



PROBLEMA DE CORTE DE ITENS IRREGULARES NA FABRICAÇÃO DE LUVAS DE COURO

Leandro Resende Mundim Marina Andretta

Instituto de Ciências Matemáticas e de Computação

Universidade de São Paulo (USP)

mundim@icmc.usp.br, andretta@icmc.usp.br

Resumo

Problemas de corte e empacotamento de itens irregulares são problemas que visam determinar um leiaute ótimo de objetos menores (itens) dentro de objetos maiores (recipientes), a fim de atender a uma demanda. Dentre estes problemas, existem os problemas de empacotamento de itens irregulares em recipientes, para os quais os recipientes têm tamanho fixo e o objetivo é encontrar um leiaute dos itens, sem sobreposição, dentro dos recipientes utilizando a menor quantidade de recipientes. Neste trabalho, resolvemos um problema de uma indústria de couro, no qual os itens e os recipientes são irregulares. Apresentamos duas heurísticas com uma abordagem alternativa para lidar com os defeitos do couro. Os experimentos computacionais validam as duas heurísticas que podem ser utilizadas em várias indústrias, como a têxtil, a de móveis e a de calçados.

PALAVRAS CHAVE: Problemas de Corte de Itens em Recipientes Irregulares, Heurísticas, Indústria de Couro.

Área principal: Otimização Combinatória.

Abstract

Nesting problems are problems that aim to determine the optimum layout of smaller objects (items) within larger objects (containers), in order to meet a demand. Among these problems are the irregular bin packing problems, for which containers are fixed in size and the goal is to find a layout of items without overlap within the containers using the least amount of containers. In this paper, we solve a problem of a leather factory, where items and containers are irregular. We present two heuristics and an alternative approach to dealing with the defects of leather. The computational experiments validate the two heuristics that can be used in various industries, such as textiles, furniture and footwear.

KEYWORDS: Nesting Problems, Heuristics, Leather Industry.

Main area: Combinatorial Optimization.

1 - Introdução

Cortar objetos grandes (recipientes) em produtos menores (itens) ou empacotar itens pequenos dentro de espaços delimitados são problemas amplamente estudados na literatura, denominados problemas de corte e empacotamento (ou *cutting and packing problems*). Estes problemas ocorrem com muita frequência em diversas indústrias. Para exemplificar, há o corte de tecido, couro, papel, placas de metal e madeira. Quando pensamos em empacotar itens, há o empacotamento em paletes, caminhões, salas, contêineres, etc.

Dentre os problemas de corte e empacotamento, existem os problemas de corte de itens irregulares (ou *nesting problems*). Nesta classe de problemas, os itens são representados por polígonos convexos e não convexos. Apesar de grande parte das indústrias enfrentarem o problema de corte de itens irregulares, o número de trabalhos que lidam com este tipo de problema ainda é pequeno comparado ao número de trabalhos que lidam com corte de itens regulares (retângulos ou círculos).

Um dos principais motivos para que isso aconteça é a dificuldade em trabalhar com itens irregulares. A principal delas é tratar da questão de não sobreposição dos itens. Para encontrar posições viáveis de empacotamento e verificar a sobreposição de itens, diversas técnicas foram desenvolvidas. As abordagens mais utilizadas são: o método *raster*, trigonometria direta, *phi function* e o espaço de obstrução (ou *nofit polygon*). Uma revisão sobre essas técnicas pode ser encontrada em Bennell e Oliveira (2008). A estratégia mais utilizada na literatura, e escolhida para este trabalho, é o *nofit polygon*, que se tornou a abordagem mais conhecida devido aos bons resultados da literatura e à eficiência desta técnica.

Diversas técnicas de resolução de problemas de corte de itens irregulares vêm sendo desenvolvidas. O progresso das estratégias computacionais utilizadas para resolver estes problemas é descrito por Dowland e Dowland (1995) e um levantamento mais recente pode ser encontrado em Bennell e Oliveira (2009). A maioria dos trabalhos se baseiam em heurísticas com busca local e/ou meta-heurísticas. Alguns trabalhos são: Gomes e Oliveira (2002), Cheng e Rao (2005), Egeblad et al. (2007), Bennell e Song (2010). Heurísticas eficientes e robustas podem ser encontradas em Dowland et al. (2002) e Burke et al. (2006).

Existem também trabalhos com métodos exatos: Martello et al. (2003), Alvarez-Valdes et al. (2013) e Toledo et al. (2013). Vale ressaltar que estes métodos, apesar de conseguirem bons resultados, possuem uma grande limitação com relação à quantidade de itens empacotados e conseguem resolver instâncias com poucos itens (no máximo 12 itens sem repetição). Gomes e Oliveira (2006) apresentou um método híbrido que combina um algoritmo de recozimento simulado com um modelo de programação linear e algoritmos de compactação e de separação. Este trabalho conseguiu melhorar vários resultados da literatura. Para o caso de recipientes com defeitos, assim como os recipientes que serão abordados neste trabalho, existem os trabalhos de Crispin et al. (2005), Yuping et al. (2005), Alves et al. (2012) e Baldacci et al. (2014).

