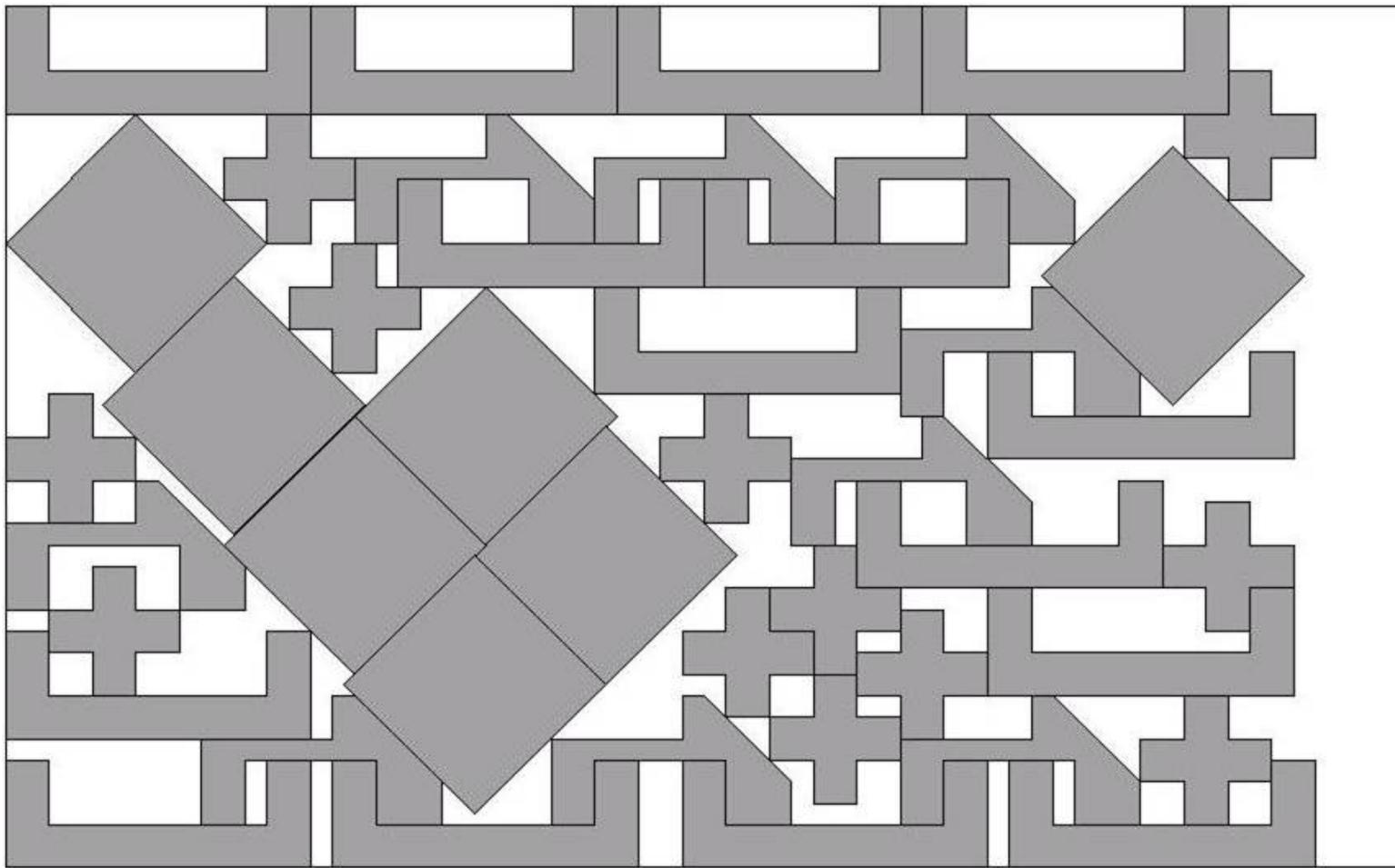
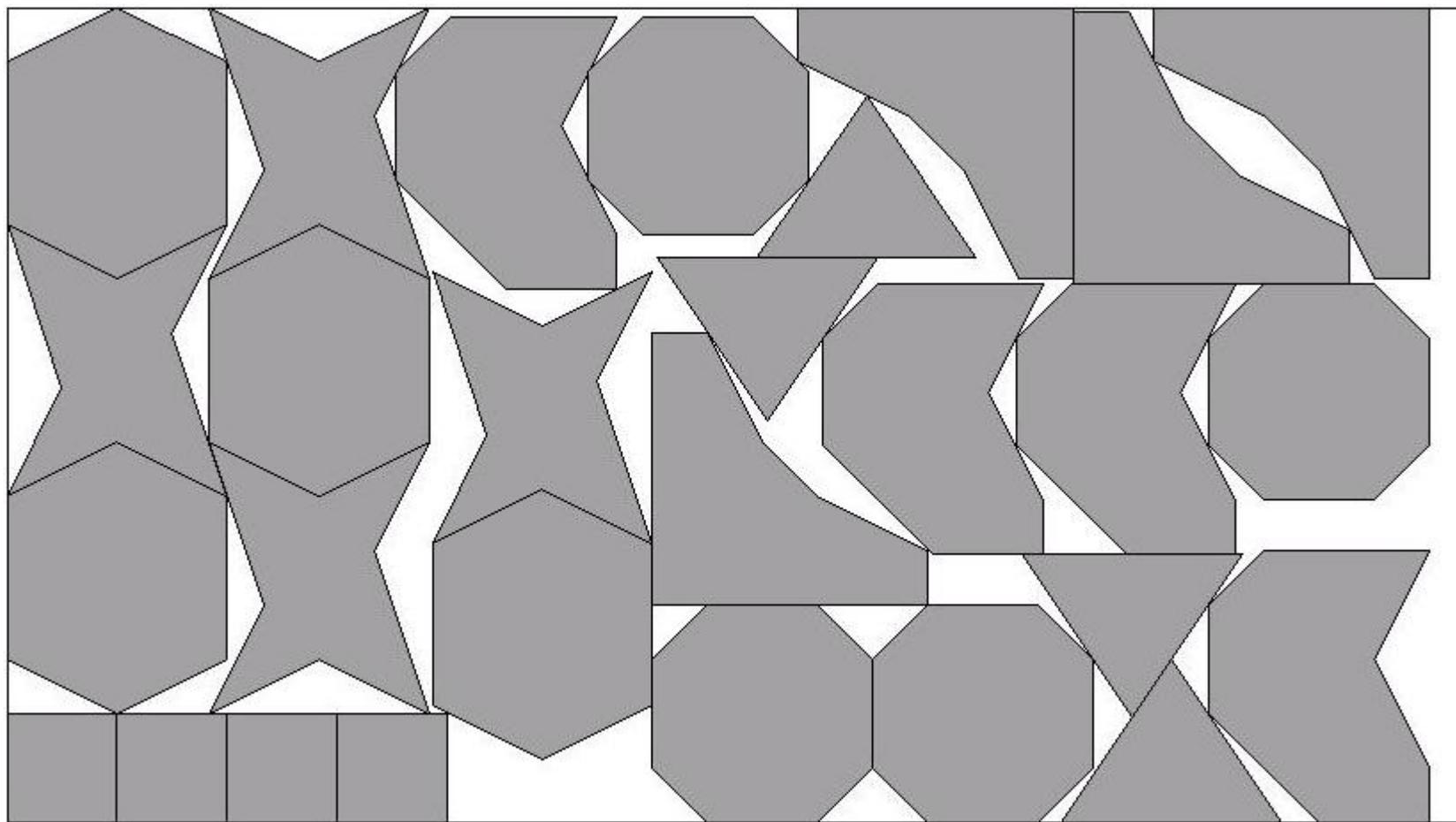


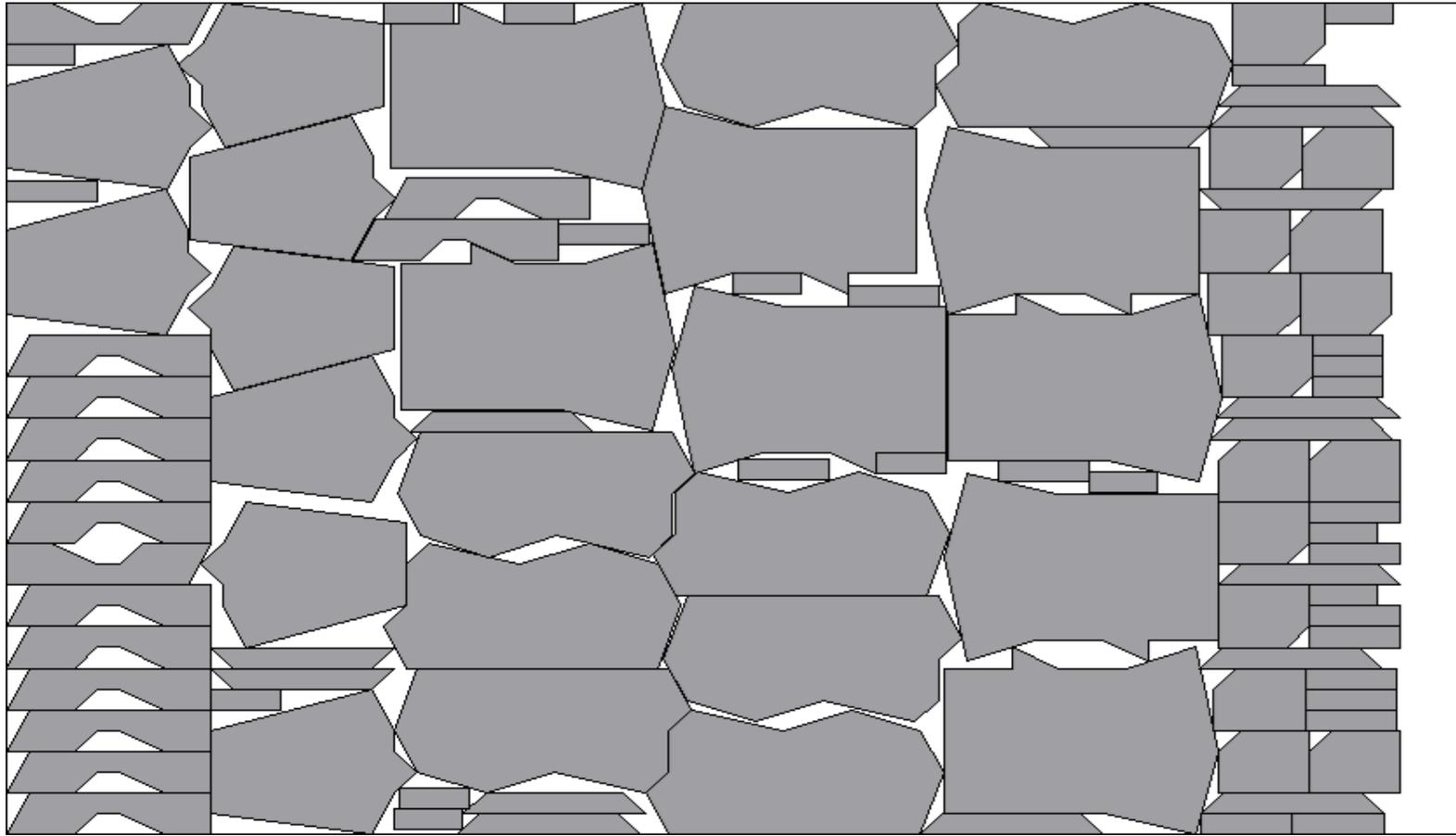
Como resolver quebra-cabeças em que as peças não encaixam certinho

