

## **Um modelo inteiro misto com trigonometria direta para o corte de polígonos convexos aplicado à indústria**

### **Leandro Resende Mundim**

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

### **Luiz Henrique Cherri**

Instituto de Ciências Matemáticas e de Computação  
Avenida Trabalhador São-carlense, 13566-590, São Carlos - SP, Brasil  
INESCTEC, Faculdade de Engenharia, Universidade do Porto  
Rua Dr. Roberto Frias, 4200-590, Porto, Portugal  
lhcherri@icmc.usp.br

### **Maria Antónia Carravilla**

INESCTEC, Faculdade de Engenharia, Universidade do Porto  
Rua Dr. Roberto Frias, 4200-590, Porto, Portugal  
mac@fe.up.pt

### **José Fernando Oliveira**

INESCTEC, Faculdade de Engenharia, Universidade do Porto  
Rua Dr. Roberto Frias, 4200-590, Porto, Portugal  
jfo@fe.up.pt

### **Franklina Maria Bragion de Toledo**

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

### **Marina Andretta**

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

### **RESUMO**

No problema de corte de itens irregulares em faixa, desejamos alocar um conjunto de itens no menor comprimento de uma faixa com altura fixa. Para resolver este problema diversas heurísticas vêm sendo propostas desde a década de 1960, mas os primeiros modelos de programação matemática foram publicados apenas nos últimos anos. Neste trabalho, apresentamos um novo modelo de programação inteira mista para o problema de corte de itens convexos em faixa. Este modelo foi desenvolvido para uma empresa têxtil e utiliza apenas informações geométricas dos itens para evitar a sobreposição entre eles. Os experimentos computacionais foram realizados utilizando problemas encontrados em uma indústria de corte de aventais e instâncias da literatura. Com os experimentos foi possível verificar a eficácia do modelo em encontrar soluções ótimas para algumas instâncias e em encontrar soluções de boa qualidade quando um critério de parada diferente da otimalidade é utilizado.

**PALAVRAS CHAVE.** Problema de corte de itens irregulares em faixa, modelagem matemática, indústria têxtil.

**Área Principal: PM - Programação Matemática, IND - PO na Indústria.**

#### **ABSTRACT**

In the irregular strip packing the aim is to pack a set of items on a strip of fixed height, minimising the used strip's length. To solve this problem several heuristics have been proposed since the sixties but the first mathematical programming model only recently (last couple of years) have been published. In this work we present a new mixed-integer programming model for the irregular strip packing problem. This model was developed for a textile company and only uses item geometric information to avoid overlap among the several items. The computational experiments were run over problems found in an apron production industry and over instances from the literature. From these experiments it was possible to prove the model's efficacy in finding for some instances optimal solutions and, when the stopping criteria is not the solution optimality proof, good quality feasible solutions.

**KEYWORDS.** Irregular strip packing, mathematical programming, textile industry.

**Main Area:** PM - Mathematical Programming, IND - OR in Industry.

## 1. Introdução

Problemas de corte (ou empacotamento) de objetos pequenos a partir de (ou dentro de) objetos maiores é um problema muito estudado na área de otimização. Estes problemas podem ser encontrados em diversas indústrias, como por exemplo durante o corte de bobinas de papel (caso unidimensional), durante o corte de vidros e placas de madeira (caso bidimensional) e durante o empacotamento de caminhões (caso tridimensional). Uma variante do caso bidimensional é o problema de corte de itens irregulares em faixa, no qual os itens devem ser alocados no interior de um recipiente com altura fixa e comprimento considerado infinito. O objetivo é encontrar um posicionamento factível (sem sobreposição entre itens e todos os itens completamente dentro do recipiente) no menor comprimento de faixa possível. Este problema é encontrado em diversas indústrias, como a moveleira, a de calçados, a de produção automobilística, entre outras.

A grande maioria dos trabalhos de empacotamento em faixa bidimensional lidam com objetos (itens e recipientes) retangulares. Apesar deste ser o caso de muitas indústrias, existe uma grande quantidade de empresas cujo processo de fabricação envolve o corte de itens irregulares. A indústria têxtil, estudada neste trabalho, corta itens diversos e na grande maioria das vezes itens irregulares (não retangulares e não circulares).

Segundo a tipologia proposta por Wäscher et al. (2007), o problema de corte de itens irregulares em faixa (ou *irregular strip packing problem*) é um problema de dimensão aberta. Este problema é estudado por diversos autores, como por exemplo Jakobs (1996), Dowsland et al. (1998, 2002), Gomes e Oliveira (2002), Egeblad et al. (2007), Imamichi et al. (2009), Sato et al. (2012), Elkeran (2013) e muitos outros. A grande maioria dos trabalhos utilizam abordagens heurísticas, que em geral obtêm soluções de boa qualidade em um baixo tempo computacional.

