

AN APPLICATION MULTIOBJECTIVE FOR OPEN DIMENSIONAL PROBLEM GUILLOTINED

Dayanne Gouveia Coelho^a, Marcelus Xavier Oliveira^a, Elizabeth Fialho Wanner^a, Sérgio Ricardo de Souza^a, Marcone Jamilson Freitas^b and Eduardo Gontijo Carrano^a

^a*Centro Federal de Educação Tecnológica de Minas Gerais - CEFET-MG, Av. Amazonas, 7675, Nova Gameleira, 30510-000, Belo Horizonte, MG, Brasil, <http://www.mmc.cefetmg.br>*

^b*Departamento de Computação, ICEB, Universidade Federal de Ouro Preto/n, Morro do Cruzeiro, 35400-000, Ouro Preto, MG, Brasil, <http://www.decom.ufop.br>*

Keywords: Open Dimensional Problem; SPEA 2; Level Algorithms.

Abstract. Este artigo apresenta uma aplicação do algoritmo genético multiobjetivo SPEA 2 (Strength Pareto Evolutionary Algorithm 2), em conjunto com duas heurísticas de encaixe, First-Fit (FF), Best-Fit (BF), First-Fit Decreasing Height (FFDH) e Best-Fit Decreasing Height (BFDH), para resolver o Problema de Corte com Dimensão Aberta (PCDA) guilhotinado. O PCDA multiobjetivo consiste em alocar um conjunto de peças menores (itens) em uma peça maior (objeto), de forma que sejam minimizados a altura utilizada deste objeto e o número de cortes realizados. Para resolver o problema, incorpora-se ao SPEA 2 a fase de construção do algoritmo GRASP (Greedy Randomized Adaptative Search Procedure), para gerar parte da população inicial. A metodologia proposta foi testada em um conjunto de problemas-testes da literatura. Os resultados mostram que todos os métodos foram capazes de gerar um conjunto representativo de soluções não-dominadas, em que cada ponto deste conjunto representa uma solução de compromisso entre esses dois objetivos.

1 INTRODUÇÃO

O interesse deste trabalho está voltado para o Problema de Corte (PC), que possui uma grande aplicação em problemas práticos de várias indústrias. Na maioria delas, o arranjo dos cortes é determinado pela experiência do funcionário que realiza o corte e não é baseado em métodos científicos. Este fato pode acarretar em uma sobra excessiva de material. Em alguns casos, esta sobra não pode ser aproveitada, refletindo no custo final do produto. O Problema de Corte é tratado, principalmente, com o objetivo de minimizar o número de objetos utilizados ou o desperdício da matéria-prima. Porém, existem outros objetivos que também podem interferir no arranjo final do produto. Como exemplo, pode-se considerar o tempo de preparação da máquina para realizar a troca de direção do corte. Assim, um bom planejamento do corte evita a necessidade de constantes preparações das máquinas para os tamanhos dos produtos requisitados, o que pode minimizar os efeitos negativos gerados pelo desperdício sobre os custos de produção. Neste trabalho, o número de cortes realizado durante o processo será utilizado como uma outra função objetivo a ser minimizado.

Neste trabalho, é estudado o Problema de Corte com Dimensão Aberta (PCDA), caso particular dos PC, que é denotado por [Wäscher et al. \(2007\)](#) como *Open Dimensional Problem*- ODP. O PCDA consiste em determinar o melhor arranjo de um conjunto de itens retangulares, sobre um objeto maior, que possui largura fixa e altura variável, com o objetivo de minimizar a altura utilizada e o número de cortes realizados. Esta variação do problema pode ser encontrada, principalmente, em indústrias que realizam o corte de bobinas ou rolos. Ao PCDA tratado, são acrescentadas restrições relativas ao tipo de corte (guilhotinado); a limitação do número de cortes (2-estágios) e a orientação dos itens (não rotacionados).

Os trabalhos de [Dyckhoff and Finke \(1992\)](#), [Lodi et al. \(1999\)](#) e [Lodi et al. \(2002\)](#) fazem uma revisão bibliográfica do PCDA, apresentando os modelos matemáticos propostos e os métodos de resolução através do uso de algoritmos aproximados, heurísticas, metaheurísticas e métodos exatos. Os trabalhos que utilizam métodos exatos trabalham com instâncias da literatura de até 100 ou 200 itens. Este fato é justificado pelo alto custo computacional que esses métodos necessitam para resolver problemas com instâncias que possuem uma grande quantidade de itens. Quando o problema apresenta um número maior de itens, os trabalhos propõem o uso de métodos heurísticos, como por exemplo, heurísticas baseadas nos algoritmos evolutivos para tratar do PCDA numa versão mono-objetivo, (ver [Hifi \(1998\)](#), [Liu and Teng \(1999\)](#) e [de Andrade \(2009\)](#)).

Trabalhos que fazem uma abordagem multiobjetivo para o PCDA são recentes. Em [Tiwari and Chakraborti \(2006\)](#) é feito um estudo sobre o problema guilhotinado e sobre o problema não-guilhotinado, avaliando os resultados obtidos para cada uma dessas versões. Além disso, é feita uma comparação entre esses dois problemas, verificando-se que, com o acréscimo da restrição da guilhotina, o número de cortes para uma determinada instância aumenta. Em [Illich et al. \(2007\)](#) estuda-se o PCDA não-guilhotinado admitindo-se a rotação dos itens. São comparados dois algoritmos multiobjetivo, um puramente determinístico e outro de natureza evolucionária. Segundo os autores ambos os algoritmos retornam um conjunto de soluções razoável, mas verifica-se para algumas instâncias os métodos retornam poucos pontos para representar a curva de compromisso entre os objetivos. Nos trabalhos de [de Armas et al. \(2009\)](#) e [Miranda et al. \(2010\)](#), o PCDA possui a restrição do corte guilhotinado e admiti-se, além disso, a rotação dos itens durante o processo. Em todos esses trabalhos, as funções objetivo tratadas são o desperdício de matéria-prima e a velocidade de operação do equipamento que realiza o corte. O segundo objetivo é alcançado minimizando-se o número de cortes realizados durante

o processo.

Este trabalho apresenta uma metodologia de solução para o PCDA multiobjetivo, considerando o corte guilhotinado e não admitindo-se a rotação dos itens, variação do problema diferente dos trabalhos acima. É utilizado, para a solução do problema tratado, o algoritmo genético multiobjetivo SPEA 2 (Strength Pareto Evolutionary Algorithm 2) em conjunto com quatro heurísticas de encaixe: *First Fit* (FF), *Best Fit* (BF), *First Fit Decreasing Height* (FFDH) e *Best-Fit Decreasing Height* (BFDH). Essas heurísticas de encaixe são usadas para realizar o encaixe dos itens no objeto, e a metaheurística SPEA2 é combinada com cada uma dessas quatro heurísticas.

O artigo está estruturado como segue: a seção 2 descreve o PCDA multiobjetivo; a seção 3 apresenta a metodologia proposta; a seção 4 apresenta e faz uma análise dos resultados computacionais gerados, aplicando-se a metodologia proposta para instâncias da literatura; a seção 5 apresenta as conclusões obtidas a respeito do trabalho realizado.

2 PROBLEMA DE CORTE COM DIMENSÃO ABERTA

O Problema de Corte com Dimensão Aberta (PCDA) também é tratado na literatura como *2D strip packing problem* (Lodi et al. (1999), Lodi (1999)), ou como *2D-(Rectangular) Open Dimensional Problem* (ODP) Wäscher et al. (2007). Este problema define uma categoria de problemas, em que, um conjunto de itens pequenos devem ser completamente acomodados em um único objeto grande. O objeto a ser cortado pode ter duas ou mais dimensões, e uma dessas dimensões é considerada variável. Uma das decisões que envolve o PCDA diz respeito a extensão da dimensão variável.

A Figura 1 apresenta um exemplo do PCDA, que consiste em determinar o melhor arranjo para um conjunto de itens retangulares, sem sobreposição destes itens, sobre um objeto com largura (W) fixa e altura (H) variável.

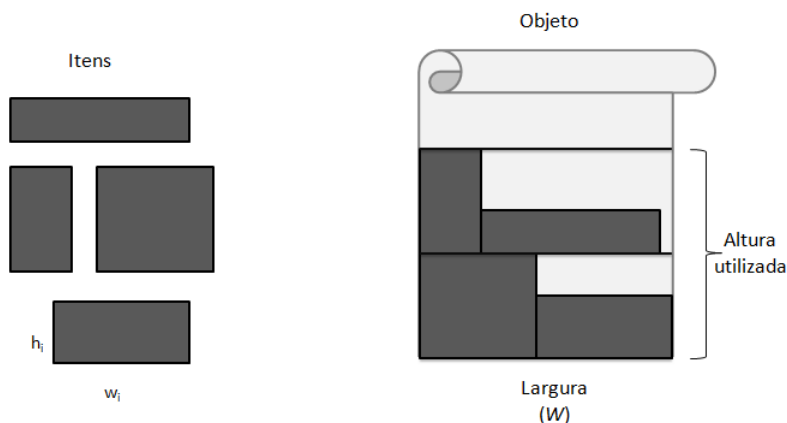


Figure 1: Problema de Corte com Dimensão Aberta (PCDA).

