



Universidade do Minho

Escola de Engenharia

Aula Prática 4: Algoritmos específicos para a obtenção da árvore dos caminhos mais curtos

Elementos de Engenharia de Sistemas

2019/2020

Árvore dos caminhos mais curtos

- **Objetivo:** Determinar a árvore dos caminhos mais curtos entre a raiz e os restantes nodos.
- Vamos começar por determinar se um grafo é **cíclico** ou **acíclico** e, consoante o resultado, aplicar um determinado algoritmo para obter a árvore dos caminhos mais curtos.

Ordenação topológica

- Um grafo está ordenado topologicamente se o nodo de origem de qualquer arco tiver índice inferior ao índice do nodo de destino, isto é,

$$i < j, \quad \forall ij \in A.$$

- Um algoritmo de ordenação topológica é apresentado na pág. 55 da sebenta. Este algoritmo permite determinar se uma rede é acíclica (caso em que retorna a ordenação topológica) ou cíclica (caso em que a ordenação fica incompleta).

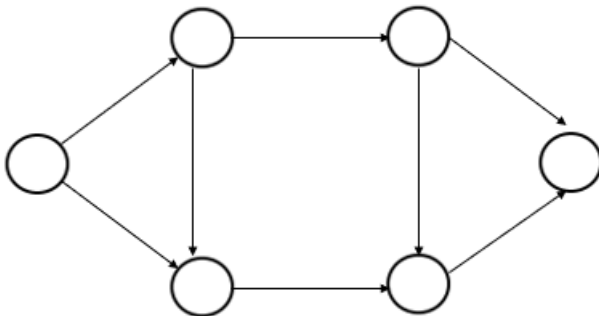
Ordenação topológica

```
// Algoritmo de ordenação topológica
// Vector index fica com o novo índice para cada nodo
j=1
Enquanto existirem nodos com grau de entrada = 0
    Seleccionar um nodo i com grau de entrada = 0
    index[i]=j
    Remover da rede todos os arcos com origem em i
    Remover da rede o nodo i
    j=j+1
Se rede está vazia, ordenação topológica concluída
Se não, existe um ciclo (orientado)
```

Algoritmo de ordenação topológica (pág. 55 da sebenta).

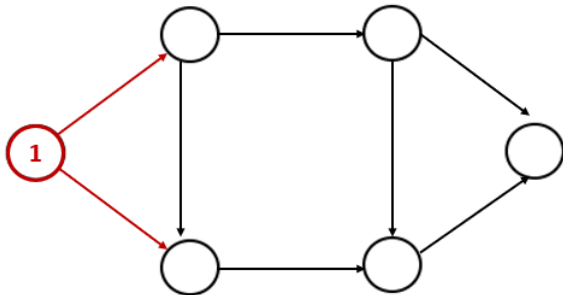
Ordenação topológica

- **Exemplo 1:** Determine se o grafo da figura é cíclico ou acíclico através da utilização de um algoritmo.



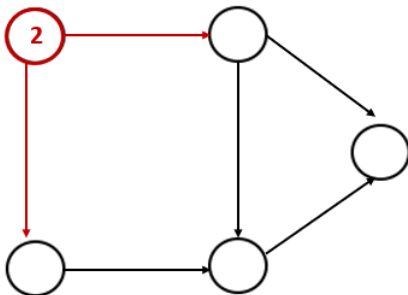
Ordenação topológica

- **Iteração 1:** O único nodo com grau de entrada 0 é o nodo mais à esquerda, identificado a vermelho. Logo, é-lhe atribuído o índice 1. De seguida, o nodo é removido da rede, bem como os arcos com origem nesse nodo.



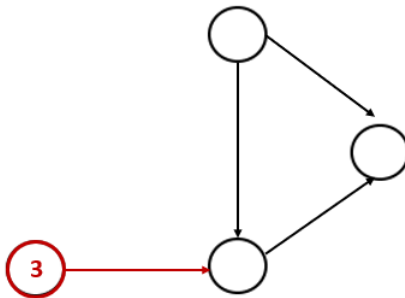
Ordenação topológica

■ Iteração 2



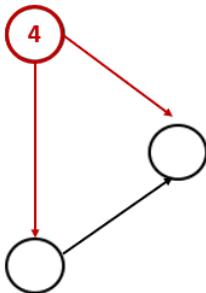
Ordenação topológica

- **Iteração 3:** Existem dois nodos com grau de entrada 0. Escolhemos arbitrariamente um deles (neste caso, é escolhido o que está em baixo, à esquerda), atribuindo-lhe o índice 3.



