MDIO - Trabalho Prático

D. Alves A79751, D. Braga A82547, J. Silva A82005, R. Caçador A81064

December 2018

Contents

1 Questão 1

1.a

Dados:
n - ordem da matriz
m - número de nodos da matriz
cNorte - Matriz de custos de ir para Norte
cSul - Matriz de custos de ir para Sul
cEste - Matriz de custos de ir para Este
cOeste - Matriz de custos de ir para Oeste
c_{ij} - custo associado ao arco da origem i para o destino j,

$$i \in \{1, ..., m\}, j \in \{1, ..., m\}$$

Variáveis de decisão:
x_{ij} - indica o número de caminhos de que faz parte o arco ij,

$$i \in \{1, ..., m\}, j \in \{1, ..., m\}$$

Função Objetivo:

$$MinZ = \sum_{i=1}^{m} \sum_{j=1}^{m} c_{ij} \times x_{ij}$$

Sujeito a:

$$x_{ii} = 0, i \in \{1, ..., m\}$$

$$\sum_{j=1}^{m} (x_{1j} - x_{j1}) = m - 1$$

$$\sum_{j=1}^{m} (x_{ij} - x_{ji}) = -1, i \in \{2, ..., m\}$$

$$x_{ij} \geq 0, i \in \{1, ..., m\}, j \in \{1, ..., m\}$$

Em anexo, na figura 7, segue o grafo utilizado como exemplo no exercício.

Explicação da Função Objetivo

A nossa função objetivo tem como propósito minimizar o instante de chegada do fogo a cada nodo assumindo que o nodo de ignição é o nodo 1.

Explicação das Restrições

A primeira restrição verifica que não existem arcos no qual a origem é igual ao destino.

A segunda restrição verifica que o número de caminhos que sai do primeiro nodo é o valor do número de nodos total menos um, sendo este denominado nodo de ignição. Por exemplo, na instância em causa:

$$x_{18} + x_{12} - x_{21} - x_{81} = 48$$

A terceira restrição verifica que o número de caminhos que saem num nodo menos os que entram é igual a -1. Isto acontece porque um dos caminhos é contabilizado no nodo em causa. Por exemplo, na instância em causa:

Para o nodo 2:

$$x_{23} + x_{29} + x_{21} - x_{32} - x_{39} - x_{31} = -1$$

Para o nodo 8:

$$x_{81} + x_{89} + x_{8,15} - x_{18} - x_{98} - x_{15,8} = -1$$

Para o nodo 11:

$$x_{11,4} + x_{11,10} + x_{11,12} + x_{11,18} - x_{4,11} - x_{10,11} - x_{12,11} - x_{18,11} = -1$$

Para o nodo 39:

$$x_{39,32} + x_{39,38} + x_{39,40} + x_{39,46} - x_{32,39} - x_{38,39} - x_{40,39} - x_{46,39} = -1$$

Para o nodo 49:

$$x_{49,42} + x_{49,48} - x_{42,49} - x_{48,49} = -1$$

1.b

Dados:

n - ordem da matriz
m - número de nodos da matriz
cNorte - Matriz de custos de ir para Norte
cSul - Matriz de custos de ir para Sul
cEste - Matriz de custos de ir para Este
cOeste - Matriz de custos de ir para Oeste
c_{ij} - custo atribuido ao arco ij,

$$i \in \{1, ..., m\}, j \in \{1, ..., m\}$$

Variáveis de decisão:

y_i - indica o tempo que o fogo demora a chegar à célula i

Função Objetivo:

$$MaxZ = \sum_{i=2}^{m} y_i$$

Sujeito a:

$$y_1 = 0$$
$$y_j - y_i \leq c_{ij}, i \in \{1, ..., m\}, j \in \{1, ..., m\}$$
$$y_i \geq 0, i \in \{1, ..., m\}$$

Explicação da Função Objetivo

A nossa função objetivo tem como propósito maximizar o tempo de chegada às células.

Explicação das Restrições

A primeira restrição verifica que o tempo que o fogo demora a chegar à célula 1 é 0, uma vez que esta é a célula de ignição.

A segunda restrição verifica que o tempo que demora ir de uma célula para outra tem de ser menor que o custo desse mesmo arco.

1.c

Na seguinte figura 1 é possível verificar a solução ótima primal. Importante referir que os valores dos arcos representam o número de caminhos de que estes fazem parte.