O PCDA multiobjetivo será descrito conforme os trabalhos de Lodi et al. (2002) e Lodi et al. (2004). Para tal, considere um conjunto de itens retangulares $I = r_1, \dots, r_n$, em que cada item $r_i = (w_i, h_i)$, possui largura w_i e altura h_i , e um objeto maior retangular S de largura fixa W e altura grande o suficiente (“altura infinita”) para alocar todos os itens (ver Figura 1). Os itens são encaixados com a borda, correspondente a sua largura, paralela a borda correspondente a largura do objeto. Além disso, é necessário que $w_i \leq W$ para $i = 1, \dots, n$. Dessa forma, o PCDA consiste em alocar, ortogonalmente e por faixas, todos os itens I no objeto S , atendendo as funções objetivo impostas.

O PCDA pode ser classificado de acordo com duas restrições: o tipo de corte (guilhotinado ou não) e a orientação dos itens (rotacionando ou não). A partir dessas duas restrições o problema pode ser dividido em quatro subtipos:

OG: não é permitido rotação dos itens e o corte é guilhotinado;

OF: não é permitido rotação dos itens e o corte é não guilhotinado;

RG: é permitido rotação dos itens e o corte é guilhotinado;

RF: é permitido rotação dos itens e o corte é não guilhotinado.

Os trabalhos encontrados na literatura que estudam o PCDA multiobjetivo, trabalham com variações do problema diferentes do problema tratado neste trabalho. A versão deste problema aqui tratado, onde não é permitido a rotação dos itens e o corte é guilhotinado, na prática é o mais utilizado. Em [Tiwari and Chakraborti \(2006\)](#) é estudado o PCDA RG; em [Illich \(2007\)](#), [Illich et al. \(2007\)](#) e [Illich and While \(2009\)](#) o PCDA OF e RF; e em [de Armas et al. \(2009\)](#) e [Miranda et al. \(2010\)](#) o PCDA RG. Na abordagem do problema proposta neste trabalho é estudado o PCDA OG, diferente dos trabalhos acima, assim é imposta ao problema as seguintes restrições:

- i) Os itens são alocados de forma ortogonal no objeto;
- ii) Os itens apresentam orientação fixa, ou seja, não podem ser rotacionados em 90° ;
- iii) O corte é do tipo guilhotinado em 2-estágios.

Este problema pode ser encontrado em indústrias que realizam o corte de bobinas ou rolos, tendo como objetivo cortar os itens, utilizando o mínimo de comprimento do rolo ou desperdiçar o mínimo possível da matéria-prima. No PCDA multiobjetivo, deve-se atender duas funções objetivo independentes: otimizar o desperdício da matéria-prima e a eficiência no processo de corte.

2.1 Formulação Matemática do PCDA

O modelo matemático do PCDA multiobjetivo é baseado no modelo proposto por [Lodi et al \(2004\)](#) para o PCDA mono-objetivo, considerando o caso especial em que o encaixe é feito por faixas.

No PCDA, a primeira faixa formada será chamada de primeiro nível. Os itens começam a ser inseridos nessa primeira faixa e a altura dela é determinada pela altura do item mais alto. Assim, a próxima faixa se inicia na linha horizontal desenhada no topo do item mais alto da faixa anterior. Sendo assim, considere as seguintes proposições para a formulação matemática do problema:

- i) Em cada faixa, o item mais à esquerda possui maior altura;
- ii) A primeira faixa do objeto (mais baixa) é a mais alta;
- iii) Os itens são ordenados de forma decrescente em relação à altura ($h_1 \geq h_2 \geq \dots \geq h_n$).

Para modelagem do problema, assume-se que o número de faixas formadas para alocar os itens é dada por n . A altura de cada faixa corresponde à altura h_i do primeiro item i a ser inserido (o item mais alto da faixa). Dessa forma, considerando n o número de faixas formadas para alocar os itens, a variável de decisão y_i é dada por:

$$y_i = \begin{cases} 1, & \text{se o item } i \text{ inicia a faixa } i \text{ (para } i = 1, \dots, n); \\ 0, & \text{caso contrário.} \end{cases}$$

Além disso, apenas os itens j , tais que $j > i$, podem ser alocados na faixa. Isso se justifica pois, dado um item j , tal que $j = i$ inicia a faixa i , ele não poderá ser atribuído novamente a nenhuma outra faixa. Dessa forma, a variável de decisão x_{ij} é definida como:

$$x_{ij} = \begin{cases} 1, & \text{se o item } j \text{ está alocado na faixa } i \text{ (para } i = 1, \dots, n-1); j > i \\ 0, & \text{caso contrário.} \end{cases}$$

Logo, a formulação matemática do problema é dada por:

$$\text{Minimizar} \quad \sum_{i=1}^n h_i \cdot y_i \quad (1)$$

$$\text{NumeroCortes} \quad (2)$$

$$\text{Sujeito a:} \quad \sum_{i=1}^{j-1} x_{ij} + y_j = 1, \quad j = 1, \dots, n; \quad (3)$$

$$\sum_{j=i+1}^n w_j x_{ij} \leq (W - w_i) y_i, \quad i = 1, \dots, (n-1); \quad (4)$$

$$\begin{aligned} x_{ij} &\in \{1, 0\} & i = 1, \dots, (n-1) j > i; \\ y_i &\in \{0, 1\} & i = 1, \dots, n; \end{aligned} \quad (5)$$

A função objetivo (1) tem como critério, minimizar a altura utilizada do objeto. A função objetivo (2) tem como objetivo otimizar a eficiência no processo de corte, minimizando o número de cortes realizados. Este cálculo é dado pelo número de arestas distintas ou “linhas” dentro do objeto a ser cortado, excluindo as margens no exterior do objeto. Como não existe expressão analítica para a função (2), o cálculo do número de cortes é dado de acordo com o Algoritmo 1. As restrições utilizadas na formulação do problema, (3), (4) e (5), garantem que o item só será alocado uma vez no objeto; que a largura do objeto não será ultrapassada em cada faixa; e definem o tipo das variáveis do problema, respectivamente.

3 METODOLOGIA PROPOSTA

Cada solução do PCDA é representada por uma sequência de números inteiros, em que cada posição da sequência representa a ordem na qual o item é encaixado. Para um problema com seis itens, por exemplo, uma possível solução é $s = (2 \ 1 \ 3 \ 6 \ 5 \ 4)$. A solução s indica que o primeiro item a ser encaixado é o item 2 e o último é o item 4. Assim, os indivíduos de uma determinada população serão representados por permutações deste conjunto de itens

Nesta seção é mostrado os algoritmos de encaixe utilizados para alocar os itens dentro do objeto; o Algoritmo Genético multiobjetivo utilizado para resolver o problema; a metodologia utilizada para construir a população inicial; a função de avaliação de uma solução e os operadores genéticos utilizados.

Algoritmo 1: Número de Cortes

Entrada: Conjunto de itens (I) com dimensões (w_i, h_i), número de faixas ($faixa$) e itens em cada faixa $itensFaixas(j)$

Saída: $NumeroCortes$

```
1 início
2    $NumeroCortes = faixa$ ;
3    $j = 1$ ;
4   para  $j = 1$  até  $faixa$  faça
5        $itens =$  quantidade de itens de  $itensFaixa(j)$ ;
6       se a largura disponível da faixa for igual a zero então
7            $NumeroCortes = NumeroCortes + 2 \times itens - 2$ ;
8       senão
9            $NumeroCortes = NumeroCortes + 2 \times itens - 1$ ;
10      fim
11  fim
12  para  $t = 1$  até  $itens$  faça
13      se houver itens com alturas iguais então
14           $NumeroCortes = NumeroCortes - 1$ ;
15      fim
16  fim
17  fim
18 fim
```

3.1 Algoritmos de Nível

Técnicas de encaixe (ou algoritmos de níveis) são algoritmos aproximados, desenvolvidos para tratar do encaixe de itens em problemas de corte [Ortmann \(2010\)](#). Nesses algoritmos, os itens são alocados em faixas, da esquerda para a direita, sendo que o início de uma nova faixa coincide com o topo do item mais alto da faixa anterior.

