de itens	Quantidade de tipos de itens
Shirts	Regular	99	8

Tabela 19 – Dados dos itens da instância *Shirts*.

ID	Cópias	Pesos															
1	8	47,17	64,97	43,16	79,66	22,25	30,71	70,76	41,38								
2	8	125,78	56,12	114,16	76,75	73,53	86,43	90,30	89,01								
3	8	138,46	116,96	46,44	138,46	95,46	82,56	106,64	86,86								
4	15	14,72	17,48	12,65	9,32	19,90	6,44	10,35	9,08	6,21	21,74	6,78	6,32	6,78	8,05	11,27	
5	15	7,42	11,13	8,89	12,25	5,11	4,27	8,12	4,48	5,25	3,64	8,40	13,02	11,55	4,76	11,34	
6	15	2,64	5,84	4,60	7,12	3,20	5,60	3,48	6,88	6,76	3,64	6,48	3,20	5,92	2,92	4,00	
7	15	2,94	3,87	1,80	5,28	3,15	4,05	2,67	2,19	4,47	3,42	3,81	3,63	3,03	2,79	4,20	
8	15	9,14	23,34	12,47	26,10	27,84	24,36	17,40	11,46	20,44	20,59	24,94	14,94	24,94	17,54	18,27	

Figura 41 – Tipos de itens da instância *Shirts*.

Fonte: Elaborada pelo autor.

A.8 Instância Swim

Tabela 20 – Dados gerais da instância *Swim*.

Nome	Forma do recipiente	Quantidade de itens	Quantidade de tipos de itens
Swim	Regular	48	10

Tabela 21 – Dados dos itens da instância *Swim*.

ID	Cópias	Pesos					
1	3	1872583,61	1493326,17	628145,14			
2	6	419426,60	558207,46	262141,62	166537,03	434846,70	234385,45
3	6	1142768,18	883048,14	1135347,61	601066,38	1365385,36	1165029,90
4	6	183600,26	231916,12	275400,39	335795,22	463832,24	253658,26
5	6	719221,43	342687,86	698067,86	283457,86	274996,43	757297,86
6	3	824719,20	1417486,13	730220,13			
7	3	1069113,92	1174116,18	1536851,26			
8	6	369278,13	356433,67	455978,21	166977,94	539467,18	436711,52
9	6	198582,13	263765,58	162200,67	206161,60	83374,18	137946,36
10	3	1605476,46	1472609,45	1516898,45			

Figura 42 – Tipos de itens da instância *Swim*.

Fonte: Elaborada pelo autor.

A.9 Instância Trousers

Tabela 22 – Dados gerais da instância *Trousers*.

Nome	Forma do recipiente	Quantidade de itens	Quantidade de tipos de itens
Trousers	Regular	64	17

Tabela 23 – Dados dos itens da instância *Trousers*.

ID	Cópias	Pesos							
1	8	388,53	1316,68	1122,42	496,46	964,13	1115,22	1223,15	1403,02
2	8	900,24	532,40	1219,68	1035,76	1442,32	813,12	803,44	755,04
3	1	367,65							
4	1	267,80							
5	1	308,00							
6	1	386,40							
7	8	31,92	78,96	40,32	24,78	59,64	57,54	44,94	26,88
8	8	63,54	117,24	96,28	100,22	106,11	101,52	48,47	97,60
9	1	57,75							
10	1	135,80							
11	1	70,56							
12	1	99,45							
13	2	46,41	31,85						
14	2	36,60	61,80						
15	8	102,90	143,08	179,34	84,28	195,02	121,52	120,54	124,46
16	8	104,96	45,44	120,96	67,84	101,76	94,72	110,08	69,12
17	4	40,96	54,72	19,52	45,44				

Figura 43 – Tipos de itens da instância *Trousers*.

Fonte: Elaborada pelo autor.

A.10 Instância Car

A instância *Car* é uma instância com recipiente irregular composta pelos itens da instância *Shapes0*. Para informações sobre os itens, ver [seção A.5](#). Na [Figura 44](#), a área útil do recipiente é dada pela região em branco. Os vértices dessa região estão descritos na [Tabela 25](#). O retângulo exterior é dado pelo vértices $(-4,4; -3,75)$, $(-4,4; 48,25)$, $(75,6;48,25)$ e $(75,6; -3,75)$.