Neste trabalho, estudamos o problema de empacotamento de itens irregulares em recipientes também irregulares (ou *irregular bin packing problem*). Este problema consiste em reduzir a quantidade de recipientes utilizados para cortar uma demanda de itens. O estudo de caso apresentado neste trabalho é um problema da indústria de couro, em que precisamos minimizar o número de peças de couro utilizadas para cortar uma demanda específica de luvas. Este problema é interessante tanto do ponto de vista econômico como do ponto de vista ambiental, já que uma boa estratégia de corte pode reduzir a quantidade de couro utilizada.

Visando melhorar a qualidade dos planos de corte da indústria de couro, apresentamos duas heurísticas eficientes para a resolução do problema estudado neste trabalho. Uma heurística é a mais utilizada no empacotamento de itens irregulares, a *Bottom Left*.

A segunda heurística propõe dividir o recipiente em duas regiões e é baseada na heurística *Bottom Left*. As duas heurísticas utilizam uma abordagem alternativa para lidar com os defeitos dos recipientes. Esta nova abordagem é bastante simples e pode ser facilmente adaptada para outras variantes do problema de corte de itens irregulares.

O trabalho está organizado da seguinte forma. Na Seção 2, apresentamos o problema estudado neste trabalho. Na Seção 3, apresentamos as heurísticas desenvolvidas. A Seção 4 contém os resultados numéricos obtidos. As conclusões são apresentadas na Seção 5.

2 - O Problema

Problemas de corte e empacotamento podem ser classificados segundo seus objetivos e as características de seus objetos (itens e recipientes). Para mais detalhes, veja a classificação proposta por Dyckhoff (1990), que foi revisada e estendida por Wascher et al. (2007). O problema estudado neste trabalho, pela classificação de Wascher et al. (2007), trata-se da versão bidimensional do problema de empacotamento de itens irregulares em recipientes (ou *irregular bin packing problem*). O objetivo deste problema é, dados uma lista de m recipientes irregulares e uma lista de n itens irregulares, determinar como empacotar todos os n itens utilizando o menor número de recipientes possível, respeitando as configurações dos recipientes e a não sobreposição dos itens.

O problema estudado neste trabalho é um problema real da indústria de luvas, que necessita minimizar o número de peças de couro utilizadas na fabricação de diferentes tipos de luvas. Nesta variante do problema, os recipientes são peças de couro e os itens são as partes de uma luva. Cada peça de couro possui um formato único e vários defeitos, com diferenças entre a frente e o verso, o que proíbe cortar itens dos dois lados do recipiente. No corte do couro, a rotação dos itens é livre. Apesar disso, neste trabalho, para facilitar a análise da solução, consideramos as rotações de 0, 90 e 180 graus, as mais utilizadas nos problemas de corte de itens irregulares.

Definido as características dos itens e dos recipientes, nosso objetivo é cortar uma demanda de luvas utilizando o menor número de recipientes possível. Note que a demanda de luvas está diretamente relacionada com a demanda dos itens. A seguir, na Seção 2.1, apresentamos como os objetos (itens e recipientes) são representados. A Seção 2.2 descreve o procedimento para alocar um item no recipiente.

2.1 - Representação dos objetos

Os objetos podem ser representados de diferentes maneiras, as mais comuns são a representação por malha ou por polígonos. As abordagens com malhas são mais flexíveis e amplamente utilizadas em métodos heurísticos. Estas abordagens dividem o recipiente em áreas discretas, reduzindo a complexidade geométrica a uma matriz. Reduzir os itens a uma matriz facilita a verificação de sobreposição e a torna muito mais eficiente. Porém, a abordagem discreta pode não ser uma boa aproximação dos itens, dado que um item pode ser muito pequeno ou muito grande quando comparado à malha. A qualidade da aproximação depende da discretização da malha, ou seja, quanto menor for a discretização, melhor a aproximação de um item real e maior o custo computacional. Para maiores detalhes sobre as representações existentes, veja Bennell e Oliveira (2008).

A abordagem mais utilizada em problemas de corte e empacotamento de itens irregulares é a representação poligonal dos itens, ou seja, os itens são aproximados por polígonos. Neste trabalho, utilizamos um conjunto de polígonos convexos para representar cada item. Como trabalhamos com vários itens iguais, representamos apenas os itens diferentes (que chamamos de tipos de itens) e armazenamos a demanda de cada tipo de item.

Cada recipiente é representado por um retângulo, que cobre toda a peça do couro, e um conjunto de defeitos. A abordagem proposta neste trabalho para lidar com defeitos é aproximar a região defeituosa e o contorno do recipiente por um conjunto de polígonos convexo que possuem uma posição fixa no recipiente. Essa nova abordagem facilita encontrar posições viáveis dentro do recipiente, o que não é trivial quando estamos trabalhando com itens irregulares e recipientes também irregulares.

2.2 - Verificação de não-sobreposição de itens

Uma das maiores dificuldades em problemas de corte e empacotamento de itens irregulares é lidar com a geometria dos itens, principalmente evitar a sobreposição de itens irregulares e encontrar posições factíveis de empacotamento no interior do recipiente. Neste trabalho, utilizamos o espaço de obstrução (ou *noft polygon*).