A utilização destas técnicas se deve ao fato delas serem mais rápidas e gerarem padrões guilhotinados. Neste trabalho, são utilizadas as técnicas de encaixe *First Fit* (FF), *Best-Fit* (BF), *First Fit Decreasing Height* (FFDH) e *Best-Fit Decreasing Height* (BFDH) (para maiores detalhes a respeito dessas técnicas, ver [Coffman-Jr et al. \(1996\)](#), [Johnson \(1974\)](#), [Ntene \(2007\)](#) e [Ortmann \(2010\)](#)). Em ambas as técnicas, o encaixe dos itens é feito na ordem em que eles aparecem na solução. Os itens são inseridos no canto inferior esquerdo do objeto até que a largura do objeto não seja mais suficiente. As técnicas são descritas a seguir:

First Fit (FF): Nesta técnica o item é inserido na faixa atual. Se o item não couber nesta faixa, forma-se uma nova faixa. Assim, após inserir o primeiro item, verifica-se a possibilidade de inserir o próximo item nesta mesma faixa. Caso não seja possível, uma nova faixa é formada.

Best Fit (BF): Nesta técnica após inserir o primeiro item, pesquisa-se a largura disponível da faixa. Se o item seguinte não puder ser inserido nesta faixa, verifica-se entre os demais, na ordem em que aparecem na solução, se é possível inseri-lo ou não. Uma nova faixa só é formada quando nenhum outro item puder ser inserido na faixa atual.

First Fit Decreasing Height (FFDH): Nesta técnica após inserir o primeiro item, é pesquisada a largura disponível das faixas existentes e o item é inserido na primeira faixa em que ele couber. Uma nova faixa só é formada, quando o item não puder ser inserido em nenhuma das faixas existentes.

Best Fit Decreasing Height (BFDH): Nesta técnica após inserir o primeiro item, pesquisa-se a largura disponível das faixas existentes. O próximo item é inserido na faixa que resultar no menor espaço residual em relação à largura. Uma nova faixa só é formada se o item não puder ser inserido em nenhuma das faixas existentes.

Nas técnicas FFDH e BFDH é permitido que as faixas formadas fiquem abertas, possibilitando que o item seja inserido apenas na faixa que resultar no melhor encaixe, como pode ser observado em [Ntene \(2007\)](#) e [Ortmann \(2010\)](#).

3.2 Strength Pareto Evolutionary Algorithm 2 (SPEA 2)

Os Algoritmos Genéticos (AGs) são métodos populacionais de busca, flexíveis e robustos que conseguem produzir soluções de boa qualidade em problemas complexos com um tempo computacional viável [Holland \(1975\)](#). Devido a essa flexibilidade e a eficácia na realização de uma busca global em diversos ambientes, os AGs vêm sendo aplicados com sucesso na resolução de problemas de otimização combinatória.

Os problemas multiobjetivo admitem diferentes funções objetivo que, em geral, são conflitantes entre si. Resolver um problema como este não é uma tarefa trivial, uma vez que a melhora em um objetivo ocorre em detrimento do outro objetivo. A solução de um problema multiobjetivo não é representada por apenas uma solução, mas por um conjunto de soluções eficientes, que representam relações de compromisso entre os objetivos considerados no problema.

Para resolver estes problemas, os métodos baseados em metaheurísticas têm sido uma alternativa muito utilizada ([Zitzler et al. \(2002\)](#)). A preferência pelos AGs se deve ao fato desses algoritmos trabalharem com uma população de soluções que podem conter informações sobre diferentes espaços de buscas, além de poder gerar, em apenas uma execução, um conjunto de soluções eficientes, denominado por conjunto Pareto-ótimo. Esse conjunto contém as soluções das diferentes funções objetivo do problema. As soluções presentes neste conjunto caracterizam a relação conflitante existente neste tipo de problema, ou seja, neste conjunto não é possível melhorar o valor de um critério sem tornar o outro pior.

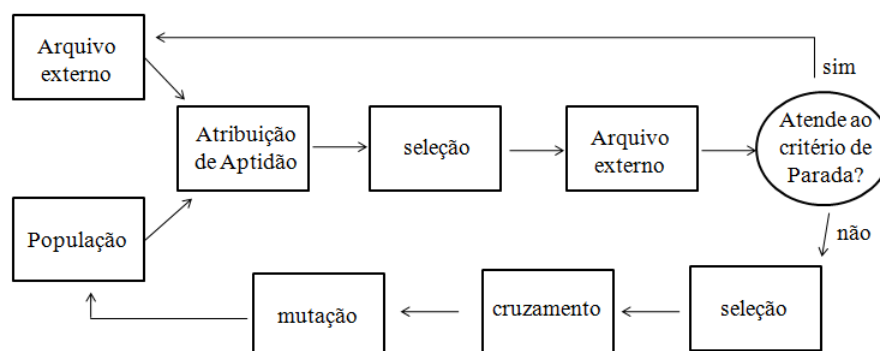


Figure 2: Estrutura básica do Algoritmo SPEA2.

Neste artigo, para resolver o PCDA multiobjetivo, é utilizado o *Strength Pareto Evolutionary Algorithm 2* (SPEA2) ([Zitzler et al. \(2002\)](#)). O SPEA2 possui a estrutura básica de um algoritmo genético multiobjetivo, utilizando uma população regular e um arquivo externo. Este arquivo externo contém as soluções não dominadas encontradas pelo algoritmo, de modo a garantir a preservação das melhores soluções encontradas. Este algoritmo se destaca por apresentar uma

boa convergência e garantir diversidade no conjunto de soluções retornado. Ele apresenta um esquema para atribuir melhor aptidão, o que determina, para cada indivíduo, quantos outros indivíduos ele domina e por quantos ele é dominado.

Em geral, o SPEA 2 apresenta a estrutura descrita na Figura 2. O algoritmo utiliza uma estratégia para atribuir aptidão que incorpora informações de densidade. O tamanho do arquivo externo é fixo. Assim, quando o número de indivíduos não dominados é menor que o tamanho pré-definido para este arquivo, ele será preenchido por indivíduos não dominados.

3.3 Construção da População Inicial

A população inicial do SPEA 2 é constituída por 30% de indivíduos gerados de forma aleatória e 70% utilizando a fase de construção do Algoritmo GRASP (*Greedy Randomized Adaptive Search Procedure*), proposto por [Feo and Resende \(1995\)](#).

Na fase de construção, cada solução é construída pela inserção dos itens, sendo um de cada vez. Cada indivíduo é construído através da metaheurística *Greedy Randomized Adaptive Search Procedure* -GRASP, cujo pseudocódigo está ilustrado no Algoritmo 2. O GRASP é um método constituído de duas fases, denominadas fase de construção e fase de refinamento, que são aplicadas repetidamente, retornando a melhor das soluções encontradas. As subseções seguintes mostram como esse algoritmo foi adaptado ao PCDA.

Algoritmo 2: GRASP

Entrada: $f(\cdot)$, $g(\cdot)$, $N(\cdot)$, $GRASP_{max,s}$
Saída: s

```

1 início
2    $f^* \leftarrow \infty$ ;
3   para  $Iter=1$  até  $GRASP_{max}$  faça
4     Construção(  $g(\cdot)$ ,  $k$ ,  $s$ );
5     Buscalocal(  $f(\cdot)$ ,  $N(\cdot)$ ,  $s$ );
6     se ( $f(s) < f^*$ ) então
7        $s^* \leftarrow s$ ;
8        $f^* \leftarrow f(s)$ ;
9     fim
10  fim
11   $s \leftarrow s^*$ ;
12  Retorne  $s$ ;
13 fim

```

3.3.1 Fase de Construção

Na fase de construção do Algoritmo GRASP (linha 2 do Algoritmo 2), uma solução é construída pela inserção dos itens, sendo um de cada vez. O pseudocódigo dessa fase está apresentado no Algoritmo 3.

Este procedimento se inicia na linha 3 do Algoritmo 3, com a construção de uma lista de itens candidatos C , que é formada pelo conjunto dos itens I que serão inseridos. Os elementos (itens) dessa lista são ordenados de forma decrescente, de acordo com uma função $g(t)$, que avalia o benefício do candidato ser incluído na solução. A função $g(\cdot)$ retorna, para cada item t da lista C , uma nota, que corresponde a altura deste item. A cada iteração da fase de construção, os k melhores itens da lista C , ou seja, os k itens mais altos, são colocados em uma Lista Restrita de Candidatos (*LRC*). O componente aleatório do método consiste em, para cada inserção na solução, selecionar aleatoriamente um item dentre os k itens candidatos da LRC. Em seguida,

atualizam-se as listas C e LRC , e repete-se o processo, até que todos os itens tenham sido incluídos na solução.

O encaixe dos itens da solução inicial gerada é feita utilizando-se uma técnica de encaixe, conforme a linha 3 do Algoritmo 3. Em seguida, é determinado o valor da função de avaliação desta solução inicial.

