# Problema de Corte de Estoque Unidimensional

Lucas Fernandes Nogueira
Orientadora: Prof.ª Dr.ª Adriana Cristina Cherri

# Introdução

# Introdução

- Contexto histórico
  - Revolução industrial
- Problemas de otimização
  - Pesquisa operacional
- Obter itens demandados a partir de barras maiores

# Classificação quanto à dimensão

- Número de dimensões relevantes no processo de cortagem
  - Unidimensional
  - Bidimensional
  - Tridimensional
  - 1.5-dimensional
  - 2.5-dimensional
  - Multidimensional

# Problema

# Problema

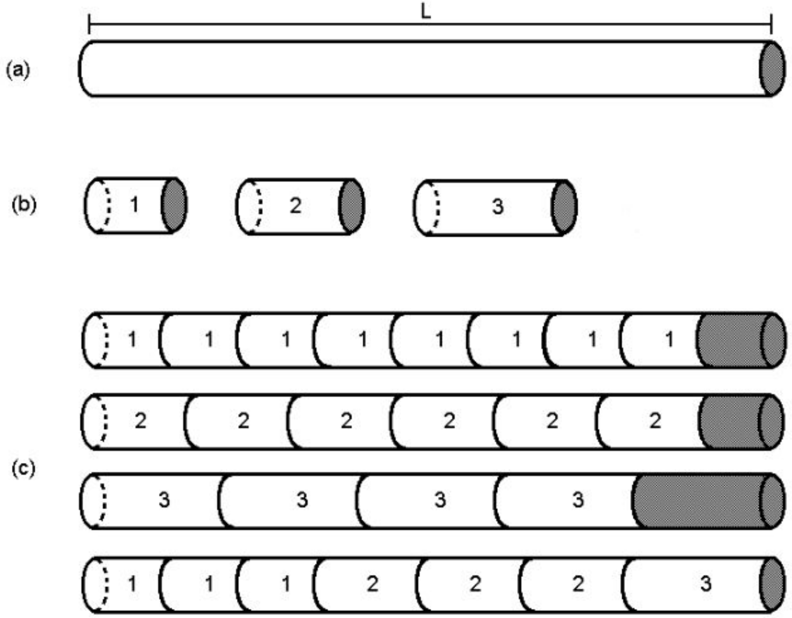
- Conjunto de barras de tamanho $L_1$, $L_2$, …, $L_k$ disponíveis em estoque
- Produzir itens menores em quantidades encomendadas
- Minimizar um objetivo
- Encomenda: demanda por $d_i$ itens de tamanho $l_i$ $\quad\quad$ i = 1, 2, …, m

# Padrão de corte

- Meio de se cortar determinada barra em estoque
- Representação:
  - $a = (\alpha_1, \alpha_2, \ldots, \alpha_m)$
  - $\alpha_i$ representa a quantidade de itens do tipo *i* no padrão de corte
- Definição: padrão de corte homogêneo
  - Produz apenas um item

# Representação do problema

Figura 1 - (a) Barra em estoque; (b) Itens demandados; (c) padrões de corte




Fonte: (POLDI, 2003)

# Objetivos

# Objetivos

- Objetivos gerais
  - Entender o PCE;
  - Entender e implementar o modelo matemático através do método simplex com geração de colunas e
  - Entender e implementar procedimentos heurísticos
- Objetivos específicos
  - Entender o modelo matemático que representa o problema de corte de estoque;
  - Implementar o modelo matemático relacionado ao PCE na linguagem C++ utilizando o solver CPLEX;
  - Implementar um procedimento heurístico para a obtenção de soluções inteiras;
  - Realizar testes computacionais.

# Modelo Matemático

# Modelo básico

minimizar $f(x) = \sum_{j=1}^{n} x_j$

sujeito a:

$Ax = d$

$x \geq 0$ e inteiro

$A \in R^{m \times n}$

# Modelo para vários tipos de barras em estoque

Dados de estoque:

$K$: número de barras em estoque

$L_k$: dimensão da barra $k$

Dados de demanda:

$m$: número de tipos de itens

$l_i$: dimensão do item tipo $i$, $i = 1, 2, \ldots, m$

$d_i$: demanda do item tipo $i$, $i = 1, 2, \ldots, m$

# Modelo para vários tipos de barras em estoque

- Padrões de corte definidos para cada barra em estoque

  $l_1\alpha_{1k} + l_2\alpha_{2k} + \ldots + l_m\alpha_{mk} \leq L_k$

  $\alpha_{ik} \geq 0$ e inteiro, $i = 1, \ldots, m$ e $k = 1, \ldots, K$

# Modelo para vários tipos de barras em estoque

- Considerando que há $N_k$ padrões de corte para cada barra *k*:

$$\text{minimizar } \mathrm{f}(x_{1_1}, x_{1_2}, \ldots) = \sum_{j=1}^{N_1} c_{j_1} x_{j_1} + \sum_{j=1}^{N_2} c_{j_2} x_{j_2} + \ldots + \sum_{j=1}^{N_k} c_{j_k} x_{j_k}$$

sujeito a

$$\sum_{j=1}^{N_1} a_{j_1} x_{j_1} + \sum_{j=1}^{N_2} a_{j_2} x_{j_2} + \ldots + \sum_{j=1}^{N_k} a_{j_k} x_{j_k} = d$$

$$x_{j_k} \geq 0 \text{ e inteiro, j = 1, ..., } N_k \text{, k = 1, ..., K}$$

# Método Simplex com Geração de Colunas

# Método Simplex

- Problema

  maximizar $f(x) = c^T x$
  sujeito a
  $Ax = d, x \geq 0$
- $A \in R^{mxn}$, $posto(A) = m$
- Partição básica $A = [B, N]$ e $c^T = [c_B, c_N]^T$

  maximizar $f(x) = c_B^T x_B + c_N^T x_N$
  sujeito a:
  $Bx_B + Nx_N = d$
  $x_B \geq 0, x_N \geq 0$

# Método Simplex - Solução

$$Bx_B + Nx_N = d \Leftrightarrow x_B = B^{-1}d - B^{-1}Nx_N$$

- Solução básica

$$x_N^o = 0 \Rightarrow x_B^o = B^{-1}d$$

- Vetor multiplicador simplex (vetor das variáveis duais) $\pi \in R^m$

$$\pi = c_B^T B^{-1}d$$

# Método Simplex - Estratégia

- Perturbação de uma componente de $x_N$
- $c_k - \pi^T a_k < 0$
- $x_k = \varepsilon \geq 0$ e $x_j = 0$ para as demais componentes
- Pode-se escolher qualquer componente $k$ com custo reduzido < 0
- Regra de Dantzig
    - Escolher menor custo reduzido
    - Analisar todas as colunas de A
        - Inviável para o PCE
- Geração de colunas
    - Gerar a coluna $j$ cujo valor $c_j - \pi a_j$ *seja mínimo*