Ordenação topológica

■ Iteração 4



Ordenação topológica

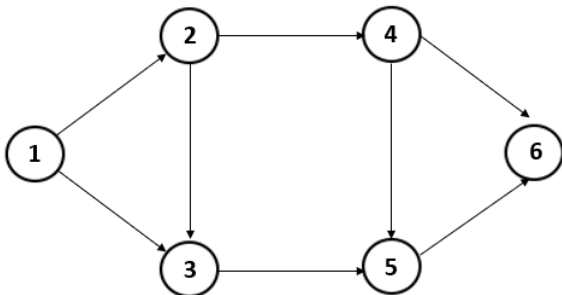
■ Iterações 5 e 6



- Não existem mais nodos com grau de entrada 0 e a rede está vazia. Logo, conclui-se que o grafo é acíclico e a ordenação topológica está completa.

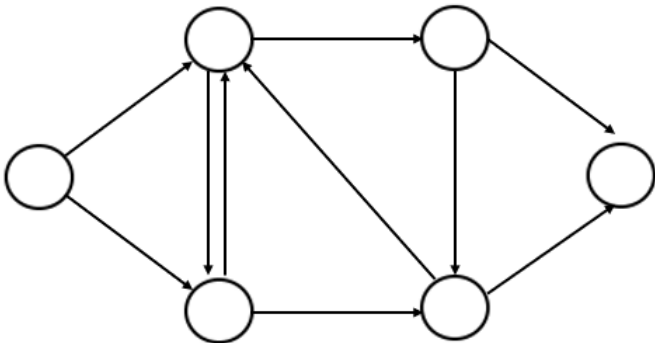
Ordenação topológica

- Assim, uma ordenação topológica para a rede inicial é:



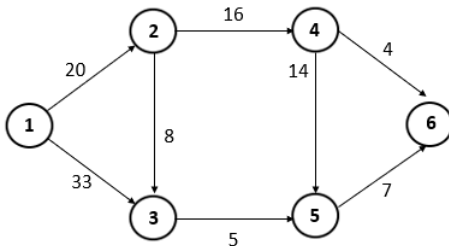
Ordenação topológica

- **Exercício:** Determine, através da utilização de um algoritmo, se a seguinte rede é cíclica ou acíclica.



Algoritmo para determinar a árvore dos caminhos mais curtos para redes acíclicas

- **Exemplo 2:** Obtenha a árvore dos caminhos mais curtos com raiz em 1 para o grafo da figura seguinte, utilizando o algoritmo para redes acíclicas.



Algoritmo para determinar a árvore dos caminhos mais curtos para redes acíclicas

- **Nota:** Este algoritmo é utilizado em redes acíclicas **ordenadas topologicamente**.
- Será construída uma tabela com n linhas e n colunas, onde cada célula (i, j) é constituída por um par (distância, predecessor) $= (d_j, i)$, onde d_j corresponde à distância da raiz ao nodo j e i corresponde ao índice do nodo a ser analisado numa dada iteração.
- Os nodos são analisados por ordem crescente, de acordo com a ordenação topológica obtida.

Algoritmo para determinar a árvore dos caminhos mais curtos para redes acíclicas

- **Iteração 0:** ($i = 0$) é feita a inicialização dos pares (d_j, i) para cada nodo da rede. Para o nodo 1, a distância d_1 é inicializada com 0. Para os restantes nodos, a distância é inicializada com um valor “muito elevado”, $d_j = \text{inf}$.
- Nas iterações seguintes, vamos comparar o valor de d_j com $d'_j = d_i + c_{ij}$, onde c_{ij} é o comprimento do arco ij . Se $d_j > d'_j$, significa que foi encontrado um caminho mais curto para o nodo j , logo, atualizamos o valor de d_j para d'_j .

