UNIVERSIDADE FEDERAL DE SANTA CATARINA
CENTRO TECNOLÓGICO
DEPARTAMENTO DE INFORMÁTICA E ESTATÍSTICA
CIÊNCIAS DA COMPUTAÇÃO

Gabriel Medeiros Lopes Carneiro

Problema de empacotamento de retângulos:
avaliação de métodos de solução baseados em *bottom-left*

Florianópolis, SC
2023

Gabriel Medeiros Lopes Carneiro

Problema de empacotamento de retângulos: avaliação de métodos de solução baseados em *bottom-left*

Trabalho de conclusão de curso submetido ao curso de Ciências da Computação da Universidade Federal de Santa Catarina para a obtenção do título de Bacharel em Ciências da Computação.

Universidade Federal de Santa Catarina
Centro Tecnológico
Departamento de Informática e Estatística
Ciências da Computação

Orientador: Prof. Dr. Pedro Belin Castellucci
Coorientador: Prof. Dr. Rafael de Santiago

Florianópolis, SC
2023

Ficha de identificação da obra elaborada pelo autor,
através do Programa de Geração Automática da Biblioteca Universitária da UFSC.

Carneiro, Gabriel Medeiros Lopes
Problema de empacotamento de retângulos : avaliação de métodos de solução baseados em bottom-left / Gabriel Medeiros Lopes Carneiro ; orientador, Pedro Belin Castellucci, coorientador, Rafael de Santiago, 2023.
111 p.

Trabalho de Conclusão de Curso (graduação) - Universidade Federal de Santa Catarina, Centro Tecnológico, Graduação em Ciências da Computação, Florianópolis, 2023.

Inclui referências.

1. Ciências da Computação. 2. problema de empacotamento. 3. bottom-left. 4. heurística. 5. pesquisa operacional. I. Castellucci, Pedro Belin. II. Santiago, Rafael de. III. Universidade Federal de Santa Catarina. Graduação em Ciências da Computação. IV. Título.

Resumo

Problemas de empacotamento consistem em alocar um conjunto de itens $\mathcal{I}$ em uma caixa $\mathcal{B}$. No problema de empacotamento da mochila, foco deste trabalho, cada item é associado a um valor e busca-se uma solução que maximize a soma dos valores dos itens alocados. Este trabalho compara 40 métodos de solução criados com base na heurística construtiva *bottom-left* para o problema de empacotamento de retângulos. A escolha dessa heurística se deve a sua simplicidade e a dificuldade de usar métodos exatos para resolução do problema em tempo hábil. Os métodos criados são uma combinação de diferentes formas de ordenação dos itens e criação de regiões, as quais evitam as sobreposições e o domínio contínuo presentes no problema. Algoritmos foram implementados em Python e testados com instâncias da literatura, dados como qualidade de solução, porcentagem de itens alocados e tempo de execução foram coletados. O principal resultado foi a alta competitividade de diferentes modos de ordenação, não sendo a área a única relevante, com o perímetro obtendo os melhores resultados.

Palavras-chave: problema de empacotamento, *bottom-left*, heurística, pesquisa operacional.

Abstract

Packing problems consist of allocating a set of items $\mathcal{I}$ into a box $\mathcal{B}$. In the knapsack packing problem, the focus of this work, each item is associated with a value and a solution is sought that maximizes the sum of the values of the allocated items. This work compare 40 created solution methods based on *bottom-left* constructive heuristic for the rectangle packing problem. The choice of this heuristic is due to its simplicity and the difficulty of using exact methods to solve the problem in a timely manner. The methods created are a combination of different ways of ordering items and creating regions, which avoid superposition and continuous domain present in the problem. Algorithms were implemented in Python and tested with instances from the literature, data such as solution quality, percentage of allocated items and execution time were collected. The main result was the high competitiveness of different ordering modes, the area not being the only relevant one, with the perimeter obtaining the best results.

Keywords: packing problem, *bottom-left*, heuristic, operational research.

Lista de ilustrações

Lista de tabelas

Sumário

Introdução

Serviços de loja *online* com entrega como Amazon e Mercado Livre estão tornando-se cada vez mais presentes no dia a dia. Para tornar as entregas mais rápidas é necessário fazer uma série de estudos de logística e planejamento sobre como organizar os produtos nos estoques e nos veículos de entrega (SILVA et al., 2022; MORABITO NETO et al., 1992), muitas vezes sendo necessário considerar a ordem em que eles precisarão ser retirados. Além de tornar o processo mais rápido, a organização também pode permitir o melhor uso de espaços, aumentando a quantidade máxima de itens ou evitando o desperdício dos espaços.

Ainda sobre evitar desperdícios, esse quesito é muito importante para as indústrias de papel, móveis, têxtil e metal-mecânica (QUEIROZ, 2022; CAVALI, 2004; BELLUZZO et al., 2005). Todas essas áreas querem gerar o máximo de produtos com o mínimo de recursos materiais utilizados, para evitar o descarte desnecessário do material e prejuízos financeiros.

Os problemas citados são considerados problemas de corte e empacotamento. Problemas de corte envolvem cortar um objeto, como blocos de gesso, chapas de aço e barras de ferro, em itens menores. Enquanto problemas de empacotamento tratam sobre alocar um conjunto de itens $\mathcal{I}$ em um recipiente $\mathcal{B}$. Ambos são equivalentes entre si e é possível separá-los de acordo com sua dimensão.

Problemas unidimensionais podem ser associados ao corte de barras ou canos, para atender uma demanda por peças de diferentes tamanhos, ou ao empilhamento de caixas em busca de uma altura mínima. Outra aplicação viável do caso 1D é o famoso problema da mochila (IORI; DE LIMA et al., 2021), onde se deseja alocar itens de valor v_i e peso w_i em uma mochila com capacidade C, de forma que se maximize a soma dos valores dos itens escolhidos para entrar na mochila.

As indústrias de móveis, tecido, couro e papel usam o caso 2D para minimizar o desperdício ao se cortar suas peças (QUEIROZ, 2022; CAVALI, 2004; BELLUZZO et al., 2005), enquanto os setores de entregas usam para organizar paletes (MORABITO NETO et al., 1992). O caso 3D é facilmente associável ao carregamento de *containers* (MORABITO NETO et al., 1992), onde objetos são geralmente caixas a serem alocadas em algum veículo, ou ao corte de blocos de gesso, já citado.

Basicamente, o problema é aplicável em qualquer área que precise de organização ou logística, bem como em situações que envolvam o corte de algum material. Ao utilizar soluções para resolver problemas de corte e empacotamento, é possível reduzir o desperdício de materiais e impacto ambiental, diminuir tempo de entregas e otimizar espaços de estoque.

O caso 2D, dimensão de estudo deste trabalho, possui uma vasta literatura de métodos

de solução. As abordagens de solução se dividem entre exatas, que buscam a solução ótima do problema, e heurísticas, as quais podem não encontrar uma solução ótima, mas conseguem uma solução aceitável em tempo hábil. Dentre os métodos de solução exatos, um que se destaca é o procedimento de busca em árvore (BEASLEY, 1985b), mas existem muitos outros na literatura (IORI; DE LIMA et al., 2021; FEKETE et al., 1997; DELORME et al., 2016; KENMOCHI et al., 2009). Na parte de heurísticas, tem-se a *bottom-left* (BAKER et al., 1980; CHEHRAZAD et al., 2022) e *skyline* (WEI et al., 2011), as heurísticas também possuem grande presença na literatura (BURKE et al., 2004; RAKOTONIRAINY et al., 2020; HOPPER, E. B. C. H. et al., 2001; CHEN et al., 2019; HUANG et al., 2007; HOPPER, E. et al., 2001).

Este trabalho visa criar métodos de solução para o problema de empacotamento no espaço de duas dimensões, onde as peças são retangulares e com um recipiente também retangular, mais especificamente na versão do empacotamento 2D da mochila, considerado NP-difícil (IORI; LIMA et al., 2022). Nessa versão do problema, dado um conjunto de itens $\mathcal{I}$, com cada item i possuindo um valor p_i, e uma caixa $\mathcal{B}$, o objetivo é maximizar a soma dos valores dos itens alocados dentro do recipiente. A abordagem escolhida para resolver o problema foi utilizar a heurística *bottom-left*, devido a sua simplicidade e aos limites computacionais e de tempo ao escolher algum método exato. Mesmo com a heurística sendo proposta em 1980, ela ainda está presente na literatura recente (CHEHRAZAD et al., 2022; HOPPER, E. B. C. H. et al., 2001; WEI et al., 2011).

Empacotamento de retângulos tem importância prática (FIRAT et al., 2020) e o fato de somente eles serem considerados não é um grande demérito, já que é possível transformar qualquer polígono em um retângulo com algumas manipulações, ainda que haja um desperdício de área no recipiente ao resolver o problema desse modo. A escolha de focar somente no empacotamento de retângulos foi feita justamente por isso, é possível utilizá-lo para empacotar qualquer tipo de item, com as devidas adaptações.

Antes de abordar o problema (Capítulo 2) e buscar soluções (Capítulo 3), alguns conceitos básicos são mostrados. O Capítulo 1 foca em modelos de otimização e definições (seções 1.1 e 1.2) e a diferença entre métodos exatos e heurísticos (seção 1.3). No Capítulo 2 é dada a definição do problema (seção 2.1), para então mostrar algumas classificações (seção 2.2) e variantes (seção 2.3). Depois, é explicada a heurística *bottom-left* (Capítulo 3) e os métodos feitos com base nela, os quais serão utilizados na resolução das instâncias de teste. Por fim, no Capítulo 4, os principais resultados obtidos são mostrados.

O principal objetivo deste trabalho é criar métodos de solução para o problema de empacotamento da mochila de peças retangulares, todos baseados na heurística *bottom-left*. Outros objetivos mais específicos são: implementar a *bottom-left* e os métodos derivados em Python, executá-los com instâncias de teste da literatura, comparar seus resultados e identificar vantagens e desvantagens de cada um.

1 Conceitos básicos

Antes de estudar o problema, são necessários alguns conceitos básicos e definição formal de termos importantes para a área de Pesquisa Operacional e otimização. Pesquisa Operacional pode ser entendida como o estudo e a aplicação de métodos científicos para tomada de decisões em problemas complexos (ARENALES et al., 2007, p.IX). Ela permite modelar, analisar e solucionar tais problemas de modo, geralmente, satisfatório.

Neste capítulo é explicado sobre modelos de otimização (seção 1.1) e seus tipos (seção 1.2), além de algumas definições sobre otimização, finalizando com a diferença entre métodos heurísticos e exatos (seção 1.3). O problema de empacotamento de retângulos (detalhado no Capítulo 2), alvo deste estudo, é determinístico e pode ser modelado utilizando programação linear inteira mista (WOLSEY, 2020). No trabalho, o problema será explorado do ponto de vista heurístico, com o método apresentado no Capítulo 3.

1.1 Modelos de otimização e outras definições

Modelos de otimização são aproximações da realidade, representam o problema de maneira simples e objetiva, usando restrições. De forma geral, um modelo de otimização busca minimizar ou maximizar uma função $f(x)$ com x obedecendo algumas restrições (CALAFIORE et al., 2014). Pode-se então representar o modelo do seguinte modo:

$$\min/\max f(x), x \in \mathcal{X}.$$

Onde

- x: variáveis de decisão, $x = (x_1, x_2, \ldots, x_n)$.
- $\mathcal{X}$: conjunto factível ou domínio, possui todas as soluções viáveis para o problema.
- $f(x)$: função objetivo, a qual determinará o critério de escolha da solução.

A seguir serão dadas as definições de quatro expressões que aparecem com frequência no estudo de problemas de otimização.

Uma solução x' é **factível** somente se satisfaz todas as restrições dadas ao problema, ou seja, $x' \in \mathcal{X}$. Existem casos onde o problema não tem solução, possivelmente por muitas restrições terem sido aplicadas. Isso é chamado **problema infactível** e $\mathcal{X} = \emptyset$. Se para toda solução for possível encontrar outra melhor o problema é dito **ilimitado**.

Uma solução x' é **ótima** somente se for **factível** e possuir resultado melhor, ou igual, que as demais soluções, isto é, $f(x') \leq f(x), \forall x \in \mathcal{X}$ (caso seja um problema de maximização é necessário substituir "$\leq$" por "$\geq$"). Importante observar que somente existe solução ótima se o problema não for infactível nem ilimitado.

Usando o problema da mochila como exemplo, onde se busca maximizar o valor dos itens alocados sem que seus pesos ultrapassem a capacidade da mochila (IORI; DE LIMA et al., 2021), um modelo possível é:

$$\max \sum_{i \in I} v_i x_i : \sum_{i \in I} w_i x_i \leq C$$

Em que I é um conjunto de itens, v_i e w_i são, respectivamente o valor e o peso do item $i \in I$, C é a capacidade da mochila e x_i são variáveis binárias indicando se o item foi escolhido para mochila ou não.

O problema não é ilimitado, já que a melhor solução (solução ótima) é ocupar toda a capacidade C da mochila, caso seja possível com os itens I. Qualquer solução que não ultrapasse o limite da mochila é factível. O problema não é considerado infactível, pois colocar nenhum item é uma solução factível.

1.2 Tipos de modelo

É importante saber diferenciar os modelos devido ao método de resolução que varia para cada um deles.

1.2.1 Modelo Linear × Não-linear

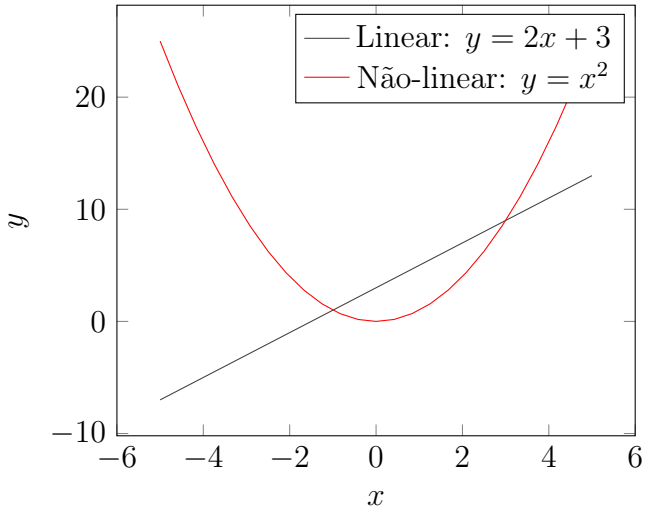
Modelos lineares possuem como função objetivo uma função linear e todas as restrições também são lineares. Exemplos:

- $f(x) = ax + b$.
- $f(x_1, x_2) = x_1 + x_2 - 5$.

Já os não-lineares não obedecem essa regra, podendo ter suas variáveis se multiplicando ou funções trigonométricas e logarítmicas. Exemplos:

- $f(x_1, x_2) = x_1^2 + x_2^2$.
- $f(x_1, x_2) = \tan(x_1 + x_2)$.

Figura 1 – Exemplo de modelo linear e não-linear.

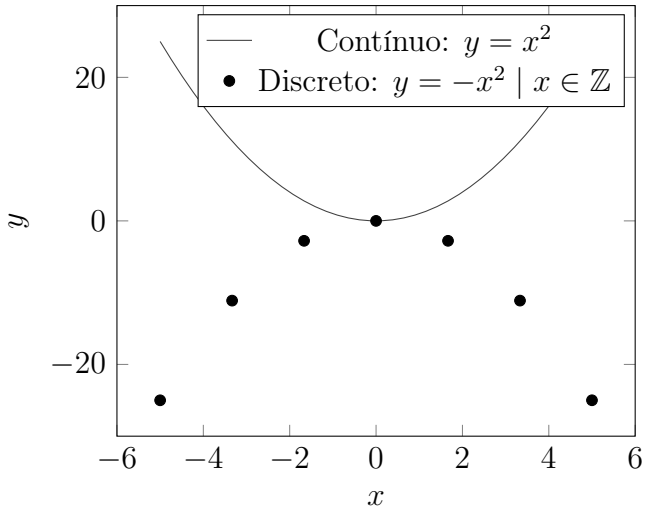


Fonte: feito pelo autor.

1.2.2 Modelo Contínuo × Discreto

Um modelo é contínuo quando sua região factível é contínua, ou seja, dado um ponto dessa região todos os seus vizinhos também serão uma solução. Modelos discretos não possuem seu domínio contínuo. A Figura 2 mostra um gráfico com exemplos de um modelo contínuo e outro discreto.

Figura 2 – Exemplo de modelo contínuo e discreto.



Fonte: feito pelo autor.

1.2.3 Modelo Determinístico × Estocástico

Em modelos determinísticos seus dados são conhecidos, enquanto os estocásticos possuem uma incerteza quanto aos dados.

1.2.4 Tipos de Programação

Com base nas categorias de modelo é possível também dividir métodos de programação (planejamento) para sua solução. HILLIER (1967) traz alguns exemplos algoritmos de para os tipos de programação.

- Linear: modelo linear contínuo determinístico. Algoritmos importantes para essa classe de problemas são: o algoritmo Simplex e algoritmos de pontos interiores.
- Inteira: modelo linear discreto determinístico. Aqui se destacam os algoritmos *branch-and-bound* e *branch-and-cut*.
- Estocástica: modelo linear contínuo estocástico. Alguns métodos para solução podem se basear em simulação dos eventos aleatórios envolvidos.
- Não-linear: modelo não-linear contínuo determinístico. Os algoritmos para solução de problemas não lineares podem ser baseados em gradiente, mas com frequência dependem bastante do problema a ser resolvido.

1.3 Métodos exatos × heurísticos

Métodos exatos sempre vão garantir a solução ótima para o problema, porém encontrar tal solução pode requerer grande tempo e/ou muitos recursos computacionais. Já heurísticas buscam por soluções factíveis e são geralmente usadas em problemas de grande porte.

Um dos métodos exatos mais conhecidos é o algoritmo *branch-and-bound*, ele realiza a enumeração implícita das soluções viáveis de um problema de programação linear inteira mista, mantendo valores para os limitantes inferior e superior de um problema de otimização. O algoritmo termina sua execução quando ambos os limitantes se igualam, garantindo a otimalidade da solução. Detalhes do algoritmo *branch-and-bound* e outros para problemas de programação inteira, como *branch-and-cut* e planos de corte, podem ser vistos em WOLSEY (2020).

Como o problema de empacotamento de retângulos é NP-difícil e o principal interesse é em instâncias de médio e grande porte, utilizar um método exato seria bastante desafiador e, provavelmente, não seria possível obter um resultado em tempo hábil devido aos recursos computacionais disponíveis. Portanto, métodos heurísticos serão usados, já que eles tendem a diminuir a demanda computacional, porém não garantem otimalidade da solução resultante.

Soluções heurísticas tipicamente alternam entre explorar o espaço de busca de forma mais ampla e se concentrar em uma vizinhança de uma solução viável já encontrada. Por isso, em geral, uma heurística garante apenas a otimalidade local da solução. Para escapar de ótimos locais e buscar atingir um resultado melhor, mecanismos de fuga são usados.

Alguns exemplos desses mecanismos são o *multi-start* e o *simulated annealing* (FIRAT et al., 2020; RAKOTONIRAINY et al., 2020; HOPPER, E. B. C. H. et al., 2001).

Continuando o exemplo do problema da mochila (seção 1.1), uma heurística possível para sua solução é ordenar os itens de maneira decrescente de acordo com $\frac{v_i}{w_i}$ e colocá-los na mochila enquanto couber.

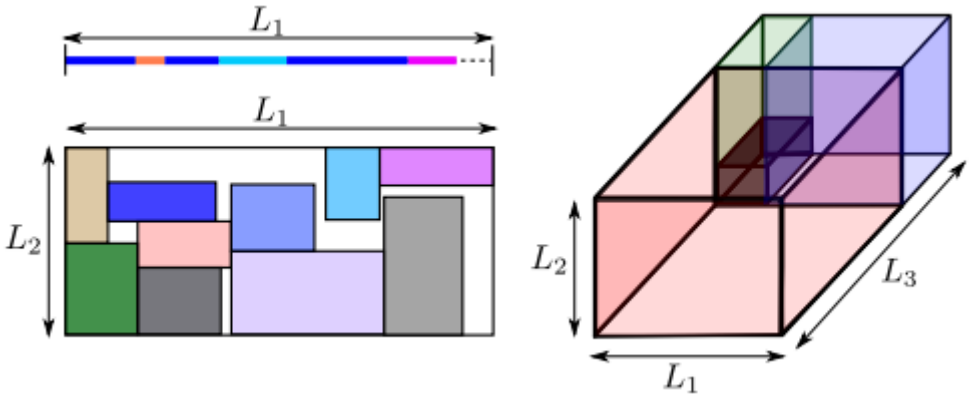
As heurísticas podem ser divididas entre heurísticas construtivas e heurísticas de melhoria. Heurísticas construtivas, como diz o nome, constroem uma solução para o problema, enquanto heurísticas de melhoria, partem de uma solução viável e realizam tentativas de melhorar tal solução (MICHALEWICZ et al., 2013). A heurística *bottom-left* (detalhada no Capítulo 3), a qual será usada para resolver o problema de empacotamento de retângulos (Capítulo 2), é uma heurística construtiva.

2 Problema de empacotamento

Nas últimas três décadas, as publicações na área de corte e empacotamento tiveram um aumento considerável (IORI; DE LIMA et al., 2021; WÄSCHER et al., 2007). Devido a isso, a categorização e organização desses problemas é cada vez mais importante. O trabalho de WÄSCHER et al. (2007) fornece a classificação de problemas de corte e empacotamento baseado na sua dimensão, tipos de itens, tipo de recipiente e função objetivo, fornecendo uma atualização da tipologia definida em DYCKHOFF (1990).

O problema de empacotamento, é um problema de otimização de difícil resolução. Seu objetivo é simples, colocar peças em um espaço N-dimensional, na Figura 3 é possível ver representações para os casos 1D, 2D e 3D. Tanto as peças quanto o espaço, podem ser de formato convexo ou côncavo. Pensando no caso 2D, triângulos, retângulos, círculos e outros polígonos convexos são considerados convexos, enquanto estrelas e outros são côncavos. Os polígonos também podem ser classificados como regulares ou irregulares.

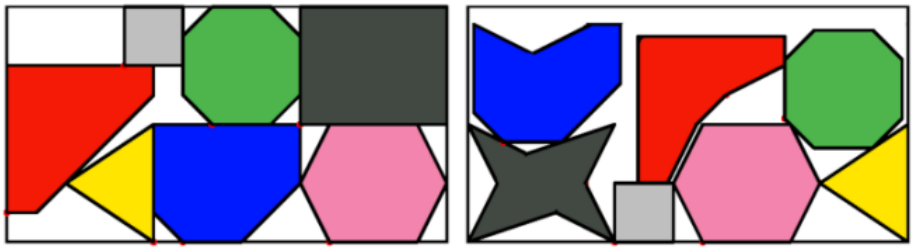
Figura 3 – Representação para o problema de empacotamento 1D, 2D e 3D.



Fonte: CASTELLUCCI (2019).

Existem diversas formas de identificar se um polígono é convexo ou não. A primeira delas é verificando se existe alguma diagonal que não pertença à região interna do polígono, caso exista, o polígono é côncavo, caso contrário, convexo. Também é possível identificar através dos ângulos internos, polígonos côncavos possuem pelo menos um ângulo interno com mais de 180 graus. Uma forma de definir se uma peça é regular ou não, é identificar o número de parâmetros necessários para representá-la. Se for preciso três ou mais é irregular, caso contrário, regular (BARTMEYER et al., 2021). A Figura 4 mostra alguns exemplos de peças convexas e côncavas (à direita) e seus contornos convexos (à esquerda).

Figura 4 – Exemplos de peças convexas e côncavas.



Fonte: BARTMEYER et al. (2021).

O foco deste trabalho será em problemas de empacotamento 2D de peças e objetos retangulares ortogonais, sem qualquer variante (seção 2.3). Por mais simples que seja, é uma categoria muito importante do problema, visto que, no mundo real, a maioria do que temos interesse em resolver se encaixa nessas características. Inclusive, existem vários trabalhos como WEI et al. (2011) e outros mais recentes (MARTIN et al., 2020; FIRAT et al., 2020; CHEN et al., 2019) com o mesmo propósito. Esse escopo também é utilizado para resolver outros problemas, como o planejamento de integração em larga escala (HUANG et al., 2007) e para o roteamento de veículos levando paletes (SILVA et al., 2022). Existem até mesmo instâncias padronizadas para realizar comparativos entre algoritmos (IORI; LIMA et al., 2022), as instâncias usadas neste trabalho serão explicadas no Capítulo 4.

Tratar somente de objetos retangulares não é um grande limitador para resolver com outros tipos de itens, já que é possível usar os contornos convexos de um polígono côncavo para transformá-lo em um polígono convexo (Figura 4). Após isso, basta transformar o polígono em um retângulo. Assim é possível empacotar qualquer polígono usando o empacotamento de retângulos, ainda que haja uma área desperdiçada devido às transformações.

2.1 Definição

Com o escopo do estudo definido como problema de empacotamento de retângulos, é possível ver sua definição formal. De acordo com IORI; LIMA et al. (2022), dado uma caixa retangular $\mathcal{B} = (W, H)$ de comprimento $W \in \mathbb{Z}_+$ e altura $H \in \mathbb{Z}_+$ e um conjunto $\mathcal{I}$ de itens também retangulares, onde cada item $i \in \mathcal{I}$ com comprimento $w_i \in \mathbb{Z}_+, w_i \leq W$ e altura $h_i \in \mathbb{Z}_+, h_i \leq H$. Um empacotamento $\mathcal{I}' \subseteq \mathcal{I}$ em $\mathcal{B}$ pode ser descrito como uma função $\mathcal{F} : \mathcal{I}' \to \mathbb{Z}_+^2$ que mapeie cada item $i \in \mathcal{I}'$ para um par de coordenadas $\mathcal{F}(i) = (x_i, y_i)$, de forma:

$$x_i \in \{0, \ldots, W - w_i\}, y_i \in \{0, \ldots, H - h_i\} \, (i \in \mathcal{I}') \quad (1)$$

$$[x_i, x_i + w_i) \cap [x_j, x_j + w_j) = \emptyset \text{ ou } [y_i, y_i + h_i) \cap [y_j, y_j + h_j) = \emptyset \, (i, j \in \mathcal{I}', i \neq j) \quad (2)$$

Nesse modo de representação a caixa está posicionada no plano cartesiano, com seu canto inferior esquerdo na origem. Já as coordenadas $\mathcal{F}(i) = (x_i, y_i)$ representam a posição em que o canto inferior esquerdo da peça será alocado. A Restrição 1 garante que cada item deve estar inteiramente na caixa, enquanto a Restrição 2 impede sobreposição entre itens. Ambas restrições indicam uma orientação fixa, ou seja, peças não podem ser rotacionadas.

2.2 Classificação

Por existirem diferentes objetivos na solução de um problema de empacotamento foram criadas algumas classificações. Algumas delas (as principais) são mostradas em IORI; DE LIMA et al. (2021) e IORI; LIMA et al. (2022), as quais serão exploradas em seguida, com alguns exemplos já vistos na Introdução.

O objetivo do **Empacotamento 2D em Faixa**, em inglês *Two-Dimensional Strip Packing Problem* (2D-SPP), é encontrar um empacotamento de altura mínima H para um dado conjunto de itens $\mathcal{I}$ em uma caixa $\mathcal{B} = (W, H)$ com comprimento fixo W. Muito aplicado na área têxtil para minimizar o comprimento de tecido cortado para fazer peças de roupas.

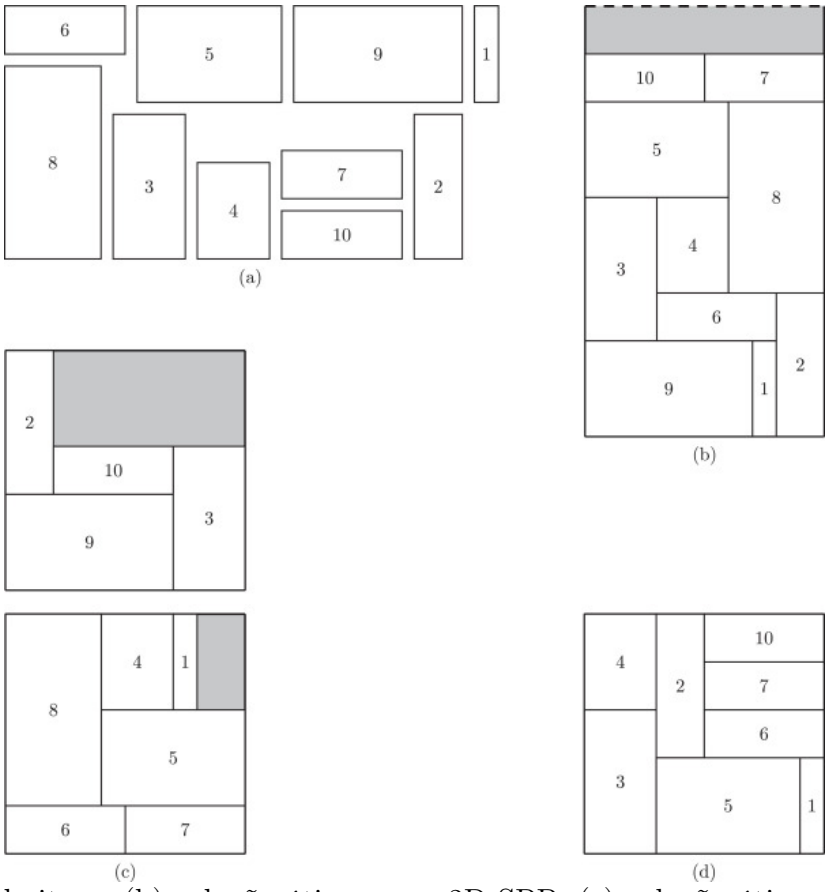
No **Empacotamento 2D da Mochila**, em inglês *Two-Dimensional Knapsack Problem* (2D-KP), dado um conjunto de itens $\mathcal{I}$, onde cada item $i \in \mathcal{I}$ é associado a um valor p_i, e uma caixa $\mathcal{B}$, deve-se encontrar um subconjunto $\mathcal{I}' \subseteq \mathcal{I}$ que maximize $\sum_{i \in \mathcal{I}'} p_i$. Geralmente o valor p_i é dado pela área do item, dessa forma, outra interpretação do problema seria minimizar a área desperdiçada (vazia) da caixa $\mathcal{B}$. Pode ser utilizado para maximizar o número de peças cortadas de um pedaço de couro, por exemplo.

Já o **Empacotamento 2D em Caixas**, em inglês *Two-Dimensional Cutting Stock Problem* (2D-CSP), envolve encontrar uma solução que minimize o número de caixas idênticas necessárias para empacotar todos os itens $i \in \mathcal{I}$, onde cada item possui uma demanda $d_i \in \mathbb{Z}_+$ (número mínimo de cópias do item que precisam ser empacotadas). Existe uma versão menos genérica do 2D-CSP, o *Two-Dimensional Bin Packing Problem* (2D-BPP), onde a demanda d_i de cada item é 1. As caixas podem possuir diferentes tamanhos ao utilizar uma variante (seção 2.3), mas a maioria dos problemas lida com as mesmas dimensões. Facilmente aplicável na área logística e de transporte, seja minimizando o número de paletes ou veículos de entrega.

Por fim, no **Empacotamento 2D Ortogonal**, em inglês *Two-Dimensional Orthogonal Packing Problem* (2D-OPP), busca-se uma solução, caso exista, para empacotar **todos** os itens $i \in \mathcal{I}$ na caixa $\mathcal{B}$. Usado em situações onde se precisa alocar todos os itens dentro de um caminhão.

Todos os problemas descritos são NP-difícil, com exceção do Ortogonal, sendo NP-completo (IORI; LIMA et al., 2022). Existem resultados recentes para todas as classes do problema (CÔTÉ et al., 2014; DELORME et al., 2017; VELASCO et al., 2019; MARTIN et al., 2020; MRAD, 2015; CINTRA et al., 2008; FURINI et al., 2016). A Figura 5 traz exemplos de solução ótima para 2D-SPP, 2D-BPP e 2D-KP, para um dado conjunto de itens.

Figura 5 – Exemplo de solução ótima para algumas classes.



(a) conjunto de itens; (b) solução ótima para 2D-SPP; (c) solução ótima para 2D-BPP; (d) solução ótima para 2D-KP (se os valores dos itens corresponderem a sua área).

Fonte: IORI; DE LIMA et al. (2021).

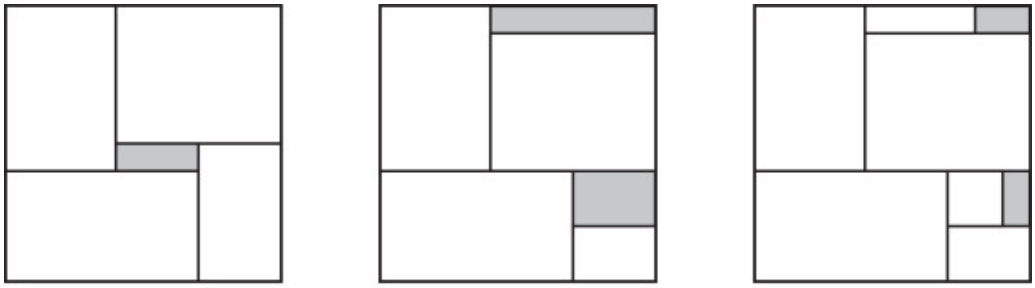
Este trabalho resolverá as instâncias de teste somente para o 2D-KP, onde o valor p_i de cada item i será sua própria área, independente do valor original da instância (caso tenha).

2.3 Variantes

Variantes são pequenas alterações no escopo do problema, também podem ser vistas como restrições ou relaxamento. Existem quatro comuns (IORI; LIMA et al., 2022; IORI; DE LIMA et al., 2021), as quais são descritas a seguir.

Corte guilhotinado consiste em cortar a caixa de forma paralela a um de seus lados de ponta a ponta recursivamente, é útil na resolução de problemas de corte (problemas de empacotamento são equivalentes aos de corte). Máquinas de corte, usualmente, só conseguem produzir peças por meio de sequências de corte guilhotinados, ou seja, de ponta a ponta e paralelos aos lados do recipiente. Isso pode ser um problema, já que, com essa restrição, nem todos os conjuntos de itens são compatíveis com tal método (Figura 6). Frequentemente ainda existe um limite k de cortes por recipiente, conhecidos como problemas de k-estágios. Geralmente, k é igual a 2 ou 3, com um corte extra chamado *trimming*.

Figura 6 – Exemplos de corte guilhotinado.



(a) não permite corte guilhotinado; (b) corte de 2-estágios com *trimming*; (c) corte de 3-estágios com *trimming*.

Fonte: IORI; DE LIMA et al. (2021).

Rotações ortogonais são um modo de relaxar o problema, permitindo rotações de 90 graus para os itens a serem alocados. Porém, isso também leva a desafios mais complexos, pois aumenta o número de decisões a serem tomadas, fazendo o modelo possuir mais varáveis e restrições. Essa variante, geralmente, é tratada adicionando um parâmetro de decisão binária r_i para cada item $i \in \mathcal{I}$, indicando se a rotação do item é permitida ou não.

Restrições de carga e descarga implicam que algumas peças **devem** ser posicionadas em dada posição para não ser necessário mover outros itens quando tal peça for carregada/descarregada. Usando como exemplo um caminhão de entregas, visa evitar situações onde um produto precisa ser descarregado para se ter acesso a um item mais ao fundo e então carregar novamente o primeiro item. O tratamento da variante pode ser feito ao fixar as coordenadas (x_i, y_i), dos itens $i \in \mathcal{I}$ que possuem essa restrição, no modelo.

Existem variantes aplicáveis somente a algumas categorias do problema, é o caso de **caixas de tamanho variável**, aplicável ao 2D-CSP e ao 2D-BPP e define que caixas não precisam possuir as mesmas dimensões, custos e disponibilidade. Com essa variante, o objetivo passa a ser empacotar todos os itens com um custo mínimo de recipientes.

3 Métodos de solução

Como descrito na seção 2.2, a maioria das classes do problema são NP-difíceis. Isso torna métodos de soluções exatos, os quais buscam pela solução ótima, extremamente custosos em tempo e recursos computacionais em instâncias de porte moderado, muitas vezes sendo inviáveis por falta de algum desses dois motivos. Consequentemente a literatura é dominada por abordagens que usam heurísticas e meta-heurísticas, sendo a *bottom-left* uma das principais estratégias de solução e será usada no estudo deste trabalho.

A *bottom-left* é uma heurística construtiva (seção 1.3) proposta por BAKER et al. em 1980. Embora tenha sido proposta a décadas, ainda é bastante usada na literatura atual, além de poder ser usada como componente de algoritmos mais sofisticados e para diferentes classes e variantes do problema. Ela foi utilizada nos trabalhos de HOPPER, E. B. C. H. et al. (2001) na comparação de vários métodos de solução, WEI et al. (2011) trazendo uma revisão do método e seus derivados e, mais recentemente, CHEHRAZAD et al. (2022) através de uma adaptação para o empacotamento de itens irregulares de forma gulosa.

Sua premissa é simples, dado uma fila de itens como entrada, enquanto ela não estiver vazia, basta retirar o primeiro item dela e alocar no canto mais a baixo e à esquerda quanto for possível (BARTMEYER et al., 2021), sem sobreposições entre peças (seção 2.1). Caso não exista uma posição válida, a peça é desconsiderada e passa-se para próxima da fila. A Figura 7 mostra um exemplo de alocação para um dado conjunto de peças regulares.

Figura 7 – Representação de alocação usando *bottom-left*.

(a) Sequência para alocação.

(b) Alocação gerada pela regra *bottom-left*.

Fonte: BARTMEYER et al. (2021).

Vale destacar que a própria ordem da fila pode gerar resultados diferentes, alterando a qualidade da solução. Um dos resultados esperados deste trabalho é identificar se há alguma forma de ordenação que se destaque na qualidade de solução, através da comparação

entre os diferentes modos. Para isso, serão usados conjuntos de instâncias frequentemente utilizados na literatura.

