Universidade Do Minho
Engenharia Informática
Investigação Operacional
A91677 - Vicente Castro; A91971 - Nuno Pereira
A94662 - Bruno Ribeiro; A100600 - Diogo Barros

TP2:
Problema do Fluxo Máximo

11/05/2024

1 - Formulação do Problema

O problema consiste em determinar o fluxo máximo entre dois vértices não adjacentes, denominados **O (origem) e D (destino), num grafo G = (V, A).** Este grafo é composto por um conjunto de vértices (V) e um conjunto de arestas (A), onde cada aresta representa duas possíveis rotas de fluxo entre dois vértices.

O **Problema de Fluxo Máximo** consiste na **maior quantidade de fluxo, que pode ser transportada de uma origem (O), para um destino (D)** dentro do grafo, respeitando as capacidades das arestas e as restrições do problema.

Com tudo isto e tendo em conta que o maior nº de aluno do grupo é **100600(xABCDE)**, obtivemos a seguinte tabela das capacidades **(oferta e/ou consumo)** de cada vértice para pudermos iniciar o nosso trabalho prático:

VÉRTICE	CAPACIDADE
1 (ORIGEM)	10 X (0+6+1) = **70, LOGO É INFINITA**
2	10 X (0+0+1) = **10**
3	10 X (6+1) = **70**
4 (DESTINO)	10 X (0+1) = **10, LOGO É INFINITA**
5	10 X (0+1) = **10**
6	10 X (0+0+1) = **10**

Fig. 1 - Tabela com as capacidades dos vértices.

NOTA: quando dizemos **(LOGO É INFINITA)**, pretendemos dizer que apesar dos vértices **1 e 4** terem uma capacidade definida através de cálculos, estes estão identicados como a origem e o destino do nosso grafo **(k = DE mod(14) = 0** $\times$ **14 + 0 = 0, ou seja, 0: (1,4))**, logo a sua capacidade passa a estar definida como **INFINITA.**

Seguindo as regras, conseguimos selecionar todos os caminhos possíveis que o fluxo pode percorrer, e acabamos por obter o seguinte **grafo de compatibilidades**.

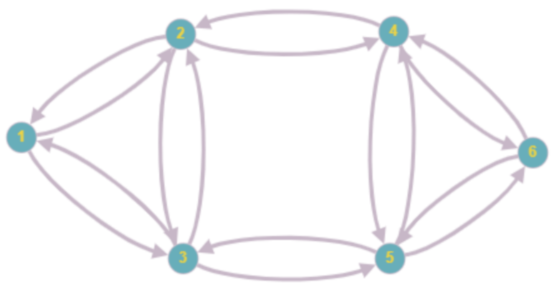


Fig. 2 - Grafo de Compatibilidades.

2 - Modelo do Problema de Fluxo Máximo

Neste tipo de problemas, existem ligações feitas entre um determinado vértice e o seu adjacente, com uma determinada direção. Estas ligações chamam-se de **arcos.** Cada arco possui uma **capacidade**, que limita a quantidade de fluxo que pode passar por ele.

Inicialmente, o nosso problema tinha capacidades infinitas atribuídas aos nossos **arcos**. Em seguida, verificamos que tinhamos capacidades que estavam apenas, atribuídas aos nossos **vértices.** Logo, para resolver este problema, tivemos de definir arcos, que representavam apenas **um vértice.**

Para compreender melhor estes arcos criados para o problema das capacidades, aqui está uma tabela de como funcionam:

VÉRTICE INICIAL	CAPACIDADE DO VÉRTICE	VÉRTICE FINAL CRIADO	ARCO EXEMPLO (DIREÇÃO)	CAPACIDADE DO ARCO
2	10	7	2 → 7	10
3	70	8	3 → 8	70
5	10	9	5 → 9	10
6	10	10	6 → 10	10

Fig. 3 - Tabela dos arcos de um vértice e as suas capacidades.

