

Investigação Operacional

Licenciatura em Gestão

3.º Ano

Ano Lectivo 2013/14

Optimização em Redes

Texto elaborado por:

Maria João Cortinhal (Coordenadora)

Anabela Costa

Maria João Lopes

Ana Catarina Nunes

1. Introdução

A otimização em redes foca-se no estudo e resolução de problemas que possam ser representados através de uma rede. São exemplos disso, problemas de planeamento de produção, problemas de rotas, etc.

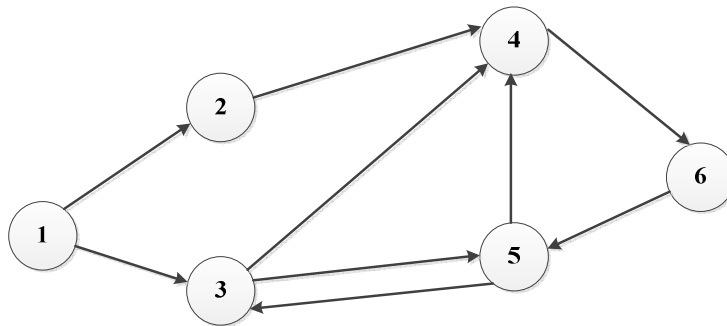
2. Definições

Considere-se V um conjunto finito de vértices (ou nodos) e A um conjunto de arcos, contido em $V \times V$. Designa-se por **Grafo** ao par ordenado $G=(V, A)$.

A explicitação de um grafo pode ser feita de forma extensiva, enumerando o conjunto de vértices e de arcos ou por meio de uma representação gráfica, tal como se ilustra no exemplo que segue.

Exemplo:

Dados $V=\{1,2,3,4,5,6\}$ e $A=\{(1,2), (1,3), (2,4), (3,4), (3,5), (4,6), (5,3), (5,4), (6,5)\}$, $G=(V, A)$ é um grafo, o qual pode ser representado graficamente do seguinte modo:

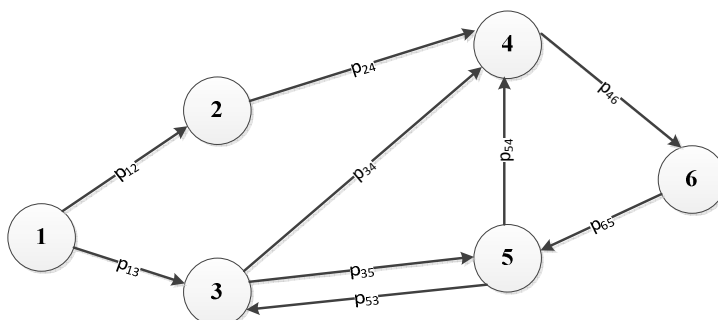


Outras representações deste grafo podem ser obtidas. Para tal, basta posicionar os vértices de forma diferente.

Considere-se ainda, P um conjunto de pesos associados aos arcos. Designa-se por **Rede** ao terno ordenado $R=(V, A, P)$. Tal como grafos, também as redes podem ser definidas explicitamente e por representação gráfica.

Exemplo:

Dados $V=\{1,2,3,4,5,6\}$, $A=\{(1,2), (1,3), (2,4), (3,4), (3,5), (4,6), (5,3), (5,4), (6,5)\}$ e $P=\{p_{12}, p_{13}, p_{24}, p_{34}, p_{35}, p_{46}, p_{53}, p_{54}, p_{65}\}$, $R=(V, A, P)$ é uma rede, a qual pode ser representada graficamente do seguinte modo:



Numa rede pode considerar-se mais do que um peso associado a cada arco. Por exemplo, um dos pesos pode representar um custo, enquanto que outro pode referir-se a uma duração.

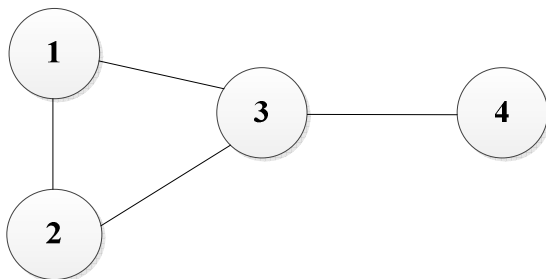
Nas definições anteriores, considerou-se que as ligações entre os vértices têm um determinado sentido, designando-se por isso por arcos. Existem contudo situações em que não interessa considerar o sentido – por exemplo, se as ligações representarem redes de telecomunicações – ou em que importa considerar, em simultâneo, ambos os tipos de ligação, com e sem sentido – por exemplo, num mapa de estradas. Às ligações em que não interessa considerar o sentido dá-se o nome de **arestas**. Surgem assim as definições que se seguem.

Considere-se V um conjunto finito de vértices (ou nodos), A um conjunto de arcos e E um conjunto de arestas.

Designa-se por **Grafo não Orientado** ao par ordenado $G=(V, E)$.

Exemplo:

Dados $V=\{1,2,3,4\}$ e $E=\{(1,2), (1,3), (2,3), (3,4)\}$, $G=(V, E)$ é um grafo não orientado, o qual pode ser representado graficamente do seguinte modo:

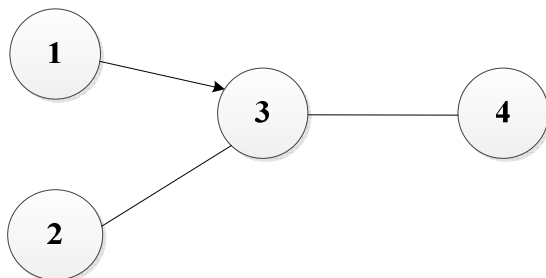


Note-se uma aresta (a,b) pode ser também designada por aresta (b,a) .

Designa-se por **Grafo Misto** ao par ordenado $G=(V, A \cup E)$.

Exemplo:

Dados $V=\{1,2,3,4\}$, $A=\{(1,3)\}$ e $E=\{(2,3), (3,4)\}$, $G=(V, A \cup E)$ é um grafo misto, o qual pode ser representado graficamente do seguinte modo:



Notas:

1. Cada aresta (i,j) pode ser representada por meio de 2 arcos – o arco de i para j e o arco de j para i . Sendo assim, a explicitação do conjunto de arestas pode ser feita através da enumeração de arcos;
2. Seja (i,j) um arco de um grafo $G=(V,A)$. Os vértices i e j são os extremos do arco, sendo i o vértice ou extremo inicial e j o vértice ou extremo final do arco. Seja (a,b) uma aresta de um grafo não orientado. Os vértices a e b são os extremos da aresta.

Nalgumas situações, existe interesse em estudar não o grafo/rede por completo mas apenas uma parte dele, que resulte de se considerar um subconjunto de arcos/arestas e/ou vértices.

Considere-se então grafo $G=(V, A)$ e os subconjuntos $V_1 \subseteq V$ e $A_1 \subseteq A$. Define-se:

Subgrafo gerado por V_1 é o grafo $G_{V_1}=(V_1, A_1)$ cujos vértices são todos os elementos de V_1 e os arcos (ou arestas) são os elementos de A cujos extremos pertencem a V_1 (i.e., $A_1=\{(v_i, v_j) \in A: v_i, v_j \in V_1\}$).

Nota

Dado o grafo $G=(V, A)$, o subgrafo gerado por $V_1 \subseteq V$ obtém-se eliminando os vértices que não pertencem a V_1 e considerando apenas os arcos (ou arestas) que ligam vértices pertencentes a V_1 .

Grafo Parcial gerado por A_1 é o grafo $G_{A_1}=(V, A_1)$ cujos vértices são todos os elementos de V e os arcos (ou arestas) são os elementos de A_1 .

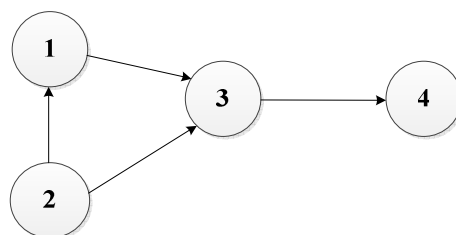
Nota

Dado o grafo $G=(V, A)$, o grafo parcial gerado por $A_1 \subseteq A$ obtém-se considerando todos os vértices de V e apenas os arcos (ou arestas) de A_1 .

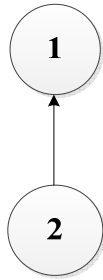
E juntando as duas definições anteriores numa só, obtém-se a definição de **Subgrafo Parcial gerado por V_1 e A_1** que não é mais que um grafo parcial $G_{V_1, A_1}=(V_1, A_1)$ gerado pelos vértices de V_1 e os arcos (ou arestas) de A_1 cujos extremos pertencem a V_1 .

Exemplo:

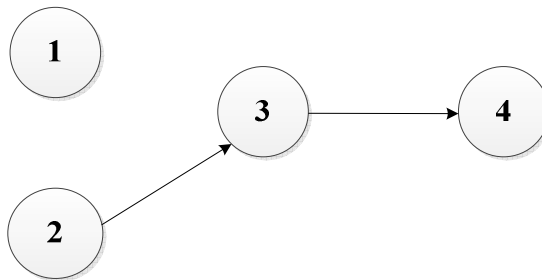
Considere o grafo $G=(V, A)$ com a seguinte representação gráfica:



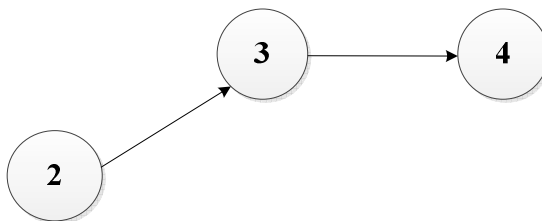
- O subgrafo gerado por $V_1=\{1, 2\}$ tem a seguinte representação gráfica:



- O grafo parcial gerado por $A_1=\{(2, 3), (3, 4)\}$ tem a seguinte representação gráfica:



- O subgrafo parcial gerado por $V_1=\{2, 3, 4\}$ e $A_1=\{(2, 3), (3, 4)\}$ tem a seguinte representação gráfica:



Tal como foi anteriormente referido, a otimização em redes faz o estudo de problemas que podem ser representados por meio de uma rede. No estudo desse tipo de problemas, podem surgir situações em que seja necessário determinar um subconjunto de vértices e/ou arcos. Estas situações conduzem às seguintes definições:

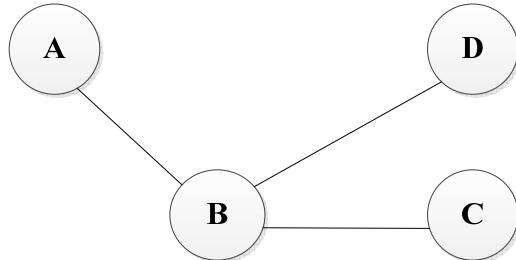
Cadeia entre dois nodos s e t: uma sequência de arcos que ligam esses dois nodos, em que os arcos consecutivos têm que ter um extremo em comum. Os nodos **s** e **t** designam-se por extremos da cadeia. Chama-se **Ciclo** a uma cadeia cujos extremos coincidem

Caminho de s para t: uma sequência de arcos entre **s** e **t** em que, para cada par de arcos consecutivos, o vértice final do primeiro arco tem que ser igual ao vértice inicial do seu sucessor na sequência. Chama-se **Circuito** a um caminho cujos extremos coincidem.