3.1 Critérios de ordenação

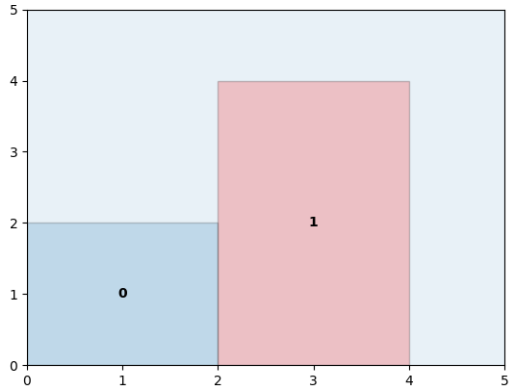
Para determinar o impacto da ordenação da fila, cinco critérios de ordenação foram escolhidos, sendo eles: área, perímetro, largura, altura e *id*. A ordenação por *id* considera a ordem em que os itens foram colocados na lista, ou seja, seria a forma padrão de resolver e ele será a base para definir se os demais critérios possuem algum benefício. Além disso, cada critério pode ser usado para ordenar a fila em ordem crescente ou decrescente, algo que também será analisado. Na literatura o mais comum é utilizar a ordenação decrescente pela área (CHEN et al., 2019).

3.2 Sobreposição e domínio contínuo

Por mais simples que a heurística *bottom-left* seja, ainda existem dois desafios, respeitar a Restrição 2 (sobreposição de peças) e o domínio contínuo de coordenadas ao tentar alocar um item. As Figuras 8 a 12 serão usadas para demonstrar os desafios citados.

Supondo uma instância com recipiente de altura e largura 5 e três itens retangulares a serem alocados, nesta ordem, com as seguintes dimensões: altura e largura 2, altura 4 e largura 2 e altura 1 e largura 3. A Figura 8 mostra um estado intermediário do algoritmo de solução, onde o item 0 foi alocado na coordenada (0, 0) e o item 1 foi alocado a sua direita na posição (2, 0) para respeitar a Restrição 1, porque não cabe logo acima na posição (0, 2).

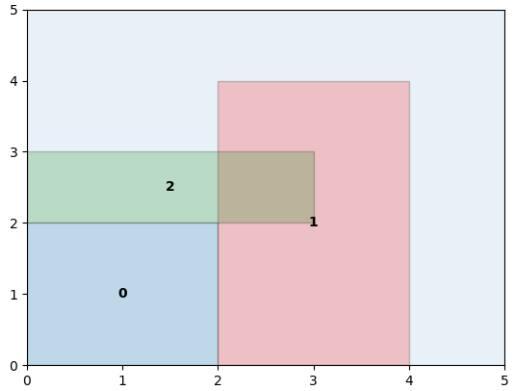
Figura 8 – Itens 0 e 1 posicionados.



Fonte: feito pelo autor.

Como a próxima peça da fila tem largura 3 e altura 1, ao posicioná-la na posição (0, 2) percebe-se que a Restrição 1 é satisfeita, mas a Restrição 2 não (Figura 9).

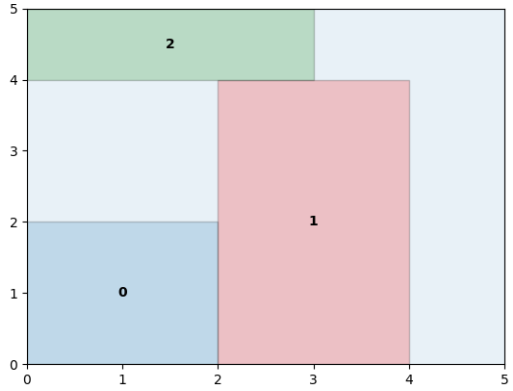
Figura 9 – Itens 0, 1 e 2 posicionados, mas com sobreposição.



Fonte: feito pelo autor.

Nesse caso, com poucas peças alocadas e auxílio visual, é fácil identificar que a posição correta, seguindo a lógica *bottom-left*, seria a de coordenadas (0, 4) (Figura 10). Porém, encontrar tal posição pode ser extremamente complexo. Entre as coordenadas (0, 2) e (0, 4) o espaço é contínuo, existindo infinitas outras coordenadas entre elas, sendo impossível checar todas.

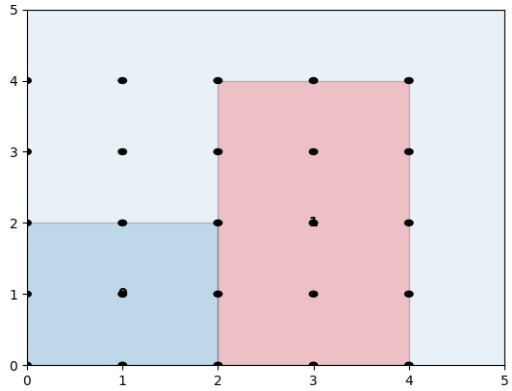
Figura 10 – Itens 0, 1 e 2 posicionados, sem sobreposição.



Fonte: feito pelo autor.

Como todas as instâncias usadas para testes tratam somente de peças e recipientes com valores inteiros, uma abordagem possível seria discretizar o domínio, conforme a Figura 11.

Figura 11 – Itens 0 e 1 posicionados, com domínio discreto.

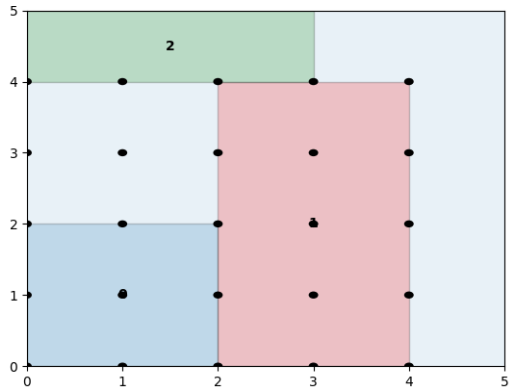


Fonte: feito pelo autor.

Dessa forma somente coordenadas de valores inteiros precisariam ser verificadas, resolvendo parcialmente o problema com o domínio, já que ainda existiriam muitos pontos, principalmente em instâncias com recipientes grandes. Mas isso ainda não resolve a sobreposição de itens. Para cada ponto ainda é necessário verificar se existe sobreposição com cada uma das peças já alocadas, algo custoso.

Na Figura 12 é possível notar que três coordenadas precisaram ser checadas até encontrar uma que cumprisse as Restrições 1 e 2. Por se tratar de um estado inicial de uma instância pequena esse processo não é tão custoso, porém em instâncias maiores ou estados mais avançados isso se tornará cada vez mais custoso, visto que o domínio será maior e, principalmente, mais peças estarão alocadas para checar possíveis sobreposições.

Figura 12 – Itens 0, 1 e 2 posicionados, com domínio discreto.



Fonte: feito pelo autor.

Outro fator o qual deve ser observado é de que a discretização do domínio pode não funcionar bem em casos onde os valores trabalhados não sejam inteiros. Nesses casos a discretização poderia ocorrer com algum nível de precisão nas casas decimais, mas ainda

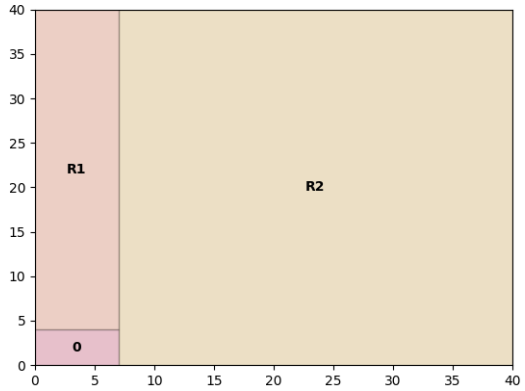
assim as chances de desperdício de área são grandes. Ou seja, muitos problemas reais seriam complexos de serem resolvidos com a discretização.

3.3 Criação de regiões

Os dois problemas expostos na seção 3.2 podem ser resolvidos utilizando a estratégia de criação de regiões. Com essa técnica, a Restrição 2 é trivialmente satisfeita. Nela, ao posicionar uma peça, duas regiões são criadas e o item seguinte será somente posicionado se couber em uma das regiões disponíveis.

Supondo um recipiente com altura e largura 40 e um item 0 com altura 4 e largura 7. Quando o item for posicionado na coordenada (0, 0), duas regiões, R1 e R2, serão criadas (Figura 13). A região R1 começará na coordenada (0, 4) e a R2 na (7, 0).

Figura 13 – Regiões criadas traçando uma linha vertical.



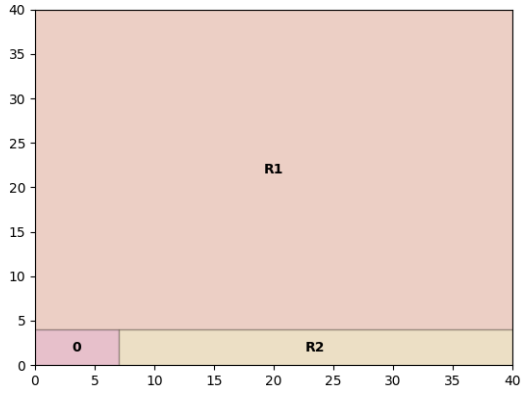
Fonte: feito pelo autor.

Agora o domínio passa a ser somente o canto inferior esquerdo de cada uma das regiões e sobreposições deixam de ser possíveis. Além disso, a regra para definir se uma peça cabe em dada região é igual a Restrição 1, simplificando o algoritmo. A fim de identificar o impacto das regiões na solução do modelo, quatro formas de criação delas foram usadas.

A primeira delas é **traçando uma linha vertical** a partir do canto superior direito de cada peça alocada (Figura 13). Nela a região R1 terá altura 36 e largura 7, indo até à coordenada (7, 40). Enquanto a R2 possuirá altura 40 e largura 33, chegando até à coordenada (40, 40).

A segunda é semelhante à primeira, porém **traçando uma linha horizontal** (Figura 14). Nesse caso, R1 terá altura 36 e largura 40, indo até à coordenada (40, 40). Já R2 possuíra altura 4 e largura 33, chegando até à coordenada (40, 4).

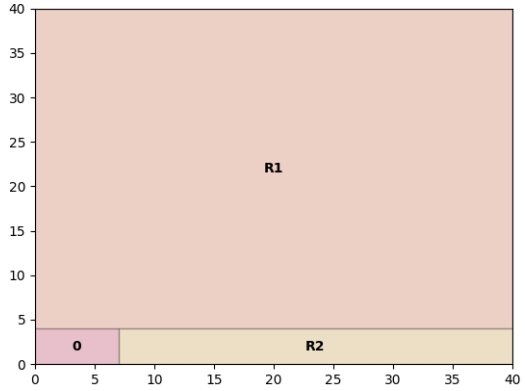
Figura 14 – Regiões criadas traçando uma linha horizontal.



Fonte: feito pelo autor.

Na terceira, a linha traçada (vertical ou horizontal) depende da área das regiões criadas com cada linha. Nesse modo o objetivo é maximizar a área de uma das regiões geradas, ele **identifica qual linha irá gerar a região de maior área e a traça**. Por exemplo, a Figura 13 gerou uma região com 252 de área e outra com 1320, enquanto a Figura 14 obteve regiões com 1440 e 132, então, nesse caso, a linha traçada será a horizontal (Figura 15).

Figura 15 – Regiões criadas maximizando uma das regiões.



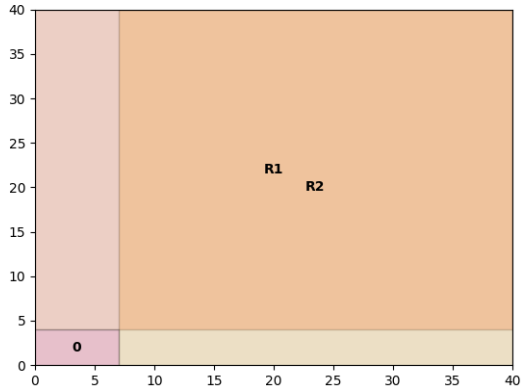
Fonte: feito pelo autor.

Maximizar uma região pode ser interessante, pois aumenta as chances do próximo item conseguir ser alocado, visto que uma das regiões será mais espaçosa. Em contrapartida, esse método também pode acabar gerando muitas regiões pequenas que não sejam utilizadas, diminuindo a qualidade da solução.

No quarto e último modo de criar regiões nenhuma linha é traçada, todas as regiões vão até o final do recipiente (Figura 16), criando **regiões sobrepostas**. R1 terá altura 40 e largura 36, enquanto R2 possuirá altura 40 e largura 33. Então, R1 e R2 terminarão na

coordenada (40, 40). Nesse caso, sobreposições de peças podem ocorrer, então verificações são necessária para cumprir a Restrição 2. Ao fazer isso, é possível que mais peças sejam alocadas, visto que todas as regiões possuem área máxima. Esse modo foi criado para identificar se é de fato melhor que os demais e qual seu custo.

Figura 16 – Regiões criadas possibilitando sobreposições.



Fonte: feito pelo autor.

Com os critérios para criação de regiões explicados, é possível diferenciá-los em dois tipos. O primeiro é dos que permitem sobreposição entre peças e, por isso, precisam de verificações para respeitar a Restrição 2 (regiões complexas), nesse tipo se encaixa somente o quarto modo. O segundo tipo contém os três primeiros critérios, onde somente a Restrição 1 precisa ser checada (regiões simples).

A Tabela 1 mostra, de forma resumida, os quatro modos de criar regiões, seu tipo (regiões simples ou complexas) e sua relação com a Restrição 2. A coluna "Divisão" indica o critério usado ao criar as regiões, enquanto a coluna "Restrição 2" mostra se a Restrição 2 é trivialmente satisfeita ou não. A última coluna ("Tipo"), indica se a região é simples ou complexa e está diretamente relacionada a Restrição 2.

Tabela 1 – Modos de criar regiões, seu tipo e sua relação com a Restrição 2.

Modo	Divisão	Restrição 2	Tipo
1	Vertical	Trivial	Simples
2	Horizontal	Trivial	Simples
3	Maior área	Trivial	Simples
4	Regiões sobrepostas	Não trivial	Complexas

Fonte: feito pelo autor.

Todas as variações propostas foram implementadas em Python e avaliadas computacionalmente, os resultados são apresentados no Capítulo 4.

4 Resultados

Para testar os métodos de solução criados foram usadas 45 instâncias de teste da literatura, separadas em cinco conjuntos de instância de características diferentes: BKW, GCUT, NGCUT, OF e OKP. Todas as elas foram obtidas através da biblioteca pública 2DPackLib[1] (IORI; LIMA et al., 2022).

O foco do trabalho é no 2D-KP e com o critério de maximização sendo a área ocupada do espaço. Mas nem todos conjuntos foram feitos para ser resolvidos dessa forma, nesses casos foram feitas leves adaptações para usá-los. O motivo de usar instâncias feitas com outro objetivo é para não viciar o modelo em instâncias específicas.

As instâncias BKW foram propostas para 2D-SPP (BURKE et al., 2004), esse conjunto é interessante, pois existe uma solução ótima onde todos os itens podem ser alocados. CÔTÉ et al. (2014) apresentam alguns resultados para esse conjunto, já DELORME et al. (2017) trazem resultados usando rotações ortogonais.

Instâncias GCUT foram propostas para 2D-KP com corte guilhotinados (BEASLEY, 1985a). Esse conjunto já foi usado na literatura no 2D-SPP (CÔTÉ et al., 2014), 2D-SPP com corte guilhotinado (MRAD, 2015), 2D-SPP com rotações ortogonais (DELORME et al., 2017) e 2D-CSP com corte guilhotinado (CINTRA et al., 2008).

NGCUT é um conjunto proposto para 2D-KP (BEASLEY, 1985b). Ele possui resultados recentes para 2D-KP, 2D-SPP (CÔTÉ et al., 2014) e 2D-SPP com rotações ortogonais (DELORME et al., 2017).

As instâncias OF foram inicialmente elaboradas para 2D-KP com cortes guilhotinados (OLIVEIRA et al., 1990) e foram resolvidas recentemente com o mesmo propósito (VELASCO et al., 2019; MARTIN et al., 2020).

Por fim, instâncias OKP foram criadas para 2D-KP (FEKETE et al., 1997) e já foram resolvidas para versões sem e com corte guilhotinado (FURINI et al., 2016).

Como são cinco critérios de ordenação (seção 3.1), com cada critério podendo ser crescente ou decrescente, quatro formas de criar regiões (seção 3.3) e 45 instâncias, tem-se o total de 1800 casos de teste. Além disso, para conseguir resultados mais fiéis, a média, mediana e desvio padrão do tempo de execução foram calculados. Por isso, cada caso foi executado cinco vezes, totalizando 9000 execuções. Outros dados como a qualidade de solução (objetivo do trabalho), porcentagem de itens alocados e tempo, também foram computados. Todas as execuções foram feitas em um mesmo computador, com configurações conforme a Tabela 2.

Ao analisar a média, a mediana e o desvio padrão do tempo de execução, observou-se que a média e mediana possuem valores quase idênticos, enquanto o desvio padrão

[1] Disponível em: https://site.unibo.it/operations-research/en/research/2dpacklib. Acessado em: 7 de julho de 2023.

Tabela 2 – Configuração do computador de testes.

CPU	AMD Ryzen™ 5 3600X
RAM	16 GiB
Python	3.11.0
SO	Linux Mint 21.1 Cinnamon
Kernel	5.15

Fonte: feito pelo autor.

é pequeno ao ponto de poder ser ignorado, indicando que cinco execuções por caso de teste são suficientes. Portanto, a mediana e desvio padrão serão omitidos no restante do trabalho, podendo ser encontrados na versão completa dos dados gerados no Github[2]. No Apêndice A é possível ver uma versão resumida de todos os dados gerados (sem mediana e desvio padrão) para cada instância.

Nas tabelas apresentadas nas seções seguintes, as colunas "Vitórias" e "Empates" trazem os resultados de forma quantitativa, enquanto a coluna "Qualidade %" de modo qualitativo. Nos testes, o valor p_i dos itens sempre é sua área, desconsiderando os valores dados pelas instâncias, caso hajam.

A qualidade de solução é determinada pela área ocupada do recipiente, no caso de todos os itens serem alocados, a qualidade é 100% independente da área do recipiente. A coluna "Vitórias" indica quantas vezes tal método de solução obteve o melhor resultado em comparação com os demais métodos em outras linhas. Enquanto a coluna "Empates" mostra a quantidade de vezes que o método conseguiu a melhor qualidade, mas outros também conseguiram. Essas colunas foram feitas da seguinte forma: entre cada combinação de critério de ordenação, modo de criar regiões e instâncias, é feita a comparação se a qualidade de solução foi melhor para ordenação crescente ou decrescente. No caso de ambas conseguirem o melhor resultado, é acrescido 1 tanto na coluna "Vitórias", quanto na "Empates" de ambas. Por fim, a coluna "Tempo (s)" mostra o tempo médio de execução do método em segundos.

4.1 Ordenação crescente × decrescente

Uma primeira observação é a discrepância na qualidade de solução entre a ordenação crescente e a decrescente, algo já esperado. Na Tabela 3 é possível notar que ordenando de forma decrescente é possível ocupar cerca de 20% a mais do espaço (coluna "Qualidade %"), quando comparado a ordenação crescente, em média.

Com isso, fica claro que ordenar a fila de entrada da *bottom-left* de modo decrescente é vantajoso em termos de qualidade, quantidade e tempo de execução. Isso se deve a como as regiões são criadas, as Figuras 17 a 20 serão utilizadas para exemplificar, elas representam estados intermediários do algoritmo, usando ordenação crescente pela altura e

[2] Disponível em: https://github.com/G-Carneiro/packing-problem/. Acessado em: 7 de julho de 2023.

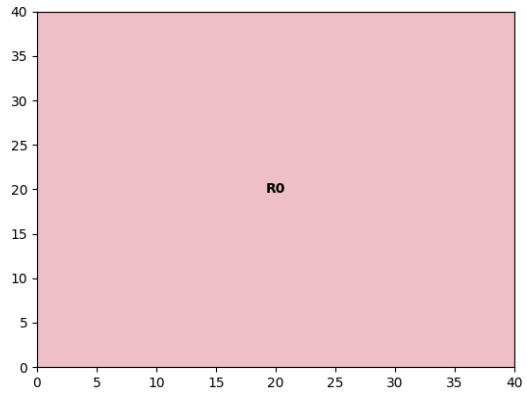
Tabela 3 – Resultado da comparação entre ordenação crescente e decrescente.

Decrescente	Vitórias	Empates	Qualidade %	Tempo (s)
Sim	736	8	78.9136	1.7798e+00
Não	167	8	57.3060	2.3715e+00

Fonte: feito pelo autor.

linhas horizontais para criação de regiões para a instância BKW01. A Figura 17 representa o estado inicial do algoritmo de solução, onde ainda não existem peças alocadas.

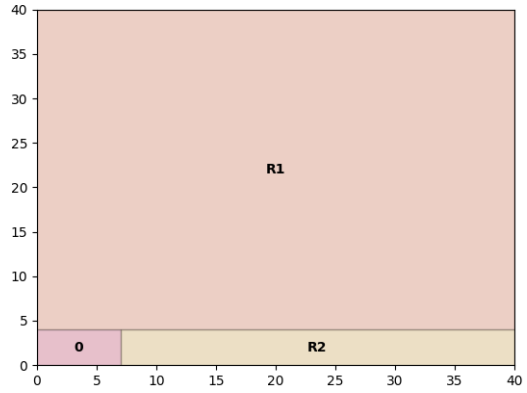
Figura 17 – Regiões criadas na ordenação crescente - estado inicial.



Fonte: feito pelo autor.

Ao posicionar uma peça, uma das regiões ficará com a mesma altura do item recém-posicionado (Figura 18), como a ordenação é crescente a próxima peça terá no mínimo a mesma altura, mas o provável é que seja mais alta, impossibilitando seu posicionamento nessa região.

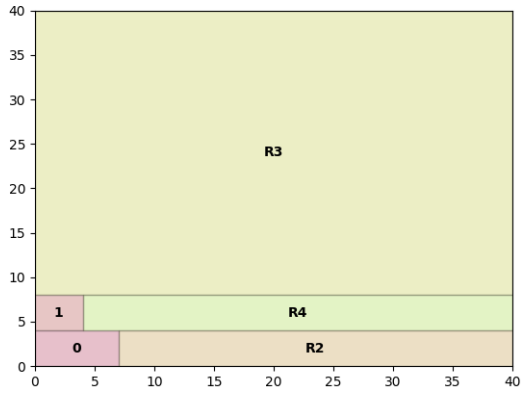
Figura 18 – Regiões criadas na ordenação crescente - estado 1.



Fonte: feito pelo autor.

O mesmo irá ocorrer para todos os itens seguintes (Figura 19), fazendo com que muitas regiões fiquem sem poder receber peças.

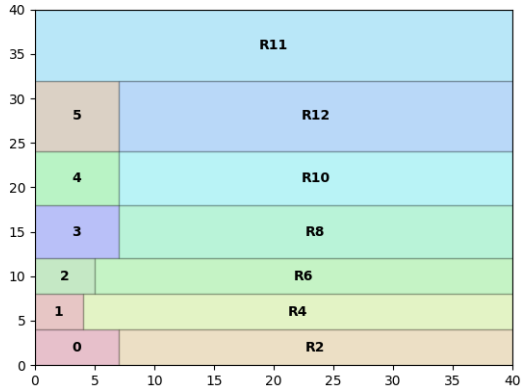
Figura 19 – Regiões criadas na ordenação crescente - estado 2.



Fonte: feito pelo autor.

A Figura 20 mostra o estado final do algoritmo de solução e grande parte do espaço ainda está livre. Na Tabela 11 do Apêndice A é possível ver que a qualidade de solução do modelo com essas combinações foi de 12.75%, enquanto ao utilizar ordenação decrescente foi possível encontrar uma solução de 100%. Algo semelhante ocorre com outros critérios de ordenação e criação de regiões.

Figura 20 – Regiões criadas na ordenação crescente - estado final.



Fonte: feito pelo autor.

4.2 Comparativo entre critérios de ordenação

Ao comparar os diferentes critérios de ordenação (Tabela 4), inicialmente tem-se um resultado curioso, a ordenação por *id* (nenhuma ordenação) está com os melhores resul-

tados qualitativos. Além disso, todos os critérios conseguiram praticamente a mesma qualidade média.

Tabela 4 – Resultado da comparação entre critérios de ordenação.

Ordenação	Vitórias	Empates	Qualidade %	Tempo (s)
Área	81	56	67.7915	2.0726e+00
Perímetro	93	52	68.9870	2.0366e+00
Altura	73	24	65.9848	1.8254e+00
Largura	94	32	69.1286	2.3444e+00
Id	127	12	68.6572	2.0995e+00

Fonte: feito pelo autor.

Como mencionado anteriormente (seção 4.1), ordenar a fila de forma crescente gera resultados ruins. Mas isso não ocorre quando o critério de ordenação é o *id*, porque nele tem-se peças sem seguirem alguma ordem. Com esse critério o impacto da escolha entre ordenação crescente ou decrescente é praticamente nulo. Por isso ele também se sai muito melhor que os demais na ordenação crescente, sendo o vitorioso na esmagadora maioria. Já as qualidades próximas se justificam nos critérios diferentes do *id* pois a discrepância entre ordenação crescente e decrescente é muito grande, mas ao tirar média isso acaba não ficando evidente.

A Tabela 5 mostra o mesmo comparativo que a Tabela 4, porém agora somente considerando a ordenação decrescente, já que não existem motivos para usar a crescente. Representando os dados dessa forma fica fácil identificar que utilizar algum critério de ordenação para a fila de entrada é vantajoso, pois ao usar o *id* os resultados foram os piores. Além disso, percebe-se que as ordenações por área e perímetro obtiveram os melhores resultados, ainda que os demais também sejam competitivos.

Tabela 5 – Resultado da comparação entre critérios de ordenação decrescente.

Ordenação	Vitórias	Empates	Qualidade %	Tempo (s)
Área	63	39	82.7353	1.5874e+00
Perímetro	71	38	84.6986	1.5769e+00
Altura	40	16	77.4182	1.5655e+00
Largura	66	24	81.1899	2.0805e+00
Id	16	5	68.5261	2.0889e+00

Fonte: feito pelo autor.

A alta competitividade entre os critérios de ordenação é interessante, pois a maioria dos trabalhos na literatura, como o de CHEN et al. (2019), usam somente ordenação pela área, e isso pode ser um forte indicativo que os demais critérios devem ser mais explorados em certas circunstâncias. Ao considerar somente a ordenação decrescente deixa claro que a alta quantidade vitórias do critério *id* na Tabela 4 realmente se deve a ordenação crescente.

4.3 Comparativo entre criação de regiões

O comparativo entre métodos de criação de regiões gerou resultados interessantes, na Tabela 6 ocorre algo semelhante ao da Tabela 4. Regiões criadas de modo a ser necessário verificar sobreposições obtiveram excelentes resultados qualitativos (última linha), pois são pouco afetadas pela ordenação crescente (seção 4.1), assim como o critério de ordenação por *id*, já que as regiões sempre possuem área máxima (seção 3.3).

Tabela 6 – Resultado da comparação entre criação de regiões.

Divisão	Vitórias	Empates	Qualidade %	Tempo (s)
Vertical	155	102	65.0025	2.4311e-03
Horizontal	122	101	63.0163	7.2323e-03
Maior área	189	158	69.5409	1.3313e-02
Sobrepostas	334	195	75.0333	8.4208e+00

Fonte: feito pelo autor.

Ao considerar somente a ordenação decrescente (Tabela 5), a diferença em relação aos demais métodos de criação diminui, ainda que o último modo permaneça sendo o melhor qualitativa e quantitativamente. Os modos criados traçando uma linha vertical ou horizontal apresentaram qualidades semelhantes e os menores tempos de execução, mas o método o qual traça uma linha vertical obteve mais vitórias. Regiões criadas para maximizar uma das mesmas conseguiram o segundo melhor resultado qualitativo e quantitativo, ao custo de um pequeno acréscimo no tempo de execução em relação aos dois primeiros. O último modo de fato conseguiu os melhores resultados, porém a um custo altíssimo, levando cerca de 1000 vezes mais tempo que métodos mais rápidos.

Tabela 7 – Resultado da comparação entre criação de regiões - ordenação decrescente.

Divisão	Vitórias	Empates	Qualidade %	Tempo (s)
Vertical	98	79	76.4030	2.7157e-03
Horizontal	70	60	75.9970	6.2101e-03
Maior área	104	89	79.7175	1.3743e-02
Sobrepostas	176	119	83.6420	7.2176e+00

Fonte: feito pelo autor.

A Tabela 8 traz um comparativo entre os dois tipos de criação de regiões, os que é preciso checar sobreposição e os que não (seção 3.3). Na primeira linha da tabela a coluna "Qualidade %" representa a média do melhor resultado obtido em cada instância e a coluna "Tempo Total (s)" mostra a soma dos tempos que cada método de solução levou para cada instância. As duas colunas consideram somente métodos de solução que usam regiões onde não são necessárias verificações de sobreposição, ou seja, são considerados 30 modos de solução. A segunda linha considera somente método de solução o qual utiliza ordenação decrescente pela área e criação de regiões onde é necessário verificar sobreposições, esse método foi escolhido para o comparativo por apresentar os melhores resultados quantitativos e ser o segundo melhor qualitativamente (seção 4.4).

Tabela 8 – Resultado da comparação entre tipos de regiões.

Sobreposição	Qualidade %	Tempo Total (s)
Não	90.8278	1.6299e+01
Sim	87.2957	2.8313e+02

Fonte: autor

Com a Tabela 8 fica claro que, por mais que usar regiões onde é preciso checar sobreposições apresente o melhor resultado, é melhor executar todos demais métodos os quais não precisem e escolher somente o de melhor solução, pois assim é possível obter, na média, soluções de maior qualidade e ainda levando 10 vezes menos tempo.

4.4 Comparativo entre combinações

A Tabela 9 contém os resultados para cada um dos quarenta métodos de solução feitos. Com ela é possível identificar qual dos métodos apresenta melhores resultados qualitativos e quantitativos.

Dentre os métodos que usam regiões onde é preciso checar sobreposições, os mais interessantes são o de ordenação decrescente pela área (linha 30), pelo perímetro (linha 32) e pela largura (linha 36). Utilizando a área conseguiu-se o segundo melhor resultado quantitativo e também qualitativo. Com o perímetro foi possível atingir o terceiro maior número de vitórias e a melhor qualidade de solução. Por fim, a largura obteve o melhor resultado quantitativo e ficou em terceiro na qualidade de solução.

Nos métodos que usam regiões mais simples, os resultados foram bem variados nas combinações, dentre eles se destacam: regiões criadas usando linha vertical e ordenação pela largura (linha 6) e maximizar uma região e ordenar pela área (linha 20) ou pelo perímetro (linha 22). O método da linha 6 ficou em terceiro lugar no critério quantitativo (empate com a linha 32) e conseguiu a quinta posição na qualidade de solução. As linhas 20 e 22 tiveram a quarta maior quantidade de vitórias (empate também com a linha 2), já qualitativamente a linha 20 obteve a sexta melhor média nas soluções, enquanto a linha 22 conseguiu a quarta.

Ainda que os resultados dos métodos que usam regiões as quais calculam sobreposições sejam melhores, não compensa utilizá-los, como visto na seção 4.3.

4.5 Conjuntos de instâncias

A Tabela 10 considera os melhores resultados para cada conjunto de instância. As colunas “Qualidade %” e “Itens %” representam, respectivamente, a média percentual da qualidade de solução e itens alocados no melhor resultado de cada instância, considerando apenas os métodos que utilizam regiões simples. Já a coluna “Tempo Total (s)” traz o

Tabela 9 – Resultado da comparação entre todos os métodos de solução.

	Divisão	Ordenação	Decres.	Vitórias	Empates	Qualidade %	Tempo (s)
0	Vertical	Área	Sim	6	6	78.9961	3.0834e-03
1	Vertical	Área	Não	0	0	50.9443	2.4805e-03
2	Vertical	Perímetro	Sim	7	6	82.6210	2.3285e-03
3	Vertical	Perímetro	Não	0	0	51.2033	2.1488e-03
4	Vertical	Altura	Sim	4	4	70.7811	2.5334e-03
5	Vertical	Altura	Não	0	0	55.2624	2.0178e-03
6	Vertical	Largura	Sim	9	8	84.5497	2.4820e-03
7	Vertical	Largura	Não	0	0	47.9606	1.6620e-03
8	Vertical	Id	Sim	1	1	65.0670	3.1510e-03
9	Vertical	Id	Não	1	1	62.6394	2.4236e-03
10	Horizontal	Área	Sim	5	5	81.5022	5.9963e-03
11	Horizontal	Área	Não	1	1	44.9575	8.8805e-03
12	Horizontal	Perímetro	Sim	4	3	82.6390	4.7573e-03
13	Horizontal	Perímetro	Não	0	0	45.0368	8.5250e-03
14	Horizontal	Altura	Sim	4	4	79.2274	6.1442e-03
15	Horizontal	Altura	Não	2	2	43.4125	7.7093e-03
16	Horizontal	Largura	Sim	4	4	74.9317	7.6157e-03
17	Horizontal	Largura	Não	0	0	51.7897	1.0063e-02
18	Horizontal	Id	Sim	1	1	61.6848	6.5370e-03
19	Horizontal	Id	Não	2	2	64.9816	6.0956e-03
20	Maior área	Área	Sim	7	7	83.2483	1.3233e-02
21	Maior área	Área	Não	2	2	53.1636	1.7284e-02
22	Maior área	Perímetro	Sim	7	6	85.8682	1.2944e-02
23	Maior área	Perímetro	Não	1	1	53.7023	1.7675e-02
24	Maior área	Altura	Sim	4	4	78.5353	1.3269e-02
25	Maior área	Altura	Não	2	2	54.8203	1.0842e-02
26	Maior área	Largura	Sim	5	5	79.4570	1.4847e-02
27	Maior área	Largura	Não	0	0	62.4641	4.0100e-03
28	Maior área	Id	Sim	2	2	71.4787	1.4421e-02
29	Maior área	Id	Não	2	2	72.6713	1.4607e-02
30	Sobrepostas	Área	Sim	13	11	87.2957	6.4349e+00
31	Sobrepostas	Área	Não	0	0	62.5408	1.0376e+01
32	Sobrepostas	Perímetro	Sim	9	6	87.7336	6.3945e+00
33	Sobrepostas	Perímetro	Não	0	0	63.3835	1.0127e+01
34	Sobrepostas	Altura	Sim	6	5	81.2132	6.3465e+00
35	Sobrepostas	Altura	Não	2	2	64.9412	8.4619e+00
36	Sobrepostas	Largura	Sim	16	10	85.9266	8.4384e+00
37	Sobrepostas	Largura	Não	0	0	66.2589	1.0595e+01
38	Sobrepostas	Id	Sim	5	3	76.0408	8.4736e+00
39	Sobrepostas	Id	Não	4	4	74.9990	8.5603e+00

Fonte: feito pelo autor.

tempo total, em segundos, de execução para todos os métodos em todas as instâncias do conjunto.

É notável que ótimos resultados foram obtidos em um baixo período. No conjunto BKW, o qual é possível alocar todos os itens, levou mais tempo que os demais, pois possui a maior quantidade de instâncias e algumas com várias peças. Os conjuntos OF e OKP possuem poucas instâncias e, por isso, tiveram pouco tempo de execução. De forma geral, todos os conjuntos conseguiram uma excelente qualidade média, sendo o GCUT e NGCUT os únicos a ficarem abaixo de 90%.

Tabela 10 – Resultados para os conjuntos de instância.

Conjunto	Qualidade %	Itens %	Tempo Total (s)
BKW	94.4783	85.7782	1.2688e+01
GCUT	84.6060	20.0994	2.0189e-01
NGCUT	88.2085	35.0307	8.1531e-01
OF	92.0714	34.0580	1.0821e-02
OKP	93.9360	22.8232	1.1026e-01

Fonte: autor

4.6 Complexidade

As seções 4.3 e 4.4 deixaram claro que regiões onde a sobreposição pode ocorrer são mais custosas no tempo de execução. Isso acontece devido à complexidade do algoritmo de solução do modelo.

Em regiões simples, onde não podem acontecer sobreposições, é necessário verificar se o item a ser alocado cabe em uma região. O Algoritmo 1 é um pseudocódigo de solução para esse caso (código fonte feito em Python pode ser visto no Apêndice B). Onde $\mathcal{I}$ é o conjunto de itens a serem alocados e R são as regiões disponíveis no momento (ordenadas para respeitar a *bottom-left*). A função `CriarNovasRegiões()` elimina r, cria duas novas regiões com base no posicionamento de i e atualiza R com as novas regiões. A linha 3 verifica se um item i cabe em uma região r, se for possível, o item é alocado e a função `CriarNovasRegiões()` é chamada. Quando um item é alocado, o conjunto $\mathcal{I}'$ (conjunto de itens alocados) é atualizado. Após, um `break` é acionado para avançar ao próximo item. No pior caso, todas as peças só conseguiriam ser alocadas na última região checada. É possível descobrir o número máximo de regiões disponíveis no momento de alocação do item i da fila.

