Relatório de Trabalho
Modelagem do problema do fluxo máximo

Lucas Lima de Araújo
Carlos Augusto Pereira Maia
Matheus Fernandes de Souza

CENTRO DE INFORMÁTICA
UNIVERSIDADE FEDERAL DA PARAÍBA

João Pessoa, 2021

LISTA DE FIGURAS

Sumário

1 INTRODUÇÃO

O Problema de Fluxo Máximo é um clássico problema de modelagem da área da Pesquisa Operacional. Amplamente presente em situações do cotidiano (fluxo de mercadorias, por exemplo), o PFM consiste na elaboração de um modelo que represente o fluxo máximo em uma rede de fluxo. Um caso particular do PFM é o Problema de Fluxo de Custo Mínimo, amplamente utilizado em diversas áreas da sociedade para reduzir custos no transporte de bens, por exemplo. Este relatório possui o fim de apresentar a definição de um PFCM, bem como sua resolução através da modelagem matemática e da programação linear.

2 DEFINIÇÃO DO PROBLEMA

O projeto constitui-se em algumas partes, sendo elas: introdução, modelagem do PFM para o PFCM e o exercício 9.4-3, que se encontra na página 395 do referente livro. Abaixo é demonstrado a definição geral de um PFM.

Um grafo direcionado G = (V, A) consiste de um conjunto V de vértices v, também chamados de nós, e um conjunto A de arcos a, onde cada arco é um par ordenado (i, j) de vértices. O vértice i do par é fonte do arco e o vértice j o alvo.

Sabe-se que:

• O fluxo atraves de uma rede direcionada e conectada origina-se de um **nó**, denominado origem e termina em outro nó, denominado **escoadouro**.
• Os nós restantes são chamados **nós de transbordo**
• Tem como objetivo maximizar a quantidade total de fluxo da origem até o escoadouro. Essa quantidade é medida em qualquer uma das duas maneiras equivalentes, ou seja, a quantidade que sai da origem ou, então, a quantidade que chega ao escoadouro.

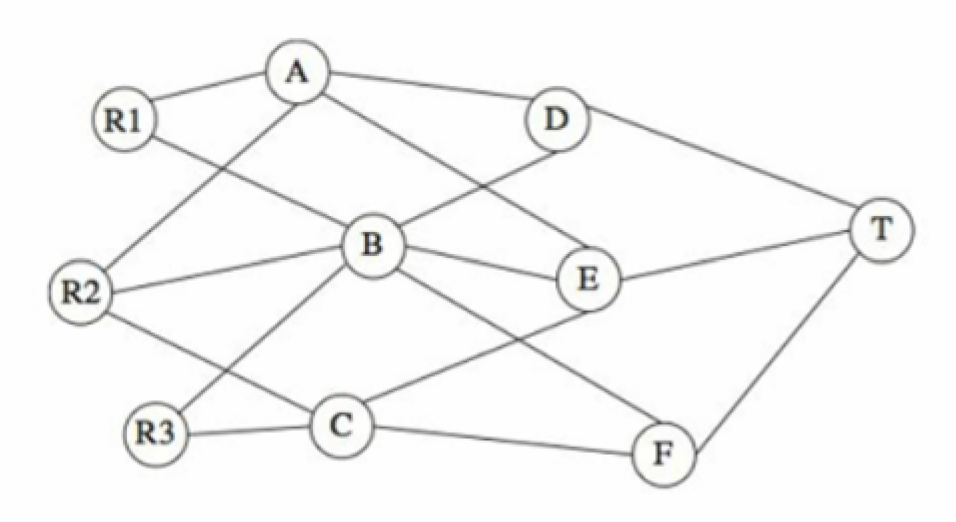

Figura 1: Grafo do problema 9.4-3, resolvido ao fim desse relatório, como exemplo de uma rede PFM.

3 MODELAGEM

Nesta seção é descrita os passos necessários paraa efetuar a modelagem do problema.

