

Um Enigma das Galáxias - Parte 1

SME0110 - Programação Matemática

Angelo A. R. Tessaro, N^oUSP: 10310551

Bruno Mitsuo Homma, N^oUSP: 9293605

Leandro Antonio Silva, N^oUSP: 9805341

Marilene Andrade Garcia, N^oUSP: 10276974

1 Tarefa 1

O objetivo da tarefa foi desenvolver um modelo de otimização linear-mista para resolução do problema do astrônomo. A modelagem para o problema do enigma da galáxias visa encontrar o melhor movimento possível que seria possível obter o menor movimento do telescópio e encontre N galáxias.

1.1 Parâmetros

N : número de galáxias

$G = \{0, 1, 2, \dots, N - 1\}$: conjunto de galáxias que podem ser visitadas pelo astrônomo

d_{ij} : distância da galáxia i para a galáxia j , com $i = 0, 1, \dots, N - 1$; $j = 0, 1, \dots, N - 1$, $i \neq j$

Observação 1: Assume-se que a distância para ir da galáxia i para a galáxia j é a mesma distância do caminho inverso, de j para i ($d_{ij} = d_{ji}$).

Observação 2: A distância de uma galáxia para ela mesma (d_{ii}) é modelada como um valor muito alto para permitir o funcionamento do modelo - valor este que é muito maior que o valor máximo de distância entre quaisquer duas galáxias.

1.2 Variáveis

$$x_{ij} = \begin{cases} 1 & \text{se for percorrida a trajetória da galáxia } i \text{ para a galáxia } j \\ 0 & \text{caso contrário} \end{cases}$$

$$i = 0, 1, \dots, N - 1; j = 0, 1, \dots, N - 1, i \neq j$$

1.3 Restrições

I. $\sum_{i \in G - \{k\}} x_{ik} = 1, \quad k = 0, 1, \dots, N - 1$
Garante que um destino tem apenas uma galáxia como origem

II. $\sum_{j \in G - \{k\}} x_{kj} = 1, \quad k = 0, 1, \dots, N - 1$
Garante que uma origem tem apenas uma galáxia como destino

III. $\sum_{i \in S, j \notin S} x_{ij} \geq 1 \quad S \subseteq G, 2 \leq |S| \leq N - 2$
Evita sub-ciclos

$x_{ij} \in \{0, 1\}$
Indica que a variável x é binária

1.4 Função Objetivo

$$\min \sum_{i \in G} \sum_{j \in G} d_{ij} x_{ij}$$

A função objetivo busca minimizar a distância percorrida quando se visita todas as galáxias e retorna à galáxia de origem

2 Tarefa 2

A partir do modelo, foi-se desenvolvido um código computacional para resolução do problema. Esse código foi feito na linguagem Python usando a biblioteca OR-Tools, uma aplicação livre de resolução de problemas de modelagem.

Todo o projeto pode ser encontrado no seguinte repositório do GitHub:
github.com/MarileneGarcia/SME0110LinearOptimization

2.1 Código

No geral, o código foi segmentado exatamente nas partes características de um problema de modelagem - parâmetros, variáveis, restrições e função objetivo.

É dada a opção de se usar um problema-exemplo com 5 galáxias e parâmetros fixos. Além disso, também é possível personalizar o problema com valores aleatórios de parâmetro, bastando apenas escolher um número de galáxias entre 2 e 10.

2.1.1 Parâmetros

```
1 galaxias = ['Andromeda', 'OlhoNegro', 'Girassol ', 'CataVento', 'Magalhaes',  
             'Charuto', 'Redshift 7', 'Hoag', 'Sombreiro', 'Girino']  
2 d_maxima = 100  
3 d_infinita = 100000 * d_maxima  
4
```

```

5 # Entrada da preferencia de uso do programa
6 begin = int(input("----> Digite 0 para o caso exemplo e 1 para personalizar
                        : "))
7
8 # Parametros pre-fixados (begin = 0)
9 N = 5
10 d = [[d_infinita, 100, 1, 1, 1],
11       [100, d_infinita, 1, 100, 1],
12       [1, 1, d_infinita, 1, 100],
13       [1, 100, 1, d_infinita, 1],
14       [1, 1, 100, 1, d_infinita]]
15
16 # Parametros aleatorios por input (begin = 1)
17 N = int(input("----> Digite o numero de galaxias: "))
18 d = gerar_aleatorio(G)
19
20 # Escolha da galaxia de inicio
21 inicio = int(input("----> De que galaxia voce quer partir, senhor astrono-
                        ? "))

```

2.1.2 Variáveis

$$x_{ij} = \begin{cases} 1 & \text{se for percorrida a trajetória da galáxia } i \text{ para a galáxia } j \\ 0 & \text{caso contrário} \end{cases}$$

```

1 x = [[0 for i in range(N)] for j in range(N)]
2 for i in range(0, N):
3     for j in range(0, N):
4         name_x = 'x' + str(i) + str(j)
5         x[i][j] = solver.BoolVar(name_x)