Algoritmo 1: Solução usando regiões simples.

```
1 forall i ∈ I do
2     forall r ∈ R do
3         if i.largura ≤ r.largura and i.altura ≤ r.altura then
4             aloca item i na região r;
              /* elimina r e cria duas novas regiões com base em i      */
5             CriarNovasRegioes(i, r);
6             break;
7         end
8     end
9 end
```

Para o primeiro item, existe somente uma região, o próprio espaço. Após alocar o primeiro item, tem-se duas regiões disponíveis para posicionar o segundo (ver seção 4.1). Ao posicionar uma peça o número de regiões sempre é acrescido em um (uma região é eliminada e duas são criadas). Com isso, é possível afirmar que para alocar o item i

será preciso verificar no máximo i regiões. Somando o máximo de regiões para cada item chega-se no número máximo de regiões a serem verificadas no pior caso. Considerando n itens, tem-se a Equação 1.

$$\sum_{i=1}^{n} i \tag{1}$$

Como a soma de 1 até n pode ser reescrita utilizando a Equação 2 (MERCA, 2015), tem-se que o número máximo de regiões a serem verificadas para solucionar o modelo é $\frac{n^2+n}{2}$.

$$\sum_{k=1}^{n} k = \frac{n(n+1)}{2} \tag{2}$$

Para regiões complexas, além desse número de regiões, ainda é necessário verificar sobreposições, conforme o Algoritmo 2. O conjunto $\mathcal{I}'$ se refere aos itens já alocados no recipiente e a função `Sobreposicao` verifica se existe sobreposição entre duas peças. Agora, para cada região $r \in R$ onde é possível alocar o item $i \in \mathcal{I}$, também é preciso verificar todos itens $i' \in \mathcal{I}'$. A linha 6 garante que i e i' são diferentes e, caso haja sobreposição, o item i é removido da região r (foi alocado na linha 4) e um `break` é acionado, pois não existe necessidade de verificar outros itens $i' \in \mathcal{I}'$. Na linha 11 é checado se as coordenadas do item i existem, ou seja, se o item ainda está posicionado (não houve sobreposição), então a função `CriarNovasRegiões` é chamada.

Algoritmo 2: Solução usando regiões complexas.

```
1  forall i ∈ I do
2      forall r ∈ R do
3          if i.largura ≤ r.largura and i.altura ≤ r.altura then
4              aloca item i na região r;
5              forall i' ∈ I' do
6                  if i ≠ i' and Sobreposicao(i, i') then
7                      remove item i na região r;
8                      break;
9                  end
10             end
11             if i.coordenadas ≠ null then
                   /* elimina r e cria duas novas regiões com base em i   */
12                 CriarNovasRegioes(i, r);
13                 break;
14             end
15         end
16     end
17 end
```

No pior caso, ao alocar o item i, para cada região será preciso verificar sobreposições com todos os itens $i-1$ (Equação 3).

$$\sum_{i=1}^{n} i(i-1) \tag{3}$$

Usando a Fórmula de Faulhaber (MERCA, 2015) é possível reescrever a soma dos quadrados dos n primeiros números inteiros como mostra a Equação 4.

$$\sum_{k=1}^{n} k^2 = \frac{n(n+1)(2n+1)}{6} \tag{4}$$

Assim, ao reescrever a Equação 3 utilizando as Equações 2 e 4, tem-se que, no pior caso, será necessário verificar se existe sobreposição entre peças $\frac{n^3-n}{3}$ vezes. Para $n = 3152$, seriam quase cinco milhões de regiões e mais 10 bilhões de sobreposições a serem checadas, explicando o tempo elevado ao executar um método de solução utilizando regiões complexas.

Conclusão e trabalhos futuros

No trabalho foram avaliados experimentalmente 40 métodos de solução baseados na heurística *bottom-left* para o problema de empacotamento de retângulos, considerando a versão da mochila e com o valor dos itens sua própria área. O resultado mais óbvio obtido foi a supremacia da ordenação decrescente sobre a crescente (seção 4.1), não compensando utilizar a segunda para solucionar o problema.

O alto uso da ordenação decrescente pela área na literatura (CHEN et al., 2019) foi justificado pelos resultados, já que essa composição conseguiu alguns dos melhores números (seção 4.2). Os resultados também indicam que qualquer critério de ordenação obtêm melhores resultados do que deixar a fila desordenada (ordenação por *id*). Mas a alta competitividade de outros critérios de ordenação, esses não tão comuns na literatura, conseguindo melhores resultados em algumas instâncias, mostra que ainda há espaço para outras formas de ordenação.

Em relação às diferentes regiões criadas, por mais que regiões complexas tenham obtido melhores resultados (seção 4.3), não compensa utilizá-las. Ao executar todos os métodos de regiões simples é possível conseguir melhores resultados e em menor tempo, quando comparado ao método de ordenação decrescente pela área e usando regiões complexas (ver a Tabela 8 na seção 4.3). Isso devido à complexidade do algoritmo que precisa verificar a existência de sobreposições de peças em regiões complexas (seção 4.6).

Como trabalho futuro, cabe a investigação, a fundo, de abordagens meta-heurísticas para a *bottom-left*, considerando os outros critérios promissores que não seja a área. Buscando identificar se há alguma característica na instância que proporcione melhores soluções com outros critérios, a suspeita inicial é que esteja relacionado a quão quadrados são os itens a serem alocados no recipiente.

Outros trabalhos possíveis são: expandir o algoritmo de solução para o caso 3D (ou N-dimensões) e possibilitar a resolução de outras classes do problema (2D-SPP, 2D-CSP, 2D-OPP) e suas variantes. Também é possível implementar métodos de solução da literatura diferentes da *bottom-left*, sejam eles exatos ou heurísticos. Com isso, seria possível criar uma ampla biblioteca em Python para problemas de corte e empacotamento.

Por fim, seria interessante buscar soluções quânticas para o problema. No momento, existem poucos trabalhos disponíveis na literatura que abordem o tema. Um dos mais recentes, feito por ANDOIN et al. (2022), traz uma heurística híbrida clássica-quântica para o *bin packing problem* (BPP) unidimensional, conseguindo com essa nova heurística bons resultados para o problema. Outro resultado interessante foi o de FUJIMURA (2021), com um algoritmo quântico foi possível chegar a uma complexidade $3(\log_2 k)n$ para o BPP, onde k representa o número de caixas e n o número de peças. Trabalhos como esses ainda são raros, principalmente para dimensões maiores, mas os poucos que existem indicam

bons resultados para soluções quânticas, ou híbridas, do problema de empacotamento.

Referências

ANDOIN, Mikel Garcia de et al. Hybrid quantum-classical heuristic for the bin packing problem. In: PROCEEDINGS of the Genetic and Evolutionary Computation Conference Companion. [S.l.: s.n.], 2022. P. 2214–2222.

ARENALES, Marcos et al. **Pesquisa Operacional**. [S.l.]: Elsevier, 2007.

BAKER, Brenda S; COFFMAN JR, E G; RIVEST, Ronald L. Orthogonal Packings in Two Dimensions. **SIAM Journal on Computing**, Society for Industrial e Applied Mathematics, v. 9, n. 4, p. 846, 1980.

BARTMEYER, Petra Maria et al. Aprendizado por reforço aplicado ao problema de empacotamento de peças irregulares em faixas. **Anais**, 2021. Disponível em: <https://repositorio.usp.br/directbitstream/455094df-864a-4fad-8a97-c5f59fd3d6ca/3051981.pdf>.

BEASLEY, J E. Algorithms for unconstrained two-dimensional guillotine cutting. **Journal of the Operational Research Society**, Taylor & Francis, v. 36, n. 4, p. 297–306, 1985.

_____. An Exact Two-Dimensional Non-Guillotine Cutting Tree Search Procedure. **Operations Research**, v. 33, n. 1, 1985.

BELLUZZO, Luciano; MORABITO, Reinaldo. Otimização nos padrões de corte de chapas de fibra de madeira reconstituída: um estudo de caso. **Pesquisa Operacional**, SciELO Brasil, v. 25, p. 391–415, 2005. Disponível em: <https://www.scielo.br/j/pope/a/tTXXckvGTHbDfZQkmzCqdkp>.

BURKE, E K; KENDALL, G; WHITWELL, G. A new placement heuristic for the orthogonal stock-cutting problem. **Operations Research**, INFORMS, v. 52, n. 4, p. 655–671, 2004.

CALAFIORE, Giuseppe C; EL GHAOUI, Laurent. **Optimization models**. [S.l.]: Cambridge university press, 2014.

CASTELLUCCI, Pedro Belin. **Consolidation problems in freight transportation systems: mathematical models and algorithms**. 2019. Tese (Doutorado) – Universidade de São Paulo. Disponível em: <https://pdfs.semanticscholar.org/90e7/bd898951e1350c2694478b63fbcde508e189.pdf>.

CAVALI, Roberto. Problemas de corte e empacotamento na indústria de Móveis: um estudo de caso. Universidade Estadual Paulista (Unesp), 2004. Disponível em: <https://repositorio.unesp.br/bitstream/handle/11449/94286/cavali_r_me_sjrp.pdf>.

CHEHRAZAD, Sahar; ROOSE, Dirk; WAUTERS, Tony. A fast and scalable bottom-left-fill algorithm to solve nesting problems using a semi-discrete representation. **European Journal of Operational Research**, Elsevier, v. 300, n. 3, p. 809–826, 2022.

CHEN, Mao et al. An efficient deterministic heuristic algorithm for the rectangular packing problem. **Computers & Industrial Engineering**, Elsevier, v. 137, p. 106097, 2019.

CINTRA, G F et al. Algorithms for two-dimensional cutting stock and strip packing problems using dynamic programming and column generation q. **European Journal of Operational Research**, v. 191, p. 61–85, 2008.

CÔTÉ, Jean-François; DELL'AMICO, Mauro; IORI, Manuel. Combinatorial Benders' cuts for the strip packing problem. **Operations Research**, INFORMS, v. 62, n. 3, p. 643–661, 2014.

DELORME, Maxence; IORI, Manuel; MARTELLO, Silvano. Bin packing and cutting stock problems: Mathematical models and exact algorithms. **European Journal of Operational Research**, Elsevier, v. 255, n. 1, p. 1–20, 2016.

_____. Logic based Benders' decomposition for orthogonal stock cutting problems. **Computers & Operations Research**, Elsevier, v. 78, p. 290–298, 2017.

DYCKHOFF, Harald. A typology of cutting and packing problems. **European journal of operational research**, Elsevier, v. 44, n. 2, p. 145–159, 1990.

FEKETE, Sándor P; SCHEPERS, Jörg. A new exact algorithm for general orthogonal d-dimensional knapsack problems. In: SPRINGER. ALGORITHMS—ESA'97: 5th Annual European Symposium Graz, Austria, September 15–17, 1997 Proceedings 5. [S.l.: s.n.], 1997. P. 144–156.

FIRAT, Hüseyin; ALPASLAN, Nuh. An effective approach to the two-dimensional rectangular packing problem in the manufacturing industry. **Computers & Industrial Engineering**, Elsevier, v. 148, p. 106687, 2020.

FUJIMURA, Toru. Quantum Algorithm for Bin-Packing Problem by Quarter Method. **Global Journal of Pure and Applied Mathematics**, v. 17, n. 1, p. 9–15, 2021.

FURINI, Fabio; MALAGUTI, Enrico; THOMOPULOS, Dimitri. Modeling two-dimensional guillotine cutting problems via integer programming. **INFORMS Journal on Computing**, INFORMS, v. 28, n. 4, p. 736–751, 2016.

HILLIER, Frederick S. Introduction to operations research, 1967.

HOPPER, E B C H; TURTON, Brian C H. An empirical investigation of meta-heuristic and heuristic algorithms for a 2D packing problem. **European Journal of Operational Research**, Elsevier, v. 128, n. 1, p. 34–57, 2001.

HOPPER, Eva; TURTON, Brian CH. A review of the application of meta-heuristic algorithms to 2D strip packing problems. **Artificial Intelligence Review**, Springer, v. 16, p. 257–300, 2001.

HUANG, Wenqi; CHEN, Duanbing. An efficient heuristic algorithm for rectangle-packing problem. **Simulation Modelling Practice and Theory**, Elsevier, v. 15, n. 10, p. 1356–1365, 2007.

IORI, Manuel; DE LIMA, Vinícius L. et al. Exact solution techniques for two-dimensional cutting and packing. **European Journal of Operational Research**, v. 289, n. 2, p. 399–415, 2021. ISSN 0377-2217. DOI: https://doi.org/10.1016/j.ejor.2020.06.050. Disponível em: <https://www.sciencedirect.com/science/article/pii/S0377221720306111>.

IORI, Manuel; LIMA, Vinícius Loti de et al. 2DPackLib: a two-dimensional cutting and packing library. **Optimization Letters**, Springer, v. 16, n. 2, p. 471–480, 2022.

KENMOCHI, Mitsutoshi et al. Exact algorithms for the two-dimensional strip packing problem with and without rotations. **European Journal of Operational Research**, Elsevier, v. 198, n. 1, p. 73–83, 2009.

MARTIN, Mateus et al. Models for the two-dimensional rectangular single large placement problem with guillotine cuts and constrained pattern. **International Transactions in Operational Research**, Wiley Online Library, v. 27, n. 2, p. 767–793, 2020.

MERCA, Mircea. An alternative to Faulhaber's formula. **The American Mathematical Monthly**, Taylor & Francis, v. 122, n. 6, p. 599–601, 2015.

MICHALEWICZ, Zbigniew; FOGEL, David B. **How to solve it: modern heuristics**. [S.l.]: Springer Science & Business Media, 2013.

MORABITO NETO, Reinaldo; WIDMER, Joao Alexandre. **Abordagem em grafo-e-ou para o problema do empacotamento: aplicacao ao carregamento de paletes e conteineres**. 1992. Tese (Doutorado). Disponível em: <https://repositorio.usp.br/item/000734666>.

MRAD, Mehdi. An arc flow-based optimization approach for the two-stage guillotine strip cutting problem. **Journal of the Operational Research Society**, Taylor & Francis, v. 66, n. 11, p. 1850–1859, 2015.

OLIVEIRA, Jose Fernando; FERREIRA, Jose Soeiro. An improved version of Wang's algorithm for two-dimensional cutting problems. **European Journal of Operational Research**, Elsevier, v. 44, n. 2, p. 256–266, 1990.

QUEIROZ, Layane Rodrigues de Souza. **Estudo de problemas de corte de itens irregulares com incertezas**. 2022. Tese (Doutorado) – Universidade de São Paulo. Disponível em: <https://www.teses.usp.br/teses/disponiveis/55/55134/tde-10032022-110656/en.php>.

RAKOTONIRAINY, Rosephine G; VUUREN, Jan H van. Improved metaheuristics for the two-dimensional strip packing problem. **Applied Soft Computing**, Elsevier, v. 92, p. 106268, 2020.

SILVA, Lorrany Cristina da; QUEIROZ, Thiago Alves de;
TOLEDO, Franklina Maria Bragion de. Integer formulations for the integrated vehicle routing problem with two-dimensional packing constraints. **Pesquisa Operacional**, SciELO Brasil, v. 42, 2022.

VELASCO, André Soares; UCHOA, Eduardo. Improved state space relaxation for constrained two-dimensional guillotine cutting problems. **European Journal of Operational Research**, Elsevier, v. 272, n. 1, p. 106–120, 2019.

WÄSCHER, Gerhard; HAUSSNER, Heike; SCHUMANN, Holger. An improved typology of cutting and packing problems. **European journal of operational research**, Elsevier, v. 183, n. 3, p. 1109–1130, 2007.

WEI, Lijun et al. A skyline heuristic for the 2D rectangular packing and strip packing problems. **European Journal of Operational Research**, Elsevier, v. 215, n. 2, p. 337–346, 2011.

WOLSEY, Laurence A. **Integer programming**. [S.l.]: John Wiley & Sons, 2020.

APÊNDICE A – Resultados das instâncias

A.1 BKW

Tabela 11 – Resultados da instância BKW01.

Instância	Divisão	Ordenação	Decrescente	Qualidade %	Tempo (s)	Itens %
BKW01	Vertical	Área	Sim	100.0000	7.4816e-05	100
BKW01	Vertical	Área	Não	60.0000	8.3637e-05	90
BKW01	Vertical	Perímetro	Sim	100.0000	7.1335e-05	100
BKW01	Vertical	Perímetro	Não	60.0000	8.4829e-05	90
BKW01	Vertical	Altura	Sim	60.0000	9.0504e-05	90
BKW01	Vertical	Altura	Não	24.0000	8.1444e-05	80
BKW01	Vertical	Largura	Sim	100.0000	6.1607e-05	100
BKW01	Vertical	Largura	Não	60.0000	8.6117e-05	90
BKW01	Vertical	Id	Sim	60.0000	1.0233e-04	90
BKW01	Vertical	Id	Não	60.0000	1.0214e-04	90
BKW01	Horizontal	Área	Sim	100.0000	6.5231e-05	100
BKW01	Horizontal	Área	Não	12.7500	7.0524e-05	60
BKW01	Horizontal	Perímetro	Sim	100.0000	6.4421e-05	100
BKW01	Horizontal	Perímetro	Não	12.7500	7.0429e-05	60
BKW01	Horizontal	Altura	Sim	100.0000	7.0000e-05	100
BKW01	Horizontal	Altura	Não	12.7500	7.0763e-05	60
BKW01	Horizontal	Largura	Sim	88.7500	7.0429e-05	80
BKW01	Horizontal	Largura	Não	24.0000	8.9598e-05	80
BKW01	Horizontal	Id	Sim	17.7500	8.0585e-05	70
BKW01	Horizontal	Id	Não	52.7500	7.1764e-05	70
BKW01	Maior área	Área	Sim	100.0000	1.5569e-04	100
BKW01	Maior área	Área	Não	60.0000	1.8935e-04	90
BKW01	Maior área	Perímetro	Sim	100.0000	1.5464e-04	100
BKW01	Maior área	Perímetro	Não	60.0000	1.9536e-04	90
BKW01	Maior área	Altura	Sim	98.7500	1.5717e-04	90
BKW01	Maior área	Altura	Não	60.0000	1.8964e-04	90
BKW01	Maior área	Largura	Sim	88.7500	1.4067e-04	80
BKW01	Maior área	Largura	Não	60.0000	1.8468e-04	90
BKW01	Maior área	Id	Sim	60.0000	1.8449e-04	90
BKW01	Maior área	Id	Não	52.7500	2.3007e-04	70
BKW01	Sobrepostas	Área	Sim	100.0000	1.6430e-03	100
BKW01	Sobrepostas	Área	Não	60.0000	1.6438e-03	90
BKW01	Sobrepostas	Perímetro	Sim	100.0000	1.6285e-03	100
BKW01	Sobrepostas	Perímetro	Não	60.0000	1.6748e-03	90
BKW01	Sobrepostas	Altura	Sim	60.0000	1.3000e-03	90
BKW01	Sobrepostas	Altura	Não	60.0000	1.9024e-03	90
BKW01	Sobrepostas	Largura	Sim	100.0000	2.0637e-03	100
BKW01	Sobrepostas	Largura	Não	60.0000	1.8050e-03	90
BKW01	Sobrepostas	Id	Sim	60.0000	1.6782e-03	90
BKW01	Sobrepostas	Id	Não	52.7500	8.4772e-04	70

Fonte: feito pelo autor.

Tabela 12 – Resultados da instância BKW02.

Instância	Divisão	Ordenação	Decrescente	Qualidade %	Tempo (s)	Itens %
BKW02	Vertical	Área	Sim	61.0000	1.5020e-04	70
BKW02	Vertical	Área	Não	29.1333	1.7204e-04	60
BKW02	Vertical	Perímetro	Sim	63.4000	1.7619e-04	75
BKW02	Vertical	Perímetro	Não	29.1333	1.2631e-04	60
BKW02	Vertical	Altura	Sim	57.2667	1.5454e-04	65
BKW02	Vertical	Altura	Não	34.2000	1.4596e-04	65
BKW02	Vertical	Largura	Sim	81.0667	1.7471e-04	90
BKW02	Vertical	Largura	Não	26.7333	1.1444e-04	55
BKW02	Vertical	Id	Sim	40.7333	1.4687e-04	65
BKW02	Vertical	Id	Não	57.4667	1.8272e-04	60
BKW02	Horizontal	Área	Sim	82.6667	1.6789e-04	65
BKW02	Horizontal	Área	Não	40.1333	1.7276e-04	70
BKW02	Horizontal	Perímetro	Sim	82.6667	1.6990e-04	65
BKW02	Horizontal	Perímetro	Não	49.3333	1.7524e-04	75
BKW02	Horizontal	Altura	Sim	82.5333	1.9927e-04	80
BKW02	Horizontal	Altura	Não	47.4667	1.6994e-04	70
BKW02	Horizontal	Largura	Sim	73.4000	1.8406e-04	85
BKW02	Horizontal	Largura	Não	60.2000	2.5406e-04	85
BKW02	Horizontal	Id	Sim	50.4000	1.6894e-04	75
BKW02	Horizontal	Id	Não	78.5333	1.8544e-04	80
BKW02	Maior área	Área	Sim	71.9333	4.1785e-04	85
BKW02	Maior área	Área	Não	43.8667	3.2215e-04	75
BKW02	Maior área	Perímetro	Sim	71.9333	3.7384e-04	85
BKW02	Maior área	Perímetro	Não	60.2000	3.5372e-04	85
BKW02	Maior área	Altura	Sim	68.2000	3.5043e-04	80
BKW02	Maior área	Altura	Não	49.3333	3.3765e-04	75
BKW02	Maior área	Largura	Sim	79.4667	3.5648e-04	90
BKW02	Maior área	Largura	Não	55.2667	3.1128e-04	75
BKW02	Maior área	Id	Sim	54.1333	3.2859e-04	80
BKW02	Maior área	Id	Não	76.8000	3.8748e-04	85
BKW02	Sobrepostas	Área	Sim	85.8667	5.5543e-03	70
BKW02	Sobrepostas	Área	Não	43.5333	1.1909e-02	75
BKW02	Sobrepostas	Perímetro	Sim	89.8667	6.3050e-03	80
BKW02	Sobrepostas	Perímetro	Não	49.9333	1.0754e-02	80
BKW02	Sobrepostas	Altura	Sim	82.0000	7.3275e-03	85
BKW02	Sobrepostas	Altura	Não	59.1333	1.5066e-02	85
BKW02	Sobrepostas	Largura	Sim	96.5333	1.5720e-02	90
BKW02	Sobrepostas	Largura	Não	65.5333	1.8248e-02	80
BKW02	Sobrepostas	Id	Sim	64.3333	1.3025e-02	85
BKW02	Sobrepostas	Id	Não	76.8000	7.9223e-03	85

Fonte: feito pelo autor.

Tabela 13 – Resultados da instância BKW03.

Instância	Divisão	Ordenação	Decrescente	Qualidade %	Tempo (s)	Itens %
BKW03	Vertical	Área	Sim	84.1333	2.2397e-04	76.67
BKW03	Vertical	Área	Não	52.0000	2.4667e-04	80.00
BKW03	Vertical	Perímetro	Sim	80.1333	2.4357e-04	73.33
BKW03	Vertical	Perímetro	Não	52.0000	2.1973e-04	80.00
BKW03	Vertical	Altura	Sim	75.4667	2.3298e-04	73.33
BKW03	Vertical	Altura	Não	55.3333	2.5473e-04	83.33
BKW03	Vertical	Largura	Sim	88.6000	2.3975e-04	80.00
BKW03	Vertical	Largura	Não	52.0000	2.2697e-04	80.00
BKW03	Vertical	Id	Sim	65.9333	2.4056e-04	83.33
BKW03	Vertical	Id	Não	66.4000	2.5501e-04	83.33
BKW03	Horizontal	Área	Sim	88.8000	3.0799e-04	76.67
BKW03	Horizontal	Área	Não	20.2000	2.3384e-04	56.67
BKW03	Horizontal	Perímetro	Sim	93.0000	2.9578e-04	90.00
BKW03	Horizontal	Perímetro	Não	20.2000	2.2769e-04	56.67
BKW03	Horizontal	Altura	Sim	83.0667	3.4061e-04	83.33
BKW03	Horizontal	Altura	Não	20.2000	2.4662e-04	56.67
BKW03	Horizontal	Largura	Sim	58.1333	2.8653e-04	80.00
BKW03	Horizontal	Largura	Não	39.6667	3.0808e-04	76.67
BKW03	Horizontal	Id	Sim	67.1333	3.2792e-04	86.67
BKW03	Horizontal	Id	Não	33.8667	4.2829e-04	66.67
BKW03	Maior área	Área	Sim	92.4000	5.5542e-04	83.33
BKW03	Maior área	Área	Não	35.0667	5.0235e-04	73.33
BKW03	Maior área	Perímetro	Sim	92.6667	5.8484e-04	93.33
BKW03	Maior área	Perímetro	Não	38.1333	4.8871e-04	73.33
BKW03	Maior área	Altura	Sim	90.4000	5.3940e-04	83.33
BKW03	Maior área	Altura	Não	38.1333	5.3577e-04	73.33
BKW03	Maior área	Largura	Sim	78.3333	6.4588e-04	93.33
BKW03	Maior área	Largura	Não	55.3333	5.1866e-04	83.33
BKW03	Maior área	Id	Sim	72.8667	6.0782e-04	90.00
BKW03	Maior área	Id	Não	62.9333	5.9104e-04	90.00
BKW03	Sobrepostas	Área	Sim	93.8000	2.9598e-02	90.00
BKW03	Sobrepostas	Área	Não	58.3333	5.1224e-02	86.67
BKW03	Sobrepostas	Perímetro	Sim	92.6667	3.0483e-02	93.33
BKW03	Sobrepostas	Perímetro	Não	55.7333	4.5757e-02	86.67
BKW03	Sobrepostas	Altura	Sim	89.7333	2.5991e-02	83.33
BKW03	Sobrepostas	Altura	Não	51.4667	5.3401e-02	80.00
BKW03	Sobrepostas	Largura	Sim	93.8000	5.1422e-02	86.67
BKW03	Sobrepostas	Largura	Não	71.6000	5.5932e-02	93.33
BKW03	Sobrepostas	Id	Sim	78.9333	5.2150e-02	93.33
BKW03	Sobrepostas	Id	Não	80.0000	3.4366e-02	93.33

Fonte: feito pelo autor.

Tabela 14 – Resultados da instância BKW04.

Instância	Divisão	Ordenação	Decrescente	Qualidade %	Tempo (s)	Itens %
BKW04	Vertical	Área	Sim	91.0781	4.1385e-04	85.00
BKW04	Vertical	Área	Não	46.2812	3.6044e-04	92.50
BKW04	Vertical	Perímetro	Sim	93.4062	4.1132e-04	90.00
BKW04	Vertical	Perímetro	Não	44.4688	3.5396e-04	90.00
BKW04	Vertical	Altura	Sim	88.4375	3.8142e-04	80.00
BKW04	Vertical	Altura	Não	48.2812	4.1633e-04	95.00
BKW04	Vertical	Largura	Sim	94.5000	4.8122e-04	87.50
BKW04	Vertical	Largura	Não	42.2812	3.3688e-04	87.50
BKW04	Vertical	Id	Sim	46.2812	3.9468e-04	92.50
BKW04	Vertical	Id	Não	81.5938	4.1642e-04	85.00
BKW04	Horizontal	Área	Sim	87.3125	5.5556e-04	85.00
BKW04	Horizontal	Área	Não	18.6875	4.8118e-04	70.00
BKW04	Horizontal	Perímetro	Sim	85.7812	3.6364e-04	77.50
BKW04	Horizontal	Perímetro	Não	17.5625	3.9864e-04	65.00
BKW04	Horizontal	Altura	Sim	74.6094	4.3383e-04	90.00
BKW04	Horizontal	Altura	Não	17.5625	4.1633e-04	65.00
BKW04	Horizontal	Largura	Sim	75.6406	4.2057e-04	85.00
BKW04	Horizontal	Largura	Não	18.6875	4.0050e-04	70.00
BKW04	Horizontal	Id	Sim	33.8594	4.3645e-04	85.00
BKW04	Horizontal	Id	Não	62.7656	4.4909e-04	85.00
BKW04	Maior área	Área	Sim	86.5000	8.5955e-04	90.00
BKW04	Maior área	Área	Não	38.5938	7.4091e-04	90.00
BKW04	Maior área	Perímetro	Sim	88.1250	8.7609e-04	92.50
BKW04	Maior área	Perímetro	Não	38.5938	7.6685e-04	90.00
BKW04	Maior área	Altura	Sim	92.7969	8.0295e-04	90.00
BKW04	Maior área	Altura	Não	49.3125	7.7963e-04	90.00
BKW04	Maior área	Largura	Sim	90.3125	9.8920e-04	95.00
BKW04	Maior área	Largura	Não	48.2812	8.5306e-04	95.00
BKW04	Maior área	Id	Sim	61.6719	8.5254e-04	95.00
BKW04	Maior área	Id	Não	85.5312	1.1236e-03	92.50
BKW04	Sobrepostas	Área	Sim	96.1875	4.5247e-02	82.50
BKW04	Sobrepostas	Área	Não	44.6719	1.1442e-01	87.50
BKW04	Sobrepostas	Perímetro	Sim	95.0625	5.5670e-02	75.00
BKW04	Sobrepostas	Perímetro	Não	48.2812	1.3407e-01	95.00
BKW04	Sobrepostas	Altura	Sim	94.0000	5.2721e-02	92.50
BKW04	Sobrepostas	Altura	Não	58.1562	1.0393e-01	95.00
BKW04	Sobrepostas	Largura	Sim	93.8750	1.1813e-01	95.00
BKW04	Sobrepostas	Largura	Não	63.7812	1.5432e-01	97.50
BKW04	Sobrepostas	Id	Sim	48.2812	5.2246e-02	95.00
BKW04	Sobrepostas	Id	Não	88.7656	8.4950e-02	92.50

Fonte: feito pelo autor.

Tabela 15 – Resultados da instância BKW05.

Instância	Divisão	Ordenação	Decrescente	Qualidade %	Tempo (s)	Itens %
BKW05	Vertical	Área	Sim	64.9200	5.2748e-04	78
BKW05	Vertical	Área	Não	22.2700	3.1900e-04	58
BKW05	Vertical	Perímetro	Sim	74.0100	8.0299e-04	82
BKW05	Vertical	Perímetro	Não	22.2700	3.2158e-04	58
BKW05	Vertical	Altura	Sim	55.1000	5.0688e-04	64
BKW05	Vertical	Altura	Não	30.8300	4.8060e-04	68
BKW05	Vertical	Largura	Sim	87.0500	5.4159e-04	82
BKW05	Vertical	Largura	Não	22.2700	3.1123e-04	58
BKW05	Vertical	Id	Sim	27.3500	4.3411e-04	64
BKW05	Vertical	Id	Não	43.1400	5.5504e-04	72
BKW05	Horizontal	Área	Sim	90.4700	4.9443e-04	70
BKW05	Horizontal	Área	Não	41.1000	5.4550e-04	78
BKW05	Horizontal	Perímetro	Sim	90.8800	5.5623e-04	80
BKW05	Horizontal	Perímetro	Não	41.1000	6.1522e-04	78
BKW05	Horizontal	Altura	Sim	77.4200	6.2652e-04	84
BKW05	Horizontal	Altura	Não	25.0800	3.4971e-04	46
BKW05	Horizontal	Largura	Sim	83.9600	5.4994e-04	78
BKW05	Horizontal	Largura	Não	42.8100	6.8064e-04	80
BKW05	Horizontal	Id	Sim	55.2900	5.9762e-04	74
BKW05	Horizontal	Id	Não	77.5000	5.7979e-04	86
BKW05	Maior área	Área	Sim	92.1400	1.2037e-03	84
BKW05	Maior área	Área	Não	50.9200	1.0511e-03	86
BKW05	Maior área	Perímetro	Sim	93.2000	1.2055e-03	90
BKW05	Maior área	Perímetro	Não	50.9200	1.0590e-03	86
BKW05	Maior área	Altura	Sim	80.4800	1.2160e-03	90
BKW05	Maior área	Altura	Não	44.7700	1.0118e-03	74
BKW05	Maior área	Largura	Sim	89.7000	1.0605e-03	84
BKW05	Maior área	Largura	Não	52.9200	1.3261e-03	86
BKW05	Maior área	Id	Sim	62.7500	1.0735e-03	92
BKW05	Maior área	Id	Não	88.0400	1.0009e-03	92
BKW05	Sobrepostas	Área	Sim	92.0500	1.7748e-01	76
BKW05	Sobrepostas	Área	Não	62.7500	2.4776e-01	92
BKW05	Sobrepostas	Perímetro	Sim	93.1300	1.9128e-01	72
BKW05	Sobrepostas	Perímetro	Não	67.8800	2.7685e-01	94
BKW05	Sobrepostas	Altura	Sim	80.4800	2.0130e-01	90
BKW05	Sobrepostas	Altura	Não	62.5900	2.8900e-01	92
BKW05	Sobrepostas	Largura	Sim	93.5600	1.9373e-01	76
BKW05	Sobrepostas	Largura	Não	59.5700	2.8289e-01	90
BKW05	Sobrepostas	Id	Sim	67.8800	2.3627e-01	94
BKW05	Sobrepostas	Id	Não	88.0400	1.5671e-01	92

Fonte: feito pelo autor.

Tabela 16 – Resultados da instância BKW06.

Instância	Divisão	Ordenação	Decrescente	Qualidade %	Tempo (s)	Itens %
BKW06	Vertical	Área	Sim	70.0200	6.5041e-04	80.00
BKW06	Vertical	Área	Não	33.8600	4.4980e-04	85.00
BKW06	Vertical	Perímetro	Sim	70.5800	7.0281e-04	90.00
BKW06	Vertical	Perímetro	Não	33.8600	4.4298e-04	85.00
BKW06	Vertical	Altura	Sim	69.3800	6.1293e-04	86.67
BKW06	Vertical	Altura	Não	33.8600	4.8304e-04	85.00
BKW06	Vertical	Largura	Sim	86.9000	5.1322e-04	56.67
BKW06	Vertical	Largura	Não	32.4200	4.2357e-04	83.33
BKW06	Vertical	Id	Sim	45.2600	5.3544e-04	88.33
BKW06	Vertical	Id	Não	56.1800	5.4092e-04	88.33
BKW06	Horizontal	Área	Sim	95.7400	5.9204e-04	81.67
BKW06	Horizontal	Área	Não	13.2400	6.1703e-04	61.67
BKW06	Horizontal	Perímetro	Sim	96.0600	6.4635e-04	88.33
BKW06	Horizontal	Perímetro	Não	13.2400	6.2704e-04	61.67
BKW06	Horizontal	Altura	Sim	81.6000	9.8410e-04	93.33
BKW06	Horizontal	Altura	Não	22.8800	6.5327e-04	65.00
BKW06	Horizontal	Largura	Sim	88.0200	7.4892e-04	86.67
BKW06	Horizontal	Largura	Não	22.4600	7.7348e-04	76.67
BKW06	Horizontal	Id	Sim	45.1400	7.5245e-04	83.33
BKW06	Horizontal	Id	Não	48.3200	7.3385e-04	81.67
BKW06	Maior área	Área	Sim	95.9200	1.2887e-03	85.00
BKW06	Maior área	Área	Não	35.7800	1.1890e-03	86.67
BKW06	Maior área	Perímetro	Sim	95.3400	1.2143e-03	85.00
BKW06	Maior área	Perímetro	Não	35.7800	1.2651e-03	86.67
BKW06	Maior área	Altura	Sim	72.5000	1.4461e-03	91.67
BKW06	Maior área	Altura	Não	28.5200	1.2993e-03	80.00
BKW06	Maior área	Largura	Sim	92.7600	1.4110e-03	91.67
BKW06	Maior área	Largura	Não	42.5000	1.2162e-03	88.33
BKW06	Maior área	Id	Sim	77.4000	1.4139e-03	95.00
BKW06	Maior área	Id	Não	72.5000	1.4293e-03	91.67
BKW06	Sobrepostas	Área	Sim	96.9400	1.6137e-01	78.33
BKW06	Sobrepostas	Área	Não	56.9000	3.8633e-01	90.00
BKW06	Sobrepostas	Perímetro	Sim	97.3800	1.8133e-01	81.67
BKW06	Sobrepostas	Perímetro	Não	60.0000	4.0037e-01	91.67
BKW06	Sobrepostas	Altura	Sim	72.5000	1.6896e-01	91.67
BKW06	Sobrepostas	Altura	Não	57.8800	3.6767e-01	91.67
BKW06	Sobrepostas	Largura	Sim	98.1400	3.6012e-01	95.00
BKW06	Sobrepostas	Largura	Não	42.5000	4.1001e-01	88.33
BKW06	Sobrepostas	Id	Sim	63.0000	2.9472e-01	93.33
BKW06	Sobrepostas	Id	Não	72.5000	2.4477e-01	91.67

Fonte: feito pelo autor.

Tabela 17 – Resultados da instância BKW07.

Instância	Divisão	Ordenação	Decrescente	Qualidade %	Tempo (s)	Itens %
BKW07	Vertical	Área	Sim	78.8000	7.6075e-04	84.29
BKW07	Vertical	Área	Não	41.5375	6.2904e-04	85.71
BKW07	Vertical	Perímetro	Sim	75.7750	7.0801e-04	84.29
BKW07	Vertical	Perímetro	Não	40.6375	6.2118e-04	84.29
BKW07	Vertical	Altura	Sim	67.8000	7.4782e-04	81.43
BKW07	Vertical	Altura	Não	42.0875	7.0863e-04	87.14
BKW07	Vertical	Largura	Sim	91.6625	8.6551e-04	95.71
BKW07	Vertical	Largura	Não	38.7250	5.4336e-04	80.00
BKW07	Vertical	Id	Sim	66.6000	8.4338e-04	88.57
BKW07	Vertical	Id	Não	46.7250	7.8974e-04	84.29
BKW07	Horizontal	Área	Sim	74.0125	8.8582e-04	85.71
BKW07	Horizontal	Área	Não	8.7250	8.2049e-04	62.86
BKW07	Horizontal	Perímetro	Sim	83.7750	9.6598e-04	88.57
BKW07	Horizontal	Perímetro	Não	11.5250	8.5373e-04	67.14
BKW07	Horizontal	Altura	Sim	86.1000	8.2316e-04	95.71
BKW07	Horizontal	Altura	Não	6.7500	7.8702e-04	58.57
BKW07	Horizontal	Largura	Sim	46.6750	7.3056e-04	78.57
BKW07	Horizontal	Largura	Não	26.7375	9.5806e-04	80.00
BKW07	Horizontal	Id	Sim	46.8250	7.1940e-04	78.57
BKW07	Horizontal	Id	Não	59.8000	1.1274e-03	87.14
BKW07	Maior área	Área	Sim	82.7250	1.5545e-03	90.00
BKW07	Maior área	Área	Não	40.1375	1.4470e-03	88.57
BKW07	Maior área	Perímetro	Sim	90.5875	1.4915e-03	91.43
BKW07	Maior área	Perímetro	Não	42.0875	1.4538e-03	90.00
BKW07	Maior área	Altura	Sim	82.3500	1.5525e-03	95.71
BKW07	Maior área	Altura	Não	38.9000	1.4123e-03	81.43
BKW07	Maior área	Largura	Sim	49.0750	1.3748e-03	82.86
BKW07	Maior área	Largura	Não	61.2125	1.4251e-03	94.29
BKW07	Maior área	Id	Sim	79.4375	1.6688e-03	94.29
BKW07	Maior área	Id	Não	64.3375	1.4823e-03	91.43
BKW07	Sobrepostas	Área	Sim	90.2375	3.1498e-01	95.71
BKW07	Sobrepostas	Área	Não	50.2875	6.5706e-01	92.86
BKW07	Sobrepostas	Perímetro	Sim	91.5875	2.3442e-01	92.86
BKW07	Sobrepostas	Perímetro	Não	60.7125	6.3953e-01	92.86
BKW07	Sobrepostas	Altura	Sim	82.3500	2.5686e-01	95.71
BKW07	Sobrepostas	Altura	Não	56.6875	5.4043e-01	88.57
BKW07	Sobrepostas	Largura	Sim	94.5000	1.9643e-01	95.71
BKW07	Sobrepostas	Largura	Não	64.3000	7.2890e-01	94.29
BKW07	Sobrepostas	Id	Sim	79.4375	3.3753e-01	94.29
BKW07	Sobrepostas	Id	Não	75.8125	3.3454e-01	97.14

Fonte: feito pelo autor.