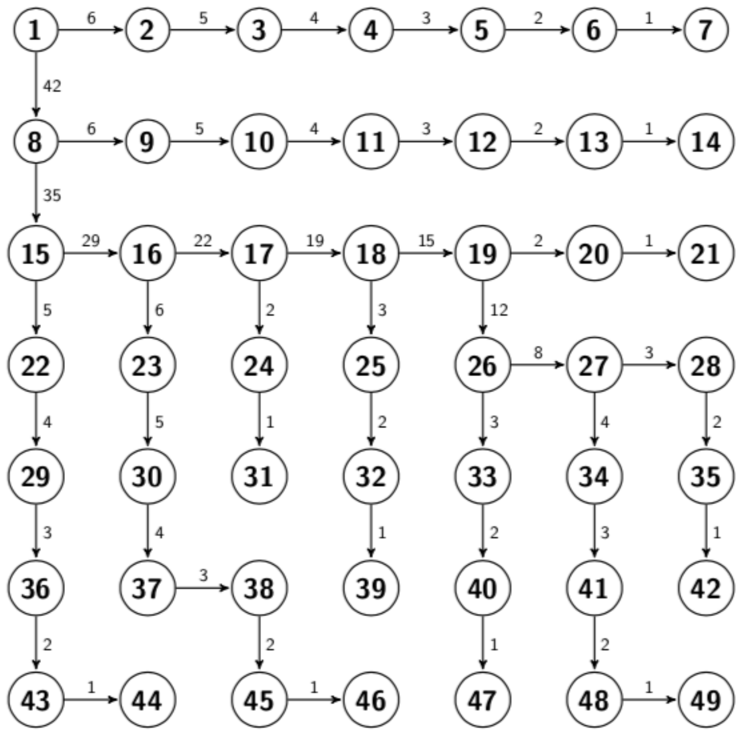

Figure 1: Solução ótima do Problema Primal

1.d

Na seguinte figura 2 é possível verificar a solução ótima dual. Importante referir que os valores dos arcos representam o tempo que o fogo demora a chegar ao nodo a que este se destina.

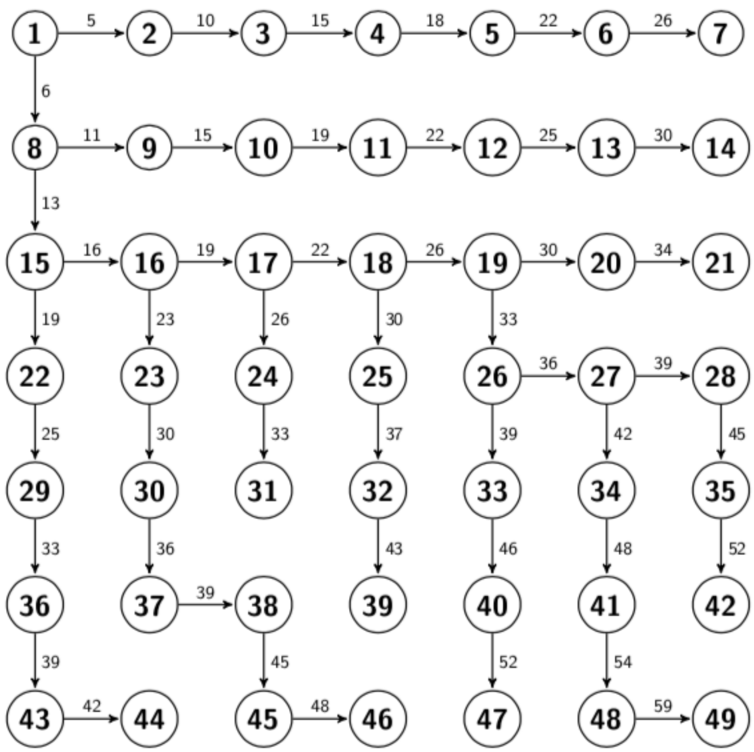

Figure 2: Solução ótima do Problema Dual

2 Questão 2

2.a

Dados:

n - ordem da matriz
m - número de nodos da matriz
cNorte - Matriz de custos de ir para Norte
cSul - Matriz de custos de ir para Sul
cEste - Matriz de custos de ir para Este
cOeste - Matriz de custos de ir para Oeste
retardamento - quantidade de tempo acrecentado na propagação para os nodos adjacentes
nodoProtegido - nodo que se pretende que aumente o tempo de chegada do fogo
recursos - número de recursos disponíveis para proteger um determinada nodo
c_{ij} - custo atribuido ao arco ij,

