Caminho Mínimo (1)

Zenilton Patrocínio

Caminho Mínimo

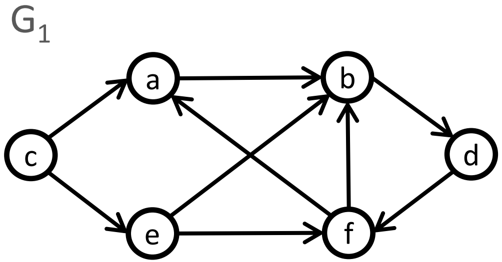
O caminho mais curto entre os vértices de um grafo não ponderado é aquele que possui o menor número de arestas e pode ser obtido por meio de uma busca em largura.

Já o caminho mais curto entre os vértices de um grafo ponderado é aquele cuja soma dos pesos das arestas possui o menor valor possível dentre todos os caminhos existentes.

Claramente, em grafos ponderados, o menor caminho pode não ser aquele com menor número de arestas.

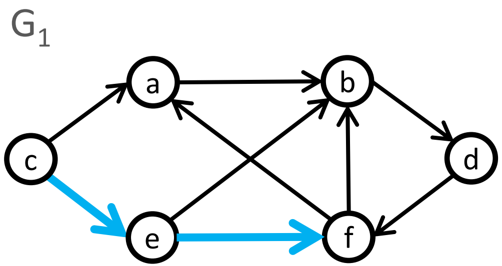
Exemplo

No grafo não ponderado G_1, o caminho mínimo do vértice c para o vértice f possui **duas** arestas.



Exemplo

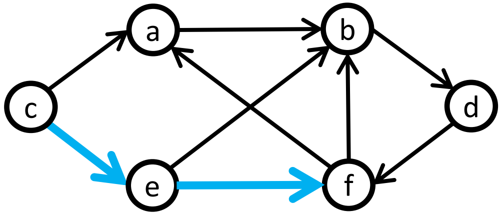
No grafo não ponderado G_1, o caminho mínimo do vértice c para o vértice f possui **duas** arestas.



Exemplo

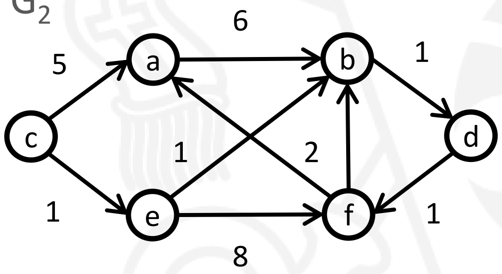
No grafo não ponderado G_1, o caminho mínimo do vértice c para o vértice f possui **duas** arestas.

G_1



G_2

d_{ab} = Peso ou comprimento da aresta (a,b)

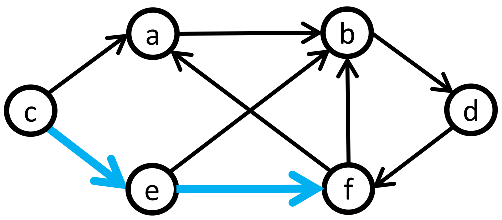


Exemplo

No grafo não ponderado G_1, o caminho mínimo do vértice c para o vértice f possui **duas** arestas.

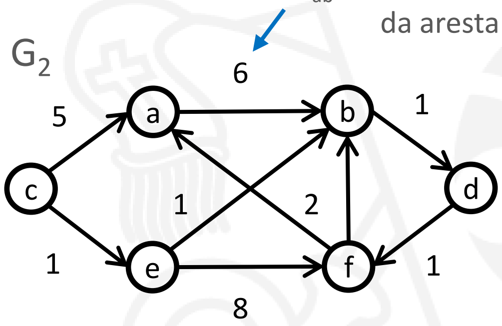
Já no grafo ponderado G_2, o caminho mínimo do vértice c para o vértice f possui **quatro** arestas.

G_1



G_2

d_{ab} = Peso ou comprimento da aresta (a,b)

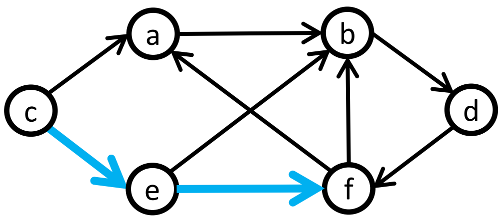


Exemplo

No grafo não ponderado G_1, o caminho mínimo do vértice c para o vértice f possui **<u>duas</u>** arestas.