Tabela 18 – Resultados da instância BKW08.

Instância	Divisão	Ordenação	Decrescente	Qualidade %	Tempo (s)	Itens %
BKW08	Vertical	Área	Sim	81.7000	9.8100e-04	82.50
BKW08	Vertical	Área	Não	47.8500	7.1449e-04	85.00
BKW08	Vertical	Perímetro	Sim	86.9375	1.0267e-03	80.00
BKW08	Vertical	Perímetro	Não	58.3625	8.5321e-04	90.00
BKW08	Vertical	Altura	Sim	70.2375	9.7189e-04	80.00
BKW08	Vertical	Altura	Não	57.6750	9.3403e-04	92.50
BKW08	Vertical	Largura	Sim	92.7000	8.8458e-04	86.25
BKW08	Vertical	Largura	Não	51.1875	7.1197e-04	81.25
BKW08	Vertical	Id	Sim	54.4875	1.4294e-03	85.00
BKW08	Vertical	Id	Não	70.1875	8.9645e-04	83.75
BKW08	Horizontal	Área	Sim	72.1875	1.1306e-03	78.75
BKW08	Horizontal	Área	Não	16.3875	7.1249e-04	51.25
BKW08	Horizontal	Perímetro	Sim	72.9500	1.2429e-03	78.75
BKW08	Horizontal	Perímetro	Não	16.3875	7.1640e-04	51.25
BKW08	Horizontal	Altura	Sim	83.3125	1.8450e-03	90.00
BKW08	Horizontal	Altura	Não	10.9500	6.4631e-04	35.00
BKW08	Horizontal	Largura	Sim	53.8875	8.4953e-04	71.25
BKW08	Horizontal	Largura	Não	24.7625	9.6712e-04	68.75
BKW08	Horizontal	Id	Sim	32.3500	1.1181e-03	73.75
BKW08	Horizontal	Id	Não	36.7625	9.4471e-04	70.00
BKW08	Maior área	Área	Sim	93.7875	2.0847e-03	93.75
BKW08	Maior área	Área	Não	50.5500	1.5254e-03	90.00
BKW08	Maior área	Perímetro	Sim	96.3000	1.5986e-03	82.50
BKW08	Maior área	Perímetro	Não	57.6750	1.7844e-03	92.50
BKW08	Maior área	Altura	Sim	91.7000	2.2449e-03	88.75
BKW08	Maior área	Altura	Não	39.8750	2.0337e-03	81.25
BKW08	Maior área	Largura	Sim	80.5000	1.9263e-03	91.25
BKW08	Maior área	Largura	Não	58.3500	1.6934e-03	92.50
BKW08	Maior área	Id	Sim	63.0750	1.7603e-03	91.25
BKW08	Maior área	Id	Não	82.7125	1.8297e-03	95.00
BKW08	Sobrepostas	Área	Sim	94.2000	5.4258e-01	68.75
BKW08	Sobrepostas	Área	Não	66.7250	1.1957e+00	95.00
BKW08	Sobrepostas	Perímetro	Sim	95.5375	5.9883e-01	77.50
BKW08	Sobrepostas	Perímetro	Não	66.7250	1.1602e+00	95.00
BKW08	Sobrepostas	Altura	Sim	91.0375	7.2394e-01	88.75
BKW08	Sobrepostas	Altura	Não	53.0000	1.0804e+00	85.00
BKW08	Sobrepostas	Largura	Sim	94.1750	6.7467e-01	97.50
BKW08	Sobrepostas	Largura	Não	70.4375	1.3156e+00	93.75
BKW08	Sobrepostas	Id	Sim	82.4500	9.1109e-01	96.25
BKW08	Sobrepostas	Id	Não	86.9500	6.1646e-01	95.00

Fonte: feito pelo autor.

Tabela 19 – Resultados da instância BKW09.

Instância	Divisão	Ordenação	Decrescente	Qualidade %	Tempo (s)	Itens %
BKW09	Vertical	Área	Sim	70.5467	9.3708e-04	80
BKW09	Vertical	Área	Não	35.6800	8.6551e-04	85
BKW09	Vertical	Perímetro	Sim	73.5867	1.0125e-03	80
BKW09	Vertical	Perímetro	Não	35.6800	7.4091e-04	85
BKW09	Vertical	Altura	Sim	50.7067	9.2530e-04	82
BKW09	Vertical	Altura	Não	34.3467	7.9546e-04	83
BKW09	Vertical	Largura	Sim	90.0400	1.0225e-03	85
BKW09	Vertical	Largura	Não	31.8267	6.6066e-04	78
BKW09	Vertical	Id	Sim	42.7600	1.0644e-03	88
BKW09	Vertical	Id	Não	35.3467	9.3207e-04	82
BKW09	Horizontal	Área	Sim	91.7867	1.6902e-03	95
BKW09	Horizontal	Área	Não	22.1200	1.3567e-03	72
BKW09	Horizontal	Perímetro	Sim	94.0933	1.1711e-03	65
BKW09	Horizontal	Perímetro	Não	24.6267	1.4016e-03	73
BKW09	Horizontal	Altura	Sim	83.4533	1.5490e-03	94
BKW09	Horizontal	Altura	Não	14.6933	1.1772e-03	49
BKW09	Horizontal	Largura	Sim	70.4800	1.5297e-03	89
BKW09	Horizontal	Largura	Não	34.1200	1.5114e-03	86
BKW09	Horizontal	Id	Sim	57.1200	1.7036e-03	90
BKW09	Horizontal	Id	Não	54.9200	1.4885e-03	83
BKW09	Maior área	Área	Sim	87.3467	2.6760e-03	95
BKW09	Maior área	Área	Não	37.5200	2.1771e-03	87
BKW09	Maior área	Perímetro	Sim	93.1067	2.7644e-03	98
BKW09	Maior área	Perímetro	Não	32.9867	2.1779e-03	85
BKW09	Maior área	Altura	Sim	86.1733	2.4829e-03	94
BKW09	Maior área	Altura	Não	27.4533	2.3566e-03	77
BKW09	Maior área	Largura	Sim	76.4533	2.7125e-03	92
BKW09	Maior área	Largura	Não	63.9867	2.2154e-03	94
BKW09	Maior área	Id	Sim	63.9867	2.6590e-03	94
BKW09	Maior área	Id	Não	78.2000	2.6271e-03	94
BKW09	Sobrepostas	Área	Sim	94.2933	1.1927e+00	97
BKW09	Sobrepostas	Área	Não	62.5733	2.0321e+00	96
BKW09	Sobrepostas	Perímetro	Sim	96.9200	1.2871e+00	79
BKW09	Sobrepostas	Perímetro	Não	58.2533	1.9654e+00	95
BKW09	Sobrepostas	Altura	Sim	85.2133	1.3337e+00	93
BKW09	Sobrepostas	Altura	Não	54.2800	2.0387e+00	94
BKW09	Sobrepostas	Largura	Sim	96.2800	1.8623e+00	98
BKW09	Sobrepostas	Largura	Não	68.4667	2.4013e+00	96
BKW09	Sobrepostas	Id	Sim	72.5733	1.5391e+00	97
BKW09	Sobrepostas	Id	Não	82.0000	1.4539e+00	96

Fonte: feito pelo autor.

Tabela 20 – Resultados da instância BKW10.

Instância	Divisão	Ordenação	Decrescente	Qualidade %	Tempo (s)	Itens %
BKW10	Vertical	Área	Sim	61.8381	2.5306e-03	88.50
BKW10	Vertical	Área	Não	22.5714	1.3769e-03	84.00
BKW10	Vertical	Perímetro	Sim	62.5714	2.4922e-03	90.00
BKW10	Vertical	Perímetro	Não	21.9810	1.4228e-03	83.00
BKW10	Vertical	Altura	Sim	50.9905	2.6208e-03	85.00
BKW10	Vertical	Altura	Não	24.3048	1.6850e-03	86.00
BKW10	Vertical	Largura	Sim	84.0667	2.4504e-03	93.50
BKW10	Vertical	Largura	Não	21.9810	1.2523e-03	83.00
BKW10	Vertical	Id	Sim	37.2762	2.0342e-03	87.50
BKW10	Vertical	Id	Não	41.6286	1.6489e-03	85.50
BKW10	Horizontal	Área	Sim	95.3714	4.0924e-03	99.00
BKW10	Horizontal	Área	Não	16.3714	4.3102e-03	76.00
BKW10	Horizontal	Perímetro	Sim	95.5048	3.5373e-03	95.50
BKW10	Horizontal	Perímetro	Não	16.3714	4.2977e-03	76.00
BKW10	Horizontal	Altura	Sim	77.0952	3.4287e-03	93.50
BKW10	Horizontal	Altura	Não	9.8857	3.0872e-03	51.50
BKW10	Horizontal	Largura	Sim	87.8571	2.6926e-03	95.00
BKW10	Horizontal	Largura	Não	28.9810	3.5844e-03	89.50
BKW10	Horizontal	Id	Sim	40.4095	3.0013e-03	93.50
BKW10	Horizontal	Id	Não	75.4952	3.1981e-03	95.00
BKW10	Maior área	Área	Sim	91.3429	6.1746e-03	97.50
BKW10	Maior área	Área	Não	31.3143	5.5194e-03	94.00
BKW10	Maior área	Perímetro	Sim	95.9429	5.9940e-03	99.00
BKW10	Maior área	Perímetro	Não	31.5619	5.7390e-03	94.00
BKW10	Maior área	Altura	Sim	72.2952	6.3191e-03	97.50
BKW10	Maior área	Altura	Não	30.5810	5.3077e-03	90.50
BKW10	Maior área	Largura	Sim	94.0381	5.1808e-03	98.50
BKW10	Maior área	Largura	Não	54.6000	4.5886e-03	97.00
BKW10	Maior área	Id	Sim	58.6286	4.8657e-03	98.50
BKW10	Maior área	Id	Não	83.0667	6.0828e-03	96.00
BKW10	Sobrepostas	Área	Sim	96.1238	9.7142e+00	98.00
BKW10	Sobrepostas	Área	Não	45.6381	1.7891e+01	97.50
BKW10	Sobrepostas	Perímetro	Sim	97.5619	7.9507e+00	69.50
BKW10	Sobrepostas	Perímetro	Não	50.3619	1.8647e+01	98.00
BKW10	Sobrepostas	Altura	Sim	77.8952	9.2436e+00	94.50
BKW10	Sobrepostas	Altura	Não	48.5905	1.2567e+01	97.50
BKW10	Sobrepostas	Largura	Sim	98.1143	1.2305e+01	82.50
BKW10	Sobrepostas	Largura	Não	55.8000	2.1395e+01	97.50
BKW10	Sobrepostas	Id	Sim	58.6286	1.3919e+01	98.50
BKW10	Sobrepostas	Id	Não	76.2476	1.3409e+01	98.50

Fonte: feito pelo autor.

Tabela 21 – Resultados da instância BKW11.

Instância	Divisão	Ordenação	Decrescente	Qualidade %	Tempo (s)	Itens %
BKW11	Vertical	Área	Sim	65.7238	3.9003e-03	86.33
BKW11	Vertical	Área	Não	37.9238	2.4142e-03	85.67
BKW11	Vertical	Perímetro	Sim	73.9429	3.8857e-03	87.00
BKW11	Vertical	Perímetro	Não	37.9238	2.6056e-03	85.67
BKW11	Vertical	Altura	Sim	56.7238	3.7015e-03	86.67
BKW11	Vertical	Altura	Não	35.7524	2.9558e-03	86.00
BKW11	Vertical	Largura	Sim	88.9714	3.4059e-03	64.67
BKW11	Vertical	Largura	Não	37.9238	2.0863e-03	85.67
BKW11	Vertical	Id	Sim	46.3905	2.9467e-03	88.33
BKW11	Vertical	Id	Não	39.2000	3.1803e-03	84.33
BKW11	Horizontal	Área	Sim	93.7905	6.6441e-03	85.67
BKW11	Horizontal	Área	Não	21.7810	5.3345e-03	68.33
BKW11	Horizontal	Perímetro	Sim	94.8857	4.8075e-03	85.33
BKW11	Horizontal	Perímetro	Não	21.7810	5.5768e-03	68.33
BKW11	Horizontal	Altura	Sim	82.1810	6.3877e-03	95.00
BKW11	Horizontal	Altura	Não	10.7810	4.4535e-03	35.33
BKW11	Horizontal	Largura	Sim	80.4286	5.3077e-03	84.33
BKW11	Horizontal	Largura	Não	27.2667	6.8587e-03	78.67
BKW11	Horizontal	Id	Sim	52.4571	6.2678e-03	89.33
BKW11	Horizontal	Id	Não	54.4952	5.9869e-03	89.00
BKW11	Maior área	Área	Sim	96.7143	9.3718e-03	94.33
BKW11	Maior área	Área	Não	41.2667	9.0139e-03	90.33
BKW11	Maior área	Perímetro	Sim	97.7714	8.8081e-03	83.33
BKW11	Maior área	Perímetro	Não	41.8286	9.3330e-03	90.00
BKW11	Maior área	Altura	Sim	82.2571	1.1729e-02	95.33
BKW11	Maior área	Altura	Não	27.8762	9.7571e-03	75.33
BKW11	Maior área	Largura	Sim	89.9905	9.6921e-03	95.33
BKW11	Maior área	Largura	Não	61.1238	6.5022e-03	94.67
BKW11	Maior área	Id	Sim	80.5143	1.0383e-02	97.67
BKW11	Maior área	Id	Não	80.2762	9.0831e-03	97.33
BKW11	Sobrepostas	Área	Sim	97.2095	4.7685e+01	92.00
BKW11	Sobrepostas	Área	Não	69.4381	6.7367e+01	97.00
BKW11	Sobrepostas	Perímetro	Sim	98.1524	4.7298e+01	86.67
BKW11	Sobrepostas	Perímetro	Não	69.9905	6.5980e+01	96.33
BKW11	Sobrepostas	Altura	Sim	83.3810	3.7135e+01	96.33
BKW11	Sobrepostas	Altura	Não	58.0381	5.2771e+01	93.33
BKW11	Sobrepostas	Largura	Sim	98.0952	6.0246e+01	90.00
BKW11	Sobrepostas	Largura	Não	72.7238	7.0334e+01	96.33
BKW11	Sobrepostas	Id	Sim	78.7143	5.5696e+01	97.33
BKW11	Sobrepostas	Id	Não	85.2095	4.1413e+01	98.00

Fonte: feito pelo autor.

Tabela 22 – Resultados da instância BKW12.

Instância	Divisão	Ordenação	Decrescente	Qualidade %	Tempo (s)	Itens %
BKW12	Vertical	Área	Sim	80.6733	8.5429e-03	92.00
BKW12	Vertical	Área	Não	51.1267	4.5707e-03	92.40
BKW12	Vertical	Perímetro	Sim	81.0333	8.8156e-03	92.20
BKW12	Vertical	Perímetro	Não	56.2433	4.5100e-03	93.20
BKW12	Vertical	Altura	Sim	74.5833	9.6094e-03	90.00
BKW12	Vertical	Altura	Não	50.2133	5.6847e-03	93.00
BKW12	Vertical	Largura	Sim	87.8000	5.8658e-03	62.60
BKW12	Vertical	Largura	Não	55.9967	3.8399e-03	91.40
BKW12	Vertical	Id	Sim	59.3267	8.0027e-03	91.60
BKW12	Vertical	Id	Não	49.8300	6.6992e-03	91.60
BKW12	Horizontal	Área	Sim	89.3767	2.0509e-02	89.00
BKW12	Horizontal	Área	Não	8.5500	9.2856e-03	36.60
BKW12	Horizontal	Perímetro	Sim	91.2000	1.4939e-02	78.00
BKW12	Horizontal	Perímetro	Não	8.5500	9.3717e-03	36.60
BKW12	Horizontal	Altura	Sim	86.9067	2.0183e-02	92.40
BKW12	Horizontal	Altura	Não	8.5500	9.3761e-03	36.60
BKW12	Horizontal	Largura	Sim	74.5833	1.6376e-02	92.60
BKW12	Horizontal	Largura	Não	26.6167	1.4265e-02	81.80
BKW12	Horizontal	Id	Sim	45.3133	2.5891e-02	89.80
BKW12	Horizontal	Id	Não	34.7533	1.5724e-02	82.60
BKW12	Maior área	Área	Sim	95.1100	2.0580e-02	78.00
BKW12	Maior área	Área	Não	28.7067	2.9351e-02	86.00
BKW12	Maior área	Perímetro	Sim	94.7000	1.8316e-02	68.40
BKW12	Maior área	Perímetro	Não	28.2100	2.1613e-02	85.40
BKW12	Maior área	Altura	Sim	88.0967	3.6373e-02	97.20
BKW12	Maior área	Altura	Não	24.4900	2.1656e-02	78.80
BKW12	Maior área	Largura	Sim	89.6933	2.7099e-02	99.20
BKW12	Maior área	Largura	Não	67.2400	1.2437e-02	97.60
BKW12	Maior área	Id	Sim	84.9233	2.8526e-02	98.60
BKW12	Maior área	Id	Não	75.5567	2.8829e-02	97.80
BKW12	Sobrepostas	Área	Sim	95.9100	2.2323e+02	76.60
BKW12	Sobrepostas	Área	Não	66.8933	3.6634e+02	97.80
BKW12	Sobrepostas	Perímetro	Sim	95.8400	2.2349e+02	74.40
BKW12	Sobrepostas	Perímetro	Não	68.8833	3.5608e+02	98.00
BKW12	Sobrepostas	Altura	Sim	90.0500	2.3005e+02	97.00
BKW12	Sobrepostas	Altura	Não	65.8767	3.0229e+02	97.20
BKW12	Sobrepostas	Largura	Sim	96.2800	2.9520e+02	92.20
BKW12	Sobrepostas	Largura	Não	73.9967	3.6886e+02	98.00
BKW12	Sobrepostas	Id	Sim	88.5733	2.9970e+02	98.60
BKW12	Sobrepostas	Id	Não	78.0967	3.1879e+02	98.60

Fonte: feito pelo autor.

A instância BKW13, mostrada na Tabela 23, não foi executada com regiões complexas, pois seu tempo de execução ultrapassaria uma hora e conseguiria resultados semelhantes aos demais modos (como mostrado na Seção 3.3).

Tabela 23 – Resultados da instância BKW13.

Instância	Divisão	Ordenação	Decrescente	Qualidade %	Tempo (s)	Itens %
BKW13	Vertical	Área	Sim	94.9339	1.1675e-01	84.01
BKW13	Vertical	Área	Não	72.6849	9.5678e-02	96.45
BKW13	Vertical	Perímetro	Sim	91.0596	8.2055e-02	74.24
BKW13	Vertical	Perímetro	Não	73.5465	8.0993e-02	96.13
BKW13	Vertical	Altura	Sim	86.7168	9.0644e-02	86.96
BKW13	Vertical	Altura	Não	78.2632	7.3115e-02	96.99
BKW13	Vertical	Largura	Sim	97.9108	9.2664e-02	90.74
BKW13	Vertical	Largura	Não	73.9661	6.0921e-02	88.83
BKW13	Vertical	Id	Sim	61.0007	1.2061e-01	83.95
BKW13	Vertical	Id	Não	60.3755	8.9735e-02	82.20
BKW13	Horizontal	Área	Sim	51.3125	2.3023e-01	57.87
BKW13	Horizontal	Área	Não	4.2135	3.7217e-01	32.23
BKW13	Horizontal	Perímetro	Sim	63.4264	1.8303e-01	63.26
BKW13	Horizontal	Perímetro	Não	4.2656	3.5591e-01	29.95
BKW13	Horizontal	Altura	Sim	85.3234	2.3699e-01	86.04
BKW13	Horizontal	Altura	Não	1.2526	3.2270e-01	18.78
BKW13	Horizontal	Largura	Sim	32.3698	3.1027e-01	54.31
BKW13	Horizontal	Largura	Não	11.5620	4.1873e-01	55.74
BKW13	Horizontal	Id	Sim	38.9907	2.5004e-01	77.16
BKW13	Horizontal	Id	Não	38.8179	2.4046e-01	76.74
BKW13	Maior área	Área	Sim	92.3503	5.4441e-01	94.13
BKW13	Maior área	Área	Não	32.1917	7.1785e-01	81.85
BKW13	Maior área	Perímetro	Sim	91.9723	5.3467e-01	93.81
BKW13	Maior área	Perímetro	Não	33.3984	7.4306e-01	82.04
BKW13	Maior área	Altura	Sim	97.1108	5.2685e-01	95.81
BKW13	Maior área	Altura	Não	9.5863	4.3582e-01	44.42
BKW13	Maior área	Largura	Sim	55.9587	6.1040e-01	85.03
BKW13	Maior área	Largura	Não	82.3021	1.4048e-01	95.34
BKW13	Maior área	Id	Sim	94.4312	5.8916e-01	98.76
BKW13	Maior área	Id	Não	94.6325	5.9675e-01	98.79

Fonte: feito pelo autor.

A.2 GCUT

Tabela 24 – Resultados da instância GCUT01.

Instância	Divisão	Ordenação	Decrescente	Qualidade %	Tempo (s)	Itens %
GCUT01	Vertical	Área	Sim	77.3888	3.7432e-05	30
GCUT01	Vertical	Área	Não	62.0688	4.8780e-05	40
GCUT01	Vertical	Perímetro	Sim	67.3808	2.6512e-05	20
GCUT01	Vertical	Perímetro	Não	63.4800	4.9114e-05	40
GCUT01	Vertical	Altura	Sim	67.3808	2.6894e-05	20
GCUT01	Vertical	Altura	Não	46.7840	3.5667e-05	30
GCUT01	Vertical	Largura	Sim	77.3888	3.3188e-05	30
GCUT01	Vertical	Largura	Não	43.4768	3.6573e-05	30
GCUT01	Vertical	Id	Sim	63.1040	4.7302e-05	40
GCUT01	Vertical	Id	Não	77.3888	3.3569e-05	30
GCUT01	Horizontal	Área	Sim	67.7568	3.3093e-05	20
GCUT01	Horizontal	Área	Não	43.8528	3.8767e-05	30
GCUT01	Horizontal	Perímetro	Sim	67.3808	2.7037e-05	20
GCUT01	Horizontal	Perímetro	Não	43.8528	3.9434e-05	30
GCUT01	Horizontal	Altura	Sim	67.3808	2.8086e-05	20
GCUT01	Horizontal	Altura	Não	31.1552	2.9707e-05	20
GCUT01	Horizontal	Largura	Sim	67.7568	2.5702e-05	20
GCUT01	Horizontal	Largura	Não	43.8528	4.0340e-05	30
GCUT01	Horizontal	Id	Sim	47.4752	4.1199e-05	30
GCUT01	Horizontal	Id	Não	67.7568	2.4080e-05	20
GCUT01	Maior área	Área	Sim	77.3888	7.0143e-05	30
GCUT01	Maior área	Área	Não	62.0688	9.4986e-05	40
GCUT01	Maior área	Perímetro	Sim	67.3808	5.0068e-05	20
GCUT01	Maior área	Perímetro	Não	63.4800	9.4366e-05	40
GCUT01	Maior área	Altura	Sim	67.3808	5.1784e-05	20
GCUT01	Maior área	Altura	Não	46.7840	7.0381e-05	30
GCUT01	Maior área	Largura	Sim	77.3888	6.7854e-05	30
GCUT01	Maior área	Largura	Não	62.0688	9.5272e-05	40
GCUT01	Maior área	Id	Sim	63.1040	9.4795e-05	40
GCUT01	Maior área	Id	Não	77.3888	6.7329e-05	30
GCUT01	Sobrepostas	Área	Sim	77.3888	1.2126e-04	30
GCUT01	Sobrepostas	Área	Não	62.0688	4.6997e-04	40
GCUT01	Sobrepostas	Perímetro	Sim	67.3808	5.6219e-05	20
GCUT01	Sobrepostas	Perímetro	Não	63.4800	5.8246e-04	40
GCUT01	Sobrepostas	Altura	Sim	67.3808	5.4932e-05	20
GCUT01	Sobrepostas	Altura	Não	49.7472	3.7170e-04	30
GCUT01	Sobrepostas	Largura	Sim	77.3888	1.1854e-04	30
GCUT01	Sobrepostas	Largura	Não	62.0688	2.2902e-04	40
GCUT01	Sobrepostas	Id	Sim	63.1040	2.1100e-04	40
GCUT01	Sobrepostas	Id	Não	77.3888	1.1430e-04	30

Fonte: feito pelo autor.

Tabela 25 – Resultados da instância GCUT02.

Instância	Divisão	Ordenação	Decrescente	Qualidade %	Tempo (s)	Itens %
GCUT02	Vertical	Área	Sim	75.9760	4.0197e-05	15
GCUT02	Vertical	Área	Não	29.5232	4.2391e-05	15
GCUT02	Vertical	Perímetro	Sim	75.9760	4.0436e-05	15
GCUT02	Vertical	Perímetro	Não	50.1248	5.5313e-05	20
GCUT02	Vertical	Altura	Sim	60.1696	3.2568e-05	10
GCUT02	Vertical	Altura	Não	46.0144	5.3597e-05	20
GCUT02	Vertical	Largura	Sim	67.6032	4.3535e-05	15
GCUT02	Vertical	Largura	Não	39.1968	4.2582e-05	15
GCUT02	Vertical	Id	Sim	69.6704	5.1212e-05	20
GCUT02	Vertical	Id	Não	56.8624	5.5790e-05	20
GCUT02	Horizontal	Área	Sim	65.6560	3.2902e-05	10
GCUT02	Horizontal	Área	Não	56.2416	7.1335e-05	25
GCUT02	Horizontal	Perímetro	Sim	65.6560	3.3331e-05	10
GCUT02	Horizontal	Perímetro	Não	63.6400	7.4577e-05	25
GCUT02	Horizontal	Altura	Sim	60.1696	3.3522e-05	10
GCUT02	Horizontal	Altura	Não	44.3440	4.5872e-05	15
GCUT02	Horizontal	Largura	Sim	68.1568	5.5313e-05	20
GCUT02	Horizontal	Largura	Não	56.2416	7.2479e-05	25
GCUT02	Horizontal	Id	Sim	76.7008	7.0715e-05	25
GCUT02	Horizontal	Id	Não	56.8624	5.7697e-05	20
GCUT02	Maior área	Área	Sim	77.6864	7.8821e-05	15
GCUT02	Maior área	Área	Não	41.5536	1.0476e-04	20
GCUT02	Maior área	Perímetro	Sim	77.6864	7.5579e-05	15
GCUT02	Maior área	Perímetro	Não	41.5536	1.0128e-04	20
GCUT02	Maior área	Altura	Sim	60.1696	5.7554e-05	10
GCUT02	Maior área	Altura	Não	54.6640	1.0295e-04	20
GCUT02	Maior área	Largura	Sim	68.1568	9.9945e-05	20
GCUT02	Maior área	Largura	Não	39.1968	7.8201e-05	15
GCUT02	Maior área	Id	Sim	69.9840	1.0247e-04	20
GCUT02	Maior área	Id	Não	73.8544	1.2260e-04	25
GCUT02	Sobrepostas	Área	Sim	77.6864	1.2984e-04	15
GCUT02	Sobrepostas	Área	Não	56.2416	1.2731e-03	25
GCUT02	Sobrepostas	Perímetro	Sim	77.6864	1.2975e-04	15
GCUT02	Sobrepostas	Perímetro	Não	76.2352	1.3390e-03	30
GCUT02	Sobrepostas	Altura	Sim	72.2000	1.2803e-04	15
GCUT02	Sobrepostas	Altura	Não	54.6640	7.7462e-04	20
GCUT02	Sobrepostas	Largura	Sim	68.1568	3.5162e-04	20
GCUT02	Sobrepostas	Largura	Não	50.1248	8.4643e-04	20
GCUT02	Sobrepostas	Id	Sim	69.9840	6.5637e-04	20
GCUT02	Sobrepostas	Id	Não	56.8624	4.5171e-04	20

Fonte: feito pelo autor.

Tabela 26 – Resultados da instância GCUT03.

Instância	Divisão	Ordenação	Decrescente	Qualidade %	Tempo (s)	Itens %
GCUT03	Vertical	Área	Sim	87.4512	5.8222e-05	13.33
GCUT03	Vertical	Área	Não	54.9824	7.9632e-05	16.67
GCUT03	Vertical	Perímetro	Sim	86.0608	5.2261e-05	10.00
GCUT03	Vertical	Perímetro	Não	68.3104	9.4748e-05	20.00
GCUT03	Vertical	Altura	Sim	76.6000	6.6233e-05	13.33
GCUT03	Vertical	Altura	Não	72.8752	7.8440e-05	16.67
GCUT03	Vertical	Largura	Sim	87.0368	6.1274e-05	13.33
GCUT03	Vertical	Largura	Não	40.2928	5.0402e-05	10.00
GCUT03	Vertical	Id	Sim	68.7360	7.9060e-05	16.67
GCUT03	Vertical	Id	Não	66.1776	6.3515e-05	13.33
GCUT03	Horizontal	Área	Sim	87.0160	5.8746e-05	13.33
GCUT03	Horizontal	Área	Não	61.6464	7.7105e-05	16.67
GCUT03	Horizontal	Perímetro	Sim	93.5504	6.1894e-05	13.33
GCUT03	Horizontal	Perímetro	Não	61.6464	8.0872e-05	16.67
GCUT03	Horizontal	Altura	Sim	68.1520	5.5885e-05	10.00
GCUT03	Horizontal	Altura	Não	36.6032	5.7793e-05	10.00
GCUT03	Horizontal	Largura	Sim	75.7456	6.2752e-05	13.33
GCUT03	Horizontal	Largura	Não	49.6064	6.9666e-05	13.33
GCUT03	Horizontal	Id	Sim	56.0832	6.7473e-05	13.33
GCUT03	Horizontal	Id	Não	57.9184	8.6498e-05	13.33
GCUT03	Maior área	Área	Sim	87.3984	1.0753e-04	13.33
GCUT03	Maior área	Área	Não	41.5568	1.1282e-04	13.33
GCUT03	Maior área	Perímetro	Sim	93.5504	1.0753e-04	13.33
GCUT03	Maior área	Perímetro	Não	41.5568	1.1158e-04	13.33
GCUT03	Maior área	Altura	Sim	76.6000	1.0991e-04	13.33
GCUT03	Maior área	Altura	Não	56.2256	1.0886e-04	13.33
GCUT03	Maior área	Largura	Sim	75.7456	1.0104e-04	13.33
GCUT03	Maior área	Largura	Não	58.9504	1.1392e-04	13.33
GCUT03	Maior área	Id	Sim	56.0832	1.1158e-04	13.33
GCUT03	Maior área	Id	Não	66.1776	1.0591e-04	13.33
GCUT03	Sobrepostas	Área	Sim	87.3984	4.2763e-04	13.33
GCUT03	Sobrepostas	Área	Não	67.7792	3.5841e-03	20.00
GCUT03	Sobrepostas	Perímetro	Sim	86.0608	3.6655e-04	10.00
GCUT03	Sobrepostas	Perímetro	Não	68.1040	3.1793e-03	20.00
GCUT03	Sobrepostas	Altura	Sim	76.6000	2.2597e-04	13.33
GCUT03	Sobrepostas	Altura	Não	56.2256	1.3146e-03	13.33
GCUT03	Sobrepostas	Largura	Sim	87.0368	3.5462e-04	13.33
GCUT03	Sobrepostas	Largura	Não	58.9504	6.3920e-04	13.33
GCUT03	Sobrepostas	Id	Sim	77.1840	1.0312e-03	20.00
GCUT03	Sobrepostas	Id	Não	57.9184	1.2184e-03	13.33

Fonte: feito pelo autor.

Tabela 27 – Resultados da instância GCUT04.

Instância	Divisão	Ordenação	Decrescente	Qualidade %	Tempo (s)	Itens %
GCUT04	Vertical	Área	Sim	92.8160	6.8474e-05	8
GCUT04	Vertical	Área	Não	58.8480	1.0767e-04	12
GCUT04	Vertical	Perímetro	Sim	92.8160	7.4196e-05	8
GCUT04	Vertical	Perímetro	Não	58.8480	1.0757e-04	12
GCUT04	Vertical	Altura	Sim	90.8624	7.6675e-05	10
GCUT04	Vertical	Altura	Não	61.5680	1.0748e-04	12
GCUT04	Vertical	Largura	Sim	86.7136	6.8951e-05	8
GCUT04	Vertical	Largura	Não	44.5504	6.6090e-05	8
GCUT04	Vertical	Id	Sim	72.9616	7.7677e-05	8
GCUT04	Vertical	Id	Não	63.7264	8.5640e-05	10
GCUT04	Horizontal	Área	Sim	83.6768	6.5374e-05	6
GCUT04	Horizontal	Área	Não	71.9872	1.4682e-04	14
GCUT04	Horizontal	Perímetro	Sim	83.6768	6.1178e-05	6
GCUT04	Horizontal	Perímetro	Não	54.7504	1.3065e-04	12
GCUT04	Horizontal	Altura	Sim	82.7280	7.1049e-05	8
GCUT04	Horizontal	Altura	Não	34.8656	7.4434e-05	6
GCUT04	Horizontal	Largura	Sim	96.1424	8.4353e-05	10
GCUT04	Horizontal	Largura	Não	64.0320	1.2999e-04	12
GCUT04	Horizontal	Id	Sim	79.3360	8.5401e-05	8
GCUT04	Horizontal	Id	Não	67.7552	8.9026e-05	8
GCUT04	Maior área	Área	Sim	94.3856	1.1673e-04	8
GCUT04	Maior área	Área	Não	64.5616	1.9250e-04	14
GCUT04	Maior área	Perímetro	Sim	94.3856	1.1697e-04	8
GCUT04	Maior área	Perímetro	Não	64.5616	1.9188e-04	14
GCUT04	Maior área	Altura	Sim	93.4368	1.3871e-04	10
GCUT04	Maior área	Altura	Não	53.4192	1.5845e-04	10
GCUT04	Maior área	Largura	Sim	96.1424	1.4410e-04	10
GCUT04	Maior área	Largura	Não	71.2128	1.6317e-04	12
GCUT04	Maior área	Id	Sim	72.9616	1.2188e-04	8
GCUT04	Maior área	Id	Não	67.7552	1.3151e-04	8
GCUT04	Sobrepostas	Área	Sim	94.3856	4.2419e-04	8
GCUT04	Sobrepostas	Área	Não	75.2704	6.3355e-03	16
GCUT04	Sobrepostas	Perímetro	Sim	94.3856	3.6907e-04	8
GCUT04	Sobrepostas	Perímetro	Não	75.2704	6.3654e-03	16
GCUT04	Sobrepostas	Altura	Sim	93.4368	3.7985e-04	10
GCUT04	Sobrepostas	Altura	Não	62.8480	3.6749e-03	12
GCUT04	Sobrepostas	Largura	Sim	96.1424	3.8924e-04	10
GCUT04	Sobrepostas	Largura	Não	71.2128	5.2459e-03	12
GCUT04	Sobrepostas	Id	Sim	79.3360	8.5273e-04	8
GCUT04	Sobrepostas	Id	Não	67.7552	8.7338e-04	8

Fonte: feito pelo autor.

Tabela 28 – Resultados da instância GCUT05.