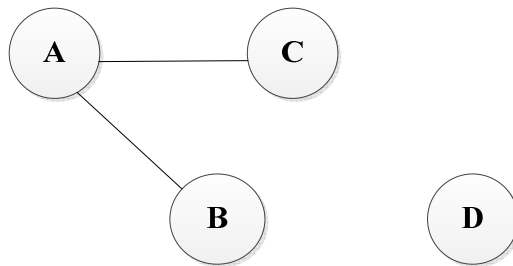
Grafo Conexo: se existir uma cadeia entre qualquer par de vértices. Se o grafo não for conexo é possível identificar subgrafos conexos, designados por componentes conexas, tal como se ilustra no exemplo que se segue.

Exemplo:

- Grafo conexo:



- Grafo não conexo, no qual existem duas componentes conexas:



Note-se que não existe nenhuma cadeia entre D e os restantes nodos. No entanto, os subgrafos gerados, respectivamente, por {A, B, C} e por {D} são conexos.

3.Problema da **Árvore de Suporte de Custo Mínimo**

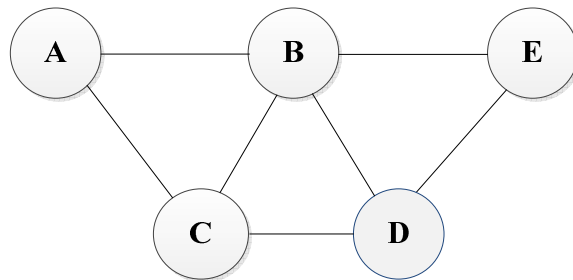
Considere o grafo $G=(V, A)$, em que V representa o conjunto de vértices e A um conjunto de arestas, e $A_1 \subseteq A$. Chama-se **Árvore de Suporte do grafo G** ao grafo parcial $G_{A_1}=(V, A_1)$ em que as arestas de A_1 permitem estabelecer uma e uma só cadeia entre qualquer par de nodos de V .

Dada uma rede que representa uma situação real, de forma intuitiva podemos dizer que a determinação de uma **Árvore de Suporte** corresponde a seleccionar o número mínimo de ligações (arestas) que devem ser estabelecidas de forma a garantir que, de qualquer vértice da rede, é possível aceder a qualquer outro vértice da mesma rede.

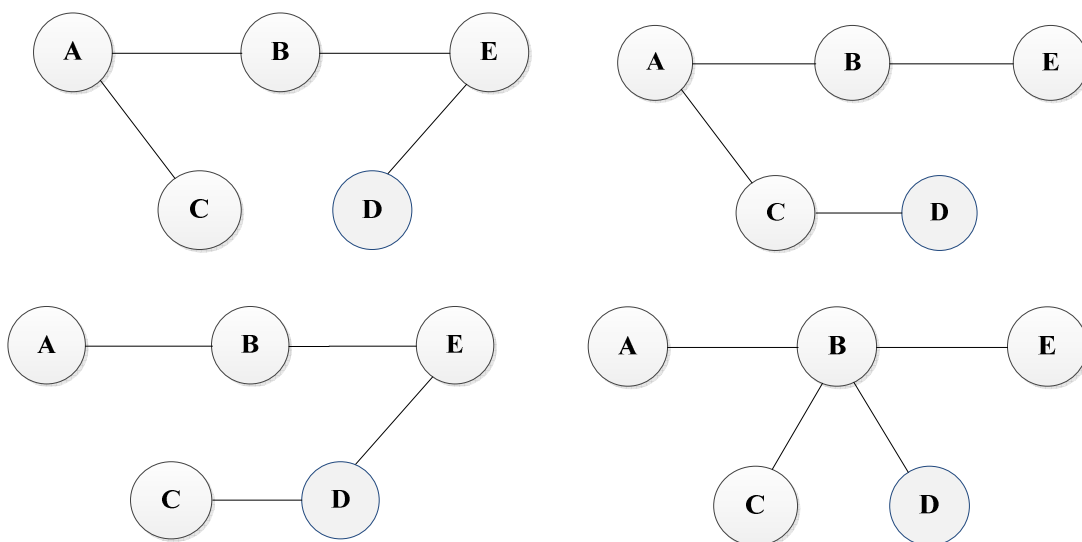
Note-se que cada grafo pode conter um número exponencial de árvores de suporte, tal como é ilustrado no exemplo que se segue.

Exemplo:

Considere o grafo $G=(V, A)$ com a seguinte representação gráfica:



Este grafo admite várias árvores de suporte, das quais se apresentam as seguintes:



Se a cada aresta estiver associado um peso, é então possível determinar árvores de suporte com diferentes pesos totais. Chama-se **Árvore de Suporte de Custo Mínimo** à árvore de suporte para a qual a soma dos pesos das arestas é mínima.

Resultado

Dado um grafo conexo $G=(V, A)$ em que V é constituído por N nodos, são válidas as seguintes afirmações:

- i) Uma árvore de suporte de G é um grafo conexo sem ciclos.
- ii) Uma árvore de suporte de G é um grafo conexo com $N-1$ arestas.

3.1 Determinação da Árvore de Suporte de Custo Mínimo

Para determinar a árvore de suporte de custo mínimo associada à rede conexa $R=(V, A, C)$ pode aplicar-se o Algoritmo de Kruskal ou o Algoritmo de Prim.

Algoritmo de Kruskal

Em termos gerais, o Algoritmo de Kruskal começa por considerar $|V|$ componentes conexas e vai inserindo arestas, até obter uma única componente conexa.

Definindo $G_S=(V_S, T_S)$ como sendo o grafo que representa a árvore de suporte de custo mínimo da rede $R=(V, A, C)$, o Algoritmo de Kruskal é constituído pelos seguintes passos:

PASSO 1

- $V_S = V$;
- $T_S = \emptyset$ (Não existem arestas no grafo G_S);
- $C_S = 0$ (O custo da árvore G_S é nulo).

PASSO 2

- Ordenar arestas por ordem não decrescente dos respectivos custos (ou pesos).

PASSO 3

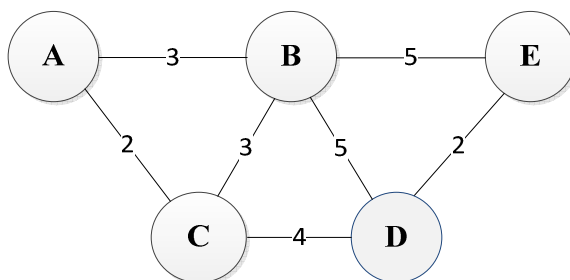
- Seleccionar a aresta do topo da lista de arestas e adicionar ao conjunto T_S desde que a inclusão desta aresta não origine um ciclo em G_S . Retirar a aresta da lista ordenada. Adicionar a C_S o custo da aresta.

PASSO 4

- Se G_S é uma árvore de suporte então STOP;
- Senão voltar ao PASSO 3.

Exemplo de Aplicação:

Considere a rede $R=(V, A, C)$ com a seguinte representação gráfica:

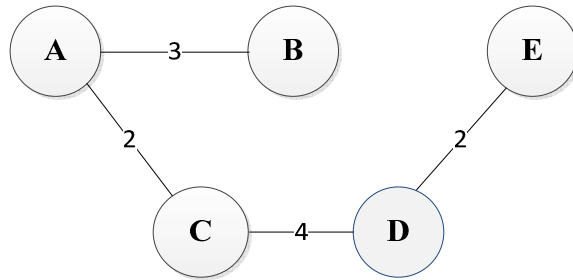


- Resumo de aplicação do Algoritmo de Kruskal:

Arestas	Custos	Estado da aresta	Conjunto T_S	Custo de G_S
(A, C)	2	Adicionada	{(A,C)}	$C_S = 0 + 2 = 2$
(D, E)	2	Adicionada	{(A,C), (D,E)}	$C_S = 2 + 2 = 4$
(A, B)	3	Adicionada	{(A, C), (D, E), (A,B)}	$C_S = 4 + 3 = 7$
(B, C)	3	Excluída (a sua	{(A,C), (D,E), (A,B)}	$C_S = 7$

		inclusão criava um ciclo)		
(C, D)	4	Adicionada e STOP (T_S já tem $5-1=4$ arestas)	$\{(A,C), (D,E), (A,B), (C,D)\}$	$C_S=7+4=11$
(B, D)	5	–	–	–
(B, E)	5	–	–	–

- **Árvore de suporte de custo mínimo obtida pelo Algoritmo de Kruskal:**



(custo da árvore de suporte é $C_S=11$)

Algoritmo de Prim

Em termos gerais, o Algoritmo de Prim começa por considerar uma única componente conexa, contendo apenas um vértice e vai alargando a componente conexa, por inserção de vértices e arestas, até que tenha $|V|$ vértices.

Definindo $G_S=(V_S, T_S)$ como sendo o grafo que representa a árvore de suporte de custo mínimo da rede $R=(V, A, C)$, o Algoritmo de Prim é constituído pelos seguintes passos:

PASSO 1: Inicialização

- $V_S=\{v_s\}$ onde v_s é um nodo (ou vértice) qualquer de R ;
- $T_S=\emptyset$ (Não existem arestas no grafo G_S);
- $C_S=0$ (O custo da árvore G_S é nulo).

PASSO 2: Determinação do vértice v_{j^*} mais “perto” de G_S

- Seleccionar a aresta (v, v_{j^*}) de menor custo (ou peso) c , entre as arestas (v, v_j) , com $v \in V_S$ e $v_j \notin V_S$.