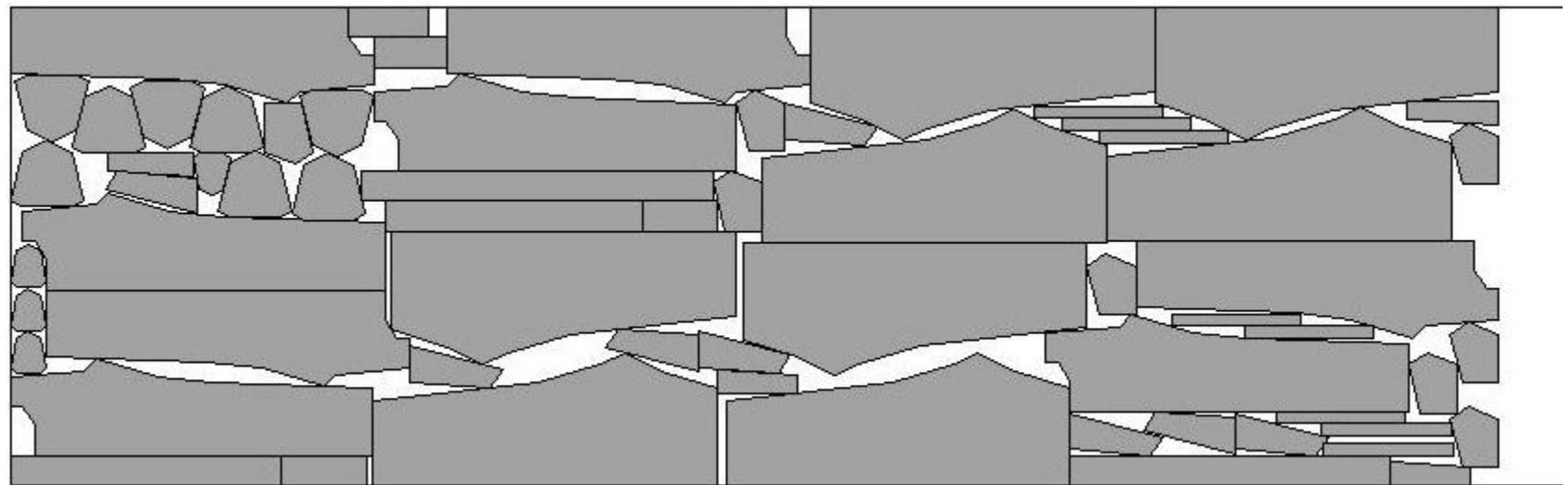


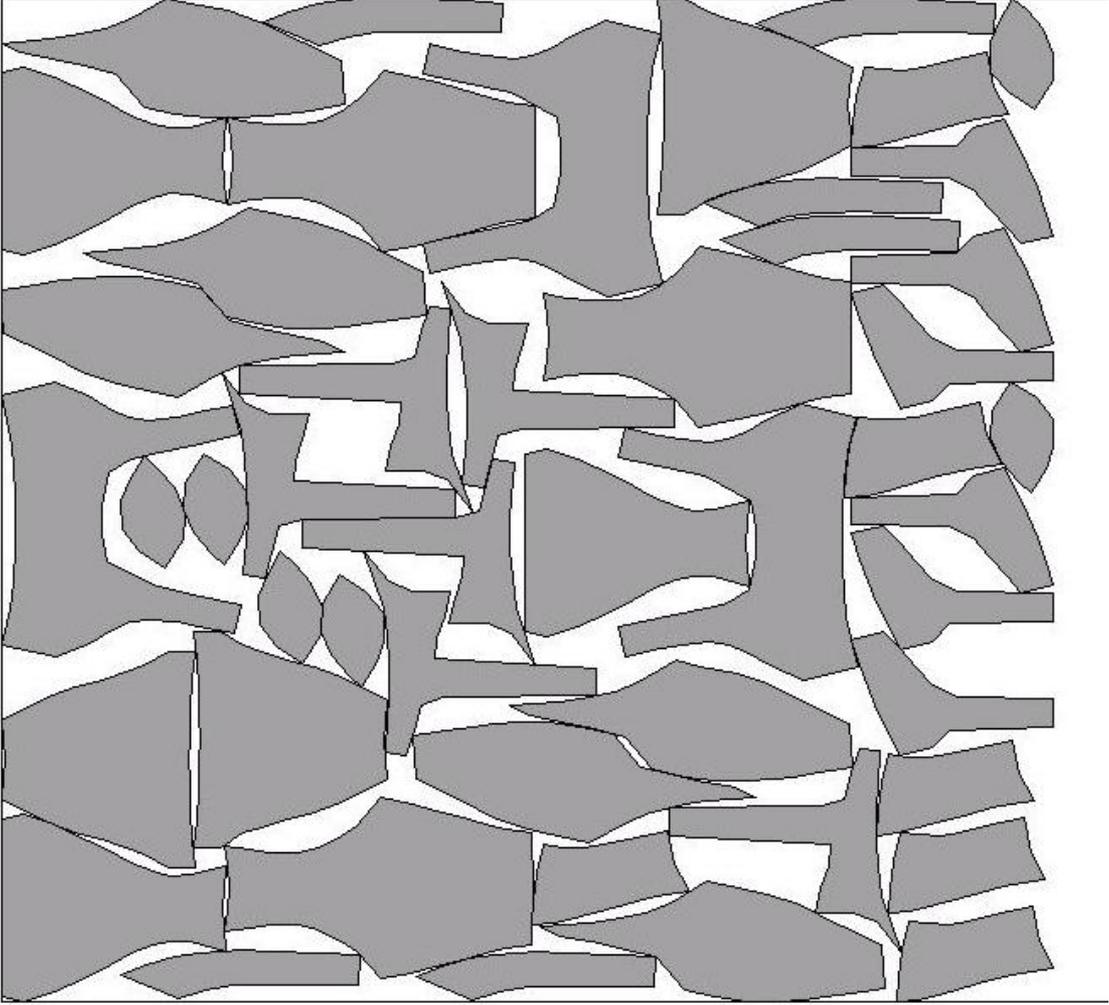
*José Fernando Oliveira
Universidade do Porto, Portugal*