Depois de ultrapassado este problema das representações das capacidades dos vértices em arcos, obtivemos o seguinte grafo:

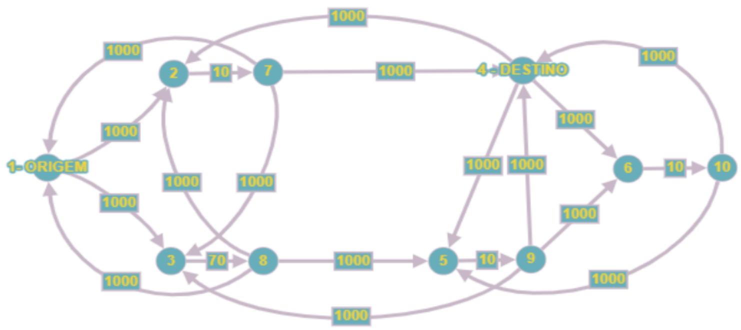


Fig. 4 - Grafo resultante com os seus arcos e suas capacidades.

NOTA: 1000 é igual a infinito, por causa do software **Relax4.**

Agora com o grafo desenhado com sucesso, vamos explicar alguns dos seus caminhos:

- **Caminho de 2 para 7:** neste caminho, o máximo que pode ser enviado, através deste arco, é **10**. Este valor foi atribuído a este arco, para respeitar a capacidade máxima do vértice **2**;
- **Caminho de 3 para 8:** neste caminho, o máximo que pode ser enviado, através deste arco, é **70**. Este valor foi atribuído a este arco, para respeitar a capacidade máxima do vértice **3**;
- **Caminho de 1 para 2:** neste caminho, o arco encontra-se com capacidade de **1000**, simbolizando que a capacidade máxima é infinita, para depois ser representada com um valor no software Relax4. Este valor foi atribuído a este arco, porque o **vértice 1 não tem capacidade definida (infinita);**
- **Caminho de 7 para 4:** neste caminho, o arco encontra-se com capacidade de **1000**, simbolizando que a capacidade máxima é infinita, para depois ser representada com um valor no software Relax4. Este valor foi atribuído a este arco, porque o **vértice 4 não tem capacidade definida (infinita).**

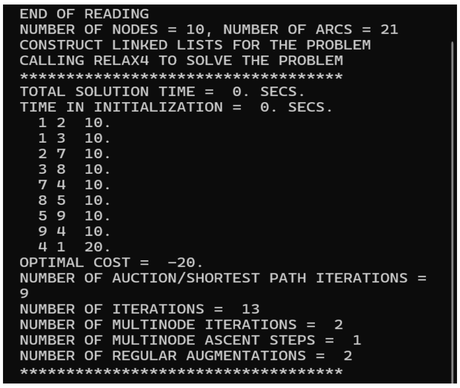
```
10
21
1 2 0 1000
1 3 0 1000
7 1 0 1000
8 1 0 1000
2 7 0 10
7 3 0 1000
8 2 0 1000
3 8 0 70
4 2 0 1000
7 4 0 1000
8 5 0 1000
9 3 0 1000
4 5 0 1000
5 9 0 10
9 4 0 1000
10 5 0 1000
9 6 0 1000
4 6 0 1000
10 4 0 1000
6 10 0 10
4 1 -1 1000
0
0
0
0
0
0
0
0
0
0
```

Fig. 5 - Input do software do Relax4.

Depois de atribuir todas as capacidades aos arcos, de maneira a respeitar as capacidades dos vértices, reunimos todas as informações e criamos este input da **Fig. 5**, para o software do Relax4. Aqui temos a explicação de algumas das partes mais importantes do input:

- **Linha 1:** nesta linha temos o número **10**, que representa o número total de vértices que o nosso grafo contém;
- **Linha 2:** nesta linha temos o número **21**, que representa o número total de arestas que o nosso input contém, sendo **20** arestas desse valor total, o número de ligações diretas entre os vértices do nosso grafo;
- **Linha 3:** nesta linha temos o número **1**, que representa o vértice origem do arco, temos o número **2**, que representa o vértice destino do arco, temos o número **0**, que representa o custo do arco, visto que para iniciar este tipo de problema todos os custos são iniciados com zero, para serem todos calculados com sucesso, e por fim, temos o número **1000**, que representa a capacidade infinita deste arco;
- **Linha 23:** nesta linha temos o número **4**, que representa o vértice origem do arco, temos o número **1**, que representa o vértice destino do arco. O software Relax4 assume que todos os problemas são de **minimização.** Portanto, para maximizar o fluxo **(que é o objetivo do problema)**, devemos associar ao arco de retorno um custo unitário de transporte igual a **-1.** Isto significa que enviar uma unidade de fluxo por este arco resulta em uma economia de uma unidade, o que incentiva o aumento do fluxo.
- **Linhas 24 até 33:** nestas linhas temos 10 números **0**, que representam a oferta e a demanda de cada um dos **10 vértices do grafo**. Esses valores são zero, porque como dito anteriormente este tipo de problemas de fluxo máximo, para calcular o fluxo máximo, necessita que todos os custos sejam nulos, pois vão ser calculados e demonstrados pelo output do Relax4.