PASSO 3: Aumento de G_S

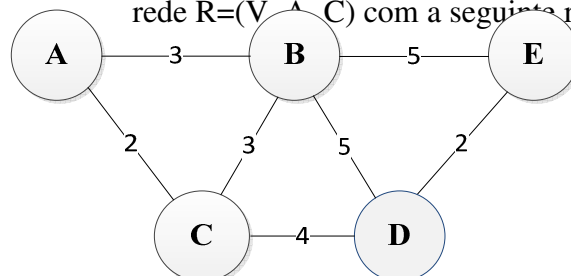
- $V_S = V_S \cup \{v_{j^*}\}$ e $T_S = T_S \cup \{(v, v_{j^*})\}$;
- $C_S = C_S + c$.

PASSO 4: Teste

- Se G_S é uma árvore de suporte então STOP;
- Senão voltar ao PASSO 2.

Exemplo de Aplicação:

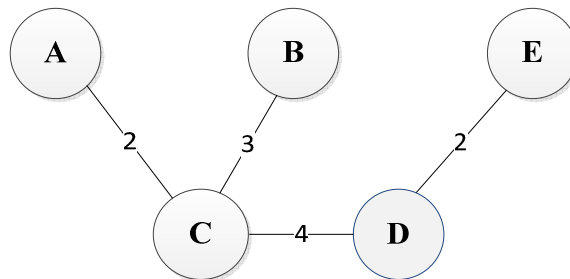
Considere a rede $R=(V, A, C)$ com a seguinte representação gráfica:



- Resumo de aplicação do Algoritmo de Prim considerando $V_S=\{B\}$ (o primeiro nodo a incluir em V_S pode ser um qualquer da rede R):

Vértice mais “perto” de G_S	Conjunto V_S	Conjunto T_S	Custo de G_S
C	{B, C}	{(B,C)}	$C_S=0+3=3$
A	{B, C, A}	{(B,C),(C,A)}	$C_S=3+2=5$
D	{B, C, A, D}	{(B,C),(C,A),(C,D)}	$C_S=5+4=9$
E	{B, C, A, D, E}	{(B,C),(C,A),(C,D),(D,E)} e STOP (T_S já tem $5-1=4$ arestas)	$C_S=9+2=11$

- Árvore de suporte de custo mínimo obtida pelo Algoritmo de Prim:



(custo da árvore de suporte é $C_S=11$)

Nota: Como se pode observar, a aplicação dos dois algoritmos pode gerar soluções diferentes, no entanto, o custo terá que ser igual.

4.Problema do Caminho mais Curto

Considere a rede $R=(V, A, P)$ em que P representa o conjunto de “custos” associados aos arcos (ou arestas) de A. Chama-se **Caminho mais Curto entre os vértices s e t** ao caminho entre s e t para o qual a soma dos custos dos arcos (ou arestas) é mínima.

Em termos práticos, determinar o Caminho mais Curto entre os vértices s e t corresponde a seleccionar as ligações (arcos ou arestas) que devem ser estabelecidas de forma a aceder, da forma mais eficiente, a um ponto de destino t a partir de um ponto de origem s.

Para além do problema base, podem ser consideradas suas variantes:

- Dado um vértice s , o problema de determinar o caminho mais curto entre s e todos os outros vértices da rede $R=(V, A, C)$;
- O problema de determinar o caminho mais curto entre todos os pares de vértices da rede $R=(V, A, C)$.

4.1 Determinação do Caminho mais Curto

Para determinar o caminho mais curto entre s e t pode recorrer-se a algoritmos iterativos que atribuem etiquetas aos vértices.

Estes algoritmos podem ser divididos em dois grupos:

- Algoritmos de fixação de etiquetas – aplicáveis sempre que não existem circuitos de custo negativo – **Algoritmo de Dijkstra** e suas variantes;
- Algoritmos de correcção de etiquetas – aplicáveis quer existam ou não circuitos de custo negativo – **Algoritmo de Floyd**.

Nota

Nesta Unidade Curricular apenas será estudado o Algoritmo de Dijkstra, o qual determina o caminho mais curto entre um vértice $s \in V$ e todos os outros vértices da rede $R=(V, A, C)$. Salienta-se, ainda, que o Algoritmo de Dijkstra só pode ser aplicado em redes que não contenham arcos de custo negativo.

Algoritmo de Dijkstra (Fixação de Etiquetas)

Admitindo que se pretende determinar o caminho mais curto entre um vértice $s \in V$ e todos os outros vértices da rede $R=(V, A, C)$, o Algoritmo de Dijkstra é constituído pelos seguintes passos:

PASSO 1: Inicialização

- $S=\{s\}$, onde S é o conjunto dos vértices que têm etiqueta permanente;
- $\bar{S}=V-S$;
- $\Pi(s)=0$, onde $\Pi(i)$ é o comprimento do caminho mais curto entre s e i , obtido até ao momento;
- $\rho(s)=s$, onde $\rho(i)$ é o vértice antecessor de i no caminho mais curto entre s e i .
- $\Pi(i) = c(s,i)$ se $(s,i) \in A$, $\Pi(i)=+\infty$ se $(s,i) \notin A$, $\rho(i)=s$.

PASSO 2: Determinação do vértice $j^* \in \bar{S}$

- $j^*: \Pi(j^*) = \min_{j \in \bar{S}} \Pi(j)$;
- $S = S \cup \{j^*\}$;
- $\bar{S} = \bar{S} - \{j^*\}$
- Se $\bar{S}=\emptyset$ então STOP;

NOTA: No caso de se pretender o caminho mais curto entre o vértice s e um vértice t , substitui-se o critério de paragem anterior por: Se $t \in S$ então STOP;

- Senão ir para o PASSO 3.

PASSO 3: Actualização das etiquetas

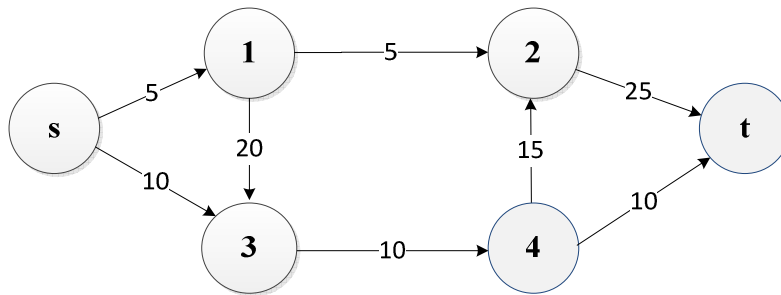
- $\Pi(j) = \min_{j \in (T_{j^*} \cap \bar{S})} \{\Pi(j), \Pi(j^*) + c_{j^*j}\}$, onde T_{j^*} é o conjunto de sucessores directos de j^* ;
- $\rho(j) = j^*$, se $\Pi(j)$ foi melhorado;
- Voltar ao PASSO 2.

Observação

Podem existir mais que um caminho mais curto entre s e t , mas o algoritmo de Dijkstra só permite determinar um deles.

Exemplo de Aplicação:

Considere a rede $R=(V, A, C)$ com a seguinte representação gráfica:



- Resumo de aplicação do Algoritmo de Dijkstra – determinação do caminho mais curto entre s e t :

j	s✓	1	2	3	4	t	Informação
$\Pi(j)$	0	5✓	∞	10	∞	∞	$S=\{s\}$; $\min_{j \in S} \Pi(j)=5; j^*=1; S=\{s,1\}$
$\rho(j)$	s	s	s	s	s	s	
$\Pi(j)$	–	–	10✓	10	∞	∞	Ver Cálculos Auxiliares (i) $\min_{j \in S} \Pi(j)=10; j^*=2; S=\{s,1,2\}$
$\rho(j)$	–	–	1	s	s	s	
$\Pi(j)$	–	–	–	10✓	∞	35	Ver Cálculos Auxiliares (ii) $\min_{j \in S} \Pi(j)=10; j^*=3; S=\{s,1,2,3\}$
$\rho(j)$	–	–	–	s	s	2	
$\Pi(j)$	–	–	–	–	20✓	35	Ver Cálculos Auxiliares (iii) $\min_{j \in S} \Pi(j)=20; j^*=4; S=\{s,1,2,3,4\}$
$\rho(j)$	–	–	–	–	3	2	
$\Pi(j)$	–	–	–	–	–	30✓	Ver Cálculos Auxiliares (iv) $\min_{j \in S} \Pi(j)=30; j^*=t; S=\{s,1,2,3,4,t\}$
$\rho(j)$	–	–	–	–	–	4	STOP porque $t \in S$.

Cálculos Auxiliares

(i) $\Pi(2) = \min\{\Pi(2); \Pi(1) + c_{12}\} = \min\{\infty; 5+5\} = 10$ (logo, $\rho(2) = 1$);
 $\Pi(3) = \min\{\Pi(3); \Pi(1) + c_{13}\} = \min\{10; 5+20\} = 10$ (logo, $\rho(3)$ não sofre alteração);

(ii) $\Pi(t) = \min\{\Pi(t); \Pi(2) + c_{2t}\} = \min\{\infty; 10+25\} = 35$ (logo, $\rho(t) = 2$);

(iii) $\Pi(4) = \min\{\Pi(4); \Pi(3) + c_{34}\} = \min\{\infty; 10+10\} = 20$ (logo, $\rho(4) = 3$);

(iv) $\Pi(t) = \min\{\Pi(t); \Pi(4) + c_{4t}\} = \min\{35; 20+10\} = 30$ (logo, $\rho(t) = 4$).

Caminho mais Curto entre s e t

- Custo: $\Pi(t) = 30$ e $\rho(t) = 4$, $\rho(4) = 3$, $\rho(3) = 1$



Note-se que a aplicação do algoritmo de Dijkstra também permite determinar os caminhos mais curtos entre s e todos os nodos que estiverem etiquetados, com etiqueta permanente, quando o algoritmo terminar. No exemplo anterior, após a aplicação do algoritmo é mesmo possível determinar o caminho mais curto entre s e todos os restantes nodos, dado que todos eles se encontram etiquetados, com etiqueta permanente,

Notas

- Para determinar o caminho mais curto entre um vértice s e todos os outros vértices aplica-se o algoritmo de Dijkstra em que a condição de paragem é todos os vértices estarem etiquetados com etiqueta permanente (também se pode aplicar o algoritmo de Floyd).
- Para obter o caminho mais curto entre todos os pares de vértices determina-se, para cada vértice da rede, o caminho mais curto a todos os outros vértices utilizando o algoritmo de Dijkstra (também se pode aplicar o algoritmo de Floyd).