Algoritmo para determinar a árvore dos caminhos mais curtos para redes acíclicas

i, j	1	2	3	4	5	6
0	(0,0)	(inf,0)	(inf,0)	(inf,0)	(inf,0)	(inf,0)
1						

Inicialização do algoritmo para redes acíclicas.

Algoritmo para determinar a árvore dos caminhos mais curtos para redes acíclicas

- **Iteração 1:** os arcos com origem no nodo 1 são o arco 12 e 13. Queremos determinar se d_2 e d_3 vão ser “atualizados” nesta iteração. Temos:
- $d_2 = \text{inf}$ e $d'_2 = d_1 + c_{12} = 0 + 20 = 20$. Como $d_2 > d'_2$, d_2 é atualizado para o novo valor 20.
 - $d_3 = \text{inf}$ e $d'_3 = d_1 + c_{13} = 0 + 33 = 33$. Como $d_3 > d'_3$, d_3 é atualizado para o novo valor 33.

i, j	1	2	3	4	5	6
0	(0,0)	(inf,0)	(inf,0)	(inf,0)	(inf,0)	(inf,0)
1		(20,1)	(33,1)			

Algoritmo para determinar a árvore dos caminhos mais curtos para redes acíclicas

- **Iteração 2:** os arcos com origem no nodo 2 são o arco 23 e o 24. Queremos determinar se d_3 e d_4 vão ser “atualizados” nesta iteração. Temos:
 - $d_3 = 33$ e $d'_3 = d_2 + c_{23} = 20 + 8 = 28$. Como $d_3 > d'_3$, d_3 é atualizado para o novo valor 28.
 - $d_4 = \text{inf}$ e $d'_4 = d_2 + c_{24} = 20 + 16 = 36$. Como $d_4 > d'_4$, d_4 é atualizado para o novo valor 36.

i, j	1	2	3	4	5	6
0	(0,0)	(inf,0)	(inf,0)	(inf,0)	(inf,0)	(inf,0)
1		(20,1)	(33,1)			
2			(28,2)	(36,2)		

Algoritmo para determinar a árvore dos caminhos mais curtos para redes acíclicas

- **Iteração 3:** o único arco com origem no nodo 3 é o arco 35.

Temos:

- $d_5 = \text{inf}$ e $d'_5 = d_3 + c_{35} = 28 + 5 = 33$. Como $d_5 > d'_5$, d_5 é atualizado para o novo valor 33.

i, j	1	2	3	4	5	6
0	(0,0)	(inf,0)	(inf,0)	(inf,0)	(inf,0)	(inf,0)
1		(20,1)	(33,1)			
2			(28,2)	(36,2)		
3					(33,3)	

Algoritmo para determinar a árvore dos caminhos mais curtos para redes acíclicas

■ Iteração 4:

- $d_5 = 33$ e $d'_5 = d_4 + c_{45} = 36 + 14 = 50$. Como $d_5 < d'_5$, d_5 não é atualizado.
- $d_6 = \text{inf}$ e $d'_6 = d_4 + c_{46} = 36 + 4 = 40$. Como $d_6 > d'_6$, d_6 é atualizado para 40.

i, j	1	2	3	4	5	6
0	(0,0)	(inf,0)	(inf,0)	(inf,0)	(inf,0)	(inf,0)
1		(20,1)	(33,1)			
2			(28,2)	(36,2)		
3					(33,3)	
4					(50,4)	(40,4)

Algoritmo para determinar a árvore dos caminhos mais curtos para redes acíclicas

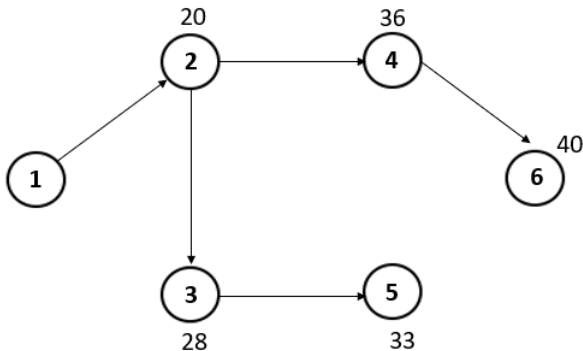
■ Iteração 5:

- $d_6 = 40$ e $d'_6 = d_5 + c_{56} = 33 + 7 = 40$. Como $d_6 = d'_6$, o valor de d_6 não é atualizado.

i, j	1	2	3	4	5	6
0	(0,0)	(inf,0)	(inf,0)	(inf,0)	(inf,0)	(inf,0)
1		(20,1)	(33,1)			
2			(28,2)	(36,2)		
3					(33,3)	
4					(50,4)	(40,4)
5						(40,5)

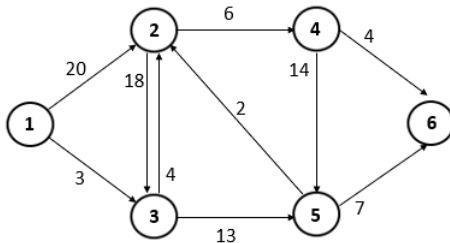
Algoritmo para determinar a árvore dos caminhos mais curtos para redes acíclicas

- A árvore dos caminhos mais curtos resultante é apresentada a seguir.



Árvore dos caminhos mais curtos para redes sem custos negativos: algoritmo de Dijkstra

- **Exemplo 3:** Obtenha a árvore dos caminhos mais curtos para o grafo da figura seguinte.



Árvore dos caminhos mais curtos para redes sem custos negativos: algoritmo de Dijkstra

- Algoritmo aplicado em redes sem custos negativos (em particular, em redes cíclicas.)
- À semelhança do algoritmo para redes acíclicas, é criada uma tabela onde cada célula contém o par (d_j, i) , (distância, predecessor). Algumas destas células são marcadas como permanentes (identificadas com uma etiqueta à sua volta) e outras serão células temporárias.
- Em cada iteração, são feitas atualizações de acordo com a fórmula $d_j > d_i + c_{ij}$. De seguida, entre **todas** as células temporárias, é marcada como permanente aquela que tem a menor distância. O nodo a ser analisado na iteração seguinte é o correspondente à coluna onde foi marcada a célula permanente.

Árvore dos caminhos mais curtos para redes sem custos negativos: algoritmo de Dijkstra

- **Iteração 0:** ($i = 0$) é feita a inicialização dos pares (d_j, i) para cada nodo da rede. Para o nodo 1, a distância d_1 é inicializada com 0. Para os restantes nodos $j = 2, \dots, n$, a distância é inicializada como um valor "muito elevado", $d_j = \infty$.

i, j	1	2	3	4	5	6
0	(0,0)	(inf,0)	(inf,0)	(inf,0)	(inf,0)	(inf,0)
1						

Inicialização do algoritmo de Dijkstra.

Árvore dos caminhos mais curtos para redes sem custos negativos: algoritmo de Dijkstra

■ Iteração 1:

- 1 $d_2 = \text{inf}$ e $d'_2 = d_1 + c_{12} = 0 + 20 = 20$. Como $d_2 > d'_2$, d_2 é atualizado para o novo valor 20.
- 2 $d_3 = \text{inf}$ e $d'_3 = d_1 + c_{13} = 0 + 3 = 3$. Como $d_3 > d'_3$, d_3 é atualizado para o novo valor 3.
- 3 As células com etiquetas temporárias são (20,1) e (3,1). A célula com menor distância é (3,1), logo, esta é marcada como permanente.

i, j	1	2	3	4	5	6
0	(0,0)	(inf,0)	(inf,0)	(inf,0)	(inf,0)	(inf,0)
1		(20,1)	(3,1)			

Árvore dos caminhos mais curtos para redes sem custos negativos: algoritmo de Dijkstra

- **Iteração 2:** Uma vez que na iteração anterior foi marcada como permanente a célula na coluna 3, iremos analisar o nodo 3. Os arcos com origem neste nodo são o 32 e o 35.
 - 1 $d_2 = 20$ e $d'_2 = d_3 + c_{32} = 3 + 4 = 7$. Como $d_2 > d'_2$, d_2 é atualizado para o novo valor 7.
 - 2 $d_5 = \infty$ e $d'_5 = d_3 + c_{35} = 3 + 13 = 16$. Como $d_5 > d'_5$, d_5 é atualizado para o novo valor 16.
 - 3 As células com etiquetas temporárias são (7, 3) e (16, 3). Assim, a célula que passa a ter uma etiqueta permanente é a célula (7, 3). Logo, na próxima iteração, irá ser analisado o nodo 2.

