

a. Formule o problema de Jake como um problema do fluxo de custo mínimo e desenhe a rede para esse problema. Identifique os nós de suprimento e de demanda para a rede.

Modelagem:

[Variáveis]

x_{ij} : Fluxo de dinheiro em U\$ da moeda i para moeda j

[Função Objetivo]

Minimizar o custo total das transações

(min) $\sum (i, j) C_{ij} \cdot X_{ij}$

Em que:

C_{ij} é o custo de transação por unidade de U\$ ao converter da moeda i para a moeda j.

X_{ij} é a quantidade em U\$ convertida da moeda i para a moeda j.

[Condições]

Capacidade das arestas:

$X_{ij} \leq \text{Capacidade}_{ij}$ (Capacidade das arestas)

Balanco de Fluxo:

Fonte:

$\sum_i X_{kj} - f = 0$ para todo k pertencente às fontes. f é o valor daquela fonte.

Sumidouro:

$f - \sum_i X_{ik} = 0$ para todo k pertencente ao sumidouro. f é a soma das fontes.

Intermediários:

$\sum_i X_{kj} - \sum_i X_{ik} = 0$ para todo k intermediário

Que pode ser resumido para:

$\sum_i X_{ik} - \sum_i X_{kj} = S_k$ para todo k

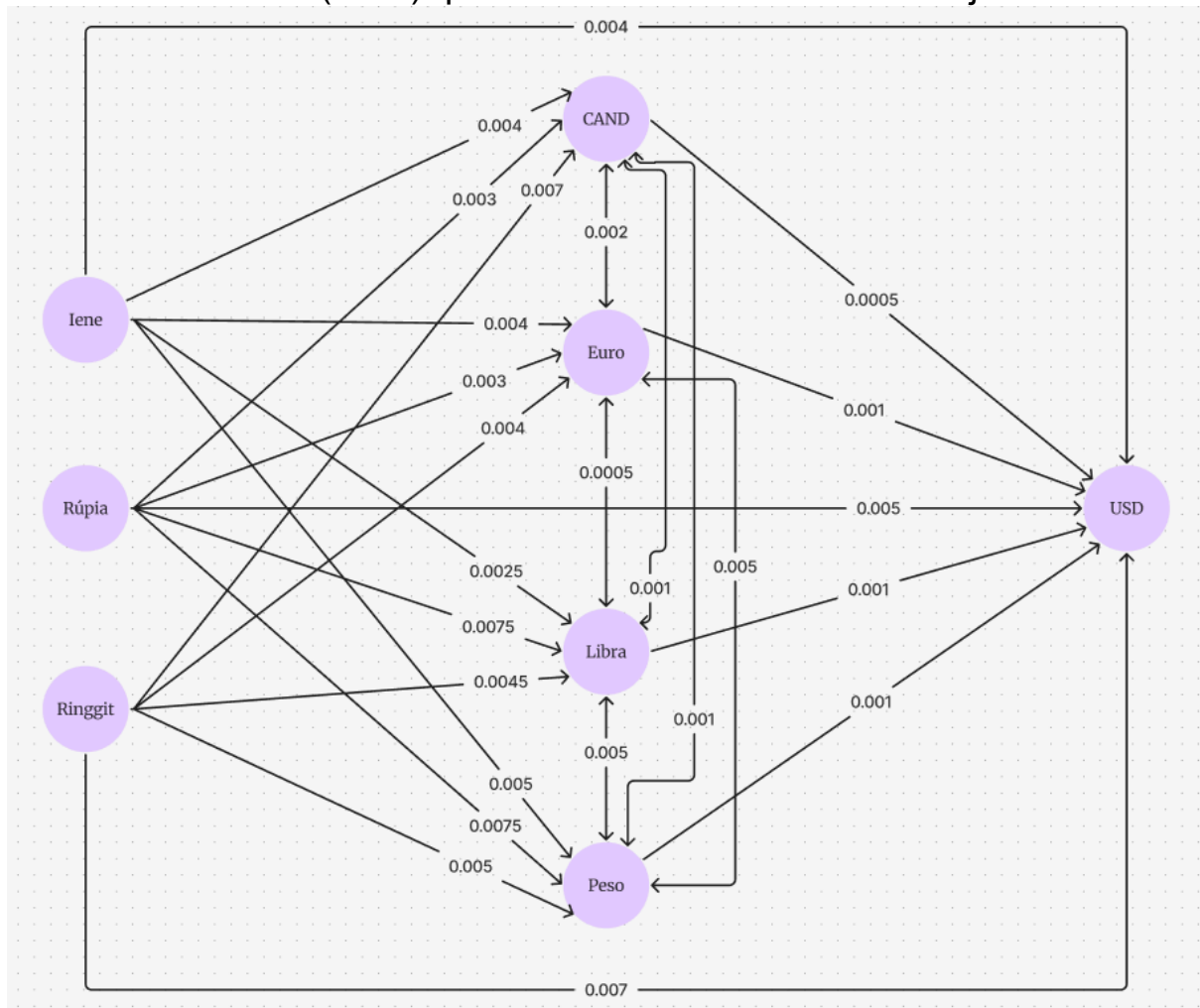
Em que:

k : nó/moeda

S_k : Suprimento líquido no nó k ($-f$ para fonte; $+f$ para sumidouro e 0 para intermediário).