Questões:

1. E se, na determinação do caminho mais curto entre dois vértices, for necessário passar por um ponto intermédio da rede? E por um arco da rede?
2. E se se quiser determinar o caminho mais longo?

5.Problemas de Fluxos numa rede

Nalguns problemas de redes, a cada arco/aresta encontra-se associado uma capacidade que representa a quantidade máxima de um certo bem – fluxo – que pode passar nessa ligação, numa determinada unidade de tempo. Por exemplo, o número máximo de veículos que pode circular numa determinada estrada, por unidade de tempo, de forma que não exista congestionamento de tráfego. Esses problemas são designados por problemas de fluxos. Nesta UC iremos estudar dois deles: o problema do fluxo máximo

e o problema do fluxo de custo mínimo. Em ambos considera-se um nodo inicial que gera fluxo e um nodo final ou destino que absorverá o fluxo gerado pela origem. Antes de prosseguirmos, destaca-se aqui alguma da notação que irá ser utilizada.

Dada a rede $R=(V, A, U)$, denota-se por:

- u_{ij} a **capacidade** do arco $(i, j) \in A$ (assume-se que $u_{ij} \geq 0$);
- x_{ij} a variável que representa a **quantidade de fluxo** que está a passar no arco $(i, j) \in A$ (assume-se que $0 \leq x_{ij} \leq u_{ij}$);
- Arco **saturado**: qualquer arco (i, j) em que $x_{ij} = u_{ij}$;
- **cadeia de aumento de fluxo**: qualquer cadeia em que seja possível aumentar o fluxo entre o nodo origem e o nodo destino.

É também importante realçar que o estudo destes problemas requer a assumpção das seguintes hipóteses:

Hipótese 1 – Princípio de conservação de fluxo:

Exceptuando os vértices inicial e final, o fluxo que entra num vértice é sempre igual ao fluxo que dele sai.

Hipótese 2:

A rede $R=(V, A, U)$ tem apenas um vértice inicial e outro final.

Hipótese 3:

A capacidade de cada arco $(i, j) \in A$, u_{ij} , é um valor inteiro.

No caso da segunda ou da terceira hipótese não se verificar, é possível efectuar alterações de forma a obter uma nova rede que satisfaça estas hipóteses.

5.1 Fluxo máximo numa rede

Considere a rede $R=(V, A, U)$ em que U representa o conjunto de capacidades associadas aos arcos de A . Chama-se **Fluxo Máximo entre os vértices s e t** à quantidade máxima, de um certo bem, que pode ser enviada entre os vértices s e t .

5.1.1. Determinação do Fluxo Máximo

Para determinar o fluxo máximo entre os vértices s e t recorre-se ao Algoritmo de Ford-Fulkerson. Em termos gerais, o Algoritmo de Ford-Fulkerson consiste em identificar sucessivas cadeias de aumento de fluxo entre os vértices s e t . Para construir as cadeias de aumento de fluxo, o algoritmo utiliza um processo de etiquetagem de vértices:

- **Vértice inicial**
 $s^{[-, \infty]}$, o que significa que ao vértice s podem chegar ∞ unidades de fluxo vindas de nenhum vértice.
- **Restantes vértices**
Caso 1: $A \rightarrow B^{[+A, 2]}$, o que significa que ao vértice B podem chegar **2** unidades de fluxo vindas do vértice A ;

Caso 2: $A \xrightarrow{[-B,3]} B$, o que significa que pelo vértice **A** podem deixar de se enviar **3** unidades de fluxo para o vértice **B**.

Algoritmo de Ford-Fulkerson

Admitindo que se pretende determinar o fluxo máximo entre os vértices s e t da rede $R=(V, A, U)$, o Algoritmo de Ford-Fulkerson é constituído pelos seguintes passos:

PASSO 0

Atribuir a cada arco um fluxo admissível (em geral atribui-se o fluxo nulo).

PASSO 1

Etiquetar o nodo origem s com a etiqueta $[-, \infty]$.

PASSO 2

Encontrar um arco (i,j) para o qual:

Caso 1: o nodo i esteja etiquetado e o nodo j não esteja etiquetado e $x_{ij} < u_{ij}$, ou seja, que o fluxo que está a passar no arco (i,j) não atinge ainda a capacidade do arco; ou

Caso 2: o nodo j esteja etiquetado e o nodo i não esteja etiquetado e $x_{ij} > 0$;

Se tal arco não existir então ir para o PASSO 5.

PASSO 3

- Se Caso 1 então etiquetar o nodo j com $[a_j, b_j]$, em que $a_j = +i$ e $b_j = \min(b_i, u_{ij} - x_{ij})$;
- Se Caso 2 então etiquetar o nodo i com $[a_i, b_i]$, em que $a_i = -j$ e $b_i = \min(b_j, x_{ij})$;
- Se o nodo final t já está etiquetado então ir para o PASSO 4;
- Senão repetir o PASSO 2.

PASSO 4

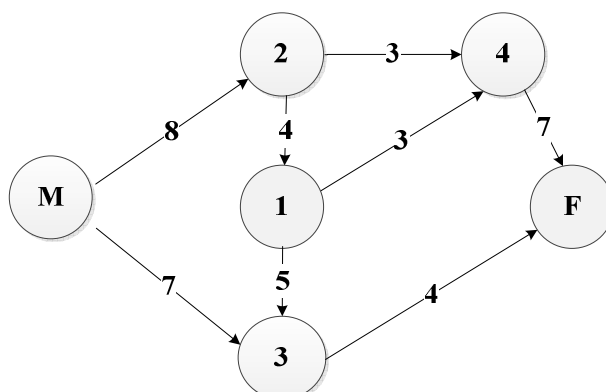
- Foi determinada uma cadeia de aumento de fluxo entre s e t . Incrementar o fluxo na cadeia de aumento de fluxo na quantidade b_t . Começando pelo nodo t : a. Fazer $i=t$; b. Se o nodo i foi etiquetado com $[+k, b_i]$ então incrementar o fluxo no arco (k,i) na quantidade b_t . Se, pelo contrário, foi etiquetado com $[-k, b_i]$, decrementar o fluxo no arco (i,k) na quantidade b_t ; c. Fazer $i=k$; 4. Repetir os passos b,c até atingir o nodo s .
- Ir para o PASSO 1.

PASSO 5

- O fluxo máximo foi determinado. STOP.

Exemplo de Aplicação:

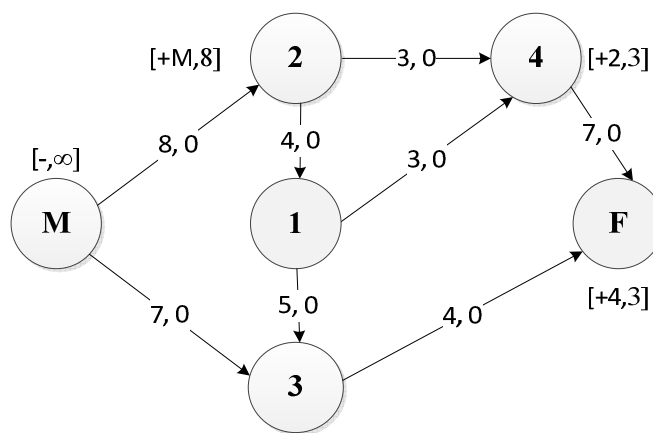
Considere a rede $R=(V, A, U)$ com a seguinte representação gráfica:



em que o valor indicado junto aos arcos representa, a capacidade do arco.

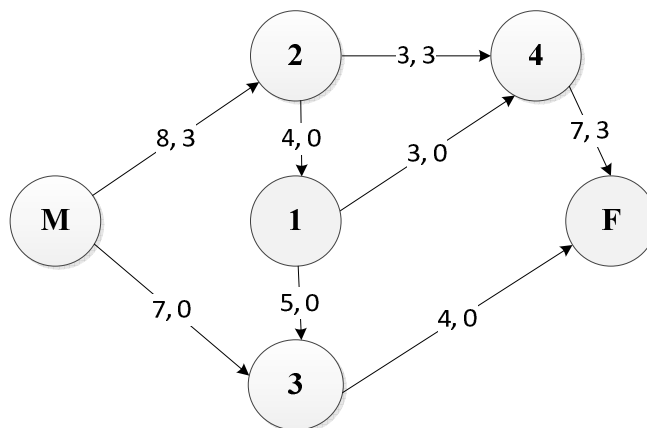
Resumo de aplicação do Algoritmo de Ford-Fulkerson: determinação do fluxo máximo entre os nodos M e F. Começemos por atribuir o fluxo nulo a todos os arcos. Como se verifica o princípio de conservação de fluxo e nenhuma das capacidades é excedida, o fluxo é admissível. Nas redes seguintes os valores a, b junto dos arcos representam, respectivamente, a capacidade do arco e o valor do fluxo nesse arco.