3 - Solução Ótima do Modelo



```
END OF READING
NUMBER OF NODES = 10, NUMBER OF ARCS = 21
CONSTRUCT LINKED LISTS FOR THE PROBLEM
CALLING RELAX4 TO SOLVE THE PROBLEM
**********************************
TOTAL SOLUTION TIME =  0. SECS.
TIME IN INITIALIZATION =  0. SECS.
   1 2  10.
   1 3  10.
   2 7  10.
   3 8  10.
   7 4  10.
   8 5  10.
   5 9  10.
   9 4  10.
   4 1  20.
OPTIMAL COST =  -20.
NUMBER OF AUCTION/SHORTEST PATH ITERATIONS =
9
NUMBER OF ITERATIONS =  13
NUMBER OF MULTINODE ITERATIONS =  2
NUMBER OF MULTINODE ASCENT STEPS =  1
NUMBER OF REGULAR AUGMENTATIONS =  2
**********************************
```

Fig. 6 - Output do software do Relax4.

- **Linha 1:** este 1º output, simboliza que no arco, de origem **1** e destino **2**, o fluxo máximo enviado foi **10**. Apesar deste envio, é seguro de dizer que este arco não atingiu a sua capacidade máxima de **1000 (INFINITA)**;
- **Linha 2:** este 2º output, simboliza que no arco, de origem **1** e destino **3**, o fluxo máximo enviado foi **10**. Apesar deste envio, é seguro de dizer que este arco não atingiu a sua capacidade máxima de **1000 (INFINITA)**;
- **Linha 3:** este 3º output, simboliza que no arco, de origem **2** e destino **7**, o fluxo máximo enviado foi **10**. Este fluxo levou ao impedimento da passagem de fluxo adicional por este arco, visto que atingiu a capacidade máxima, que por ele deve passar **(10)**;
- **Linha 4:** este 4º output, simboliza que no arco, de origem **3** e destino **8**, o fluxo máximo enviado foi **10**. Este fluxo levou ao impedimento da passagem de fluxo adicional por este arco, visto que atingiu a capacidade máxima, que por ele deve passar **(10)**;
- **Linha 5:** este 5º output, simboliza que no arco, de origem **7** e destino **4**, o fluxo máximo enviado foi **10**. Apesar deste envio, é seguro de dizer que este arco não atingiu a sua capacidade máxima de **1000 (INFINITA)**;
- **Linha 6:** este 6º output, simboliza que no arco, de origem **8** e destino **5**, o fluxo máximo enviado foi **10**. Apesar deste envio, é seguro de dizer que este arco não atingiu a sua capacidade máxima de **1000 (INFINITA)**;
- **Linha 7:** este 7º output, simboliza que no arco, de origem **5** e destino **9**, o fluxo máximo enviado foi **10**. Este fluxo levou ao impedimento da passagem de fluxo adicional por este arco, visto que atingiu a capacidade máxima, que por ele deve passar **(10)**;
- **Linha 8:** este 8º output, simboliza que no arco, de origem **9** e destino **4**, o fluxo máximo enviado foi **10**. Apesar deste envio, é seguro de dizer que este arco não atingiu a sua capacidade máxima de **1000 (INFINITA)**;

- **Linha 9:** este 9º output, simboliza que no arco, de origem **4** e destino **1**, o fluxo máximo enviado foi **20**. Apesar deste envio, é seguro de dizer que este arco não atingiu a sua capacidade máxima de **1000 (INFINITA)**. No contexto do problema de fluxo máximo, as 20 unidades de fluxo que estão a ser enviadas ao longo deste arco, é igual ao fluxo enviado pela origem e recebido pelo destino do grafo.

No problema de fluxo máximo, o custo associado aos arcos pode ser interpretado como uma **poupança ou ganho**, em vez de um custo direto. Um **custo negativo** indica que enviar fluxo por uma determinada aresta resulta numa poupança.