A ideia do espaço de obstrução é reduzir a complexidade do cálculo de sobreposição durante o empacotamento dos itens. Trocamos a verificação de sobreposição entre dois itens por verificar a posição de um ponto de referência em relação a um conjunto de polígonos. Estes polígonos são construídos fixando um dos itens, enquanto o outro orbita ao redor do item fixo, de maneira que os itens sempre se encostem e nunca se sobreponham. Assim, podemos verificar a sobreposição entre dois itens verificando a distância entre o ponto de referência do item orbital e os segmentos de reta que ligam cada par de vértices consecutivos dos polígonos que formam o espaço de obstrução. Se a distância do ponto de referência para um dos segmentos de reta for zero, os itens estão encostados; se a distância for maior do que zero para todos os pares, o ponto de referência está no interior de um polígono e os itens estão se sobrepondo; e se a distância for negativa para algum par de vértices, o ponto de referência está de fora dos polígonos e os itens estão separados.

Utilizamos o algoritmo de Cuninghame-Green (1989) para calcular o espaço de obstrução, que pode ser calculado de forma rápida e eficiente para polígonos convexos. Em linhas gerais, o algoritmo de Cuninghame-Green (1989) constrói o espaço de obstrução entre um par de polígonos convexos, um denominado A , fixo no plano e com orientação anti-horária, e o outro chamado de B , com orientação horária e com um ponto de referência dado pelo vértice de maior coordenada no eixo das ordenadas (eixo y) - em caso de empate, o ponto de referência é dado pelo vértice dentre estes que tem maior coordenada no eixo das abscissas (eixo x). A região do espaço de obstrução é construída trasladando todas as arestas de A e B para o vértice de A com a menor coordenada no eixo y (em caso de empate, dentre estes, o de menor coordenada no eixo x). Em seguida, basta concatenar todas as arestas dos polígonos no sentido anti-horário. A Figura 1 ilustra a obtenção do espaço de obstrução dos polígonos A e B .

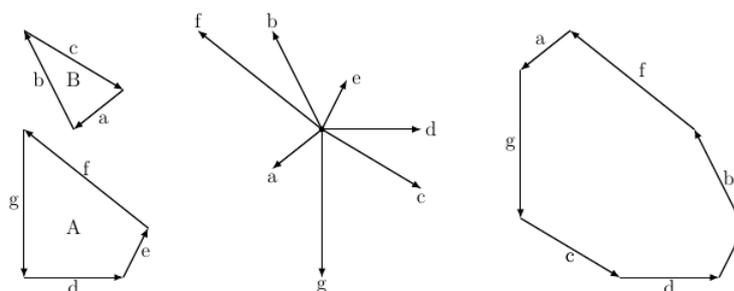


Figura 1: Espaço de obstrução entre os polígonos A (fixo) e B (orbital).

Note que o espaço de obstrução entre os pares de itens necessita ser calculado

apenas uma vez, em uma etapa de pré-processamento.

Lidar com os defeitos dos recipientes é uma grande dificuldade abordada neste trabalho. Em geral, trabalhos com recipientes irregulares dividem os defeitos em regiões proibidas e regiões de baixa qualidade. Para evitar a sobreposição durante o empacotamento, cada item é deslocado no interior do recipiente, criando a região onde o item pode ser empacotado. Essa técnica é conhecida como *innerfit polygon* e, quando estamos trabalhando com itens irregulares, essa técnica pode ser muito custosa. Para evitar o cálculo desta região, aproximamos os defeitos por polígonos convexos e os consideramos como itens empacotados nos recipientes que não podem ser movidos. Esta nova técnica de lidar com os defeitos é um diferencial deste trabalho.

Podemos definir a maneira de empacotar um item no recipiente da seguinte maneira: como um item é composto por um conjunto de polígonos convexos que não podem se separar, quando empacotamos um item, devemos procurar uma posição onde todos os polígonos convexos que o compõem possam ser empacotados. Os polígonos podem ser empacotados quando nenhum dos polígonos convexos que estão sendo empacotados sobrepõem um polígono convexo que já está no recipiente. Os polígonos que estão no recipiente, quando vamos empacotar um item, são todos os defeitos do recipiente e todos os polígonos que compõem os itens já empacotados. Além disso, o item deve ser empacotado dentro do retângulo que cobre todo o recipiente.

3 - Heurísticas propostas

Esta seção apresenta os métodos de resolução desenvolvidos para resolver o problema de empacotamento de itens irregulares em recipientes apresentados na Seção 2. Neste trabalho, os métodos de resolução utilizados são estratégias heurísticas, ou seja, estratégias que objetivam obter uma solução razoável em um tempo computacional pequeno, neste caso, um tempo inferior a dois minutos. A escolha de métodos heurísticos e rápidos se deu pelas características do problema estudado neste trabalho. Como cada peça de couro possui um formato diferente e não é possível cortar muitos itens de cada um, o método de resolução precisa ser eficiente.