É possível classificar os modelos para a resolução do problema corte de itens irregulares como de domínio contínuo e domínio discreto. Denominamos por modelos contínuos aqueles em que os itens podem ser posicionados em qualquer posição do recipiente. Fischetti e Luzzi (2009) propuseram o primeiro modelo de programação inteira mista para o problema de corte de itens irregulares em faixa em que os itens e a placa são definidos por polígonos convexos e não convexos. Recentemente, Alvarez-Valdes et al. (2013) formalizam a geração de algumas estruturas geométricas utilizadas no modelo de Fischetti e Luzzi (2009) e propõem um algoritmo *branch-and-bound* para a resolução do problema. Além disso, Alvarez-Valdes et al. (2013) estendem o modelo de compactação de Gomes e Oliveira (2006) para o problema do corte de itens irregulares em faixa. No caso dos modelos discretos, o posicionamento dos itens pode ser feito em um número finito de pontos da placa. Nesta linha, Carravilla et al. (2003) propõem um método de solução via programação por restrições para a resolução do problema. Este foi o primeiro método exato para a resolução do problema corte de itens irregulares em faixa. Um modelo de programação inteira mista para a resolução deste problema baseado em pontos de posicionamento finitos foi proposto por Toledo et al. (2013). Com este modelo foi possível resolver até a otimalidade instâncias de maior porte. É importante ressaltar que a otimalidade da solução dos modelos de domínio discreto é condicionada à discretização utilizada.

Todos estes métodos exatos existentes na literatura utilizam a técnica de *no-fit polygon* para evitar a sobreposição entre os itens, seja nos modelos de domínio contínuo ou discreto. Neste trabalho, propomos um novo modelo inteiro misto para polígonos convexos que utiliza trigonometria direta para evitar a sobreposição. Este modelo é mais simples de implementar por não requerer estruturas geométricas complexas. Em experimentos computacionais foi possível verificar que o modelo é competitivo com o melhor modelo contínuo da literatura (HS2 proposto por Alvarez-Valdes et al. (2013)).

O modelo proposto neste trabalho foi desenvolvido para um estudo de caso de uma indústria de aventais do interior de São Paulo. Nesta indústria, os itens que compõem os aventais são polígonos convexos e no problema real são permitidas rotações livres. Por uma questão de simplificação não consideramos rotações no modelo.

Este trabalho está organizado da seguinte forma. Na Seção 2 apresentamos a definição do problema e algumas estruturas básicas, essenciais para a compreensão do restante do trabalho. Na Seção 3, propomos um novo modelo para o problema. Em seguida, na Seção 4 reportamos os resultados computacionais obtidos e na Seção 5 apresentamos as conclusões e direções futuras.

## 2. Definição do problema e estruturas básicas

O problema de corte e empacotamento de itens em faixa (ou *strip packing problem*) é uma das variantes dos problemas de corte e empacotamento. Nestes problemas,  $N$  itens devem ser alocados sem sobreposição no interior de um recipiente de altura  $H$  fixa e comprimento  $L$  ilimitado (ou suficientemente grande para alocar todos os itens).

Os itens são representados por um conjunto de vértices ordenado (em sentido horário). Destes pontos, um é escolhido para ponto de referência do item. A posição do item na placa é definida a partir da localização de seu ponto de referência. A Figura 1 ilustra um item  $i$  e seus vértices. O ponto de referência do item é destacado em preto. Nesta figura também são definidos  $h_i^{\min}$  ( $h_i^{\max}$ ), que representa a distância entre o ponto de referência e o ponto mais acima (abaixo) do item, e  $l_i^{\min}$  ( $l_i^{\max}$ ), que é a distância entre o ponto de referência e o ponto mais à esquerda (direita) do item. Estas distâncias são úteis para garantir que o item está inteiramente contido na placa.

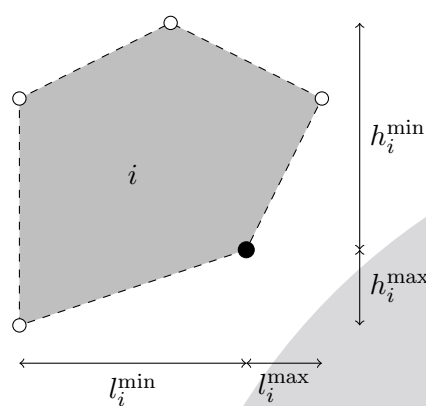


Figura 1: Representação de um item e algumas medidas importantes.