# Geração de Colunas

- Requer apenas uma solução básica inicial
- Trata as variáveis não-básicas de forma implícita
  - Fase de determinação de vetor a entrar na base
  - Problema da mochila

# Problema da Mochila

- Barra a ser cortada de tamanho $L$
- Itens de tamanho $l_1$, $l_2$, …, $l_m$

$$maximizar\ f(x) = v_1\alpha_1 + v_2\alpha_2 + \ldots + v_m\alpha_m$$

*sujeito a*

$$l_1\alpha_1 + l_2\alpha_2 + \ldots + l_m\alpha_m \leq L$$

$$\alpha_i \text{ inteiro}, i = 1, \ldots, m$$

$$0 \leq \alpha_i \leq d_i, i = 1, \ldots, m$$

# Problema da Mochila no PCE

- Valor de utilidade do item *i*
  - *i*-ésimo multiplicador simplex
- Restrito quando um padrão de corte deve poder ser utilizado ao menos uma vez
- Um para cada tamanho de barra em estoque
  - Menor objetivo entra na base

# Heurísticas

# Heurísticas

- Solução fracionária pelo método simplex
    - Relaxação da restrição de integralidade
    - Inviável
- Heurísticas de construção
- Heurísticas residuais

# Heurísticas de Construção

# Heurísticas de Construção - Algoritmo Geral

1. Construa um bom padrão de corte;
2. Use o padrão do passo 1 tanto quanto for possível, sem gerar excessos de itens cortados;
3. Atualize a demanda.
4. Se a demanda for não nula, volte ao passo 1. Caso contrário, fim do algoritmo.

# Heurística FFD

- Utilizar o maior item demandado o máximo possível em um padrão de corte

1. Ordene os itens demandados de acordo com seus tamanhos, em ordem decrescente
2. Faça $r = d$ (demanda residual) e $D = r_1 + r_2 + \ldots + r_m$ (demanda total)
3. Enquanto $D > 0$, faça:
    a. *Encontre o padrão de corte de menor custo*
    b. Faça $x = \infty$
    c. Para cada item $i$, faça:
        i. $r_i = r_i - \alpha_i$
        ii. $D = D - \alpha_i$
        iii. $x = min(x, floor(r_i / \alpha_i), r_i \neq 0, \alpha_i \neq 0$
    d. *Armazene o padrão de corte e sua frequência x*

# Heurística FFD - Padrão de Corte

- Dado o tamanho da barra em estoque *L*

1. Faça Resto = L
2. Para cada item *i* faça, se *Resto* ≥ $l_m$:
   a. $\alpha_i = min(floor(Resto / l_i), r_i)$
   b. $Resto = Resto - \alpha_i l_i$
3. *Padrão de corte resultante* $\alpha = (\alpha_1, \alpha_2, ..., \alpha_m)$

# Heurística Gulosa

- Gerar um bom padrão de corte
  - HINXMAN (1980)
    - Problema da mochila
    - Valor de utilidade do item *i* igual a $l_i$
- *Utilizá-lo à exaustão, sem que haja excessos*

# Heurística Gulosa - Algoritmo

1. Faça $r = d$
2. Gere o padrão de corte resolvendo o problema da mochila restrito
3. Faça $x = \infty$
4. Para cada item $i$, faça:
    a. $x = min(x, floor(r_i / \alpha_i)),\ r_i \neq 0,\ \alpha_i \neq 0$
5. Para cada item $i$, faça:
    a. $r_i = r_i - x\alpha_i$
6. Armazene o padrão de corte e sua frequência $x$
7. Se r = 0 então pare
8. Senão, volte ao passo 2

# Heurísticas Residuais

# Heurísticas Residuais

- Geram uma solução inteira a partir da solução fracionária obtida pelo método simplex com geração de colunas
  - Seja $x$ a solução fracionária
  - Considere $y$ uma aproximação inteira para $x$ tal que $Ay \leq d$
    - Arredondamento comum: por truncamento $y = (floor(x_1), floor(x_2), \ldots, floor(x_n))$
  - Demanda residual $r = d - Ay$
  - Se há demanda residual, resolve novo problema com a mesma
  - Critério de parada comum: $y$ não alterar $r$ $(y = 0)$
  - Se após a parada ainda houver $r$, resolve com outra heurística

# Heurísticas Residuais - Algoritmo

1. **Faça** $r = d$
2. **Faça**:
   a. Resolve o problema residual relaxado utilizando gerador de colunas restrito, obtendo $\boldsymbol{x}$ e $\boldsymbol{B}$
   b. Determine a solução inteira aproximada $\boldsymbol{y}$
   c. Guarde $\boldsymbol{B}$ e $\boldsymbol{y}$
   d. **Faça** $r = r - By$
   e. **Se** algum critério de parada for satisfeito, **então**
      i. pare
3. **Se** $r \neq 0$, **então**
   a. aplique uma heurística para satisfazer $r$

# Heurística Residual por Truncamento

- Passo 2b
  - Arredondamento: por truncamento
- Passo 2e
  - Critério de parada: $y = 0$
- Passo 3a
  - Heurística de construção (Gulosa ou FFD)
    - Heurística Residual Gulosa
    - Heurística Residual FFD

# Heurística Residual *Nova*

- Apostar na qualidade dos padrões obtidos pelo modelo de otimização linear
- A cada iteração:
    - Resolve-se um PCE relaxado
    - Ordena-se o vetor solução de forma não-crescente
    - O padrão 1 (o mais utilizado), merece maior atenção
    - Para cada padrão, fazemos:
        - Sua frequência é arredondada para o inteiro superior e testa-se a factibilidade da solução $S = (y_1, 0, 0, \ldots, 0)$ (não gerar excessos)
        - Se não for factível, reduz frequência de 1 unidade até que se encontre uma solução que não viole a demanda
    - Atualizar demanda

# Heurística Residual *Nova* - Algoritmo

- Passo 2b do algoritmo base das heurísticas residuais

1. Ordene os padrões tal que $x_1 \geq x_2 \geq \ldots \geq x_m$
2. **Faça** y = 0
3. **Para cada** padrão de corte *j* **faça**
   a. $y_j = floor(x_j) + 1$
   b. **Enquanto** *By* > *d* faça (solução infactível)
      i. $y_i = y_i - 1$

# Heurística Residual *Nova - versão 2*

- Difere da versão original na ordenação dos padrões de corte
  - Ordem não-crescente de perda no padrão
  - Em caso de perdas iguais, o padrão com maior frequência $x_j$ possui prioridade

# Heurística Residual $Nova_p$

- Difere das versões originais por utilizar outro critério de arredondamento
- Parâmetro *p* relacionado ao desperdício de um padrão de corte
  - Define se a sua frequência será arredondada para cima ou para baixo
  - *p* pode ser um valor fixo ou um valor que depende do problema
    - *p = menor item demandado*
    - *p = média dos comprimentos dos itens*
- Critério de ordenação: menor perda