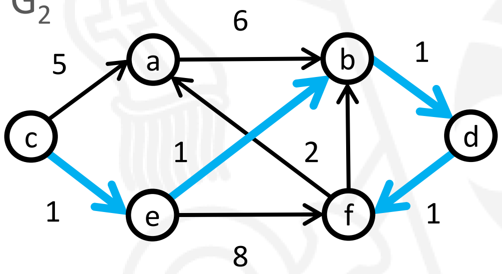
Já no grafo ponderado G_2, o caminho mínimo do vértice c para o vértice f possui **<u>quatro</u>** arestas.

G_1



d_{ab} = Peso ou comprimento da aresta (a,b)

G_2



Corte – Lembrete

Dado um grafo direcionado $G = (V, E)$ e um subconjunto de vértice $S \subset V$, um corte é um subconjunto das arestas com um extremo em S e o outro não, isto é,

$$\text{corte}(S) = \{ (v, w) \in E \mid v \in S, w \notin S \}.$$

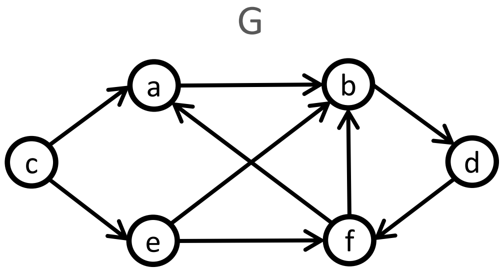
O conjunto de vértices S não pode ser vazio nem conter todos os vértices.

Corte – Lembrete

Dado um grafo direcionado $G = (V, E)$ e um subconjunto de vértice $S \subset V$, um corte é um subconjunto das arestas com um extremo em S e o outro não, isto é,

$$\text{corte}(S) = \{ (v, w) \in E \mid v \in S, w \notin S \}.$$

O conjunto de vértices S não pode ser vazio nem conter todos os vértices.

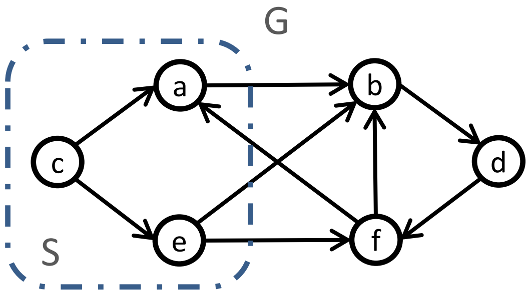


Corte – Lembrete

Dado um grafo direcionado $G = (V, E)$ e um subconjunto de vértice $S \subset V$, um corte é um subconjunto das arestas com um extremo em S e o outro não, isto é,

$$\text{corte}(S) = \{ (v, w) \in E \mid v \in S, w \notin S \}.$$

O conjunto de vértices S não pode ser vazio nem conter todos os vértices.



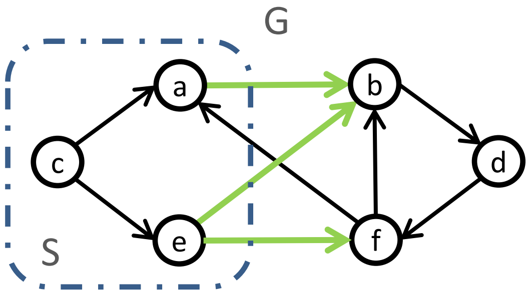
Seja $S = \{a, c, e\}$

Corte – Lembrete

Dado um grafo direcionado $G = (V, E)$ e um subconjunto de vértice $S \subset V$, um corte é um subconjunto das arestas com um extremo em S e o outro não, isto é,

$$\text{corte}(S) = \{ (v, w) \in E \mid v \in S, w \notin S \}.$$

O conjunto de vértices S não pode ser vazio nem conter todos os vértices.