As heurísticas propostas são algoritmos construtivos, baseados predominantemente na ordenação dos itens. A ideia envolve criar uma solução em pouco tempo computacional, através de um procedimento guloso, que utiliza os conceitos do espaço de obstrução (Seção 2.2) para encontrar empacotamentos viáveis. Os algoritmos propostos recebem uma lista de itens e de recipientes. A lista de itens recebe, para cada um dos n tipos de itens, sua demanda e suas possíveis rotações. A lista dos m recipientes recebe, para cada recipiente, as dimensões do retângulo que o envolve e sua lista de defeitos.

A Seção 3.1 apresenta a heurística *Bottom Left*. Na Seção 3.2 apresentamos a heurística de Duas Regiões.

3.1 - Heurística *Bottom Left*

O algoritmo *Bottom Left* é um dos mais utilizados para a resolução de problemas de corte de itens irregulares e existe uma grande variedade de estratégias desenvolvidas. Na heurística desenvolvida neste trabalho, o recipiente é discretizado em uma malha de pontos, no eixo x e no eixo y , utilizando, respectivamente, os valores da largura e da altura do recipiente. A discretização considera apenas os pontos com coordenadas inteiras do recipiente. Definida a malha, verificamos, para cada ponto, se o item pode ser empacotado ou não. Começamos a verificação do ponto inferior esquerdo (origem da malha, ponto $(0, 0)$), avançando uma unidade no eixo y até alcançar o ponto superior (altura da malha), recomen-

çando a verificação na parte inferior (altura zero), uma unidade à direita. Esse processo é repetido até que se encontre uma posição para alocação do item ou até ser atingido o ponto superior direito (o ponto de maior coordenada no eixo x e no eixo y). Se nenhum ponto de alocação for encontrado, repete-se o procedimento para o próximo recipiente. O Algoritmo 1 apresenta este procedimento, que chamamos de heurística *Bottom Left*.

Algoritmo 1 Heurística *Bottom Left*

Entrada: *Itens*, n , *Recipientes*, m .

Itens é a lista com os tipos de itens.

n é a quantidade de tipos de itens.

Recipientes é a lista com os recipientes.

m é a quantidade de recipientes.

Saida: S

S é a solução obtida, formada pelas posições e rotações dos itens empacotados em cada recipiente.

```
1:  $S \leftarrow$  vazio;
2: Itens é ordenada, de forma decrescente, pela área de cada item;
3: for ( $i = 0$ ;  $i < n$ ;  $i++$ ) do
4:    $item_i \leftarrow$   $i$ -ésimo elemento da lista Itens;
5:   for ( $j = 0$ ;  $j < m$ ;  $j++$ ) do
6:      $recipiente_j \leftarrow$   $j$ -ésimo elemento da lista Recipientes;
7:     if (demanda do  $item_i > 0$ ) then
8:       Busque o ponto do  $recipiente_j$ , com coordenadas inteiras, mais à esquerda e mais abaixo possível, em que o  $item_i$  pode ser empacotado em alguma de suas possíveis rotações;
9:       if ( $item_i$  pode ser empacotado no recipiente  $j$ ) then
10:        A solução  $S$  recebe o  $item_i$  no recipiente  $j$  na posição e na rotação encontrada na linha 8;
11:        A demanda do  $item_i$  é reduzida uma unidade;
12:      end if
13:    end if
14:  end for
15: end for
16: Retorne  $S$  como solução;
```

A heurística *Bottom Left* ordena de forma decrescente os itens por suas áreas, empacotando um item por vez, até satisfazer sua demanda ou testar todas as possibilidades de alocação em todos os recipientes. Na linha 8, o Algoritmo 1 procura, nas posições da discretização, o ponto mais à esquerda e abaixo em que o item pode ser empacotado, respeitando o espaço de obstrução dos itens empacotados contra este item e dos defeitos contra este item. Além disso, o item deve estar dentro do retângulo que limita o recipiente. Se tal ponto existir, a solução recebe as coordenadas do ponto encontrado, bem como a rotação do item utilizada, e a demanda do item é atualizada. O processo é repetido até que todos os itens sejam empacotados ou não reste espaço disponível para alocar mais itens.

3.2 - Heurística de Duas Regiões

A heurística de Duas Regiões, apresentada nesta seção, é baseada na heurística *Bottom Left*. A diferença é que ela divide o recipiente em Duas Regiões, como é apresentado na Figura 2. A primeira região (A) é formada pelos cantos do recipiente e a segunda região (B) é formada pelo interior do recipiente.

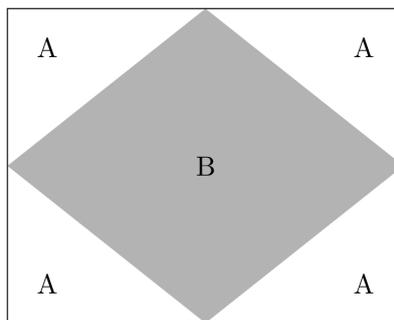


Figura 2: Regiões A e B.

Para a heurística de Duas Regiões, os itens são ordenados em ordem decrescente de suas áreas. O objetivo deste algoritmo é empacotar os itens na região A e, depois que todos os itens tentarem ser empacotados na região A, é possível empacotar os itens na região B.

Algoritmo 2 Heurística de Duas Regiões

Entrada: *Itens*, n , *Recipientes*, m .