Instância	Divisão	Ordenação	Decrescente	Qualidade %	Tempo (s)	Itens %
GCUT05	Vertical	Área	Sim	65.7440	3.8052e-05	30
GCUT05	Vertical	Área	Não	36.6852	3.6621e-05	30
GCUT05	Vertical	Perímetro	Sim	65.7440	3.6573e-05	30
GCUT05	Vertical	Perímetro	Não	36.6852	3.8004e-05	30
GCUT05	Vertical	Altura	Sim	65.7440	3.8004e-05	30
GCUT05	Vertical	Altura	Não	56.6948	4.7255e-05	40
GCUT05	Vertical	Largura	Sim	63.9648	3.9482e-05	30
GCUT05	Vertical	Largura	Não	36.6852	3.9387e-05	30
GCUT05	Vertical	Id	Sim	46.2440	3.8814e-05	30
GCUT05	Vertical	Id	Não	52.5148	4.8733e-05	40
GCUT05	Horizontal	Área	Sim	56.5568	2.7990e-05	20
GCUT05	Horizontal	Área	Não	56.6948	5.0449e-05	40
GCUT05	Horizontal	Perímetro	Sim	56.5568	2.7275e-05	20
GCUT05	Horizontal	Perímetro	Não	56.9356	4.9639e-05	40
GCUT05	Horizontal	Altura	Sim	56.5568	4.0340e-05	20
GCUT05	Horizontal	Altura	Não	57.0664	3.7241e-05	30
GCUT05	Horizontal	Largura	Sim	63.9648	3.6907e-05	30
GCUT05	Horizontal	Largura	Não	52.9212	5.1594e-05	40
GCUT05	Horizontal	Id	Sim	57.3420	5.0306e-05	40
GCUT05	Horizontal	Id	Não	52.5148	5.2166e-05	40
GCUT05	Maior área	Área	Sim	72.7928	7.5912e-05	30
GCUT05	Maior área	Área	Não	56.6948	9.5892e-05	40
GCUT05	Maior área	Perímetro	Sim	72.7928	7.1907e-05	30
GCUT05	Maior área	Perímetro	Não	52.9212	9.5797e-05	40
GCUT05	Maior área	Altura	Sim	72.7928	7.1144e-05	30
GCUT05	Maior área	Altura	Não	66.2536	9.4986e-05	40
GCUT05	Maior área	Largura	Sim	63.9648	7.1907e-05	30
GCUT05	Maior área	Largura	Não	52.9212	9.1744e-05	40
GCUT05	Maior área	Id	Sim	57.3420	9.0838e-05	40
GCUT05	Maior área	Id	Não	68.9148	1.1787e-04	50
GCUT05	Sobrepostas	Área	Sim	72.7928	1.2221e-04	30
GCUT05	Sobrepostas	Área	Não	56.6948	4.7331e-04	40
GCUT05	Sobrepostas	Perímetro	Sim	72.7928	1.2221e-04	30
GCUT05	Sobrepostas	Perímetro	Não	52.5148	5.6872e-04	40
GCUT05	Sobrepostas	Altura	Sim	72.7928	1.1520e-04	30
GCUT05	Sobrepostas	Altura	Não	66.2536	4.2410e-04	40
GCUT05	Sobrepostas	Largura	Sim	63.9648	1.1826e-04	30
GCUT05	Sobrepostas	Largura	Não	52.9212	3.9582e-04	40
GCUT05	Sobrepostas	Id	Sim	73.3356	3.0842e-04	50
GCUT05	Sobrepostas	Id	Não	52.5148	3.1509e-04	40

Fonte: feito pelo autor.

Tabela 29 – Resultados da instância GCUT06.

Instância	Divisão	Ordenação	Decrescente	Qualidade %	Tempo (s)	Itens %
GCUT06	Vertical	Área	Sim	82.0112	5.1212e-05	20
GCUT06	Vertical	Área	Não	57.9372	5.1832e-05	20
GCUT06	Vertical	Perímetro	Sim	81.9180	5.1451e-05	20
GCUT06	Vertical	Perímetro	Não	53.0764	5.3740e-05	20
GCUT06	Vertical	Altura	Sim	59.9172	3.2806e-05	10
GCUT06	Vertical	Altura	Não	55.6788	5.4884e-05	20
GCUT06	Vertical	Largura	Sim	67.9596	4.4203e-05	15
GCUT06	Vertical	Largura	Não	60.3732	4.9067e-05	20
GCUT06	Vertical	Id	Sim	64.4944	4.5395e-05	15
GCUT06	Vertical	Id	Não	75.8792	5.3740e-05	20
GCUT06	Horizontal	Área	Sim	89.7596	5.4932e-05	20
GCUT06	Horizontal	Área	Não	51.5460	6.0272e-05	20
GCUT06	Horizontal	Perímetro	Sim	79.2576	4.5538e-05	15
GCUT06	Horizontal	Perímetro	Não	51.5460	5.9795e-05	20
GCUT06	Horizontal	Altura	Sim	59.9172	3.3903e-05	10
GCUT06	Horizontal	Altura	Não	54.9244	4.6825e-05	15
GCUT06	Horizontal	Largura	Sim	67.9596	4.4203e-05	15
GCUT06	Horizontal	Largura	Não	57.9372	6.2609e-05	20
GCUT06	Horizontal	Id	Sim	58.5072	4.6778e-05	15
GCUT06	Horizontal	Id	Não	81.9180	5.8079e-05	20
GCUT06	Maior área	Área	Sim	89.7596	1.0109e-04	20
GCUT06	Maior área	Área	Não	51.5460	1.0257e-04	20
GCUT06	Maior área	Perímetro	Sim	89.2368	9.7799e-05	20
GCUT06	Maior área	Perímetro	Não	74.4260	1.3299e-04	25
GCUT06	Maior área	Altura	Sim	76.0284	7.7152e-05	15
GCUT06	Maior área	Altura	Não	64.9036	1.0247e-04	20
GCUT06	Maior área	Largura	Sim	67.9596	7.4196e-05	15
GCUT06	Maior área	Largura	Não	60.3732	1.0514e-04	20
GCUT06	Maior área	Id	Sim	58.5072	7.9203e-05	15
GCUT06	Maior área	Id	Não	81.9180	1.0242e-04	20
GCUT06	Sobrepostas	Área	Sim	89.7596	2.9903e-04	20
GCUT06	Sobrepostas	Área	Não	51.5460	1.3199e-03	20
GCUT06	Sobrepostas	Perímetro	Sim	89.2368	2.9921e-04	20
GCUT06	Sobrepostas	Perímetro	Não	51.5460	1.4173e-03	20
GCUT06	Sobrepostas	Altura	Sim	76.0284	1.3037e-04	15
GCUT06	Sobrepostas	Altura	Não	64.9036	6.6676e-04	20
GCUT06	Sobrepostas	Largura	Sim	67.9596	1.3151e-04	15
GCUT06	Sobrepostas	Largura	Não	60.3732	9.2411e-04	20
GCUT06	Sobrepostas	Id	Sim	64.4944	3.6778e-04	15
GCUT06	Sobrepostas	Id	Não	81.9180	3.2120e-04	20

Fonte: feito pelo autor.

Tabela 30 – Resultados da instância GCUT07.

Instância	Divisão	Ordenação	Decrescente	Qualidade %	Tempo (s)	Itens %
GCUT07	Vertical	Área	Sim	72.4060	4.5157e-05	10.00
GCUT07	Vertical	Área	Não	63.7420	6.9809e-05	16.67
GCUT07	Vertical	Perímetro	Sim	72.4060	4.3869e-05	10.00
GCUT07	Vertical	Perímetro	Não	63.7420	7.1239e-05	16.67
GCUT07	Vertical	Altura	Sim	72.4060	4.5347e-05	10.00
GCUT07	Vertical	Altura	Não	63.7420	6.7616e-05	16.67
GCUT07	Vertical	Largura	Sim	72.4060	4.4298e-05	10.00
GCUT07	Vertical	Largura	Não	53.3224	5.6791e-05	13.33
GCUT07	Vertical	Id	Sim	53.3224	6.4135e-05	13.33
GCUT07	Vertical	Id	Não	67.1940	5.9223e-05	13.33
GCUT07	Horizontal	Área	Sim	72.4060	4.6778e-05	10.00
GCUT07	Horizontal	Área	Não	29.6964	6.2418e-05	10.00
GCUT07	Horizontal	Perímetro	Sim	72.4060	4.7731e-05	10.00
GCUT07	Horizontal	Perímetro	Não	29.6964	5.9557e-05	10.00
GCUT07	Horizontal	Altura	Sim	72.4060	4.6301e-05	10.00
GCUT07	Horizontal	Altura	Não	19.4796	4.5252e-05	6.67
GCUT07	Horizontal	Largura	Sim	72.4060	4.7111e-05	10.00
GCUT07	Horizontal	Largura	Não	29.6964	6.1560e-05	10.00
GCUT07	Horizontal	Id	Sim	50.1948	7.2098e-05	13.33
GCUT07	Horizontal	Id	Não	68.8716	6.4755e-05	13.33
GCUT07	Maior área	Área	Sim	72.4060	8.0538e-05	10.00
GCUT07	Maior área	Área	Não	61.1696	1.2293e-04	16.67
GCUT07	Maior área	Perímetro	Sim	72.4060	8.4305e-05	10.00
GCUT07	Maior área	Perímetro	Não	61.1696	1.2589e-04	16.67
GCUT07	Maior área	Altura	Sim	72.4060	8.2684e-05	10.00
GCUT07	Maior área	Altura	Não	48.4644	1.0667e-04	13.33
GCUT07	Maior área	Largura	Sim	72.4060	8.0347e-05	10.00
GCUT07	Maior área	Largura	Não	53.3224	1.0333e-04	13.33
GCUT07	Maior área	Id	Sim	65.2168	1.3280e-04	16.67
GCUT07	Maior área	Id	Não	76.0008	1.0648e-04	13.33
GCUT07	Sobrepostas	Área	Sim	72.4060	1.4997e-04	10.00
GCUT07	Sobrepostas	Área	Não	61.1696	2.2301e-03	16.67
GCUT07	Sobrepostas	Perímetro	Sim	72.4060	1.5025e-04	10.00
GCUT07	Sobrepostas	Perímetro	Não	61.1696	2.1540e-03	16.67
GCUT07	Sobrepostas	Altura	Sim	72.4060	1.6727e-04	10.00
GCUT07	Sobrepostas	Altura	Não	38.2476	1.0423e-03	10.00
GCUT07	Sobrepostas	Largura	Sim	72.4060	1.4644e-04	10.00
GCUT07	Sobrepostas	Largura	Não	53.3224	2.3964e-03	13.33
GCUT07	Sobrepostas	Id	Sim	71.0972	8.1229e-04	16.67
GCUT07	Sobrepostas	Id	Não	76.0008	3.9163e-04	13.33

Fonte: feito pelo autor.

Tabela 31 – Resultados da instância GCUT08.

Instância	Divisão	Ordenação	Decrescente	Qualidade %	Tempo (s)	Itens %
GCUT08	Vertical	Área	Sim	79.6752	5.6314e-05	6
GCUT08	Vertical	Área	Não	63.7732	1.0734e-04	12
GCUT08	Vertical	Perímetro	Sim	79.6752	5.8031e-05	6
GCUT08	Vertical	Perímetro	Não	52.9528	9.3079e-05	10
GCUT08	Vertical	Altura	Sim	79.6752	6.2084e-05	6
GCUT08	Vertical	Altura	Não	62.9792	1.0819e-04	12
GCUT08	Vertical	Largura	Sim	81.3776	7.2765e-05	8
GCUT08	Vertical	Largura	Não	35.9152	6.1512e-05	6
GCUT08	Vertical	Id	Sim	72.6640	7.7343e-05	8
GCUT08	Vertical	Id	Não	74.4596	9.2077e-05	10
GCUT08	Horizontal	Área	Sim	72.1296	6.4468e-05	4
GCUT08	Horizontal	Área	Não	61.8724	1.2908e-04	12
GCUT08	Horizontal	Perímetro	Sim	72.1296	4.9353e-05	4
GCUT08	Horizontal	Perímetro	Não	61.8724	1.3809e-04	12
GCUT08	Horizontal	Altura	Sim	72.1296	6.4516e-05	4
GCUT08	Horizontal	Altura	Não	39.6812	7.4053e-05	6
GCUT08	Horizontal	Largura	Sim	63.4032	5.8413e-05	6
GCUT08	Horizontal	Largura	Não	66.0228	1.3223e-04	12
GCUT08	Horizontal	Id	Sim	72.6640	8.5926e-05	8
GCUT08	Horizontal	Id	Não	66.9140	9.0170e-05	8
GCUT08	Maior área	Área	Sim	79.6752	9.2030e-05	6
GCUT08	Maior área	Área	Não	48.0484	1.5783e-04	10
GCUT08	Maior área	Perímetro	Sim	79.6752	9.1696e-05	6
GCUT08	Maior área	Perímetro	Não	48.0484	1.4796e-04	10
GCUT08	Maior área	Altura	Sim	79.6752	9.4318e-05	6
GCUT08	Maior área	Altura	Não	39.6812	1.0753e-04	6
GCUT08	Maior área	Largura	Sim	63.4032	9.3556e-05	6
GCUT08	Maior área	Largura	Não	55.2024	1.4873e-04	10
GCUT08	Maior área	Id	Sim	84.0720	1.4963e-04	10
GCUT08	Maior área	Id	Não	66.9140	1.2717e-04	8
GCUT08	Sobrepostas	Área	Sim	79.6752	1.7033e-04	6
GCUT08	Sobrepostas	Área	Não	61.8724	4.3475e-03	12
GCUT08	Sobrepostas	Perímetro	Sim	79.6752	1.6751e-04	6
GCUT08	Sobrepostas	Perímetro	Não	61.8724	3.9010e-03	12
GCUT08	Sobrepostas	Altura	Sim	79.6752	2.0213e-04	6
GCUT08	Sobrepostas	Altura	Não	47.5604	2.3296e-03	8
GCUT08	Sobrepostas	Largura	Sim	81.3776	2.6455e-04	8
GCUT08	Sobrepostas	Largura	Não	72.9048	4.3007e-03	12
GCUT08	Sobrepostas	Id	Sim	84.0720	5.3163e-04	10
GCUT08	Sobrepostas	Id	Não	66.9140	1.1696e-03	8

Fonte: feito pelo autor.

Tabela 32 – Resultados da instância GCUT09.

Instância	Divisão	Ordenação	Decrescente	Qualidade %	Tempo (s)	Itens %
GCUT09	Vertical	Área	Sim	72.2042	3.6287e-05	30
GCUT09	Vertical	Área	Não	54.3733	5.0402e-05	40
GCUT09	Vertical	Perímetro	Sim	72.2042	3.7813e-05	30
GCUT09	Vertical	Perímetro	Não	54.3733	4.9305e-05	40
GCUT09	Vertical	Altura	Sim	60.5416	3.5763e-05	30
GCUT09	Vertical	Altura	Não	72.1475	6.2943e-05	50
GCUT09	Vertical	Largura	Sim	80.6112	5.1212e-05	40
GCUT09	Vertical	Largura	Não	47.4760	3.7289e-05	30
GCUT09	Vertical	Id	Sim	60.5416	3.7098e-05	30
GCUT09	Vertical	Id	Não	77.7122	5.6410e-05	40
GCUT09	Horizontal	Área	Sim	77.7122	5.0783e-05	40
GCUT09	Horizontal	Área	Não	24.9023	2.9850e-05	20
GCUT09	Horizontal	Perímetro	Sim	77.7122	4.8161e-05	40
GCUT09	Horizontal	Perímetro	Não	24.9023	2.9087e-05	20
GCUT09	Horizontal	Altura	Sim	48.8453	2.9421e-05	20
GCUT09	Horizontal	Altura	Não	24.9023	2.8562e-05	20
GCUT09	Horizontal	Largura	Sim	76.1225	4.9400e-05	40
GCUT09	Horizontal	Largura	Não	38.1883	4.0102e-05	30
GCUT09	Horizontal	Id	Sim	57.3318	3.9721e-05	30
GCUT09	Horizontal	Id	Não	69.6847	5.0068e-05	40
GCUT09	Maior área	Área	Sim	82.2804	9.6083e-05	40
GCUT09	Maior área	Área	Não	54.3733	9.6178e-05	40
GCUT09	Maior área	Perímetro	Sim	82.2804	9.6703e-05	40
GCUT09	Maior área	Perímetro	Não	54.3733	9.8229e-05	40
GCUT09	Maior área	Altura	Sim	60.5416	7.3242e-05	30
GCUT09	Maior área	Altura	Não	55.9625	1.0066e-04	40
GCUT09	Maior área	Largura	Sim	82.2004	1.0176e-04	40
GCUT09	Maior área	Largura	Não	63.6610	9.7179e-05	40
GCUT09	Maior área	Id	Sim	60.5416	7.4673e-05	30
GCUT09	Maior área	Id	Não	77.7122	9.5701e-05	40
GCUT09	Sobrepostas	Área	Sim	72.2042	2.8634e-04	30
GCUT09	Sobrepostas	Área	Não	54.3733	5.5189e-04	40
GCUT09	Sobrepostas	Perímetro	Sim	72.2042	2.7690e-04	30
GCUT09	Sobrepostas	Perímetro	Não	54.3733	5.3296e-04	40
GCUT09	Sobrepostas	Altura	Sim	60.5416	1.2355e-04	30
GCUT09	Sobrepostas	Altura	Não	55.9625	5.4507e-04	40
GCUT09	Sobrepostas	Largura	Sim	82.2004	3.2101e-04	40
GCUT09	Sobrepostas	Largura	Não	63.6610	4.4079e-04	40
GCUT09	Sobrepostas	Id	Sim	60.5416	1.1601e-04	30
GCUT09	Sobrepostas	Id	Não	72.5837	4.2529e-04	40

Fonte: feito pelo autor.

Tabela 33 – Resultados da instância GCUT10.

Instância	Divisão	Ordenação	Decrescente	Qualidade %	Tempo (s)	Itens %
GCUT10	Vertical	Área	Sim	85.6445	4.1056e-05	15
GCUT10	Vertical	Área	Não	63.9774	5.7650e-05	20
GCUT10	Vertical	Perímetro	Sim	85.6445	4.3344e-05	15
GCUT10	Vertical	Perímetro	Não	63.9774	5.5075e-05	20
GCUT10	Vertical	Altura	Sim	59.7263	3.1996e-05	10
GCUT10	Vertical	Altura	Não	65.2485	5.8556e-05	20
GCUT10	Vertical	Largura	Sim	87.7079	4.4012e-05	15
GCUT10	Vertical	Largura	Não	38.7183	3.1376e-05	10
GCUT10	Vertical	Id	Sim	70.8190	3.2473e-05	10
GCUT10	Vertical	Id	Não	73.1757	5.3692e-05	20
GCUT10	Horizontal	Área	Sim	85.6445	4.0960e-05	15
GCUT10	Horizontal	Área	Não	44.2233	4.6349e-05	15
GCUT10	Horizontal	Perímetro	Sim	85.6445	4.3678e-05	15
GCUT10	Horizontal	Perímetro	Não	44.5464	4.9639e-05	15
GCUT10	Horizontal	Altura	Sim	59.7263	3.3665e-05	10
GCUT10	Horizontal	Altura	Não	51.2196	4.4298e-05	15
GCUT10	Horizontal	Largura	Sim	68.4206	3.3283e-05	10
GCUT10	Horizontal	Largura	Não	51.2763	4.8447e-05	15
GCUT10	Horizontal	Id	Sim	70.8190	3.3951e-05	10
GCUT10	Horizontal	Id	Não	66.4464	5.7077e-05	20
GCUT10	Maior área	Área	Sim	85.6445	7.7772e-05	15
GCUT10	Maior área	Área	Não	44.2233	8.9073e-05	15
GCUT10	Maior área	Perímetro	Sim	85.6445	7.6580e-05	15
GCUT10	Maior área	Perímetro	Não	44.5464	7.8964e-05	15
GCUT10	Maior área	Altura	Sim	59.7263	5.8508e-05	10
GCUT10	Maior área	Altura	Não	51.2196	7.8487e-05	15
GCUT10	Maior área	Largura	Sim	68.4206	5.8842e-05	10
GCUT10	Maior área	Largura	Não	53.6220	8.0442e-05	15
GCUT10	Maior área	Id	Sim	70.8190	5.7650e-05	10
GCUT10	Maior área	Id	Não	84.2369	1.0438e-04	20
GCUT10	Sobrepostas	Área	Sim	85.6445	1.3237e-04	15
GCUT10	Sobrepostas	Área	Não	61.3080	8.7261e-04	20
GCUT10	Sobrepostas	Perímetro	Sim	85.6445	1.3699e-04	15
GCUT10	Sobrepostas	Perímetro	Não	84.6795	6.9551e-04	25
GCUT10	Sobrepostas	Altura	Sim	59.7263	6.4087e-05	10
GCUT10	Sobrepostas	Altura	Não	51.2196	5.2104e-04	15
GCUT10	Sobrepostas	Largura	Sim	87.7079	1.4372e-04	15
GCUT10	Sobrepostas	Largura	Não	53.6220	1.8420e-04	15
GCUT10	Sobrepostas	Id	Sim	70.8190	6.3419e-05	10
GCUT10	Sobrepostas	Id	Não	73.1757	4.5605e-04	20

Fonte: feito pelo autor.

Tabela 34 – Resultados da instância GCUT11.

Instância	Divisão	Ordenação	Decrescente	Qualidade %	Tempo (s)	Itens %
GCUT11	Vertical	Área	Sim	76.0952	5.3644e-05	10.00
GCUT11	Vertical	Área	Não	51.2010	1.1759e-04	16.67
GCUT11	Vertical	Perímetro	Sim	76.0952	4.8542e-05	10.00
GCUT11	Vertical	Perímetro	Não	51.2010	7.8630e-05	16.67
GCUT11	Vertical	Altura	Sim	59.2969	3.9864e-05	6.67
GCUT11	Vertical	Altura	Não	59.5818	7.9298e-05	16.67
GCUT11	Vertical	Largura	Sim	82.6283	5.2547e-05	10.00
GCUT11	Vertical	Largura	Não	38.3174	5.3453e-05	10.00
GCUT11	Vertical	Id	Sim	66.3607	7.4673e-05	16.67
GCUT11	Vertical	Id	Não	59.7295	5.2404e-05	10.00
GCUT11	Horizontal	Área	Sim	66.0312	4.7159e-05	6.67
GCUT11	Horizontal	Área	Não	66.5978	1.0076e-04	20.00
GCUT11	Horizontal	Perímetro	Sim	66.0312	4.4346e-05	6.67
GCUT11	Horizontal	Perímetro	Não	66.5978	1.0047e-04	20.00
GCUT11	Horizontal	Altura	Sim	59.2969	4.2486e-05	6.67
GCUT11	Horizontal	Altura	Não	46.0496	5.7030e-05	10.00
GCUT11	Horizontal	Largura	Sim	81.7617	6.0749e-05	13.33
GCUT11	Horizontal	Largura	Não	65.9970	1.0128e-04	20.00
GCUT11	Horizontal	Id	Sim	54.3848	5.6457e-05	10.00
GCUT11	Horizontal	Id	Não	62.8011	5.3501e-05	10.00
GCUT11	Maior área	Área	Sim	78.1734	8.7404e-05	10.00
GCUT11	Maior área	Área	Não	51.2010	1.2975e-04	16.67
GCUT11	Maior área	Perímetro	Sim	78.1734	8.4019e-05	10.00
GCUT11	Maior área	Perímetro	Não	51.2010	1.3919e-04	16.67
GCUT11	Maior área	Altura	Sim	59.2969	6.3991e-05	6.67
GCUT11	Maior área	Altura	Não	56.1136	1.1291e-04	13.33
GCUT11	Maior área	Largura	Sim	81.7617	1.0877e-04	13.33
GCUT11	Maior área	Largura	Não	58.7358	1.3380e-04	16.67
GCUT11	Maior área	Id	Sim	79.2448	1.3432e-04	16.67
GCUT11	Maior área	Id	Não	71.8717	1.0791e-04	13.33
GCUT11	Sobrepostas	Área	Sim	78.1734	1.5130e-04	10.00
GCUT11	Sobrepostas	Área	Não	65.9970	1.8229e-03	20.00
GCUT11	Sobrepostas	Perímetro	Sim	78.1734	1.4205e-04	10.00
GCUT11	Sobrepostas	Perímetro	Não	66.5978	1.8986e-03	20.00
GCUT11	Sobrepostas	Altura	Sim	59.2969	1.0004e-04	6.67
GCUT11	Sobrepostas	Altura	Não	68.2558	1.6176e-03	16.67
GCUT11	Sobrepostas	Largura	Sim	82.6283	2.6569e-04	10.00
GCUT11	Sobrepostas	Largura	Não	58.7358	1.2592e-03	16.67
GCUT11	Sobrepostas	Id	Sim	64.4488	6.6929e-04	13.33
GCUT11	Sobrepostas	Id	Não	71.8717	2.0966e-04	13.33

Fonte: feito pelo autor.

Tabela 35 – Resultados da instância GCUT12.

Instância	Divisão	Ordenação	Decrescente	Qualidade %	Tempo (s)	Itens %
GCUT12	Vertical	Área	Sim	86.2188	5.8460e-05	6
GCUT12	Vertical	Área	Não	75.9497	1.0958e-04	12
GCUT12	Vertical	Perímetro	Sim	86.2188	5.8031e-05	6
GCUT12	Vertical	Perímetro	Não	77.3433	1.0786e-04	12
GCUT12	Vertical	Altura	Sim	86.2188	5.8746e-05	6
GCUT12	Vertical	Altura	Não	62.7312	9.5367e-05	10
GCUT12	Vertical	Largura	Sim	87.9958	6.5947e-05	6
GCUT12	Vertical	Largura	Não	42.5142	6.1798e-05	6
GCUT12	Vertical	Id	Sim	84.8231	9.5415e-05	10
GCUT12	Vertical	Id	Não	69.2716	6.4421e-05	6
GCUT12	Horizontal	Área	Sim	86.2188	5.9843e-05	6
GCUT12	Horizontal	Área	Não	45.9083	9.4128e-05	8
GCUT12	Horizontal	Perímetro	Sim	86.2188	6.0368e-05	6
GCUT12	Horizontal	Perímetro	Não	46.0824	9.9087e-05	8
GCUT12	Horizontal	Altura	Sim	86.2188	6.1083e-05	6
GCUT12	Horizontal	Altura	Não	46.6858	7.3814e-05	6
GCUT12	Horizontal	Largura	Sim	80.0304	6.0701e-05	6
GCUT12	Horizontal	Largura	Não	67.3601	1.0519e-04	10
GCUT12	Horizontal	Id	Sim	75.9252	8.7500e-05	8
GCUT12	Horizontal	Id	Não	81.7816	9.0074e-05	8
GCUT12	Maior área	Área	Sim	86.2188	9.4175e-05	6
GCUT12	Maior área	Área	Não	88.4597	2.2726e-04	14
GCUT12	Maior área	Perímetro	Sim	86.2188	9.5892e-05	6
GCUT12	Maior área	Perímetro	Não	88.4597	1.9875e-04	14
GCUT12	Maior área	Altura	Sim	86.2188	9.4461e-05	6
GCUT12	Maior área	Altura	Não	62.4328	1.2975e-04	8
GCUT12	Maior área	Largura	Sim	80.0304	9.6607e-05	6
GCUT12	Maior área	Largura	Não	56.8740	1.3413e-04	8
GCUT12	Maior área	Id	Sim	75.9252	1.3671e-04	8
GCUT12	Maior área	Id	Não	81.6102	1.2927e-04	8
GCUT12	Sobrepostas	Área	Sim	86.2188	1.7200e-04	6
GCUT12	Sobrepostas	Área	Não	72.7781	2.4309e-03	12
GCUT12	Sobrepostas	Perímetro	Sim	86.2188	1.6713e-04	6
GCUT12	Sobrepostas	Perímetro	Não	58.4183	2.6996e-03	10
GCUT12	Sobrepostas	Altura	Sim	86.2188	1.6260e-04	6
GCUT12	Sobrepostas	Altura	Não	62.4328	1.4399e-03	8
GCUT12	Sobrepostas	Largura	Sim	80.0304	2.3909e-04	6
GCUT12	Sobrepostas	Largura	Não	70.6035	1.0150e-03	10
GCUT12	Sobrepostas	Id	Sim	75.9252	3.9520e-04	8
GCUT12	Sobrepostas	Id	Não	81.6102	5.0712e-04	8

Fonte: feito pelo autor.

Tabela 36 – Resultados da instância GCUT13.

Instância	Divisão	Ordenação	Decrescente	Qualidade %	Tempo (s)	Itens %
GCUT13	Vertical	Área	Sim	91.2187	1.5383e-04	34.38
GCUT13	Vertical	Área	Não	22.6051	1.9054e-04	37.50
GCUT13	Vertical	Perímetro	Sim	91.2187	1.5855e-04	34.38
GCUT13	Vertical	Perímetro	Não	34.3835	2.1954e-04	46.88
GCUT13	Vertical	Altura	Sim	91.4087	1.8144e-04	40.62
GCUT13	Vertical	Altura	Não	30.7244	2.1696e-04	43.75
GCUT13	Vertical	Largura	Sim	81.8049	2.3775e-04	53.12
GCUT13	Vertical	Largura	Não	10.1909	1.0586e-04	21.88
GCUT13	Vertical	Id	Sim	81.9018	1.8449e-04	40.62
GCUT13	Vertical	Id	Não	10.1909	1.0376e-04	21.88
GCUT13	Horizontal	Área	Sim	84.1311	2.3031e-04	46.88
GCUT13	Horizontal	Área	Não	34.0000	2.5044e-04	46.88
GCUT13	Horizontal	Perímetro	Sim	84.7578	2.5582e-04	50.00
GCUT13	Horizontal	Perímetro	Não	34.0502	2.5339e-04	46.88
GCUT13	Horizontal	Altura	Sim	83.4785	2.5315e-04	50.00
GCUT13	Horizontal	Altura	Não	16.3304	1.6928e-04	31.25
GCUT13	Horizontal	Largura	Sim	47.0389	2.7003e-04	56.25
GCUT13	Horizontal	Largura	Não	37.3764	2.7122e-04	50.00
GCUT13	Horizontal	Id	Sim	47.0389	2.6679e-04	56.25
GCUT13	Horizontal	Id	Não	37.3764	2.7294e-04	50.00
GCUT13	Maior área	Área	Sim	91.2187	2.7618e-04	34.38
GCUT13	Maior área	Área	Não	40.5105	4.6577e-04	53.12
GCUT13	Maior área	Perímetro	Sim	91.2187	2.6989e-04	34.38
GCUT13	Maior área	Perímetro	Não	40.5105	4.6430e-04	53.12
GCUT13	Maior área	Altura	Sim	91.4087	3.1524e-04	40.62
GCUT13	Maior área	Altura	Não	32.9340	3.7761e-04	43.75
GCUT13	Maior área	Largura	Sim	73.4412	5.1422e-04	62.50
GCUT13	Maior área	Largura	Não	40.0229	3.9906e-04	46.88
GCUT13	Maior área	Id	Sim	75.5894	4.9195e-04	59.38
GCUT13	Maior área	Id	Não	40.0229	4.0412e-04	46.88
GCUT13	Sobrepostas	Área	Sim	91.2187	1.2565e-02	34.38
GCUT13	Sobrepostas	Área	Não	58.9385	5.0150e-02	65.62
GCUT13	Sobrepostas	Perímetro	Sim	91.2187	1.2575e-02	34.38
GCUT13	Sobrepostas	Perímetro	Não	54.0947	4.5969e-02	62.50
GCUT13	Sobrepostas	Altura	Sim	91.2187	1.2634e-02	34.38
GCUT13	Sobrepostas	Altura	Não	58.9385	4.9757e-02	65.62
GCUT13	Sobrepostas	Largura	Sim	72.3215	2.6286e-02	65.62
GCUT13	Sobrepostas	Largura	Não	47.4392	2.4019e-02	46.88
GCUT13	Sobrepostas	Id	Sim	77.8546	2.5180e-02	62.50
GCUT13	Sobrepostas	Id	Não	47.4392	2.4177e-02	46.88

Fonte: feito pelo autor.

A.3 NGCUT

Tabela 37 – Resultados da instância NGCUT01.

Instância	Divisão	Ordenação	Decrescente	Qualidade %	Tempo (s)	Itens %
NGCUT01	Vertical	Área	Sim	78	5.2118e-05	40
NGCUT01	Vertical	Área	Não	70	4.1628e-05	40
NGCUT01	Vertical	Perímetro	Sim	62	3.1900e-05	30
NGCUT01	Vertical	Perímetro	Não	60	3.3283e-05	30
NGCUT01	Vertical	Altura	Sim	78	4.9734e-05	40
NGCUT01	Vertical	Altura	Não	70	4.3726e-05	40
NGCUT01	Vertical	Largura	Sim	72	3.6716e-05	40
NGCUT01	Vertical	Largura	Não	78	4.6730e-05	40
NGCUT01	Vertical	Id	Sim	76	4.6873e-05	40
NGCUT01	Vertical	Id	Não	78	4.8113e-05	40
NGCUT01	Horizontal	Área	Sim	62	3.2616e-05	30
NGCUT01	Horizontal	Área	Não	92	4.5109e-05	50
NGCUT01	Horizontal	Perímetro	Sim	62	2.9373e-05	30
NGCUT01	Horizontal	Perímetro	Não	76	3.9148e-05	40
NGCUT01	Horizontal	Altura	Sim	78	4.1008e-05	40
NGCUT01	Horizontal	Altura	Não	92	4.3249e-05	50
NGCUT01	Horizontal	Largura	Sim	92	4.3726e-05	50
NGCUT01	Horizontal	Largura	Não	78	4.5204e-05	40
NGCUT01	Horizontal	Id	Sim	76	4.7541e-05	40
NGCUT01	Horizontal	Id	Não	58	3.4761e-05	30
NGCUT01	Maior área	Área	Sim	78	9.8467e-05	40
NGCUT01	Maior área	Área	Não	92	9.6226e-05	50
NGCUT01	Maior área	Perímetro	Sim	62	6.8140e-05	30
NGCUT01	Maior área	Perímetro	Não	60	6.2895e-05	30
NGCUT01	Maior área	Altura	Sim	78	9.4843e-05	40
NGCUT01	Maior área	Altura	Não	92	9.2697e-05	50
NGCUT01	Maior área	Largura	Sim	92	8.9216e-05	50
NGCUT01	Maior área	Largura	Não	78	9.4032e-05	40
NGCUT01	Maior área	Id	Sim	76	9.8419e-05	40
NGCUT01	Maior área	Id	Não	78	9.4271e-05	40
NGCUT01	Sobrepostas	Área	Sim	62	2.3079e-04	30
NGCUT01	Sobrepostas	Área	Não	70	3.5377e-04	40
NGCUT01	Sobrepostas	Perímetro	Sim	62	1.1468e-04	30
NGCUT01	Sobrepostas	Perímetro	Não	76	1.9755e-04	40
NGCUT01	Sobrepostas	Altura	Sim	78	1.9264e-04	40
NGCUT01	Sobrepostas	Altura	Não	92	4.0889e-04	50
NGCUT01	Sobrepostas	Largura	Sim	92	2.8644e-04	50
NGCUT01	Sobrepostas	Largura	Não	78	1.9646e-04	40
NGCUT01	Sobrepostas	Id	Sim	76	2.0156e-04	40
NGCUT01	Sobrepostas	Id	Não	58	2.1682e-04	30

Fonte: feito pelo autor.

Tabela 38 – Resultados da instância NGCUT02.

Instância	Divisão	Ordenação	Decrescente	Qualidade %	Tempo (s)	Itens %
NGCUT02	Vertical	Área	Sim	97	6.4945e-05	29.41
NGCUT02	Vertical	Área	Não	76	6.9380e-05	41.18
NGCUT02	Vertical	Perímetro	Sim	97	4.2677e-05	29.41
NGCUT02	Vertical	Perímetro	Não	76	7.1621e-05	41.18
NGCUT02	Vertical	Altura	Sim	78	4.8161e-05	29.41
NGCUT02	Vertical	Altura	Não	56	6.3896e-05	35.29
NGCUT02	Vertical	Largura	Sim	97	4.5204e-05	29.41
NGCUT02	Vertical	Largura	Não	78	4.6492e-05	29.41
NGCUT02	Vertical	Id	Sim	70	5.0878e-05	23.53
NGCUT02	Vertical	Id	Não	64	5.1641e-05	35.29
NGCUT02	Horizontal	Área	Sim	91	5.7173e-05	29.41
NGCUT02	Horizontal	Área	Não	52	7.7677e-05	35.29
NGCUT02	Horizontal	Perímetro	Sim	91	5.7316e-05	29.41
NGCUT02	Horizontal	Perímetro	Não	52	7.7677e-05	35.29
NGCUT02	Horizontal	Altura	Sim	90	5.9843e-05	41.18
NGCUT02	Horizontal	Altura	Não	52	7.8297e-05	35.29
NGCUT02	Horizontal	Largura	Sim	91	8.0681e-05	29.41
NGCUT02	Horizontal	Largura	Não	88	6.1321e-05	41.18
NGCUT02	Horizontal	Id	Sim	64	8.2493e-05	35.29
NGCUT02	Horizontal	Id	Não	82	8.4543e-05	52.94
NGCUT02	Maior área	Área	Sim	91	1.0653e-04	29.41
NGCUT02	Maior área	Área	Não	52	1.3828e-04	35.29
NGCUT02	Maior área	Perímetro	Sim	91	1.1015e-04	29.41
NGCUT02	Maior área	Perímetro	Não	52	1.4281e-04	35.29
NGCUT02	Maior área	Altura	Sim	78	9.8372e-05	29.41
NGCUT02	Maior área	Altura	Não	52	1.3766e-04	35.29
NGCUT02	Maior área	Largura	Sim	91	1.0285e-04	29.41
NGCUT02	Maior área	Largura	Não	78	1.0209e-04	29.41
NGCUT02	Maior área	Id	Sim	64	1.3709e-04	35.29
NGCUT02	Maior área	Id	Não	82	1.6999e-04	52.94
NGCUT02	Sobrepostas	Área	Sim	97	3.8362e-04	29.41
NGCUT02	Sobrepostas	Área	Não	76	1.1484e-03	41.18
NGCUT02	Sobrepostas	Perímetro	Sim	97	3.7675e-04	29.41
NGCUT02	Sobrepostas	Perímetro	Não	76	1.0783e-03	41.18
NGCUT02	Sobrepostas	Altura	Sim	90	7.0357e-04	41.18
NGCUT02	Sobrepostas	Altura	Não	86	1.3473e-03	47.06
NGCUT02	Sobrepostas	Largura	Sim	91	5.8994e-04	29.41
NGCUT02	Sobrepostas	Largura	Não	88	5.5323e-04	41.18
NGCUT02	Sobrepostas	Id	Sim	88	7.5531e-04	41.18
NGCUT02	Sobrepostas	Id	Não	82	9.3389e-04	52.94

Fonte: feito pelo autor.

Tabela 39 – Resultados da instância NGCUT03.