# Heurística Residual $Nova_p$ - Algoritmo

- Passo 2b do algoritmo base das heurísticas residuais

1. Ordene os padrões tal que $x_1 \geq x_2 \geq \ldots \geq x_m$
2. **Faça** y = 0
3. **Para cada** padrão de corte *j* **faça**
    a. $c_j = L_j\, x_j\, (l_1 + l_2 + \ldots + l_m)$ (perda no padrão *j*, com $L_j$ sendo o tamanho da barra relacionada)
    b. **Se** $c_j < p$ **então**
        i. $y_j = floor(x_j) + 1$
    c. **Senão**
    d. $y_j = floor(x_j)$
    e. **Enquanto** $By > d$ **faça** (solução infactível)
        i. $y_i = y_i - 1$
4. **Se** y = 0 **então**
    a. **Faça** $p = min(c_j,\ j = 1,\ \ldots,\ n)$ e volte ao passo 3

# Heurística Residual *Nova*$_p$ - Exemplo

- $L = 10$ e $p = 2$

Tabela 1 - Dados do exemplo de PCE para a heurística

| Item | Comprimento | Demanda |
|---|---|---|
| 1 | 5 | 49 |
| 2 | 7 | 155 |
| 3 | 9 | 177 |

Fonte: Produzido pelo autor

# Heurística Residual *Nova*$_p$ - Exemplo

Tabela 2 - Solução relaxada obtida pelo método simplex com geração de colunas

| Quantidade de barras | Perda no padrão | Padrão de corte |
|---|---|---|
| 24,5 | 0 | (2, 0, 0) |
| 177 | 1 | (0, 0, 1) |
| 155 | 3 | (0, 1, 0) |

Fonte: Produzido pelo autor

# Heurística Residual $Nova_p$ - Exemplo

- Ordenando o vetor solução
  - $x_1$ (perda = 0), $x_2$ (perda = 1), $x_3$ (perda = 3)
- Arredondando $x_1$
  - $c_1 = 0 < p \Rightarrow y_1 = 25$
  - $y_1 = 24$
- Arredondando $x_2$
  - $c_2 = 1 < p \Rightarrow y_2 = 178$
  - $y_2 = 177$
- Arredondando $x_3$
  - $c_3 = 3 > p \Rightarrow y_3 = 155$

# Heurística Residual $Nova_p$ - Exemplo

Tabela 3 - Solução inteira obtida pela heurística $Nova_p$

| Quantidade de barras | Padrão de corte |
|---|---|
| 24 | (2, 0, 0) |
| 177 | (0, 0, 1) |
| 155 | (0, 1, 0) |

Fonte: Produzido pelo autor

# Heurística Residual $Nova_p$ - Exemplo

Tabela 4 - Dados do problema residual

| Item | Comprimento | Demanda |
|---|---|---|
| 1 | 5 | 1 |
| 2 | 7 | 0 |
| 3 | 9 | 0 |

Fonte: Produzido pelo autor

# Heurística Residual $Nova_p$ - Exemplo

Tabela 5 - Solução relaxada do problema residual

| Quantidade de barras | Perda no padrão | Padrão de corte |
|---|---|---|
| 1 | 5 | (1, 0, 0) |

Fonte: Produzido pelo autor

# Heurística Residual $Nova_p$ - Exemplo

Tabela 6 - Solução final do exemplo

| Quantidade de barras | Padrão de corte |
|---|---|
| 24 | (2, 0, 0) |
| 177 | (0, 0, 1) |
| 155 | (0, 1, 0) |
| 1 | (1, 0, 0) |

Fonte: Produzido pelo autor

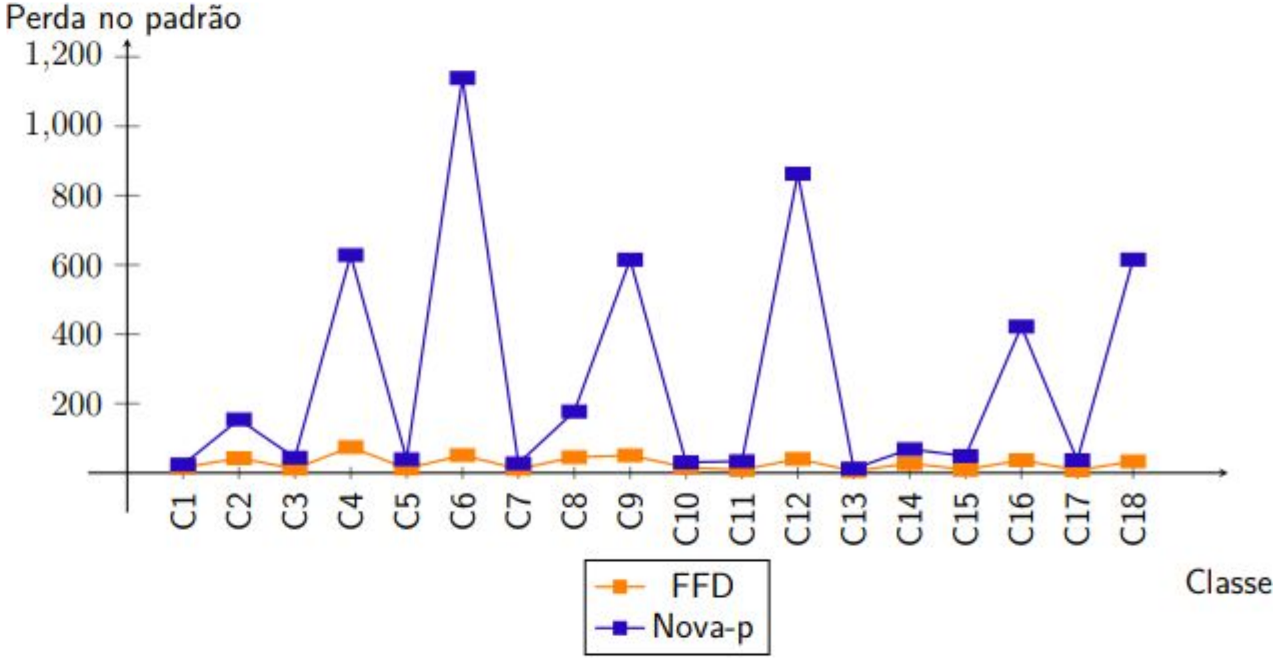
# Análise de Resultados

# Gerador Aleatório

- *$K$ = 3, 5 e 7*
- *$L_k$ no intervalo [1, 100]*
- *$m$ = 5, 20 e 40*
- *$l_i$ no intervalo [1, $vL$]*
  - *$v$ = 0,2 ou 0,8*
  - *$L$ = média entre os valores $L_k$*
  - *$d_i$ no intervalo [1, 10]*
- *18 classes de problemas com 20 exemplos cada*