Itens é a lista com os tipos de itens.

n é a quantidade de tipos de itens.

Recipientes é a lista com os recipientes.

m é a quantidade de recipientes.

Saída: S

S é a solução obtida, formada pelas posições e rotações dos itens empacotados em cada recipiente.

```
1:  $S \leftarrow$  vazio;
2: Itens é ordenada, de forma decrescente, pela área de cada item;
3: for ( $j = 0$ ;  $j < m$ ;  $j++$ ) do
4:   recipiente $_j \leftarrow$  j-ésimo elemento da lista de Recipientes;
5:   for ( $i = 0$ ;  $i < n$ ;  $i++$ ) do
6:     item $_i \leftarrow$  i-ésimo elemento da lista Itens;
7:     if (demanda do item $_i > 0$ ) then
8:       Procure o ponto  $(x, y)$  inferior esquerdo da região A do recipiente $_j$ , com coordenadas inteiras,
          e em que o item $_i$ , com alguma de suas possíveis rotações ( $r$ ), pode ser empacotado;
9:       if (item $_i$  pode ser empacotado em  $(x, y)$  no recipiente $_j$ ) then
10:        A solução  $S$  recebe o item $_i$  no recipiente $_j$  na posição  $(x, y)$  e na rotação  $r$ ;
11:        A demanda do item $_i$  é reduzida de uma unidade;
12:       end if
13:     end if
14:   end for
15:   for ( $i = 0$ ;  $i < n$ ;  $i++$ ) do
16:     item $_i \leftarrow$  i-ésimo elemento da lista de Itens;
17:     if (demanda do item $_i > 0$ ) then
18:       Procure o ponto  $(x, y)$  inferior esquerdo da região B do recipiente $_j$ , com coordenadas inteiras,
          e em que o item $_i$ , com alguma de suas rotações possíveis ( $r$ ) pode ser empacotado;
19:       if (item $_i$  pode ser empacotado em  $(x, y)$  no recipiente $_j$ ) then
20:        A solução  $S$  recebe o item $_i$  no recipiente $_j$  na posição  $(x, y)$  e na rotação  $r$ ;
21:        A demanda do item $_i$  é reduzida de uma unidade;
22:       end if
23:     end if
24:   end for
25: end for
26: Retorne  $S$  como solução;
```

O Algoritmo 2 descreve a heurística de Duas Regiões. Ao contrário da heurística *Bottom Left*, que empacota os itens nos recipientes até cada demanda ser atendida, a heurística de Duas Regiões tenta maximizar a ocupação do recipiente empacotando a maior

quantidade de itens ainda disponíveis, até não ser possível empacotar mais nenhum item.

Utilizamos a discretização descrita na heurística *Bottom Left* para procurar o ponto onde o ponto referência dos itens podem ser empacotado na região A (linha 8) e na região B (linha 18). A discretização é idêntica à discretização da heurística *Bottom Left*, o diferencial na busca é que as regiões A e B são separadas. Primeiramente, a busca do ponto inferior esquerdo acontece na região A (linha 8) e, depois, a busca acontece na região B (linha 18).

4 - Resultados numéricos

Nesta seção apresentamos os experimentos computacionais para as heurísticas *Bottom Left* e de Duas Regiões, implementadas em linguagem C. Todos os experimentos foram realizados em um computador Intel Core2 Duo de 2.80GHz e 4GB de memória, usando sistema operacional Linux Mint 16 Petra. Na Seção 4.1, descrevemos a instância utilizada e a Seção 4.2 apresenta os resultados numéricos.

4.1 - Instância

A instância utilizada neste trabalho para resolver o problema de empacotamento de itens irregulares em recipientes é uma instância real, obtida em uma indústria de corte de luvas de couro do interior de São Paulo. Nosso objetivo é cortar uma demanda de luvas utilizando o menor número de recipientes possível. Cada luva é formada por dez tipos de itens, apresentados Figura 3. Estes itens são todas as partes que compõem uma luva, divididas em um conjunto de polígonos convexos.

Um par de luvas é composto por oito unidades do item (6) e uma unidade de cada um dos demais itens. Observe que os itens 1-2, 3-4, 7-8 e 9-10 são os mesmos itens, refletidos sobre o eixo y . Apesar de todos os itens poderem ser livremente rotacionados no couro, para deixar o algoritmo mais eficiente, utilizamos apenas três rotações, que são as mais utilizadas em problemas de corte de itens irregulares: 0, 90 e 180 graus.

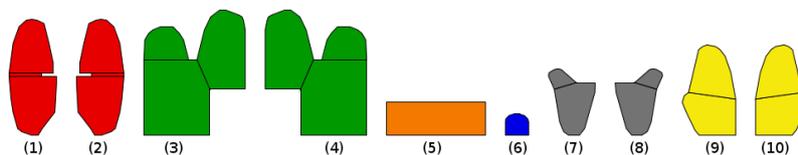


Figura 3: Representação dos itens.

Os recipientes foram criados a partir de uma amostra de peças de couro. A Figura 4 apresenta em detalhes os três recipientes utilizados em nossos experimentos computacionais. O número de defeitos nos recipientes (a), (b) e (c) são, respectivamente, iguais a 16, 26 e 19.