3.1 Modelagem do PFM

Modelando o problema como um PFM obtém-se o seguinte grafo:

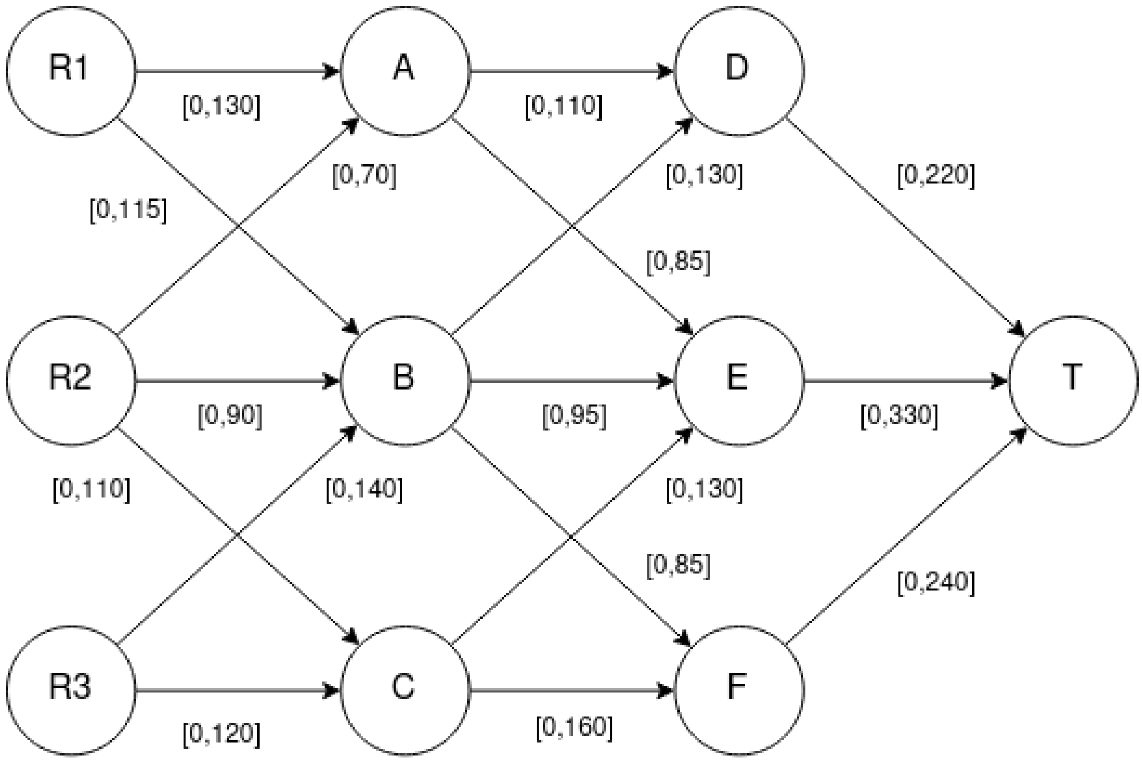

Figura 2: Grafo do modelo PFM

3.2 Transformando o PFM para PFCM

Segundo [1] para transformar um PFM em um PFCM deve-se fazer as seguintes alterações no problema original:

1. Atribuir um custo $c =\ 0$ para todos os arcos existentes
2. Atribuir a todos os nós ,de origem, escoadouro e transbordo, demanda de valor 0
3. Estabelecer um valor $\overline{F}$ de fluxo viável máximo para a rede
4. Criar um arco que vai diretamente do nó de suprimento para o nó de demanda, e atribuir uma oferta e uma demanda de valor $\bar{F}$ para os nós de suprimento e demanda,

respectivamente. Atribuir um valor de custo M grande e um valor de capacidade U infinito.