Vários métodos para lidar com a sobreposição já foram propostos, a saber: trigonometria direta (*D-function*) (utiliza diretamente as características geométricas dos itens, sem necessidade de pré-processamento), *no-fit polygon* (região, calculada a priori, onde os itens se sobrepõem) e *phi-function* (equação matemática que informa a posição relativa de dois itens). O cálculo dos *no-fit polygons* e a definição das *phi-functions* requer ferramentas geométricas que são complexas e de difícil implementação.

Neste trabalho, a sobreposição entre os itens é evitada utilizando trigonometria diretamente sobre as informações geométricas referentes aos itens. A única função utilizada é a *D-function*. Considerando os pontos  $a = (a_x, a_y)$ ,  $b = (b_x, b_y)$  e  $r = (r_x, r_y)$ , a *D-function* é dada por:

$$D_{abr} = (a_x - b_x)(a_y - r_y) - (a_y - b_y)(a_x - r_x). \quad (1)$$

A Figura 2, mostra os valores que  $D_{abr}$  pode assumir. Se  $D_{abr} > 0$  (Figura 2i), o ponto  $r$  está à esquerda da linha que passa pelos pontos  $a$  e  $b$ ; se  $D_{abr} = 0$  (Figura 2ii), o ponto  $r$  está sobre da linha que passa pelos pontos  $a$  e  $b$ ; e  $D_{abr} < 0$  (Figura 2iii) se o ponto  $r$  está a direita da reta que passa pelos pontos  $a$  e  $b$ .

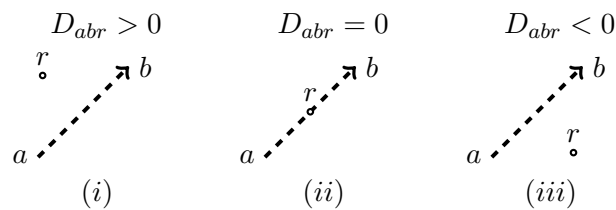


Figura 2: Exemplo de valores que a *D-function* pode assumir.

O conceito de *inner-fit polygon* é utilizado para delimitar as posições viáveis para o posicionamento dos itens dentro do recipiente. Mais especificamente, o *inner-fit polygon* determina a região dentro do recipiente onde o ponto de referência do item pode ser alocado de forma que o item esteja completamente contido no recipiente. Quando o recipiente é retangular o *inner-fit polygon* é um retângulo e pode ser calculado por uma simples adição de coordenadas. A Figura 3 ilustra o *inner-fit polygon* de um item *i*.

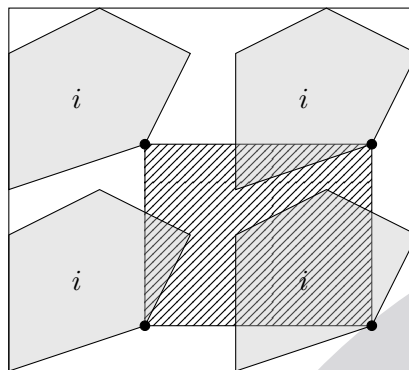


Figura 3: Exemplo do *inner-fit polygon* (região hachurada) de um item *i* em um recipiente retangular.

### 3. Modelo inteiro misto

Um modelo de programação inteira mista para a resolução do problema de corte de itens irregulares em faixa é apresentado nesta seção. A maior vantagem deste modelo comparado com outros da literatura está na forma como as restrições de não sobreposição são tratadas. As restrições de não sobreposição entre itens são as que apresentam a maior dificuldade nos problemas de corte de itens irregulares.

Para evitar a sobreposição entre cada par de itens, utilizaremos apenas análises sobre os vértices dos polígonos que compõem os itens, não utilizando portanto ferramentas geométricas mais complexas que necessitam de cálculo a priori, como é o caso do *no-fit polygon*.

Para definir as restrições de não sobreposição, considere os pontos  $a^k = (a_x^k, a_y^k)$  e  $b^k = (b_x^k, b_y^k)$ , dois vértices consecutivos que compõem uma aresta *k* de um item *i*. Definimos então  $g_x^{irj}$  e  $g_y^{irj}$  como as distâncias horizontal e vertical entre o ponto de posicionamento de um item *i* ( $x_i, y_i$ ), e o vértice *r* de um item *j*. Com isto, podemos redefinir a *D-function* apresentada em (1) criando a desigualdade (2) que, se satisfeita, elimina a sobreposição entre os itens *i* e *j*.

$$D_{abg} = (a_x^k - b_x^k)(a_y^k - g_y^{irj}) - (a_y^k - b_y^k)(a_x^k - g_x^{irj}) \leq 0. \quad (2)$$