- $F(\text{fluxo que passa na rede})=0$;
- Cadeia de Aumento de Fluxo (C.A.F.) entre M e F:

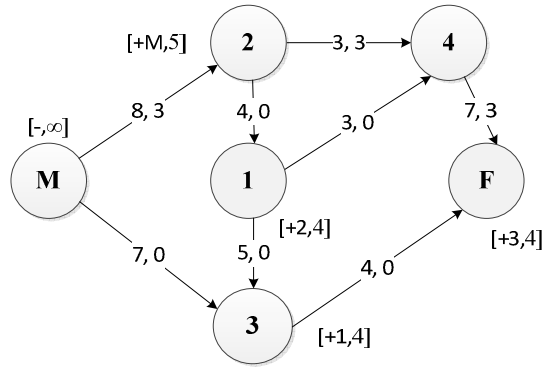


C.A.F. entre M e F: (M,2), (2,4) e (4,F) (fluxo que passa na cadeia é 3) \Rightarrow $F=0+3=3$ (fluxo que passa na rede entre os vértices M e F);

- Actualização da rede:

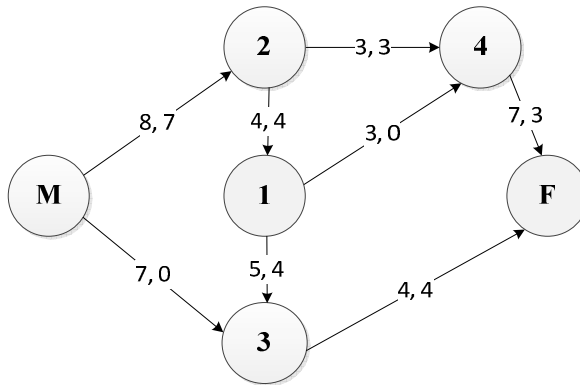


- Cadeia de Aumento de Fluxo (C.A.F.) entre M e F:

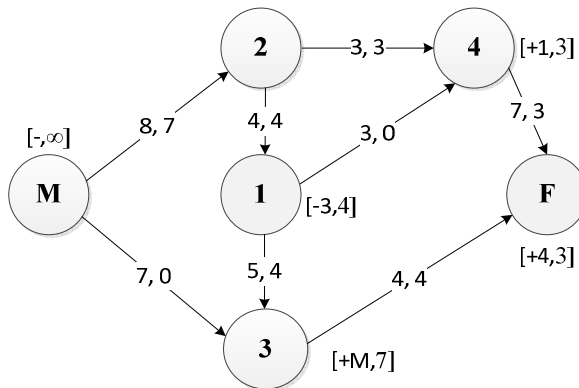


C.A.F. entre M e F: (M,2), (2,1), (1,3) e (3,F) (fluxo que passa na cadeia é 4) $\Rightarrow F=3+4=7$ (fluxo que passa na rede entre os vértices M e F);

➤ Actualização da rede:

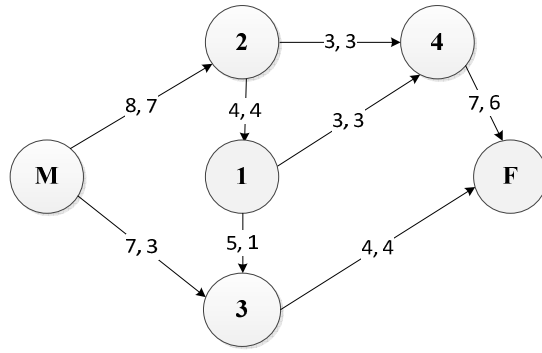


➤ Cadeia de Aumento de Fluxo (C.A.F.) entre M e F:

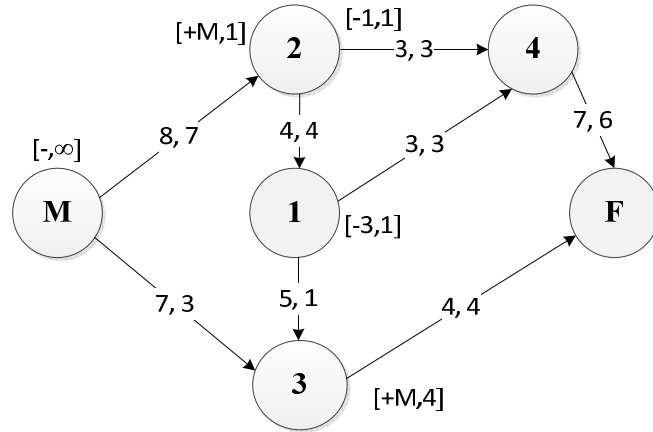


C.A.F. entre M e F: (M,3), (1,3), (1,4) e (4,F) (fluxo que passa na cadeia é 3) $\Rightarrow F=7+3=10$ (fluxo que passa na rede entre os vértices M e F);

➤ Actualização da rede:



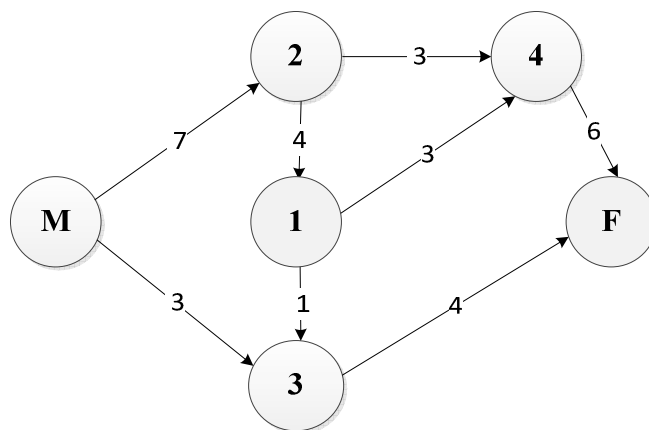
➤ Cadeia de Aumento de Fluxo (C.A.F.) entre M e F:



Na rede actualizada já não é possível construir uma C.A.F entre M e F ⇒ STOP.

➤ Valor do Fluxo Máximo entre M e F: F=10.

➤ Sistema Óptimo de Fluxos (valor apresentado junto a cada arco representa o fluxo que está a passar por esse arco):



5.1.2. Corte de Capacidade Mínima

Relacionado com o problema da determinação do fluxo máximo está um outro problema, e não menos importante, que se designa por problema do corte de capacidade mínima

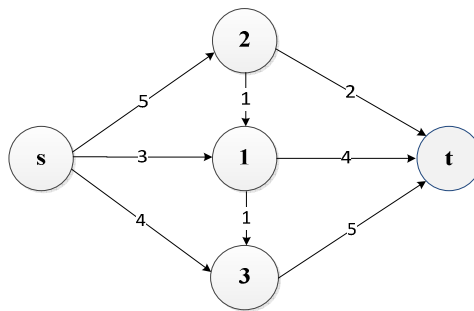
Seja $G=(V, A)$ um grafo e s e t os vértices origem e destino, respectivamente. Considere-se S um subconjunto de vértices que contém o vértice s e não contém o vértice t e $T=V-S$ (ou seja, T é o conjunto complementar de S). Chama-se **Corte (C) em G** ao conjunto de arcos que têm o extremo inicial em S e o extremo final em T . A remoção destes arcos torna impossível qualquer caminho de s para t .

Dado que a cada arco está associada uma capacidade, chama-se **capacidade do corte C** à soma das capacidades dos arcos que o compõem.

Note-se que numa rede com N vértices, podem ser definidos 2^{N-2} cortes.

Exemplo:

Considere a rede $R=(V, A, U)$ com a seguinte representação gráfica, em que o valor indicado junto aos arcos representa a capacidade do arco:



Neste exemplo, são cortes: $C1=\{(s,1), (s,2), (s,3)\}$; $C2=\{(s,2), (s,3), (1,3), (1,t)\}$; $C3=\{(2,t), (1,t), (3,t)\}$; (...),

Sendo as suas capacidades: capacidade do corte $C1=3+5+4=12$; capacidade do corte $C2=5+4+1+4=14$; capacidade do $C3=2+4+5=11$. O número total de Cortes é, neste exemplo, $2^{N-2}=2^{5-2}=8$

A relação entre o problema do fluxo máximo e o corte de capacidade mínima é expressa na proposição e no teorema que seguidamente se enunciam.

Proposição

Num problema de fluxo máximo na rede $R=(V, A, U)$, qualquer fluxo entre o vértice origem s e o vértice destino t não pode exceder a capacidade de qualquer corte em $G=(V, A)$.

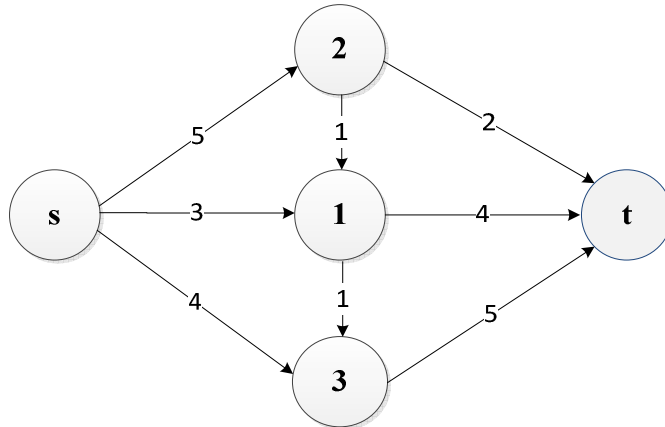
Teorema do Fluxo Máximo – Corte de Capacidade Mínima

O valor do fluxo máximo na rede $R=(V, A, U)$ é igual à capacidade do corte de capacidade mínima no grafo $G=(V, A)$.

Note-se que pode existir mais do que um corte de capacidade mínima.

Exemplo:

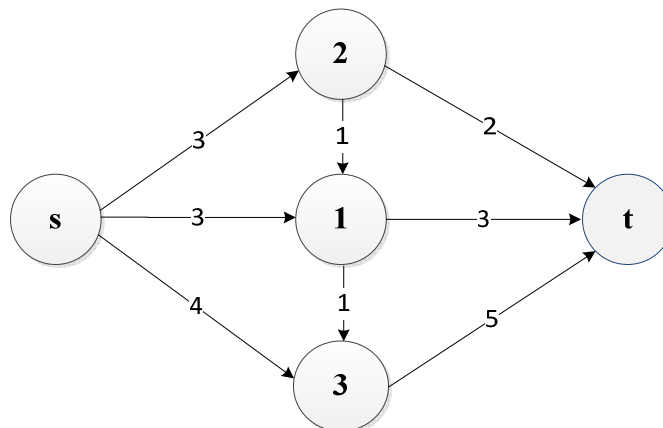
Considere novamente a rede $R=(V, A, U)$, em que o valor indicado junto aos arcos representa a capacidade do arco:



- O corte de capacidade mínima no grafo $G=(V, A)$ tem valor 10 e é constituído pelos arcos: $(s,1)$, $(s,3)$, $(2,1)$ e $(2,t)$.
- Atendendo ao **Teorema do Fluxo Máximo-Corte de Capacidade Mínima**, o valor do fluxo máximo entre s e t na rede $R=(V, A, U)$ é igual a 10.