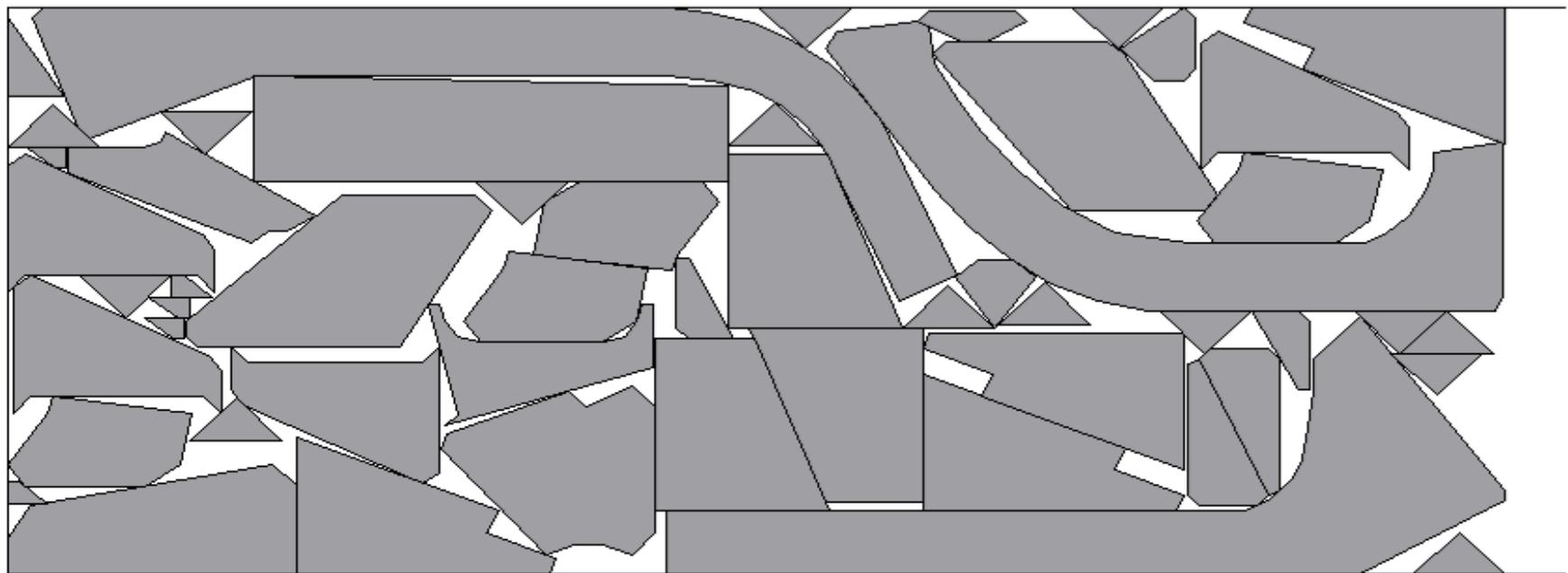


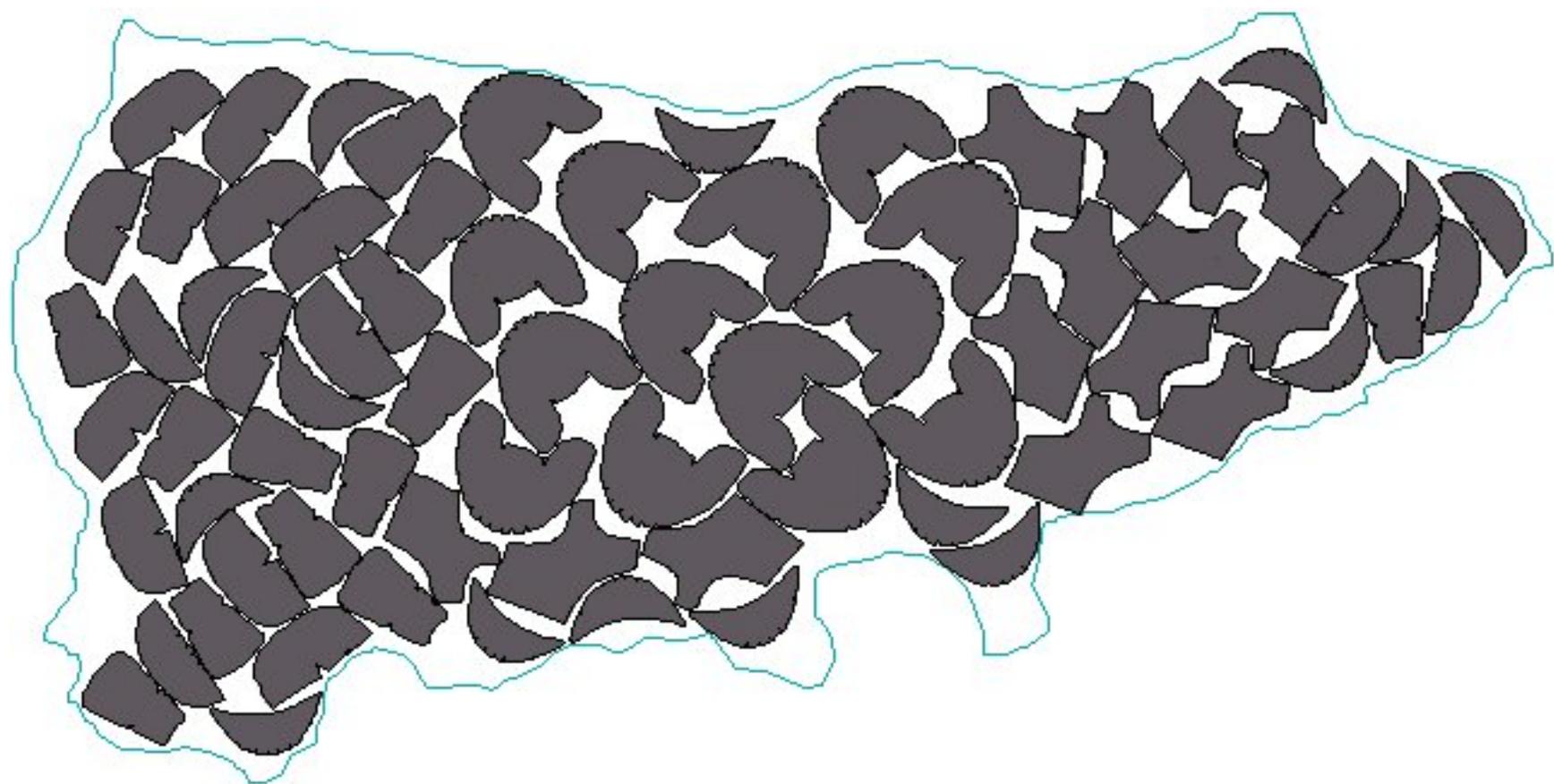


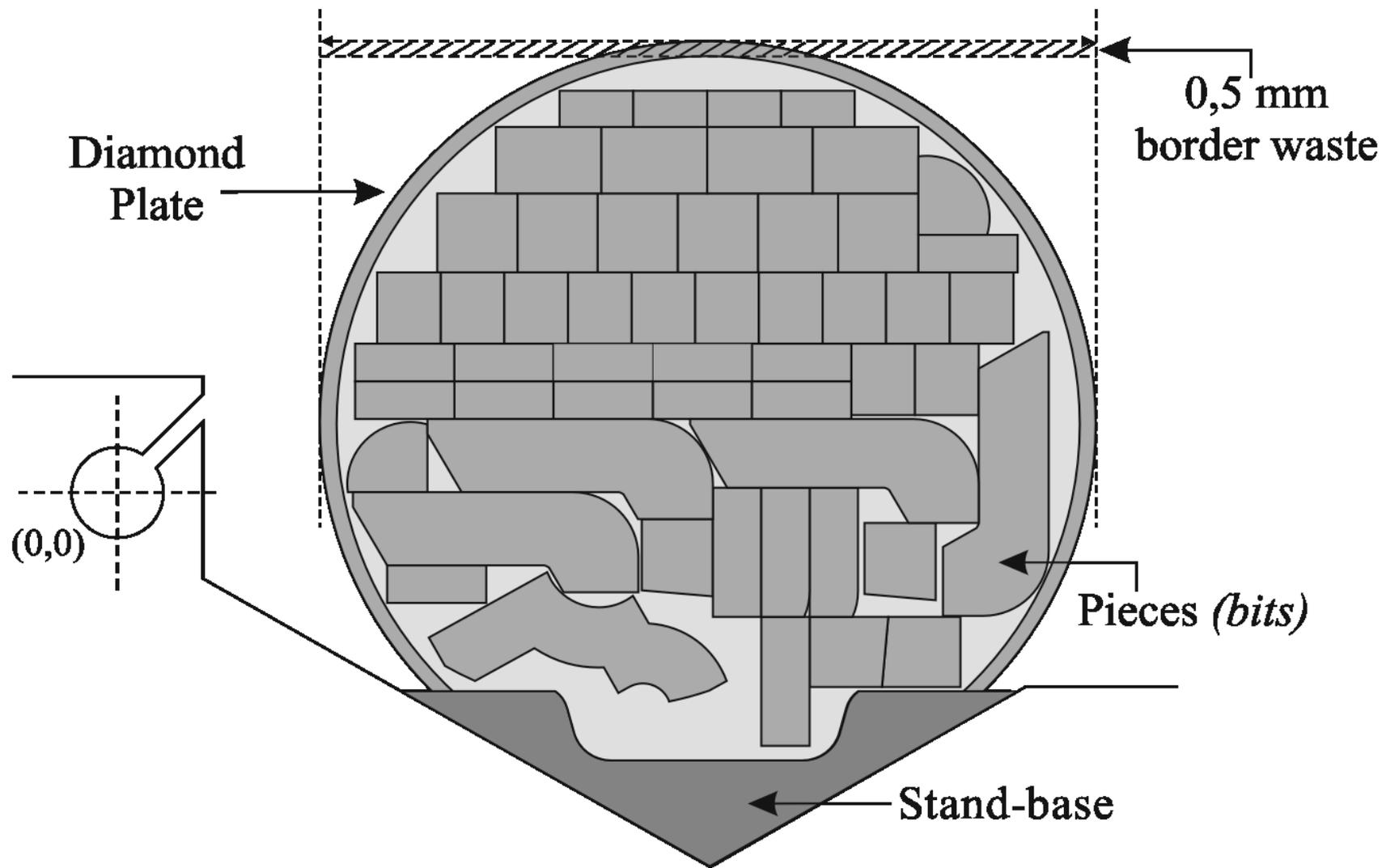






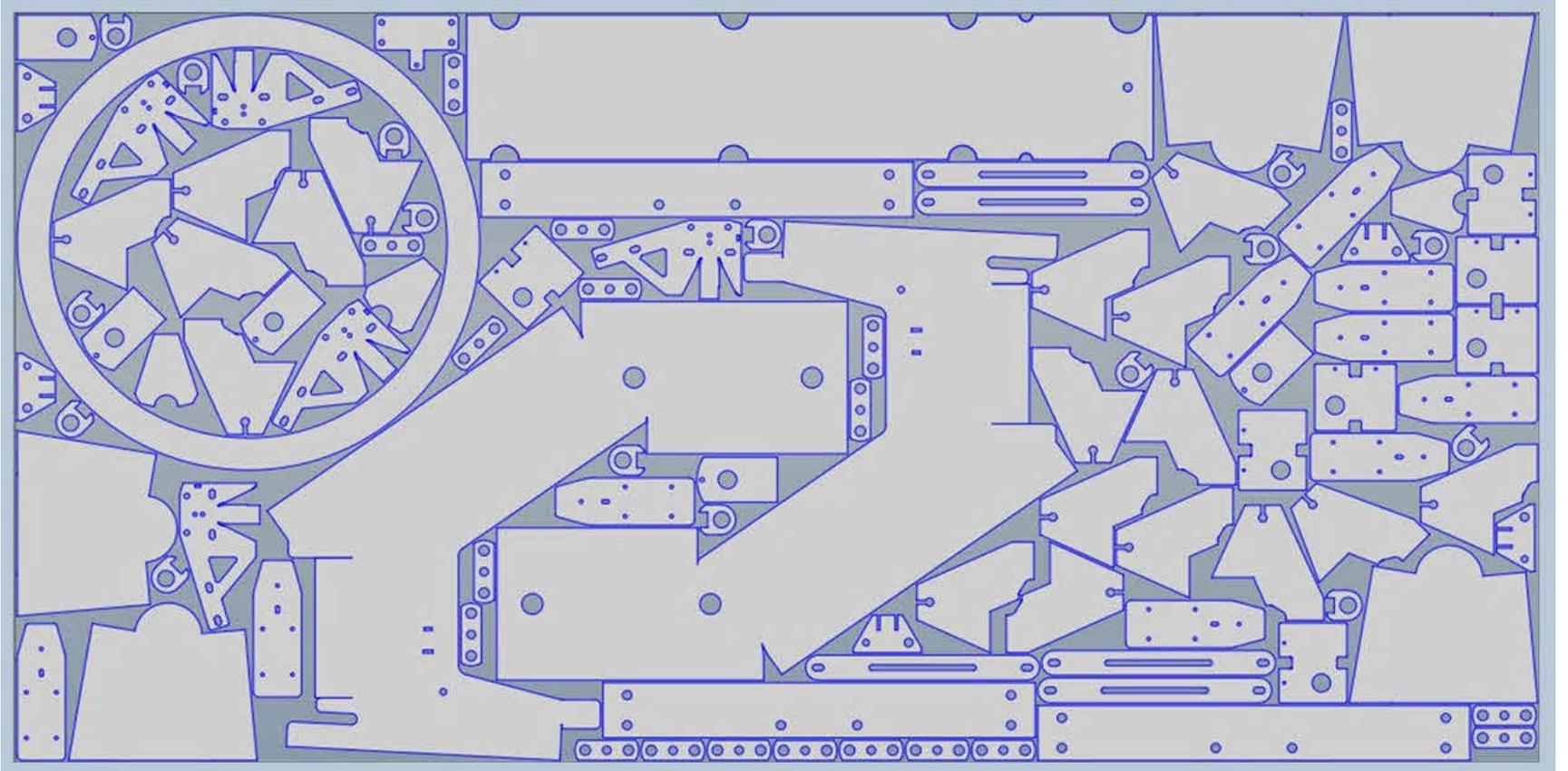


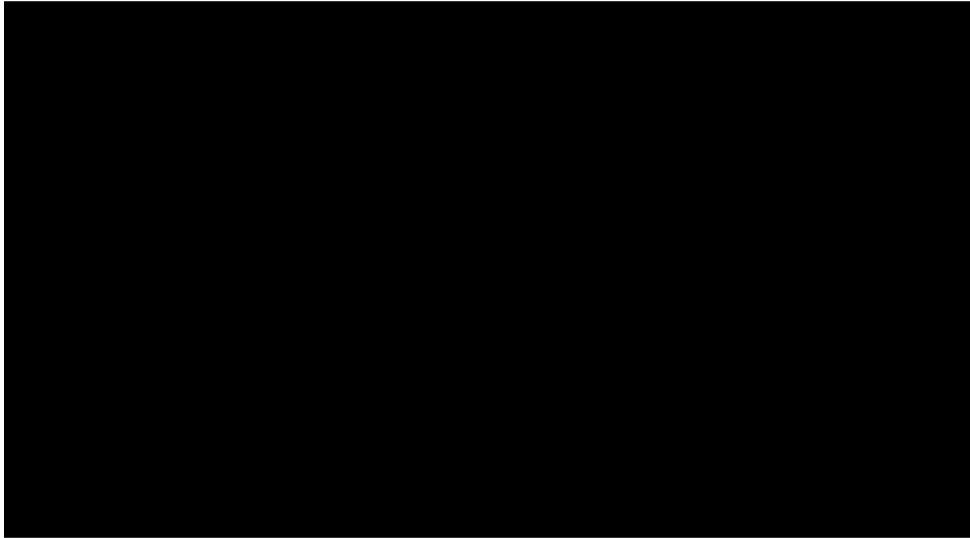




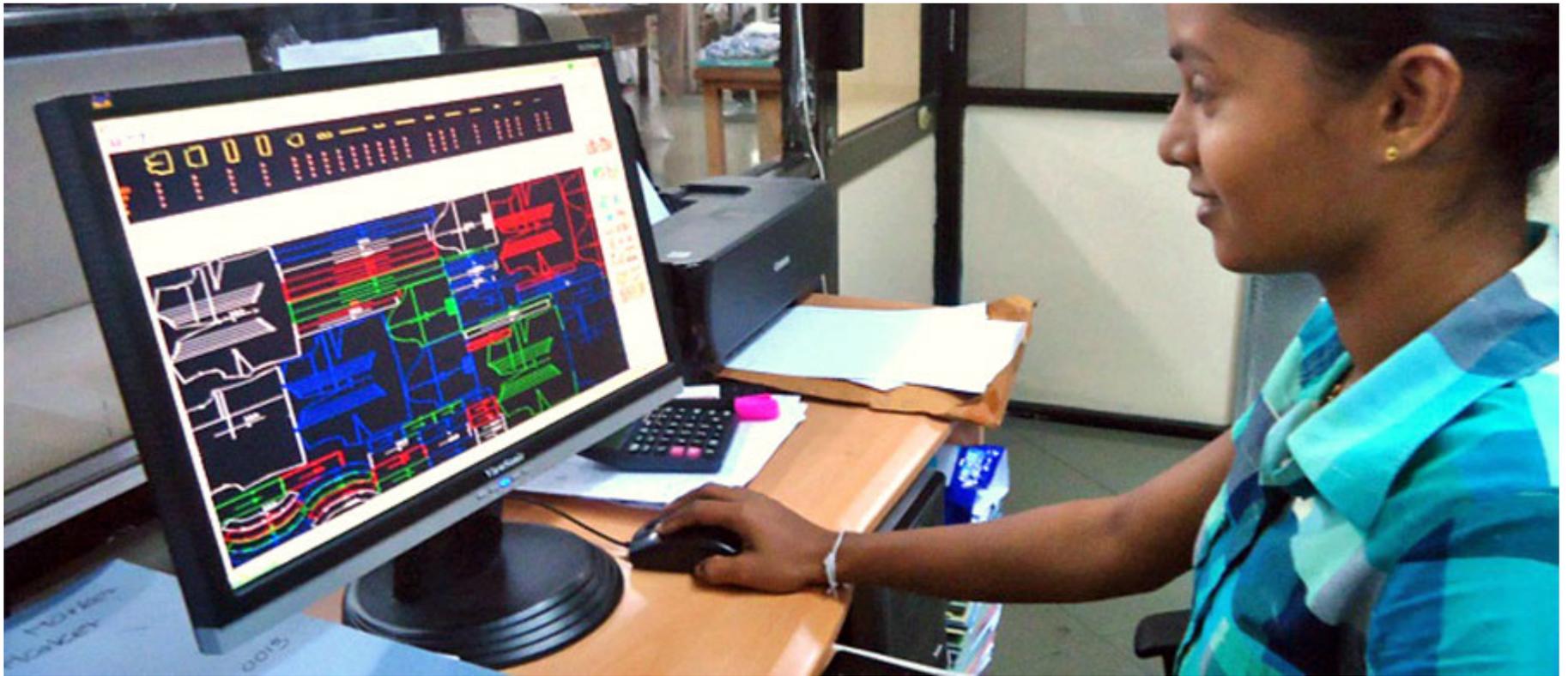
NA PRÁTICA...





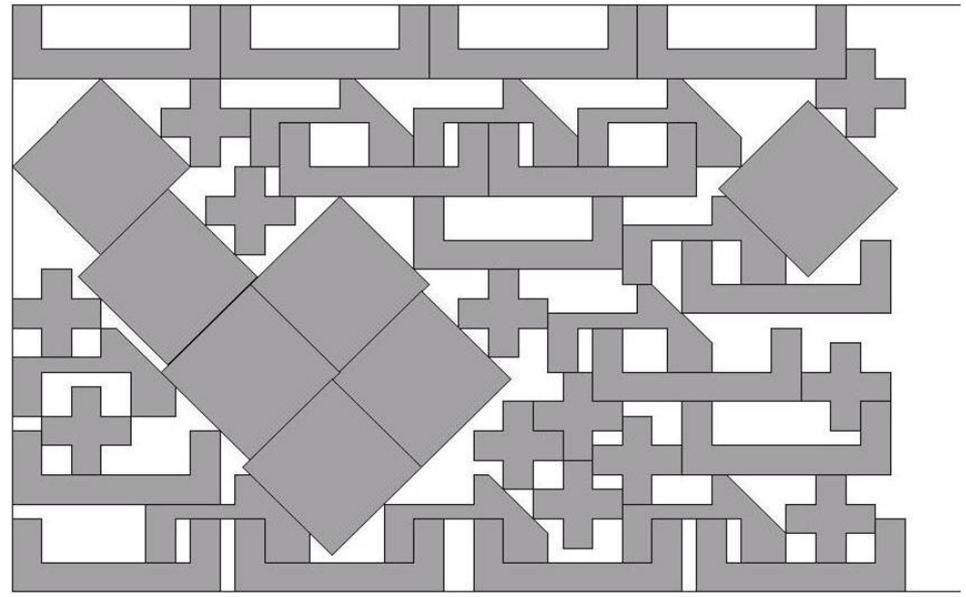
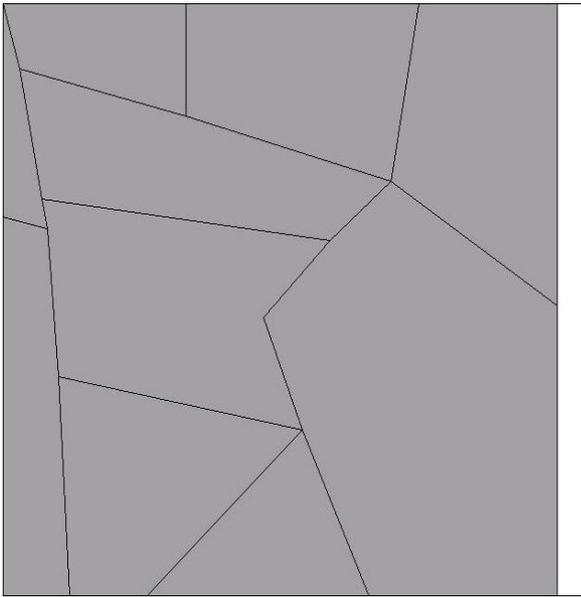






Yes, we can do it better!

1. Quando é que gente sabe que as peças estão “bem” encaixadas?
2. Como implementar isso num computador?

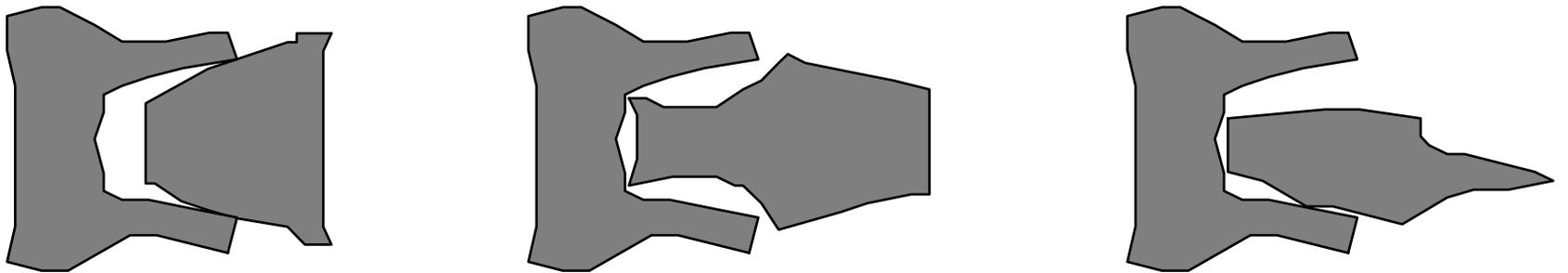


Medida local:

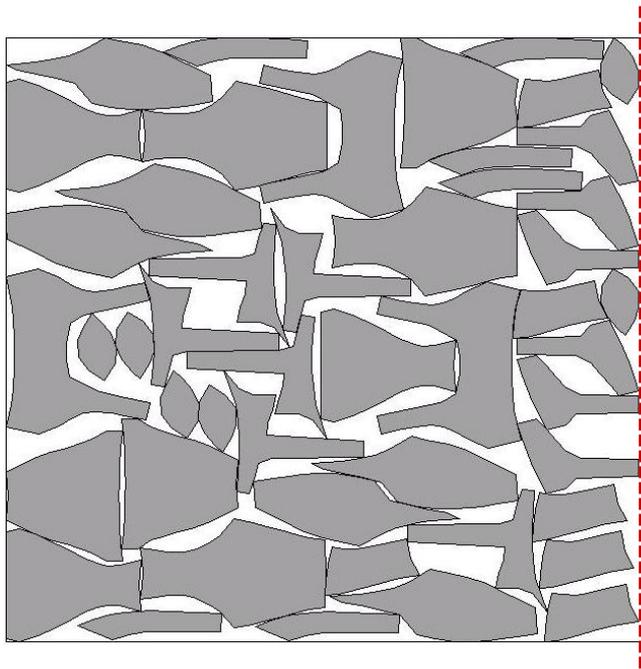
Cada peça encaixa na outra com perda nula