Efetuando as devidas alterações, obtém-se o seguinte grafo

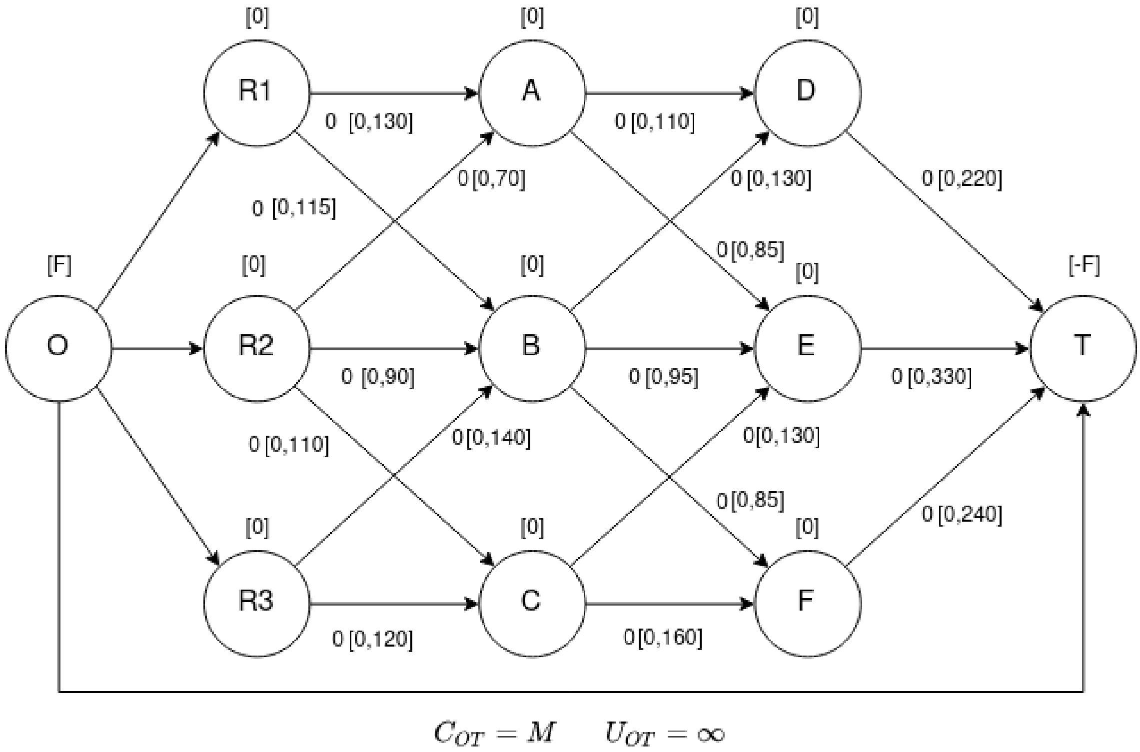

Figura 3: Grafo do modelo PFCM

3.3 Modelagem matemárica do PFCM

As fórmulas que descrevem o PFCM acima são as seguintes

Min.

$$\sum_{(i,j)\in A} c_{ij}x_{ij}$$

S.a.

$$\sum_{(j,i)\in A} x_{ji} - \sum_{(i,j)\in A} x_{ij} = b_i;\ \forall\ i \in A$$

$$0 \leq x_{ij} \leq u_{ij}\ \forall\ i, j \in A$$

4 INSTRUÇÕES

Nesta seção será descrito como instalar o pacote de ferramentas com o solver utilizado, e também como utilizar o código desenvolvido.

4.1 Instalando o OR-Tools

Os comandos abaixo foram utilizados e testados no sistema operacional Ubuntu 20.04 LTS.

Instalando python 3.6:

```
$ sudo apt-get install python3-dev python3-pip python3-venv python3-six
```

Instalando o OR-Tools:

```
$ python3 -m pip install -U --user ortools
```

4.2 Utilizando o código

Para utilizar o código desenvolvido é necessário definir um arquivo no seguinte formato:

Formato do arquivo

```
n #numero de vertices (vertices numerados de 1 a n)
m #numero de arcos (arcos numerados de 1 a m)
s #indice da origem
t #indice do escoadouro
i j c_1 #dados do arco 1
...
i j c_m #dados do arco m
```

Figura 4: Formato do arquivo com entrada de dados

Em seguida, deve-se abrir o arquivo com o código python e alterar a sexta linha que contém uma variável String de nome fileName que receberá o caminho e o nome do arquivo definido como entrada de dados.

