

# Trabalho Computacional 3 - Teoria da Decisão (ELE088)

Daniel Felipe de Almeida Araújo  
Universidade Federal de Minas Gerais  
Matrícula: 2023422617

Milton Pereira Bravo Neto  
Universidade Federal de Minas Gerais  
Matrícula: 2018072549

Raphael Henrique Braga Leivas  
Universidade Federal de Minas Gerais  
Matrícula: 2020028101

**Abstract—**The abstract goes here.

## I. INTRODUÇÃO

This demo file is intended to serve as a “starter file” for IEEE conference papers produced under L<sup>A</sup>T<sub>E</sub>X using IEEEtran.cls version 1.6b and later.

May all your publication endeavors be successful.

mds

November 18, 2002

2) *Problema 2:* O custo da falha de cada equipamento é dado pelo produto da probabilidade de falha  $p_{ij}$  pelo custo da falha do equipamento, dada por  $d_i$ . Assim, temos

$$\min f_2 = \sum_{i=1}^N \sum_{j=1}^J p_{ij} d_i x_{ij} \quad (4)$$

onde

## II. METODOLOGIA

### A. Modelagem do Problema

Inicialmente podemos ver o trabalho como sendo dois problemas mono-objetivo distintos:

- Problema 1: minimização do custo de manutenção total  $f_1(\cdot)$
- Problema 2: minimização do custo esperado de falha total  $f_2(\cdot)$

1) *Problema 1:* Temos essencialmente um problema de designação simples. Seja  $N$  o número de equipamentos e  $J$  o número de políticas de manutenção, definimos a variável de decisão  $x_{ij}$  por

$$x_{ij} : \text{ se o equipamento } i \text{ executa a manutenção } j \quad (1)$$

onde

$$x_{ij} \in \{0, 1\} \quad , \quad i = \{1, 2, \dots, N\} \quad , \quad j = \{1, 2, \dots, J\}$$

Para a função objetivo, seja  $c_j$  o custo de executar a manutenção  $j$ . Note que esse custo independe do equipamento  $i$  que estamos executando a manutenção. Temos a função objetivo

$$\min f_1 = \sum_{i=1}^N \sum_{j=1}^J c_j x_{ij} \quad (2)$$

sujeito a

$$\sum_{j=1}^J x_{ij} = 1 \quad , \quad \forall i = 1, 2, \dots, N \quad (3)$$

A 3 indica que todo equipamento executa exatamente uma política de manutenção. Além disso, note que solução da 2 é trivial: basta escolher o plano de manutenção com o menor custo para todos os equipamentos.

$$x_{ij} \in \{0, 1\} \quad , \quad i = \{1, 2, \dots, N\} \quad , \quad j = \{1, 2, \dots, J\}$$

$$p_{ij} = \frac{F_i(t_0 + k_j \Delta t) - F_i(t_0)}{1 - F_i(t_0)} \quad (5)$$

$$F_i(t) = 1 - \exp \left[ - \left( \frac{t}{\eta_i} \right)^{\beta_i} \right] \quad (6)$$

sujeito a

$$\sum_{j=1}^J x_{ij} = 1 \quad , \quad \forall i = 1, 2, \dots, N \quad (7)$$

Note que na Equação 4 temos essencialmente um problema de programação linear inteira. Assim, é possível usar o método Simplex visto em Pesquisa Operacional para resolver esse problema com garantia de otimalidade. Usando o Simplex, a solução encontrada foi

$$\mathbf{x}^* = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 0 & 1 \\ \vdots & \vdots & \vdots \\ 0 & 0 & 1 \end{bmatrix} \quad , \quad f_2(\mathbf{x}^*) = 1048.17$$

Assim, antes mesmo de começar a implementar o BVNS para resolver os problemas isoladamente, já sabemos as soluções ótimas para eles.

3) *Modelagem Multiobjetivo:* Juntando as modelagens dos problemas mono-objetivos acima, temos a modelagem multi-objetivo do problema.

$$\min f_1 = \sum_{i=1}^N \sum_{j=1}^J c_j x_{ij}$$

$$\min f_2 = \sum_{i=1}^N \sum_{j=1}^J p_{ij} d_i x_{ij}$$

$x_{ij}$  : se o equipamento  $i$  executa a manutenção  $j$

sujeito a

$$\sum_{j=1}^J x_{ij} = 1 \quad , \quad \forall i = 1, 2, \dots, N$$

$$x_{ij} \in \{0, 1\} \quad , \quad i = \{1, 2, \dots, N\} \quad , \quad j = \{1, 2, \dots, J\}$$

onde

- $N = 500$ : número de equipamentos
- $J = 3$ : número de planos de manutenção
- $c_j$ : custo de executar a manutenção  $j$
- $p_{ij}$ : probabilidade de falha do equipamento  $i$  executando a manutenção  $j$ , definido em 5 e 6
- $d_i$ : custo de falha do equipamento  $i$

A partir dessa modelagem, temos o nosso problema multi-objetivo

$$\min \mathbf{f}(\mathbf{x}) = [f_1(\mathbf{x}), f_2(\mathbf{x})] \quad (8)$$

Considerando o problema de (8), podemos aplicar duas abordagens escalares para obter a fronteira Pareto-ótima no espaço de objetivos, descritas a seguir.