Quando não há solução com perda nula,

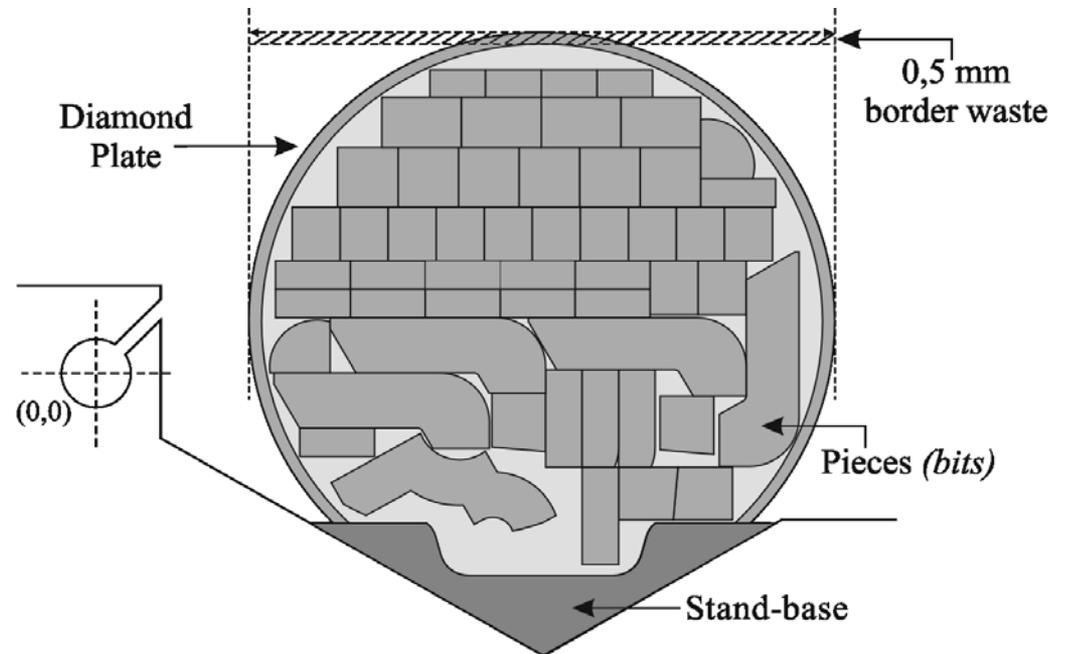


é precisa uma medida global



Encaixar todas as peças
no menor
comprimento possível

Encaixar o maior
número de peças
possível dentro do
espaço disponível

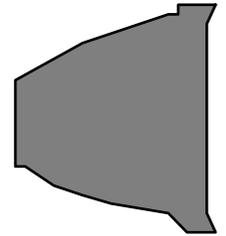
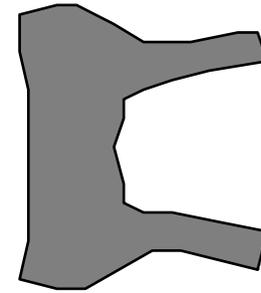
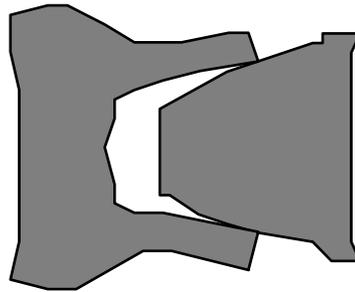
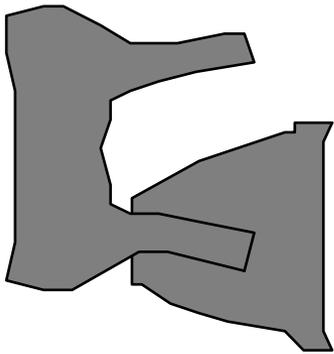


Implementar num computador

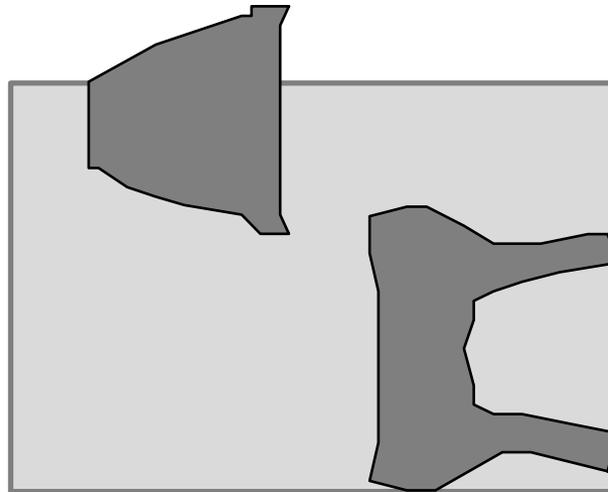
LIDANDO COM A GEOMETRIA

1. Geometria computacional

Em contacto?

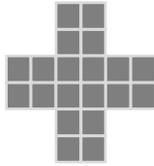


Sobreposição?



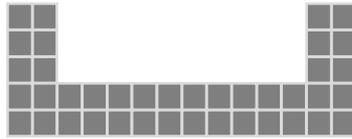
Afastado?

Contido?

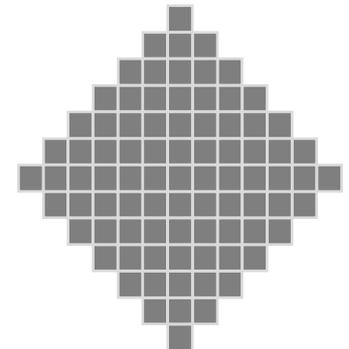
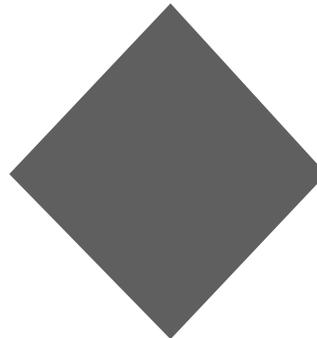
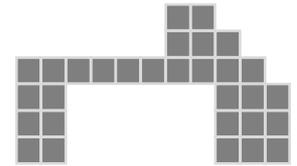


0	0	1	1	0	0
0	0	1	1	0	0
1	1	1	1	1	1
1	1	1	1	1	1
0	0	1	1	0	0
0	0	1	1	0	0

Rasterizar
e somar
matrizes

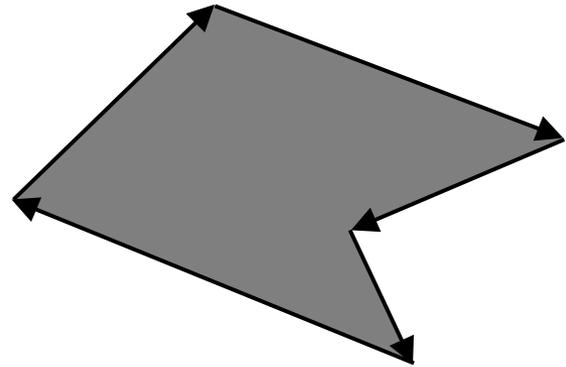


mas...



Representação poligonal

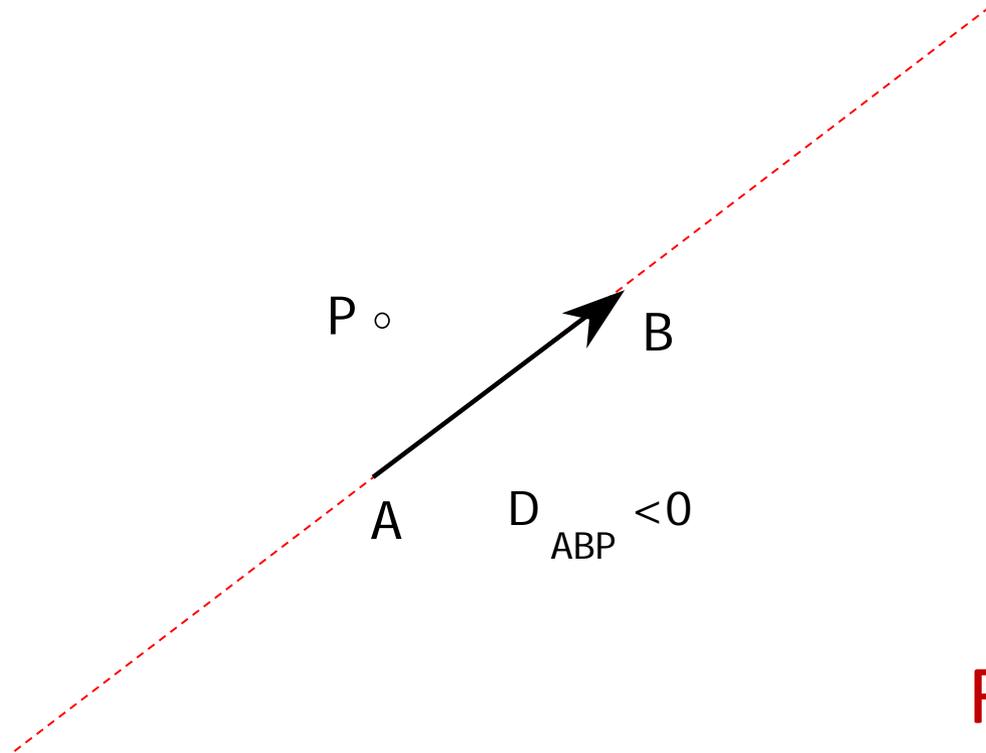
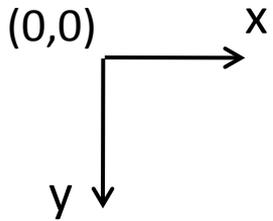
- As peças são representadas como polígonos com arestas orientadas



- A orientação das arestas é tal que o exterior do polígono está sempre do lado direito da aresta

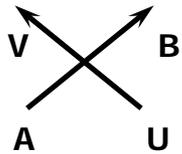
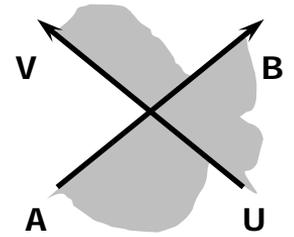
Trigonometria...

$$D_{ABP} = \text{sign}((x_A - x_B) \cdot (y_A - y_P) - (y_A - y_B) \cdot (x_A - x_P))$$

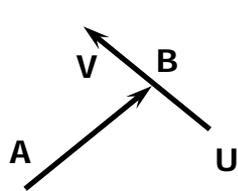


Funções-D

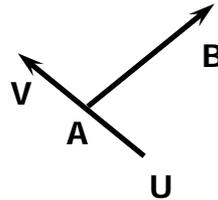
Trigonometria...



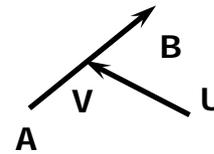
(a)



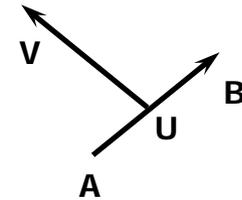
(b)



(c)



(d)



(e)

...

$\left\{ \begin{array}{l} D_{ABU} \neq 0 \\ D_{ABV} \neq 0 \\ D_{ABU} \neq D_{ABV} \\ D_{UVA} \neq 0 \\ D_{UVB} \neq 0 \\ D_{UVA} \neq D_{UVB} \end{array} \right.$	$\left\{ \begin{array}{l} D_{ABU} \neq 0 \\ D_{ABV} \neq 0 \\ D_{ABU} \neq D_{ABV} \\ D_{UVA} \neq 0 \\ D_{UVB} = 0 \end{array} \right.$	$\left\{ \begin{array}{l} D_{ABU} \neq 0 \\ D_{ABV} \neq 0 \\ D_{ABU} \neq D_{ABV} \\ D_{UVA} = 0 \\ D_{UVB} \neq 0 \end{array} \right.$	$\left\{ \begin{array}{l} D_{ABU} \neq 0 \\ D_{ABV} = 0 \\ D_{UVA} \neq 0 \\ D_{UVB} \neq 0 \\ D_{UVA} \neq D_{UVB} \end{array} \right.$	$\left\{ \begin{array}{l} D_{ABU} = 0 \\ D_{ABV} \neq 0 \\ D_{UVA} \neq 0 \\ D_{UVB} \neq 0 \\ D_{UVA} \neq D_{UVB} \end{array} \right.$
$D_{UVA} < 0$	$D_{UVB} > 0$	$D_{ABU} > 0$	$D_{ABV} < 0$	

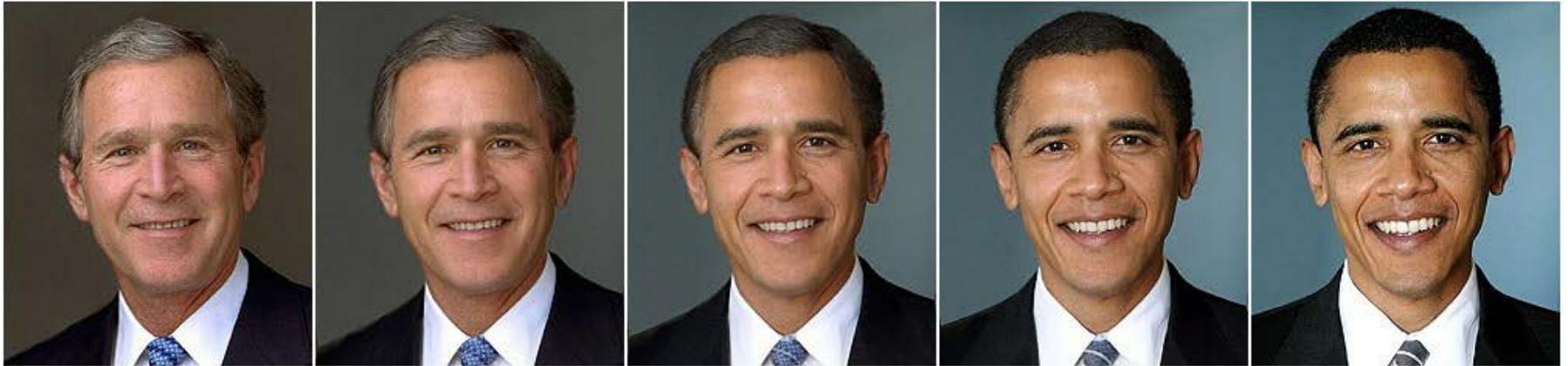
Computacionalmente pesado pois
todo o cálculo é feito em *runtime* para
todos os pares de arestas de todos os
pares de polígonos



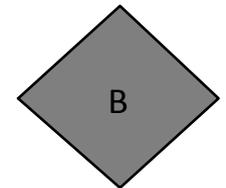
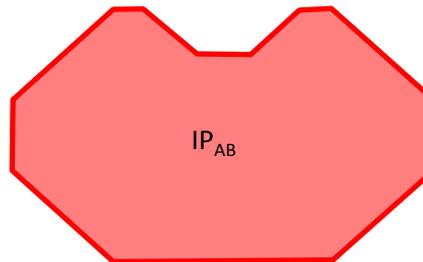
Quem é?