Seja S = {a, c, e}

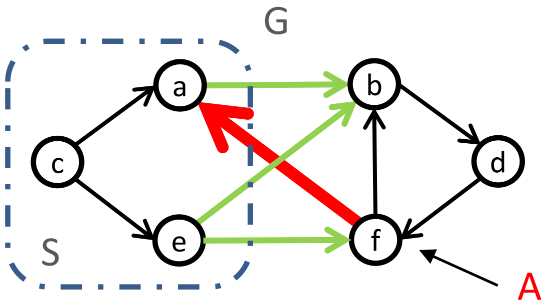
Então corte(S) = { (a,b), (e,b), (e,f) }

Corte – Lembrete

Dado um grafo direcionado G = (V, E) e um subconjunto de vértice $S \subset V$, um corte é um subconjunto das arestas com um extremo em S e o outro não, isto é,

$$\text{corte}(S) = \{ (v, w) \in E \mid v \in S, w \notin S \}.$$

O conjunto de vértices S não pode ser vazio nem conter todos os vértices.



Seja S = {a, c, e}

Então corte(S) = { (a,b), (e,b), (e,f) }

Aresta (f, a) não pertence ao corte(S) !!

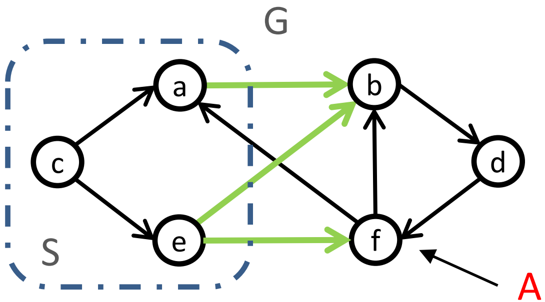
Corte – Lembrete

Dado um grafo direcionado G = (V, E) e um subconjunto de vértice $S \subset V$, um corte é um subconjunto das arestas com um extremo em S e o outro não, isto é,

$$\text{corte}(S) = \{ (v, w) \in E \mid v \in S, w \notin S \}.$$

O conjunto de vértices S não pode ser vazio nem conter todos os vértices.

A remoção das arestas do corte(S) elimina os todos caminhos entre os vértices pertencentes a S e os demais.



Seja S = {a, c, e}

Então corte(S) = { (a,b), (e,b), (e,f) }

Aresta (f, a) não pertence ao corte(S) !!

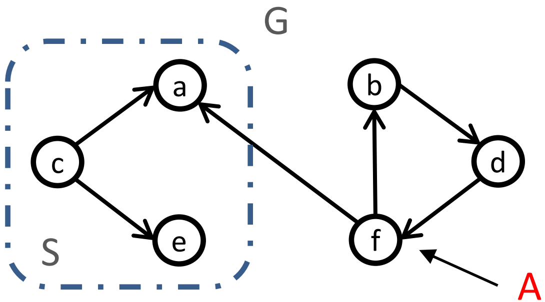
Corte – Lembrete

Dado um grafo direcionado G = (V, E) e um subconjunto de vértice $S \subset V$, um corte é um subconjunto das arestas com um extremo em S e o outro não, isto é,

$$\text{corte}(S) = \{ (v, w) \in E \mid v \in S, w \notin S \}.$$

O conjunto de vértices S não pode ser vazio nem conter todos os vértices.

A remoção das arestas do corte(S) elimina os todos caminhos entre os vértices pertencentes a S e os demais.



Seja S = {a, c, e}

Então corte(S) = { (a,b), (e,b), (e,f) }

Aresta (f, a) não pertence ao corte(S) !!

Método de Dijkstra

Método de Dijkstra

O método proposto por Dijkstra em 1959 rotula vértices durante a exploração do grafo de forma a obter o menor caminho entre um vértice de origem (ou raiz) e todos os demais (sendo semelhante a uma busca em largura).

Inicialmente, um conjunto contendo apenas a raiz é usado para se definir um corte.