Algoritmo 3: Construção

Entrada: $g(\cdot)$, k, s
Saída: s

```

1 início
2    $s^* \leftarrow \emptyset$ ;
3   Inicia o conjunto  $C$  de itens candidatos;
4   Ordene o Conjunto  $C$  de acordo com  $g(\cdot)$ ;
5   enquanto  $C \neq \emptyset$  faça
6      $LRC = \{\text{conj. dos } k \text{ melhores itens de } C\}$ ;
7     Selecione, aleatoriamente, um item  $t \in LRC$ ;
8      $s^* \leftarrow s^* \cup \{t\}$ ;
9     Atualize o conjunto  $C$  de itens candidatos;
10  fim
11   $s \leftarrow$  Técnica de Encaixe  $s^*$ ;
12  Retorne  $s$ ;
13 fim

```

3.3.2 Fase de busca local

A fase de busca local utilizada é baseada na estrutura de vizinhança de uma dada solução. Esta vizinhança de s é o conjunto de soluções $N(s)$, em que cada solução $s \in 2N(s)$ é obtida a partir de um movimento feito na solução corrente s . Neste trabalho, para determinar a vizinhança de uma dada solução, é aplicado o movimento de troca de dois itens de s . Assim, a estrutura de vizinhança utilizada permuta cada item da solução corrente com os demais itens. Como exemplo, se a solução é $s = (5 \ 1 \ 3 \ 6 \ 2 \ 4)$, então $s' = (2 \ 1 \ 3 \ 6 \ 5 \ 4)$ é um vizinho de s , obtido pela troca de posição dos itens 5 e 2.

A fase de busca local do algoritmo proposto (linha 2 do Algoritmo 2) consiste na aplicação do procedimento de busca local *Best Improvement* (BI). Neste método, parte-se de uma solução s e, a cada iteração, são analisados todos os possíveis vizinhos, movendo-se somente para aquele que tiver o valor mais favorável da função de avaliação, isto é, a melhor solução é aquela cujo somatório das alturas dos itens que estão na primeira posição de cada faixa é menor. Dada uma solução, aplica-se uma técnica de encaixe, descrita na subseção 3.1, para determinar o valor da função de avaliação. O método para quando encontra um ótimo local. O Algoritmo 4 ilustra o pseudocódigo deste procedimento de busca local.

Para garantir a diversidade do conjunto de soluções, será utilizada apenas uma iteração do algoritmo GRASP, ou seja, $GRASP_{max} = 1$, para gerar cada indivíduo. Além disso, para garantir que a população inicial tenha uma boa qualidade em relação aos objetivos propostos, diferentes valores para o tamanho da lista restrita de candidatos LRC são utilizados. Dessa forma, os 70% da população construída pelo GRASP foram geradas de forma que: 10% com $k = 2$, 20% com $k = 4$, 20% com $k = 5$ e 20% com $k = 7$.

Algoritmo 4: Best Improvement

Entrada: $f(\cdot)$, s

Saída: s

```
1 início
2    $V = \{s' \in N(s) \mid f(s') < f(s)\}$ ; (Avalia toda a vizinhança de  $s$ )
3   enquanto  $V \neq \emptyset$  faça
4     Selecione  $s' \in V$ , sendo  $s' = \arg \min \{f(s') \mid s' \in V\}$ ;
5     se  $(f(s') < f(s))$  então
6        $s \leftarrow s'$ ;
7        $f(s) \leftarrow f(s')$ ;
8     fim
9      $V = \{s' \in N(s) \mid f(s') < f(s)\}$ ;
10  fim
11  Retorne  $s$ ;
12 fim
```

3.4 Função de Avaliação

Neste trabalho, são utilizadas, como função de avaliação, as duas funções objetivo do problema. Dessa forma, cada indivíduo recebe duas notas, que correspondem à altura utilizada do objeto e ao número de cortes realizado.

3.5 Operadores Genéticos

Para descrever os operadores genéticos utilizados neste trabalho será usada, como exemplo, uma solução com 6 itens.

3.5.1 Crossover OX

Este operador constrói os cromossomos filhos escolhendo, de um pai, uma sequência de corte, e preservando, do outro pai, a ordem de corte dos demais itens. Será usada uma definição deste operador de acordo com [Souza \(2009\)](#).

Considere os pais P_1 e P_2 abaixo e a escolha aleatória de dois pontos desses pais para fazer o corte. Para este exemplo, o corte será feito nas posições 3 e 4:

$$P_1 = [6 \ 1 \ | \ 3 \ 4 \ | \ 5 \ 2] \quad \text{e} \quad P_2 = [1 \ 2 \ | \ 4 \ 6 \ | \ 5 \ 3]$$

Os filhos F_1 e F_2 herdam a sequência de itens dos pais P_1 e P_2 , respectivamente, nas posições que foram feitos os cortes:

$$F_1 = [X \ X \ | \ 3 \ 4 \ | \ X \ X] \quad \text{e} \quad F_2 = [X \ X \ | \ 4 \ 6 \ | \ X \ X]$$

Partindo-se do segundo corte de P_1 , é criada uma lista com os n itens deste pai:

$$L_1 = 5 - 2 - 6 - 1 - 3 - 4.$$

De forma análoga, cria-se uma lista para P_2 , com a ordem de itens a partir do segundo corte:

$$L_2 = 5 - 3 - 1 - 2 - 4 - 6.$$

Os itens contidos nas listas L_1 e L_2 são inseridos nos filhos F_2 e F_1 , respectivamente, a partir do segundo corte, seguindo a sequência em que os itens se encontram nas listas. Os itens que já estão contidos nos filhos não são inseridos. Deste modo, os filhos formados são:

$$F_1 = [2 \ 6 \ 3 \ 4 \ 5 \ 1] \quad \text{e} \quad F_2 = [1 \ 3 \ 4 \ 6 \ 5 \ 2].$$

3.5.2 Mutação

O operador de mutação nos problemas que utilizam permutação deve garantir que o indivíduo obtido seja factível. Para fazer a mutação, são escolhidos p pontos de corte no cromossomo e, então, realiza-se a troca dos genes entre esses pontos. Para a mutação usada neste trabalho, foi escolhido $p = 2$.

Considere um indivíduo representado por uma sequência com seis números inteiros $S = [6 \ 1 \ 4 \ 3 \ 5 \ 2]$. Deste indivíduo, são escolhidas duas posições aleatórias, 2 e 5, por exemplo. O operador age trocando os genes dessas posições, criando-se, assim, um novo indivíduo $B = [6 \ 5 \ 4 \ 3 \ 1 \ 2]$.

3.5.3 Seleção

O operador de seleção é utilizado para selecionar os indivíduos que irão para a próxima geração. Neste trabalho, é utilizado o Método da Roleta. Este método é usado para selecionar X indivíduos para a próxima geração a partir de Y pais. Cada indivíduo recebe uma probabilidade de seleção, que pode ser calculada usando-se a seguinte fórmula:

$$Probabilidade_{seleo} = \frac{aptidao(individuo_i)}{\sum aptidao(individuo_i)}$$

Com o valor gerado pela função, é criada uma roleta, em que seus espaços são proporcionais à aptidão relativa de cada indivíduo. Assim, os indivíduos com alta aptidão terão maiores chances de participar do processo de criação da nova população.

4 ANÁLISE DOS RESULTADOS

Nesta seção são apresentados os resultados obtidos pela metodologia proposta para resolver o Problema de Corte com Dimensão Aberta multiobjetivo. Os algoritmos propostos para resolver o problema foram desenvolvidos utilizando o MATLAB R2009b, e os testes foram feitos em um PC Intel Core 2 Quad 2.67 GHz, com 4 GB de memória RAM, em ambiente Windows7.

4.1 Problemas testes

Para avaliar o método proposto para resolver o PCDA multiobjetivo são utilizadas duas classes de instâncias da literatura: os problemas-testes de Hopper e Turton (2001) e de Mumford-Valenzuela e Wang (2001).

Os problemas-testes de Hopper e Turton (ver [Hopper and Turton \(2001\)](#)), denotados por C_i^j estão divididos em sete categorias ($i = 1, \dots, 7$), sendo que cada categoria é subdividida em três problemas ($i = 1, 2, 3$), formando um total de 21 problemas como mostra a Tabela 1.

Os problemas-testes de Mumford-Valenzuela e Wang (ver [Mumford-Valenzuela and Wang \(2001\)](#)) são divididos em duas classes: *Nice* e *Path*. Essas duas classes são caracterizadas pelo conjunto dos itens possuírem dimensões reais, e as instâncias serem de tamanhos n , em que $n = 25, 50, 100, 200, 500$, como mostra a Tabela 2. No trabalho, serão utilizadas para os testes um problema para cada tamanho de cada classe.