O invólucro de posicionamento e o *morphing*

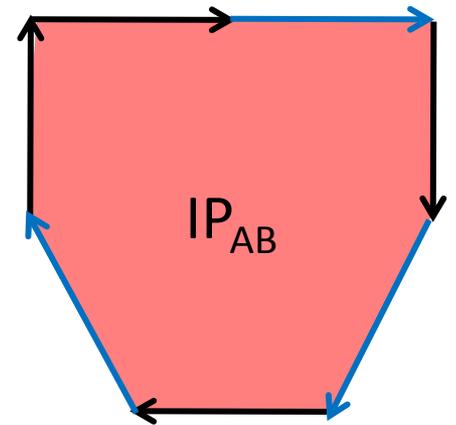
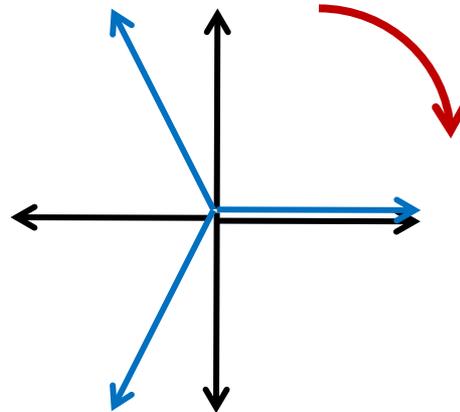
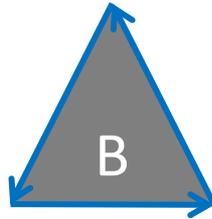
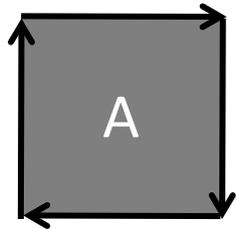


<http://paulbakaus.com/2009/10/08/underestimated-ui-techniques-morphing/>

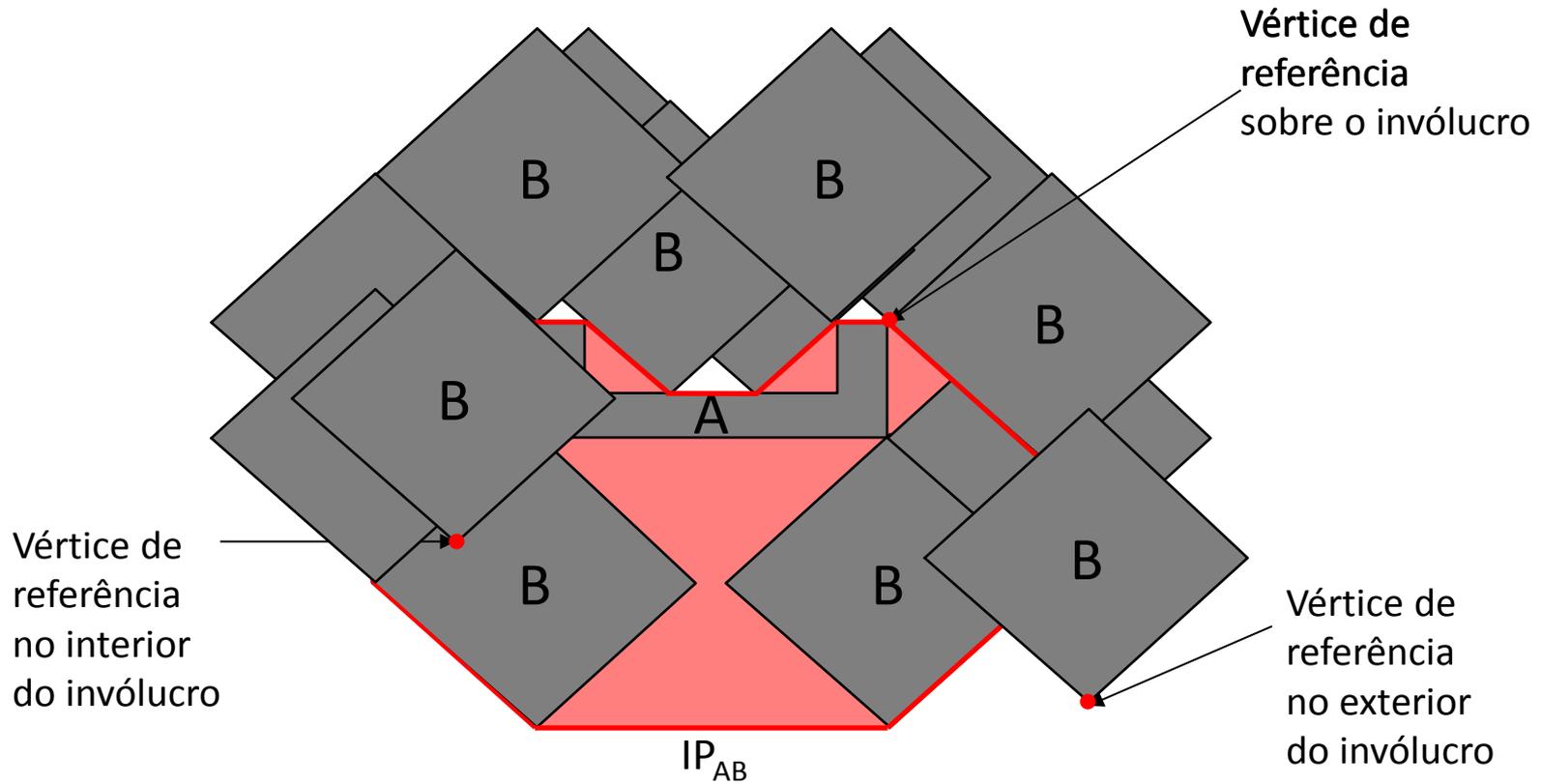


O invólucro de posicionamento contém toda a informação geométrica dos dois polígonos que lhe dão origem.

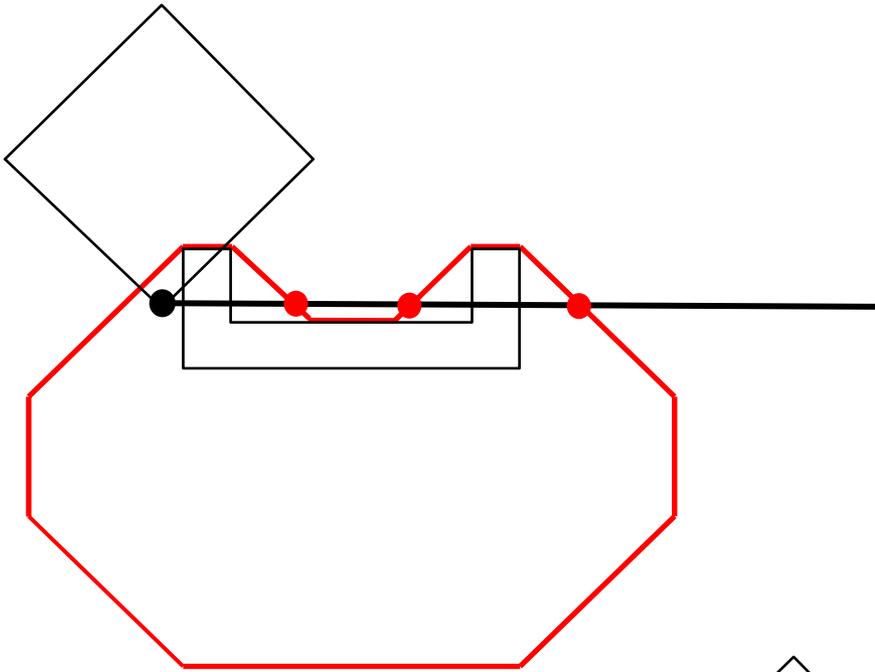
Cálculo do invólucro de posicionamento: somadas de Minkowski



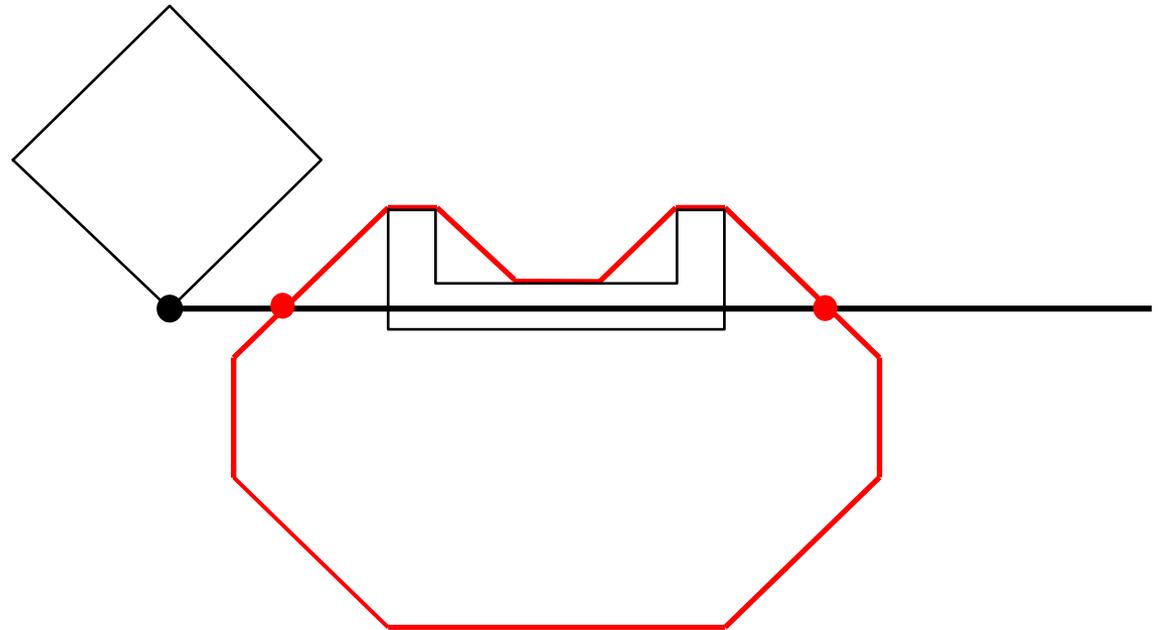
Cálculo do invólucro de posicionamento: algoritmo de deslizamento



O cálculo *online*
é simplificado

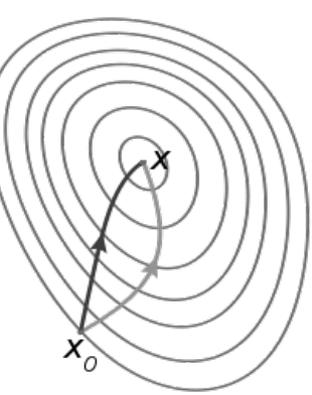


Ponto dentro
de polígono?