Esta expressão pode ser reescrita substituindo a distância ao vértice *r* do item *j* e o ponto de referência do item *i* pela soma da distância entre os pontos de referência dos itens *i* e *j* mais a distância do ponto de referência do item *j* ao vértice *r*. Considere assim a distância horizontal e

vertical entre o ponto de referência de um item  $j$  e o seu vértice  $r$ , dadas respectivamente por  $g_x^{jrr}$  e  $g_y^{jrr}$ . Utilizando este fato, podemos reescrever a desigualdade (2) como apresentado em (3).

$$(a_x^k - b_x^k)(a_y^k + y_i - y_j - g_y^{jrr}) - (a_y^k - b_y^k)(a_x^k + x_i - x_j - g_x^{jrr}) \leq 0 \quad (3)$$

Com algumas manipulações algébricas, obtemos a seguinte desigualdade (4).

$$C_{ij}^{kr} + (a_x^k - b_x^k)(y_i - y_j) + (a_y^k - b_y^k)(x_i - x_j) \leq 0, \quad (4)$$

em que  $C_{ij}^{kr}$  é a constante definida por  $(a_x^k - b_x^k)(a_y^k - g_y^{jrr}) - (a_y^k - b_y^k)(a_x^k - g_x^{jrr})$ .

Vale a pena ressaltar que não é necessário criar uma restrição para cada vértice  $r$  de um item  $j$ , visto que este somente é relevante para a definição da constante  $C_{ij}^{kr}$ . De fato, definindo  $C_{ij}^k$  como a constante dada por  $\max_r \{C_{ij}^{kr}\}$ , a restrição (4) pode ser simplificada como apresentado em (5).

$$C_{ij}^k + (a_x^k - b_x^k)(y_i - y_j) + (a_y^k - b_y^k)(x_i - x_j) \leq 0 \quad (5)$$

A restrição (4) ou (5) garante a não sobreposição entre dois itens convexos, impondo que todos os vértices de um item  $j$  estejam à direita da reta definida pela aresta  $k$  de um item  $i$ . No entanto esta condição não tem que ser verificada para todas as arestas do item  $i$ , bastando que seja verificada para uma das arestas. Deste modo, precisamos ainda definir qual reta será ativada para garantir a não sobreposição entre os itens minimizando o comprimento da placa utilizado. Para isto, considere as variáveis  $v_{ij}^k$ , que assumem o valor 1 se a reta  $k$  de um item  $i$  é a reta que separa os itens  $i$  e  $j$ , e 0 caso contrário. Considere  $K_i$  o conjunto de retas que contêm uma aresta do polígono  $i$ . A nova restrição pode ser formulada como abaixo:

$$C_{ij}^k + (a_x^k - b_x^k)(y_i - y_j) + (a_y^k - b_y^k)(x_i - x_j) \leq (1 - v_{ij}^k)M, \\ i = 1, \dots, N, j = 1, \dots, N, i \neq j, k \in K_i,$$

em que  $M$  é um número real suficientemente grande para manter a desigualdade sempre válida.

Sabemos que, para cada par de itens  $i$  e  $j$ , uma desigualdade deve ser ativa para evitar a sobreposição entre os itens. Desta forma, é necessário, criar uma restrição que garanta que uma das desigualdades relacionadas a cada par de itens será válida. A seguir, apresentamos a restrição que garante que exatamente uma das restrições que separam os itens  $i$  e  $j$  será ativa.

$$\sum_{k \in K_i} v_{ij}^k + \sum_{k \in K_j} v_{ji}^k = 1, \quad 1 \leq i < j \leq N.$$

Para garantir a factibilidade da solução, é necessário que todos os itens a serem cortados estejam dentro da placa. Como a placa é retangular, esta restrição pode ser garantida utilizando informações geométricas simples sobre cada item, apresentadas na Figura 1. As restrições abaixo garantem que os itens estão inteiramente alocados dentro da placa.

$$l_i^{\min} \leq x_i, \quad i = 1, \dots, N, \\ h_i^{\min} \leq y_i \leq H - h_i^{\max}, \quad i = 1, \dots, N.$$

Nosso objetivo é cortar todos os itens dentro da faixa de menor comprimento possível  $L$ . Para dar suporte a esta função objetivo, introduzimos as restrições

$$x_i \leq L - l_i^{\max}, \quad i = 1, \dots, N,$$

que garantem que o comprimento da placa utilizado pelo corte dos itens mais à direita seja contabilizado.