Por exemplo:

```
fileName = 'instancias/instance1.txt'
```

Por fim, basta rodar o comando que vai executar o script python:

```
$ python3 [nomeDoArquivoDeCodigo].py
```

4.3 O código

```
from __future__ import print_function
from ortools.graph import pywrapgraph

# Nome e caminho do arquivo:

fileName = 'instancias/instance1.txt'

def main():

  #
  # LEITURA DO ARQUIVO
  #

  arq = open(fileName)

  verticesNumber = int(arq.readline())
  arcsNumber     = int(arq.readline())
  sourceNode     = int(arq.readline())
  sinkNode       = int(arq.readline())

  lines = [0]*arcsNumber

  for i in range(arcsNumber):
      lines[i] = arq.readline().split(' ')

  arq.close()

  #
  # FIM DA LEITURA DO ARQUIVO
  #

  #
```

```
# INÍCIO DA MODELAGEM
#

# Somando +1 para considerar uma das etapas da modelagem de PFM para PFCM
# em que se adiciona um arco da origem ao escoadouro

totalArcs = arcsNumber + 1
totalVertices = verticesNumber + 1

# Declaração dos arrays que o Solver usa como entrada de dados

start_nodes = [0]*totalArcs
end_nodes   = [0]*totalArcs
capacities  = [0]*totalArcs
unit_costs  = [0]*totalArcs
supplies    = [0]*totalVertices

# Declaração de variáveis que vão auxiliar no processo de modelagem
capacitiesSum = 0

# Atribuindo os valores lidos do arquivo aos arrays de entrada
# de dados do Solver

for i in range(arcsNumber):
    start_nodes[i] = int(lines[i][0])
    end_nodes[i]   = int(lines[i][1])
    capacities[i]  = int(lines[i][2])

    # PRIMEIRA ETAPA DA MODELAGEM ONDE Cij = 0
    unit_costs[i]  = 0

    # Somatório utilizado para dar o chute dos valores de capacidade,
    # custo e suprimento do arco da modelagem
    capacitiesSum += capacities[i]


# SEGUNDA ETAPA DE MODELAGEM: ATRIBUINDO UM LIMITE SUPERIOR SEGURO
# COMO OFERTA E DEMANDA DO NÓ DE INÍCIO E NÓ ESCOADOURO
for i in range(verticesNumber):
```

```
    supplies[i] = 0

supplies[sourceNode] = capacitiesSum
supplies[sinkNode]   = -capacitiesSum

# TERCEIRA ETAPA DA MODELAGEM, ADICIONANDO UM ARCO QUE VAI DO NÓ INICIAL
# AO NÓ ESCOADOURO
start_nodes[totalArcs - 1] = sourceNode
end_nodes[totalArcs - 1]   = sinkNode

# CONTINUANDO TERCEIRA ETAPA DE MODELAGEM, ATRIBUINDO CAPACIDADE "INFINITA"
# E CUSTO ALTO AO ARCO ADICIONADO ANTERIORMENTE
capacities[totalArcs - 1]  = capacitiesSum*3
unit_costs[totalArcs - 1]  = capacitiesSum*3

print('Start Nodes:')
print(start_nodes)
print('End Nodes:')
print(end_nodes)
print('Capacities:')
print(capacities)
print('Unit Costs:')
print(unit_costs)
print('Supplies:')
print(supplies)

#
# FIM DA MODELAGEM
#

# Instanciando o solver SimpleMinCostFlow.
min_cost_flow = pywrapgraph.SimpleMinCostFlow()

# Adicionando os arcos
for i in range(0, len(start_nodes)):
  min_cost_flow.AddArcWithCapacityAndUnitCost(start_nodes[i], end_nodes[i],
                                            capacities[i], unit_costs[i])

# Adicionando os nós
```

```
  for i in range(0, len(supplies)):
    min_cost_flow.SetNodeSupply(i, supplies[i])

  # Variável usada para calcular Z
  z = 0

  # Encontrando o fluxo de custo mínimo
  if min_cost_flow.Solve() == min_cost_flow.OPTIMAL:
    print('')
    print('  Arc    Flow / Capacity  Cost')
    for i in range(min_cost_flow.NumArcs()):
      cost = min_cost_flow.Flow(i) * min_cost_flow.UnitCost(i)
      if(not(min_cost_flow.Tail(i) == sourceNode
      and min_cost_flow.Head(i) == sinkNode)
      and min_cost_flow.Flow(i) != 0):
        print('%1s -> %1s   %3s  / %3s       %3s' % (
            min_cost_flow.Tail(i),
            min_cost_flow.Head(i),
            min_cost_flow.Flow(i),
            min_cost_flow.Capacity(i),
            cost))
      if(min_cost_flow.Head(i) == sinkNode):
        if(min_cost_flow.Tail(i) != sourceNode):
          z += min_cost_flow.Flow(i)
  else:
    print('There was an issue with the min cost flow input.')

  print('')
  print('Z = ', z)
  print('')

if __name__ == '__main__':
  main()
```

