



## **Tópicos Especiais em Pesquisa Operacional** **(Período 2020.3)**

### **1ª Atividade Assíncrona** **(Problemas de Transporte e Designação)**

1. Ler os capítulos e/ou seções do livro "*Pesquisa Operacional para cursos de engenharia*" (de Arenales et al., 2015) abaixo mencionados e elaborar um resumo de no máximo uma página.
  - a) Capítulo 1 (completo);
  - b) Capítulo 2 (seções 2.1, 2.2, 2.2.1 e 2.2.2);
  - c) Capítulo 3 (seção 3.4.3).
2. Implementar os modelos matemáticos dos exemplos 2.3, 2.4, 2.5 e 3.8 (páginas 22, 23, 24 e 179).
  - a) Para o exemplo 3.8, considere os dados abaixo:  
 $C = [16 \ 62 \ 3 \ 92 \ 88 \ 65 \ 67 \ 47;$   
 $21 \ 29 \ 77 \ 51 \ 52 \ 80 \ 34 \ 76;$   
 $20 \ 23 \ 42 \ 41 \ 25 \ 5 \ 88 \ 43]$   
 $A = [30 \ 66 \ 16 \ 84 \ 52 \ 61 \ 31 \ 52;$   
 $22 \ 21 \ 73 \ 84 \ 92 \ 54 \ 31 \ 73;$   
 $22 \ 53 \ 38 \ 62 \ 85 \ 65 \ 33 \ 30]$   
 $b = [200 \ 210 \ 190]$
3. **[Opcional]** Escreva um código que resolva o modelo matemático do problema de designação generalizado para qualquer instância (dados de entrada) que seja apresentada em um arquivo .txt organizado da seguinte maneira:  
*NrAgentes: 3*  
*NrTarefas: 8*  
*MatrizCustos:*  
*11 12 13 14 15 16 17 18*  
*21 22 23 24 25 26 27 28*  
*31 32 33 34 35 36 37 38*  
*MatrizRecursos:*  
*11 12 13 14 15 16 17 18*  
*21 22 23 24 25 26 27 28*  
*31 32 33 34 35 36 37 38*  
*Capacidades:*  
*99 100 122*

#### **Enviar os seguintes arquivos:**

1. Arquivo .pdf contendo as soluções dos modelos implementados e o resumo dos capítulos e/ou seções mencionados. Nomear o arquivo da seguinte maneira: Atividade1-resumo-NOME-SOBRENOME.pdf
2. Os códigos (arquivos .jl). Nomear os arquivos da seguinte maneira: Atividade1-codigo1-NOME-SOBRENOME.jl, Atividade1-codigo2-NOME-SOBRENOME.jl, etc.

#### **Referência:**

**ARENALES, M.; ARMENTANO, V.; YANASSE, H.; MORABITO, R. Pesquisa operacional para cursos de engenharia. 2. ed. Elsevier, 2015.**

#### **- Como declarar uma matriz bi-dimensional:**

modo1) `matriz = Array{Int64}(undef, 2, 4)` # Declara uma matrix 2x4 de valores inteiros  
modo2) `matriz = zeros{Bool, 2, 3}` # Declara uma matrix 2x3 de valores binários, inicializados com zeros  
modo3) `matriz = [1 4 3 1; 1 1 1 2]` # Declara uma matrix 2x4, atribuindo os valores manualmente  
Exemplos de outros tipos de variáveis: `Float64` (real); `Bool` (binário); `Char` (caractere); `Int64` (inteiro)  
Mais detalhes em: <https://docs.julialang.org/en/v1/manual/types/>

#### **- P/ acessar um elemento de uma matriz bi-dimensional:**

`valor = matriz[1, 2]` # acessa o elemento (valor) da (linha 1, coluna 2).

#### **- Como declarar uma restrição de igualdade em Julia+JuMP**

Para uma restrição do tipo:  $x + y = 100$ , fazemos: `@constraint(model, x + y == 100)`

#### **- P/ declarar um conj. de variáveis binárias com 2 índices (ex: $x_{11}, x_{12}, \dots, x_{19}; x_{21}, x_{22}, \dots, x_{29}; (\dots) x_{91}, x_{92}, \dots, x_{99}$ )**

`@variable(model, x[i=1:9, j=1:9], Bin)`. Após resolver o modelo, o valor da variável  $x_{ij}$  podem ser obtido assim: `sol[i,j] = value.(x[i,j])`

OBS: Atenção para distinguir uma variável de Julia com uma variável do modelo matemático. Variáveis do modelo matemático são declaradas com `@variable(param1, param2, param3)`, onde param1 se refere ao nome modelo, param2 se refere ao nome da variável e param3 ao tipo da variável.