O modelo completo, denominado Modelo de Trigonometria Direta (MTD), é apresentado em (6)-(12).

$$\text{minimizar } L \tag{6}$$

$$\text{sujeito a } l_i^{\min} \leq x_i \leq L - l_i^{\max}, \quad i = 1, \dots, N, \tag{7}$$

$$h_i^{\min} \leq y_i \leq H - h_i^{\max}, \quad i = 1, \dots, N, \tag{8}$$

$$C_{ij}^k + (a_x^k - b_x^k)(y_i - y_j) + (a_y^k - b_y^k)(x_i - x_j) \leq (1 - v_{ij}^k)M, \quad i, j = 1, \dots, N, \quad i \neq j, k \in K_i, \tag{9}$$

$$\sum_{k \in K_i} v_{ij}^k + \sum_{k \in K_j} v_{ji}^k = 1, \quad 1 \leq i < j \leq N, \tag{10}$$

$$(x_i, y_i) \in \mathcal{R}^2, \quad i = 1, \dots, N, \tag{11}$$

$$v_{ij}^k \in [0, 1], \quad i, j = 1, \dots, N, \quad i \neq j, k \in K_i. \tag{12}$$

A relaxação linear do MTD (6)-(12) fornece limitantes inferiores de má qualidade. Bons limitantes inferiores podem ajudar na velocidade de convergência dos métodos de solução. Para este problema, um limitante trivial é dado pelo comprimento do item mais longo a ser cortado. Outro limitante válido está associado à área necessária para efetuar o corte dos itens, e é calculado somando a área de todos os itens a serem cortados e dividindo pela altura da placa. Note que este limitante corresponde ao valor da solução se não houver desperdício no corte. O máximo entre estes dois limitantes é tido como o limitante inferior inicial para o valor da solução do MTD.

#### 4. Experimentos computacionais

Nesta seção apresentamos os experimentos computacionais para o MTD proposto neste trabalho, implementado em linguagem C++ e usando o software de otimização IBM ILOG CPLEX 12.6. Todos os experimentos foram realizados em um computador Intel Core i7-2600 com 16 GB de memória RAM usando o sistema operacional Ubuntu 12.04.2 LTS. O Tempo Limite (TL) para o modelo provar a otimalidade foi restringido a 3600 segundos (uma hora).

Na Seção 4.1, são descritas as instâncias utilizadas nos experimentos computacionais. Na Seção 4.2 é apresentada uma análise comparativa entre os resultados obtidos e os resultados de modelo HS2 proposto por Alvarez-Valdes et al. (2013), que é atualmente o melhor modelo da literatura. Na Seção 4.3, apresentamos experimentos com instâncias reais.

##### 4.1. Instâncias

Para realizar os testes computacionais, dois conjuntos de instâncias foram utilizados. O primeiro conjunto é composto por instâncias com itens convexos utilizadas em Alvarez-Valdes et al. (2013). Com este conjunto pretende-se avaliar a eficiência do MTD frente ao modelo HS2 de Alvarez-Valdes et al. (2013). As instâncias utilizadas são: "fu" e suas variações (fu5, fu6, fu7, fu8, fu9 e fu10) e "three", também com suas variações (threep2, threep2w9, threep3 e threep3w9). Mais detalhes sobre estas instâncias podem ser encontrados em Alvarez-Valdes et al. (2013).

O segundo conjunto de experimentos é composto por instâncias reais provenientes do corte de aventais na indústria têxtil. Consideramos um tipo específico de avental, denominado avental de churrasco, que é composto pelo corpo do avental e o bolso.

#### 4.2. Analisando a performance do MTD

Nesta seção, é feita a comparação entre o MTD e o modelo HS2 de Alvarez-Valdes et al. (2013). Os resultados do modelo HS2 foram retirados de Alvarez-Valdes et al. (2013). Sob a perspectiva computacional, os computadores utilizados para realizar os testes são semelhantes diferindo apenas na versão do software de otimização IBM ILOG CPLEX (Alvarez-Valdes et al. (2013) utilizam a versão 12.1).

A Tabela 1 apresenta os resultados computacionais para o MTD e o modelo HS2. Na tabela, o nome da instância resolvida é apresentado na primeira coluna, seguido pelo número de itens da instância, apresentado na segunda coluna. As colunas três, quatro e cinco apresentam, respectivamente, o comprimento da solução (limitante superior), gap de otimalidade e tempo computacional em segundos para o modelo HS2. Para o MTD, o limitante superior, gap de otimalidade e tempo computacional são apresentados nas colunas seis, sete e oito. O gap de otimalidade é dado pela diferença entre os Limitantes Superior (LS) e Limitante Inferior (LI), dividida pelo limitante inferior, ou seja,  $(LS-LI)/LS$ .

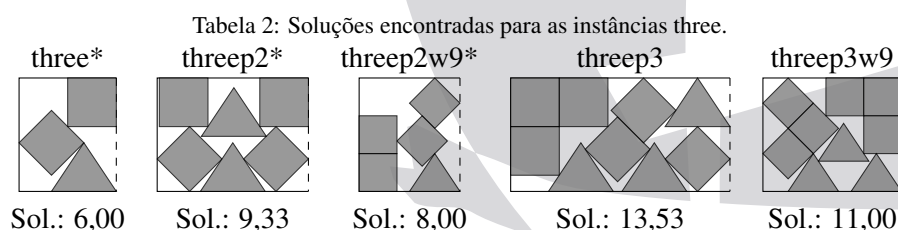
Tabela 1: Resultados computacionais para as instâncias do grupo 1.