Identificação dos nós de suprimento e demanda:

- Nós de Suprimento (Fonte): Representam as moedas asiáticas que a Grant Hill Associates possui.
- Nós de demanda (Sumidouro): Representam os dólares norte-americanos (USD) que a Grant Hill Associates deseja obter.



b. Quais transações em moeda Jake deve realizar de modo a converter os investimentos feitos em ienes, rúpias e ringgits em dólares norte-americanos para garantir que a Grant Hill Associates tenha a quantia máxima em dólares após todas as transações terem sido feitas? Quanto dinheiro Jake tem de investir em obrigações do mercado norteamericano?

Primeiramente precisamos saber quantos U\$ a empresa possui atualmente investido em moedas asiáticas. O problema nos informa que ela possui 10.5 bilhões de rúpias e 28 milhões de ringgits. Utilizando a tabela de taxa de câmbio disponível no problema, obtemos que a empresa possui $10.500.000.000 * 0.00016 = \text{U\$}1.680.000$ investidos em rúpias e $28.000.000 * 0.2 = \text{U\$}5.600.000$ investidos em ringgits. Como o problema informa que Jake havia levado os investimentos da empresa para U\$15.000.000 de dólares investidos em ienes com a cotação 1 dólar = 80 ienes, sabemos que a empresa possui 1.2 bilhões de ienes. Como a cotação atualmente é de 1 dólar = 125 ienes, obtemos que a empresa possui $1.200.000.000/125 = \text{U\$}9.600.000$. A empresa possui então U\$16.880.000 investidos no total.

Implementamos um solver com o modelo “GLOP” do ortools baseado na modelagem do problema. O solver recebe a matriz de custos e capacidade, que são as seguintes:

```
Costs = [
#
#           |R|      |D|      |D|
#           |i|      |l|      |l|
#           |n|      |a|      |a|      |L|
#           |e|      |p|      |g|      |r|      |r|      |E|      |i|      |P|
#           |n|      |i|      |i|      |N|      |C|      |r|      |r|      |s|
#           |e|      |a|      |t|      |A|      |A|      |o|      |a|      |o|
#
# [ 0.0000, 0.0050, 0.0050, 0.0040, 0.0040, 0.0040, 0.0025, 0.0050 ], #Iene 0
# [ 0.0050, 0.0000, 0.0070, 0.0050, 0.0030, 0.0030, 0.0075, 0.0075 ], #Rúpia 1
# [ 0.0050, 0.0070, 0.0000, 0.0070, 0.0070, 0.0040, 0.0045, 0.0050 ], #Ringgit 2
# [ 0.0040, 0.0050, 0.0070, 0.0000, 0.0005, 0.0010, 0.0010, 0.0010 ], #Dólar Norte Americano 3
# [ 0.0040, 0.0030, 0.0070, 0.0005, 0.0000, 0.0020, 0.0010, 0.0010 ], #Dólar Canadense 4
# [ 0.0040, 0.0030, 0.0040, 0.0010, 0.0020, 0.0000, 0.0005, 0.0050 ], #Euro 5
# [ 0.0025, 0.0075, 0.0045, 0.0010, 0.0010, 0.0005, 0.0000, 0.0050 ], #Libra 6
# [ 0.0050, 0.0075, 0.0050, 0.0010, 0.0010, 0.0050, 0.0050, 0.0000 ] #Peso 7
]
```

Como os custos estavam em porcentagem, dividimos todos os valores por 100

```
Capacity = [
#
#           |D|      |D|
#           |ó|      |ó|
#           |i|      |l|      |l|
#           |R|      |n|      |a|      |a|      |L|
#           |I|      |ú|      |g|      |r|      |r|      |E|      |i|      |P|
#           |e|      |p|      |g|      |r|      |r|      |u|      |b|      |e|
#           |n|      |i|      |i|      |N|      |C|      |r|      |r|      |s|
#           |e|      |a|      |t|      |A|      |A|      |o|      |a|      |o|
#
# [ 0, 5, 5, 2, 2, 2, 2, 2, 4 ], #Iene
# [ 5, 0, 2, 0.2, 0.2, 1, 0.5, 0.2 ], #Rúpia
# [ 3, 4.5, 0, 1.5, 1.5, 2.5, 1, 1 ] #Ringgit
]
```

Dividimos cada capacidade por 1000 para trabalharmos com a capacidade em milhões de U\$ (Ela estava sendo trabalhada em mil U\$), já que também vamos trabalhar com a quantidade de fluxo em milhões de U\$.