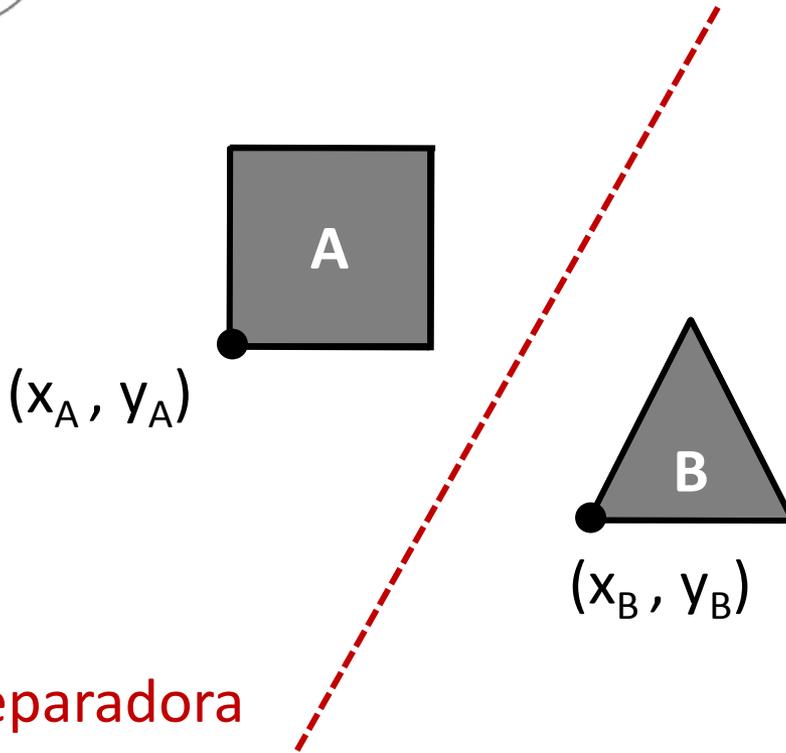


Implementar num computador

**PROCURANDO UMA
SOLUÇÃO COM UM BOM
ENCAIXE**

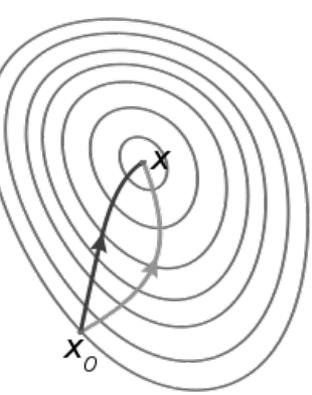


2. Otimização matemática

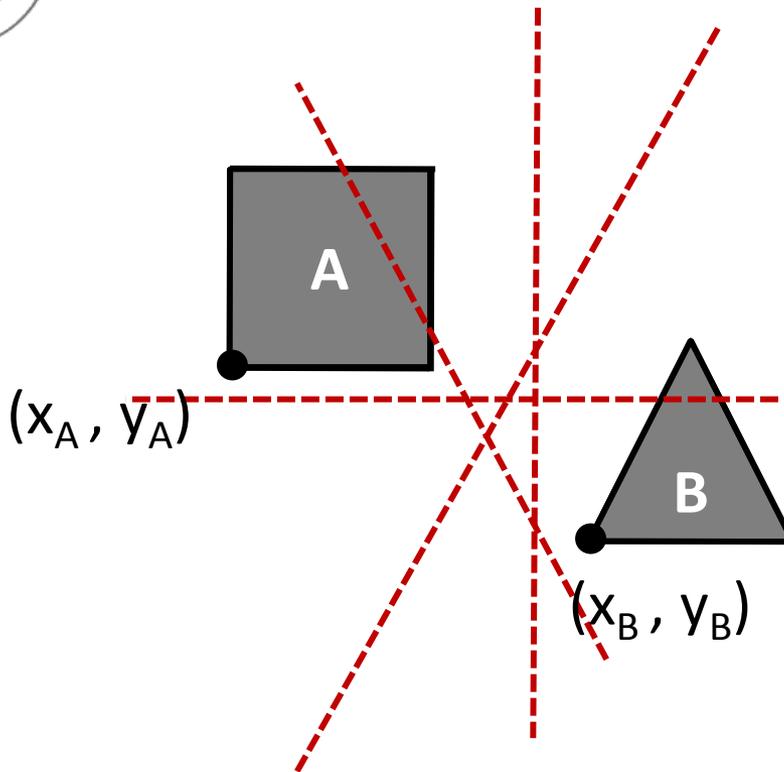


Todos os vértices de **A** para um lado da reta e todos os vértices de **B** para o outro lado da reta: 7 inequações ou condições.

$$D_{ABP} = \text{sign}((x_A - x_B) \cdot (y_A - y_P) - (y_A - y_B) \cdot (x_A - x_P))$$



2. Otimização matemática

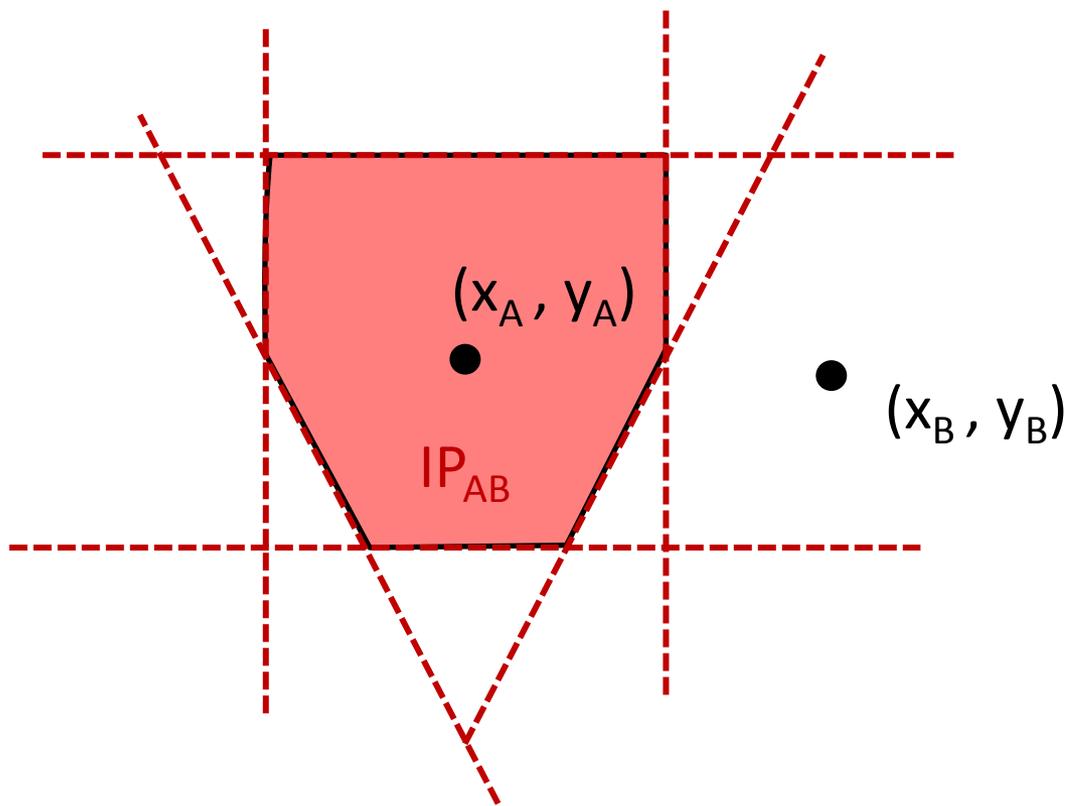


Todos os vértices de **A** para um lado da reta e todos os vértices de **B** para o outro lado da reta: 7 inequações.

Mas que reta testar?

Prova-se que, se existir, há sempre uma reta separadora paralela a uma das arestas de um dos dois polígonos.

Ou usando o invólucro de posicionamento?



Testar todas as retas a
ver se pelo menos uma
verifica



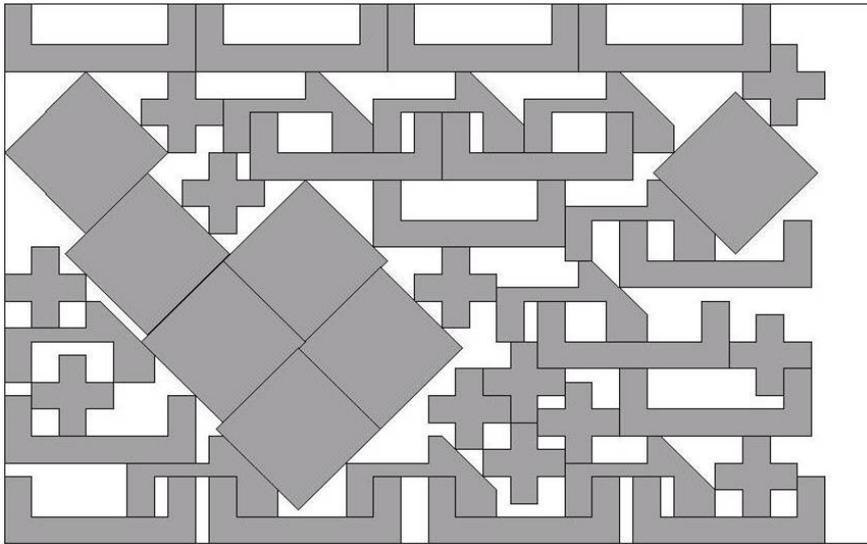
**Problema
combinatório**

Problemas combinatórios...

Enumerar todas as
possibilidades



Problemas combinatórios...



43 polígonos

343 vértices

2×10^{722} possibilidades

Idade do sistema solar:

$1,45 \times 10^{17}$ segundos

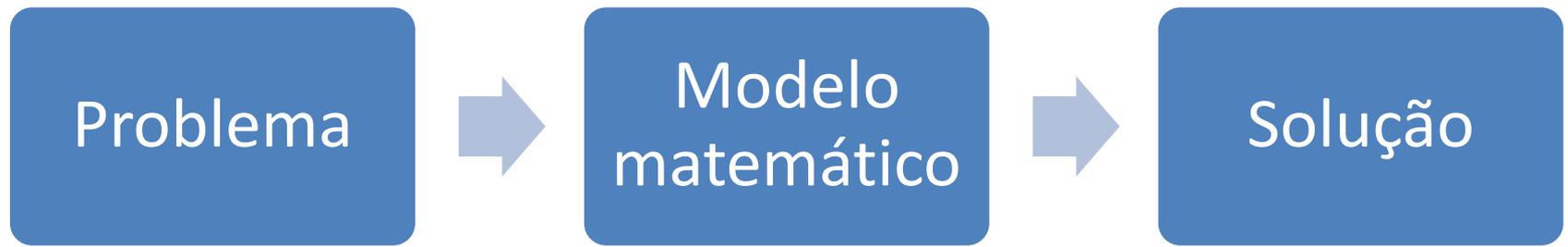
E este é um problema bem pequeno quando comparado com os problemas reais....



Otimização matemática: enumeração **implícita** de soluções factíveis, guiada por um objetivo

Modelação

Solver



Dados

Representação
geométrica

Objetivo

Condições a impor
(restrições)

Diferentes
modelos

Diferentes
solvers

Otimização matemática: enumeração implícita de soluções factíveis, guiada por um objetivo

Objetivo:

minimizar (gastar o menos possível) o comprimento do recipiente utilizado para encaixar toda as peças.

Restrições:

todas as peças estão contidas dentro do recipiente
nenhuma peça sobrepõe outra
todas a peças são colocadas

Ou seja:

Otimização matemática: enumeração implícita de soluções factíveis, guiada por um objetivo

Min L

s.t. $x_i \leq L - l_i \quad i = 1, \dots, N$

$y_i \leq W - w_i \quad i = 1, \dots, N$

Peças dentro do recipiente

$$\alpha_{ijk}(x_j - x_i) + \beta_{ijk}(y_j - y_i) \leq q_{ijk} + M(1 - v_{ijk}) \quad 1 \leq i < j \leq N$$
$$k = 1, \dots, m_{ij}$$

Peças
não se
sobrepoem

$$\sum_{k=1}^{m_{ij}} v_{ijk} \geq 1 \quad 1 \leq i < j \leq N$$

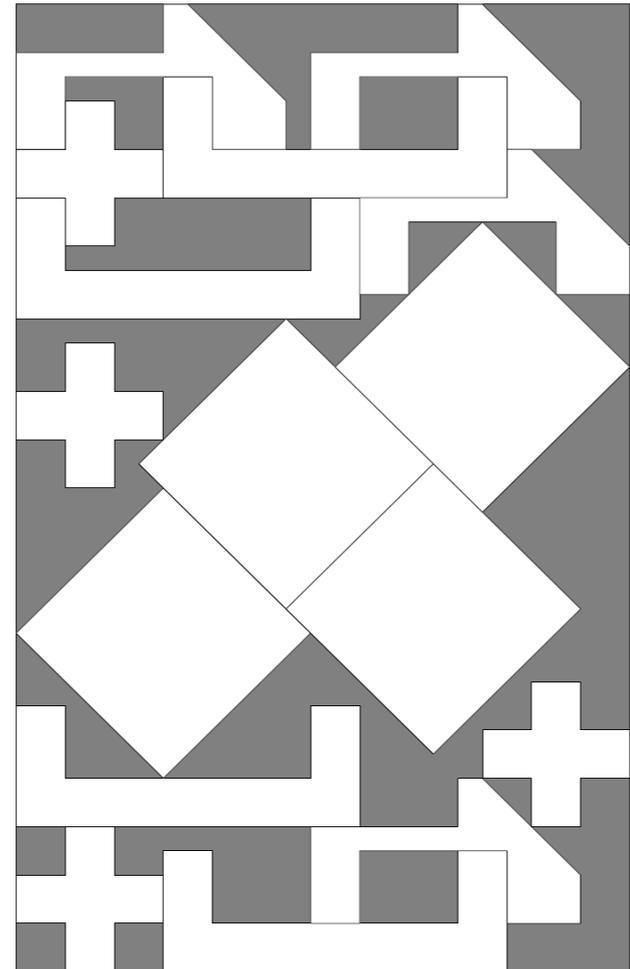
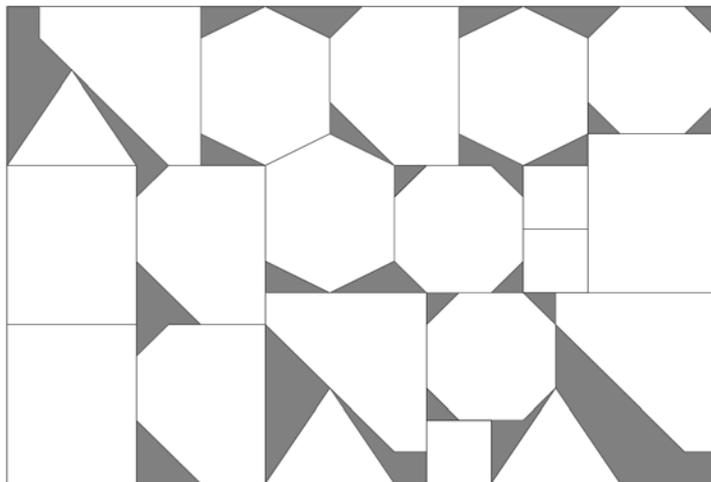
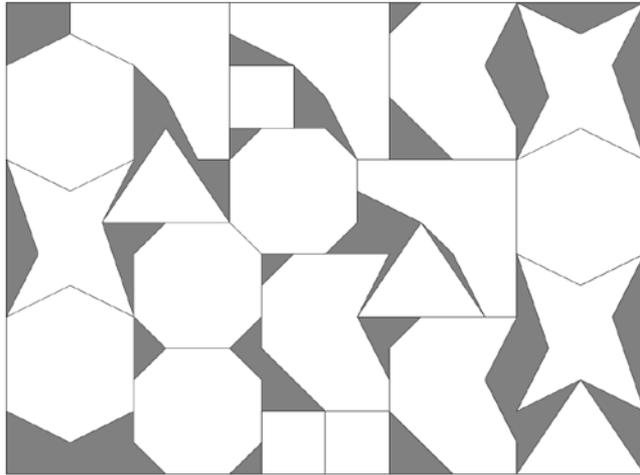
$$v_{ijk} \in \{0, 1\} \quad 1 \leq i < j \leq N$$

Que reta separadora é escolhida

$$x_i, y_i \geq 0 \quad 1 \leq i \leq N$$

Onde ficam colocadas as peças

Soluções ótimas* que se conseguem atualmente achar



* Solução que se consegue provar não haver outra com melhor valor de objetivo do que ela

3. Otimização algorítmica



Heurísticas

Heurística é um método ou processo criado com o objetivo de encontrar soluções para um problema. É um procedimento simplificador (embora não simplista) que, em face de questões difíceis envolve a substituição destas por outras de resolução mais fácil a fim de encontrar respostas viáveis, ainda que imperfeitas.