$$i \in \{1, ..., m\}, j \in \{1, ..., m\}$$

Variáveis de decisão:

x_i - tempo que o fogo demora a chegar ao nodo i,

$$i \in \{1, ..., m\}$$

$$y_j = \begin{cases} 1, & \text{se recurso é colocado no nodo j} \\ 0, & \text{caso contrário} \end{cases}$$

$$j \in \{1, ..., m\}$$

Função Objetivo:

$$MaxZ = x_{nodoProtegido}$$

Sujeito a:

$$x_1 = 0$$

$$\sum_{i=1}^{m} y_i \leq recursos$$

$$x_j - x_i \leq c_{ij} + y_i \times retardamento, i \in \{1, ..., m\}, j \in \{1, ..., m\}$$

$$x_i \geq 0, i \in \{1, ..., m\}$$

Explicação da Função Objetivo

A nossa função objetivo tem como propósito maximizar o tempo de chegada à célula que pretendemos proteger.

Explicação das Restrições

A primeira restrição verifica que o tempo que o fogo demora a chegar à célula 1 é 0, uma vez que esta é a célula de ignição.

A segunda restrição verifica que os recursos usados não são maiores que o total de recursos disponibilizados.

A terceira restrição verifica que ao tempo que demora ir de uma célula para a outra, é adicionado um retardamento se a célula for protegida com um recurso.

2.b

Tendo em conta que o número em causa é o 82547, o maior elemento é o 8. Desta forma, são 8 os recursos disponíveis, e tambem 8 o retardamento colocado em cada nodo selecionado para proteger. O nodo escolhido para proteger é o 49 (linha 7, coluna 7).

Executando o modelo em causa, a solução ótima acontece quando são colocados os recursos nos nodos **1, 2, 8, 35, 41, 42, 47 e 48**.

Na figura 3 representa-se essa mesma solução, sendo que cada nodo possui um par com o seu número e o tempo que o fogo demora a chegar a esse local, tendo em conta o retardamento.

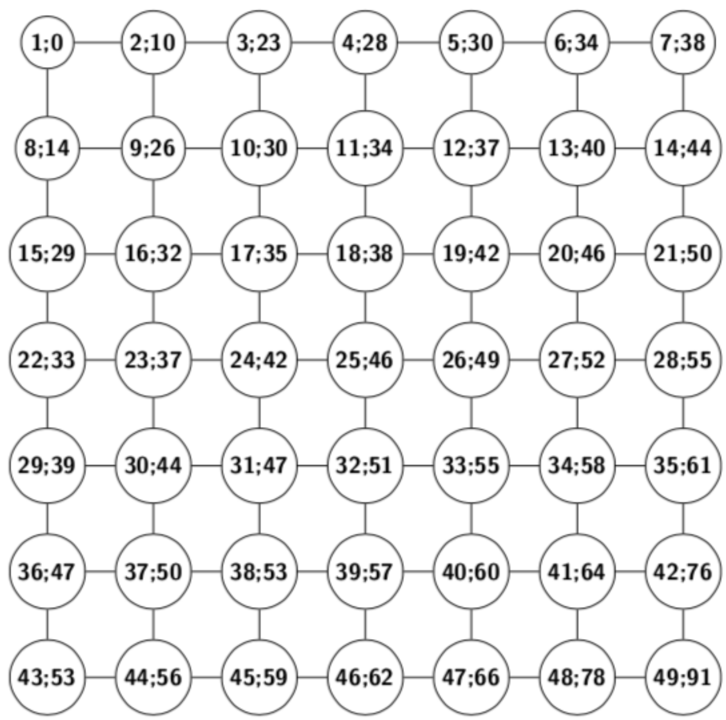

Figure 3: Solução ótima

2.c

Na pequena amostra representada pelo gráfico, onde apenas consideramos o intervalo de valores para os recursos entre 8 e 20, repara-se que existe uma tendência de aumento do tempo de chegada do fogo ao nodo protegido quanto maior for o número de recursos utilizados.

Prevê-se que o valor do tempo continue a aumentar, quantos mais recursos forem utilizados.