Nota

De facto, aplicando o Algoritmo de Ford-Fulkerson, o valor do fluxo máximo entre s e t na rede $R=(V, A, U)$ é igual a 10, sendo o sistema óptimo de fluxos dado por:



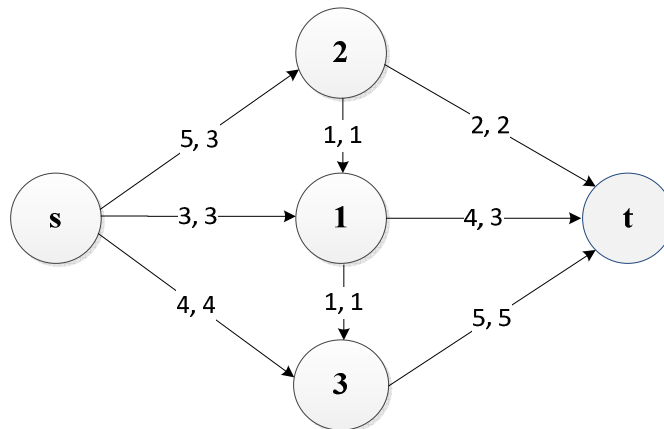
(o valor indicado junto aos arcos representa o fluxo que passa no arco)

5.1.3. Importância do Corte de Capacidade Mínima

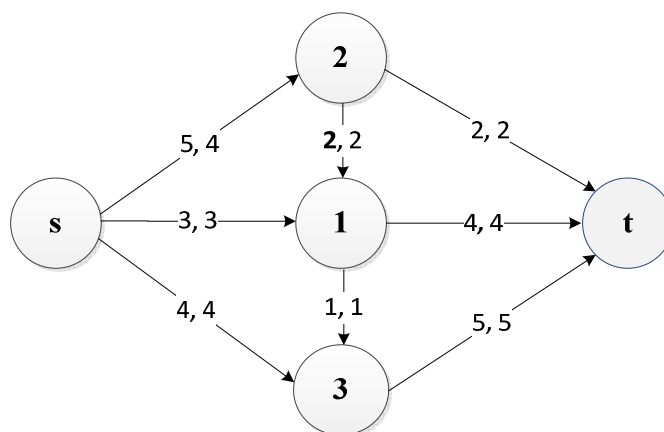
Considere uma rede $R=(V, A, U)$, para a qual se pretende aumentar o valor actual do fluxo máximo entre o vértice origem s e o vértice destino t . Para este efeito, é necessário aumentar a capacidade de arcos da rede R , para aumentar a capacidade do corte ou dos cortes de capacidade mínima. Eventualmente, apenas será necessário aumentar a capacidade de alguns arcos do corte de capacidade mínima.

Exemplo:

Considere novamente a rede $R=(V, A, C)$, em que os valores a, b indicados junto aos arcos representam, respectivamente, a capacidade do arco e o valor do fluxo nesse arco que permite atingir o fluxo máximo entre s e t de valor 10:



Admita que se pretende aumentar o valor do fluxo máximo entre s e t em 1 unidade. Para este efeito, pode-se, por exemplo, aumentar a capacidade do arco $(2,1)$ em 1 unidade. De facto, se a capacidade do arco $(2,1)$ passar a ser 2, é possível construir uma C.A.F. entre s e t que permite transportar mais uma unidade de fluxo: $(s,2)$, $(2,1)$ e $(1,t)$. Deste modo, obtém-se o sistema de fluxos que permite atingir o fluxo máximo entre s e t de valor 11 (o fluxo que passa em cada arco corresponde ao segundo valor junto ao arco):

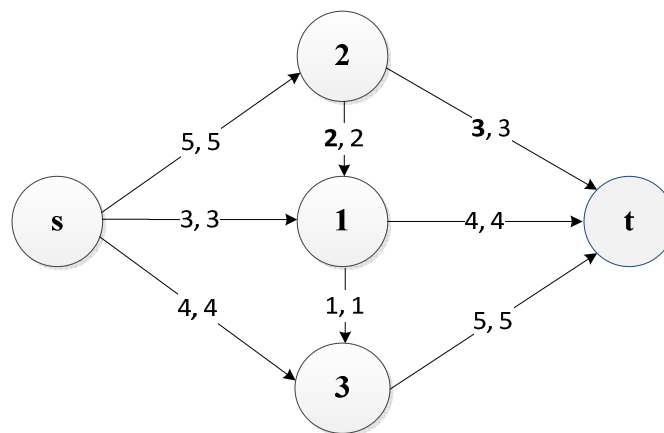


Aumentou-se a capacidade do arco $(2,1)$, que é um dos arcos do corte de capacidade mínima. Este é constituído pelos arcos $(s,1)$, $(s,3)$, $(2,1)$ e $(2,t)$.

Caso se pretenda aumentar o valor do fluxo máximo entre s e t em mais de uma unidade, não é suficiente aumentar a capacidade do arco $(2,1)$. Por exemplo, se o objectivo é que o valor do fluxo máximo passe a ser 12 então, pode-se, por exemplo, aumentar em 1 unidade a capacidade dos arcos $(2,1)$ e $(2,t)$. De facto, se as capacidades dos arcos $(2,1)$ e $(2,t)$ passarem a ser 2 e 3, respectivamente, é possível construir duas C.A.F. entre s e t , em que cada uma delas permite transportar mais uma unidade de fluxo:

- C.A.F.1: $(s,2)$, $(2,1)$ e $(1,t)$;
- C.A.F.2: $(s,2)$, $(2,t)$.

Deste modo, obtém-se o sistema de fluxos que permite atingir o fluxo máximo entre s e t de valor 12 (o fluxo que passa em cada arco corresponde ao segundo valor junto ao arco):

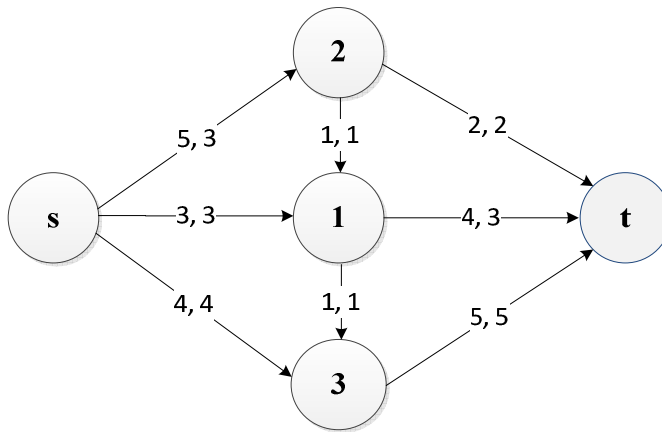


Será possível aumentar o valor do fluxo máximo entre s e t se a capacidade do arco $(2,1)$ passar de 2 para 3? Não. Com o aumento da capacidade do arco $(2,1)$ de 1 para 2 o corte $\{(s,1), (s,3), (2,1), (2,t)\}$ passou a ter capacidade igual a 11, tal como o corte $\{(2,t), (1,t), (3,t)\}$. Passaram a existir dois cortes de capacidade mínima, em vez de um e, conseqüentemente, para se aumentar o fluxo será necessário aumentar a capacidade destes dois cortes.

Questão:

Como identificar um corte de capacidade mínima?

Esta identificação pode ser feita com recurso à aplicação do Algoritmo de Ford-Fulkerson. Este algoritmo termina quando não é possível etiquetar o nodo final. Todos os nodos que podem ser etiquetados são incluídos no conjunto S e os que não podem ser etiquetados são incluídos no conjunto T . Estes dois conjuntos permitem definir um corte de capacidade mínima. Para ilustrar esta situação, consideremos a rede dada no exemplo anterior:



Facilmente se conclui que, por aplicação do Algoritmo de Ford-Fulkerson, apenas os nodos s e 2 podem ser etiquetados, ou seja, não é possível determinar uma cadeia de aumento de fluxo entre s e t . Sendo assim, para definir um dos possíveis cortes de capacidade mínima, basta fazer $S=\{s,2\}$ e $T=\{1,3,t\}$. Considerado agora os arcos pertencentes à rede e cujo nodo inicial pertença a S e o nodo final a T , verifica-se que existem apenas três arcos nesta situação, a saber $(s,1)$, $(s,3)$, e $(2,1)$. O conjunto destes três arcos, é então um corte de capacidade mínima (o fluxo máximo na rede é 10 e a capacidade deste conjunto de arcos é também 10)

5.1.4. Formulação em Programação Linear

Dada uma rede $R=(V, A, U)$, considere x_{ij} a variável que representa o fluxo que passa no arco (i,j) e u_{ij} a capacidade desse arco. Designando por F o fluxo total que passa do vértice origem s ao vértice destino t , o problema de Fluxo Máximo pode ser formulado em Programação Linear do seguinte modo:

$$\text{Max } Z = F$$

$$s. a: \sum_{i: (s,i) \in A} x_{si} - \sum_{i: (i,s) \in A} x_{is} = F \quad (\text{a quantidade de fluxo no vértice } s \text{ é } F)$$

$$\sum_{i: (t,i) \in A} x_{ti} - \sum_{i: (i,t) \in A} x_{it} = -F \quad (\text{a quantidade de fluxo no vértice } t \text{ é } F)$$

$$\sum_{i: (j,i) \in A} x_{ji} - \sum_{i: (i,j) \in A} x_{ij} = 0, j \in V: j \neq s, t \quad (\text{todos os vértices diferentes de } s \text{ e } t \text{ são pontos de passagem de fluxo})$$

$$x_{ij} \leq u_{ij}, (i, j) \in A \quad (\text{a quantidade de fluxo que passa em cada arco respeita a sua capacidade})$$

$$x_{ij} \geq 0, (i, j) \in A$$

5.1.5. Algumas Questões

Para terminar, apresentam-se duas questões:

Questão 1: O que fazer se a rede apresentar mais do que um vértice origem?

Sabendo que numa determinada localidade existem 2 vias de entrada possíveis e uma única via de saída. Conhecem-se as vias de comunicação entre as vias de entrada e de saída bem como o tráfego máximo, por unidade de tempo, que nelas pode circular sem que exista engarrafamento. Qual o tráfego máximo que pode dar entrada nesta localidade, por unidade de tempo, para que se consiga evitar engarrafamentos?

Questão 2: O que fazer se um ou mais vértices tiverem restrições de capacidade?

Existe uma rede viária entre uma localidade A e uma localidade B que obriga à passagem por outras localidades intermédias. Sabendo que, numa dessas localidades intermédias, o tráfego viário por hora terá que ser limitado, qual o impacto desta alteração no tráfego que pode circular, por hora, entre a localidade A e a localidade B?

5.2. Problema de Fluxo de Custo Mínimo numa rede

Considere a rede $R=(V, A, P)$ em que $P=(C,U)$, representando C o conjunto de custos unitários associados aos arcos de A e U o conjunto das capacidades associadas aos arcos de A. Chama-se **Fluxo de Custo Mínimo entre os vértices s e t de valor θ** à forma mais económica de enviar θ unidades de fluxo, de um certo bem, do vértice origem s para o vértice destino t.