Árvore dos caminhos mais curtos para redes sem custos negativos: algoritmo de Dijkstra

i, j	1	2	3	4	5	6
0	(0,0)	(inf,0)	(inf,0)	(inf,0)	(inf,0)	(inf,0)
1		(20,1)	(3,1)			
3		(7,3)			(16,3)	

Iteração 2 do algoritmo de Dijkstra.

Árvore dos caminhos mais curtos para redes sem custos negativos: algoritmo de Dijkstra

- **Iteração 3:** os arcos com origem no nodo 2 são o 23 e o 24. Uma vez que já foi marcado o caminho mais curto para o nodo 3, o arco 23 não será analisado.
 - 1 $d_4 = \inf$ e $d'_4 = d_2 + c_{24} = 7 + 5 = 13$. Como $d_4 > d'_4$, d_4 é atualizado para o novo valor 13.
 - 2 As células com etiquetas temporárias são (13, 2) e (16, 3) (da iteração anterior). Assim, a célula que passa a ter uma etiqueta permanente é a célula (13, 2). Logo, na próxima iteração, irá ser analisado o nodo 4.

Árvore dos caminhos mais curtos para redes sem custos negativos: algoritmo de Dijkstra

i, j	1	2	3	4	5	6
0	(0,0)	(inf,0)	(inf,0)	(inf,0)	(inf,0)	(inf,0)
1		(20,1)	(3,1)			
3		(7,3)			(16,3)	
2				(13,2)		

Iteração 3 do algoritmo de Dijkstra.

Árvore dos caminhos mais curtos para redes sem custos negativos: algoritmo de Dijkstra

- **Iteração 4:** os arcos com origem no nodo 4 são o 45 e o 46.
 - 1 $d_5 = 16$ e $d'_5 = d_4 + c_{45} = 13 + 14 = 27$. Como $d_5 < d'_5$, d_5 não é atualizado.
 - 2 $d_6 = \inf$ e $d'_6 = d_4 + c_{46} = 13 + 4 = 17$. Como $d_6 > d'_6$, d_6 é atualizado para o novo valor 17.
 - 3 As células com etiquetas temporárias são $(16, 3)$ e $(17, 4)$. Assim, a célula que passa a ter uma etiqueta permanente é a célula $(16, 3)$. Logo, na próxima iteração, irá ser analisado o nodo 5.

Árvore dos caminhos mais curtos para redes sem custos negativos: algoritmo de Dijkstra

i, j	1	2	3	4	5	6
0	(0,0)	(inf,0)	(inf,0)	(inf,0)	(inf,0)	(inf,0)
1		(20,1)	(3,1)			
3		(7,3)			(16,3)	
2				(13,2)		
4					(16,3)	(17,4)

Iteração 4 do algoritmo de Dijkstra.

Árvore dos caminhos mais curtos para redes sem custos negativos: algoritmo de Dijkstra

■ Iteração 5:

- 1 $d_6 = 17$ e $d'_6 = d_5 + c_{56} = 16 + 7 = 23$. Como $d_6 < d'_6$, d_6 não é atualizado.
- 2 Temos apenas uma célula com uma etiqueta temporária, $(17, 4)$. Assim, esta célula passa a ter uma etiqueta permanente e concluímos a aplicação do algoritmo.

Árvore dos caminhos mais curtos para redes sem custos negativos: algoritmo de Dijkstra

i, j	1	2	3	4	5	6
0	(0,0)	(inf,0)	(inf,0)	(inf,0)	(inf,0)	(inf,0)
1		(20,1)	(3,1)			
3		(7,3)			(16,3)	
2				(13,2)		
4					(16,3)	(17,4)
5						(17,4)

Iteração 5 do algoritmo de Dijkstra.

Árvore dos caminhos mais curtos para redes sem custos negativos: algoritmo de Dijkstra

- A árvore dos caminhos mais curtos resultante é apresentada a seguir.