Portanto, o custo ótimo de **-20** indica que a maximização do fluxo resulta numa poupança total de 20 unidades (ou num ganho de 20 unidades). Isto significa que, ao otimizar o fluxo no grafo de acordo com as restrições impostas, o sistema poupa **20 unidades em custos de transporte ou obtenção de recursos.**

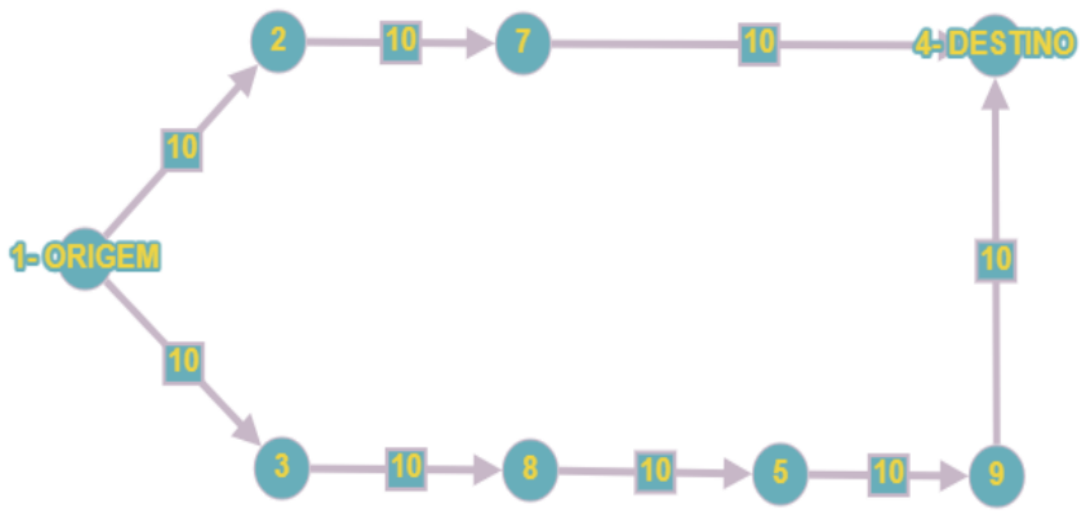


Fig. 7 - Solução Ótima em forma de grafo.

Com a ajuda deste grafo **"ótimo"**, podemos verificar que a regra deste tipo de problemas foi cumprida, ou seja, a **soma dos fluxos que entram pela origem (arcos 12 e 13)** é igual à **soma dos fluxos que saem pelo destino (arcos 24 e 54)**: **10 + 10 = 10 + 10 = 20.**

Caminhos Possíveis da Origem até ao Destino:

- **Caminho 1:** 1 - 2 - 7 - 4;
- **Caminho 2:** 1 - 3 - 8 - 5 - 9 - 4;

4 - Validação do Modelo

Para começar a nossa validação do modelo, tivemos de verificar se as nossas soluções iniciais eram válidas perante as restrições impostas pelo problema.

O próximo passo para a validação do projeto foi a verificação da solução ótima, e para que isso fosse possível, decidimos modelar o problema como um problema de programação linear, usando o software **LPSolve.** A função objetivo neste software consiste na maximização dos arcos, dos fluxos de entrada e de saída e também da maximização das capacidades dos vértices, desde que estas consigam respeitar as regras de conservação do fluxo. As variáveis de decisão usadas são todas inteiras, representam os diversos arcos do nosso grafo e no resultado final vão demonstrar o fluxo, que pode passar por qualquer um desses arcos.