A título ilustrativo, os

- Problemas de Transportes e Transbordo;
- Problemas de Planeamento de Produção

são exemplos de problemas que podem ser interpretados como um Problema de Fluxo de Custo Mínimo.

5.2.1. Determinação do Fluxo de Custo Mínimo de valor θ

Para determinar o fluxo de custo mínimo entre os vértices s e t recorre-se ao Algoritmo de Busacker-Gowen.

Algoritmo de Busacker-Gowen

Admitindo que se pretende determinar o fluxo de custo mínimo entre os vértices s e t de valor θ na rede $R=(V, A, P)$, e considerando as estruturas:

- $R^* = (V^*, A^*, P^*)$, que representa a rede incremental, onde cada elemento, $u_{ij}^* \geq 0$ e inteiro, da matriz U^* , representa a capacidade do arco $(i, j) \in A^*$ e $c_{ij}^* \geq 0$ representa o custo do arco $(i, j) \in A^*$;
- VF é a variável que representa o valor do sistema de fluxos construído até ao momento;
- f é a variável que representa o valor do fluxo corrente;

O Algoritmo de Busacker-Gowen é constituído pelos seguintes passos:

PASSO 1: Inicialização

- $R^* = R$; $VF = 0$; $f = 0$.

PASSO 2: Criação do fluxo corrente

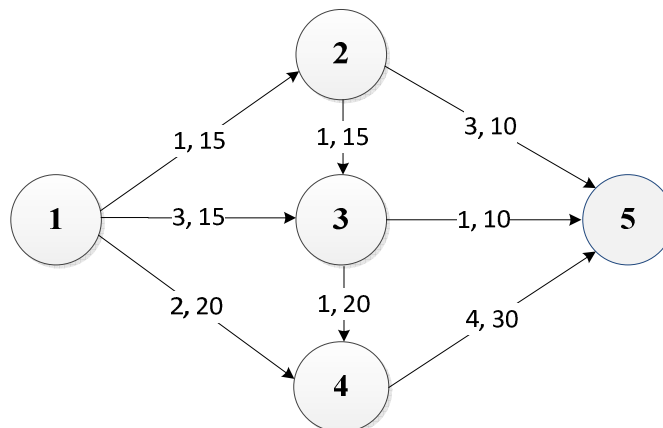
- Determina-se o caminho mais curto (ou mais económico, etc.) entre s e t na rede R^* ;
- Envia-se, pelo caminho determinado, o maior fluxo possível (ou o necessário para atingir o valor θ , i.e., $\theta - VF$). Tal valor é designado por f;
- $VF = VF + f$;
- Se $VF = \theta$ então determinou-se o fluxo de valor θ com custo mínimo e STOP;
- Senão ir para o PASSO 3.

PASSO 3: Actualização da rede incremental R^*

- A rede R^* tem a mesma estrutura básica da rede R;
- Os valores das capacidades, u_{ij}^* , e dos custos, c_{ij}^* , são actualizados da seguinte forma:
 - ❖ se no arco (i, j) circula uma quantidade não nula de fluxo, x_{ij} , então
 - $u_{ij}^* = u_{ij} - x_{ij}$;
 - se $x_{ij} = u_{ij}$ então $c_{ij}^* = +\infty$;
 - criar um arco fictício (j, i) e fazer $u_{ji}^* = x_{ij}$ e $c_{ji}^* = -c_{ij}$;
- Voltar ao PASSO 2.

Exemplo de Aplicação:

Considere a rede $R=(V, A, P)$ com a seguinte representação gráfica:



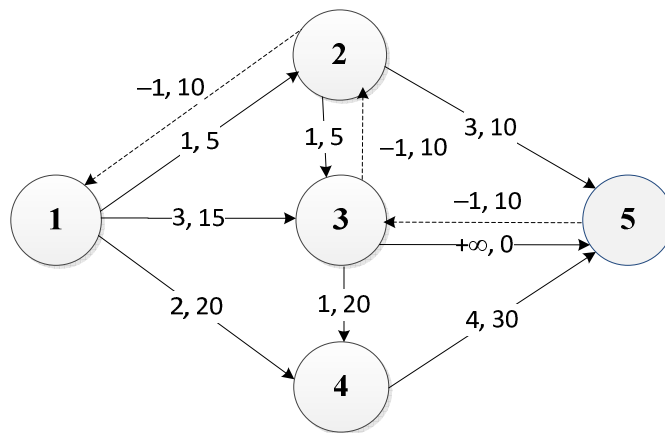
em que os valores a, b indicados junto aos arcos representam, respectivamente, o custo unitário do arco e a capacidade desse arco.

- Resumo de aplicação do Algoritmo de Busacker-Gowen: determinação do fluxo de custo mínimo entre os nodos 1 e 5 de valor $\theta=20$

➤ $R^* = R$; $VF=0$; $f=0$; $\theta=20$;

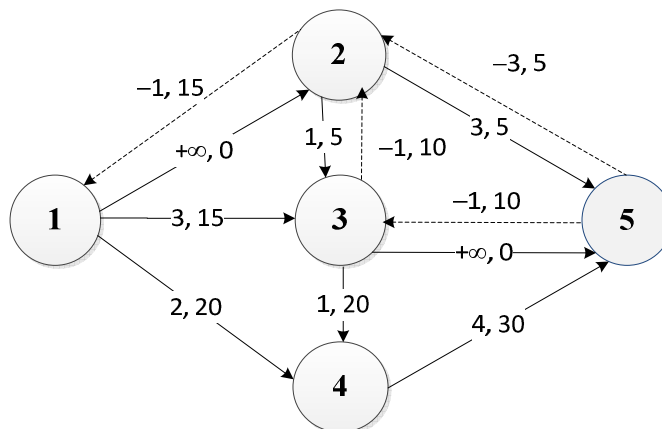
➤ Caminho mais curto ($c+c$) entre 1 e 5 na rede R^* : $1 \rightarrow 2 \rightarrow 3 \rightarrow 5$; custo unitário $= 1 + 1 + 1 = 3$; $f = \min\{15, 15, 10\} = 10$; $VF = 0 + 10 = 10 < \theta$ (avançar para Passo 3);

➤ Actualização da rede incremental R^*



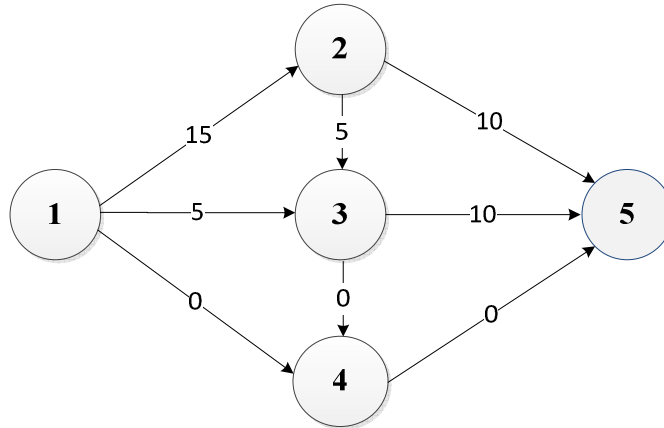
➤ $c+c$ entre 1 e 5 na rede R^* : $1 \rightarrow 2 \rightarrow 5$; custo unitário $= 1 + 3 = 4$; $f = \min\{5, 10\} = 5$; $VF = 10 + 5 = 15 < \theta$ (avançar para Passo 3);

➤ Actualização da rede incremental R^*



➤ $c+c$ entre 1 e 5 na rede R^* : $1 \rightarrow 3 \rightarrow 2 \rightarrow 5$; custo $= 3 - 1 + 3 = 5$; $f = \min\{15, 10, 5\} = 5$; $VF = 15 + 5 = 20 = \theta \Rightarrow \text{STOP}$;

- O fluxo de custo mínimo entre os vértices 1 e 5 de valor $\theta=20$ tem custo $= 3 \times 10 + 4 \times 5 + 5 \times 5 = 1 \times 15 + 3 \times 5 + 1 \times 5 + 3 \times 10 + 1 \times 10 = 75$, sendo o sistema óptimo de fluxos (valor junto a cada arco representa o fluxo que está a passar por esse arco) dado por:



5.2.2. Formulação em Programação Linear

Dada uma rede $R=(V, A, P)$, em que $P=(U,C)$, considere x_{ij} a variável que representa o fluxo que passa no arco (i,j) , c_{ij} o custo unitário (≥ 0) por transitar no arco (i,j) e u_{ij} a capacidade desse arco. Designando por θ o fluxo que passa do vértice s ao vértice t , o problema de Fluxo de Custo Mínimo entre os vértices s e t de valor θ pode ser formulado em Programação Linear do seguinte modo:

$$\text{Min } z = \sum_{(i,j) \in A} c_{ij} x_{ij}$$

$$\text{s. a: } \sum_{i: (s,i) \in A} x_{si} - \sum_{i: (i,s) \in A} x_{is} = \theta \quad (\text{a quantidade de fluxo no vértice } s \text{ é } \theta)$$

$$\sum_{i: (t,i) \in A} x_{ti} - \sum_{i: (i,t) \in A} x_{it} = -\theta \quad (\text{a quantidade de fluxo no vértice } t \text{ é } \theta)$$

$$\sum_{i: (j,i) \in A} x_{ji} - \sum_{i: (i,j) \in A} x_{ij} = 0, \quad j \in V: j \neq s, t \quad (\text{todos os vértices diferentes de } s \text{ e } t \text{ são pontos de passagem de fluxo})$$

$$x_{ij} \leq u_{ij}, \quad (i, j) \in A \quad (\text{a quantidade de fluxo que passa em cada arco respeita a sua capacidade})$$

$$x_{ij} \geq 0, \quad (i, j) \in A$$

5.2.3. Problemas que podem ser formulados como um Problema de Fluxo de Custo Mínimo

De entre os problemas que podem ser formulados como um Problema de Fluxo de Custo Mínimo, destacam-se o Problema do Caminho mais Curto, o Problema do Fluxo Máximo, o Problema de Transportes e o Problema de Afectação.