Instância	Número de itens	HS2			MTD		
		LS	GAP	Tempo	LS	GAP	Tempo
three	3	6,00	0,00	0,8	6,00	0,00	0,0
fu_5	5	17,89	0,00	0,1	17,89	0,00	0,1
fu_6	6	23,00	0,00	0,5	23,00	0,00	0,0
threep2	6	9,33	0,00	3,9	9,33	0,00	0,8
threep2w9	6	8,00	0,00	8,5	8,00	0,00	4,8
fu_7	7	24,00	0,00	1,0	24,00	0,00	0,1
fu_8	8	24,00	0,00	1,3	24,00	0,00	0,1
fu_9	9	25,00	0,00	70,0	25,00	0,00	9,6
threep3	9	13,53	0,00	3394,0	13,53	0,27	TL
threep3w9	9	11,00	0,09	TL	11,00	0,23	TL
fu_10	10	28,68	0,00	3064,0	28,68	0,00	260,0
fu	12	33,80	0,29	TL	33,14	0,14	TL

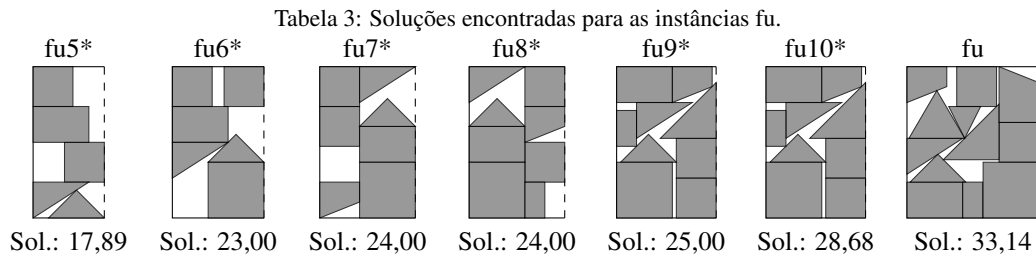
TL: O tempo limite de uma hora foi atingido.

Na Tabela 1, é possível notar que o modelo proposto (MTD) consegue provar a otimalidade para 9 das 12 instâncias avaliadas, enquanto o modelo HS2 prova a otimalidade para 10 destas instâncias. Em contrapartida, em relação ao limitante superior, o modelo MTD é sempre melhor ou igual ao HS2.

As Tabelas 3 e 2 apresentam o desenho das soluções obtidas pelo modelo MTD e também o valor da solução obtida para cada instância. As instâncias que possuem um asterisco na frente do nome atingiram a otimalidade.







### 4.3. Aplicando o método a problemas reais

Nesta seção, apresentamos a instância da indústria considerada neste trabalho e, em seguida, os experimentos computacionais. A Figura 4 ilustra o desenho de um avental de churrasco. Como podemos observar, existem duas partes a serem cortadas: corpo do avental (em cinza claro) e o bolso (em cinza escuro). As linhas pontilhadas representam as outras partes do avental que são feitas de outro material e são ilustradas por completeude.

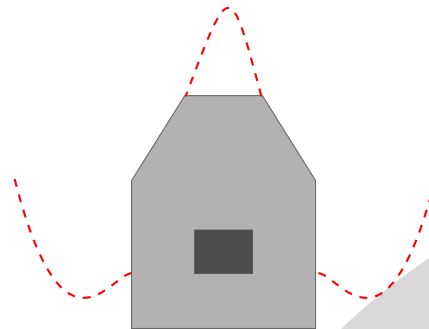


Figura 4: Desenho do avental de churrasco.

Na empresa estudada, há uma certa demanda diária de aventais a serem produzidos. Para simular alguns cenários, os experimentos computacionais apresentam instâncias denominadas  $Avental_i$ , onde o termo  $i$  representa o número produzido de aventais. São 16 instâncias para  $i = 1, \dots, 10$  e instâncias de grande porte com  $i = 20, 30, 40, 50, 60$  e  $70$ .