```

2.1.3 Restrições I

$$\sum_{i \in G - \{k\}} x_{ik} = 1 \quad k = 0, 1, \dots, N - 1$$

```

1 for g in range(0, N):
2     name_ct = 'ct' + str(g) + 'k'
3     name_ct = solver.Constraint(1, 1)
4     for i in range(0, N):
5         for j in range(0, N):
6             if g == i:
7                 name_ct.SetCoefficient(x[i][j], 1)
8             else:
9                 name_ct.SetCoefficient(x[i][j], 0)

```

2.1.4 Restrições II

$$\sum_{j \in G - \{k\}} x_{kj} = 1, \quad k = 0, 1, \dots, N-1$$

```
1 for g in range(0, N):
2     name_ct = 'ct' + 'k' + str(g)
3     name_ct = solver.Constraint(1, 1)
4     for j in range(0, N):
5         for i in range(0, N):
6             if g == j:
7                 name_ct.SetCoefficient(x[i][j], 1)
8             else:
9                 name_ct.SetCoefficient(x[i][j], 0)
```

2.1.5 Restrições III

$$\sum_{i \in S, j \notin S} x_{ij} \geq 1 \quad S \subseteq G, 2 \leq |S| \leq N-2$$

```
1 for m in range(0, N):
2     for n in range(m+1, N):
3         # (m,n) cada subciclo
4         name_ct = 'ct' + str(m) + str(n)
5         name_ct = solver.Constraint(1, solver.infinity())
6         for i in range(0, N):
7             for j in range(0, N):
8                 if i==m and j!=m and j!=n:
9                     name_ct.SetCoefficient(x[i][j], 1)
10                elif i==n and j!=m and j!=m:
11                    name_ct.SetCoefficient(x[i][j], 1)
12                else:
13                    name_ct.SetCoefficient(x[i][j], 0)
```

2.1.6 Função objetivo

$$\min \sum_{i \in G} \sum_{j \in G} d_{ij} x_{ij}$$

```
1 objective = solver.Objective()
2
3 for i in range(0, N):
4     for j in range(0, N):
5         objective.SetCoefficient(x[i][j], d[i][j])
6 objective.SetMinimization()
```

2.2 Apresentação dos resultados

Para proporcionar uma visualização do problema e dos resultados, foi usada o pacote *igraph* a partir do qual foi possível criar um grafo com as galáxias escolhidas e indicar o caminho encontrado. O código foi executado e as imagens geradas tanto no *Ubuntu18.04* quanto no *Windows10*.

Com intuito de melhorar a análise da resposta são plotados dois grafos, o do problema inicial, ainda não resolvido, e o que contém a resposta final. Além do resultado para o problema-exemplo (5 galáxias com parâmetros pré-fixados), tal como requisitado, o código foi implementado de maneira que o usuário pode escolher até 10 galáxias para serem consideradas na resolução, além de nesse caso, as distâncias entre essas galáxias serem atribuídas aleatoriamente.