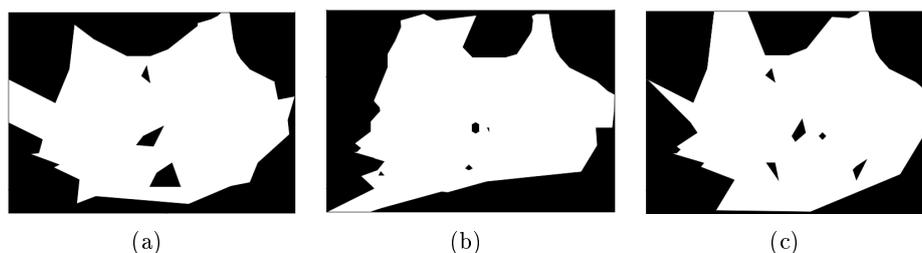


Figura 4: Peças de couro.

Os itens e os recipientes desta instância estão disponíveis na página <http://www>.

icmc.usp.br/pessoas/andretta/sbpo-2014.

4.2 - Resultados

Nesta seção apresentamos os resultados obtidos pelas heurísticas propostas neste trabalho para a resolução da instância apresentada na Seção 4.1, utilizando diferentes demandas para os pares de luvas. A Tabela 1 apresenta os resultados das duas heurísticas, *Bottom Left* e de Duas Regiões, sobre diversas demandas de pares de luvas. Na tabela, \bar{m} representa a quantidade de recipientes utilizados e t representa o tempo computacional gasto pelas heurísticas, em segundos. A melhor ocupação de um recipiente está no campo ocp^{max} .

Demanda	Quantidade de itens	<i>Bottom Left</i>			Duas Regiões		
		\bar{m}	t (s)	ocp^{max} (%)	\bar{m}	t (s)	ocp^{max} (%)
1	17	1	0.8345	18.52	1	0.3956	18.52
2	34	1	4.5501	37.03	1	1.1608	37.03
3	51	1	10.1683	55.55	1	4.8843	55.55
4	68	2	18.2553	68.82	2	16.6090	67.90
5	85	2	21.7785	72.01	2	27.6025	69.10
6	102	2	24.6639	72.58	2	33.9550	71.90
7	119	2	30.3775	71.83	3	38.8021	72.30
8	136	3	34.7555	72.47	3	41.9679	71.27
9	153	3	37.5879	73.04	3	54.7720	71.48
10	170	3	44.8973	72.01	3	61.5792	71.30
11	187	3	46.7683	71.59	-	68.3656	-

Tabela 1: Resultados numéricos.

Os experimentos mostram que a heurística *Bottom Left* se mostrou mais interessante para a resolução do problema de corte deste tipo de luva. A quantidade de recipientes utilizados e a ocupação média obtida pelas heurísticas são os mesmos para a maioria dos casos. Entretanto, para o corte de 11 pares de luvas, a heurística de Duas Regiões não foi capaz de empacotar todos os itens. O tempo de execução pela heurística de Duas Regiões foi inferior para as instâncias com demandas de 1 até 4. Para os outros casos, o tempo usado pela heurística *Bottom Left* foi menor. Com relação à ocupação máxima, ambas as heurísticas obtiveram valores próximos, com uma superioridade da heurística *Bottom Left*, que obteve resultados melhores ou iguais para dez dos onze experimentos, perdendo apenas no experimento com demanda de 7 pares de luva.

A Figura 5 apresenta o leiaute com a melhor taxa de ocupação obtida por cada uma das heurísticas. A melhor ocupação obtida pela heurística *Bottom Left* foi de 73.04%, enquanto que, para a heurística de Duas Regiões, foi de 72.3%.

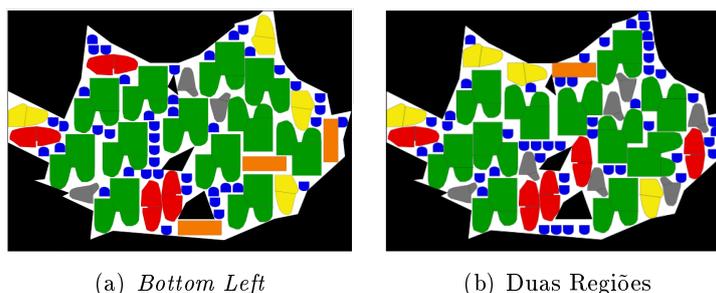


Figura 5: Recipientes com as melhores taxas de ocupação.

A Figura 6 e a Figura 7 apresentam os leiautes da heurística *Bottom Left* e da heurística de Duas Regiões para os experimentos com demanda de 10 pares de luvas.