Instância	Divisão	Ordenação	Decrescente	Qualidade %	Tempo (s)	Itens %
NGCUT03	Vertical	Área	Sim	90	5.8460e-05	28.57
NGCUT03	Vertical	Área	Não	66	8.5401e-05	38.10
NGCUT03	Vertical	Perímetro	Sim	84	5.0449e-05	23.81
NGCUT03	Vertical	Perímetro	Não	57	7.4768e-05	33.33
NGCUT03	Vertical	Altura	Sim	85	5.8031e-05	28.57
NGCUT03	Vertical	Altura	Não	66	8.5020e-05	38.10
NGCUT03	Vertical	Largura	Sim	88	5.8746e-05	28.57
NGCUT03	Vertical	Largura	Não	64	5.2404e-05	23.81
NGCUT03	Vertical	Id	Sim	93	7.2002e-05	38.10
NGCUT03	Vertical	Id	Não	70	7.5722e-05	38.10
NGCUT03	Horizontal	Área	Sim	81	5.0592e-05	19.05
NGCUT03	Horizontal	Área	Não	72	9.5749e-05	42.86
NGCUT03	Horizontal	Perímetro	Sim	90	6.5947e-05	33.33
NGCUT03	Horizontal	Perímetro	Não	72	9.6417e-05	42.86
NGCUT03	Horizontal	Altura	Sim	93	6.6376e-05	38.10
NGCUT03	Horizontal	Altura	Não	80	9.4891e-05	42.86
NGCUT03	Horizontal	Largura	Sim	90	6.8521e-05	33.33
NGCUT03	Horizontal	Largura	Não	78	8.6117e-05	38.10
NGCUT03	Horizontal	Id	Sim	90	1.5993e-04	52.38
NGCUT03	Horizontal	Id	Não	70	9.4032e-05	38.10
NGCUT03	Maior área	Área	Sim	80	9.9087e-05	23.81
NGCUT03	Maior área	Área	Não	60	1.8597e-04	38.10
NGCUT03	Maior área	Perímetro	Sim	90	1.3046e-04	33.33
NGCUT03	Maior área	Perímetro	Não	72	1.8730e-04	42.86
NGCUT03	Maior área	Altura	Sim	85	1.2250e-04	28.57
NGCUT03	Maior área	Altura	Não	80	1.8835e-04	42.86
NGCUT03	Maior área	Largura	Sim	90	1.3037e-04	33.33
NGCUT03	Maior área	Largura	Não	78	1.6723e-04	38.10
NGCUT03	Maior área	Id	Sim	90	2.0771e-04	52.38
NGCUT03	Maior área	Id	Não	70	1.7476e-04	38.10
NGCUT03	Sobrepostas	Área	Sim	90	6.0635e-04	28.57
NGCUT03	Sobrepostas	Área	Não	86	5.3571e-03	42.86
NGCUT03	Sobrepostas	Perímetro	Sim	84	5.4984e-04	23.81
NGCUT03	Sobrepostas	Perímetro	Não	72	1.9072e-03	42.86
NGCUT03	Sobrepostas	Altura	Sim	93	9.1248e-04	38.10
NGCUT03	Sobrepostas	Altura	Não	88	2.9168e-03	47.62
NGCUT03	Sobrepostas	Largura	Sim	90	7.1287e-04	33.33
NGCUT03	Sobrepostas	Largura	Não	72	3.0554e-03	33.33
NGCUT03	Sobrepostas	Id	Sim	93	3.1544e-03	38.10
NGCUT03	Sobrepostas	Id	Não	80	2.5147e-03	42.86

Fonte: feito pelo autor.

Tabela 40 – Resultados da instância NGCUT04.

Instância	Divisão	Ordenação	Decrescente	Qualidade %	Tempo (s)	Itens %
NGCUT04	Vertical	Área	Sim	70	3.1900e-05	57.14
NGCUT04	Vertical	Área	Não	20	2.4557e-05	28.57
NGCUT04	Vertical	Perímetro	Sim	78	4.2486e-05	71.43
NGCUT04	Vertical	Perímetro	Não	30	2.2507e-05	28.57
NGCUT04	Vertical	Altura	Sim	30	2.2078e-05	28.57
NGCUT04	Vertical	Altura	Não	20	2.5654e-05	28.57
NGCUT04	Vertical	Largura	Sim	78	4.4298e-05	71.43
NGCUT04	Vertical	Largura	Não	30	2.3174e-05	28.57
NGCUT04	Vertical	Id	Sim	62	3.8624e-05	57.14
NGCUT04	Vertical	Id	Não	70	3.3379e-05	57.14
NGCUT04	Horizontal	Área	Sim	92	4.6587e-05	85.71
NGCUT04	Horizontal	Área	Não	52	4.6492e-05	57.14
NGCUT04	Horizontal	Perímetro	Sim	78	4.5633e-05	71.43
NGCUT04	Horizontal	Perímetro	Não	52	4.5347e-05	57.14
NGCUT04	Horizontal	Altura	Sim	70	3.6812e-05	57.14
NGCUT04	Horizontal	Altura	Não	78	4.7064e-05	71.43
NGCUT04	Horizontal	Largura	Sim	78	4.7302e-05	71.43
NGCUT04	Horizontal	Largura	Não	52	4.5443e-05	57.14
NGCUT04	Horizontal	Id	Sim	88	5.5075e-05	85.71
NGCUT04	Horizontal	Id	Não	92	4.6778e-05	85.71
NGCUT04	Maior área	Área	Sim	92	1.0519e-04	85.71
NGCUT04	Maior área	Área	Não	52	8.8549e-05	57.14
NGCUT04	Maior área	Perímetro	Sim	78	9.6798e-05	71.43
NGCUT04	Maior área	Perímetro	Não	52	8.8215e-05	57.14
NGCUT04	Maior área	Altura	Sim	70	7.4863e-05	57.14
NGCUT04	Maior área	Altura	Não	78	1.0138e-04	71.43
NGCUT04	Maior área	Largura	Sim	78	1.3809e-04	71.43
NGCUT04	Maior área	Largura	Não	52	1.1043e-04	57.14
NGCUT04	Maior área	Id	Sim	88	1.0715e-04	85.71
NGCUT04	Maior área	Id	Não	70	7.6199e-05	57.14
NGCUT04	Sobrepostas	Área	Sim	92	4.2348e-04	85.71
NGCUT04	Sobrepostas	Área	Não	52	1.9498e-04	57.14
NGCUT04	Sobrepostas	Perímetro	Sim	78	2.9278e-04	71.43
NGCUT04	Sobrepostas	Perímetro	Não	52	1.9488e-04	57.14
NGCUT04	Sobrepostas	Altura	Sim	70	1.8783e-04	57.14
NGCUT04	Sobrepostas	Altura	Não	78	2.9354e-04	71.43
NGCUT04	Sobrepostas	Largura	Sim	78	2.9468e-04	71.43
NGCUT04	Sobrepostas	Largura	Não	52	1.9817e-04	57.14
NGCUT04	Sobrepostas	Id	Sim	88	4.2405e-04	85.71
NGCUT04	Sobrepostas	Id	Não	92	4.0999e-04	85.71

Fonte: feito pelo autor.

Tabela 41 – Resultados da instância NGCUT05.

Instância	Divisão	Ordenação	Decrescente	Qualidade %	Tempo (s)	Itens %
NGCUT05	Vertical	Área	Sim	92.6667	4.7255e-05	35.71
NGCUT05	Vertical	Área	Não	53.3333	4.8971e-05	35.71
NGCUT05	Vertical	Perímetro	Sim	92.6667	4.5824e-05	35.71
NGCUT05	Vertical	Perímetro	Não	31.3333	2.6178e-05	14.29
NGCUT05	Vertical	Altura	Sim	71.3333	4.1008e-05	28.57
NGCUT05	Vertical	Altura	Não	75.3333	5.7173e-05	42.86
NGCUT05	Vertical	Largura	Sim	92.6667	5.3263e-05	42.86
NGCUT05	Vertical	Largura	Não	31.3333	2.3508e-05	14.29
NGCUT05	Vertical	Id	Sim	75.3333	5.4264e-05	42.86
NGCUT05	Vertical	Id	Não	71.3333	4.6730e-05	28.57
NGCUT05	Horizontal	Área	Sim	79.3333	4.9973e-05	28.57
NGCUT05	Horizontal	Área	Não	70.0000	7.4196e-05	42.86
NGCUT05	Horizontal	Perímetro	Sim	79.3333	4.8017e-05	28.57
NGCUT05	Horizontal	Perímetro	Não	60.0000	6.4707e-05	35.71
NGCUT05	Horizontal	Altura	Sim	78.6667	5.1451e-05	35.71
NGCUT05	Horizontal	Altura	Não	70.0000	7.4196e-05	42.86
NGCUT05	Horizontal	Largura	Sim	79.3333	6.1131e-05	35.71
NGCUT05	Horizontal	Largura	Não	78.6667	5.0449e-05	35.71
NGCUT05	Horizontal	Id	Sim	83.3333	6.7043e-05	42.86
NGCUT05	Horizontal	Id	Não	65.3333	5.2023e-05	28.57
NGCUT05	Maior área	Área	Sim	79.3333	9.6226e-05	28.57
NGCUT05	Maior área	Área	Não	70.0000	1.3661e-04	42.86
NGCUT05	Maior área	Perímetro	Sim	79.3333	9.4366e-05	28.57
NGCUT05	Maior área	Perímetro	Não	60.0000	1.1993e-04	35.71
NGCUT05	Maior área	Altura	Sim	71.3333	8.7118e-05	28.57
NGCUT05	Maior área	Altura	Não	70.0000	1.3647e-04	42.86
NGCUT05	Maior área	Largura	Sim	79.3333	1.0939e-04	35.71
NGCUT05	Maior área	Largura	Não	78.6667	1.0514e-04	35.71
NGCUT05	Maior área	Id	Sim	83.3333	1.2956e-04	42.86
NGCUT05	Maior área	Id	Não	58.0000	7.3004e-05	21.43
NGCUT05	Sobrepostas	Área	Sim	92.6667	3.0928e-04	35.71
NGCUT05	Sobrepostas	Área	Não	73.3333	6.9175e-04	42.86
NGCUT05	Sobrepostas	Perímetro	Sim	92.6667	3.1242e-04	35.71
NGCUT05	Sobrepostas	Perímetro	Não	60.0000	3.4189e-04	35.71
NGCUT05	Sobrepostas	Altura	Sim	78.6667	2.9521e-04	35.71
NGCUT05	Sobrepostas	Altura	Não	70.0000	5.8637e-04	42.86
NGCUT05	Sobrepostas	Largura	Sim	92.6667	4.3769e-04	42.86
NGCUT05	Sobrepostas	Largura	Não	78.6667	3.3417e-04	35.71
NGCUT05	Sobrepostas	Id	Sim	83.3333	4.2019e-04	42.86
NGCUT05	Sobrepostas	Id	Não	65.3333	2.8706e-04	28.57

Fonte: feito pelo autor.

Tabela 42 – Resultados da instância NGCUT06.

Instância	Divisão	Ordenação	Decrescente	Qualidade %	Tempo (s)	Itens %
NGCUT06	Vertical	Área	Sim	80.0000	7.0620e-05	40.00
NGCUT06	Vertical	Área	Não	70.6667	8.5306e-05	46.67
NGCUT06	Vertical	Perímetro	Sim	90.6667	6.6662e-05	46.67
NGCUT06	Vertical	Perímetro	Não	70.6667	8.7786e-05	46.67
NGCUT06	Vertical	Altura	Sim	68.0000	8.3065e-05	40.00
NGCUT06	Vertical	Altura	Não	78.6667	7.3767e-05	46.67
NGCUT06	Vertical	Largura	Sim	76.0000	7.6914e-05	46.67
NGCUT06	Vertical	Largura	Não	70.6667	1.0910e-04	46.67
NGCUT06	Vertical	Id	Sim	72.0000	6.8808e-05	40.00
NGCUT06	Vertical	Id	Não	70.6667	7.6628e-05	46.67
NGCUT06	Horizontal	Área	Sim	74.6667	6.7043e-05	40.00
NGCUT06	Horizontal	Área	Não	62.6667	7.5150e-05	40.00
NGCUT06	Horizontal	Perímetro	Sim	64.0000	7.0953e-05	33.33
NGCUT06	Horizontal	Perímetro	Não	69.3333	6.6137e-05	40.00
NGCUT06	Horizontal	Altura	Sim	70.6667	8.0633e-05	46.67
NGCUT06	Horizontal	Altura	Não	57.3333	6.6996e-05	33.33
NGCUT06	Horizontal	Largura	Sim	57.3333	7.3099e-05	33.33
NGCUT06	Horizontal	Largura	Não	65.3333	6.5279e-05	40.00
NGCUT06	Horizontal	Id	Sim	72.0000	7.1764e-05	40.00
NGCUT06	Horizontal	Id	Não	74.6667	7.4863e-05	40.00
NGCUT06	Maior área	Área	Sim	60.0000	1.1730e-04	33.33
NGCUT06	Maior área	Área	Não	70.6667	1.6584e-04	46.67
NGCUT06	Maior área	Perímetro	Sim	64.0000	1.1811e-04	33.33
NGCUT06	Maior área	Perímetro	Não	70.6667	1.6823e-04	46.67
NGCUT06	Maior área	Altura	Sim	70.6667	1.7061e-04	46.67
NGCUT06	Maior área	Altura	Não	57.3333	1.1926e-04	33.33
NGCUT06	Maior área	Largura	Sim	57.3333	1.2169e-04	33.33
NGCUT06	Maior área	Largura	Não	70.6667	1.6284e-04	46.67
NGCUT06	Maior área	Id	Sim	72.0000	1.2612e-04	40.00
NGCUT06	Maior área	Id	Não	80.0000	1.5645e-04	46.67
NGCUT06	Sobrepostas	Área	Sim	85.3333	7.6637e-04	46.67
NGCUT06	Sobrepostas	Área	Não	70.6667	2.6751e-03	46.67
NGCUT06	Sobrepostas	Perímetro	Sim	77.3333	7.2384e-04	40.00
NGCUT06	Sobrepostas	Perímetro	Não	88.0000	2.2838e-03	53.33
NGCUT06	Sobrepostas	Altura	Sim	70.6667	1.5313e-03	46.67
NGCUT06	Sobrepostas	Altura	Não	73.3333	1.1046e-03	46.67
NGCUT06	Sobrepostas	Largura	Sim	70.6667	1.3248e-03	40.00
NGCUT06	Sobrepostas	Largura	Não	84.0000	1.2783e-03	53.33
NGCUT06	Sobrepostas	Id	Sim	84.0000	8.9526e-04	46.67
NGCUT06	Sobrepostas	Id	Não	70.6667	1.6067e-03	46.67

Fonte: feito pelo autor.

Tabela 43 – Resultados da instância NGCUT07.

Instância	Divisão	Ordenação	Decrescente	Qualidade %	Tempo (s)	Itens %
NGCUT07	Vertical	Área	Sim	33.7500	7.6437e-05	87.50
NGCUT07	Vertical	Área	Não	20.2500	5.6314e-05	75.00
NGCUT07	Vertical	Perímetro	Sim	100.0000	8.2302e-05	100.00
NGCUT07	Vertical	Perímetro	Não	20.2500	5.7983e-05	75.00
NGCUT07	Vertical	Altura	Sim	20.2500	6.4945e-05	75.00
NGCUT07	Vertical	Altura	Não	20.2500	6.6662e-05	75.00
NGCUT07	Vertical	Largura	Sim	100.0000	7.8249e-05	100.00
NGCUT07	Vertical	Largura	Não	8.2500	5.2738e-05	62.50
NGCUT07	Vertical	Id	Sim	20.2500	8.7023e-05	75.00
NGCUT07	Vertical	Id	Não	20.2500	6.6662e-05	75.00
NGCUT07	Horizontal	Área	Sim	100.0000	7.4863e-05	100.00
NGCUT07	Horizontal	Área	Não	33.7500	6.9475e-05	87.50
NGCUT07	Horizontal	Perímetro	Sim	100.0000	7.3004e-05	100.00
NGCUT07	Horizontal	Perímetro	Não	33.7500	6.8188e-05	87.50
NGCUT07	Horizontal	Altura	Sim	100.0000	7.5722e-05	100.00
NGCUT07	Horizontal	Altura	Não	100.0000	7.8344e-05	100.00
NGCUT07	Horizontal	Largura	Sim	100.0000	7.3433e-05	100.00
NGCUT07	Horizontal	Largura	Não	33.7500	7.2432e-05	87.50
NGCUT07	Horizontal	Id	Sim	100.0000	7.7772e-05	100.00
NGCUT07	Horizontal	Id	Não	100.0000	7.6103e-05	100.00
NGCUT07	Maior área	Área	Sim	100.0000	1.7600e-04	100.00
NGCUT07	Maior área	Área	Não	33.7500	2.1367e-04	87.50
NGCUT07	Maior área	Perímetro	Sim	100.0000	1.6670e-04	100.00
NGCUT07	Maior área	Perímetro	Não	33.7500	1.3680e-04	87.50
NGCUT07	Maior área	Altura	Sim	33.7500	1.4863e-04	87.50
NGCUT07	Maior área	Altura	Não	100.0000	1.6642e-04	100.00
NGCUT07	Maior área	Largura	Sim	100.0000	1.6522e-04	100.00
NGCUT07	Maior área	Largura	Não	33.7500	1.5197e-04	87.50
NGCUT07	Maior área	Id	Sim	100.0000	1.7247e-04	100.00
NGCUT07	Maior área	Id	Não	33.7500	1.4720e-04	87.50
NGCUT07	Sobrepostas	Área	Sim	100.0000	7.4873e-04	100.00
NGCUT07	Sobrepostas	Área	Não	33.7500	5.7478e-04	87.50
NGCUT07	Sobrepostas	Perímetro	Sim	100.0000	7.4654e-04	100.00
NGCUT07	Sobrepostas	Perímetro	Não	33.7500	5.6620e-04	87.50
NGCUT07	Sobrepostas	Altura	Sim	100.0000	7.3676e-04	100.00
NGCUT07	Sobrepostas	Altura	Não	100.0000	7.4940e-04	100.00
NGCUT07	Sobrepostas	Largura	Sim	100.0000	7.4258e-04	100.00
NGCUT07	Sobrepostas	Largura	Não	33.7500	5.6992e-04	87.50
NGCUT07	Sobrepostas	Id	Sim	100.0000	7.4096e-04	100.00
NGCUT07	Sobrepostas	Id	Não	100.0000	7.4439e-04	100.00

Fonte: feito pelo autor.

Tabela 44 – Resultados da instância NGCUT08.

Instância	Divisão	Ordenação	Decrescente	Qualidade %	Tempo (s)	Itens %
NGCUT08	Vertical	Área	Sim	79.5000	4.1962e-05	30.77
NGCUT08	Vertical	Área	Não	39.5000	6.5947e-05	46.15
NGCUT08	Vertical	Perímetro	Sim	85.2500	5.5790e-05	46.15
NGCUT08	Vertical	Perímetro	Não	33.0000	4.7016e-05	30.77
NGCUT08	Vertical	Altura	Sim	55.0000	5.0259e-05	38.46
NGCUT08	Vertical	Altura	Não	39.5000	6.5374e-05	46.15
NGCUT08	Vertical	Largura	Sim	85.2500	5.5313e-05	46.15
NGCUT08	Vertical	Largura	Não	61.0000	5.7077e-05	46.15
NGCUT08	Vertical	Id	Sim	73.0000	6.7854e-05	53.85
NGCUT08	Vertical	Id	Não	38.2500	4.9448e-05	30.77
NGCUT08	Horizontal	Área	Sim	88.7500	6.5470e-05	46.15
NGCUT08	Horizontal	Área	Não	62.7500	8.8406e-05	61.54
NGCUT08	Horizontal	Perímetro	Sim	85.2500	7.0429e-05	46.15
NGCUT08	Horizontal	Perímetro	Não	62.7500	8.6260e-05	61.54
NGCUT08	Horizontal	Altura	Sim	73.0000	6.7329e-05	53.85
NGCUT08	Horizontal	Altura	Não	58.2500	7.1907e-05	46.15
NGCUT08	Horizontal	Largura	Sim	85.2500	6.8331e-05	46.15
NGCUT08	Horizontal	Largura	Não	51.5000	8.2254e-05	53.85
NGCUT08	Horizontal	Id	Sim	69.5000	7.4482e-05	46.15
NGCUT08	Horizontal	Id	Não	62.7500	9.0456e-05	61.54
NGCUT08	Maior área	Área	Sim	85.5000	1.1330e-04	38.46
NGCUT08	Maior área	Área	Não	51.5000	1.5717e-04	53.85
NGCUT08	Maior área	Perímetro	Sim	85.2500	1.3208e-04	46.15
NGCUT08	Maior área	Perímetro	Não	53.5000	1.5636e-04	53.85
NGCUT08	Maior área	Altura	Sim	73.0000	1.5693e-04	53.85
NGCUT08	Maior área	Altura	Não	58.2500	1.3542e-04	46.15
NGCUT08	Maior área	Largura	Sim	85.2500	1.4672e-04	46.15
NGCUT08	Maior área	Largura	Não	67.5000	1.7762e-04	61.54
NGCUT08	Maior área	Id	Sim	69.5000	1.3475e-04	46.15
NGCUT08	Maior área	Id	Não	64.7500	1.7510e-04	61.54
NGCUT08	Sobrepostas	Área	Sim	88.7500	4.4518e-04	46.15
NGCUT08	Sobrepostas	Área	Não	62.7500	1.3292e-03	61.54
NGCUT08	Sobrepostas	Perímetro	Sim	85.2500	4.4403e-04	46.15
NGCUT08	Sobrepostas	Perímetro	Não	64.7500	1.5019e-03	61.54
NGCUT08	Sobrepostas	Altura	Sim	73.0000	6.6929e-04	53.85
NGCUT08	Sobrepostas	Altura	Não	58.2500	1.9360e-03	46.15
NGCUT08	Sobrepostas	Largura	Sim	85.2500	4.3097e-04	46.15
NGCUT08	Sobrepostas	Largura	Não	67.5000	1.2271e-03	61.54
NGCUT08	Sobrepostas	Id	Sim	69.5000	9.0876e-04	46.15
NGCUT08	Sobrepostas	Id	Não	64.7500	1.2555e-03	61.54

Fonte: feito pelo autor.

Tabela 45 – Resultados da instância NGCUT09.

Instância	Divisão	Ordenação	Decrescente	Qualidade %	Tempo (s)	Itens %
NGCUT09	Vertical	Área	Sim	55.5000	7.1955e-05	33.33
NGCUT09	Vertical	Área	Não	56.2500	1.1458e-04	50.00
NGCUT09	Vertical	Perímetro	Sim	84.2500	7.5579e-05	44.44
NGCUT09	Vertical	Perímetro	Não	49.2500	8.4734e-05	44.44
NGCUT09	Vertical	Altura	Sim	55.5000	7.8440e-05	33.33
NGCUT09	Vertical	Altura	Não	57.0000	9.5987e-05	50.00
NGCUT09	Vertical	Largura	Sim	83.7500	8.3828e-05	50.00
NGCUT09	Vertical	Largura	Não	46.7500	7.0763e-05	33.33
NGCUT09	Vertical	Id	Sim	55.5000	7.4673e-05	33.33
NGCUT09	Vertical	Id	Não	62.5000	9.5081e-05	50.00
NGCUT09	Horizontal	Área	Sim	76.7500	7.7724e-05	38.89
NGCUT09	Horizontal	Área	Não	57.0000	1.1282e-04	50.00
NGCUT09	Horizontal	Perímetro	Sim	79.0000	5.5695e-05	27.78
NGCUT09	Horizontal	Perímetro	Não	57.0000	1.1406e-04	50.00
NGCUT09	Horizontal	Altura	Sim	72.0000	6.9904e-05	38.89
NGCUT09	Horizontal	Altura	Não	74.0000	7.1669e-05	33.33
NGCUT09	Horizontal	Largura	Sim	78.5000	6.4087e-05	33.33
NGCUT09	Horizontal	Largura	Não	55.5000	1.0061e-04	44.44
NGCUT09	Horizontal	Id	Sim	60.7500	8.8310e-05	38.89
NGCUT09	Horizontal	Id	Não	66.2500	7.0858e-05	33.33
NGCUT09	Maior área	Área	Sim	55.5000	1.4648e-04	33.33
NGCUT09	Maior área	Área	Não	72.7500	2.1458e-04	55.56
NGCUT09	Maior área	Perímetro	Sim	84.2500	2.2631e-04	44.44
NGCUT09	Maior área	Perímetro	Não	72.7500	2.1644e-04	55.56
NGCUT09	Maior área	Altura	Sim	55.5000	1.4338e-04	33.33
NGCUT09	Maior área	Altura	Não	74.0000	1.3566e-04	33.33
NGCUT09	Maior área	Largura	Sim	78.5000	1.6065e-04	33.33
NGCUT09	Maior área	Largura	Não	46.7500	1.3752e-04	33.33
NGCUT09	Maior área	Id	Sim	55.5000	1.3971e-04	33.33
NGCUT09	Maior área	Id	Não	76.5000	2.0766e-04	50.00
NGCUT09	Sobrepostas	Área	Sim	76.7500	1.0049e-03	38.89
NGCUT09	Sobrepostas	Área	Não	72.7500	3.1530e-03	55.56
NGCUT09	Sobrepostas	Perímetro	Sim	84.2500	7.5445e-04	44.44
NGCUT09	Sobrepostas	Perímetro	Não	72.7500	3.9856e-03	55.56
NGCUT09	Sobrepostas	Altura	Sim	72.0000	1.2302e-03	38.89
NGCUT09	Sobrepostas	Altura	Não	80.5000	3.2405e-03	55.56
NGCUT09	Sobrepostas	Largura	Sim	83.7500	9.3069e-04	50.00
NGCUT09	Sobrepostas	Largura	Não	69.7500	3.8887e-03	44.44
NGCUT09	Sobrepostas	Id	Sim	76.7500	2.4679e-03	38.89
NGCUT09	Sobrepostas	Id	Não	71.5000	2.1033e-03	50.00

Fonte: feito pelo autor.

Tabela 46 – Resultados da instância NGCUT10.

Instância	Divisão	Ordenação	Decrescente	Qualidade %	Tempo (s)	Itens %
NGCUT10	Vertical	Área	Sim	91.0000	7.0810e-05	46.15
NGCUT10	Vertical	Área	Não	29.4444	3.4618e-05	30.77
NGCUT10	Vertical	Perímetro	Sim	91.0000	5.1451e-05	46.15
NGCUT10	Vertical	Perímetro	Não	29.4444	3.2997e-05	30.77
NGCUT10	Vertical	Altura	Sim	91.0000	4.1533e-05	46.15
NGCUT10	Vertical	Altura	Não	39.4444	3.5191e-05	30.77
NGCUT10	Vertical	Largura	Sim	74.0000	4.1723e-05	38.46
NGCUT10	Vertical	Largura	Não	29.4444	3.3760e-05	30.77
NGCUT10	Vertical	Id	Sim	90.1111	6.8665e-05	53.85
NGCUT10	Vertical	Id	Não	29.4444	3.3903e-05	30.77
NGCUT10	Horizontal	Área	Sim	87.6667	5.0211e-05	30.77
NGCUT10	Horizontal	Área	Não	90.1111	6.1941e-05	53.85
NGCUT10	Horizontal	Perímetro	Sim	87.6667	5.0497e-05	30.77
NGCUT10	Horizontal	Perímetro	Não	90.1111	6.3419e-05	53.85
NGCUT10	Horizontal	Altura	Sim	91.0000	4.0388e-05	46.15
NGCUT10	Horizontal	Altura	Não	79.8889	5.3978e-05	46.15
NGCUT10	Horizontal	Largura	Sim	74.0000	4.6778e-05	38.46
NGCUT10	Horizontal	Largura	Não	90.1111	6.5136e-05	53.85
NGCUT10	Horizontal	Id	Sim	80.6667	6.3944e-05	46.15
NGCUT10	Horizontal	Id	Não	83.4444	4.9305e-05	46.15
NGCUT10	Maior área	Área	Sim	87.6667	9.3460e-05	30.77
NGCUT10	Maior área	Área	Não	90.1111	1.2984e-04	53.85
NGCUT10	Maior área	Perímetro	Sim	87.6667	9.1219e-05	30.77
NGCUT10	Maior área	Perímetro	Não	90.1111	1.3857e-04	53.85
NGCUT10	Maior área	Altura	Sim	91.0000	9.6941e-05	46.15
NGCUT10	Maior área	Altura	Não	79.8889	1.1406e-04	46.15
NGCUT10	Maior área	Largura	Sim	74.0000	9.6846e-05	38.46
NGCUT10	Maior área	Largura	Não	90.1111	1.3857e-04	53.85
NGCUT10	Maior área	Id	Sim	80.6667	1.2398e-04	46.15
NGCUT10	Maior área	Id	Não	83.4444	1.0400e-04	46.15
NGCUT10	Sobrepostas	Área	Sim	87.6667	4.3998e-04	30.77
NGCUT10	Sobrepostas	Área	Não	90.1111	5.6734e-04	53.85
NGCUT10	Sobrepostas	Perímetro	Sim	87.6667	4.4494e-04	30.77
NGCUT10	Sobrepostas	Perímetro	Não	90.1111	5.6410e-04	53.85
NGCUT10	Sobrepostas	Altura	Sim	91.0000	3.9849e-04	46.15
NGCUT10	Sobrepostas	Altura	Não	79.8889	4.1452e-04	46.15
NGCUT10	Sobrepostas	Largura	Sim	74.0000	2.8892e-04	38.46
NGCUT10	Sobrepostas	Largura	Não	90.1111	5.4836e-04	53.85
NGCUT10	Sobrepostas	Id	Sim	80.6667	6.6071e-04	46.15
NGCUT10	Sobrepostas	Id	Não	83.4444	4.0836e-04	46.15

Fonte: feito pelo autor.

Tabela 47 – Resultados da instância NGCUT11.

Instância	Divisão	Ordenação	Decrescente	Qualidade %	Tempo (s)	Itens %
NGCUT11	Vertical	Área	Sim	75.4444	7.4768e-05	40.00
NGCUT11	Vertical	Área	Não	45.8889	8.9455e-05	53.33
NGCUT11	Vertical	Perímetro	Sim	79.7778	8.6832e-05	46.67
NGCUT11	Vertical	Perímetro	Não	46.0000	7.1716e-05	40.00
NGCUT11	Vertical	Altura	Sim	75.4444	7.7295e-05	40.00
NGCUT11	Vertical	Altura	Não	45.8889	8.4972e-05	53.33
NGCUT11	Vertical	Largura	Sim	78.4444	7.0381e-05	53.33
NGCUT11	Vertical	Largura	Não	31.8889	6.3896e-05	33.33
NGCUT11	Vertical	Id	Sim	50.5556	6.7711e-05	40.00
NGCUT11	Vertical	Id	Não	78.5556	8.7547e-05	53.33
NGCUT11	Horizontal	Área	Sim	69.1111	5.6887e-05	33.33
NGCUT11	Horizontal	Área	Não	44.6667	8.4257e-05	46.67
NGCUT11	Horizontal	Perímetro	Sim	78.4444	9.0074e-05	53.33
NGCUT11	Horizontal	Perímetro	Não	44.6667	8.3637e-05	46.67
NGCUT11	Horizontal	Altura	Sim	86.3333	7.1096e-05	46.67
NGCUT11	Horizontal	Altura	Não	60.7778	9.6369e-05	53.33
NGCUT11	Horizontal	Largura	Sim	78.4444	8.8835e-05	53.33
NGCUT11	Horizontal	Largura	Não	67.6667	9.6846e-05	66.67
NGCUT11	Horizontal	Id	Sim	63.3333	9.5987e-05	53.33
NGCUT11	Horizontal	Id	Não	84.7778	8.2684e-05	60.00
NGCUT11	Maior área	Área	Sim	61.3333	1.1830e-04	33.33
NGCUT11	Maior área	Área	Não	44.6667	1.4839e-04	46.67
NGCUT11	Maior área	Perímetro	Sim	64.4444	1.1682e-04	33.33
NGCUT11	Maior área	Perímetro	Não	44.6667	1.6580e-04	46.67
NGCUT11	Maior área	Altura	Sim	83.2222	1.3304e-04	40.00
NGCUT11	Maior área	Altura	Não	60.7778	1.7891e-04	53.33
NGCUT11	Maior área	Largura	Sim	78.4444	1.7652e-04	53.33
NGCUT11	Maior área	Largura	Não	67.6667	2.1253e-04	66.67
NGCUT11	Maior área	Id	Sim	58.6667	1.6160e-04	46.67
NGCUT11	Maior área	Id	Não	78.5556	1.7700e-04	53.33
NGCUT11	Sobrepostas	Área	Sim	69.1111	4.7445e-04	33.33
NGCUT11	Sobrepostas	Área	Não	63.3333	1.4092e-03	53.33
NGCUT11	Sobrepostas	Perímetro	Sim	92.5556	1.2433e-03	60.00
NGCUT11	Sobrepostas	Perímetro	Não	63.3333	1.4242e-03	53.33
NGCUT11	Sobrepostas	Altura	Sim	87.6667	3.1859e-03	46.67
NGCUT11	Sobrepostas	Altura	Não	60.7778	2.6043e-03	53.33
NGCUT11	Sobrepostas	Largura	Sim	78.4444	1.6501e-03	53.33
NGCUT11	Sobrepostas	Largura	Não	67.6667	2.6953e-03	66.67
NGCUT11	Sobrepostas	Id	Sim	63.3333	1.6173e-03	53.33
NGCUT11	Sobrepostas	Id	Não	84.7778	2.3355e-03	60.00

Fonte: feito pelo autor.

Tabela 48 – Resultados da instância NGCUT12.

Instância	Divisão	Ordenação	Decrescente	Qualidade %	Tempo (s)	Itens %
NGCUT12	Vertical	Área	Sim	74.8889	8.0109e-05	31.82
NGCUT12	Vertical	Área	Não	48.6667	1.0037e-04	40.91
NGCUT12	Vertical	Perímetro	Sim	95.0000	9.6750e-05	40.91
NGCUT12	Vertical	Perímetro	Não	52.0000	8.2874e-05	31.82
NGCUT12	Vertical	Altura	Sim	84.3333	8.4114e-05	40.91
NGCUT12	Vertical	Altura	Não	48.6667	9.6083e-05	40.91
NGCUT12	Vertical	Largura	Sim	82.5556	9.7561e-05	45.45
NGCUT12	Vertical	Largura	Não	65.6667	7.2622e-05	36.36
NGCUT12	Vertical	Id	Sim	71.1111	9.0456e-05	36.36
NGCUT12	Vertical	Id	Não	97.2222	9.8467e-05	40.91
NGCUT12	Horizontal	Área	Sim	68.4444	8.7738e-05	36.36
NGCUT12	Horizontal	Área	Não	53.3333	1.3523e-04	45.45
NGCUT12	Horizontal	Perímetro	Sim	97.6667	9.4080e-05	45.45
NGCUT12	Horizontal	Perímetro	Não	53.3333	1.3723e-04	45.45
NGCUT12	Horizontal	Altura	Sim	84.3333	7.4720e-05	40.91
NGCUT12	Horizontal	Altura	Não	53.3333	1.3557e-04	45.45
NGCUT12	Horizontal	Largura	Sim	54.4444	1.2069e-04	40.91
NGCUT12	Horizontal	Largura	Não	83.4444	9.7513e-05	40.91
NGCUT12	Horizontal	Id	Sim	65.7778	1.2121e-04	45.45
NGCUT12	Horizontal	Id	Não	53.3333	1.3099e-04	45.45
NGCUT12	Maior área	Área	Sim	57.7778	1.6332e-04	31.82
NGCUT12	Maior área	Área	Não	53.3333	2.4614e-04	45.45
NGCUT12	Maior área	Perímetro	Sim	95.0000	1.9264e-04	40.91
NGCUT12	Maior área	Perímetro	Não	53.3333	2.2607e-04	45.45
NGCUT12	Maior área	Altura	Sim	84.3333	1.7633e-04	40.91
NGCUT12	Maior área	Altura	Não	53.3333	2.4123e-04	45.45
NGCUT12	Maior área	Largura	Sim	54.4444	2.0995e-04	40.91
NGCUT12	Maior área	Largura	Não	83.4444	1.9565e-04	40.91
NGCUT12	Maior área	Id	Sim	57.7778	1.7643e-04	31.82
NGCUT12	Maior área	Id	Não	53.3333	2.3503e-04	45.45
NGCUT12	Sobrepostas	Área	Sim	85.5556	8.3594e-04	36.36
NGCUT12	Sobrepostas	Área	Não	53.3333	5.5576e-03	45.45
NGCUT12	Sobrepostas	Perímetro	Sim	95.0000	1.4291e-03	40.91
NGCUT12	Sobrepostas	Perímetro	Não	53.3333	4.6591e-03	45.45
NGCUT12	Sobrepostas	Altura	Sim	84.3333	9.3241e-04	40.91
NGCUT12	Sobrepostas	Altura	Não	53.3333	8.7070e-03	45.45
NGCUT12	Sobrepostas	Largura	Sim	65.4444	3.5925e-03	45.45
NGCUT12	Sobrepostas	Largura	Não	65.6667	5.4835e-03	36.36
NGCUT12	Sobrepostas	Id	Sim	71.1111	1.2593e-03	36.36
NGCUT12	Sobrepostas	Id	Não	53.3333	3.1496e-03	45.45

Fonte: feito pelo autor.

A.4 OF

Tabela 49 – Resultados da instância OF1.