Os resultados são apresentados na Tabela 4. A primeira coluna apresenta o nome da instância, seguida pela coluna de quantidade de itens. As colunas seguintes reportam o limitante inferior (terceira coluna), o comprimento ou limitante superior (quarta coluna), o GAP (quinta coluna), a taxa de ocupação da solução (sexta coluna) e o tempo computacional em segundos (sétima coluna). Podemos observar que o MTD foi muito eficiente para a aplicação, obtendo soluções ótimas em pouco tempo computacional para as cinco primeiras instâncias. Podemos observar que, a partir de 12 itens, o MTD não conseguiu provar a otimalidade de nenhuma instância, mas conseguiu soluções com uma ocupação superior a 80% para as instâncias com uma demanda par de aventais. Isso sugere, que sempre que possível, a empresa deve cortar uma demanda par de aventais. O MTD não encontrou solução para a instância  $Avental_{70}$  com 140 itens.

A Figura 5 apresenta o desenho das soluções  $Avental_8$  e  $Avental_{10}$  que obtiveram as melhores taxas de ocupação. A Figura 6 apresenta o desenho da solução com 120 itens (60 aventais).

Tabela 4: Resultados do modelo proposto para uma instância real.

Instância	Número de itens	MTD				
		LI	LS	GAP	Ocupação	Tempo
<i>Avental</i> <sub>1</sub>	2	6,40	6,40	0,00	47,51	0,0
<i>Avental</i> <sub>2</sub>	4	7,21	7,21	0,00	84,35	0,0
<i>Avental</i> <sub>3</sub>	6	12,80	12,80	0,00	71,27	0,0
<i>Avental</i> <sub>4</sub>	8	13,39	13,39	0,00	90,84	13,4
<i>Avental</i> <sub>5</sub>	10	19,20	19,20	0,00	79,19	10,6
<i>Avental</i> <sub>6</sub>	12	19,20	19,79	0,03	92,19	TL
<i>Avental</i> <sub>7</sub>	14	20,05	25,60	0,22	83,15	TL
<i>Avental</i> <sub>8</sub>	16	22,19	26,19	0,15	92,88	TL
<i>Avental</i> <sub>9</sub>	18	25,79	32,00	0,19	85,52	TL
<i>Avental</i> <sub>10</sub>	20	28,66	32,80	0,13	92,71	TL
<i>Avental</i> <sub>20</sub>	40	50,30	70,00	0,28	86,88	TL
<i>Avental</i> <sub>30</sub>	60	85,00	102,00	0,17	89,44	TL
<i>Avental</i> <sub>40</sub>	80	114,62	144,40	0,21	84,24	TL
<i>Avental</i> <sub>50</sub>	100	143,28	186,40	0,23	81,57	TL
<i>Avental</i> <sub>60</sub>	120	171,93	221,77	0,22	82,27	TL
<i>Avental</i> <sub>70</sub>	140	-	-	-	-	TL

TL: O tempo limete de uma hora foi atingido.

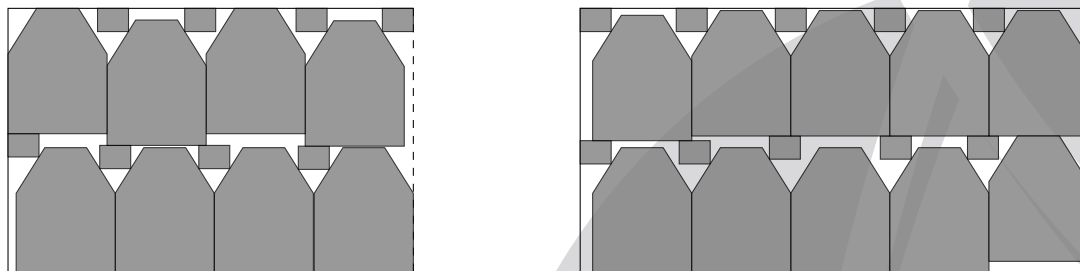


Figura 5: Desenho das soluções das instâncias *Avental*<sub>8</sub> (à esquerda) e *Avental*<sub>10</sub> (à direita).

## 5. Conclusões e pesquisas futuras

Neste trabalho apresentamos um modelo de programação inteira mista para o problema de corte em faixa em que apenas trigonometria direta é utilizada para eliminar a sobreposição entre os itens. A formulação apenas pode representar problemas que envolvem o corte de itens convexos. O modelo proposto (MTD) é mais fácil de implementar que os modelos da literatura, principalmente pelo fato de precisar apenas das informações geométricas dos itens, sem que requeira algum pré-processamento. Apesar da aparente simplicidade do modelo, ele se mostrou competitivo com a literatura, encontrando vários resultados ótimos e sempre soluções de igual ou melhor qualidade. O MTD foi testado em instâncias inspiradas em uma indústria têxtil, conseguindo bons resultados que podem impactar em uma redução do custo de material (ou matéria-prima) e reduzir o impacto ambiental, causado pela fabricação de aventais. O MTD obteve soluções com mais de 80% de ocupação, para todas as instâncias com uma demanda par, enquanto as demandas ímpares (com as demanda de 1, 3, 5, 7 e 9) obtiveram as piores taxas de ocupação. Para quatro instância o MTD obteve mais de 90% de ocupação, o que representa um leiaute menos de 10% de desperdício que vira lixo industrial.