Para resolver o PCDA multiobjetivo em que o corte é guilhotinado em dois estágios e a rotação dos itens não é permitida, é acoplado ao SPEA 2 quatro algoritmos de encaixe, formando um total de quatro algoritmos que para simplificar serão denotados por: SPEA2-FF; SPEA2-BF; SPEA2-FFDH e SPEA2-BFDH. Cada problema foi executado 30 vezes para cada um dos 21 problemas-testes e a mesma população inicial.

Table 1: Problemas-testes de Hopper e Turton (2001)

Categoria	Número de itens			Largura do objeto
	P1	P2	P3	
C1	16	17	16	20
C2	25	25	25	40
C3	28	29	28	60
C4	49	49	49	60
C5	73	73	73	60
C6	97	97	97	80
C7	196	197	196	160

Table 2: Problemas-teste de Mumford-Valenzuela e Wang (2001)

Categoria	Problemas-Testes	número de itens	Largura do objeto
Nice	P1	25	100
	P2	50	
	P3	100	
	P4	200	
	P5	500	
Path	P1	25	100
	P2	50	
	P3	100	
	P4	200	
	P5	500	

As figuras 3, 4, 5, 6, 7, 8 e 9 mostram os conjuntos de Pareto de cada uma das 21 instâncias de Hooper e Turton, obtidos para cada um dos quatro algoritmos utilizados. As figuras 10, 11, 12, 13 e 14 mostram os conjuntos de Pareto de cada um dos 10 problemas utilizados de Mumford-Valenzuela e Wang. Cada conjunto de Pareto, para cada um dos algoritmos, representa um dos conjuntos de Pareto obtido por uma das 30 execuções do algoritmo.

Deve-se ressaltar que, em todos os casos, evidencia-se o caráter multiobjetivo da formulação apresentada para o PCDA, uma vez que, conforme as figuras 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13 e 14, uma diminuição da altura utilizada corresponde em um aumento do número de cortes realizados, evidenciando-se, assim, a relação de compromisso entre as funções objetivo tratadas na versão multiobjetivo do problema.

Uma vez que não existem outros trabalhos publicados nos quais o PCDA em questão é do tipo OG, não é possível fazer uma comparação de resultados. Entretanto, no trabalho de Illich and While (2009), o PCDA multiobjetivo não guilhotinado, com e sem rotação dos itens, foi resolvido utilizando dois algoritmos multiobjetivo, um determinístico e um evolucionário. Apesar dos problemas tratados serem de naturezas distintas, é possível garantir que as soluções que resolvem o PCDA guilhotinado estão contidas no conjunto de soluções para o PCDA não guilhotinado em Illich and While (2009). Desta forma, nota-se que a metodologia aqui proposta é capaz de encontrar um conjunto de soluções não dominadas mais representativo, tanto em número de soluções quanto em diversidade e espalhamento das mesmas. Assim, a abordagem aqui apresentada se mostra eficiente para o tratamento da versão multiobjetivo do PCDA guilhotinado sem rotação dos itens.

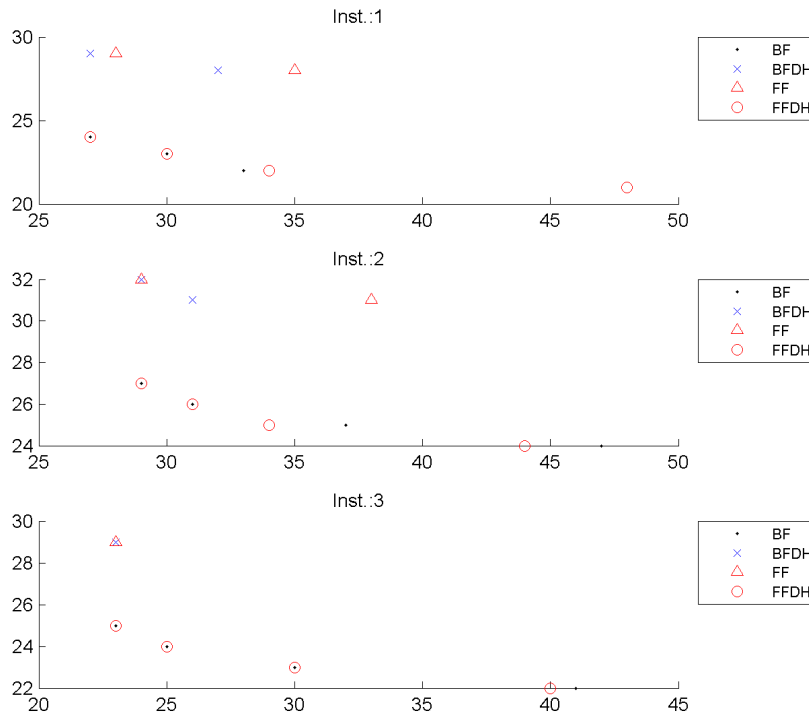


Figure 3: Conjunto de Pareto para as intâncias 1, 2 e 3 de Hopper e Turton.

5 CONCLUSÃO

Neste trabalho, o Problema de Corte com Dimensão Aberta guilhotinado e sem rotação dos itens, foi tratado, utilizando-se uma abordagem multiobjetivo. O problema proposto consiste em minimizar a altura do objeto e minimizar o número de cortes realizados. O problema foi solucionado utilizando-se uma aplicação do algoritmo genético multiobjetivo SPEA2 combinado com as heurísticas de encaixe *First Fit*, *Best-Fit*, *First Fit Decreasing Height* e *Best-Fit Decreasing Height*. Parte da população inicial dos algoritmos foi obtida através utilização da fase de construção do Algoritmo GRASP (Greedy Randomized Adaptative Search Procedure). A utilização do GRASP para construir a população inicial privilegia um conjunto de solução inicial de boa qualidade.

Foram utilizados quatros métodos, SPEA2-FF, SPEA2-BF, SPEA2-FFDH e SPEA2-BFDH, e os testes foram realizados em 21 problemas da literatura. O algoritmo multiobjetivo SPEA2 acoplado aos algoritmos de nível se mostrou uma metodologia eficiente na resolução da versão multiobjetivo do Problema de Corte com Dimensão Aberta, para todos os problemas testados. Foi possível evidenciar o caráter multiobjetivo da formulação apresentada para o PCDA, uma vez que, uma diminuição da altura utilizada corresponde a um aumento no número de cortes realizados. Assim, verificou-se que todos os métodos foram capazes de gerar um conjunto representativo de soluções não-dominadas, em que cada ponto deste conjunto representa uma solução de compromisso entre esses dois objetivos.

Uma vez que não existem trabalhos disponíveis na literatura que tratam do problema abordado, não é possível efetuar uma comparação com os resultados obtidos. Entretanto, novos testes estão sendo realizados para verificar qual das técnicas de encaixe em conjunto com o SPEA2 resulta na melhor curva de compromisso entre os objetivos tratados.

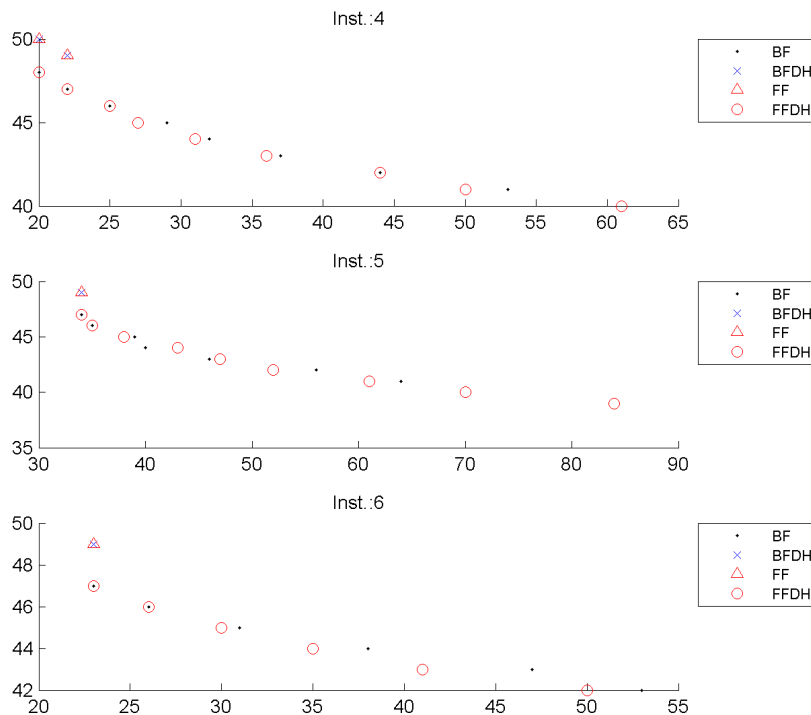


Figure 4: Conjunto de Pareto para as intâncias 4, 5 e 6 de Hopper e Turton.