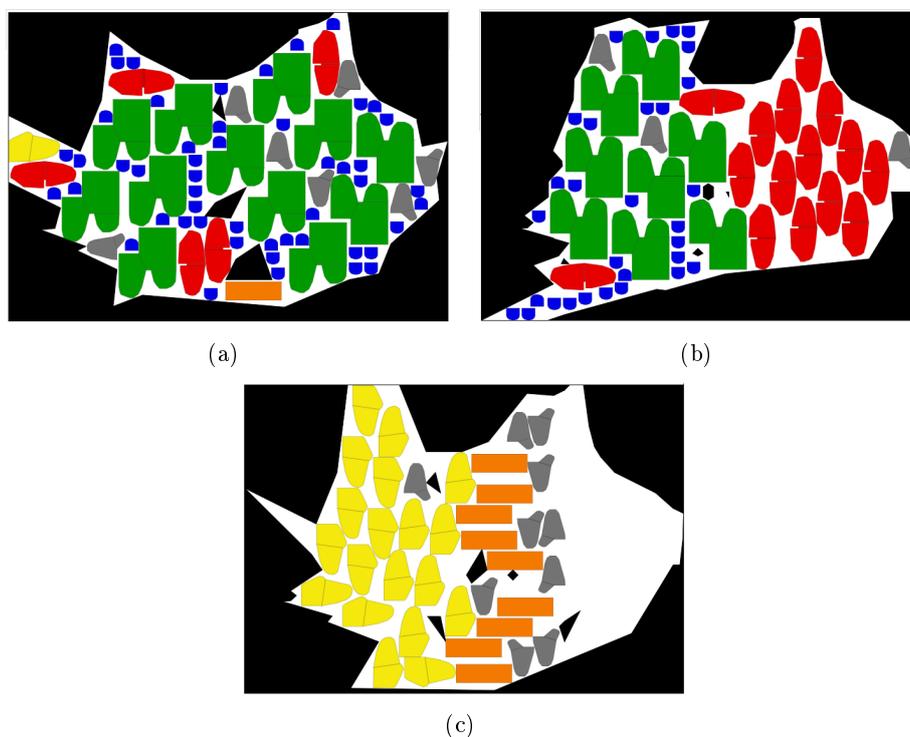


Figura 6: Solução obtida pela heurística *Bottom Left* para a instância com demanda de 10 pares de luva.

5 - Conclusões

Neste artigo foi abordado o problema de uma indústria de luvas de couro do interior de São Paulo. O problema estudado é definido na literatura como um problema de empacotamento de itens irregulares em recipientes (ou *irregular bin packing problem*). O objetivo deste problema é empacotar todos os itens utilizando o menor número de recipientes possível.

Propusemos uma representação alternativa para os itens e os recipientes com defeitos. Os itens são representados por um conjunto de polígonos convexos, enquanto os recipientes são representados por um retângulo (dados por uma largura e uma altura) e uma lista de defeitos, que são polígonos convexos, utilizados para representar regiões onde os itens não podem ser empacotados. Utilizamos o conceito de espaço de obstrução para a construção de leiautes viáveis e uma nova abordagem para lidar com regiões proibidas de recipientes defeituosos. Para o cálculo do espaço de obstrução, implementamos a abordagem proposta por Cuninghame-Green (1989).

Os experimentos computacionais foram baseados em um problema real da indústria de luvas de couro. Criamos a instância a partir de dados de três peças de couro e de um tipo de luva. Aplicamos os algoritmos apresentados para as demandas de um par de luva até onze pares para o tipo de luva escolhido, o que fez os algoritmos lidarem com demandas pequenas, com 17 itens, e grandes, quando falamos de itens irregulares, com 187 itens.

A heurística *Bottom Left* se mostrou superior à heurística de Duas Regiões, obtendo resultados bastante satisfatórios dado o tempo de execução do algoritmo, inferior a 50

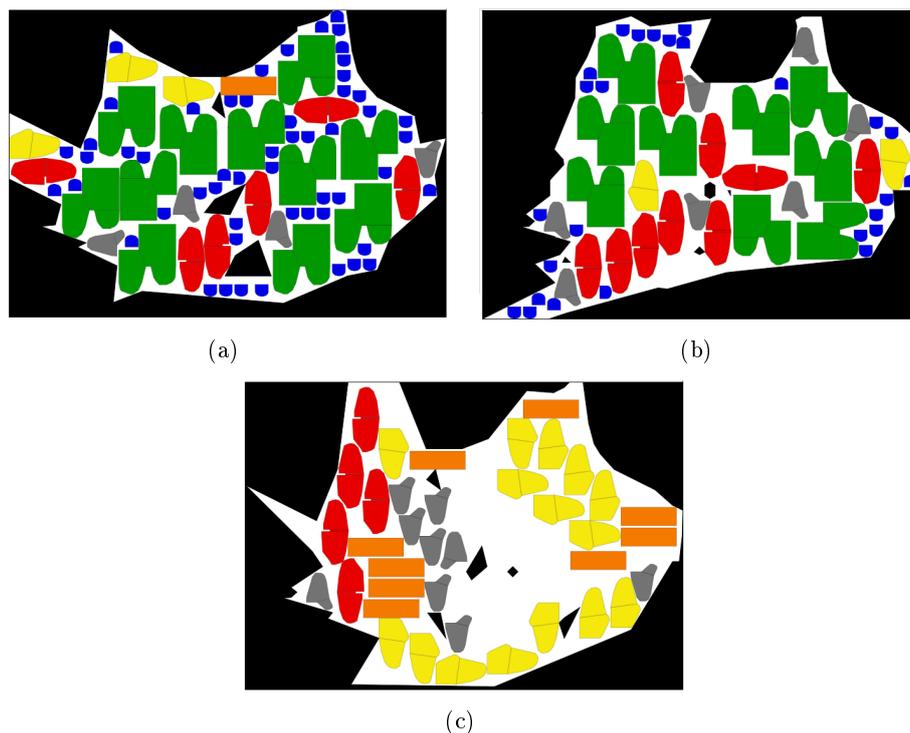


Figura 7: Solução obtida pela heurística de Duas Regiões para a instância com demanda de 10 pares de luva.

