Investigação Operacional
Problema de Fluxos em Rede

2º Trabalho

May 11, 2024

Trabalho realizado por:

Gonçalo Rocha Sousa Freitas
Henrique Guimarães Pereira
Eduarda Mafalda Martins Vieira
Maria Leonor Carvalho da Cunha

A104350
Gonçalo Rocha Sousa Freitas

A97205
Henrique Guimarães Pereira

A104098
Eduarda Mafalda Martins Vieira

A103997
Maria Leonor Carvalho da Cunha

Contents

1 Introdução

Este trabalho aborda um problema de fluxo máximo entre dois vértices não adjacentes. Considerando um grafo G=(V,A), onde V é o conjunto de vértices e A é o conjunto de arestas,procuramos determinar o fluxo máximo entre dois vértices não-adjacentes, designados por O (origem) e D (destino). Neste contexto, o objetivo é encontrar a distribuição de fluxo que maximize a quantidade de fluxo que pode ser transferida de O para D, obedecendo a certas restrições. No caso em questão, o fluxo em uma aresta pode ocorrer em qualquer um dos dois sentidos, e a capacidade das arestas é considerada virtualmente infinita. Além disso, os vértices do grafo têm capacidade, exceto O e D, que são a origem e o destino do fluxo, respetivamente.

2 Alterações ao Problema

O maior número presente no grupo, é o 104350.
Isto significa que o x = 1, A = 0, B = 4, C = 3, D = 5, E = 0
Assim, conseguimos determinar a capacidade de cada vértice e qual o vértice de origem e o de destino.

Capacidade dos Vértices:

vértices	capacidade
1	$+\infty$
2	100
3	40
4	60
5	$+\infty$
6	60

Conforme os itens disponibilizados, ficamos com origem(O) no ponto 1 e destino(D) no ponto 5.
Calculo da capacidade de cada vértice:
2. 10 x (4 + 5 + 1) = 100
3. 10 x (3 + 1) = 40
4. 10 x (5 + 1) = 60
6. 10 x (5 + 0 + 1) = 60

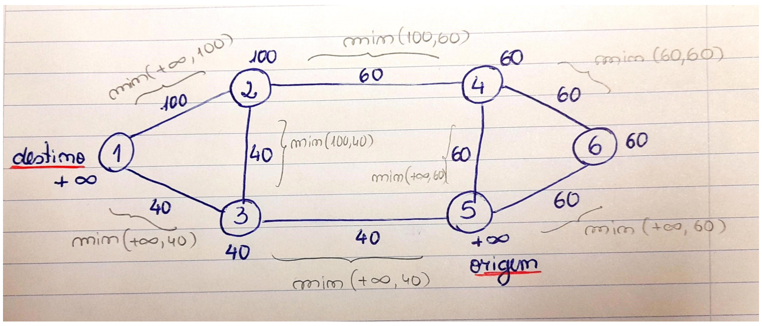

Figure 1: Grafo Resultante

3 Formulação do Problema

Neste projeto estudamos de forma a descobrir o fluxo máximo possível entre dois vértices não adjacentes. Para isso, possuímos algumas restrições de dois tipos. Neste caso em questão, o fluxo em uma aresta pode ocorrer em qualquer um dos dois sentidos, e a capacidade das arestas é considerada virtualmente infinita. Além disso, os vértices do grafo têm capacidade, exceto O e D, que são a origem e o destino do fluxo, respetivamente. De modo a resolver o problema utilizamos um modelo de programação na qual usamos:

- O custo unitário do fluxo cij do arco (i, j), (i, j) A;
- A capacidade uij do arco (i, j), (i, j) A;
- Oferta ou consumo em cada vértice bj, j V.

Assim, o vértice de origem (vértice 5) apenas poderá enviar fluxo e, o vértice de destino (vértice 1), apenas poderá receber fluxo. Concluímos portanto que, estes dois vértices referidos em cima, têm capacidade $+\infty$

Os vértices restantes têm capacidades variadas (podemos verificar as várias capacidades na figura 2, abaixo, em cima de cada vértice, sublinhado a laranja).