Tabela 24 – Dados gerais da instância *Car*.

Nome	Forma do recipiente	Quantidade de itens	Quantidade de tipos de itens
Car	Irregular	43	4

Tabela 25 – Vértices do recipiente da instância *Car*.

Quantidade de vértices: 20																				
x	0,0	8,0	10,5	15,5	18,0	52,8	55,3	60,3	62,8	70,8	70,8	62,8	60,3	55,3	52,8	18,0	15,5	10,5	8,0	0,0
y	0,0	0,0	5,0	5,0	0,0	0,0	5,5	5,5	0,0	0,0	44,5	44,5	39,5	39,5	44,5	44,5	39,5	39,5	44,5	44,5

Figura 44 – Forma do recipiente da instância *car*.

Fonte: Elaborada pelo autor.

A.11 Instância Car-dagli

A instância *Car-dagli* é baseada na instância *Dagli*, diferente apenas no formato do recipiente. Para informações sobre os itens, ver [seção A.2](#). Como na instância anterior, a área útil do recipiente é dado pela região em branco ([Figura 45](#)). O retângulo exterior é dado pelo vértices (-5,4; -4,6), (-5,4; 59,10), (92,6;59,10) e (92,6; -4,6).

Tabela 26 – Dados gerais da instância *Car-dagli*.

Nome	Forma do recipiente	Quantidade de itens	Quantidade de tipos de itens
Car-dagli	Irregular	30	3

Tabela 27 – Vértices do recipiente - instância *Car*

Quantidade de vértices: 20																				
x	0,0	9,80	12,85	18,95	22,00	64,70	67,75	73,85	76,90	86,70	86,70	76,90	73,85	67,75	64,70	22,00	18,95	12,85	9,80	0,00
y	0,0	0,0	6,10	6,10	0,00	0,00	6,10	6,10	0,00	0,00	54,50	54,50	48,40	48,40	54,50	54,50	48,40	48,40	54,50	54,50

Figura 45 – Forma do recipiente da instância *Car-dagli*.

Fonte: Elaborada pelo autor.

A.12 Instância Car-fu

A instância *Car-fu* é baseada na instância *Fu*, diferente apenas no formato do recipiente. Para informações sobre os itens, ver [seção A.3](#). O retângulo exterior é dado pelo vértices (-3,2; -2,65), (-3,2; 34,35), (53,8; 34,35) e (53,8; -2,65).

Tabela 28 – Dados gerais da instância *Car-fu*.

Nome	Forma do recipiente	Quantidade de itens	Quantidade de tipos de itens
Fu	Irregular	12	11

Tabela 29 – Vértices do recipiente da instância *Car-fu*.

Quantidade de vértices: 20																				
x	0,00	5,70	7,45	11,05	12,80	37,60	39,35	42,95	44,70	50,40	50,40	44,70	42,95	39,35	37,60	12,80	11,05	7,45	5,70	0,00
y	0,0	0,0	3,50	3,50	0,00	0,00	3,50	3,50	0,00	0,00	31,70	31,70	28,20	28,20	31,70	31,70	28,20	28,20	31,70	31,70

Figura 46 – Forma do recipiente da instância *Car-fu*.

Fonte: Elaborada pelo autor.

SOLUÇÕES OBTIDAS

Neste capítulo são apresentadas as soluções encontradas pelas heurísticas SDESLOC, ADESLOC, RDESLOC, SDESLOC-IRREGULAR, ADESLOC-IRREGULAR e RDESLOC-IRREGULAR.

Nas figuras das soluções, os itens estão em escala de cinza. A intensidade da cor é proporcional ao peso do item, de forma que quanto maior o peso de um item, mais escuro ele é. O centro de gravidade final do recipiente está representado pelo círculo em branco.

B.1 Heurística SDesloc

Albano - Pequeno

Figura 47 – Solução para instância *Albano*.

Fonte: Elaborada pelo autor.