AGRADECIMENTOS

Os autores agradecem ao CAPES, FAPEMIG, CNPq e CEFET-MG, pelo apoio ao desenvolvimento deste trabalho.

REFERENCES

- Coffman-Jr E.G., Garey M.R., and Johnson D.S. *Appears in Approximation Algorithms for NP-Hard Problems*, chapter Approximation Algorithms for Bin Packing: A Survey, pages 46–93. PWS Publishing, Boston, 1996.
- de Andrade M.S.F. *Algoritmos Evolutivos Mono e Multiobjetivos Para Problemas Bidimensionais de Corte*. Master's Thesis, Centro Federal de Educação Tecnológica de Minas Gerais, 2009.
- de Armas J., León C., Miranda G., and Segura C. Optimisation of a multi-objective two-dimensional strip packing problem based on evolutionary algorithms. *International Journal of Production Research*, 48:2011–2028, 2009.
- Dyckhoff H. and Finke U. *Cutting and Packing in Production and Distribution: A Typology and Bibliography*. Physica-Verlag HD, 1 edition, 1992.
- Feo T.A. and Resende M.G.C. Greedy randomized adaptive search procedures. *Journal of Global Optimization*, 9:109–134, 1995.
- Hifi M. Exact algorithms for the guillotine strip cutting/packing problem. *Computers & Operations Research*, 25:925–940, 1998.
- Holland J.H. *Adaptation in Natural and Artificial Systems*. Univ. of Michigan Press, 1975.
- Hopper E. and Turton B.C. An empirical investigation of meta-heuristic and heuristic algorithms for a 2d packing problem. *European Journal of Operations Research*, 16:257–300,

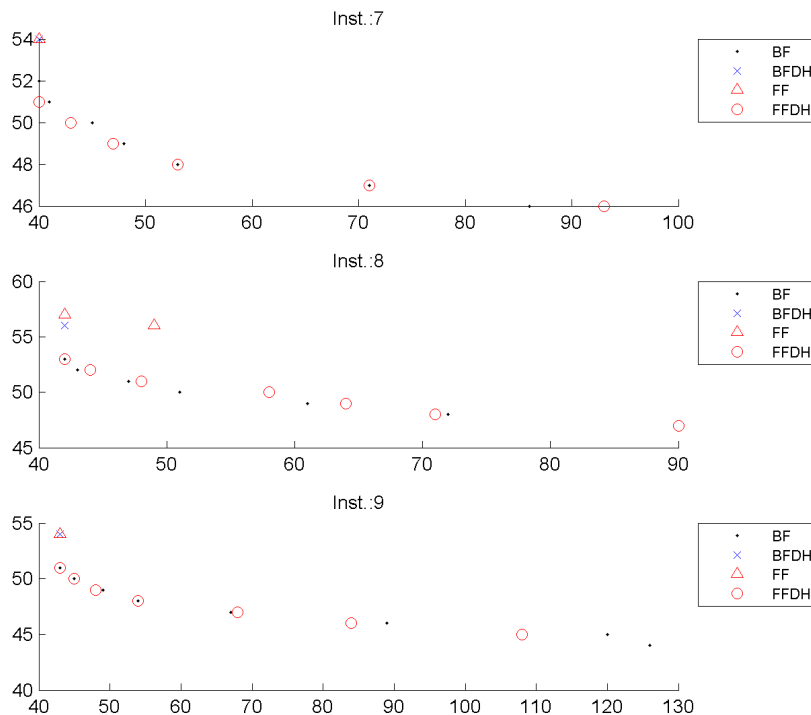


Figure 5: Conjunto de Pareto para as intâncias 7, 8 e 9 de Hopper e Turton.

2001.

- Illich S. *Multi-Objective Strip Packing*. Doutorado, School of Computer Science and Software Engineering, The University of Western Australia, 2007.
- Illich S. and While L. Multi-objective strip packing. *Journal of Advanced Research in Evolutionary Algorithms*, 1(1):1–26, 2009.
- Illich S., While L., and Barone L. Multi-objective strip packing using an evolutionary algorithm. *IEEE Congress on Evolutionary Computation (CEC 2007)*, pages 4207–4214, 2007.
- Johnson D.S. Fast algorithms for bin packing. *Journal of Computer and System Sciences*, 8(3):272 – 314, 1974.
- Liu D. and Teng H. An improved bl-algorithm for genetic algorithm of the orthogonal packing of rectangles. *European Journal of Operations Research*, 112:413–420, 1999.
- Lodi A. *Algorithms for Two-Dimensional Bin Packing and Assignment Problems*. Doutorado, UNIVERSITÀ DEGLI STUDI DI BOLOGNA, 1999.
- Lodi A., Martello S., and Monaci M. Two-dimensional packing problems: A survey. *European Journal of Operations Research*, 141:241–252, 2002.
- Lodi A., Martello S., and Vigo D. Heuristic and metaheuristic approaches for a class of two-dimensional bin packing problems. *INFORMS Journal on Computing*, 11:345–357, 1999.
- Lodi A., Martello S., and Vigo D. Models and bounds for two-dimensional level packing problems. *Journal of Combinatorial Optimization*, 8:363–379, 2004.
- Miranda G., de Armas J., Segura C., and León C. Hyperheuristic codification for the multi-objective 2d guillotine strip packing problem. *IEEE Congress on Evolutionary Computation (CEC)*, pages 1–8, 2010.
- Mumford-Valenzuela C.L. and Wang P.Y. Heuristics for large strip packing problems with

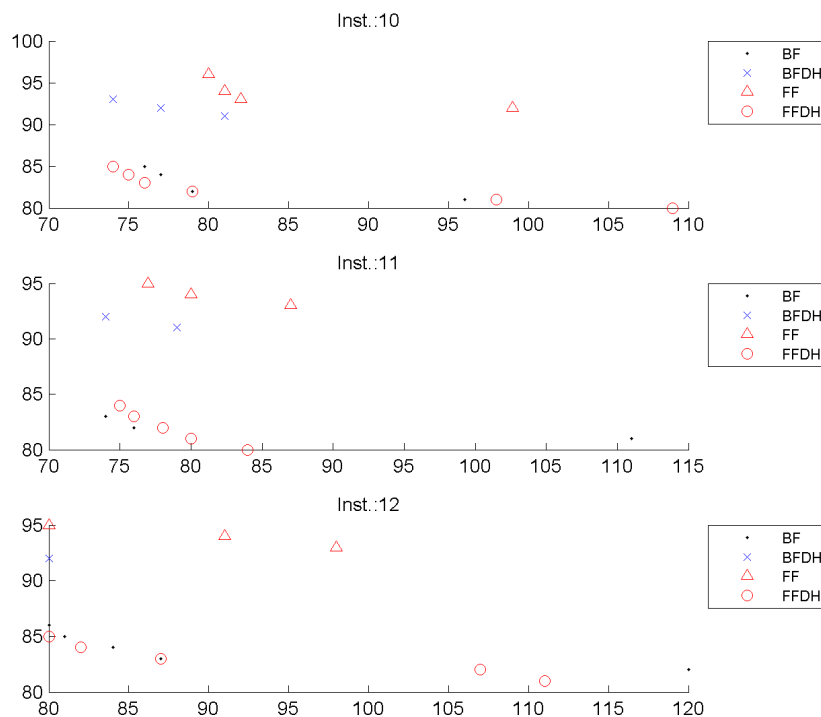


Figure 6: Conjunto de Pareto para as instâncias 10, 11 e 12 de Hopper e Turton.

guillotine patterns: an empirical study. *Proceedings of the 4Th Metaheuristics International Conference*, pages 417–421, 2001.

Ntene N. *An Algorithmic Approach to the 2D Oriented Strip Packing Problem*. Doutorado, Department of Logistics, University of Stellenbosch, South Africa, 2007.

Ortmann F.G. *Heuristics for Offline Rectangular Packing Problems*. Doutorado, Department of Logistics, Stellenbosch University, 2010.

Souza M.J.F. *Otimização Combinatória: Notas de aula*. Departamento de Computação, Universidade Federal de Ouro Preto, 2009.

Tiwari S. and Chakraborti N. Multi-objective optimization of a two-dimensional cutting problem using genetic algorithms. *Journal of Materials Processing Technology*, 173:384–393, 2006.

Wäscher G., HauBner H., and Schumann H. An improved typology of cutting and packing problems. *European Journal of Operations Research*, 183:1109–1130, 2007.

Zitzler E., Laumanns M., and Thiele L. Spea2: Improving the strength pareto evolutionary algorithm for multiobjective optimization. *Evolutionary Methods for Design, Optimization and Control*, page 6, 2002.