Calculamos através do mínimo entre as capacidades de dois vértices que forma uma aresta, o máximo de fluxo de uma aresta (máximo de fluxo que pode ser enviado entre dois vértices). Não pode haver um fluxo que ultrapasse a capacidade de um dos vértices.

Os valores estão representados na figura 2, sublinhados a roxo, e as respetivas indicações de contas a lápis (ex.: entre os vértices 2 e 4 $\rightarrow min(100, 60) = 60$).

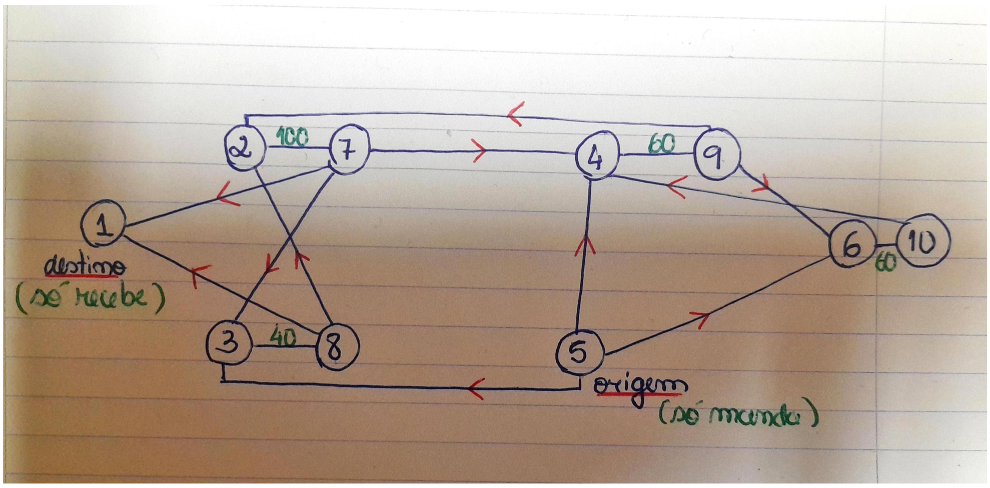

Figure 2: Grafo que representa o Fluxo

4 Modelo de Programação de Fluxos de Rede

Face ao problema e dados apresentados, determinamos o custo dos arcos na figura abaixo, através da função *min*, onde se escolhe o menos número do par apresentado.

Quanto à oferta/consumo em cada vértice, o grupo decidiu colocar todos os valores a 0, uma vez que não existe procura positiva nem negativa nos vértices.

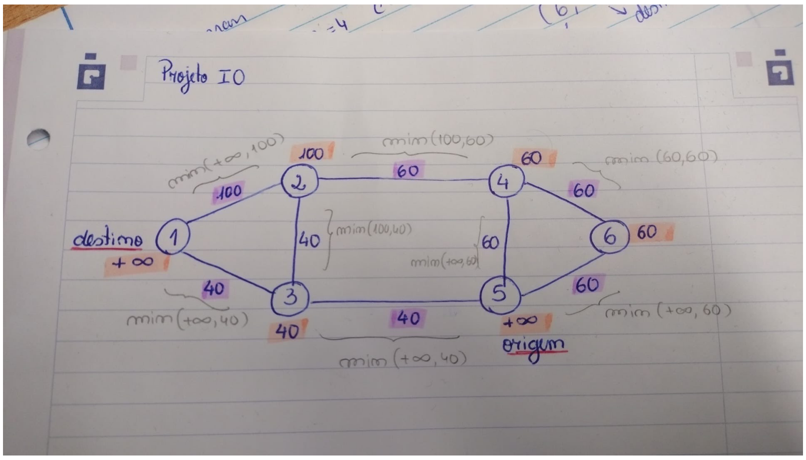

Figure 3: Grafo que representa as capacidades dos vértices

5 Ficheiro Input e Output

Ficheiro input:
10
16

7 1 0 100
7 4 0 60
2 7 0 100
7 3 0 40
8 2 0 40
8 1 0 40
3 8 0 40
9 2 0 60
9 6 0 60
4 9 0 60
5 4 0 60
5 6 0 60
5 3 0 40
10 4 0 60
6 10 0 60
1 5 -1 1000
0