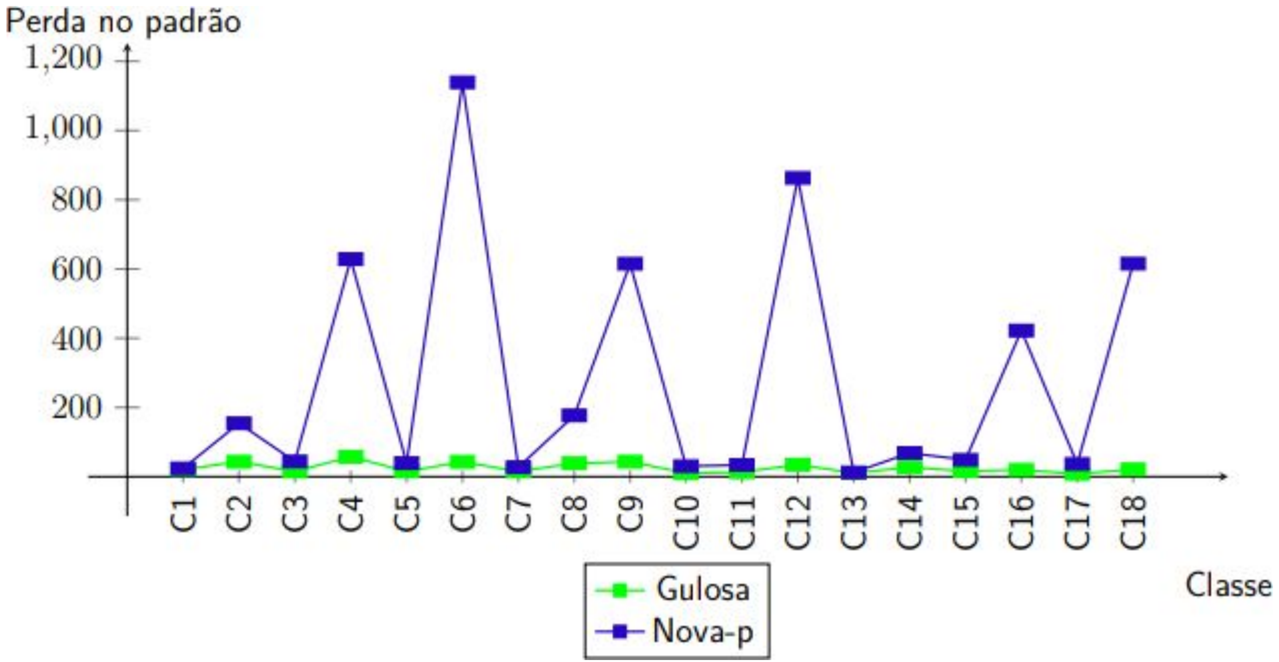
# Análise de Resultados

Figura 2 - Perda média: *Heurística FFD* e $Nova_p$



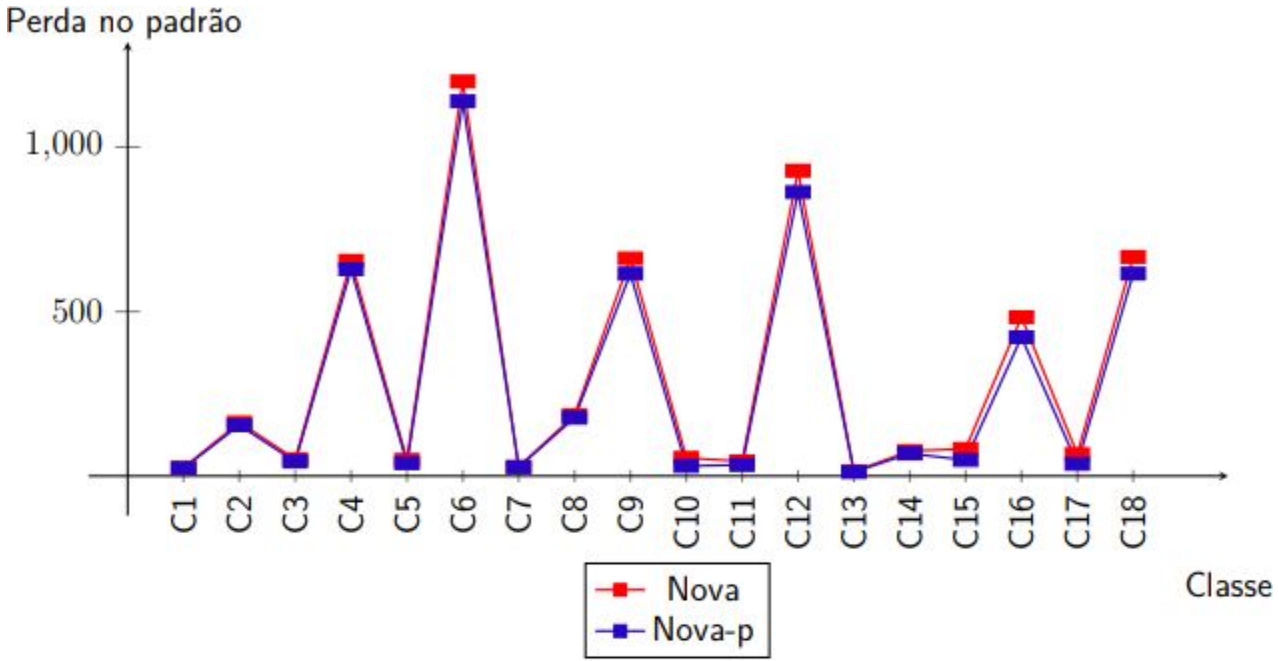
# Análise de Resultados

Figura 3 - Perda média: *Heurística Gulosa* e *Nova*$_p$



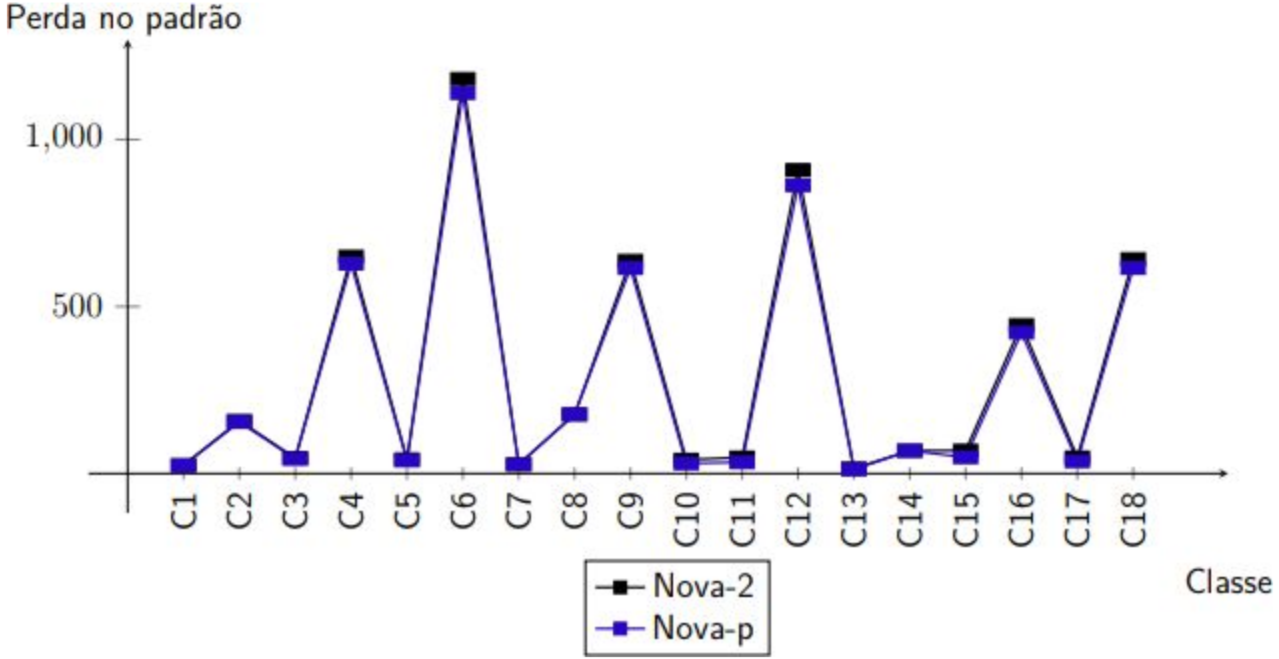
# Análise de Resultados

Figura 4 - Perda média: *Heurística Nova* e *Nova$_p$*