2.2.1 5 galáxias - problema exemplo

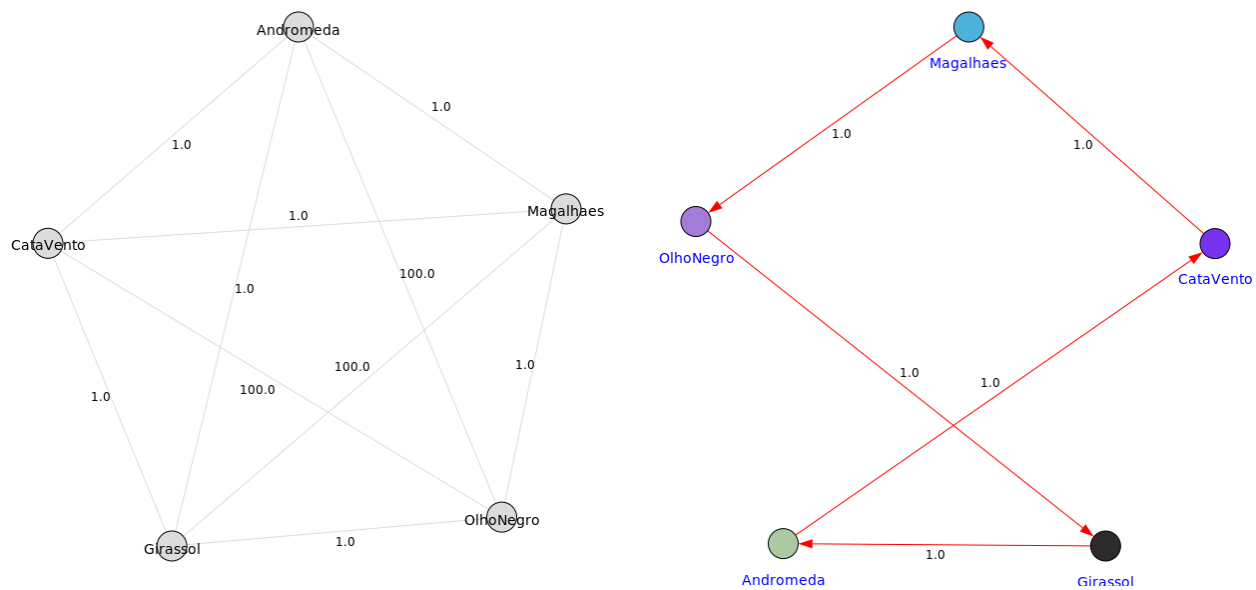


Figura 1: Problema inicial e Resposta final

```

1 Saindo de: CataVento
2
3 Numero de variaveis = 25
4
5 Numero de restricoes = 20
6
7 Solucao - Valor objetivo = 5.0

```

2.2.2 4 galáxias - geração aleatória

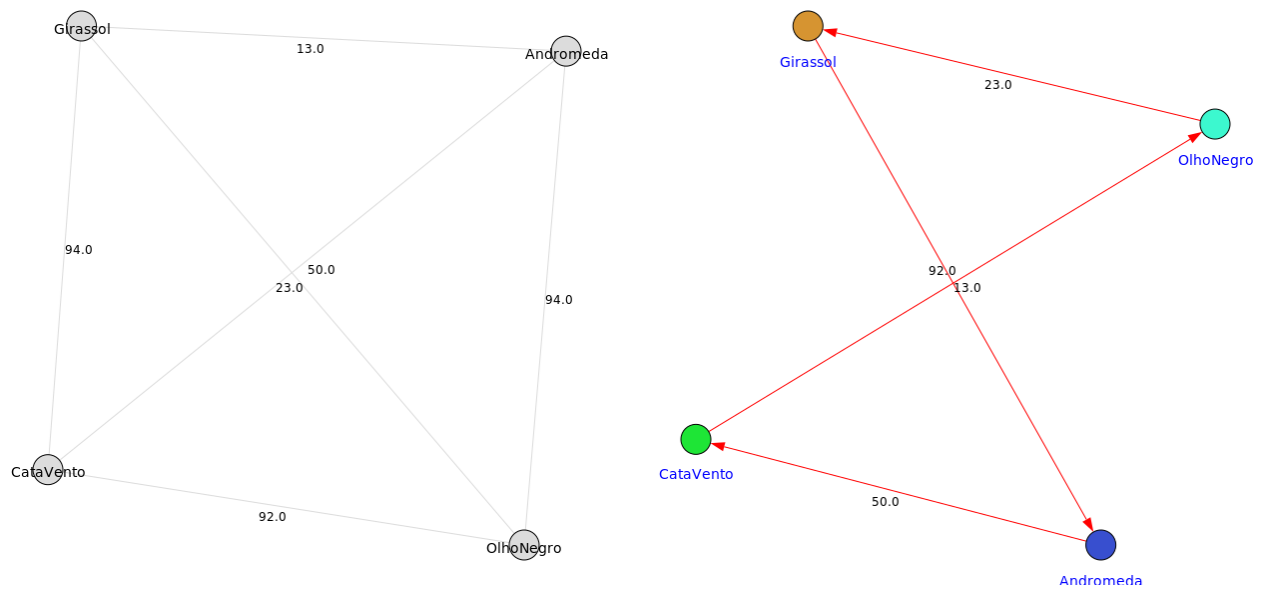


Figura 2: Problema com quatro galáxias e distâncias aleatórias

```
1 Saíndo de: OlhoNegro
2
3 Numero de variaveis = 16
4
5 Numero de restricoes = 14
6
7 Solucao - Valor objetivo = 178.0
```