segundos. A heurística de Duas Regiões se mostrou uma alternativa, dado que pelo baixo custo computacional as duas heurísticas podem ser executadas e, em caso de empate a empresa poderia optar pela heurística que gere o leiaute mais simples de ser cortado, visto que o corte do couro nesta empresa é feito de forma manual.

Nossa intenção é aplicar os conceitos dessas heurísticas na indústria. Direções futuras para este trabalho são estudar técnicas de compactação e melhorias de solução que possam fornecer padrões de cortes mais simples. Um grande desafio desta aplicação é obter leiautes compactos que possam ser utilizados na empresa.

Agradecimentos

Este trabalho contou com o apoio financeiro da CAPES, CNPq (processo 476792/2013-4) e FAPESP (processos 2010/10133-0 e 2013/07375-0). Agradecemos aos revisores pelos comentários importantes, que tornaram este trabalho melhor.

Referências

- Alvarez-Valdes, R., Parreño, F. e Tamarit, J. (2013). A branch and bound algorithm for cutting and packing irregularly shaped pieces. *Int. J. Production Economics*, 145, 478–487.
- Alves, C., Bras, P., Carvalho, J. M. V. e Pinto, T. (2012). A variable neighborhood search algorithm for the leather nesting problem. *Mathematical Problems in Engineering*, 183, 1–28.
- Baldacci, R., Boschetti, M. A., Ganovelli, M. e Maniezzo, V. (2014). Algorithms for nesting with defects. *Discrete Applied Mathematics*, 163, 17–33.

- Bennell, J. A. e Oliveira, J. F.** (2008). The geometry of nesting problems: A tutorial. *European Journal of Operational Research*, 184, 397–415.
- Bennell, J. A. e Oliveira, J. F.** (2009). A tutorial in irregular shape packing problems. *Journal of the Operational Research Society*, 60, S93–S105.
- Bennell, J. A. e Song, X.** (2010). A beam search implementation for the irregular shape packing problem. *Journal of Heuristics*, 16, 167–188.
- Burke, E., Hellier, R., Kendall, G. e Whitwell, G.** (2006). A new bottom-left-fill heuristic algorithm for the two-dimensional irregular packing problem. *Operational Research*, 54, 587–601.
- Cheng, S. K. e Rao, K. P.** (2005). Genetic algorithm coding methods for leather nesting. *Applied Intelligence*, 23, 9–20.
- Crispin, A., Clay, P., Taylor, G., Bayes, T. e Reedman, D.** (2005). Large-scale nesting of irregular patterns using compact neighborhood algorithm. *Journal of Materials Processing Technology*, 103, 135–140.
- Cuninghame-Green, R.** (1989). Geometry, shoemaking and the milk tray problem. *New Scientist*, 123, S6–S20.
- Dowland, K. A. e Dowland, W. B.** (1995). Solution approaches to irregular nesting problems. *European Journal of Operational Research*, 84, 506–521.
- Dowland, K. A., Vaid, S. e Dowland, W. B.** (2002). An algorithm for polygon placement using a bottom-left strategy. *European Journal of Operational Research*, 141, 371–381.
- Dyckhoff, H.** (1990). A typology of cutting and packing problems. *European Journal of Operational Research*, 44, 1955–1972.
- Egeblad, J., Nielsen, B. K. e Odgaard, A.** (2007). Fast neighborhood search for two and three-dimensional nesting problems. *European Journal of Operational Research*, 183, 1249–1266.
- Gomes, A. M. e Oliveira, J. F.** (2002). A 2-exchange heuristic for nesting problems. *European Journal of Operational Research*, 141, 359–370.
- Gomes, A. M. e Oliveira, J. F.** (2006). Solving irregular strip packing problems by hybridising simulated annealing and linear programming. *European Journal of Operational Research*, 171, 811–829.
- Martello, S., Monaci, M. e Vigo, D.** (2003). An exact approach to the strip-packing problem. *INFORMS Journal on Computing*, 15, 310–319.
- Toledo, F. M. B., Carravilla, M. A., Ribeiro, C., Oliveira, J. F. e Gomes, A. M.** (2013). The dotted-board model: A new mip model for nesting irregular shapes. *Int. J. Production Economics*, 145, 478–487.
- Wascher, G., Haubner, H. e Schumann, H.** (2007). An improved typology of cutting and packing problems. *European Journal of Operational Research*, 183, 1109–1130.
- Yuping, Z., Shouwei, J. e Chunli, Z.** (2005). A very fast simulated re-annealing algorithm for the leather nesting problem. *International Journal of Advanced Manufacturing Technology*, 25, 1113–1118.