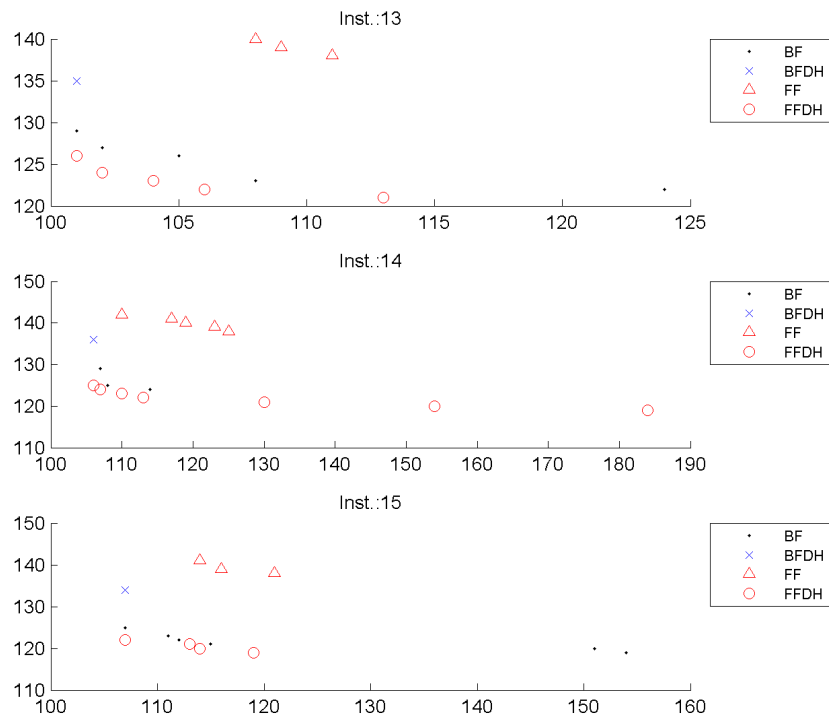


Figure 7: Conjunto de Pareto para as intâncias 13, 14 e 15 de Hopper e Turton.

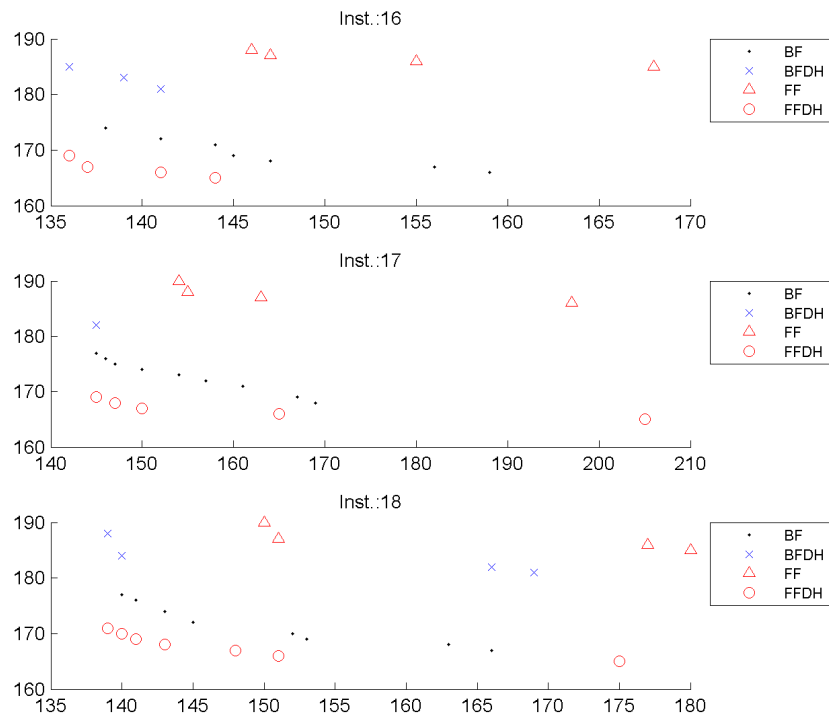


Figure 8: Conjunto de Pareto para as intâncias 16, 17 e 18 de Hopper e Turton.

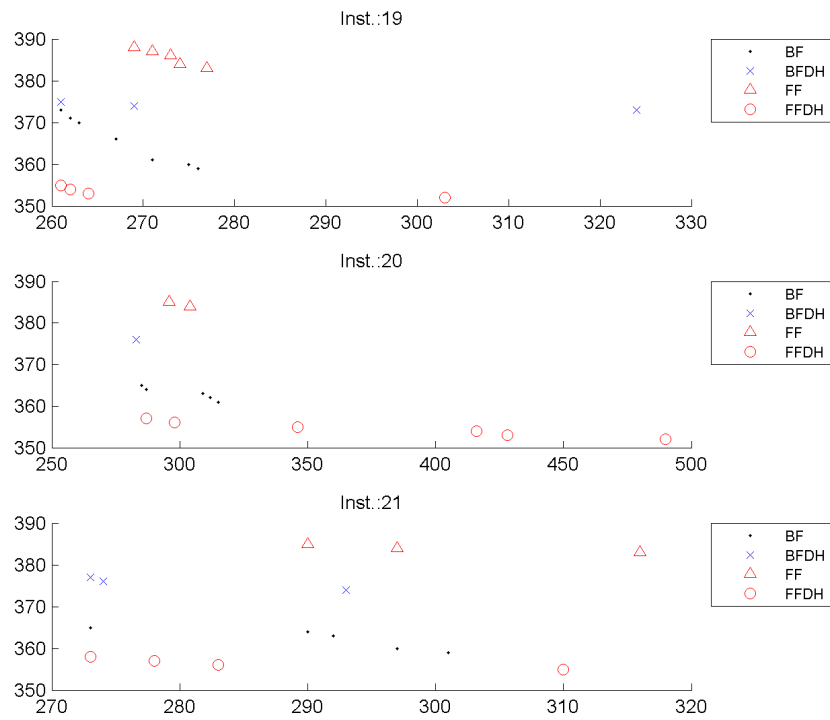


Figure 9: Conjunto de Pareto para as intâncias 19, 20 e 21 de Hopper e Turton.

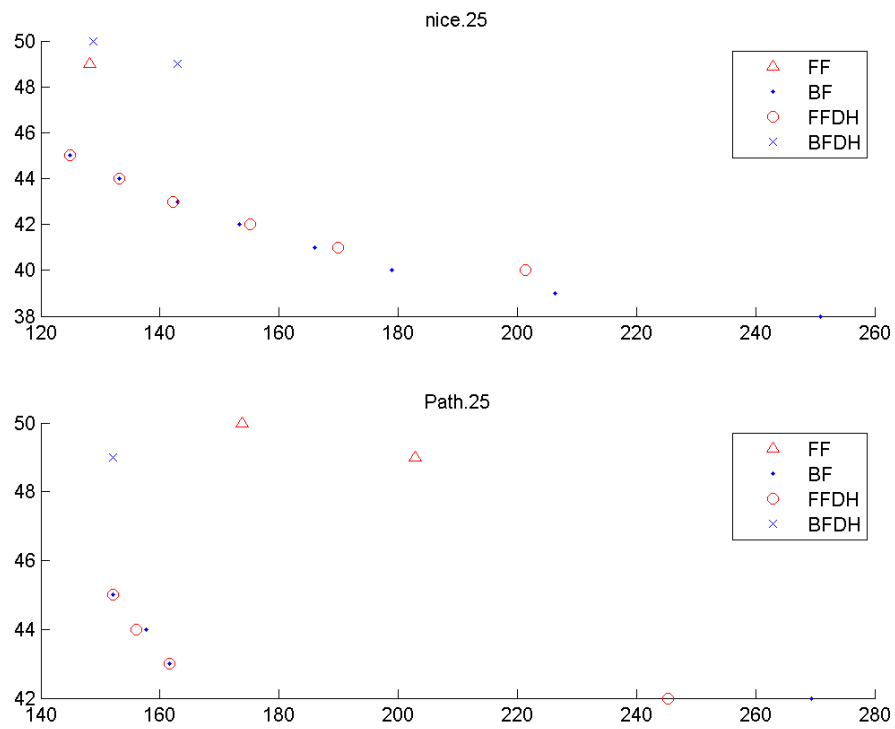


Figure 10: Conjunto de Pareto para as intâncias nice.25 e path.25 de Mumford-Valenzuela e Wang.