Dagli - PequenoFigura 48 – Solução para instância *Dagli*.

Fonte: Elaborada pelo autor.

Fu - PequenoFigura 49 – Solução para instância *Fu*.

Fonte: Elaborada pelo autor.

Marques - PequenoFigura 50 – Solução para instância *Marques*.

Fonte: Elaborada pelo autor.

Shapes0 - PequenoFigura 51 – Solução para instância *Shapes0*.

Fonte: Elaborada pelo autor.

Shapes2 - PequenoFigura 52 – Solução para instância *Shapes2*.

(a) Recipiente 1.

(b) Recipiente 2.

Fonte: Elaborada pelo autor.

Shirts - PequenoFigura 53 – Solução para instância *Shirts*.

(a) Recipiente 1.

(b) Recipiente 2.

Fonte: Elaborada pelo autor.

Swim - PequenoFigura 54 – Solução para instância *Swim*.

(a) Recipiente 1.

(b) Recipiente 2.

Fonte: Elaborada pelo autor.

Trousers - PequenoFigura 55 – Solução para instância *Trousers*.

Fonte: Elaborada pelo autor.

Albano - GrandeFigura 56 – Solução para instância *Albano* grande.

Fonte: Elaborada pelo autor.

Dagli - GrandeFigura 57 – Solução para instância *Dagli* grande.

Fonte: Elaborada pelo autor.

Fu - GrandeFigura 58 – Solução para instância *Fu* grande.

Fonte: Elaborada pelo autor.

Marques - GrandeFigura 59 – Solução para instância *Marques* grande.

Fonte: Elaborada pelo autor.

Shapes0 - GrandeFigura 60 – Solução para instância *Shapes0* grande.

Fonte: Elaborada pelo autor.

Shapes2 - GrandeFigura 61 – Solução para instância *Shapes2* grande.

Fonte: Elaborada pelo autor.

Shirts - GrandeFigura 62 – Solução para instância *Shirts* grande.

Fonte: Elaborada pelo autor.

Swim - GrandeFigura 63 – Solução para instância *Swim* grande.

Fonte: Elaborada pelo autor.

Trousers - GrandeFigura 64 – Solução para instância *Trousers* grande.

Fonte: Elaborada pelo autor.

B.2 Heurística ADesloc**Albano - Pequeno**Figura 65 – Solução para instância *Albano*.

(a) Recipiente 1.

(b) Recipiente 2.

Fonte: Elaborada pelo autor.

Dagli - PequenoFigura 66 – Solução para instância *Dagli*.

(a) Recipiente 1.

(b) Recipiente 2.

Fonte: Elaborada pelo autor.

Fu - PequenoFigura 67 – Solução para instância *Fu*.

Fonte: Elaborada pelo autor.

Marques - PequenoFigura 68 – Solução para instância *Marques*.

Fonte: Elaborada pelo autor.

Shapes0 - PequenoFigura 69 – Solução para instância *Shapes0*.

Fonte: Elaborada pelo autor.

Shapes2 - PequenoFigura 70 – Solução para instância *Shapes2*.

Fonte: Elaborada pelo autor.

Shirts - PequenoFigura 71 – Solução para instância *Shirts*.

Fonte: Elaborada pelo autor.

Swim - PequenoFigura 72 – Solução para instância *Swim*.

Fonte: Elaborada pelo autor.

Trousers - PequenoFigura 73 – Solução para instância *Trousers*.

Fonte: Elaborada pelo autor.

Albano - GrandeFigura 74 – Solução para instância *Albano* - grande.

Fonte: Elaborada pelo autor.

Dagli - GrandeFigura 75 – Solução para instância *Dagli* - grande.

Fonte: Elaborada pelo autor.

Fu - GrandeFigura 76 – Solução para instância *Fu* - grande.

Fonte: Elaborada pelo autor.

Marques - GrandeFigura 77 – Solução para instância *Marques* - grande.

Fonte: Elaborada pelo autor.

Shapes0 - GrandeFigura 78 – Solução para instância *Shapes0* - grande.

Fonte: Elaborada pelo autor.

Shapes2 - GrandeFigura 79 – Solução para instância *Shapes2*.