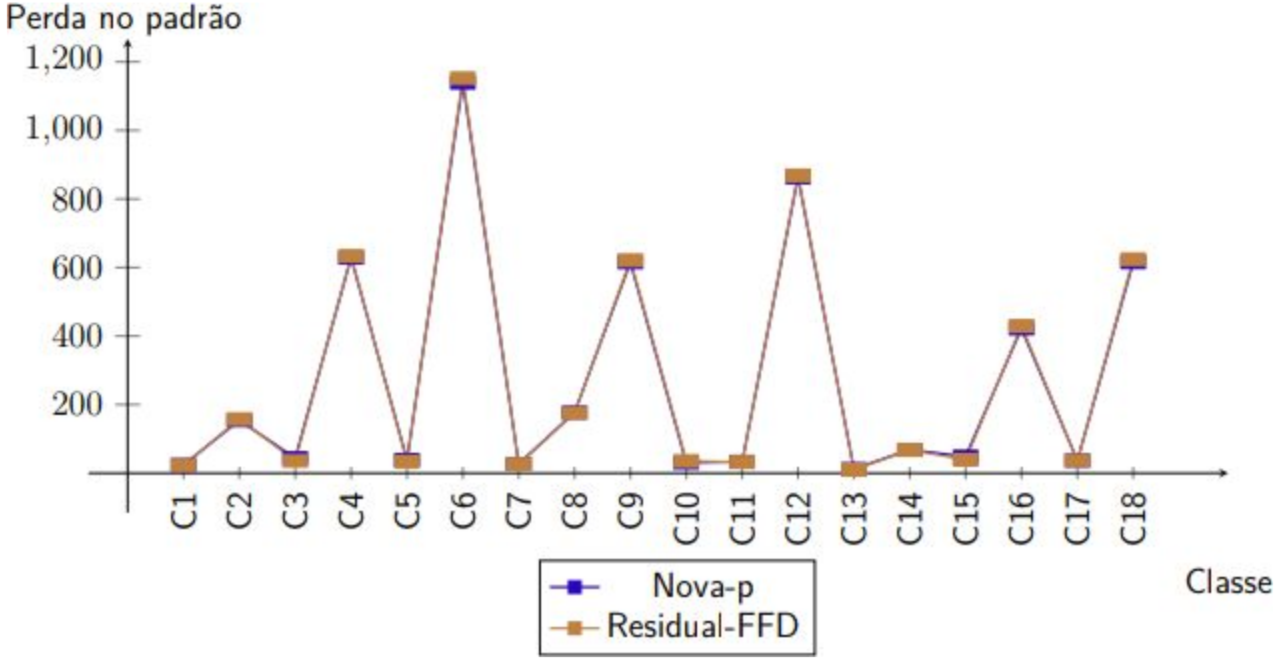
# Análise de Resultados

Figura 5 - Perda média: *Heurística Nova - versão 2* e $Nova_p$



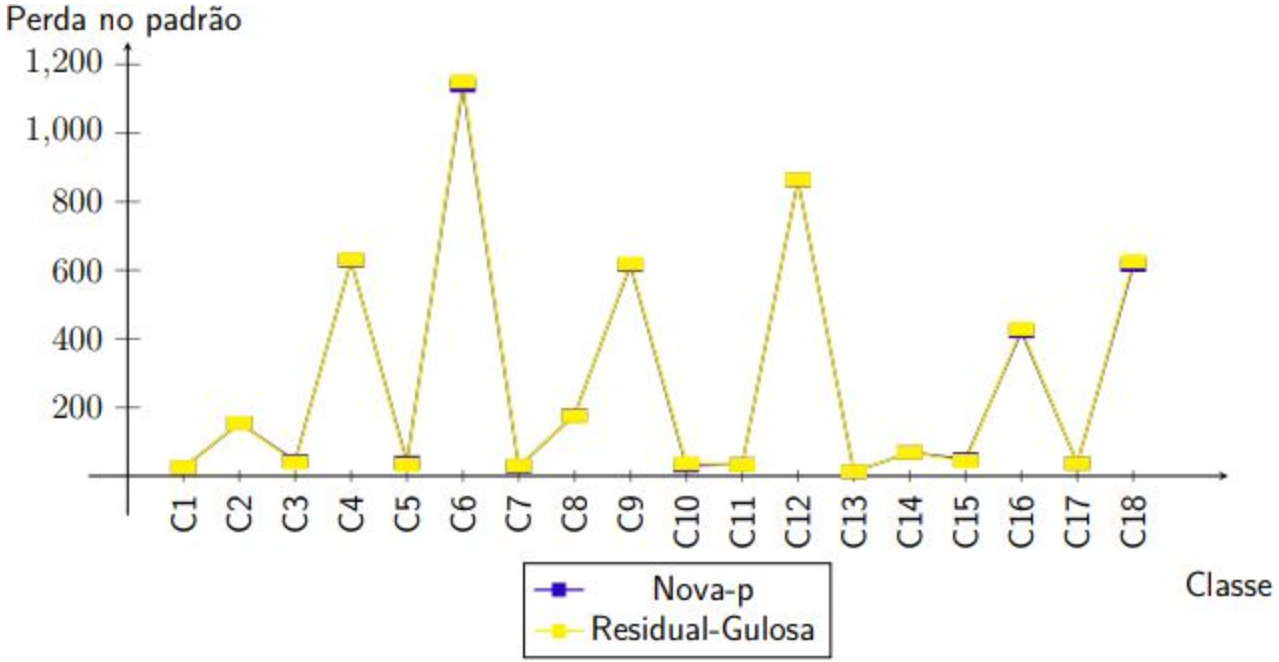
# Análise de Resultados

Figura 6 - Perda média: *Heurística Residual FFD* e $Nova_p$



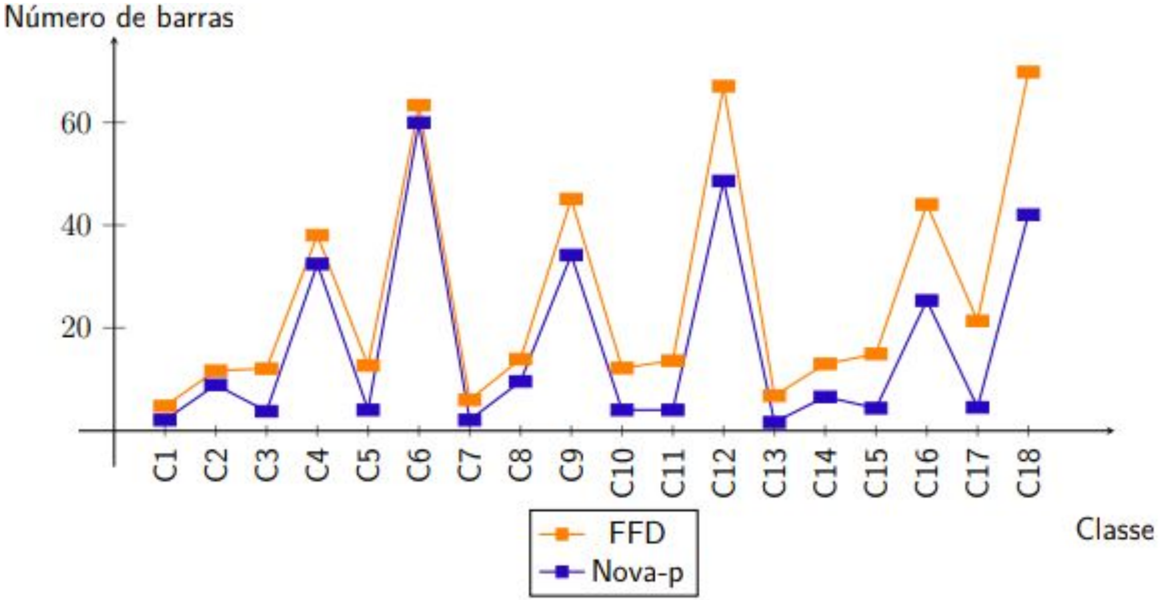