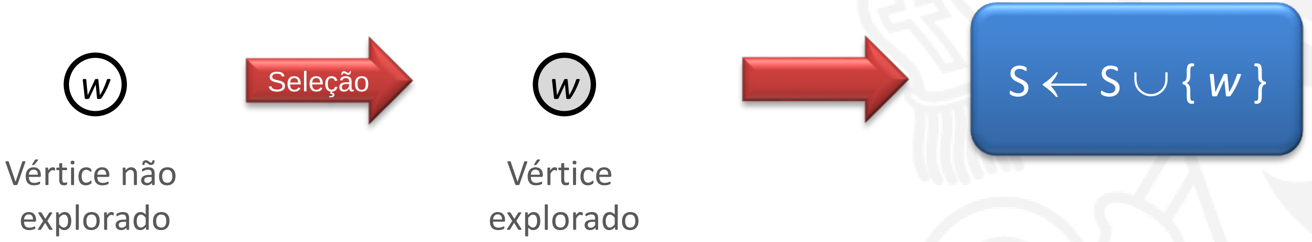
A cada iteração, calcula-se uma estimativa da distância entre os vértices já explorados e seus vizinhos não explorados (analisar apenas as arestas do corte).

A aresta que fornecer a menor estimativa é selecionada e seu extremo não selecionado passa a ser explorado e sua distância é atualizada.

Método de Dijkstra

Durante o cálculo dos caminhos, alguns atributos são definidos para os vértices:

- Inicialmente, um único vértice (denominado raiz) está no conjunto S de vértices selecionados e que define um corte representado por corte(S).
- Cada um dos vértices restantes é inicialmente não explorado (ou branco) e se torna explorado (ou cinza) quando for selecionado e inserido no conjunto S.



Método de Dijkstra

Durante o cálculo dos caminhos, alguns atributos são definidos para os vértices:

- Cada vértice tem dois valores associados a ele: **distância** (indica o comprimento do caminho da raiz até ele) e **predecessor** (indica o vértice pelo qual passa o caminho até ele).

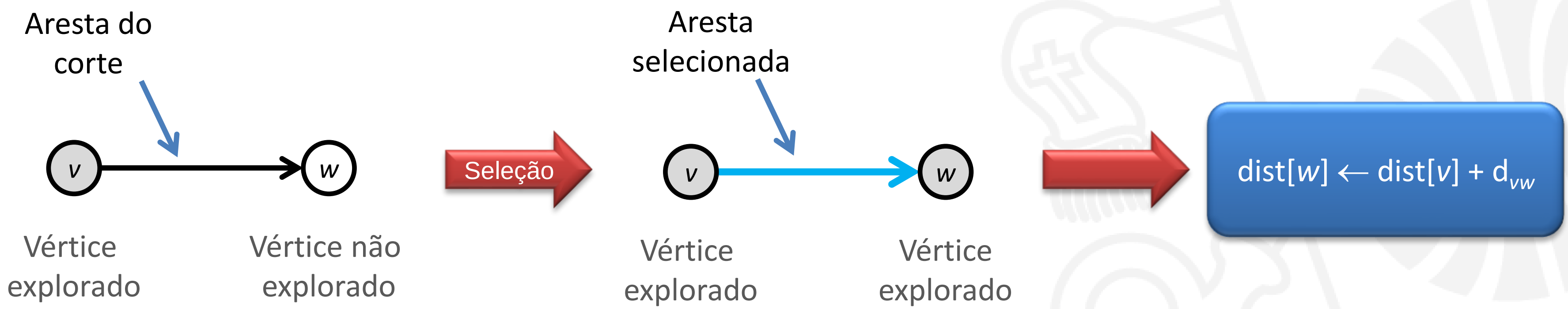
	v_1	v_2	v_2	...	v_{n-1}	v_n
Distância						
Predecessor						

- Inicialmente, todos os **predecessores** são indefinidos (ou ∅), enquanto que os valores de **distância** são infinitos (exceto para a raiz, cuja distância é zero).

Método de Dijkstra

Durante o cálculo dos caminhos, alguns atributos são definidos para os vértices:

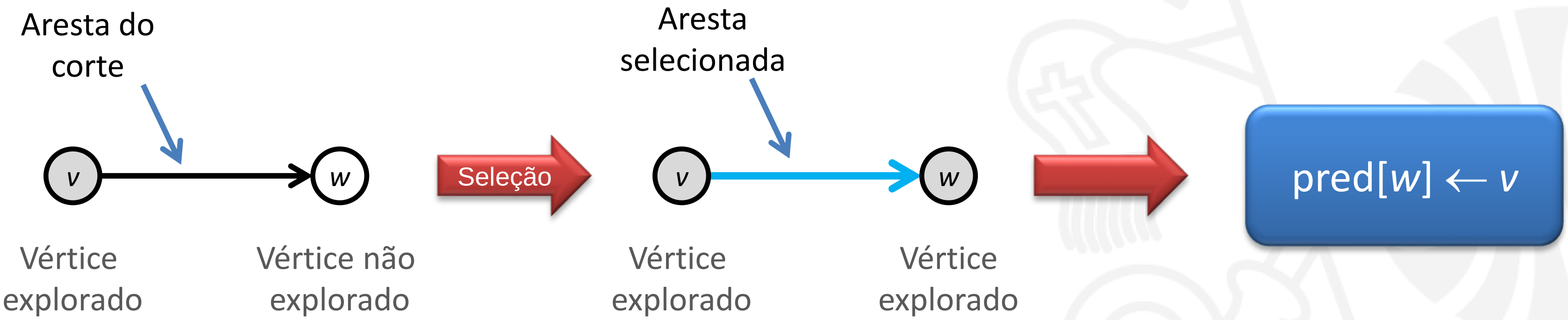
- A cada iteração, pode-se usar o peso d_{vw} da aresta $(v, w) \in$ corte(S) para uma estimativa da distância da raiz até w (ou dist-est[w]) como dist[v] + d_{vw}
- Assim, seleciona-se a aresta cujo valor seja mínimo e se atualiza dist[w].



Método de Dijkstra

Durante o cálculo dos caminhos, alguns atributos são definidos para os vértices:

- Uma vez selecionada uma aresta do corte, o caminho o vértice *w* deve passar pelo vértice *v*, então o predecessor do vértice *w* será o vértice *v*, ou ainda, pred[*w*] $\leftarrow$ *v*.

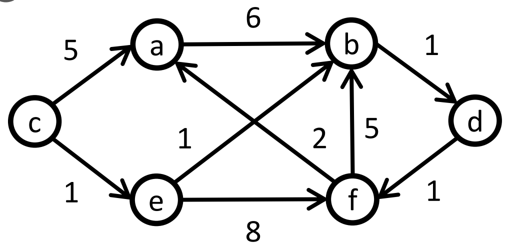


Método de Dijkstra – Algoritmo

1. <u>para</u> todo vértice $v \in V(G)$ <u>faça</u>
   a. $\text{dist}[v] \leftarrow \infty$; $\text{pred}[v] \leftarrow \varnothing$;   // Inicializar distâncias e predecessores
2. $S \leftarrow \{ s \}$;   // Inserir raiz no conj. de explorados
3. $\text{dist}[s] \leftarrow 0$;   // Fazer distância da raiz igual a zero
4. <u>para</u> $i = 1, \ldots, |V(G)| - 1$ <u>faça</u>
   a. Encontrar a aresta $(v, w) \in \text{corte}(S)$ tal que $\text{dist}[v] + d_{vw}$ seja mínimo
      // Isto é, aquela que representa a menor distância para um vértice não explorado
   b. $\text{dist}[w] \leftarrow \text{dist}[v] + d_{vw}$; $\text{pred}[w] \leftarrow v$;   // Atualizar a distância/predecessor de w
   c. $S \leftarrow S \cup \{ w \}$;   // Adicionar vértice w ao conj. explorados