Definimos então as variáveis de acordo com o modelo:

```
#Definição de Variáveis
x = []
for i in range(n):
    x.append([])
    for j in range(n):
        x[i].append(solver.NumVar(0, solver.infinity(), f"x{i}{j}"))
```

Então definimos a função objetivo de minimização:

```
#Função Objetivo de Minimização
solver.Minimize(solver.Sum(Costs[i][j]*x[i][j] for i in range(n) for j in range(n)))
```

Depois definimos as condições de capacidade baseado no modelo:

```
#Capacidade das arestas
for i in range(3):
    for j in range(n):
        solver.Add(x[i][j] <= Capacity[i][j])
```

Definimos então as condições de balanceamento de fluxo:

```
#Fontes
solver.Add(x[0][3] + x[0][4] + x[0][5] + x[0][6] + x[0][7] - 9.6 == 0) #Iene
solver.Add(x[1][3] + x[1][4] + x[1][5] + x[1][6] + x[1][7] - 1.68 == 0) #Rupia
solver.Add(x[2][3] + x[2][4] + x[2][5] + x[2][6] + x[2][7] - 5.6 == 0) #Riggit

#Sumidouro
solver.Add(16.88 - x[0][3] - x[1][3] - x[2][3] - x[4][3] - x[5][3] - x[6][3] - x[7][3] == 0)

#Intermediario
solver.Add(x[4][3] + x[4][5] + x[4][6] + x[4][7] - x[0][4] - x[1][4] - x[2][4] - x[5][4] - x[6][4] - x[7][4] == 0)
solver.Add(x[5][3] + x[5][4] + x[5][6] + x[5][7] - x[0][5] - x[1][5] - x[2][5] - x[4][5] - x[6][5] - x[7][5] == 0)
solver.Add(x[6][3] + x[6][4] + x[6][5] + x[6][7] - x[0][6] - x[1][6] - x[2][6] - x[4][6] - x[5][6] - x[7][6] == 0)
solver.Add(x[7][3] + x[7][4] + x[7][5] + x[7][6] - x[0][7] - x[1][7] - x[2][7] - x[4][7] - x[5][7] - x[6][7] == 0)
```

Dividimos os valores que saem da fonte e chegam no sumidouro por 1 milhão para trabalhar em milhões de U\$.

Explicação das condições:

Para as fontes: O valor em milhões de U\$ que a empresa possui em cada moeda é justamente o valor que irá sair da fonte para ser convertido. Como apresentado anteriormente no modelo, toda carga que sai do nó tem que ser igual ao valor que entrou, no caso das fontes, o valor que entrou é justamente o valor inicial.

Para o sumidouro: O sumidouro vai receber a soma de todos os valores que estão saindo das fontes, que no caso é o total de milhões de U\$ investidos nas moedas.

Para os vértices intermediários: Como explicado nos itens anteriores, todo valor que sai tem que ser igual ao valor que entra, dessa forma toda a carga é distribuída, sem ficar nenhum dinheiro investido em outra moeda.

Por fim, pedimos para o solver resolver o problema. Ele retorna então este resultado:

```
Solução:
Função Objetivo = 0.0834 milhões de U$
Transações de Ienes (em Milhões de U$)
Iene para Rúpia: 0.00
Iene para Ringgit: 0.00
Iene para Dólar Canadense: 2.00
Iene para Euro: 2.00
Iene para Libra: 2.00
Iene para Peso: 1.60
Iene para Dólar Norte Americano: 2.00
-----
Transações de Rúpia (em Milhões de U$)
Rúpia para Iene: 0.00
Rúpia para Ringgit: 0.00
Rúpia para Dólar Canadense: 0.20
Rúpia para Euro: 1.00
Rúpia para Libra: 0.08
Rúpia para Peso: 0.20
Rúpia para Dólar Norte Americano: 0.20
-----
Transações de Ringgit (em Milhões de U$)
Ringgit para Iene: 0.00
Ringgit para Rúpia: 0.00
Ringgit para Dólar Canadense: 0.00
Ringgit para Euro: 2.50
Ringgit para Libra: 1.00
Ringgit para Peso: 1.00
Ringgit para Dólar Norte Americano: 1.10
-----
Transações Intermediárias (em Milhões de U$)
Dólar Canadense para Dólar Americano 2.20
Euro para Dólar Norte Americano 5.50
Libra para Dólar Norte Americano 3.08
Peso para Dólar Norte Americano 2.80
```

Isso significa que o custo total dessas transações é de U\$83.400, resultando em um total de $U\$16.880.000 - U\$83.400 = U\$16.796.600$.

c. A Organização Mundial do Comércio (OMC) proíbe limites sobre transações, pois elas promovem o protecionismo. Se não existissem limites de transação, que método Jake deveria usar para converter em dólares as posições atuais nas respectivas moedas asiáticas?