# Análise de Resultados

Figura 7 - Perda média: *Heurística Residual Gulosa* e $Nova_p$



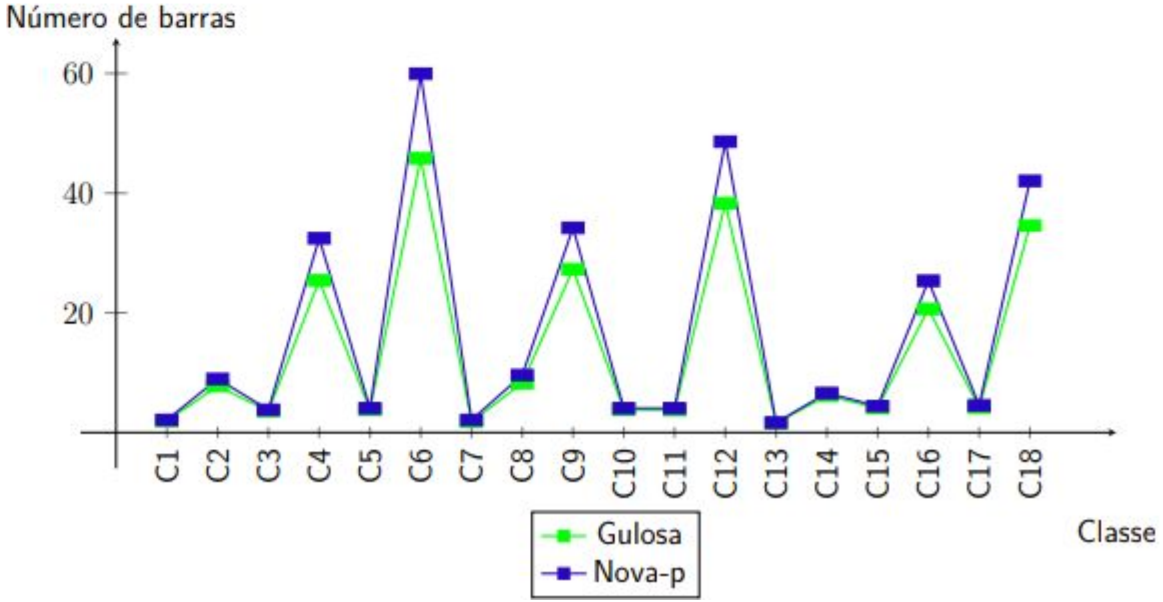
# Análise de Resultados

Figura 8 - Número de barras cortadas médio: *Heurística FFD* e $Nova_p$



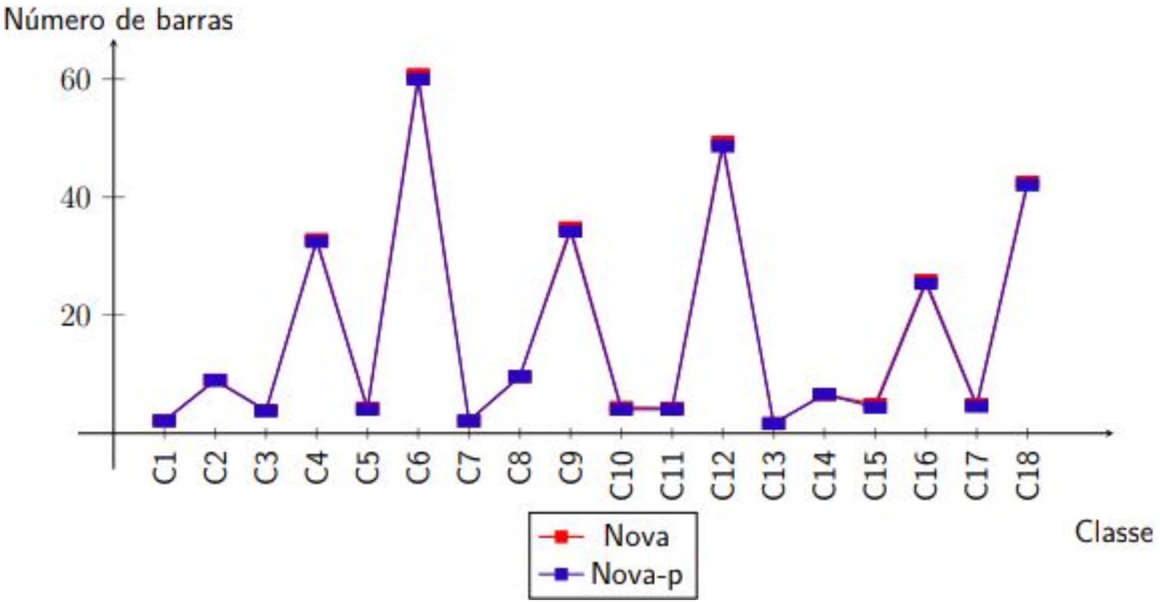
# Análise de Resultados

Figura 9 - Número de barras cortadas médio: *Heurística Gulosa* e *Nova*$_p$