Fonte: Elaborada pelo autor.

Shirts - GrandeFigura 80 – Solução para instância *Shirts*.

Fonte: Elaborada pelo autor.

Swim - GrandeFigura 81 – Solução para instância *Swim*.

Fonte: Elaborada pelo autor.

Trousers - GrandeFigura 82 – Solução para instância *Trousers*.

Fonte: Elaborada pelo autor.

B.3 Heurística RDesloc

Albano - Pequeno

Figura 83 – Solução para instância *Albano*.

Fonte: Elaborada pelo autor.

Dagli - Pequeno

Figura 84 – Solução para instância *Dagli*.

Fonte: Elaborada pelo autor.

Fu - PequenoFigura 85 – Solução para instância *Fu*.

Fonte: Elaborada pelo autor.

Marques - PequenoFigura 86 – Solução para instância *Marques*.

Fonte: Elaborada pelo autor.

Shapes0 - PequenoFigura 87 – Solução para instância *Shapes0*.

Fonte: Elaborada pelo autor.

Shapes2 - PequenoFigura 88 – Solução para instância *Shapes2*.

Fonte: Elaborada pelo autor.

Shirts - PequenoFigura 89 – Solução para instância *Shirts*.

Fonte: Elaborada pelo autor.

Swim - PequenoFigura 90 – Solução para instância *Swim*.

(a) Recipiente 1.

(b) Recipiente 2.

Fonte: Elaborada pelo autor.

Trousers - PequenoFigura 91 – Solução para instância *Trousers*.

(a) Recipiente 1.

(b) Recipiente 2.

Fonte: Elaborada pelo autor.

Albano - GrandeFigura 92 – Solução para instância *Albano* grande.

Fonte: Elaborada pelo autor.

Dagli - GrandeFigura 93 – Soluções para instância *Dagli*.

Fonte: Elaborada pelo autor.

Fu - Grande

Figura 94 – Solução para instância Fu.

Fonte: Elaborada pelo autor.

Marques - GrandeFigura 95 – Solução para instância *Marques* grande.

Fonte: Elaborada pelo autor.

Shapes0 - GrandeFigura 96 – Solução para instância *Shapes0* grande.

Fonte: Elaborada pelo autor.

Shapes2 - Grande

Figura 97 – Solução para instância *Shapes2* grande.

Fonte: Elaborada pelo autor.

Shirts - Grande

Figura 98 – Solução para instância *Shirts* grande.

Fonte: Elaborada pelo autor.

Swim - GrandeFigura 99 – Solução para instância *Swim* grande.

Fonte: Elaborada pelo autor.

Trousers - GrandeFigura 100 – Solução para instância *Trousers* grande.

Fonte: Elaborada pelo autor.

B.4 Heurística SDesloc-Irregular

Car

Figura 101 – Solução para instância *Car*.

Fonte: Elaborada pelo autor.

Car-Dagli

Figura 102 – Solução para instância *Car-dagli*.

(a) Recipiente 1.

(b) Recipiente 2.

Fonte: Elaborada pelo autor.

Car-fu

Figura 103 – Solução para instância *Car-fu*.

(a) Recipiente 1.

(b) Recipiente 2.

Fonte: Elaborada pelo autor.

B.5 Heurística ADesloc-Irregular

Car

Figura 104 – Solução para instância *Car*.

Fonte: Elaborada pelo autor.

Car-Dagli

Figura 105 – Solução para instância *Car-dagli*.

(a) Recipiente 1.

(b) Recipiente 2.

Fonte: Elaborada pelo autor.

Car-fu

Figura 106 – Solução para instância *Car-fu*.

(a) Recipiente 1.

(b) Recipiente 2.

Fonte: Elaborada pelo autor.

B.6 Heurística RDesloc-Irregular

Car

Figura 107 – Solução para instância *Car*.

Fonte: Elaborada pelo autor.

Car-Dagli

Figura 108 – Solução para instância *Car-dagli*.

Fonte: Elaborada pelo autor.

Car-fu

Figura 109 – Solução para instância *Car-fu*.

Fonte: Elaborada pelo autor.