

Trabalho Computacional 01

Teoria da Decisão (ELE088)

Raphael Henrique Braga Leivas
Milton Pereira Bravo Neto
Daniel - Completar

Curso de Bacharelado em Engenharia de Sistemas
Universidade Federal de Minas Gerais

Sumário

Modelagem

- Problema 1 Isolado

- Problema 2 Isolado

- Problema Multiobjetivo

Algoritmos

- Representação Computacional

- BVNS - Basic Variable Neighborhood Search

Resultados

- Problema 1 Isolado

- Problema 2 Isolado

Referências

Temos que modelar dois problemas mono-objetivos:

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

Problema 1 Isolado

Problema 1: minimização do custo de manutenção total

Variável de decisão:

x_{ij} : se a máquina i executa a manutenção j

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

Função Objetivo e Restrições

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

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

Problema 1 Isolado - Solução Trivial

Atribuir a manutenção mais barata a todos os equipamentos:

$$x^* = \begin{bmatrix} 1 & 0 & 0 \\ 1 & 0 & 0 \\ \vdots & \vdots & \vdots \\ 1 & 0 & 0 \end{bmatrix}$$

$$f_1(x^*) = 0$$

Problema 2 Isolado

Problema 2: minimização do custo esperado de falha total

Variável de decisão:

x_{ij} : se a máquina i executa a manutenção j

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

Problema 2 Isolado

Função Objetivo

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

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

$$p_{ij} = \frac{F_i(t_0 + k_j \Delta t) - F_i(t_0)}{1 - F_i(t_0)} \quad , \quad F_i(t) = 1 - \exp \left[- \left(\frac{t}{\eta_i} \right)^{\beta_i} \right]$$

Restrição

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

Problema 2 Isolado - Solução via Simplex

Atribuir a manutenção mais cara a todos os equipamentos:

$$x^* = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 0 & 1 \\ \vdots & \vdots & \vdots \\ 0 & 0 & 1 \end{bmatrix}$$

$$f_2(x^*) = 1048.17$$

Problema Multiobjetivo

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

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

Sujeito a

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

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

x_{ij} : se a máquina i executa a manutenção j

Modelagem

- Problema 1 Isolado

- Problema 2 Isolado

- Problema Multiobjetivo

Algoritmos

- Representação Computacional

- BVNS - Basic Variable Neighborhood Search

Resultados

- Problema 1 Isolado

- Problema 2 Isolado

Referências

Representação Computacional

- ▶ Vetor de tamanho $N = 500$
- ▶ Cada posição i indica o custo da manutenção associada ao equipamento i

$$x = [2 \quad 0 \quad 1 \quad \dots \quad 2]$$

- ▶ Equipamento 1 executa manutenção de custo 2
- ▶ Equipamento 2 executa manutenção de custo 0
- ▶ ...
- ▶ Equipamento 500 executa manutenção de custo 2

BVNS - Basic Variable Neighborhood Search

Algoritmo 1 BVNS implementado no trabalho.

```
1: procedure BVNS( $\mathbf{x}$ ,  $k_{\max}$ )
2:   while num_sol_avaliables < max_sol_avaliables 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
```

Algoritmo 2 Função Shake.

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

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

BVNS - Basic Variable Neighborhood Search

Algoritmo 3 Função FirstImprovemnt.

▷ Busca uma solução na vizinhança de x melhor que x .

```
1: procedure FIRSTIMPROVEMENT( $x$ ,  $x'$ ,  $k$ )  
2:    $N \leftarrow$  100 vizinhos aleatórios da  $k$ -ésima vizinhança  
3:   for all  $v$  in  $N$  do  
4:     if  $f(v) < f(x)$  then  
5:       return  $v$   
6:     end if  
7:   end for  
  
8:   return  $x$   
9: end procedure
```

BVNS - Basic Variable Neighborhood Search

Algoritmo 4 Heurística construtiva para gerar a solução inicial.

```
1: procedure SOLUCAOINICIAL()
2:    $x \leftarrow$  Solução aleatória
3:   for all  $i$  in  $x$  do
4:     if  $\text{variancia}(p_{ij}d_i) \geq \text{limite}$  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
```

Modelagem

- Problema 1 Isolado

- Problema 2 Isolado

- Problema Multiobjetivo

Algoritmos

- Representação Computacional

- BVNS - Basic Variable Neighborhood Search

Resultados

- Problema 1 Isolado

- Problema 2 Isolado

Referências

TESTE

TESTE

Modelagem

- Problema 1 Isolado

- Problema 2 Isolado

- Problema Multiobjetivo

Algoritmos

- Representação Computacional

- BVNS - Basic Variable Neighborhood Search

Resultados

- Problema 1 Isolado

- Problema 2 Isolado

Referências

TESTE