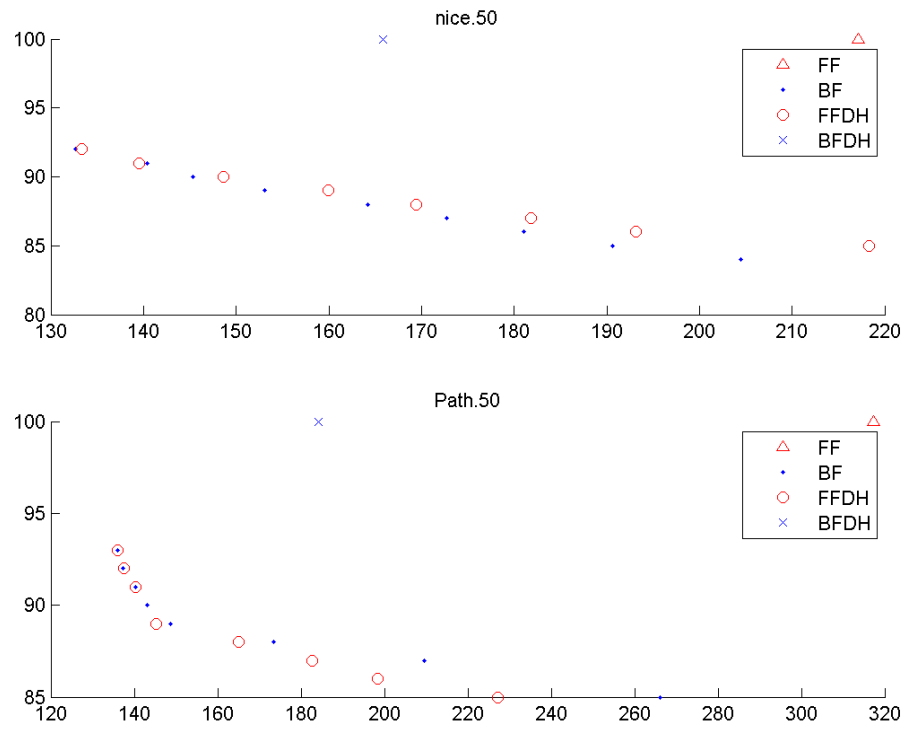


Figure 11: Conjunto de Pareto para as intâncias nice.50 e path.50 de Mumford-Valenzuela e Wang.

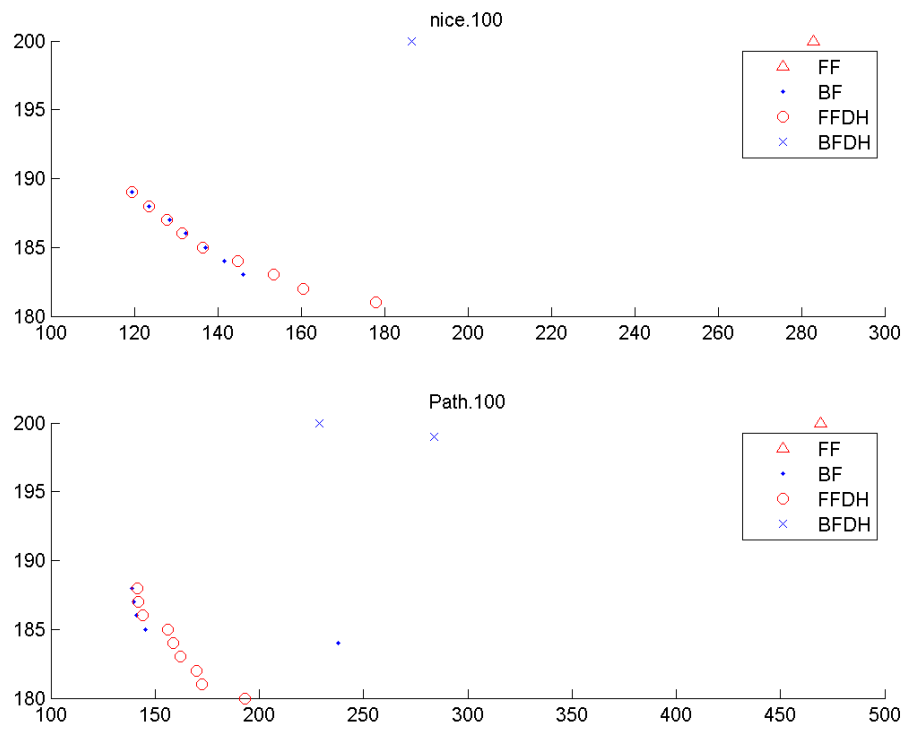


Figure 12: Conjunto de Pareto para as intâncias nice.100 e path.100 de Mumford-Valenzuela e Wang.

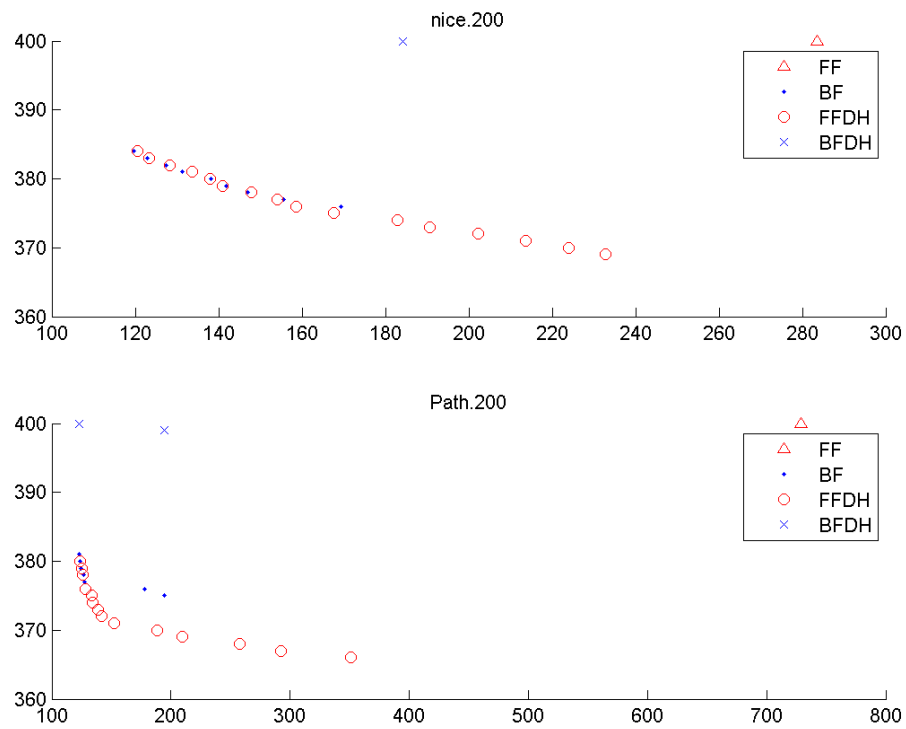


Figure 13: Conjunto de Pareto para as intâncias nice.200 e path.200 de Mumford-Valenzuela e Wang.

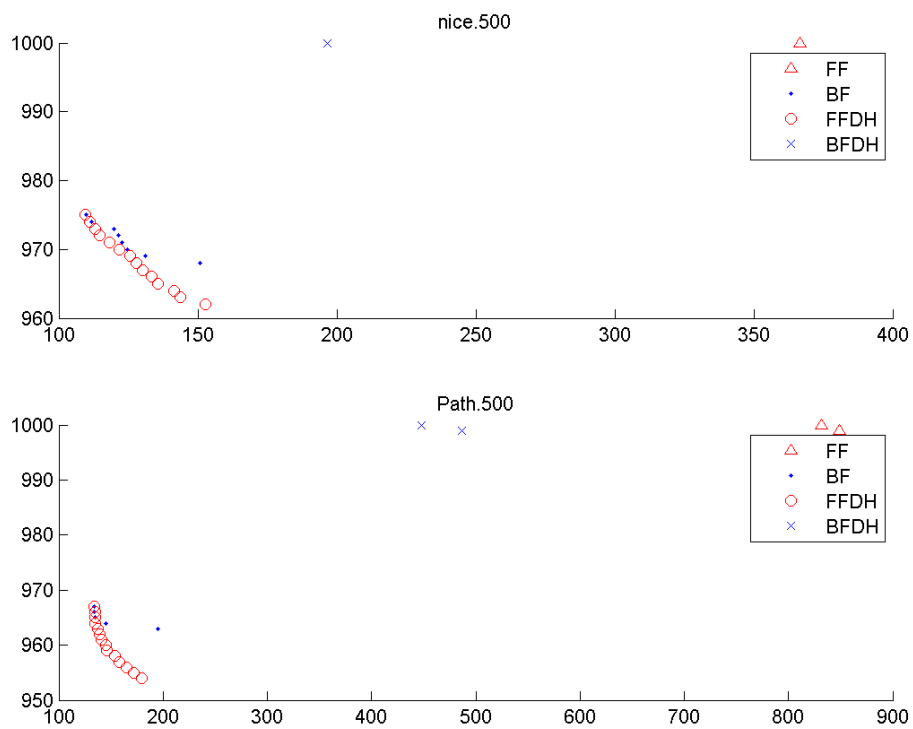


Figure 14: Conjunto de Pareto para as intâncias nice.500 e path.500 de Mumford-Valenzuela e Wang.