4.4 Implementação da modelagem

Como pode ser visto no código e nos comentários, a modelagem de PFM para PFCM foi feita logo após a leitura do arquivo de entrada. A primeira etapa foi setar os custos dos arcos para 0. A segunda etapa foi definir um limite superior seguro como oferta e demanda do nó inicial e nó escoadouro respectivamente. Nesse passo, foi usada a soma das capacidades de todos os arcos para definir esse valor. Por fim, a última etapa foi adicionar um arco que vai do nó inicial ao nó escoadouro com capacidade e custo muito altos. Para definir esses valores foi utilizado a soma das capacidades multiplicadas por 3 que geram um valor grande o suficiente.

4.5 Resultados

Segue um exemplo de utilização do código mostrado. Primeiramente será mostrado o arquivo que contém as entradas de dados.

```
7
12
1
7
1 2 10
1 3 8
1 4 3
2 3 4
2 5 4
3 5 8
3 6 2
4 3 3
4 5 7
4 6 9
5 7 10
6 7 10
```

Figura 5: Arquivo de entrada de dados: instance1.txt

Com a entrada de dados definida, e modificada a variável que guarda o nome e caminho desse arquivo, é executado o código e como resultado, obtém-se:

```
Start Nodes:
[1, 1, 1, 2, 2, 3, 3, 4, 4, 4, 5, 6, 1]
End Nodes:
[2, 3, 4, 3, 5, 5, 6, 3, 5, 6, 7, 7, 7]
Capacities:
[10, 8, 3, 4, 4, 8, 2, 3, 7, 9, 10, 10, 234]
Unit Costs:
[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 234]
Supplies:
[0, 78, 0, 0, 0, 0, 0, -78]

  Arc     Flow / Capacity  Cost
1 -> 2      5  /  10          0
1 -> 3      7  /   8          0
1 -> 4      3  /   3          0
2 -> 3      3  /   4          0
2 -> 5      2  /   4          0
3 -> 5      8  /   8          0
3 -> 6      2  /   2          0
4 -> 6      3  /   9          0
5 -> 7     10  /  10          0
6 -> 7      5  /  10          0

Z =  15
```

Figura 6: Saída do programa

O primeiro trecho da saída exibe 5 arrays que são os arrays que o solver recebe como entrada e eles já estão modelados de PFM para PFCM. Eles representam os nós iniciais, nós finais, capacidades, custo unitário e suprimentos. Assim, cada posição desses arrays representa um arco. Por exemplo: A primeira posição do array de nós iniciais tem o valor 1, enquanto que no array de nós finais tem o valor 2, isso significa um arco que vai do nó 1 para o nó 2, com capacidade 10 (primeira posição do array capacidade), e custo unitário 0 (mesma lógica que os anteriores). Já no array de suprimentos cada índice representa a oferta ou demanda do nó respectivo. Nesse caso, o nó 1 oferta 78 e o nó 7 demanda 78 (por isso está negativo).

O segundo trecho da saída exibe os arcos e seus respectivos fluxos (mostra apenas os que forem diferentes de 0) encontrados pelo solver. Por fim, exibe o valor de Z que representa o fluxo máximo.

5 EXERCÍCIO

5.1 Resolução da questão 9-4-3

Nesta seção iremos apresentar a solução do item ”a”referente a questão 9-4-3 do livro ”Introdução à Pesquisa Operacional”(LIEBERMAN, p.395).