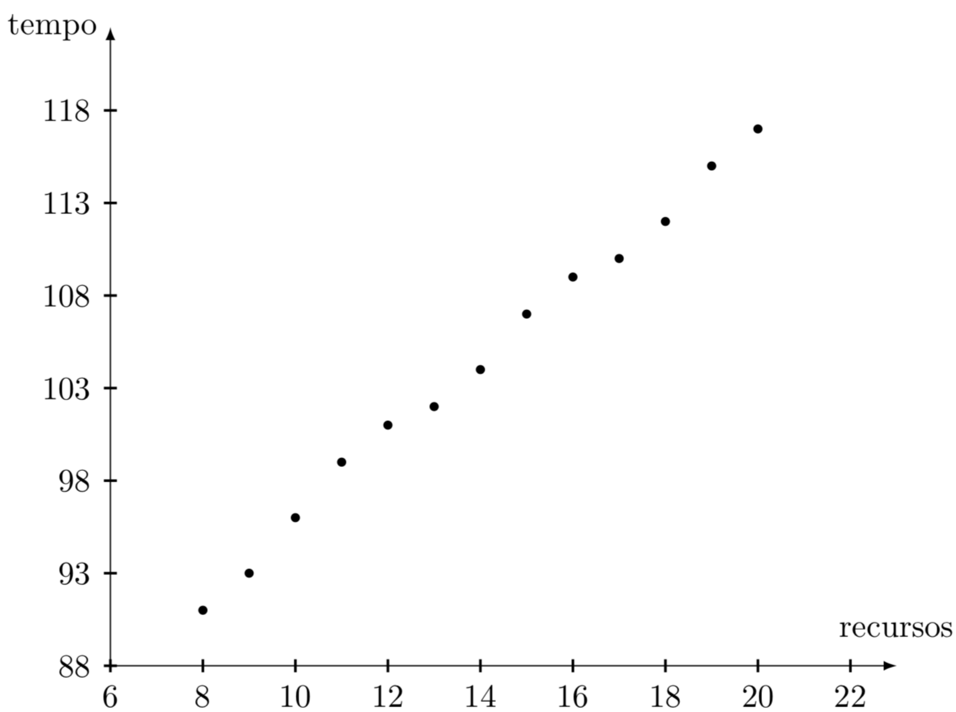

Figure 4: Gráfico que representa o tempo de chegada do fogo ao nodo protegido.

3 Questão 3

3.a

Dados:
n - ordem da matriz
m - número de nodos da matriz
cNorte - Matriz de custos de ir para Norte
cSul - Matriz de custos de ir para Sul
cEste - Matriz de custos de ir para Este
cOeste - Matriz de custos de ir para Oeste
intervalo - intervalo de tempo
retardamento - quantidade de tempo acrecentado na propagação para os nodos adjacentes
$prob_i$ - probabilidade de ignição

$$i \in \{1, ..., m\}$$

recursos - número de recursos disponíveis para proteger um determinada nodo
c_{ij} - custo atribuido ao arco ij,

$$i \in \{1, ..., m\}, j \in \{1, ..., m\}$$

Variáveis de decisão:

$$x_i = \begin{cases} 1, & \text{se célula i está queimada} \\ 0, & \text{caso contrário} \end{cases} \quad i \in \{1, ..., m\}$$

$$y_i = \begin{cases} 1, & \text{se célula i está protegida} \\ 0, & \text{caso contrário} \end{cases} \quad i \in \{1, ..., m\}$$

t_i - tempo que o fogo demora a chegar à célula i,

$$i \in \{1, ..., m\}$$

Função Objetivo:

$$MinZ = \sum_{i=1}^{m} (prob_i \times x_i)$$

Sujeito a:

$$t_1 = 0$$

$$t_j - t_i \leq c_{ij} + y_i \times retardamento, i \in \{1, ..., m\}, j \in \{1, ..., m\}$$

$$\sum_{i=1}^{m} y_i \leq recursos$$

$$x_i \geq (intervalo - t_i)/intervalo, i \in \{1, ..., m\}$$

$$t_i \geq 0, i \in \{1, ..., m\}$$

Explicação da Função Objetivo

A nossa função objetivo tem como propósito minimizar o valor esperado da área ardida tal como pedido no enunciado do problema. Por isso, utilizamos a função **minimize** , no contexto do problema analisamos a probabilidade de ignição de cada célula (**prob[i]**) e multiplicamos por **x[i]**, que é a variável responsável por verificar se uma célula está ou não ardida.

Explicação das Restrições

A primeira restrição usa a variável **t**, que indica o tempo que o fogo demora a chegar a uma célula, sendo na primeira célula que se inicia o fogo o tempo de chegada a esta é zero.