Instância	Divisão	Ordenação	Decrescente	Qualidade %	Tempo (s)	Itens %
OF1	Vertical	Área	Sim	71.0000	8.6975e-05	26.09
OF1	Vertical	Área	Não	56.6071	8.9169e-05	34.78
OF1	Vertical	Perímetro	Sim	84.8571	9.2506e-05	30.43
OF1	Vertical	Perímetro	Não	56.6071	9.0074e-05	34.78
OF1	Vertical	Altura	Sim	95.0000	1.0800e-04	34.78
OF1	Vertical	Altura	Não	74.7500	8.8119e-05	39.13
OF1	Vertical	Largura	Sim	75.5714	1.0376e-04	34.78
OF1	Vertical	Largura	Não	85.8571	1.1039e-04	34.78
OF1	Vertical	Id	Sim	74.5714	1.1315e-04	39.13
OF1	Vertical	Id	Não	78.7500	1.0476e-04	34.78
OF1	Horizontal	Área	Sim	80.8929	7.7105e-05	30.43
OF1	Horizontal	Área	Não	56.2143	9.5701e-05	30.43
OF1	Horizontal	Perímetro	Sim	77.7857	6.6757e-05	21.74
OF1	Horizontal	Perímetro	Não	70.7143	1.0166e-04	34.78
OF1	Horizontal	Altura	Sim	90.4286	8.5115e-05	34.78
OF1	Horizontal	Altura	Não	58.1429	8.5878e-05	26.09
OF1	Horizontal	Largura	Sim	60.2500	8.8167e-05	26.09
OF1	Horizontal	Largura	Não	85.8571	9.2554e-05	34.78
OF1	Horizontal	Id	Sim	69.1071	9.5463e-05	34.78
OF1	Horizontal	Id	Não	60.2500	8.0633e-05	26.09
OF1	Maior área	Área	Sim	75.7143	1.4725e-04	26.09
OF1	Maior área	Área	Não	56.6071	2.0409e-04	34.78
OF1	Maior área	Perímetro	Sim	83.1786	1.6336e-04	30.43
OF1	Maior área	Perímetro	Não	56.6071	1.8516e-04	34.78
OF1	Maior área	Altura	Sim	95.0000	1.9503e-04	34.78
OF1	Maior área	Altura	Não	64.4286	1.7972e-04	30.43
OF1	Maior área	Largura	Sim	74.7500	1.7033e-04	30.43
OF1	Maior área	Largura	Não	85.8571	2.0514e-04	34.78
OF1	Maior área	Id	Sim	51.4286	1.7285e-04	30.43
OF1	Maior área	Id	Não	69.1786	2.8148e-04	34.78
OF1	Sobrepostas	Área	Sim	80.8929	1.2308e-03	30.43
OF1	Sobrepostas	Área	Não	56.6071	3.8766e-03	34.78
OF1	Sobrepostas	Perímetro	Sim	84.8571	1.2110e-03	30.43
OF1	Sobrepostas	Perímetro	Não	56.6071	5.4671e-03	34.78
OF1	Sobrepostas	Altura	Sim	95.0000	1.4474e-03	34.78
OF1	Sobrepostas	Altura	Não	64.4286	3.5151e-03	30.43
OF1	Sobrepostas	Largura	Sim	81.0357	1.7475e-03	34.78
OF1	Sobrepostas	Largura	Não	89.0000	1.1739e-03	39.13
OF1	Sobrepostas	Id	Sim	74.5714	2.5227e-03	39.13
OF1	Sobrepostas	Id	Não	78.7500	1.7480e-03	34.78

Fonte: feito pelo autor.

Tabela 50 – Resultados da instância OF2.

Instância	Divisão	Ordenação	Decrescente	Qualidade %	Tempo (s)	Itens %
OF2	Vertical	Área	Sim	81.2857	6.4993e-05	25.00
OF2	Vertical	Área	Não	62.8214	9.1505e-05	33.33
OF2	Vertical	Perímetro	Sim	75.8214	7.3194e-05	25.00
OF2	Vertical	Perímetro	Não	76.0714	1.0028e-04	37.50
OF2	Vertical	Altura	Sim	63.3929	7.1621e-05	20.83
OF2	Vertical	Altura	Não	78.5714	1.1072e-04	37.50
OF2	Vertical	Largura	Sim	81.8214	7.7963e-05	25.00
OF2	Vertical	Largura	Não	54.7500	8.3923e-05	25.00
OF2	Vertical	Id	Sim	84.0714	8.1635e-05	29.17
OF2	Vertical	Id	Não	82.4643	9.0170e-05	29.17
OF2	Horizontal	Área	Sim	89.1429	7.9203e-05	33.33
OF2	Horizontal	Área	Não	50.2857	9.6989e-05	29.17
OF2	Horizontal	Perímetro	Sim	77.1071	9.0647e-05	29.17
OF2	Horizontal	Perímetro	Não	50.2857	9.6893e-05	29.17
OF2	Horizontal	Altura	Sim	81.3571	7.8201e-05	29.17
OF2	Horizontal	Altura	Não	35.4286	8.1396e-05	20.83
OF2	Horizontal	Largura	Sim	77.1071	9.3556e-05	29.17
OF2	Horizontal	Largura	Não	79.6071	1.0185e-04	37.50
OF2	Horizontal	Id	Sim	58.2857	6.4611e-05	20.83
OF2	Horizontal	Id	Não	82.7143	8.1825e-05	29.17
OF2	Maior área	Área	Sim	83.0714	1.6208e-04	29.17
OF2	Maior área	Área	Não	50.2857	1.7462e-04	29.17
OF2	Maior área	Perímetro	Sim	69.3214	1.4710e-04	25.00
OF2	Maior área	Perímetro	Não	50.2857	1.7419e-04	29.17
OF2	Maior área	Altura	Sim	63.3929	1.3247e-04	20.83
OF2	Maior área	Altura	Não	49.7143	1.5678e-04	25.00
OF2	Maior área	Largura	Sim	77.1071	1.7433e-04	29.17
OF2	Maior área	Largura	Não	54.7500	1.5483e-04	25.00
OF2	Maior área	Id	Sim	71.1786	1.5116e-04	25.00
OF2	Maior área	Id	Não	82.4643	1.6122e-04	29.17
OF2	Sobrepostas	Área	Sim	81.2857	1.4952e-03	25.00
OF2	Sobrepostas	Área	Não	75.3571	5.2221e-03	37.50
OF2	Sobrepostas	Perímetro	Sim	81.8214	1.8965e-03	25.00
OF2	Sobrepostas	Perímetro	Não	63.1786	4.9437e-03	33.33
OF2	Sobrepostas	Altura	Sim	73.9643	1.9861e-03	29.17
OF2	Sobrepostas	Altura	Não	64.5714	5.2644e-03	33.33
OF2	Sobrepostas	Largura	Sim	77.1071	2.9649e-03	29.17
OF2	Sobrepostas	Largura	Não	71.3214	2.4528e-03	33.33
OF2	Sobrepostas	Id	Sim	92.3571	1.4439e-03	33.33
OF2	Sobrepostas	Id	Não	82.4643	1.8828e-03	29.17

Fonte: feito pelo autor.

A.5 OKP

Tabela 51 – Resultados da instância OKP1.

Instância	Divisão	Ordenação	Decrescente	Qualidade %	Tempo (s)	Itens %
OKP1	Vertical	Área	Sim	77.5600	1.4401e-04	16
OKP1	Vertical	Área	Não	65.7400	4.0970e-04	48
OKP1	Vertical	Perímetro	Sim	91.1600	1.5883e-04	34
OKP1	Vertical	Perímetro	Não	59.8600	3.9644e-04	46
OKP1	Vertical	Altura	Sim	79.7000	3.6492e-04	38
OKP1	Vertical	Altura	Não	91.3800	2.0909e-04	42
OKP1	Vertical	Largura	Sim	91.6200	1.5159e-04	32
OKP1	Vertical	Largura	Não	65.7400	4.2324e-04	48
OKP1	Vertical	Id	Sim	86.0400	1.8191e-04	28
OKP1	Vertical	Id	Não	82.1200	4.0898e-04	46
OKP1	Horizontal	Área	Sim	93.1000	1.3990e-04	22
OKP1	Horizontal	Área	Não	58.6600	2.5072e-04	42
OKP1	Horizontal	Perímetro	Sim	83.7800	1.0734e-04	26
OKP1	Horizontal	Perímetro	Não	57.9200	2.3813e-04	38
OKP1	Horizontal	Altura	Sim	83.7000	2.6159e-04	46
OKP1	Horizontal	Altura	Não	69.3200	1.1864e-04	26
OKP1	Horizontal	Largura	Sim	83.7800	1.1826e-04	26
OKP1	Horizontal	Largura	Não	61.0600	2.8372e-04	48
OKP1	Horizontal	Id	Sim	81.5200	1.7333e-04	26
OKP1	Horizontal	Id	Não	68.5600	1.9655e-04	34
OKP1	Maior área	Área	Sim	77.5600	2.3112e-04	16
OKP1	Maior área	Área	Não	67.0600	7.0515e-04	50
OKP1	Maior área	Perímetro	Sim	88.4000	2.5392e-04	28
OKP1	Maior área	Perímetro	Não	67.1600	5.0588e-04	34
OKP1	Maior área	Altura	Sim	79.7000	5.6319e-04	38
OKP1	Maior área	Altura	Não	78.5600	2.9588e-04	30
OKP1	Maior área	Largura	Sim	88.4000	3.7198e-04	28
OKP1	Maior área	Largura	Não	65.7400	6.8054e-04	48
OKP1	Maior área	Id	Sim	77.5200	2.5144e-04	18
OKP1	Maior área	Id	Não	62.6600	5.7583e-04	40
OKP1	Sobrepostas	Área	Sim	97.2200	1.7846e-03	18
OKP1	Sobrepostas	Área	Não	67.0600	2.1143e-02	50
OKP1	Sobrepostas	Perímetro	Sim	91.1600	4.0567e-03	34
OKP1	Sobrepostas	Perímetro	Não	65.9200	2.2325e-02	50
OKP1	Sobrepostas	Altura	Sim	88.5400	8.4483e-03	42
OKP1	Sobrepostas	Altura	Não	91.3800	6.6834e-03	42
OKP1	Sobrepostas	Largura	Sim	91.6200	2.9796e-03	32
OKP1	Sobrepostas	Largura	Não	65.7400	2.2890e-02	48
OKP1	Sobrepostas	Id	Sim	90.0400	9.4479e-03	36
OKP1	Sobrepostas	Id	Não	92.1200	1.8555e-02	50

Fonte: feito pelo autor.

Tabela 52 – Resultados da instância OKP2.

Instância	Divisão	Ordenação	Decrescente	Qualidade %	Tempo (s)	Itens %
OKP2	Vertical	Área	Sim	84.6200	1.6479e-04	20.00
OKP2	Vertical	Área	Não	57.0200	1.9617e-04	40.00
OKP2	Vertical	Perímetro	Sim	86.8200	1.2054e-04	33.33
OKP2	Vertical	Perímetro	Não	67.7600	2.0738e-04	43.33
OKP2	Vertical	Altura	Sim	86.8200	1.3542e-04	30.00
OKP2	Vertical	Altura	Não	77.0200	1.7843e-04	40.00
OKP2	Vertical	Largura	Sim	87.6900	1.2140e-04	33.33
OKP2	Vertical	Largura	Não	68.2400	1.9579e-04	40.00
OKP2	Vertical	Id	Sim	83.6100	1.5059e-04	33.33
OKP2	Vertical	Id	Não	70.4200	1.6351e-04	36.67
OKP2	Horizontal	Área	Sim	84.5800	1.0796e-04	26.67
OKP2	Horizontal	Área	Não	51.7000	1.7323e-04	33.33
OKP2	Horizontal	Perímetro	Sim	84.5900	1.0056e-04	30.00
OKP2	Horizontal	Perímetro	Não	41.1800	1.0877e-04	23.33
OKP2	Horizontal	Altura	Sim	86.8200	1.3824e-04	30.00
OKP2	Horizontal	Altura	Não	55.6200	9.5940e-05	23.33
OKP2	Horizontal	Largura	Sim	84.9500	1.0800e-04	30.00
OKP2	Horizontal	Largura	Não	51.7000	1.5526e-04	33.33
OKP2	Horizontal	Id	Sim	68.0900	1.2951e-04	26.67
OKP2	Horizontal	Id	Não	63.0400	1.0557e-04	23.33
OKP2	Maior área	Área	Sim	83.6100	1.8091e-04	23.33
OKP2	Maior área	Área	Não	61.2800	3.2644e-04	40.00
OKP2	Maior área	Perímetro	Sim	90.0500	1.9298e-04	30.00
OKP2	Maior área	Perímetro	Não	60.4800	2.6722e-04	33.33
OKP2	Maior área	Altura	Sim	86.8200	2.3332e-04	30.00
OKP2	Maior área	Altura	Não	54.6200	2.1100e-04	26.67
OKP2	Maior área	Largura	Sim	84.9500	2.0037e-04	30.00
OKP2	Maior área	Largura	Não	68.2400	3.2435e-04	40.00
OKP2	Maior área	Id	Sim	83.6100	2.5225e-04	33.33
OKP2	Maior área	Id	Não	74.7100	2.8038e-04	36.67
OKP2	Sobrepostas	Área	Sim	84.6200	1.2405e-03	20.00
OKP2	Sobrepostas	Área	Não	57.1300	4.1312e-03	33.33
OKP2	Sobrepostas	Perímetro	Sim	90.0500	1.4324e-03	30.00
OKP2	Sobrepostas	Perímetro	Não	61.3200	5.4472e-03	33.33
OKP2	Sobrepostas	Altura	Sim	94.8000	1.2096e-03	33.33
OKP2	Sobrepostas	Altura	Não	66.1800	4.1755e-03	30.00
OKP2	Sobrepostas	Largura	Sim	84.9500	1.6931e-03	30.00
OKP2	Sobrepostas	Largura	Não	69.8900	5.8948e-03	46.67
OKP2	Sobrepostas	Id	Sim	78.3100	2.8646e-03	26.67
OKP2	Sobrepostas	Id	Não	78.9400	2.8905e-03	43.33

Fonte: feito pelo autor.

Tabela 53 – Resultados da instância OKP3.

Instância	Divisão	Ordenação	Decrescente	Qualidade %	Tempo (s)	Itens %
OKP3	Vertical	Área	Sim	93.6400	8.4019e-05	20.00
OKP3	Vertical	Área	Não	73.2100	1.8239e-04	40.00
OKP3	Vertical	Perímetro	Sim	92.6200	1.2507e-04	26.67
OKP3	Vertical	Perímetro	Não	59.0100	1.5645e-04	33.33
OKP3	Vertical	Altura	Sim	87.8400	1.4491e-04	30.00
OKP3	Vertical	Altura	Não	80.6400	1.6079e-04	40.00
OKP3	Vertical	Largura	Sim	78.7800	1.0500e-04	26.67
OKP3	Vertical	Largura	Não	58.5900	1.5717e-04	33.33
OKP3	Vertical	Id	Sim	87.0400	1.6317e-04	36.67
OKP3	Vertical	Id	Não	68.1300	1.2527e-04	26.67
OKP3	Horizontal	Área	Sim	87.7100	1.0405e-04	23.33
OKP3	Horizontal	Área	Não	40.8200	9.4318e-05	23.33
OKP3	Horizontal	Perímetro	Sim	82.1200	8.8882e-05	20.00
OKP3	Horizontal	Perímetro	Não	40.3200	1.0824e-04	23.33
OKP3	Horizontal	Altura	Sim	89.4000	1.7700e-04	33.33
OKP3	Horizontal	Altura	Não	40.8200	9.3031e-05	23.33
OKP3	Horizontal	Largura	Sim	84.1200	9.0694e-05	23.33
OKP3	Horizontal	Largura	Não	41.8600	1.3518e-04	26.67
OKP3	Horizontal	Id	Sim	55.7400	1.0343e-04	20.00
OKP3	Horizontal	Id	Não	47.0000	1.1153e-04	23.33
OKP3	Maior área	Área	Sim	91.8500	1.6670e-04	20.00
OKP3	Maior área	Área	Não	49.9600	2.0771e-04	26.67
OKP3	Maior área	Perímetro	Sim	92.6200	2.1229e-04	26.67
OKP3	Maior área	Perímetro	Não	51.9000	1.8759e-04	23.33
OKP3	Maior área	Altura	Sim	89.4000	2.7313e-04	33.33
OKP3	Maior área	Altura	Não	56.4600	2.4962e-04	30.00
OKP3	Maior área	Largura	Sim	85.6800	1.8134e-04	26.67
OKP3	Maior área	Largura	Não	70.7500	2.9602e-04	36.67
OKP3	Maior área	Id	Sim	68.4800	1.8425e-04	23.33
OKP3	Maior área	Id	Não	75.8800	2.0981e-04	26.67
OKP3	Sobrepostas	Área	Sim	93.6400	9.4986e-04	20.00
OKP3	Sobrepostas	Área	Não	40.3200	4.3970e-03	23.33
OKP3	Sobrepostas	Perímetro	Sim	92.6200	9.1491e-04	26.67
OKP3	Sobrepostas	Perímetro	Não	51.9000	3.6557e-03	23.33
OKP3	Sobrepostas	Altura	Sim	89.4000	1.5201e-03	33.33
OKP3	Sobrepostas	Altura	Não	71.5100	7.0192e-03	30.00
OKP3	Sobrepostas	Largura	Sim	85.6800	1.8664e-03	26.67
OKP3	Sobrepostas	Largura	Não	73.9700	3.2407e-03	40.00
OKP3	Sobrepostas	Id	Sim	87.0400	2.7803e-03	36.67
OKP3	Sobrepostas	Id	Não	67.0000	2.5690e-03	30.00

Fonte: feito pelo autor.

Tabela 54 – Resultados da instância OKP4.

Instância	Divisão	Ordenação	Decrescente	Qualidade %	Tempo (s)	Itens %
OKP4	Vertical	Área	Sim	93.8900	1.0943e-04	13.11
OKP4	Vertical	Área	Não	73.7700	4.0960e-04	37.70
OKP4	Vertical	Perímetro	Sim	90.9900	2.0833e-04	19.67
OKP4	Vertical	Perímetro	Não	71.0600	3.8357e-04	34.43
OKP4	Vertical	Altura	Sim	94.7200	2.7885e-04	24.59
OKP4	Vertical	Altura	Não	77.9900	2.1319e-04	26.23
OKP4	Vertical	Largura	Sim	86.9500	2.0637e-04	27.87
OKP4	Vertical	Largura	Não	76.8500	4.3707e-04	37.70
OKP4	Vertical	Id	Sim	91.8600	2.5077e-04	22.95
OKP4	Vertical	Id	Não	77.8500	2.9202e-04	26.23
OKP4	Horizontal	Área	Sim	66.1400	9.9468e-05	9.84
OKP4	Horizontal	Área	Não	67.5000	2.4753e-04	26.23
OKP4	Horizontal	Perímetro	Sim	89.5400	1.1768e-04	13.11
OKP4	Horizontal	Perímetro	Não	69.5200	2.3346e-04	24.59
OKP4	Horizontal	Altura	Sim	93.4600	1.8148e-04	19.67
OKP4	Horizontal	Altura	Não	68.3500	2.2850e-04	22.95
OKP4	Horizontal	Largura	Sim	82.1500	2.3079e-04	24.59
OKP4	Horizontal	Largura	Não	49.8900	2.2511e-04	26.23
OKP4	Horizontal	Id	Sim	71.5600	1.8673e-04	21.31
OKP4	Horizontal	Id	Não	82.3700	1.6150e-04	21.31
OKP4	Maior área	Área	Sim	66.1400	1.5831e-04	9.84
OKP4	Maior área	Área	Não	73.7700	6.7039e-04	37.70
OKP4	Maior área	Perímetro	Sim	90.9900	3.4185e-04	19.67
OKP4	Maior área	Perímetro	Não	68.1500	5.1789e-04	29.51
OKP4	Maior área	Altura	Sim	94.7200	4.4775e-04	24.59
OKP4	Maior área	Altura	Não	70.5500	3.6664e-04	24.59
OKP4	Maior área	Largura	Sim	82.1500	3.5472e-04	24.59
OKP4	Maior área	Largura	Não	76.8500	7.3724e-04	37.70
OKP4	Maior área	Id	Sim	88.6100	3.6578e-04	19.67
OKP4	Maior área	Id	Não	78.4100	3.2077e-04	19.67
OKP4	Sobrepostas	Área	Sim	93.8900	1.6792e-03	13.11
OKP4	Sobrepostas	Área	Não	73.7700	2.6092e-02	37.70
OKP4	Sobrepostas	Perímetro	Sim	90.9900	2.0824e-03	19.67
OKP4	Sobrepostas	Perímetro	Não	70.5700	2.3835e-02	29.51
OKP4	Sobrepostas	Altura	Sim	94.7200	2.7651e-03	24.59
OKP4	Sobrepostas	Altura	Não	82.1500	1.9110e-02	24.59
OKP4	Sobrepostas	Largura	Sim	88.3100	4.5036e-03	26.23
OKP4	Sobrepostas	Largura	Não	78.8500	1.2042e-02	39.34
OKP4	Sobrepostas	Id	Sim	91.8600	3.8030e-03	22.95
OKP4	Sobrepostas	Id	Não	77.8500	5.0374e-03	26.23

Fonte: feito pelo autor.

Tabela 55 – Resultados da instância OKP5.

Instância	Divisão	Ordenação	Decrescente	Qualidade %	Tempo (s)	Itens %
OKP5	Vertical	Área	Sim	84.8600	1.7028e-04	10.31
OKP5	Vertical	Área	Não	58.7400	4.7812e-04	22.68
OKP5	Vertical	Perímetro	Sim	96.2700	1.4076e-04	12.37
OKP5	Vertical	Perímetro	Não	64.2300	2.9836e-04	18.56
OKP5	Vertical	Altura	Sim	92.4600	2.7361e-04	12.37
OKP5	Vertical	Altura	Não	83.7900	2.3808e-04	19.59
OKP5	Vertical	Largura	Sim	98.1700	1.5202e-04	17.53
OKP5	Vertical	Largura	Não	74.8600	5.2500e-04	24.74
OKP5	Vertical	Id	Sim	82.9200	3.7689e-04	19.59
OKP5	Vertical	Id	Não	76.4600	3.5596e-04	18.56
OKP5	Horizontal	Área	Sim	87.7900	3.3321e-04	13.40
OKP5	Horizontal	Área	Não	62.2000	4.2505e-04	21.65
OKP5	Horizontal	Perímetro	Sim	96.2700	1.5454e-04	12.37
OKP5	Horizontal	Perímetro	Não	75.9600	3.9129e-04	21.65
OKP5	Horizontal	Altura	Sim	92.4600	1.7500e-04	12.37
OKP5	Horizontal	Altura	Não	58.1900	3.8648e-04	15.46
OKP5	Horizontal	Largura	Sim	98.1700	2.2922e-04	17.53
OKP5	Horizontal	Largura	Não	70.2100	3.8805e-04	22.68
OKP5	Horizontal	Id	Sim	71.3100	2.9588e-04	16.49
OKP5	Horizontal	Id	Não	80.3000	2.5239e-04	17.53
OKP5	Maior área	Área	Sim	86.8200	3.2973e-04	11.34
OKP5	Maior área	Área	Não	58.7400	7.0772e-04	22.68
OKP5	Maior área	Perímetro	Sim	96.2700	2.5525e-04	12.37
OKP5	Maior área	Perímetro	Não	69.0600	4.6005e-04	17.53
OKP5	Maior área	Altura	Sim	92.4600	4.0922e-04	12.37
OKP5	Maior área	Altura	Não	79.1100	5.3983e-04	18.56
OKP5	Maior área	Largura	Sim	98.1700	4.1027e-04	17.53
OKP5	Maior área	Largura	Não	74.8600	8.0080e-04	24.74
OKP5	Maior área	Id	Sim	77.0600	6.1326e-04	20.62
OKP5	Maior área	Id	Não	66.8600	5.2366e-04	16.49
OKP5	Sobrepostas	Área	Sim	84.8600	4.7212e-03	10.31
OKP5	Sobrepostas	Área	Não	83.7400	1.0123e-01	27.84
OKP5	Sobrepostas	Perímetro	Sim	96.2700	1.8673e-03	12.37
OKP5	Sobrepostas	Perímetro	Não	72.2400	8.4768e-02	19.59
OKP5	Sobrepostas	Altura	Sim	92.4600	1.9705e-03	12.37
OKP5	Sobrepostas	Altura	Não	74.1500	7.0392e-02	17.53
OKP5	Sobrepostas	Largura	Sim	98.1700	4.5398e-03	17.53
OKP5	Sobrepostas	Largura	Não	74.8600	8.9579e-02	24.74
OKP5	Sobrepostas	Id	Sim	82.9200	1.4897e-02	19.59
OKP5	Sobrepostas	Id	Não	89.9000	2.7700e-02	19.59

Fonte: feito pelo autor.

APÊNDICE B – Código fonte

Código fonte disponível em https://github.com/G-Carneiro/packing-problem/.

B.1 LICENSE

```
MIT License

Copyright (c) 2022 Gabriel Carneiro

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all
copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE
SOFTWARE.
```

B.2 main.py

```python
from os import listdir
from os.path import basename, dirname
from time import time

from numpy import mean, median, std

from src.model.item import Item
```

```
8  from src.model.model import Model
9  from src.utils.folders import *
10 from src.utils.functions import *
11 from src.utils.order_key import OrderKey
12 from src.utils.split_mode import SplitMode
13
14
15 def main(folder: str = INSTANCES) -> None:
16     num_tests: int = 5
17     total_time = 0
18     folder_name = basename(dirname(f"{folder}/"))
19     for file_name in [sorted(listdir(folder))[-1]]:
20         file = f"{folder}/{file_name}"
21         file_name = file_name.split(".")[0]
22         with open(file, "r") as f:
23             lines = f.readlines()
24         items: list[Item] = []
25         for line in lines[2:]:
26             _, width, height, demand, copies, profit = line.split()
27             for _ in range(int(copies)):
28                 items.append(Item(width=float(width),
                    ↪ height=float(height)))
29         width, height = lines[1].split()
30         box = Item(width=float(width), height=float(height))
31         for split in SplitMode:
32             if split == SplitMode.NONE:
33                 break
34             for order in OrderKey:
35                 for descending in [True, False]:
36                     exec_time = []
37                     model = Model(name=file_name, instance=folder_name,
                        ↪ box=box, items=items,
38                                   order=order, split=split,
                                      ↪ decrescent=descending)
39                     for _ in range(num_tests):
40                         print(f"[Starting] file={file_name}
                            ↪ split={split.name} order={order.name} "
41                               f"decrescent={descending}")
42                         model.reset()
```

```
43                          start = time()
44                          model.solve()
45                          exec_time.append(time() - start)
46                          total_time += exec_time[-1]
47                          print(f"[Finished] file={file_name}
                            ↪  split={split.name} order={order.name} "
48                                f"decrescent={descending}
                                  ↪  exec={exec_time[-1]}
                                  ↪  total={total_time}")
49
50                      media = mean(exec_time)
51                      mediana = median(exec_time)
52                      desvio = std(exec_time)
53                      model.to_csv(csv_folder=CSV, exec_time=media,
                        ↪  median=mediana, std=desvio)
54                      model.export_model(folder=FIGURES,
                        ↪  show_regions=False,
55                                         show_labels=False)
56
57      csv_to_table(csv_folder=CSV, table_folder=DATA)
58      return None
59
60
61  # TODO: run bkw13, with order=NONE
62  if __name__ == "__main__":
63      main()
```

B.3 src/

B.3.1 model/

B.3.1.1 coordinate.py

```
1  from __future__ import annotations
2
3
4  class Coordinate:
5      def __init__(self, x: float, y: float):
6          self.x: float = x
7          self.y: float = y
```

```
8
9       def __lt__(self, other: Coordinate) -> bool:
10          if (self.x != other.x):
11              return (self.x < other.x)
12
13          return (self.y < other.y)
14
15      def __eq__(self, other: Coordinate | tuple[float, float]) -> bool:
16          if isinstance(other, tuple):
17              other = Coordinate(other[0], other[1])
18
19          return (self.x == other.x and self.y == other.y)
20
21      def __le__(self, other: Coordinate) -> bool:
22          return (self < other or self == other)
23
24      def __add__(self, other: Coordinate | tuple[float, float]) ->
    ↪   Coordinate:
25          if isinstance(other, tuple):
26              other = Coordinate(other[0], other[1])
27
28          return Coordinate(self.x + other.x, self.y + other.y)
29
30      def __repr__(self) -> str:
31          return f"({self.x}, {self.y})"
```

B.3.1.2 item.py

```
1  from __future__ import annotations
2
3  from .coordinate import Coordinate
4  from .rect import Rect
5
6
7  class Item(Rect):
8      id_: int = 0
9
10     def __init__(self,
11                  width: float = float("inf"),
12                  height: float = float("inf"),
```

```
13                  ) -> None:
14          super().__init__(width=width, height=height)
15          self._id: int = Item.id_
16          Item.id_ += 1
17
18      @property
19      def id(self) -> int:
20          return self._id
21
22      def coords(self) -> tuple[Coordinate, Coordinate, Coordinate,
        ↪  Coordinate]:
23          return (self.position, self.position + (0, self.height),
24                  self.position + (self.width, self.height), self.position
                    ↪  + (self.width, 0))
25
26      def conflict(self, other: Item) -> bool:
27          # adaptation of cohen-sutherland clipping algorithm
28          def region_code(point: Coordinate) -> int:
29              code: str = ""
30              code += f"{int(point.y >= self.position.y + self.height)}"
                ↪  # 1000 top
31              code += f"{int(point.y <= self.position.y)}"  # 0100 bottom
32              code += f"{int(point.x >= self.position.x + self.width)}"  #
                ↪  0010 right
33              code += f"{int(point.x <= self.position.x)}"  # 0001 left
34              return int(code, 2)
35
36          if (other.position is None) or (self.position is None):
37              return False
38          elif (self.position == other.position) or (self.end ==
            ↪  other.end):
39              return True
40
41          coords = other.coords()
42          for idx, coord in enumerate(coords):
43              if region_code(point=coord) & region_code(coords[idx - 1])
                ↪  == 0:
44                  return True
45          return False
```

B.3.1.3 `model.py`

```
from os import makedirs, path

from matplotlib import pyplot as plt
from matplotlib.patches import Rectangle
from pandas import DataFrame

from .coordinate import Coordinate
from .item import Item
from .order_mode import OrderMode
from .ordered_queue import OrderedQueue
from .region import NotRegion, Region
from ..utils.order_key import OrderKey
from ..utils.split_mode import SplitMode


class Model:
    def __init__(self,
                 name: str,
                 instance: str,
                 box: Item,
                 items: list[Item],
                 order: OrderKey,
                 split: SplitMode,
                 decrescent: bool = True
                 ) -> None:
        self._name: str = name
        self._instance: str = instance
        self._box: Item = box
        self._items: list[Item] = items
        self._order: OrderMode = OrderMode(key=order, reverse=decrescent)
        self._items: list[Item] = self.sorted_items(order=self.order)
        self._split: SplitMode = split
        self._export_id: int = 0
        self._regions: OrderedQueue[Region] = OrderedQueue([
            Region((0, 0), (box.width, box.height))])

    @property
    def name(self):
```

```
39          return self._name
40
41      @property
42      def box(self) -> Item:
43          return self._box
44
45      @property
46      def items(self) -> list[Item]:
47          return self._items
48
49      @property
50      def regions(self) -> OrderedQueue[Region]:
51          return self._regions
52
53      @property
54      def order(self) -> OrderMode:
55          return self._order
56
57      @property
58      def split(self) -> SplitMode:
59          return self._split
60
61      @property
62      def decrescent(self) -> bool:
63          return self.order.reverse
64
65      def sorted_items(self, order: OrderMode) -> list[Item]:
66          return sorted(self._items, key=lambda x:
            ↪  eval(f"x.{order.name.lower()}"),
67                        reverse=order.reverse)
68
69      def percent_free(self) -> float:
70          return (100 - self.percent_busy())
71
72      def percent_busy(self) -> float:
73          busy_area: float = 0
74          for item in self.items:
75              if item.position is not None:
76                  busy_area += item.area
```

```
77          return (busy_area / self.box.area * 100)
78
79      def solution_quality(self) -> float:
80          for item in self.items:
81              if item.position is None:
82                  break
83          else:
84              return 100
85          return self.percent_busy()
86
87      def inside_items(self) -> int:
88          num_items: int = 0
89          for item in self.items:
90              if item.position is not None:
91                  num_items += 1
92
93          return num_items
94
95      def outside_items(self) -> int:
96          return (len(self.items) - self.inside_items())
97
98      def inside_items_percent(self) -> float:
99          return (self.inside_items() / len(self.items) * 100)
100
101     def outside_item_percent(self) -> float:
102         return (self.outside_items() / len(self.items) * 100)
103
104     def replace_region(self, original: Region, split: tuple[Region,
        ↪ Region]) -> None:
105         self._regions.remove(original)
106         r0, r1 = split
107         if r0 is not None:
108             self._regions.append(r0)
109         if r1 is not None:
110             self._regions.append(r1)
111         return None
112
113     @staticmethod
114     def new_region(start: Coordinate | tuple[float, float],
```

```
115                     end: Coordinate | tuple[float, float]) -> Region |
                        ↪ None:
116         try:
117             region = Region(start=start, end=end)
118         except NotRegion:
119             return None
120
121         return region
122
123     def split_horizontally(self, region: Region, item: Item) ->
        ↪ tuple[Region, Region]:
124         start = (item.position.x, item.position.y + item.height)
125         end = region.end
126         region0 = self.new_region(start=start, end=end)
127         start = (item.position.x + item.width, item.position.y)
128         end = (region.end.x, item.position.y + item.height)
129         region1 = self.new_region(start=start, end=end)
130         return (region0, region1)
131
132     def split_vertically(self, region: Region, item: Item) ->
        ↪ tuple[Region, Region]:
133         start = (item.position.x, item.position.y + item.height)
134         end = (item.position.x + item.width, region.end.y)
135         region0 = self.new_region(start=start, end=end)
136         start = (item.position.x + item.width, item.position.y)
137         end = region.end
138         region1 = self.new_region(start=start, end=end)
139         return (region0, region1)
140
141     def fake_split(self, region: Region, item: Item) -> tuple[Region,
        ↪ Region]:
142         start = (item.position.x, item.position.y + item.height)
143         end = region.end
144         region0 = self.new_region(start=start, end=end)
145         start = (item.position.x + item.width, item.position.y)
146         region1 = self.new_region(start=start, end=end)
147         return (region0, region1)
148
```

```
149     def choose_best_split(self, region: Region, item: Item) ->
        ↪ tuple[Region, Region]:
150         split_h = self.split_horizontally(region=region, item=item)
151         split_v = self.split_vertically(region=region, item=item)
152         regions: list[Region] = [region for region in split_h + split_v
            ↪ if region is not None]
153         try:
154             biggest = max(regions, key=lambda x: x.area)
155         except ValueError:
156             return split_h
157         # FIXME: find best solution
158         Region.id_ -= len(regions)
159         if (biggest in split_h):
160             return self.split_horizontally(region=region, item=item)
161         return self.split_vertically(region=region, item=item)
162 
163     def solve(self, export: bool = False, export_all: bool = False,
164               show_regions: bool = False, show_labels: bool = False) ->
                  ↪ float:
165         if export_all:
166             self.export_model(folder=f"output/figures",
                ↪ show_regions=show_regions,
167                               show_labels=show_labels)
168 
169         for item in self.items:
170             for region in self._regions:
171                 if (item.width > region.width) or (item.height >
                    ↪ region.height):
172                     continue
173 
174                 item.position = region.start
175                 match self.split:
176                     case SplitMode.NONE:
177                         # FIXME: two equal items in same place is
                            ↪ posible, try use OrderedSet to
178                         #  solve
179                         for other in self.items:
180                             if (other == item):
181                                 break
```

```
182                             if other.conflict(other=item) or
                                ↪ item.conflict(other=other):
183                                 item.position = None
184                                 break
185
186                         if (item.position is None):
187                             continue
188                         split = self.fake_split(region=region, item=item)
189                     case SplitMode.HORIZONTALLY:
190                         split = self.split_horizontally(region=region,
                            ↪ item=item)
191                     case SplitMode.VERTICALLY:
192                         split = self.split_vertically(region=region,
                            ↪ item=item)
193                     case _:
194                         split = self.choose_best_split(region=region,
                            ↪ item=item)
195
196                 self.replace_region(original=region, split=split)
197                 if export_all:
198                     self.export_model(folder=f"output/figures",
                        ↪ show_regions=show_regions,
199                                       show_labels=show_labels)
200                 break
201         if export and not export_all:
202             self.export_model(folder=f"output/figures",
                ↪ show_regions=show_regions,
203                               show_labels=show_labels)
204         return self.solution_quality()
205
206     def reset(self, order: OrderMode = None, split: SplitMode = None) ->
        ↪ None:
207         if order is not None:
208             self._order = order
209         if split is not None:
210             self._split = split
211         self.box.position = None
212         self._export_id = 0
213         Region.id_ = 0
```

```
214         Item.id_ = 0
215         self._regions = OrderedQueue([Region((0, 0), (self.box.width,
            ↪ self.box.height))])
216         for item in self.items:
217             item.position = None
218         return None
219
220     def export_model(self, folder: str, show_regions: bool = False,
221                      show_labels: bool = False) -> None:
222         folder = f"{folder}/{self._instance}/{self._name}/{self.split.na
            ↪ me}/{self.order.name}/"
            ↪ \
223                  f"{self.decrescent}".lower()
224         makedirs(folder, exist_ok=True)
225         num: str = "0" * (len(str(len(self.items))) -
            ↪ len(str(self._export_id))) \
226                    + f"{self._export_id}"
227         file = f"{folder}/{num}.png"
228         self._export_id += 1
229         fig, ax = plt.subplots()
230         box = Rectangle(xy=(0, 0), width=self.box.width,
            ↪ height=self.box.height, alpha=0.1)
231         ax.add_patch(box)
232         item_cmap = plt.get_cmap('brg', len(self.items))
233         self._rectangle_cmap(iterable=self.items, cmap=item_cmap, ax=ax,
            ↪ show_labels=show_labels)
234         if show_regions:
235             region_cmap = plt.get_cmap('hsv', 2 * len(self.items) + 1)
236             self._rectangle_cmap(iterable=self.regions,
                ↪ cmap=region_cmap, ax=ax,
237                                  show_labels=show_labels)
238         plt.xlim(0, self.box.width)
239         plt.ylim(0, self.box.height)
240         plt.savefig(file)
241         plt.close()
242         return None
243
244     @staticmethod
```

```
245     def _rectangle_cmap(iterable, cmap, ax, show_labels: bool = False)
        ↪  -> None:
246         for idx, item in enumerate(iterable):
247             if item.position is None:
248                 continue
249             x = item.position.x
250             y = item.position.y
251             rect = Rectangle(xy=(x, y), width=item.width,
                ↪  height=item.height,
252                              facecolor=cmap(item.id), alpha=0.2,
                                ↪  linewidth=1, edgecolor='k')
253             ax.add_patch(rect)
254             if show_labels:
255                 cx = x + rect.get_width() / 2
256                 cy = y + rect.get_height() / 2
257                 if isinstance(item, Item):
258                     label = idx
259                 else:
260                     label = f"R{item.id}"
261                 ax.annotate(label, (cx, cy), color='black',
                    ↪  weight='bold', fontsize=10, ha='center',
262                             va='center')
263         return None
264
265     def to_csv(self, csv_folder: str, exec_time: float,
266                median: float, std: float) -> None:
267         data = {"Instance": [self._name],
268                 "SplitMode": [self.split.name],
269                 "OrderMode": [self.order.name],
270                 "Decrescent": [self.decrescent],
271                 "Solution Quality %": [self.solution_quality()],
272                 "Exec. Time": [exec_time],
273                 "Median": [median],
274                 "Standard deviation": [std],
275                 "Occupied %": [self.percent_busy()],
276                 "Free %": [self.percent_free()],
277                 "Nº Items": [len(self.items)],
278                 "Inside Items": [self.inside_items()],
279                 "Outside Items": [self.outside_items()],
```

```
280                 "Inside Items %": [self.inside_items_percent()],
281                 "Outside Items %": [self.outside_item_percent()]}
282         self._to_csv(csv_folder=csv_folder, data=data)
283         self._to_csv(csv_folder=f"{csv_folder}/{self._instance}",
            ↪ data=data)
284         self._to_csv(csv_folder=f"{csv_folder}/{self._instance}/{self.na
            ↪ me.lower()}",
            ↪ data=data)
285         return None
286
287     def _to_csv(self, csv_folder: str, data) -> None:
288         csv_folder = csv_folder.lower()
289         files = [f"order/{self.order.name}", f"split/{self.split.name}",
290                  f"decrescent/{self.decrescent}", "all"]
291
292         makedirs(f"{csv_folder}/order", exist_ok=True)
293         makedirs(f"{csv_folder}/split", exist_ok=True)
294         makedirs(f"{csv_folder}/decrescent", exist_ok=True)
295         for file in files:
296             file = f"{csv_folder}/{file}".lower()
297
298             with open(file, "a") as f:
299                 DataFrame(data).to_csv(f, index=False,
300                                        header=not
                                           ↪ bool(path.getsize(file)))
301
302         return None
```