0
0
0
0
0
0
0
0
0

Ficheiro output:
NUMBER OF NODES = 10, NUMBER OF ARCS = 15
DEFAULT INITIALIZATION USED
****
Total algorithm solution time = 0.00377082825 sec.
OPTIMAL COST = -100.
NUMBER OF ITERATIONS = 5
NUMBER OF MULTINODE ITERATIONS = 1
NUMBER OF MULTINODE ASCENT STEPS = 0
NUMBER OF REGULAR AUGMENTATIONS = 1
****

f 7 1 0
f 7 4 0
f 2 7 0
f 7 3 0
f 8 2 0
f 8 1 0
f 3 8 0
f 9 2 0
f 9 6 0
f 4 9 0
f 5 4 0
f 5 6 0
f 5 3 0
f 10 4 0
f 6 10 0

6 Solução Ótima

Ao analisar o *output* do programa *relax*4 conseguimos tirar uma conclusão de quais os trajetos admissíveis:

- 5 – 3 – 1
- 5 – 4 – 2 – 1

7 Validação do Modelo

O valor ótimo é 100. No output do Relax4, o valor que é apresentado é negativo, porque o Relax4 trabalha com mínimos. No entanto, o valor que pretendemos é um máximo e por isso utilizamos o seu simétrico.

Sabendo que o fluxo máximo entre dois vértices não adjacentes é de 100, indica que a capacidade do arco entre esses vértices não pode ultrapassar esse valor, uma vez que representa sua capacidade máxima. Somando as capacidades dos arcos que saem da origem o fluxo máximo, não poderá exceder 100 (60+40).

```
/* Objective function */
max: x51;

/* conservação de fluxo */
VERTICE1: - 1 x21 - 1 x31 + 1 x12 + 1 x13 = -1 x51;
VERTICE2: - 1 x32 - 1 x42 - 1 x12 + 1 x21 + 1 x23 + 1 x24 = 0;
VERTICE3: - 1 x13 - 1 x23 - 1 x53 + 1 x31 + 1 x32 + 1 x35 = 0;
VERTICE4: - 1 x54 - 1 x64 - 1 x24 + 1 x42 + 1 x45 + 1 x46 = 0;
VERTICE5: - 1 x35 - 1 x45 + 1 x53 + 1 x54 = 1 x51;
VERTICE6: - 1 x46 - 1 x56 + 1 x64 + 1 x65 = 0;

/* capacidade dos arcos */
ARCO12: x12 <= 100;
ARCO21: x21 >= 0;
ARCO13: x13 <= 40;
ARCO31: x31 >= 0;
ARCO23: x23 <= 40;
ARCO32: x32 <= 100;
ARCO35: x35 >= 0;
ARCO53: x53 <= 40;
ARCO24: x24 <= 60;
ARCO42: x42 <= 100;
ARCO45: x45 >= 0;
```

Figure 4: Input Lpsolve

```
ARCO54: x54 <= 60;
ARCO46: x46 <= 60;
ARCO64: x64 <= 60;
ARCO56: x56 <= 60;
ARCO65: x65 >= 0;
/* Variable bounds */
```

Figure 5: Input Lpsolve (resto)

A função objetivo é x51 porque a origem é 5 e o destino é 1.

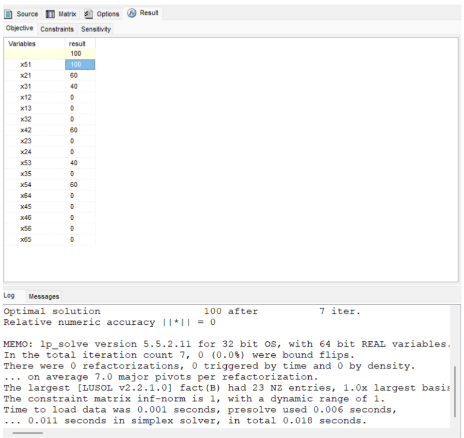

Figure 6: Output Lpsolve

Como podemos observar na figura 6, o output do lpsolve é igual ao valor ótimo do Relax4. 100 é o valor do fluxo máximo entre dois vértices não adjacentes.