A segunda restrição verifica que o tempo de chegada a uma célula tem de ser sempre menor ou igual ao tempo que o fogo demora de uma célula para outra. O tempo que o fogo demora de uma célula para a outra é somado ao retardamento, no caso de haver um recurso a proteger a célula, por isso **(retardamento*y[i])** pois **y[i]** verifica se a célula está protegida será igual a 1 em caso afirmativo e o retardamento será maior que 0.

A terceira restrição usa a variável **y** , que indica se uma célula está protegida ou não , e usa o somatório de **y[i]** para indicar que o total do mesmo tem de ser menor ou igual ao número de recursos, pois é dito no enunciado que o número de células protegidas não pode ser superior ao número de recursos.

Na quarta restrição se o fogo chegar a uma célula é porque o **intervalo** é maior que o **t[i]**, o lado direito da equação será um número positivo menor que 1 mas será lido como 1 devido a **x[i]** ser uma variável binária, e isto significa que essa mesma célula ardeu. Se a função do lado direito for ¡1 pois o **t[i]** é maior que o **intervalo**, o valor assumido será zero e significa que essa célula não ardeu.

3.b

Resolvendo o modelo através do software, concluimos que as **células queimadas** são a **1** e a **49** e a única **célula ardida** é a **1**. Na figura 5 representa-se essa mesma solução, sendo que cada nodo possui um par com o seu número e o tempo que o fogo demora a chegar a esse local.

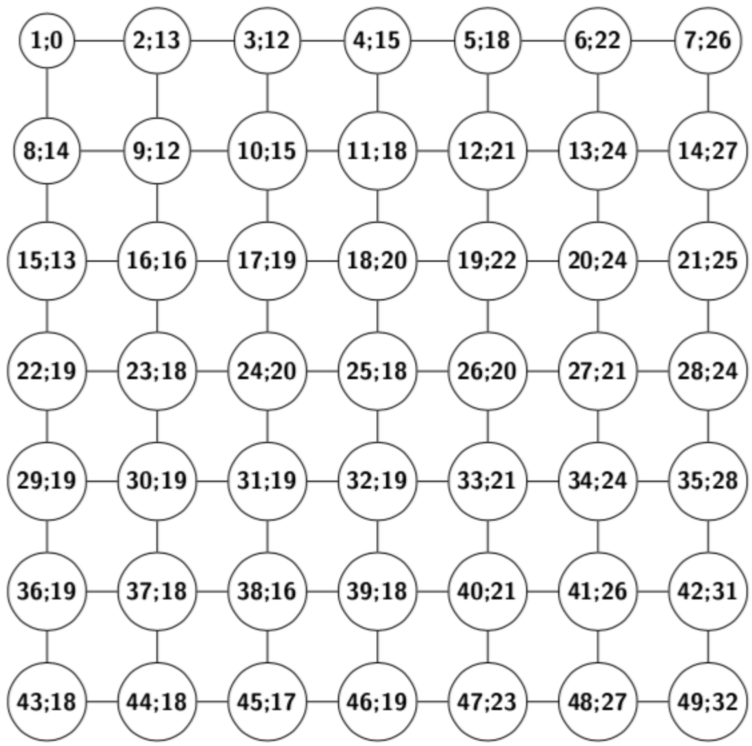

Figure 5: Tempos da Solução ótima

3.c

Na pequena amostra representada pelo gráfico, onde apenas consideramos o intervalo de valores para os intervalos entre 12 e 24, repara-se que existe uma tendência de aumento da área ardida quanto maior for o o intervalo de tempo escolhido.

Prevê-se que o valor da área ardida continue a aumentar, quanto maior for o intervalo de tempo escolhido.