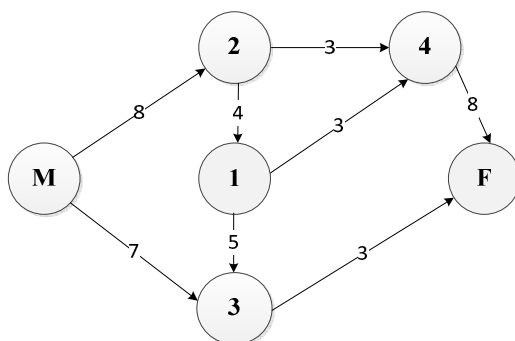
Problema do Caminho mais Curto

Dada uma rede $R=(V, A, C)$ em que C representa o conjunto de “custos” associados aos arcos de A . O problema de determinar o caminho mais curto entre s e t corresponde a determinar o fluxo de custo mínimo entre s e t , de valor $\theta=1$, na rede em que se considera que:

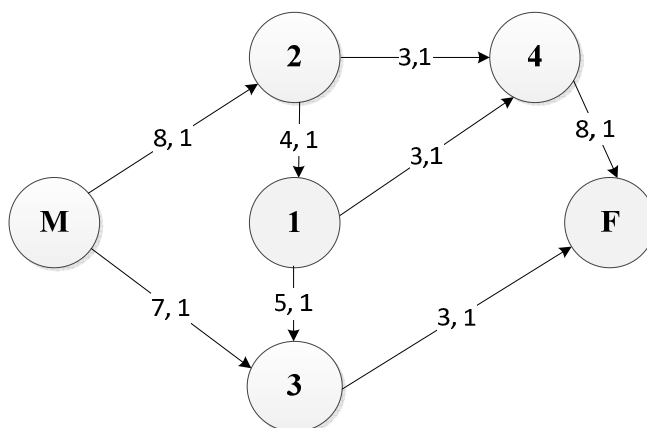
- a capacidade de cada arco $(i,j) \in A$ é igual a 1 (Nota: a capacidade pode ser qualquer valor superior ou igual a 1);
- o custo unitário de cada arco $(i,j) \in A$ é igual ao seu “custo” c_{ij} .

Exemplo:

Considere a rede $R=(V, A, C)$ em que o valor junto a cada arco representa o comprimento que lhe está associado:



O caminho mais curto entre M e F corresponde ao fluxo de custo mínimo entre M e F de valor $\theta=1$ na seguinte rede:



onde, os valores indicados junto aos arcos representam, respectivamente, o custo unitário do arco e a capacidade desse arco.

Problema do Fluxo Máximo

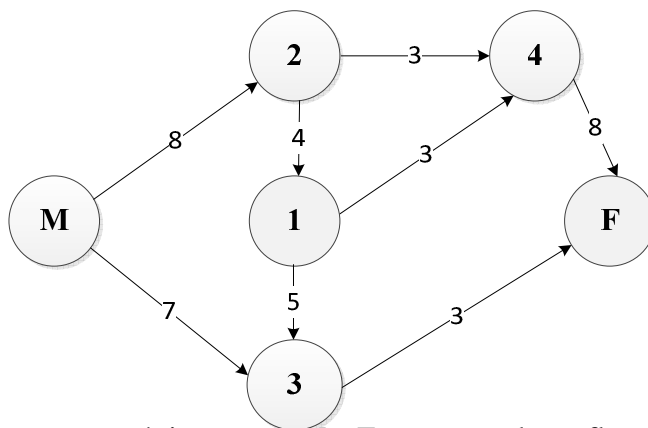
Dada uma rede $R=(V, A, U)$ em que U representa o conjunto de capacidades associadas aos arcos de A . O problema de determinar o fluxo máximo entre s e t corresponde a determinar o fluxo de custo mínimo entre s e t , de valor θ , onde θ é um limite superior do valor do fluxo máximo na rede R :

- se considera que o custo unitário em cada arco $(i,j) \in A$ é igual a zero; e que
- se cria um arco entre a origem s e o destino t com custo unitário de valor $+\infty$ e com capacidade ilimitada.

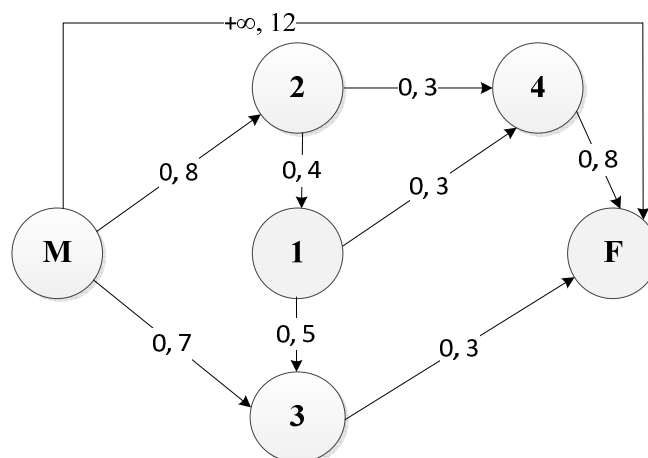
O valor do fluxo máximo entre s e t será dado por θ —Valor do fluxo que passa no arco (s,t) , ou seja, $\theta-x_{st}$.

Exemplo:

Considere a rede $R=(V, A, U)$ em que o valor junto a cada arco representa a capacidade que lhe está associada:



O fluxo máximo entre M e F corresponde ao fluxo de custo mínimo entre M e F de valor, por exemplo, $\theta=12$ (θ pode assumir qualquer valor superior ou igual ao $\min\{u_{M2}+u_{M3}; u_{4F}+u_{3F}\}=\min\{8+7, 8+3\}=11$) na seguinte rede:



onde, os valores indicados junto aos arcos representam, respectivamente, o custo unitário do arco e a capacidade desse arco.

Problema do Transportes

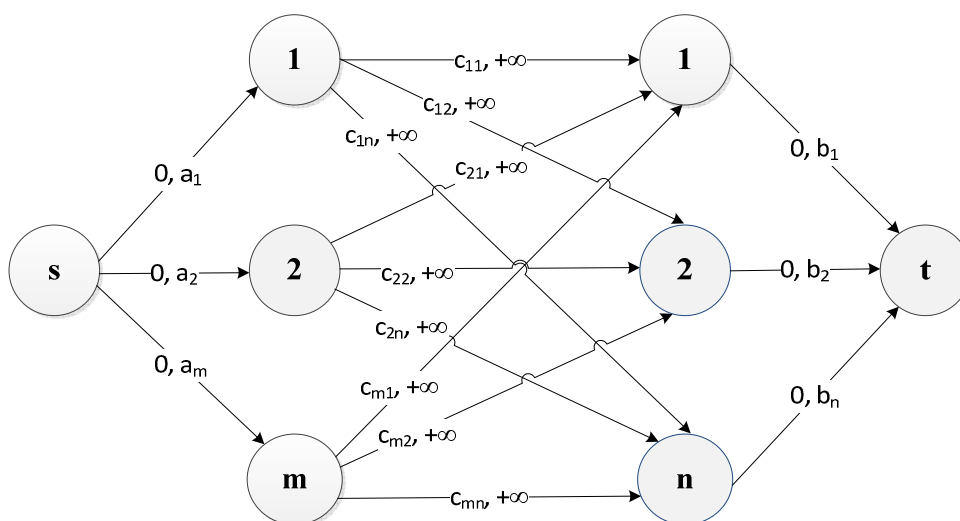
Considere um Problema de Transportes com m origens e n destinos, em que:

- cada origem i tem a_i unidades disponíveis de um certo bem;
- cada destino j necessita de b_j unidades do bem;
- o custo unitário de transporte da origem i para o destino j é c_{ij} ;
- $\sum_{i=1}^m a_i = \sum_{j=1}^n b_j$.

Para formular o Problema de Transportes como um Problema de Fluxo de Custo Mínimo é necessário construir a rede $R=(V, A, P)$ em que:

- os vértices de V são as m origens, os n destinos, o vértice inicial s e o vértice final t ;
- os arcos de A são: $(s, i), i=1, \dots, m$; $(i, j), i=1, \dots, m, j=1, \dots, n$; e $(j, t), j=1, \dots, n$;
- os arcos (s, i) têm custo unitário nulo e capacidade a_i ;
- os arcos $(i, j), i=1, \dots, m; j=1, \dots, n$, têm custo unitário c_{ij} e capacidade infinita;
- os arcos (j, t) têm custo unitário nulo e capacidade b_j .

A rede $R=(V, A, P)$ pode assumir a seguinte representação gráfica:



Na rede $R=(V, A, P)$, o Problema de Transportes corresponde a determinar o fluxo de custo mínimo entre s e t de valor $\theta = \sum_{i=1}^m a_i = \sum_{j=1}^n b_j$.

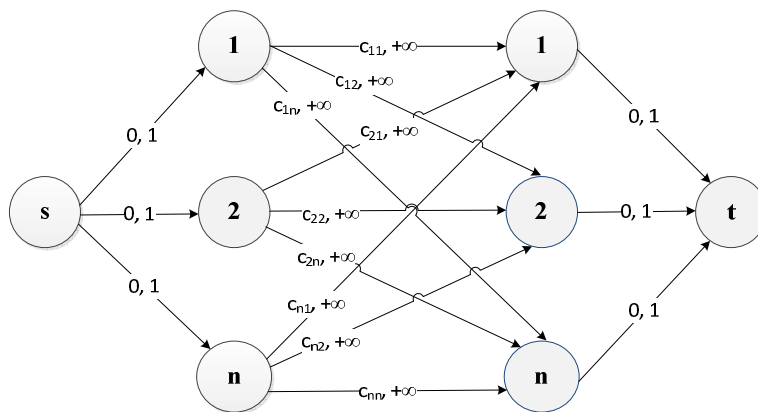
Problema de Afecção

Considere um Problema de Afecção com n tarefas e n agentes, em que o custo de afectar a tarefa i ao agente j é c_{ij} , $i, j=1, \dots, n$.

Para formular o Problema de Afecção como um Problema de Fluxo de Custo Mínimo é necessário construir a rede $R=(V, A, P)$ em que:

- os vértices de V são as n tarefas, os n agentes, o vértice inicial s e o vértice final t ;
- os arcos de A são: (s, i) , $i=1, \dots, n$; (i, j) , $i, j=1, \dots, n$; e (j, t) , $j=1, \dots, n$;
- os arcos (s, i) têm custo unitário nulo e capacidade 1 ;
- os arcos (i, j) , $i, j=1, \dots, n$, têm custo unitário c_{ij} e capacidade infinita;
- os arcos (j, t) têm custo unitário nulo e capacidade 1 .

A rede $R=(V, A, P)$ pode assumir a seguinte representação gráfica:



Na rede $R=(V, A, P)$, o Problema de Afecção corresponde a determinar o fluxo de custo mínimo entre s e t de valor $\theta=n$.