(a) Formule esse problema como um problema do fluxo máximo identificando uma origem, um escoadouro e os nós de transbordo, e depois desenhe a rede completa que mostra a capacidade de cada arco.

Figura 7: Descrição da letra ’a’

Modelando o problema como um PFM obtém-se o seguinte grafo:

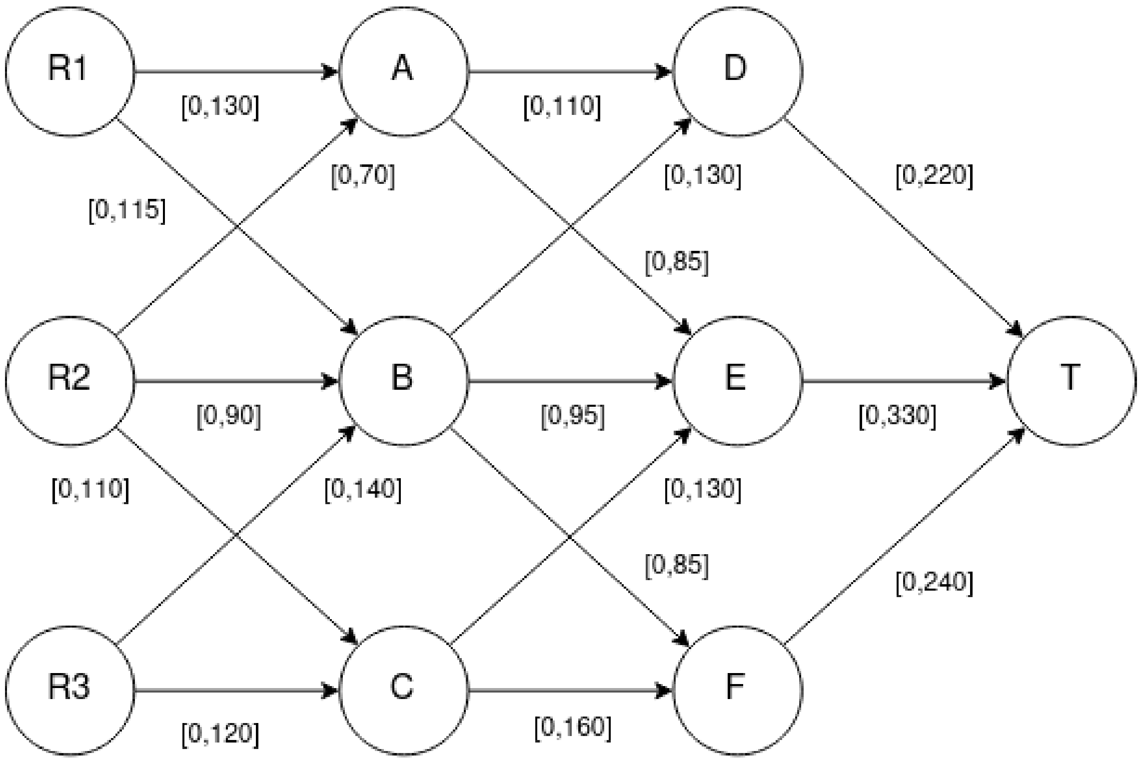

Figura 8: Grafo do modelo do PFM

5.2 Identificando os vértices

De acordo com a figura 6, os nós possuem a identificação que segue:

- Os nós R1,R2 e R3 são nós de origem

- O nó T é o nó de transbordo
- Os demais nós são escoadouros

5.3 Modelagem matemática do PFM

Dado um grafo G = (V,A), onde:

- A o conjunto de arcos do grafo G
- V o conjunto de vértices do grafo G
- 'o' é a origem
- 't' é a origem

As fórmulas que descrevem o PFM são as seguintes:

$$\textbf{Max.} \sum_{(o,j)\in A} x_{oj}$$

$$\textbf{Sa.} \sum_{(j,i)\in A} x_{ji} - \sum_{(i,j)\in A} x_{ij} = 0;\ \forall\ i \in\ V/\left\{o,t\right\}$$

REFERÊNCIAS

[1] HILLIER, F.; LIEBERMAN, G. Introdução à Pesquisa Operacional. 9. ed. , 2012.