Para resolvermos esse problema podemos simplesmente utilizar o mesmo código que implementamos anteriormente, mas dessa vez não adicionamos as condições de capacidade. Então obtemos o seguinte resultado:

```
Solução:
Função Objetivo = 0.0675 milhões de U$
Transações de Ienes (em Milhões de U$)
Iene para Rúpia: 0.00
Iene para Ringgit: 0.00
Iene para Dólar Canadense: 0.00
Iene para Euro: 0.00
Iene para Libra: 9.60
Iene para Peso: 0.00
Iene para Dólar Norte Americano: 0.00
-----
Transações de Rúpia (em Milhões de U$)
Rúpia para Iene: 0.00
Rúpia para Ringgit: 0.00
Rúpia para Dólar Canadense: 1.68
Rúpia para Euro: 0.00
Rúpia para Libra: 0.00
Rúpia para Peso: 0.00
Rúpia para Dólar Norte Americano: 0.00
-----
Transações de Ringgit (em Milhões de U$)
Ringgit para Iene: 0.00
Ringgit para Rúpia: 0.00
Ringgit para Dólar Canadense: 0.00
Ringgit para Euro: 5.60
Ringgit para Libra: 0.00
Ringgit para Peso: 0.00
Ringgit para Dólar Norte Americano: 0.00
-----
Transações Intermediárias (em Milhões de U$)
Dólar Canadense para Dólar Americano 1.68
Euro para Dólar Norte Americano 5.60
Libra para Dólar Norte Americano 9.60
Peso para Dólar Norte Americano 0.00
```

O custo total dessas transações é de U\$67.500, resultando em um total de U\$16.880.000 - U\$67.500 = U\$16.812.500. A empresa estaria economizando U\$15.900 caso não existissem essas medidas protecionistas.

d. Em resposta ao mandado da OMC proibindo limites sobre transações, o governo da Indonésia introduz um novo imposto que leva a um aumento de 500% nos custos de transação para transações de rúpias visando proteger a moeda local. Dados esses novos custos de transação, mas sem limites sobre as transações, que transações de câmbio Jake deveria realizar de modo a converter suas posições em moedas asiáticas das respectivas moedas para dólares?

Reutilizando o código anterior (Sem as restrições de capacidade), precisamos apenas atualizar os custos de transação da moeda de rúpia. Para isso implementamos uma simples função:

```
#Aumenta o custo da moeda index em tax%
def SetTax(Costs, index, tax):
    n = len(Costs)
    for i in range(n):
        Costs[index][i] *= tax/100
```

Adicionamos a chamada da função ao código:

```
#Índice referente a rúpia é 1
SetTax(Costs, 1, 500)
```


E então obtemos:

```
Solução:
Função Objetivo = 0.0876 milhões de U$
Transações de Ienes (em Milhões de U$)
Iene para Rúpia: 0.00
Iene para Ringgit: 0.00
Iene para Dólar Canadense: 0.00
Iene para Euro: 0.00
Iene para Libra: 9.60
Iene para Peso: 0.00
Iene para Dólar Norte Americano: 0.00
-----
Transações de Rúpia (em Milhões de U$)
Rúpia para Iene: 0.00
Rúpia para Ringgit: 0.00
Rúpia para Dólar Canadense: 1.68
Rúpia para Euro: 0.00
Rúpia para Libra: 0.00
Rúpia para Peso: 0.00
Rúpia para Dólar Norte Americano: 0.00
-----
Transações de Ringgit (em Milhões de U$)
Ringgit para Iene: 0.00
Ringgit para Rúpia: 0.00
Ringgit para Dólar Canadense: 0.00
Ringgit para Euro: 5.60
Ringgit para Libra: 0.00
Ringgit para Peso: 0.00
Ringgit para Dólar Norte Americano: 0.00
-----
Transações Intermediárias (em Milhões de U$)
Dólar Canadense para Dólar Americano 1.68
Euro para Dólar Norte Americano 5.60
Libra para Dólar Norte Americano 9.60
Peso para Dólar Norte Americano 0.00
```

O custo total dessas transações é de U\$87.600, resultando em um total de $U\$16.880.000 - U\$87.600 = U\$16.792.400$. A empresa estaria perdendo U\$4.200 em consequência da resposta do governo da Malásia.