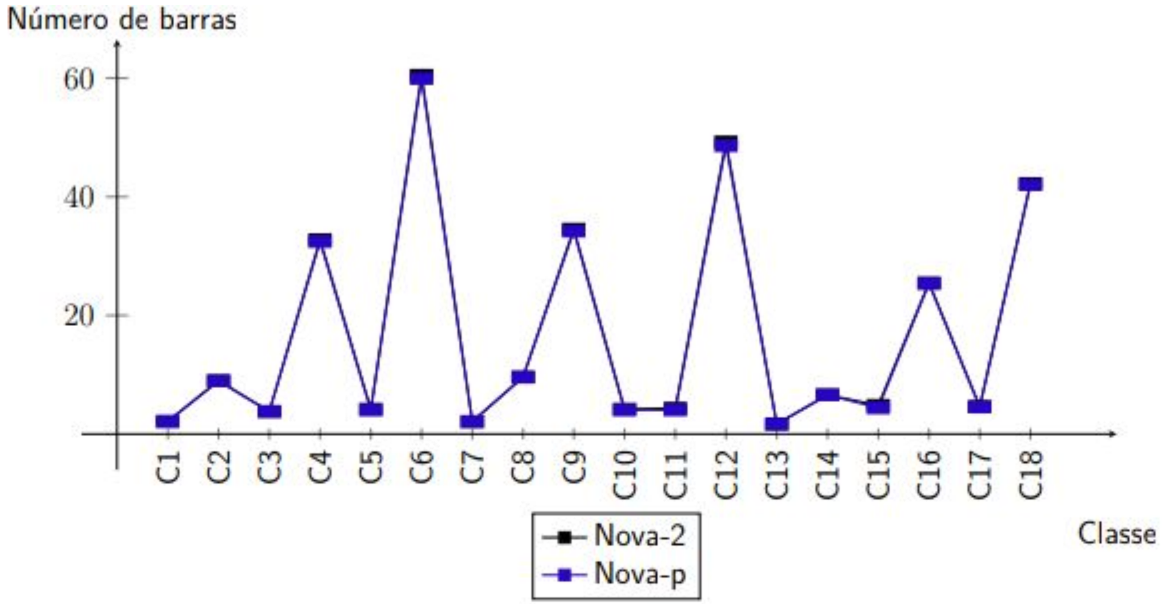
# Análise de Resultados

Figura 10 - Número de barras cortadas médio: *Heurística Nova* e $Nova_p$



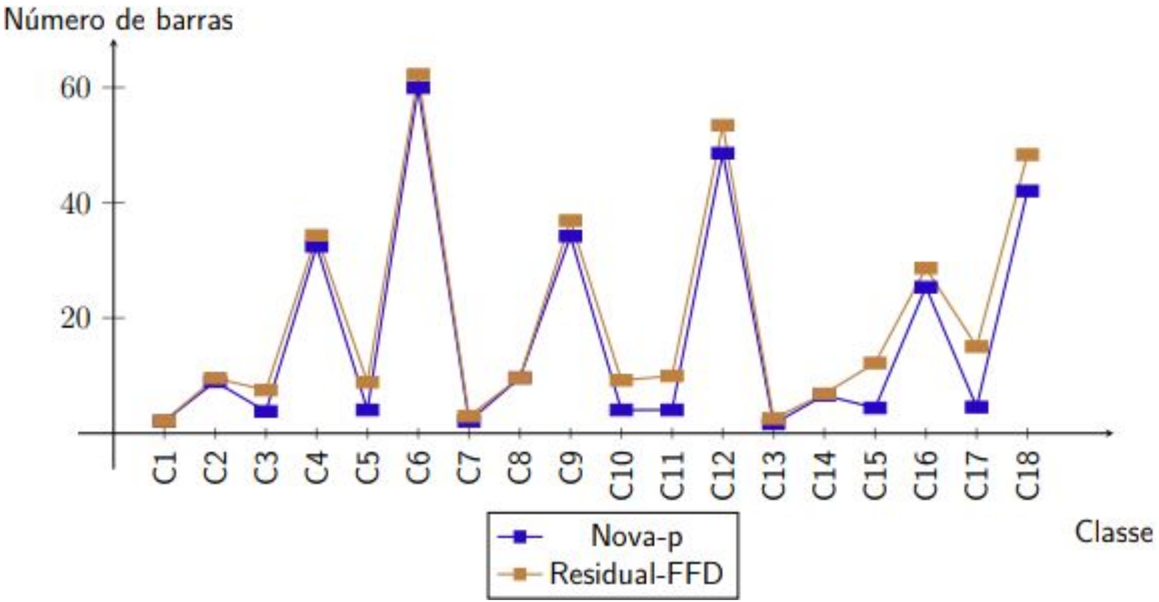
# Análise de Resultados

Figura 11 - Número de barras cortadas médio: *Heurística Nova - versão 2* e *Nova*$_p$