#### B. Formulações para resolução do problema multiobjetivo

1) *Formulação Soma Ponderada  $P_w$* : Seja  $0 \leq w \leq 1$  um peso qualquer gerado aleatoriamente de uma distribuição uniforme no intervalo  $[0, 1]$ . Usando a abordagem da soma ponderada, podemos reescrever (8) na forma de mono-objetivo de

$$\min f_w = \min w f_1 + (1 - w) f_2 \quad (9)$$

onde (9) está sujeito às mesmas restrições do problema original. Como (9) é escalar, podemos minimizar  $f_w$  através de métodos já conhecidos como o Simplex e o BVNS.

2) *Formulação  $\epsilon$ -Restrito  $P_\epsilon$* : Com a abordagem do  $\epsilon$ -Restrito, vamos minimizar apenas  $f_1$  usando  $f_2$  como restrição. Seja  $\epsilon_2$  um real qualquer tal que  $\min f_2 \leq \epsilon_2 \leq \max f_2$ . Temos

$$\min f_1 \quad (10)$$

sujeito a

$$\begin{cases} f_2 \leq \epsilon_2 \\ \sum_{j=1}^J x_{ij} = 1 \quad , \quad \forall i = 1, 2, \dots, N \end{cases} \quad (11)$$

em que (10) possui as mesmas restrições do problema original mais a restrição de  $f_2 \leq \epsilon_2$ .

Contudo, como o BVNS é usado para resolver problemas de otimização irrestritos, precisamos converter (10) em um

problema irrestrito. Para isso, adicionamos o termo um termo de penalidade  $p(x, u)$  da seguinte forma:

$$p(x, u) = u \max[0, g(x)]^2$$

onde  $g(x)$  é a nossa restrição de desigualdade, dada por

$$g(x) \leq 0 \implies f_2 - \epsilon_2 \leq 0 \implies g(x) = f_2 - \epsilon_2$$

de modo que o nosso problema irrestrito se torna:

$$\min f_1 + u \max[0, f_2 - \epsilon_2]^2 \quad (12)$$

Note que as demais restrições já estão naturalmente incluídas no BVNS devido à maneira como nós fizemos a representação computacional das variáveis de decisão, de modo que só precisamos fazer a correção para a restrição do  $\epsilon$  em (12).

3) *Normalização*: Para garantir que as abordagens escalares sejam condizentes, precisamos normalizar  $f_1$  e  $f_2$  através de

$$f_1(\mathbf{x}) = \frac{f_1(\mathbf{x}) - \min f_1}{\max f_1 - \min f_1} \quad , \quad f_2(\mathbf{x}) = \frac{f_2(\mathbf{x}) - \min f_2}{\max f_2 - \min f_2} \quad (13)$$

de modo que a formulação da soma ponderada de (9) pode ser reescrita como

$$\min \left( w \frac{f_1(\mathbf{x}) - \min f_1}{\max f_1 - \min f_1} + (1 - w) \frac{f_2(\mathbf{x}) - \min f_2}{\max f_2 - \min f_2} \right)$$

que será usado como função objetivo no código do BVNS.

4) *BVNS - Basic Variable Neighborhood Search*: O Algoritmo 1 mostra a versão do BVNS implementada no trabalho.

---

#### Algoritmo 1 BVNS implementado no trabalho.

---

```

1: procedure BVNS( $\mathbf{x}$ ,  $k_{\max}$ )
2:   while num_sol_avaliadas < max_sol_avaliadas do
3:      $k \leftarrow 1$ 
4:     while  $k < k_{\max}$  do
5:        $\mathbf{x}' \leftarrow \text{SHAKE}(\mathbf{x}, k)$ 
6:        $\mathbf{x}'' \leftarrow \text{FIRSTIMPROVEMENT}(\mathbf{x}, \mathbf{x}', k)$ 
7:        $\mathbf{x}, k \leftarrow \text{NEIGHBORHOODCHANGE}(\mathbf{x}, \mathbf{x}'', k)$ 
8:     end while
9:   end while
10: end procedure

```

---

O Algoritmo 2 mostra a função Shake. Nela estão definidas as três estruturas de vizinhança escolhidas para implementação. As duas primeiras são vizinhanças de refinamento, mas com abordagens diferentes. E a terceira é uma vizinhança de perturbação para buscar sair de mínimos locais:

- A primeira estrutura é o que chamamos de um movimento de *1-swap*, onde é escolhido aleatoriamente um equipamento e trocado seu plano para um dos outros dois restantes, a escolha do plano também é aleatória.
- A segunda estrutura é a troca ou permutação dos planos de dois equipamentos diferentes escolhidos também aleatoriamente.