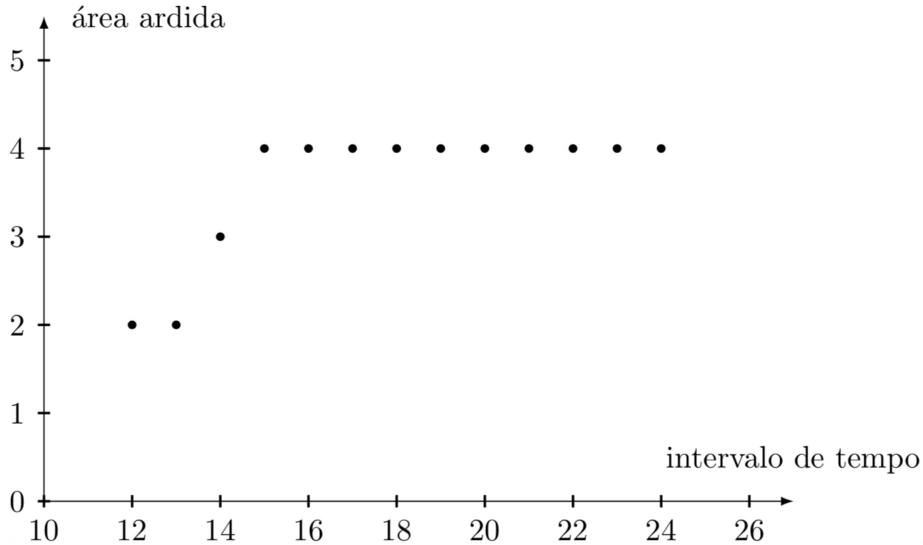

Figure 6: Gráfico que representa a área ardida em função do intervalo de tempo usado.

4 Anexo

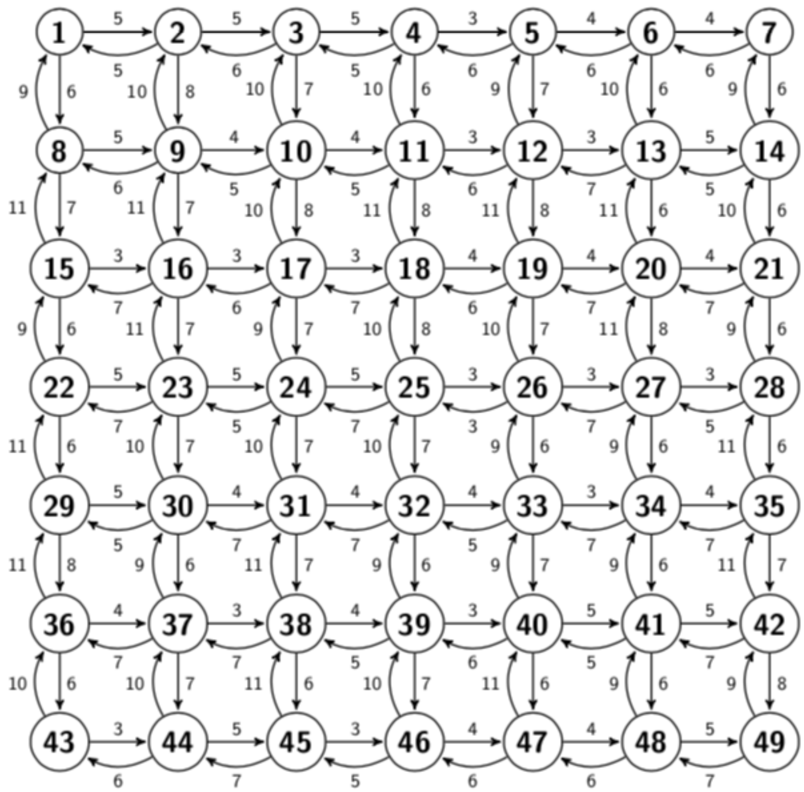

Figure 7: Instância utilizada como exemplo

Execute usado no exercício 1a, 1b e 2a:

```
for(var a = 1; a <= m; a++){
  for(var b = 1; b <= m; b++){
    var div = Opl.ceil(a/n);
    var res = a%n;
    if(res == 0) res = n;
    if((a - b) == n && div != 1) custo[a][b] = cNorte[div][res];
    else if((b - a) == n && div != n) custo[a][b] = cSul[div][res];
    else if((b - a) == 1 && res != n) custo[a][b] = cEste[div][res];
    else if((a - b) == 1 && res != 1) custo[a][b] = cOeste[div][res];
    else custo[a][b] = 1000000;
  }
}
```

Execute usado no exercício 3a:

```
for(var a = 1; a <= m; a++){
  for(var b = 1; b <= m; b++){
    var div = Opl.ceil(a/n);
    var res = a%n;
    if(res == 0) res = n;
    prob[a] = (14 - div - res)/500;
    if((a - b) == n && div != 1) custo[a][b] = cNorte[div][res];
    else if((b - a) == n && div != n) custo[a][b] = cSul[div][res];
    else if((b - a) == 1 && res != n) custo[a][b] = cEste[div][res];
    else if((a - b) == 1 && res != 1) custo[a][b] = cOeste[div][res];
    else custo[a][b] = 1000000;
  }
}
```