3. Otimização algorítmica



Uma **heurística**:

Determina uma solução para o problema rapidamente

(quando comparado com os algoritmos de otimização)

Não encontra a solução ótima ou, se a encontra, não é capaz de provar que esta é ótima.

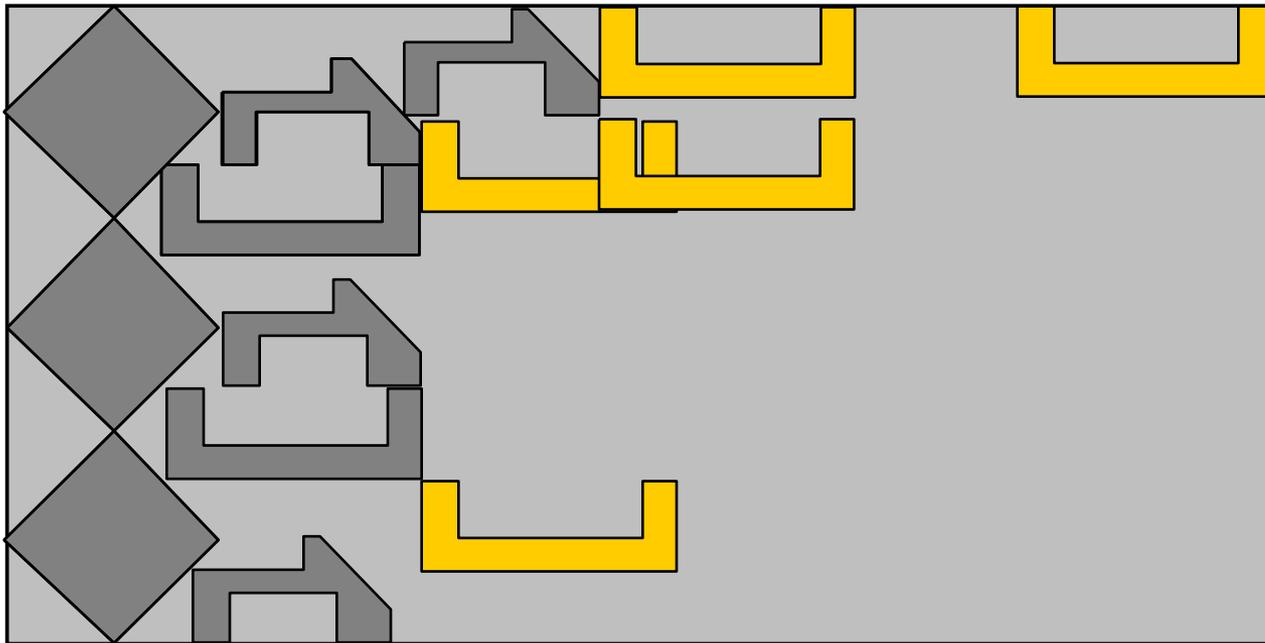
Rápida, “simples” e flexível.

Heurísticas construtivas

Construir uma solução peça a peça

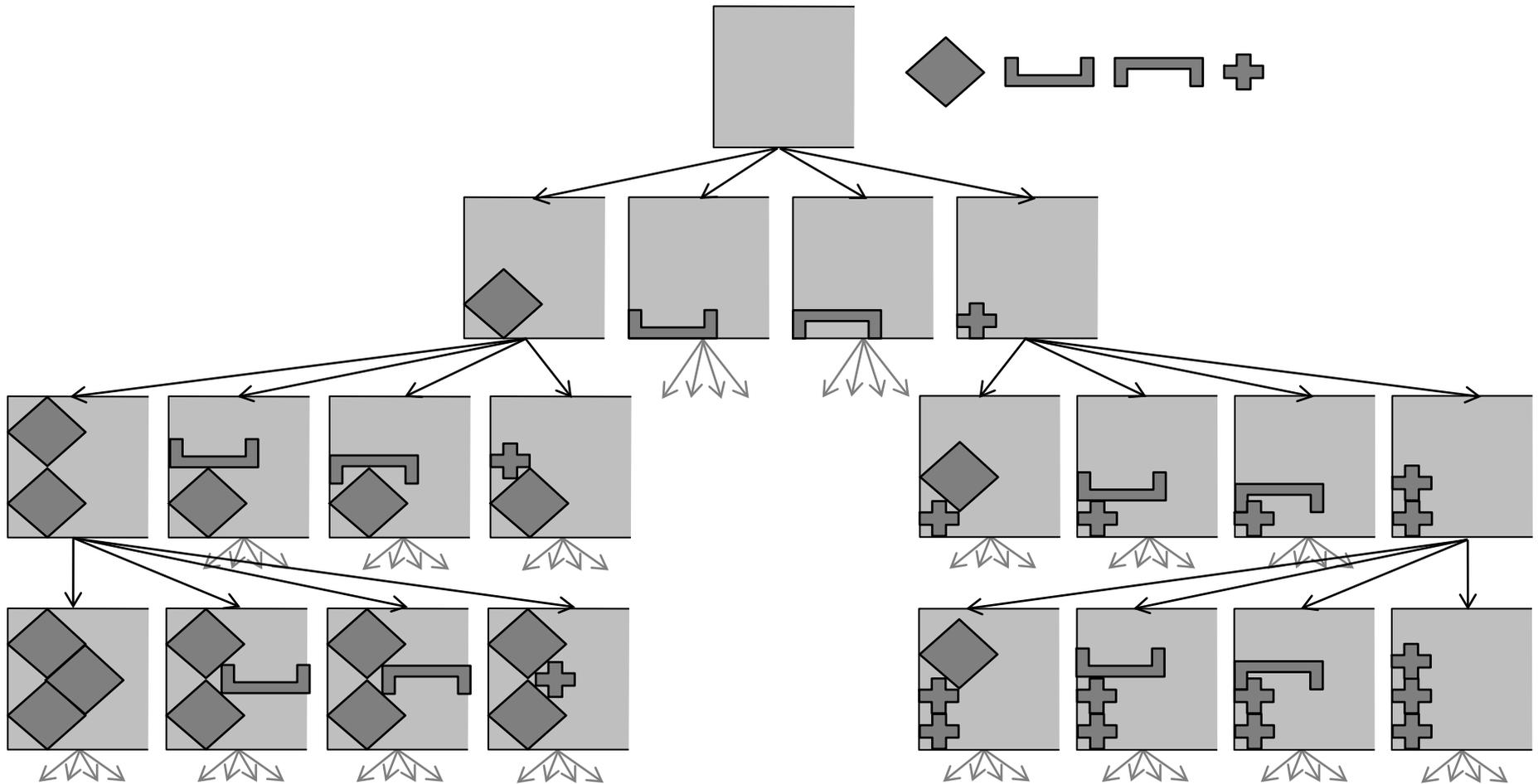


Heurísticas construtivas



Tentando todas as sequências

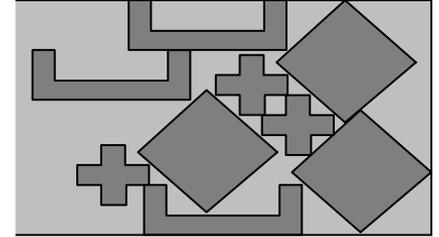
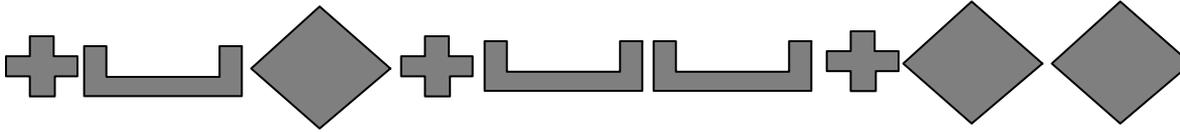
Busca em árvore



Buscando algumas seqüências

Busca local

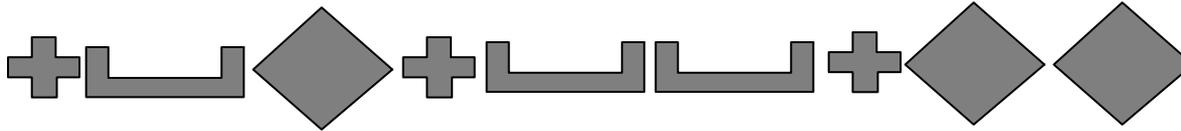
Seqüência original



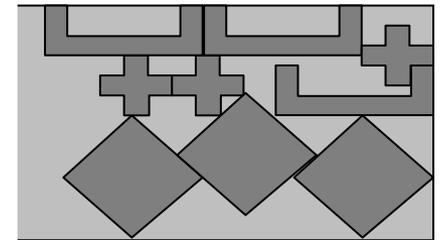
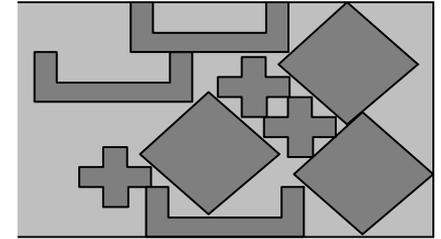
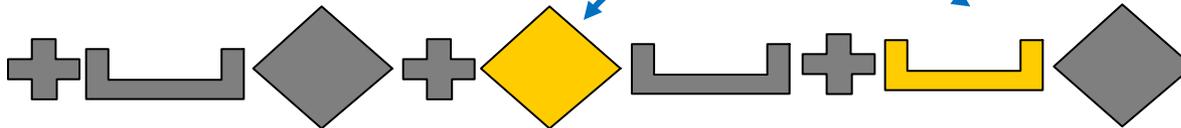
Buscando algumas seqüências

Busca local

Seqüência original



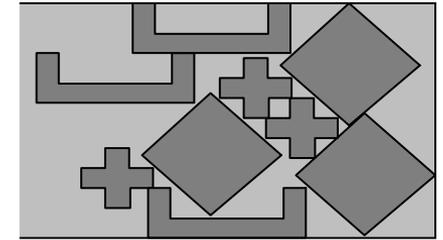
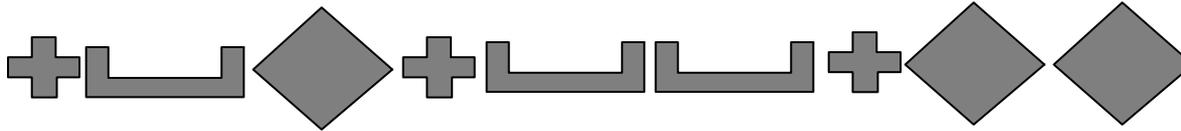
Movimento de troca



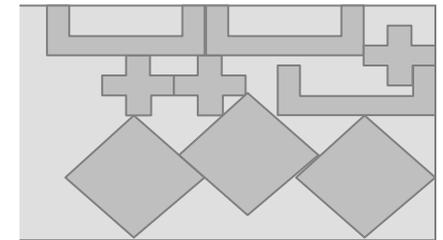
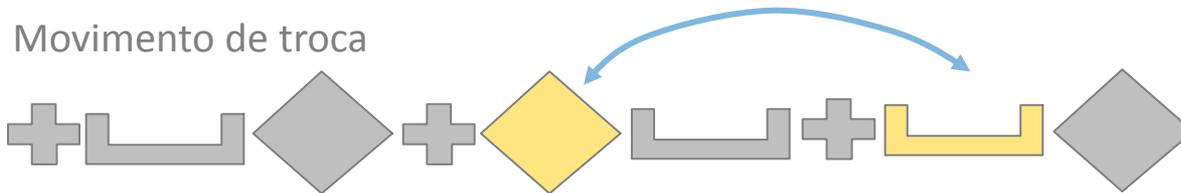
Buscando algumas seqüências

Busca local

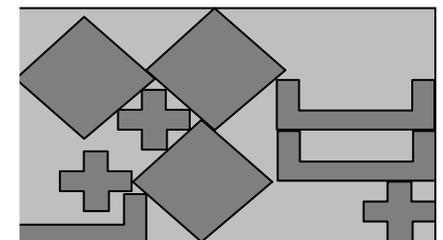
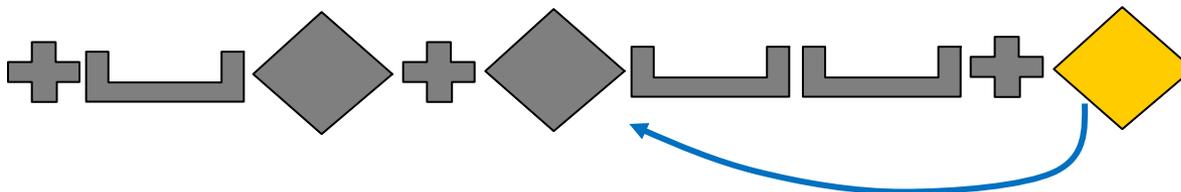
Seqüência original



Movimento de troca



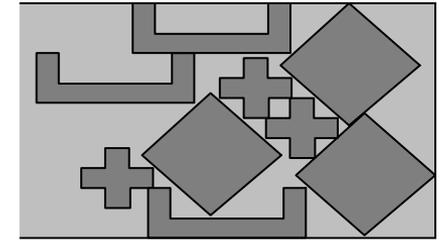
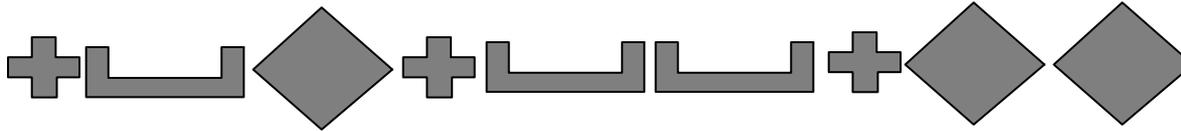
Movimento de inserção