- A terceira estrutura por sua vez, altera um bloco de 50 equipamentos em sequência, onde o início do bloco é aleatório. Nesse bloco é avaliado qual o plano mais comum e troca-se o plano de manutenção de todos os integrantes do bloco para um mesmo plano, diferente do mais comum encontrado anteriormente.

---

**Algoritmo 2** Função Shake.

---

▷ Gera uma solução aleatória na  $k$ -ésima estrutura de vizinhança.

```

1: procedure SHAKE( $x$ ,  $k$ )
2:   if  $k = 1$  then
3:      $y \leftarrow$  1-swap
4:   end if
5:   if  $k = 2$  then
6:      $y \leftarrow$  Permutação de dois planos de manutenção
7:   end if
8:   if  $k = 3$  then
9:      $y \leftarrow$  Mudança de um bloco de equipamentos para
      outro plano
10:  end if
11:  return  $y$ 
12: end procedure

```

---

5) *Estratégias de Refinamento*: O Algoritmo 3 mostra a função de busca local implementada após gerar uma solução aleatória com o Shake. Ela basicamente realiza uma busca em até  $N = 100$  vizinhos à solução inicial  $x'$  do Shake, e retorna a primeira solução  $x''$  cujo valor da solução objetivo é menor do que o valor do objetivo na solução inicial  $x'$  do Shake. Caso nenhuma solução melhor é encontrada, retorna a solução inicial  $x'$ .

---

**Algoritmo 3** Função FirstImprovement.

---

▷ Busca uma primeira solução na vizinhança de  $x'$  melhor que  $x'$ .

```

1: procedure FIRSTIMPROVEMENT( $x'$ ,  $k$ )
2:    $N \leftarrow 100$ 
3:   for all  $i$  in range( $N$ ) do
4:      $x'' \leftarrow$  SHAKE( $x'$ ,  $k$ )
5:     if  $f(x'') < f(x')$  then
6:       return  $x''$ 
7:     end if
8:   end for
9:   return  $x'$ 
10: end procedure

```

---

É possível fazer uma pequena modificação no Algoritmo 3 para obter o Best Improvement, exibido no Algoritmo 4. Note que essa função sempre executa as  $N$  buscas por uma melhor solução, e portanto o código é mais caro computacionalmente que no Algoritmo 3. No entanto, em geral, a solução encontrada pelo BestImprovement será melhor que a do FirstImprovement.

---

**Algoritmo 4** Função BestImprovement.

---

▷ Busca a melhor solução na vizinhança de  $x'$  melhor que  $x'$ .

```

1: procedure BESTIMPROVEMENT( $x'$ ,  $k$ )
2:    $N \leftarrow 100$ 
3:    $x\_melhor \leftarrow x'$ 
4:   for all  $i$  in range( $N$ ) do
5:      $x'' \leftarrow$  SHAKE( $x'$ ,  $k$ )
6:     if  $f(x'') < f(x\_melhor)$  then
7:        $x\_melhor \leftarrow x''$ 
8:     end if
9:   end for
10:  return  $x\_melhor$ 
11: end procedure

```

---

6) *Heurística Construtiva*: O Algoritmo 5 mostra a heurística construtiva utilizada para a criação da solução inicial. Basicamente o problema se reduz em escolher um plano de manutenção, dentre os três disponíveis, para cada equipamento minimizando o custo da manutenção e o custo de falha dos equipamentos. A minimização do custo da manutenção se dá escolhendo a manutenção mais barata para todos os equipamentos, e a minimização do custo de falha escolhendo a manutenção mais cara.

Olhando para o segundo problema, temos a matriz de custos de falha  $p_{ij}d_i$  onde  $i$  é cada equipamento e  $j$  os planos de manutenção. Para cada equipamento  $i$  fixo avalia-se a variância de  $p_{ij}d_i$  e caso esse valor seja maior que o limiar de 0.5 escolhe a manutenção mais cara para compor a solução inicial daquele equipamento, caso seja menor que o limiar é escolhida a manutenção mais barata.

A lógica envolvida é que se o custo de falha não varia tanto para aquele equipamento, não é necessário a manutenção mais cara.