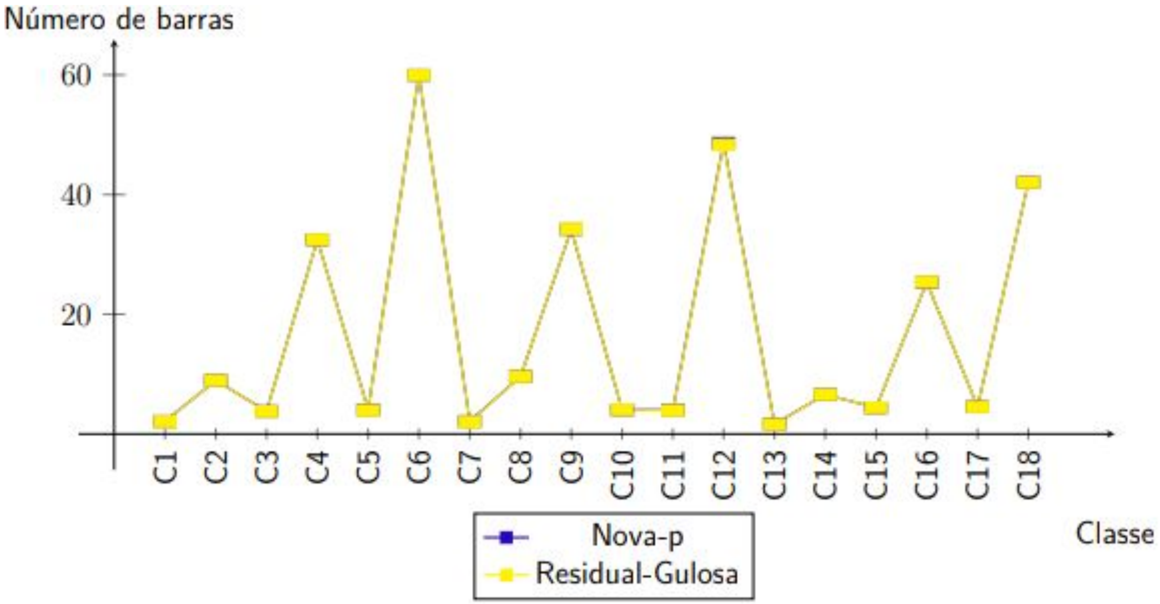
# Análise de Resultados

Figura 12 - Número de barras cortadas médio: *Heurística Residual FFD* e *Nova*$_p$



# Análise de Resultados

Figura 13 - Número de barras cortadas médio: *Residual Gulosa* e *Nova$_p$*



# Análise de Resultados

Tabela 7 - Tempo médio de execução (*ms*) por heurística

| | |
|---|---|
| FFD | 0,8731538462 |
| Gulosa | 91,46304487 |
| Nova | 97,32587821 |
| Nova - versão 2 | 100,003641 |
| $\text{Nova}_P$ | 108,3755128 |
| Residual FFD | 145,1818846 |
| Residual Gulosa | 161,4366795 |

Fonte: Produzido pelo autor

# Agradecimentos


- Orientadora Prof.ª Dr.ª Adriana Cristina Cherri
- Universidade Estadual Paulista "Júlio de Mesquita Filho"
- Colégio Técnico Industrial "Professor Isaac Portal Roldán"
- Família

# Referências

EISEMANN, K. The trim problem. Management Science, v. 3, n. 3, p. 279–284, 1957. Disponível em: <https://www.jstor.org/stable/2627456>.

FORD, J. L.; FULKERSON, D. R. A suggested computation for maximal multi-commodity network flows. Management Science, v. 5, n. 1, p. 97–101, 1958. Disponível em: <http://www.dtic.mil/dtic/tr/fulltext/u2/606440.pdf>.

GILMORE, P. C.; GOMORY, R. E. A linear programming approach to the cutting-stock problem. Oper. Res., INFORMS, Institute for Operations Research and the Management Sciences (INFORMS), Linthicum, Maryland, USA, v. 9, n. 6, p. 849–859, 1961. ISSN 0030-364X. Disponível em: <http://dx.doi.org/10.1287/opre.9.6.849>

# Referências

GILMORE, P. C.; GOMORY, R. E. A linear programming approach to the cutting stock problem—part ii. Operations research, INFORMS, v. 11, n. 6, p. 863–888, 1963.

HAESSLER, R. W. Controlling cutting pattern changes in one-dimensional trim problems. Operations Research, v. 23, p. 483–493, 1975.

HAESSLER, R. W. Technical note-a note on computational modifications to the gilmoregomory cutting stock algorithm. Oper. Res., INFORMS, Institute for Operations Research and the Management Sciences (INFORMS), Linthicum, Maryland, USA, v. 28, n. 4, p. 1001–1005, 1980. ISSN 0030-364X. Disponível em: <http://dx.doi.org/10.1287/opre.28.4.1001>.

# Referências

HINXMAN, A. I. The trim-loss and assortment problems: A survey. European Journal of Operational Research, v. 5, n. 1, p. 8–18, July 1980. Disponível em:
IBM. Modifying the model. 2018. Disponível em: <https://ideas.repec.org/a/eee/ejores/v5y1980i1p8-18.html>. Acesso em: 15 out. 2018.
PIERINI, L. M. Problemas de dimensionamento de lotes e corte de estoque no processo industrial de produção de papel. Dissertação (Mestrado em Matemática Aplicada) - Instituto de Matemática, Estatística e Computação Científica, Universidade Estadual de Campinas, v. 1, n. 1, p. 15, 2017.

# Referências

PINTO, M. J. O problema de corte de estoque inteiro. Dissertação (Mestrado em Ciências de Computação e Matemática Computacional) - Instituto de Ciências Matemáticas de Computação, Universidade de São Paulo, São Carlos, v. 1, n. 1, 1999. Disponível em: <http://www.teses.usp.br/teses/disponiveis/55/55134/tde-06032018-165206/pt-br.php>.

POLDI, K. C. Algumas extensões do problema de corte de estoque. Dissertação (Mestrado em Ciências de Computação e Matemática Computacional) - Instituto de Ciências Matemáticas 55 e de Computação, Universidade de São Paulo, São Carlos, v. 1, n. 1, p. 1–4, 2003.

# Referências

POLDI, K. C.; ARENALES, M. N. Heuristics for the one-dimensional cutting stock problem with limited multiple stock lengths. Comput. Oper. Res., Elsevier Science Ltd., Oxford, UK, UK, v. 36, n. 6, p. 2074–2081, jun. 2009. ISSN 0305-0548. Disponível em: <http://dx.doi.org/10.1016/j.cor.2008.07.001>.

STADTLER, H. A one-dimensional cutting stock problem in the aluminium industry and its solution. European Journal of Operational Research, v. 44, n. 2, p. 209–223, 1990. Disponível em: <https://ideas.repec.org/a/eee/ejores/v44y1990i2p209-223.html>.

# Referências

VAHRENKAMP, R. Random search in the one-dimensional cutting stock problem. European Journal of Operational Research, v. 95, p. 191–200,1996.

WASCHER, G.; GAU, T. Heuristics for the integer one-dimensional cutting stock problem: A computational study. OR Spectrum, v. 18, n. 3, p. 131–144, 1996. Disponível em: <https://www.researchgate.net/publication/225858913_Heuristics_for_the_integer_one-dimensional_cutting_stock_problem_A_computational_study>.