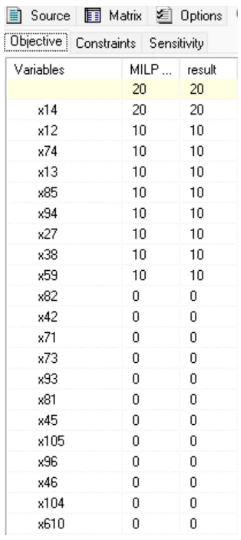
Source | Matrix | Options | Result

```
/* Objective function */
max: 1 x14;

/* Variable bounds */

VERTICE2: - 1 x82 - 1 x42 - 1 x12 + 1 x71 + 1 x73 + 1 x74 = 0;
VERTICE3: - 1 x13 - 1 x73 - 1 x93 + 1 x81 + 1 x82 + 1 x85 = 0;
VERTICE5: - 1 x85 - 1 x45 - 1 x105 + 1 x93 + 1 x94 + 1 x96 = 0;
VERTICE6: - 1 x46 - 1 x96 + 1 x104 + 1 x105 = 0;

VERTICE_2: 1 x82 + 1 x42 + 1 x12 = 1 x27;
VERTICE_3: 1 x13 + 1 x73 + 1 x93 = 1 x38;
VERTICE_5: 1 x85 + 1 x45 + 1 x105 = 1 x59;
VERTICE_6: 1 x46 + 1 x96 = 1 x610;

FLUXO.ORIGEM: - 1 x71 - 1 x81 + 1 x12 + 1 x13 = 1 x14;
FLUXO.DESTINO: - 1 x74 - 1 x94 - 1 x104 + 1 x42 + 1 x45 + 1 x46 = - 1 x14;

ARCO2: x27 <= 10;
ARCO3: x38 <= 70;
ARCO5: x59 <= 10;
ARCO6: x610 <= 10;

int x12, x13, x71, x81, x27, x73, x82, x38, x42, x74, x85, x93, x45, x59, x94, x105, x96, x46, x104, x610, x14;
```

Fig. 8 - Input para o software do LPSolve.

O nosso input para o LPSolve contém 3 tipos de restrição:

- **Restrição 1:** temos as restrições do tipo **VERTICEN**, estas são aplicadas a 4 vértices do nosso grafo **(2, 3, 5, 6)**. Nestas é garantido o **princípio de conservação do fluxo**, ou seja, garantimos que a soma dos fluxos de cada arco, que "entra" num determinado vértice é **igual** aquilo que "sai" deste determinado vértice com uma capacidade definida;
- **Restrição 2:** temos as restrições do tipo **VERTICE_N**, estas são aplicadas a 4 vértices do nosso grafo **(2, 3, 5, 6)**. Nestas é garantida a soma do fluxo de todos os arcos com destino em 2, 3, 5 e 6, não ultrapasse a capacidade da aresta criada para representar cada um destes vértices e as suas capacidades;
- **Restrição 3:** temos as restrições do tipo **FLUXO**, onde nós determinamos o valor da variável da **função objetivo**, que nós pretendemos maximizar. Esta variável é chamada de **fluxo máximo**. Como este tipo de restrição está a ser aplicada aos **vértices origem e destino**, tivemos de garantir que a conservação do fluxo era mantida. Se fosse respeitada, então teríamos calculado o nosso fluxo máximo corretamente;
- **Restrição 4:** temos as restrições do tipo **ARCO**, onde garantimos que as capacidades dos arcos do nosso grafo, não sejam ultrapassadas.


Source Matrix Options
Objective Constraints Sensitivity

Variables	MILP ...	result
	20	20
x14	20	20
x12	10	10
x74	10	10
x13	10	10
x85	10	10
x94	10	10
x27	10	10
x38	10	10
x59	10	10
x82	0	0
x42	0	0
x71	0	0
x73	0	0
x93	0	0
x81	0	0
x45	0	0
x105	0	0
x96	0	0
x46	0	0
x104	0	0
x610	0	0

Fig. 9 - Output para o software do LPSolve.

Com tudo isto, obtivemos o output da **Fig. 9** onde obtivemos os valores iguais ao do software do **Relax4**, tanto para os fluxos que passam pelos arcos, como para o valor do fluxo máximo**, 20**. Concluindo, conseguimos validar o nosso modelo do problema de fluxo máximo com sucesso.

5 - Conclusão

Neste projeto, exploramos a modelação e resolução de um problema de fluxo máximo utilizando o software **RELAX4**. Começamos pela modelação do problema, definindo os vértices, arestas e suas capacidades, bem como os custos associados ao fluxo. Em seguida, utilizamos o RELAX4 para resolver o problema, procurando maximizar o fluxo total do grafo enquanto respeitamos as capacidades das arestas.

Por fim, ao analisar o output do **Relax4** e com a validação efetuada, através do uso do **LPSolve**, conseguimos interpretar os resultados, incluindo o fluxo máximo em determinadas arestas, o custo ótimo associado à solução encontrada e etc...

Este projeto proporcionou uma oportunidade de aplicar conceitos teóricos de teoria dos grafos e programação linear em um contexto prático, permitindo uma compreensão mais profunda dos problemas de otimização e as suas aplicações no mundo real.