O MTD é o primeiro modelo a apresentar soluções para instâncias com mais de 70 itens, chegando ao limite de 120 itens. O modelo que resolveu a maior instância da literatura até o

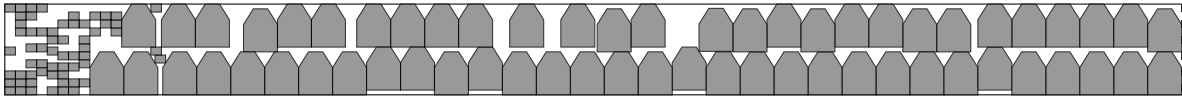


Figura 6: Desenho da solução da instância *Avental*<sub>60</sub>.

momento foi o de Toledo et al. (2013), que resolveu instâncias com 56 itens na otimalidade e encontrou solução para uma instância com 70 itens.

Como próximos passos da pesquisa, serão estudadas formas de estender o modelo proposto para polígonos não convexos, além de considerar rotações para os itens. Pretendemos estudar novos limitantes inferiores e gerar cortes no modelo a fim de acelerar a sua velocidade de convergência.

### Agradecimentos.

Este trabalho contou com o apoio da CAPES, do CNPq (306918/2014-5), da FAPESP Temático (2010/10133-0), CEPID da FAPESP (processo 2013/07375-0) e o Universal do CNPq (processo 476792/2013-4). Este trabalho foi ainda parcialmente financiado pelo ERDF - European Regional Development Fund - através do Programa ON.2, e por fundos nacionais portugueses, através da FCT, no âmbito do projeto "Smart Manufacturing and Logistics"[Projeto NORTE - 07 - 0124 - FEDER - 000057].

### Referências

- Alvarez-Valdes, R.; Martínez, A.; Tamarit, J.** (2013). A branch & bound algorithm for cutting and packing irregularly shaped pieces. *International Journal of Production Economics*, 145(2), 463 – 477.
- Carravilla, M. A.; Ribeiro, C.; Oliveira, J. F.** (2003). Solving nesting problems with non-convex polygons by constraint logic programming. *International Transactions in Operational Research*, 10, 651–663.
- Dowland, K. A.; Dowland, W. B.; Bennell, J. A.** (1998). Jostle for position: Local improvement for irregular cutting patterns. *The Journal of the Operational Research Society*, 49(6), 647–658.
- Dowland, K. A.; Vaid, S.; Dowland, W. B.** (2002). An algorithm for polygon placement using a bottom-left strategy. *European Journal of Operational Research*, 141(2), 371–381.
- Egeblad, J.; Nielsen, B. K.; Odgaard, A.** (2007). Fast neighborhood search for two- and three-dimensional nesting problems. *European Journal of Operational Research*, 183(3), 1249–1266.
- Elkeran, A.** (2013). A new approach for sheet nesting problem using guided cuckoo search and pairwise clustering. *European Journal of Operational Research*, 231(3), 757 – 769.
- Fischetti, M.; Luzzi, I.** (2009). Mixed-integer programming models for nesting problems. *Journal of Heuristics*, 15(3), 201–226.
- Gomes, A. M.; Oliveira, J. F.** (2002). A 2-exchange heuristic for nesting problems. *European Journal of Operational Research*, 141(2), 359 – 370.
- Gomes, A.; Oliveira, J. F.** (2006). Solving irregular strip packing problems by hybridising simulated annealing and linear programming. *European Journal of Operational Research*, 171(3), 811–829.
- Imamichi, T.; Yagiura, M.; Nagamochi, H.** (2009). An iterated local search algorithm based on nonlinear programming for the irregular strip packing problem. *Discrete Optimization*, 6(4), 345 – 361.
- Jakobs, S.** (1996). On genetic algorithms for the packing of polygons. *European Journal of Operational Research*, 88(1), 165–181.
- Sato, A. K.; Martins, T. C.; Tsuzuki, M. S. G.** (2012). An algorithm for the strip packing problem using collision free region and exact fitting placement. *Computer-Aided Design*, 44(8), 766 – 777.

**Toledo, F. M. B.; Carravilla, M. A.; Ribeiro, C.; Oliveira, J. F.; Gomes, A. M. (2013).** The dotted-board model: a new mip model for nesting irregular shapes. *International Journal of Production Economics*, 145(2), 478 – 487.

**Wäscher, G.; Haußner, H.; Schumann, H. (2007).** An improved typology of cutting and packing problems. *European Journal of Operational Research*, 183(3), 1109 – 1130.