---

**Algoritmo 5** Heurística construtiva para gerar a solução inicial.

---

```

1: procedure SOLUCAOINICIAL()
2:    $x \leftarrow$  Solução aleatória
3:   for all  $i$  in  $x$  do
4:     if variancia( $p_{ij}d_i$ )  $\geq$  limiar then
5:        $x[i] \leftarrow$  Manutenção mais cara
6:     else
7:        $x[i] \leftarrow$  Manutenção mais barata
8:     end if
9:   end for
10:  return  $x$ 
11: end procedure

```

---

### III. RESULTADOS

#### A. Problemas mono objetivos

A partir da execução do algoritmo BVNS para a resolução dos problemas mono-objetivo foi obtido os seguintes valores:

$$\min f_1 = 0 \quad (14)$$

$$\max f_1 = 1000 \quad (15)$$

$$\min f_2 = 1048.17 \quad (16)$$

$$\max f_2 = 1745.49 \quad (17)$$

Esses resultados são utilizados para a realizar a normalização dos valores ao se utilizar a formulação de Soma Ponderada e os mínimos estão ilustrados nas Figuras 1 e 2.

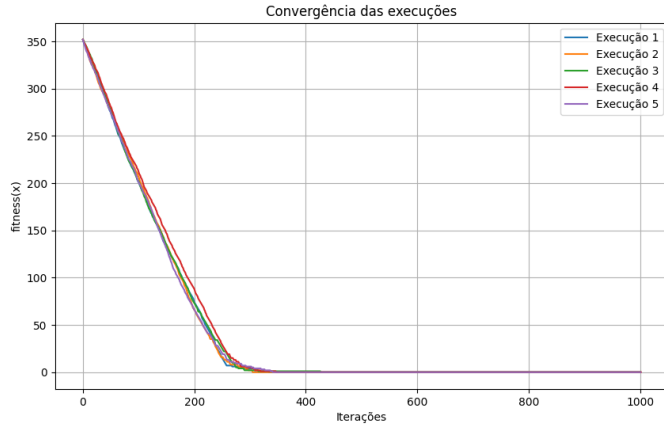


Fig. 1. Convergência do BVNS implementado para o Problema 1 Isolado.

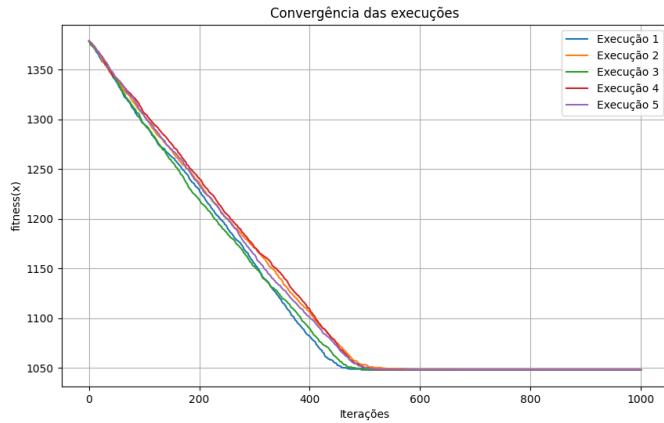


Fig. 2. Convergência do BVNS implementado para o Problema 2 Isolado.

### B. Indicador de Hipervolume (HVI)

Para a realização do cálculo a métrica de HVI foi permitido o uso de mais de 20 soluções pareto na fronteira, de maneira que o valor para a métrica atingisse os valores indicados para comparação. Dessa forma, foram utilizadas 200 soluções e a visualização da fronteira pode ser vista na Figura 3.

O valor de HVI obtido foi:

$$\text{HVI: } 0.602974 \quad (18)$$

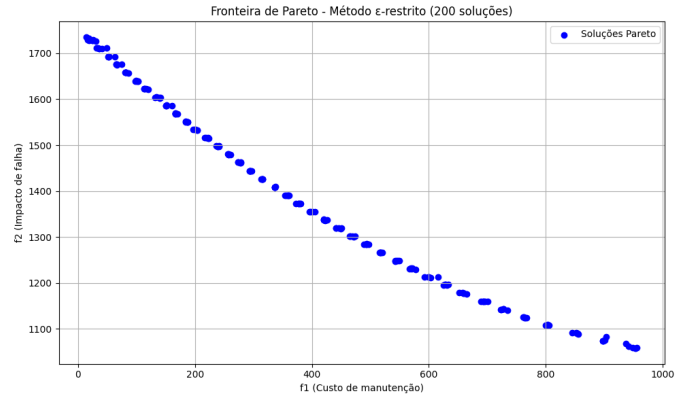


Fig. 3. Fronteira Pareto para 200 soluções com método  $\epsilon$ -restrito usada para cálculo do HVI

## IV. CONCLUSÃO

The conclusion goes here.

## REFERÊNCIAS

- [1] H. Kopka and P. W. Daly, *A Guide to L<sup>A</sup>T<sub>E</sub>X*, 3rd ed. Harlow, England: Addison-Wesley, 1999.