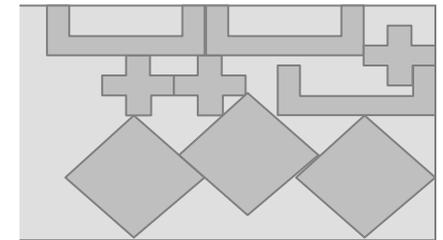
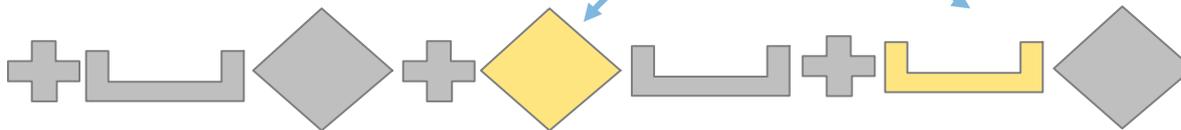
Buscando algumas seqüências

Busca local

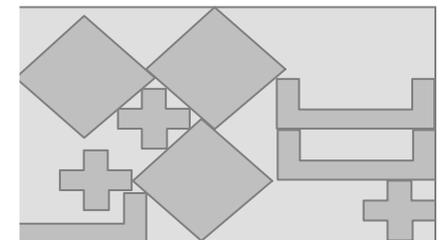
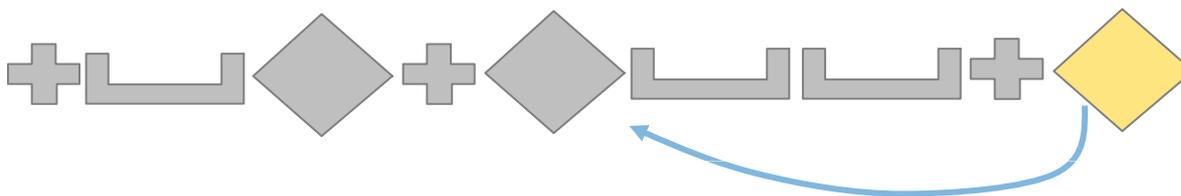
Seqüência original



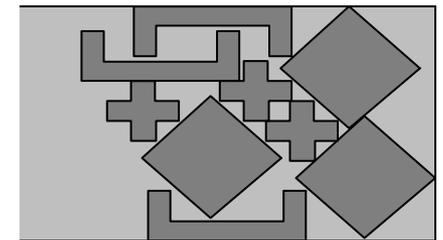
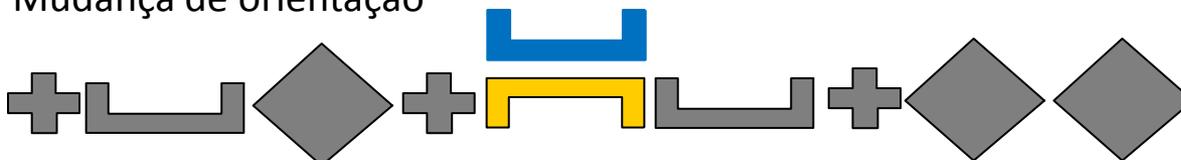
Movimento de troca



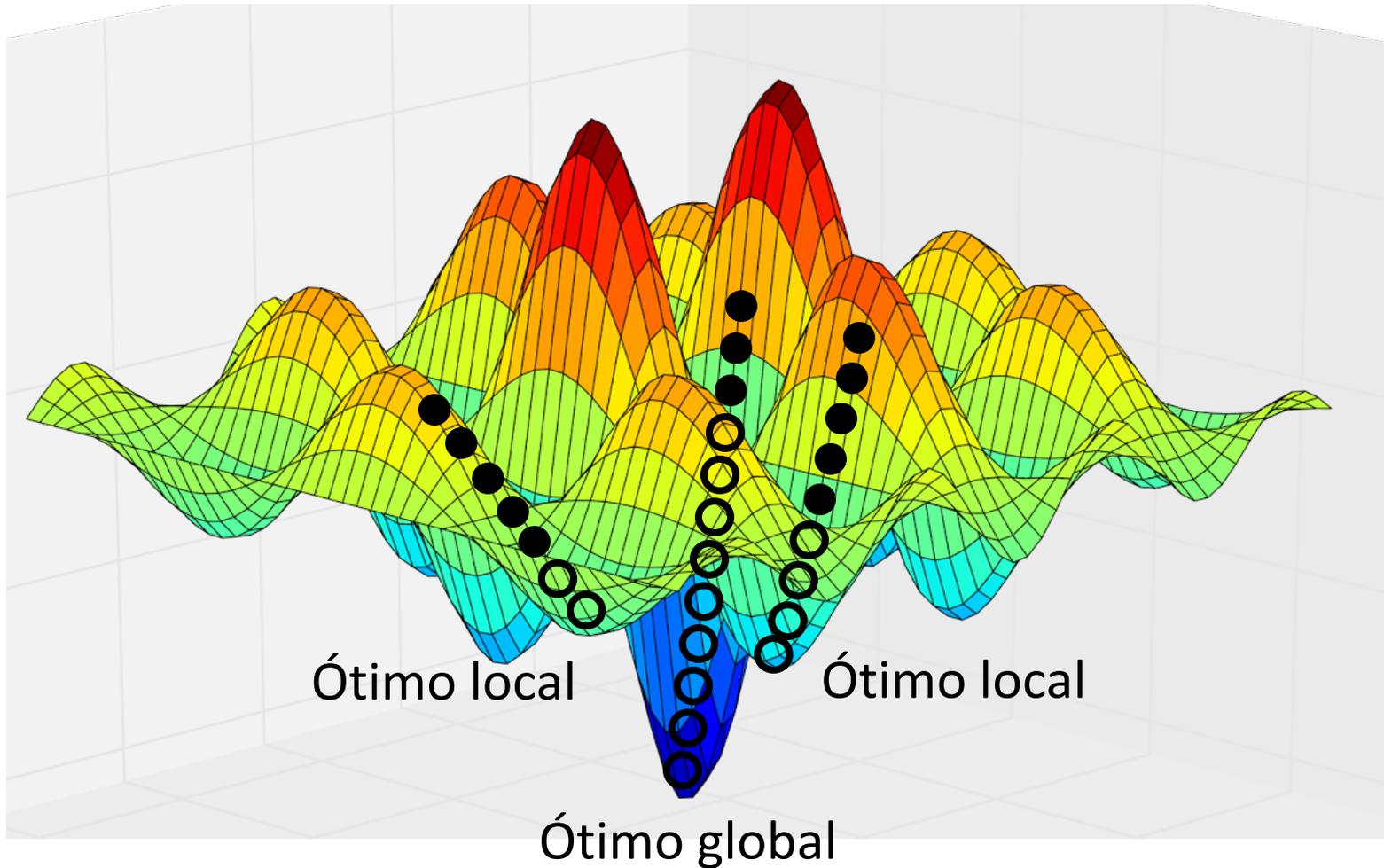
Movimento de inserção



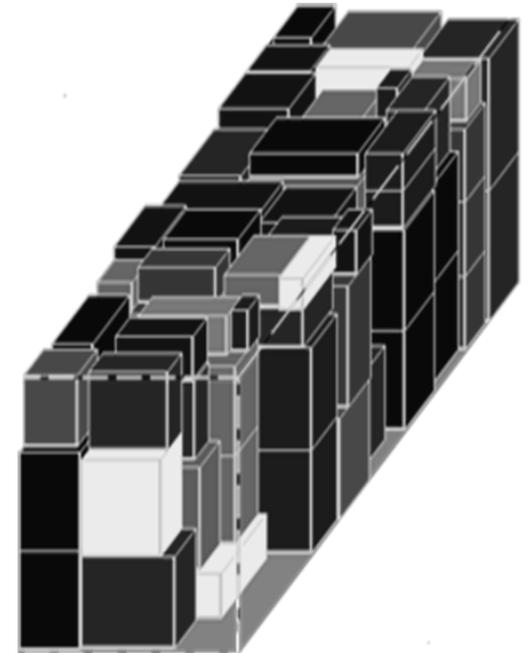
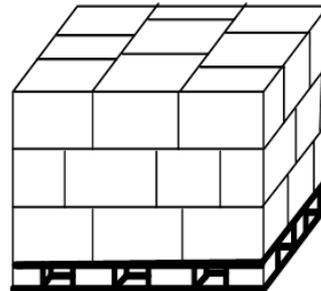
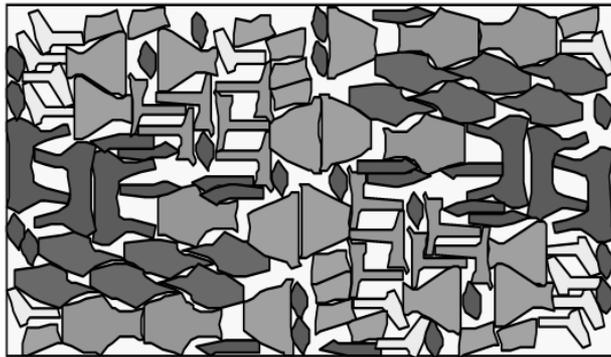
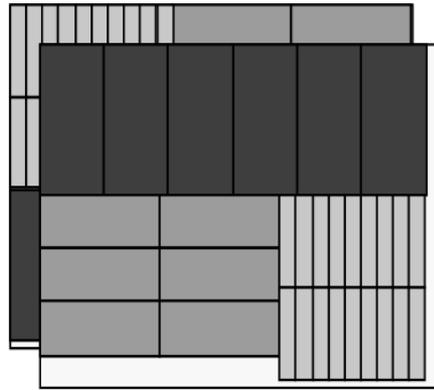
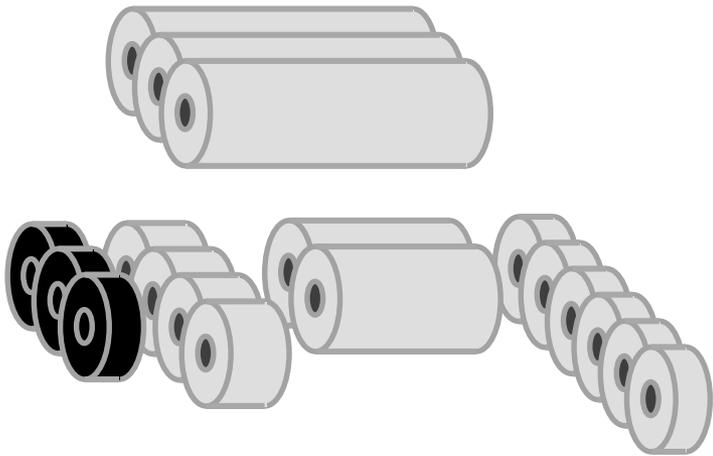
Mudança de orientação



Ótimo local e ótimo global



Problemas de cortes e empacotamentos



Problemas de cortes e empacotamentos

Objetos “pequenos” têm que ser afetados a objetos “grandes”

- problema **combinatório**
- sob restrições geométricas (não sobreposição ...)
 - problema **geométrico**
- com o objetivo de minimizar o desperdício + ...

Problemas de cortes e empacotamentos

- Menores custos de produção
- Menor desperdício de matérias-primas
 - Menor consumo de recursos naturais
 - Floresta, minérios, animais,...
 - Menor poluição ambiental
 - Desperdícios
 - Transportes
- Menor consumo de energia



Corte de rolos de papel



Corte de painéis de madeira rectangulares



Carregamento de paletes



Empacotamento de contentores



Corte de pele irregular



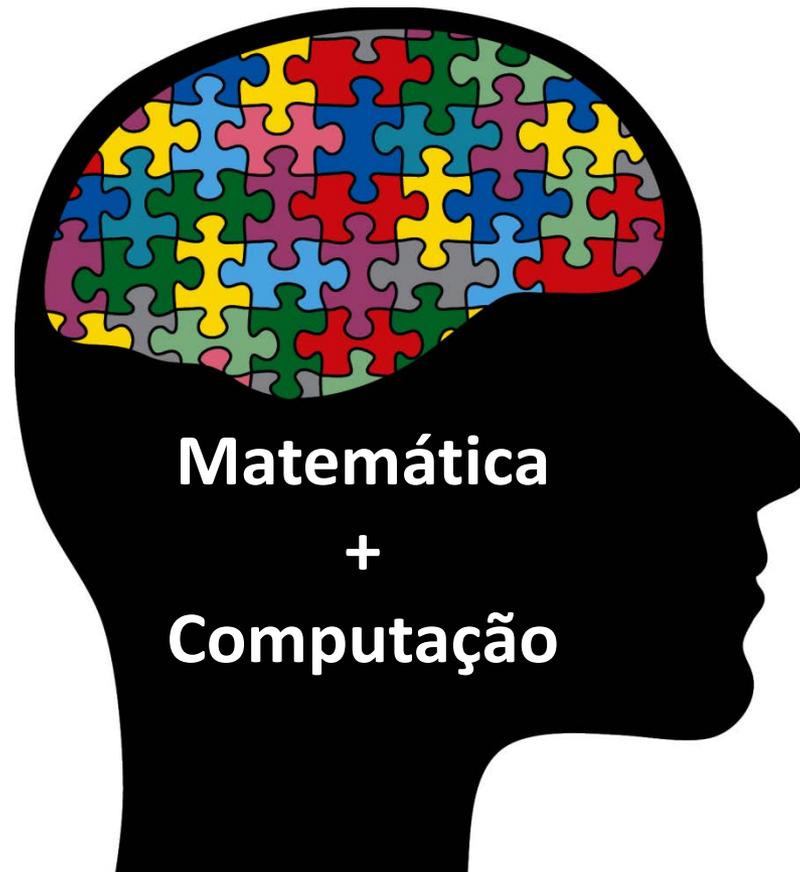
Como resolver quebra-cabeças em que as peças não encaixam certinho



Como resolver quebra-cabeças em que as peças não encaixam certinho



Como resolver quebra-cabeças em que
as peças não encaixam certinho





Obrigado pela atenção