Método de Dijkstra – Exemplo

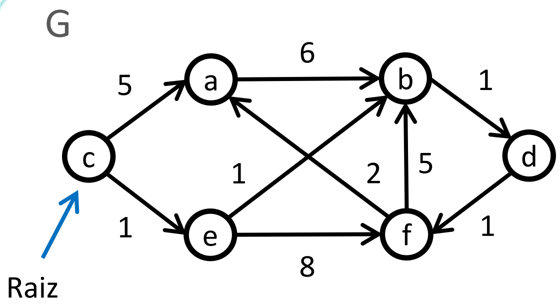
G



Vértice não explorado Vértice explorado

Aresta não explorada Aresta do corte Aresta do caminho

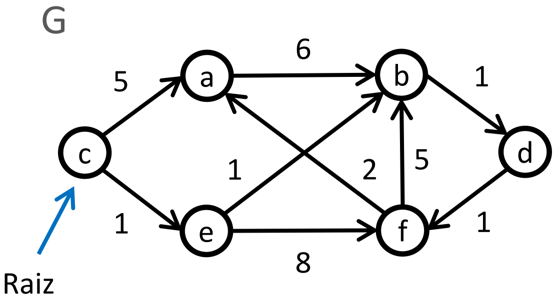
Método de Dijkstra – Exemplo



Vértice não explorado   Vértice explorado

Aresta não explorada   Aresta do corte   Aresta do caminho

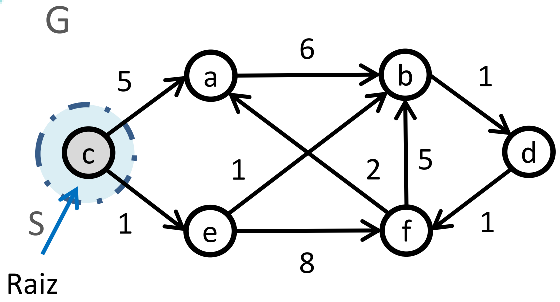
Método de Dijkstra – Exemplo



	a	b	c	d	e	f
Distância	∞	∞	∞	∞	∞	∞
Predecessor	$\varnothing$	$\varnothing$	$\varnothing$	$\varnothing$	$\varnothing$	$\varnothing$

Vértice não explorado
Vértice explorado
Aresta não explorada
Aresta do corte
Aresta do caminho

Método de Dijkstra – Exemplo



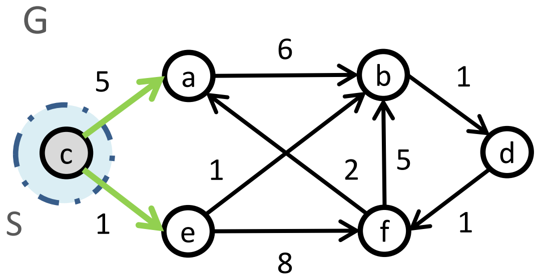
	a	b	c	d	e	f
Distância	∞	∞	0	∞	∞	∞
Predecessor	∅	∅	∅	∅	∅	∅