B.3.1.4 order_mode.py

```
1  from ..utils.order_key import OrderKey
2
3
4  class OrderMode:
5      def __init__(self, key: OrderKey, reverse: bool = True) -> None:
6          self._key: OrderKey = key
7          self._reverse: bool = reverse
8
9      @property
10     def key(self) -> OrderKey:
```

```
11          return self._key
12
13      @property
14      def reverse(self) -> bool:
15          return self._reverse
16
17      @property
18      def name(self) -> str:
19          return self.key.name
```

B.3.1.5 ordered_queue.py

```
1  from typing import TypeVar
2
3  T = TypeVar("T")
4
5
6  class OrderedQueue(list[T]):
7      def __init__(self, items: list[T] = None) -> None:
8          super().__init__()
9          if (items is not None):
10             for item in items:
11                 self.append(item)
12
13     def append(self, data: T) -> None:
14         lim_inf: int = 0
15         lim_sup: int = len(self) - 1
16         while (lim_inf <= lim_sup):
17             idx: int = (lim_sup + lim_inf) // 2
18             if (data == self[idx]):
19                 break
20             elif (data > self[idx]):
21                 lim_inf = idx + 1
22             else:
23                 lim_sup = idx - 1
24         else:
25             self.insert(lim_inf, data)
26
27         # for idx, item in enumerate(self):
28         #     if (data < item):
```

```
29          #           self.insert(idx, data)
30          #           return None
31
32          # super().append(data)
33          return None
```

B.3.1.6 rect.py

```
1  from typing import NoReturn
2
3  from .coordinate import Coordinate
4
5
6  class Rect:
7      def __init__(self,
8                   width: float = float("inf"),
9                   height: float = float("inf"),
10                  position: Coordinate | None = None
11                  ) -> None:
12          self._width: float = width
13          self._height: float = height
14          self._area: float = width * height
15          self._perimeter: float = 2 * (width + height)
16          self._position: Coordinate | None = position
17
18      @property
19      def height(self) -> float:
20          return self._height
21
22      @property
23      def width(self) -> float:
24          return self._width
25
26      @property
27      def area(self) -> float:
28          return self._area
29
30      @property
31      def perimeter(self) -> float:
32          return self._perimeter
```

```
33
34      @property
35      def position(self) -> Coordinate | None:
36          return self._position
37
38      @position.setter
39      def position(self, coordinate: Coordinate | None) -> NoReturn:
40          self._position = coordinate
41
42      @property
43      def end(self) -> Coordinate | None:
44          if self.position is None:
45              return None
46          return self.position + (self.width, self.height)
```

B.3.1.7 region.py

```
1  from __future__ import annotations
2
3  from .coordinate import Coordinate
4  from .rect import Rect
5
6
7  class NotRegion(Exception):
8      pass
9
10
11 class Region(Rect):
12     id_: int = 0
13
14     def __init__(self,
15                  start: Coordinate | tuple[float, float],
16                  end: Coordinate | tuple[float, float]
17                  ) -> None:
18         if isinstance(start, tuple):
19             start = Coordinate(x=start[0], y=start[1])
20         if isinstance(end, tuple):
21             end = Coordinate(x=end[0], y=end[1])
22
23         if (start.x == end.x) or (start.y == end.y):
```

```
24              raise NotRegion
25
26          start: Coordinate = start
27          self._end: Coordinate = end
28          height: float = end.y - start.y
29          width: float = end.x - start.x
30          super().__init__(width=width, height=height, position=start)
31          self._id: int = Region.id_
32          Region.id_ += 1
33
34      @property
35      def id(self) -> int:
36          return self._id
37
38      @property
39      def start(self) -> Coordinate:
40          return self.position
41
42      @property
43      def end(self) -> Coordinate:
44          return self._end
45
46      def __lt__(self, other: Region) -> bool:
47          return (self.start < other.start)
48
49      def __eq__(self, other: Region | None) -> bool:
50          if other is None:
51              return False
52          return (self.start == other.start and self.end == other.end)
53
54      def __repr__(self) -> str:
55          return f"{self.start} {self.end}"
```

B.3.2 utils/

B.3.2.1 descending.py

```
1  from enum import Enum
2
3
```

```
class Descending(Enum):
    TRUE = "Sim"
    FALSE = "Não"
```

B.3.2.2 folders.py

```
CSV: str = "output/csv"
DATA: str = "output/data"
FIGURES: str = "output/figures"
INSTANCES: str = "instances/BKW"
```

B.3.2.3 functions.py

```
from os import makedirs, scandir

from tabulate import tabulate


def tabulate_ins2d(folder: str = "instances/GCUT"):
    for file in scandir(folder):
        with open(file, "r") as f:
            lines = f.readlines()

        data = [lines[0].split(), lines[1].split()]
        for line in lines[2:]:
            new_line = line.split()
            data.append(new_line)

        with open(file, "w") as f:
            s = tabulate(tabular_data=data, tablefmt="plain")
            f.write(s)
    return None


def csv_to_table(csv_folder: str, table_folder: str) -> None:
    for file in scandir(csv_folder):
        file_name = file.name
        if not file.is_file():
            csv_to_table(csv_folder=f"{csv_folder}/{file_name}",
                         table_folder=f"{table_folder}/{file_name}")
            continue
```

```
29          with open(file, "r") as f:
30              lines = f.readlines()
31
32          data = []
33          for line in lines:
34              data.append(line.split(","))
35
36          makedirs(table_folder, exist_ok=True)
37
38          with open(f"{table_folder}/{file_name}.dat", "w") as f:
39              s = tabulate(tabular_data=data, headers="firstrow",
                ↪  tablefmt="plain")
40              f.write(s)
41      return None
```

B.3.2.4 instances.py

```
1  from enum import auto, Enum
2
3
4  class InstanceSet(Enum):
5      BKW = auto()
6      GCUT = auto()
7      NGCUT = auto()
8      OF = auto()
9      OKP = auto()
```

B.3.2.5 order_key.py

```
1  from enum import Enum
2
3
4  class OrderKey(Enum):
5      AREA = "Área"
6      PERIMETER = "Perímetro"
7      HEIGHT = "Altura"
8      WIDTH = "Largura"
9      ID = "Id"
```

B.3.2.6 `split_mode.py`

```
1 from enum import Enum
2
3
4 class SplitMode(Enum):
5     VERTICALLY = "Vertical"
6     HORIZONTALLY = "Horizontal"
7     MAX_AREA = "Maior área"
8     NONE = "Sobrepostas"
```

APÊNDICE C – Artigo

Problema de empacotamento de retângulos: avaliação de métodos de solução baseados em *bottom-left*

Gabriel Medeiros Lopes Carneiro[1], **Pedro Belin Castellucci**[1], **Rafael de Santiago**[1]

[1] Departamento de Informática e Estatística
Universidade Federal de Santa Catarina (UFSC) – Florianólis, SC – Brasil

gabriel.mlc@grad.ufsc.br, pedro.castellucci@ufsc.br, r.santiago@ufsc.br

Resumo. *Problemas de empacotamento consistem em alocar um conjunto de itens $\mathcal{I}$ em uma caixa $\mathcal{B}$. No problema de empacotamento da mochila, foco deste trabalho, cada item é associado a um valor e busca-se uma solução que maximize a soma dos valores dos itens alocados. Este trabalho compara 40 métodos de solução criados com base na heurística bottom-left para o problema de empacotamento de retângulos. Os métodos criados são uma combinação de diferentes formas de ordenação dos itens e criação de regiões, as quais evitam as sobreposições e o domínio contínuo presentes no problema. O principal resultado foi a alta competitividade de diferentes modos de ordenação, não sendo a área a única relevante, com o perímetro obtendo os melhores resultados.*

Abstract. *Packing problems consist of allocating a set of items $\mathcal{I}$ into a box $\mathcal{B}$. In the knapsack packing problem, the focus of this work, each item is associated with a value and a solution is sought that maximizes the sum of the values of the allocated items. This work compare 40 created solution methods based on bottom-left heuristic for the rectangle packing problem. The methods created are a combination of different ways of ordering items and creating regions, which avoid superposition and continuous domain present in the problem. The main result was the high competitiveness of different ordering modes, the area not being the only relevant one, with the perimeter obtaining the best results.*

1. Introdução

Serviços de loja *online* com entrega como Amazon e Mercado Livre estão tornando-se cada vez mais presentes no dia a dia. Para tornar as entregas mais rápidas é necessário fazer uma série de estudos de logística e planejamento sobre como organizar os produtos nos estoques e nos veículos de entrega [Silva et al. 2022, Morabito Neto and Widmer 1992], muitas vezes sendo necessário considerar a ordem em que eles precisarão ser retirados. Além de tornar o processo mais rápido, a organização também pode permitir o melhor uso de espaços, aumentando a quantidade máxima de itens ou evitando o desperdício dos espaços.

Ainda sobre evitar desperdícios, esse quesito é muito importante para as indústrias de papel, móveis, têxtil e metal-mecânica [Queiroz 2022, Cavali 2004, Belluzzo and Morabito 2005]. Todas essas áreas querem gerar o máximo de produtos com o mínimo de recursos materiais utilizados, para evitar o descarte desnecessário do material e prejuízos financeiros.

Os problemas citados são considerados problemas de corte e empacotamento. Problemas de corte envolvem cortar um objeto, como blocos de gesso, chapas de aço e barras de ferro, em itens menores. Enquanto problemas de empacotamento tratam sobre alocar um conjunto de itens $\mathcal{I}$ em um recipiente $\mathcal{B}$. Ambos são equivalentes entre si e é possível separá-los de acordo com sua dimensão.

O caso 2D, dimensão de estudo deste trabalho, possui uma vasta literatura de métodos de solução. As abordagens de solução se dividem entre exatas, que buscam a solução ótima do problema, e heurísticas, as quais podem não encontrar uma solução ótima, mas conseguem uma solução aceitável em tempo hábil. Dentre os métodos de solução exatos, um que se destaca é o procedimento de busca em árvore [Beasley 1985], mas existem muitos outros na literatura [Iori et al. 2021, Fekete and Schepers 1997, Delorme et al. 2016, Kenmochi et al. 2009]. Na parte de heurísticas, tem-se a *bottom-left* [Baker et al. 1980, Chehrazad et al. 2022] e *skyline* [Wei et al. 2011], as heurísticas também possuem grande presença na literatura [Burke et al. 2004, Rakotonirainy and van Vuuren 2020, Hopper and Turton 2001b, Chen et al. 2019, Huang and Chen 2007, Hopper and Turton 2001a].

Este trabalho visa criar métodos de solução para o problema de empacotamento no espaço de duas dimensões, onde as peças são retangulares e com um recipiente também retangular, mais especificamente na versão do empacotamento 2D da mochila, considerado NP-difícil [Iori et al. 2022]. Nessa versão do problema, dado um conjunto de itens $\mathcal{I}$, com cada item i possuindo um valor p_i, e uma caixa $\mathcal{B}$, o objetivo é maximizar a soma dos valores dos itens alocados dentro do recipiente. A abordagem escolhida para resolver o problema foi utilizar a heurística *bottom-left*, devido a sua simplicidade e aos limites computacionais e de tempo ao escolher algum método exato. Mesmo com a heurística sendo proposta em 1980, ela ainda está presente na literatura recente [Chehrazad et al. 2022, Hopper and Turton 2001b, Wei et al. 2011].

O principal objetivo deste trabalho é criar métodos de solução para o problema de empacotamento da mochila de peças retangulares, todos baseados na heurística *bottom-left*. Outros objetivos mais específicos são: implementar a *bottom-left* e os métodos derivados em Python, executá-los com instâncias de teste da literatura, comparar seus resultados e identificar vantagens e desvantagens de cada um.

2. Definição

Com o escopo do estudo definido como problema de empacotamento de retângulos, é possível ver sua definição formal. De acordo com [Iori et al. 2022], dado uma caixa retangular $\mathcal{B} = (W, H)$ de comprimento $W \in \mathbb{Z}_+$ e altura $H \in \mathbb{Z}_+$ e um conjunto $\mathcal{I}$ de itens também retangulares, onde cada item $i \in \mathcal{I}$ com comprimento $w_i \in \mathbb{Z}_+, w_i \leq W$ e altura $h_i \in \mathbb{Z}_+, h_i \leq H$. Um empacotamento $\mathcal{I}' \subseteq \mathcal{I}$ em $\mathcal{B}$ pode ser descrito como uma função $\mathcal{F} : \mathcal{I}' \to \mathbb{Z}_+^2$ que mapeie cada item $i \in \mathcal{I}'$ para um par de coordenadas $\mathcal{F}(i) = (x_i, y_i)$, de forma:

$$x_i \in \{0, \ldots, W - w_i\}, y_i \in \{0, \ldots, H - h_i\} \left(i \in \mathcal{I}'\right) \quad (1)$$

$$[x_i, x_i + w_i) \cap [x_j, x_j + w_j) = \emptyset \text{ ou } [y_i, y_i + h_i) \cap [y_j, y_j + h_j) = \emptyset \left(i, j \in \mathcal{I}', i \neq j\right) \quad (2)$$

Nesse modo de representação a caixa está posicionada no plano cartesiano, com seu canto inferior esquerdo na origem. Já as coordenadas $\mathcal{F}(i) = (x_i, y_i)$ representam a posição em que o canto inferior esquerdo da peça será alocado. A Restrição 1 garante que cada item deve estar inteiramente na caixa, enquanto a Restrição 2 impede sobreposição entre itens. Ambas restrições indicam uma orientação fixa, ou seja, peças não podem ser rotacionadas.

3. Métodos de solução

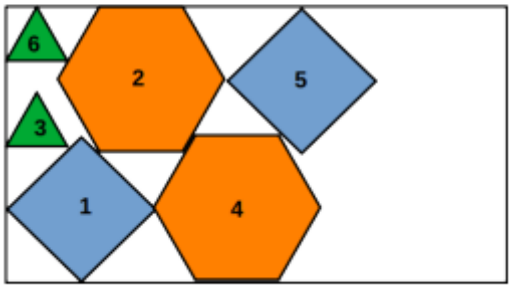
Como descrito na Seção 1, a maioria das classes do problema são NP-difíceis. Isso torna métodos de soluções exatos, os quais buscam pela solução ótima, extremamente custosos em tempo e recursos computacionais em instâncias de porte moderado, muitas vezes sendo inviáveis por falta de algum desses dois motivos. Consequentemente a literatura é dominada por abordagens que usam heurísticas e meta-heurísticas, sendo a *bottom-left* uma das principais estratégias de solução e será usada no estudo deste trabalho.

A *bottom-left* é uma heurística construtiva proposta por [Baker et al. 1980]. Embora tenha sido proposta a décadas, ainda é bastante usada na literatura atual, além de poder ser usada como componente de algoritmos mais sofisticados e para diferentes classes e variantes do problema. Ela foi utilizada nos trabalhos de [Hopper and Turton 2001b] na comparação de vários métodos de solução, [Wei et al. 2011] trazendo uma revisão do método e seus derivados e, mais recentemente, [Chehrazad et al. 2022] através de uma adaptação para o empacotamento de itens irregulares de forma gulosa.

Sua premissa é simples, dado uma fila de itens como entrada, enquanto ela não estiver vazia, basta retirar o primeiro item dela e alocar no canto mais a baixo e à esquerda quanto for possível [Bartmeyer et al. 2021], sem sobreposições entre peças. Caso não exista uma posição válida, a peça é desconsiderada e passa-se para próxima da fila. A Figura 1 mostra um exemplo de alocação para um dado conjunto de peças regulares.

Figura 1. Representação de alocação usando *bottom-left*.

(a) Sequência para alocação.



(b) Alocação gerada pela regra *bottom-left*.

Vale destacar que a própria ordem da fila pode gerar resultados diferentes, alterando a qualidade da solução. Um dos resultados esperados deste trabalho é identificar se há alguma forma de ordenação que se destaque na qualidade de solução, através da

comparação entre os diferentes modos. Para isso, serão usados conjuntos de instâncias frequentemente utilizados na literatura.

3.1. Critérios de ordenação

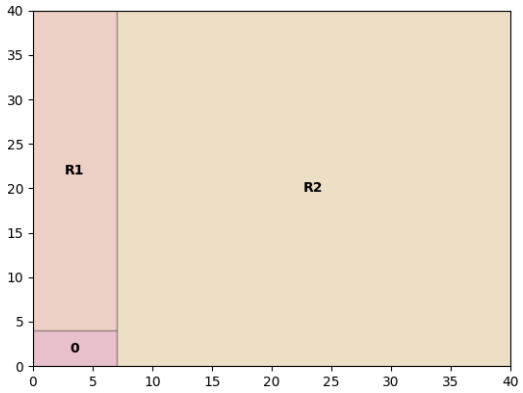
Para determinar o impacto da ordenação da fila, cinco critérios de ordenação foram escolhidos, sendo eles: área, perímetro, largura, altura e *id*. A ordenação por *id* considera a ordem em que os itens foram colocados na lista, ou seja, seria a forma padrão de resolver e ele será a base para definir se os demais critérios possuem algum benefício. Além disso, cada critério pode ser usado para ordenar a fila em ordem crescente ou decrescente, algo que também será analisado. Na literatura o mais comum é utilizar a ordenação decrescente pela área [Chen et al. 2019].

3.2. Criação de regiões

A sobreposição de peças e o domínio contínuo podem ser resolvidos utilizando a estratégia de criação de regiões. Com essa técnica, a Restrição 2 é trivialmente satisfeita. Nela, ao posicionar uma peça, duas regiões são criadas e o item seguinte será somente posicionado se couber em uma das regiões disponíveis.

Supondo um recipiente com altura e largura 40 e um item 0 com altura 4 e largura 7. Quando o item for posicionado na coordenada (0, 0), duas regiões, R1 e R2, serão criadas (Figura 2). A região R1 começará na coordenada (0, 4) e a R2 na (7, 0).

Figura 2. Regiões criadas traçando uma linha vertical.

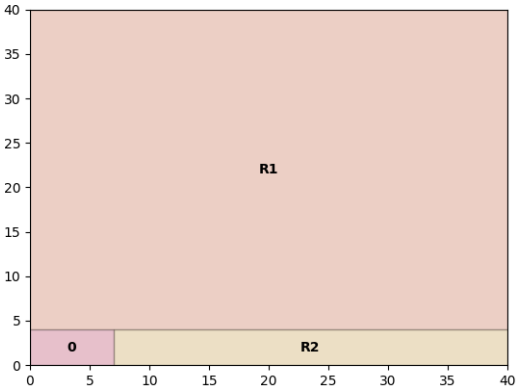


Agora o domínio passa a ser somente o canto inferior esquerdo de cada uma das regiões e sobreposições deixam de ser possíveis. Além disso, a regra para definir se uma peça cabe em dada região é igual a Restrição 1, simplificando o algoritmo. A fim de identificar o impacto das regiões na solução do modelo, quatro formas de criação delas foram usadas.

A primeira delas é **traçando uma linha vertical** a partir do canto superior direito de cada peça alocada (Figura 2). Nela a região R1 terá altura 36 e largura 7, indo até à coordenada (7, 40). Enquanto a R2 possuirá altura 40 e largura 33, chegando até à coordenada (40, 40).

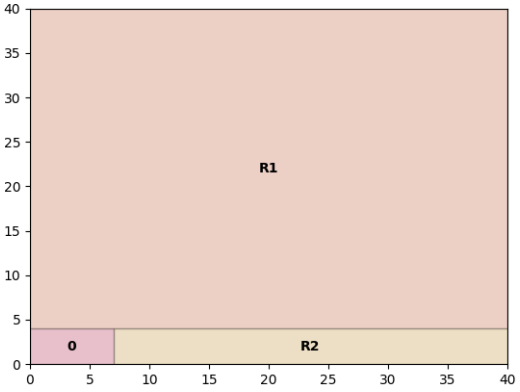
A segunda é semelhante à primeira, porém **traçando uma linha horizontal** (Figura 3). Nesse caso, R1 terá altura 36 e largura 40, indo até à coordenada (40, 40). Já R2 possuíra altura 4 e largura 33, chegando até à coordenada (40, 4).

Figura 3. Regiões criadas traçando uma linha horizontal.



Na terceira, a linha traçada (vertical ou horizontal) depende da área das regiões criadas com cada linha. Nesse modo o objetivo é maximizar a área de uma das regiões geradas, ele **identifica qual linha irá gerar a região de maior área e a traça**. Por exemplo, a Figura 2 gerou uma região com 252 de área e outra com 1320, enquanto a Figura 3 obteve regiões com 1440 e 132, então, nesse caso, a linha traçada será a horizontal (Figura 4).

Figura 4. Regiões criadas maximizando uma das regiões.

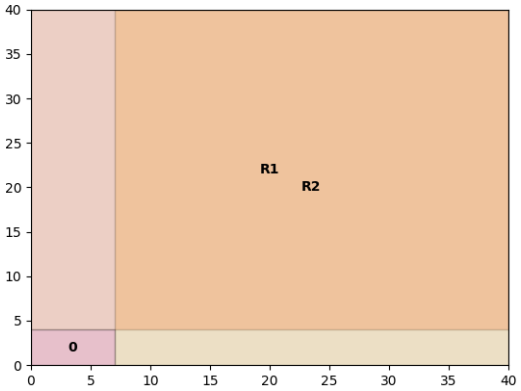


Maximizar uma região pode ser interessante, pois aumenta as chances do próximo item conseguir ser alocado, visto que uma das regiões será mais espaçosa. Em contrapartida, esse método também pode acabar gerando muitas regiões pequenas que não sejam utilizadas, diminuindo a qualidade da solução.

No quarto e último modo de criar regiões nenhuma linha é traçada, todas as regiões vão até o final do recipiente (Figura 5), criando **regiões sobrepostas**. R1 terá altura

40 e largura 36, enquanto R2 possuirá altura 40 e largura 33. Então, R1 e R2 terminarão na coordenada (40, 40). Nesse caso, sobreposições de peças podem ocorrer, então verificações são necessária para cumprir a Restrição 2. Ao fazer isso, é possível que mais peças sejam alocadas, visto que todas as regiões possuem área máxima. Esse modo foi criado para identificar se é de fato melhor que os demais e qual seu custo.

Figura 5. Regiões criadas possibilitando sobreposições.



Com os critérios para criação de regiões explicados, é possível diferenciá-los em dois tipos. O primeiro é dos que permitem sobreposição entre peças e, por isso, precisam de verificações para respeitar a Restrição 2 (regiões complexas), nesse tipo se encaixa somente o quarto modo. O segundo tipo contém os três primeiros critérios, onde somente a Restrição 1 precisa ser checada (regiões simples).

A Tabela 1 mostra, de forma resumida, os quatro modos de criar regiões, seu tipo (regiões simples ou complexas) e sua relação com a Restrição 2. A coluna "Divisão" indica o critério usado ao criar as regiões, enquanto a coluna "Restrição 2" mostra se a Restrição 2 é trivialmente satisfeita ou não. A última coluna ("Tipo"), indica se a região é simples ou complexa e está diretamente relacionada a Restrição 2.

Tabela 1. Modos de criar regiões, seu tipo e sua relação com a Restrição 2.

Modo	Divisão	Restrição 2	Tipo
1	Vertical	Trivial	Simples
2	Horizontal	Trivial	Simples
3	Maior área	Trivial	Simples
4	Regiões sobrepostas	Não trivial	Complexas

Todas as variações propostas foram implementadas em Python e avaliadas computacionalmente, os resultados são apresentados no Seção 4.

4. Resultados

Para testar os métodos de solução criados foram usadas 45 instâncias de teste da literatura, separadas em cinco conjuntos de instância de características diferentes: BKW,

Tabela 2. Configuração do computador de testes.

CPU	AMD Ryzen™ 5 3600X
RAM	16 GiB
Python	3.11.0
SO	Linux Mint 21.1 Cinnamon
Kernel	5.15

GCUT, NGCUT, OF e OKP. Todas as elas foram obtidas através da biblioteca pública 2DPackLib[1] [Iori et al. 2022].

O foco do trabalho é no empacotamento 2D da mochila, com o critério de maximização sendo a área ocupada do espaço. Mas nem todos conjuntos foram feitos para ser resolvidos dessa forma, nesses casos foram feitas leves adaptações para usá-los. O motivo de usar instâncias feitas com outro objetivo é para não viciar o modelo em instâncias específicas.

Como são cinco critérios de ordenação (Seção 3.1), com cada critério podendo ser crescente ou decrescente, quatro formas de criar regiões (Seção 3.2) e 45 instâncias, tem-se o total de 1800 casos de teste. Além disso, para conseguir resultados mais fiéis, a média, mediana e desvio padrão do tempo de execução foram calculados. Por isso, cada caso foi executado cinco vezes, totalizando 9000 execuções. Outros dados como a qualidade de solução (objetivo do trabalho), porcentagem de itens alocados e tempo, também foram computados. Todas as execuções foram feitas em um mesmo computador, com configurações conforme a Tabela 2.

Ao analisar a média, a mediana e o desvio padrão do tempo de execução, observou-se que a média e mediana possuem valores quase idênticos, enquanto o desvio padrão é pequeno ao ponto de poder ser ignorado, indicando que cinco execuções por caso de teste são suficientes. Portanto, a mediana e desvio padrão serão omitidos no restante do trabalho, podendo ser encontrados na versão completa dos dados gerados no Github[2].

Nas tabelas apresentadas nas seções seguintes, as colunas "Vitórias" e "Empates" trazem os resultados de forma quantitativa, enquanto a coluna "Qualidade %" de modo qualitativo. Nos testes, o valor p_i dos itens sempre é sua área, desconsiderando os valores dados pelas instâncias, caso hajam.

A qualidade de solução é determinada pela área ocupada do recipiente, no caso de todos os itens serem alocados, a qualidade é 100% independente da área do recipiente. A coluna "Vitórias" indica quantas vezes tal método de solução obteve o melhor resultado em comparação com os demais métodos em outras linhas. Enquanto a coluna "Empates" mostra a quantidade de vezes que o método conseguiu a melhor qualidade, mas outros também conseguiram. Essas colunas foram feitas da seguinte forma: entre cada combinação de critério de ordenação, modo de criar regiões e instâncias, é feita a comparação se a qualidade de solução foi melhor para ordenação crescente ou decrescente. No caso de am-

[1]Disponível em: https://site.unibo.it/operations-research/en/research/2dpacklib. Acessado em: 7 de julho de 2023.

[2]Disponível em: https://github.com/G-Carneiro/packing-problem/. Acessado em: 7 de julho de 2023.

bas conseguirem o melhor resultado, é acrescido 1 tanto na coluna "Vitórias", quanto na "Empates" de ambas. Por fim, a coluna "Tempo (s)" mostra o tempo médio de execução do método em segundos.

4.1. Ordenação crescente × decrescente

Uma primeira observação é a discrepância na qualidade de solução entre a ordenação crescente e a decrescente, algo já esperado. Na Tabela 3 é possível notar que ordenando de forma decrescente é possível ocupar cerca de 20% a mais do espaço (coluna "Qualidade %"), quando comparado a ordenação crescente, em média.

Tabela 3. Resultado da comparação entre ordenação crescente e decrescente.

Decrescente	Vitórias	Empates	Qualidade %	Tempo (s)
Sim	736	8	78.9136	1.7798e+00
Não	167	8	57.3060	2.3715e+00

Com isso, fica claro que ordenar a fila de entrada da *bottom-left* de modo decrescente é vantajoso em termos de qualidade, quantidade e tempo de execução.

4.2. Comparativo entre critérios de ordenação

A Tabela 4 mostra o comparativo entre os critérios de criação de regiões, somente considerando a ordenação decrescente, já que não existem motivos para usar a crescente. Representando os dados dessa forma fica fácil identificar que utilizar algum critério de ordenação para a fila de entrada é vantajoso, pois ao usar o *id* os resultados foram os piores. Além disso, percebe-se que as ordenações por área e perímetro obtiveram os melhores resultados, ainda que os demais também sejam competitivos.

Tabela 4. Resultado da comparação entre critérios de ordenação decrescente.

Ordenação	Vitórias	Empates	Qualidade %	Tempo (s)
Área	63	39	82.7353	1.5874e+00
Perímetro	71	38	84.6986	1.5769e+00
Altura	40	16	77.4182	1.5655e+00
Largura	66	24	81.1899	2.0805e+00
Id	16	5	68.5261	2.0889e+00

A alta competitividade entre os critérios de ordenação é interessante, pois a maioria dos trabalhos na literatura, como o de [Chen et al. 2019], usam somente ordenação pela área, e isso pode ser um forte indicativo que os demais critérios devem ser mais explorados em certas circunstâncias.

4.3. Comparativo entre criação de regiões

Conforme mostra a Tabela 5, os modos criados traçando uma linha vertical ou horizontal apresentaram qualidades semelhantes e os menores tempos de execução, mas o método o qual traça uma linha vertical obteve mais vitórias. Regiões criadas para maximizar uma das mesmas conseguiram o segundo melhor resultado qualitativo e quantitativo, ao custo de um pequeno acréscimo no tempo de execução em relação aos dois primeiros. O último

Tabela 5. Resultado da comparação entre criação de regiões - ordenação decrescente.

Divisão	Vitórias	Empates	Qualidade %	Tempo (s)
Vertical	98	79	76.4030	2.7157e-03
Horizontal	70	60	75.9970	6.2101e-03
Maior área	104	89	79.7175	1.3743e-02
Sobrepostas	176	119	83.6420	7.2176e+00

modo de fato conseguiu os melhores resultados, porém a um custo altíssimo, levando cerca de 1000 vezes mais tempo que métodos mais rápidos.

A Tabela 6 traz um comparativo entre os dois tipos de criação de regiões, os que é preciso checar sobreposição e os que não (Seção 3.2). Na primeira linha da tabela a coluna "Qualidade %" representa a média do melhor resultado obtido em cada instância e a coluna "Tempo Total (s)" mostra a soma dos tempos que cada método de solução levou para cada instância. As duas colunas consideram somente métodos de solução que usam regiões onde não são necessárias verificações de sobreposição, ou seja, são considerados 30 modos de solução. A segunda linha considera somente método de solução o qual utiliza ordenação decrescente pela área e criação de regiões onde é necessário verificar sobreposições, esse método foi escolhido para o comparativo por apresentar os melhores resultados quantitativos e ser o segundo melhor qualitativamente.

Tabela 6. Resultado da comparação entre tipos de regiões.

Sobreposição	Qualidade %	Tempo Total (s)
Não	90.8278	1.6299e+01
Sim	87.2957	2.8313e+02

Com a Tabela 6 fica claro que, por mais que usar regiões onde é preciso checar sobreposições apresente o melhor resultado, é melhor executar todos demais métodos os quais não precisem e escolher somente o de melhor solução, pois assim é possível obter, na média, soluções de maior qualidade e ainda levando 10 vezes menos tempo.

5. Conclusão

No trabalho foram avaliados experimentalmente 40 métodos de solução baseados na heurística *bottom-left* para o problema de empacotamento de retângulos, considerando a versão da mochila e com o valor dos itens sua própria área. O resultado mais óbvio obtido foi a supremacia da ordenação decrescente sobre a crescente (Seção 4.1), não compensando utilizar a segunda para solucionar o problema.

O alto uso da ordenação decrescente pela área na literatura [Chen et al. 2019] foi justificado pelos resultados, já que essa composição conseguiu alguns dos melhores números (Seção 4.2). Os resultados também indicam que qualquer critério de ordenação obtêm melhores resultados do que deixar a fila desordenada (ordenação por *id*). Mas a alta competitividade de outros critérios de ordenação, esses não tão comuns na literatura, conseguindo melhores resultados em algumas instâncias, mostra que ainda há espaço para outras formas de ordenação.

Em relação às diferentes regiões criadas, por mais que regiões complexas tenham obtido melhores resultados (Seção 4.3), não compensa utilizá-las. Ao executar todos os métodos de regiões simples é possível conseguir melhores resultados e em menor tempo, quando comparado ao método de ordenação decrescente pela área e usando regiões complexas (ver a Tabela 6 na Seção 4.3).

Referências

Baker, B. S., Coffman Jr, E. G., and Rivest, R. L. (1980). Orthogonal packings in two dimensions. *SIAM Journal on Computing*, 9(4):846.

Bartmeyer, P. M., Oliveira, L. T. d., Toledo, F. M. B. d., and Leao, A. A. S. (2021). Aprendizado por reforço aplicado ao problema de empacotamento de peças irregulares em faixas. *Anais*.

Beasley, J. E. (1985). An exact two-dimensional non-guillotine cutting tree search procedure. *Operations Research*, 33(1).

Belluzzo, L. and Morabito, R. (2005). Otimização nos padrões de corte de chapas de fibra de madeira reconstituída: um estudo de caso. *Pesquisa Operacional*, 25:391–415.

Burke, E. K., Kendall, G., and Whitwell, G. (2004). A new placement heuristic for the orthogonal stock-cutting problem. *Operations Research*, 52(4):655–671.

Cavali, R. (2004). Problemas de corte e empacotamento na indústria de móveis: um estudo de caso.

Chehrazad, S., Roose, D., and Wauters, T. (2022). A fast and scalable bottom-left-fill algorithm to solve nesting problems using a semi-discrete representation. *European Journal of Operational Research*, 300(3):809–826.

Chen, M., Wu, C., Tang, X., Peng, X., Zeng, Z., and Liu, S. (2019). An efficient deterministic heuristic algorithm for the rectangular packing problem. *Computers & Industrial Engineering*, 137:106097.

Delorme, M., Iori, M., and Martello, S. (2016). Bin packing and cutting stock problems: Mathematical models and exact algorithms. *European Journal of Operational Research*, 255(1):1–20.

Fekete, S. P. and Schepers, J. (1997). A new exact algorithm for general orthogonal d-dimensional knapsack problems. In *Algorithms—ESA'97: 5th Annual European Symposium Graz, Austria, September 15–17, 1997 Proceedings 5*, pages 144–156. Springer.

Hopper, E. and Turton, B. C. (2001a). A review of the application of meta-heuristic algorithms to 2d strip packing problems. *Artificial Intelligence Review*, 16:257–300.

Hopper, E. B. C. H. and Turton, B. C. H. (2001b). An empirical investigation of meta-heuristic and heuristic algorithms for a 2d packing problem. *European Journal of Operational Research*, 128(1):34–57.

Huang, W. and Chen, D. (2007). An efficient heuristic algorithm for rectangle-packing problem. *Simulation Modelling Practice and Theory*, 15(10):1356–1365.

Iori, M., de Lima, V. L., Martello, S., Miyazawa, F. K., and Monaci, M. (2021). Exact solution techniques for two-dimensional cutting and packing. *European Journal of Operational Research*, 289(2):399–415.

Iori, M., de Lima, V. L., Martello, S., and Monaci, M. (2022). 2dpacklib: a two-dimensional cutting and packing library. *Optimization Letters*, 16(2):471–480.

Kenmochi, M., Imamichi, T., Nonobe, K., Yagiura, M., and Nagamochi, H. (2009). Exact algorithms for the two-dimensional strip packing problem with and without rotations. *European Journal of Operational Research*, 198(1):73–83.

Morabito Neto, R. and Widmer, J. A. (1992). *Abordagem em grafo-e-ou para o problema do empacotamento: aplicacao ao carregamento de paletes e conteineres*. PhD thesis.

Queiroz, L. R. d. S. (2022). *Estudo de problemas de corte de itens irregulares com incertezas*. PhD thesis, Universidade de São Paulo.

Rakotonirainy, R. G. and van Vuuren, J. H. (2020). Improved metaheuristics for the two-dimensional strip packing problem. *Applied Soft Computing*, 92:106268.

Silva, L. C. d., Queiroz, T. A. d., and Toledo, F. M. B. d. (2022). Integer formulations for the integrated vehicle routing problem with two-dimensional packing constraints. *Pesquisa Operacional*, 42.

Wei, L., Oon, W.-C., Zhu, W., and Lim, A. (2011). A skyline heuristic for the 2d rectangular packing and strip packing problems. *European Journal of Operational Research*, 215(2):337–346.