2.2.3 7 galáxias - geração aleatória

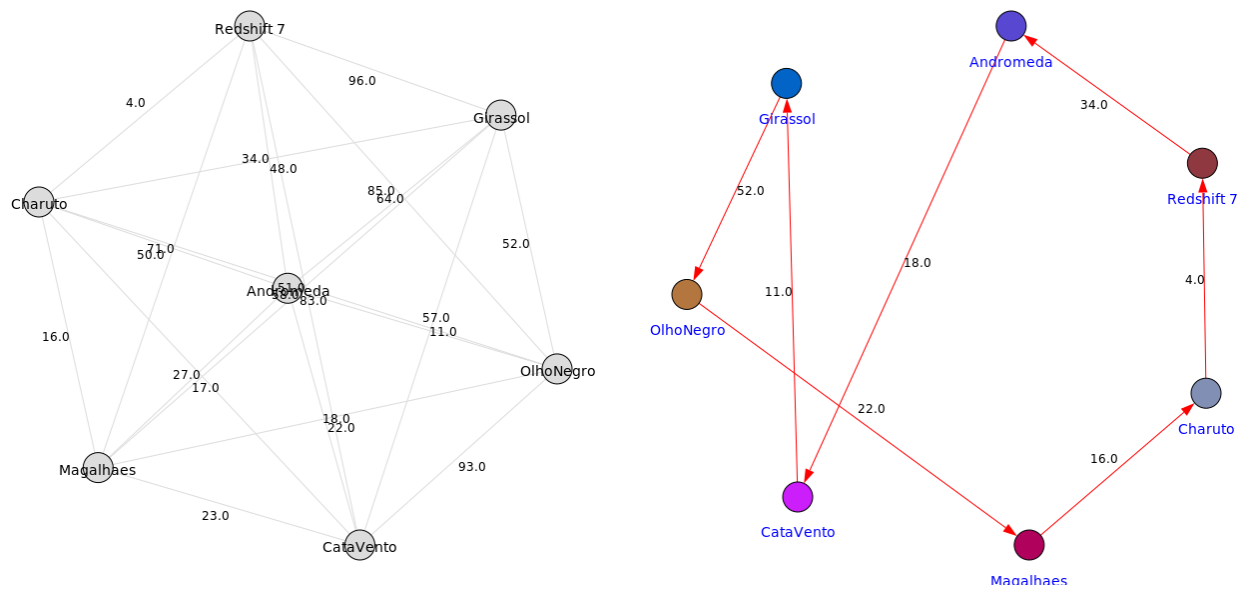


Figura 3: Problema com sete galáxias e distâncias aleatórias

```

1 Saindo de: Girassol
2
3 Numero de variaveis = 49
4
5 Numero de restricoes = 35
6
7 Solucao - Valor objetivo = 157.0

```

2.2.4 10 galáxias - geração aleatória

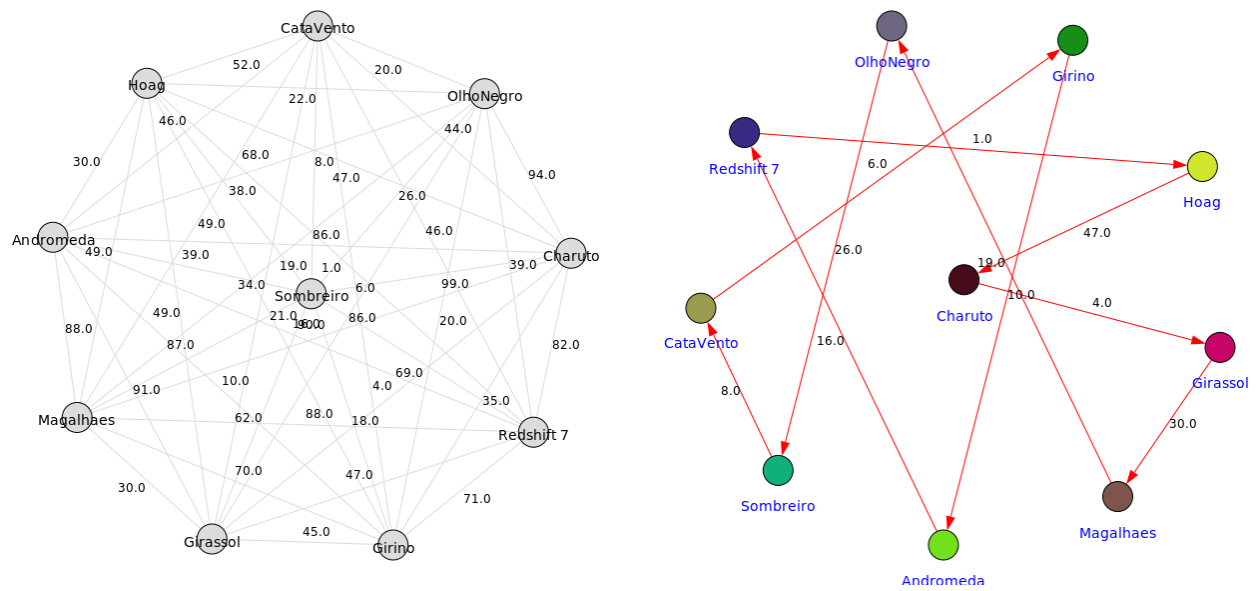


Figura 4: Problema com dez galáxias e distâncias aleatórias

```
1 Saíndo de: Hoag
2
3 Numero de variaveis = 100
4
5 Numero de restricoes = 65
6
7 Solucao - Valor objetivo = 167.0
```