S = { c }

Vértice não explorado Vértice explorado

Aresta não explorada Aresta do corte Aresta do caminho

Método de Dijkstra – Exemplo



	a	b	c	d	e	f
Distância	∞	∞	0	∞	∞	∞
Predecessor	∅	∅	∅	∅	∅	∅

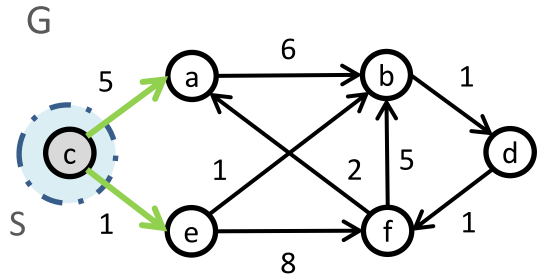
S = { c }

corte(S) = { (c,a), (c,e) }

○ Vértice não explorado ● Vértice explorado

— Aresta não explorada — Aresta do corte — Aresta do caminho

Método de Dijkstra – Exemplo



	a	b	c	d	e	f
Distância	∞	∞	0	∞	∞	∞
Predecessor	∅	∅	∅	∅	∅	∅

S = { c }

corte(S) = { (c,a), (c,e) }

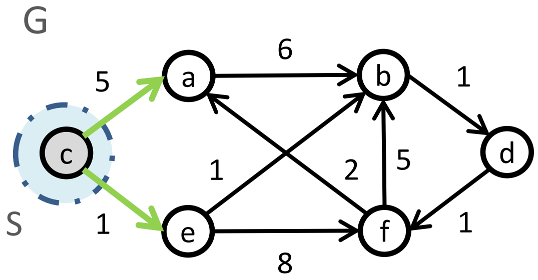
Estimativas de distância

Para aresta (c,a): dist-est[a] = dist[c] + d_{ca} = 0 + 5 = 5

Vértice não explorado   Vértice explorado

Aresta não explorada   Aresta do corte   Aresta do caminho

Método de Dijkstra – Exemplo



	a	b	c	d	e	f
Distância	∞	∞	0	∞	∞	∞
Predecessor	∅	∅	∅	∅	∅	∅

S = { c }

corte(S) = { (c,a), (c,e) }

Estimativas de distância

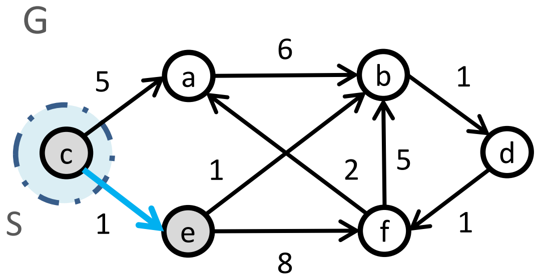
Para aresta (c,a): dist-est[a] = dist[c] + d_{ca} = 0 + 5 = 5

Para aresta (c,e): dist-est[e] = dist[c] + d_{ce} = 0 + 1 = 1

Vértice não explorado　Vértice explorado

Aresta não explorada　Aresta do corte　Aresta do caminho

Método de Dijkstra – Exemplo



	a	b	c	d	e	f
Distância	∞	∞	0	∞	1	∞
Predecessor	∅	∅	∅	∅	c	∅

S = { c }

corte(S) = { (c,a), (c,e) }

Estimativas de distância

Para aresta (c,a): dist-est[a] = dist[c] + d_{ca} = 0 + 5 = 5

Para aresta (c,e): dist-est[e] = dist[c] + d_{ce} = 0 + 1 = 1

Vértice não explorado

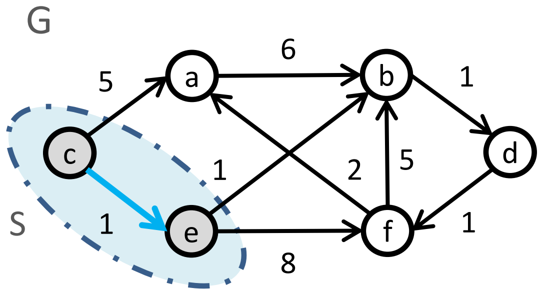
Vértice explorado

Aresta não explorada

Aresta do corte

Aresta do caminho

Método de Dijkstra – Exemplo



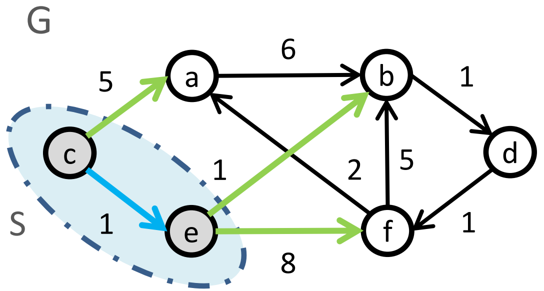
	a	b	c	d	e	f
Distância	∞	∞	0	∞	1	∞
Predecessor	∅	∅	∅	∅	c	∅

S = { c, e }

Vértice não explorado Vértice explorado

Aresta não explorada Aresta do corte Aresta do caminho

Método de Dijkstra – Exemplo



	a	b	c	d	e	f
Distância	∞	∞	0	∞	1	∞
Predecessor	∅	∅	∅	∅	c	∅

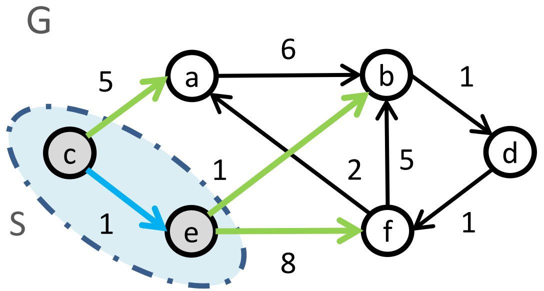
S = { c, e }

corte(S) = { (c,a), (e,b), (e,f) }

Vértice não explorado Vértice explorado

Aresta não explorada Aresta do corte Aresta do caminho

Método de Dijkstra – Exemplo



S = { c, e }

corte(S) = { (c,a), (e,b), (e,f) }

	a	b	c	d	e	f
Distância	∞	∞	0	∞	1	∞
Predecessor	∅	∅	∅	∅	c	∅

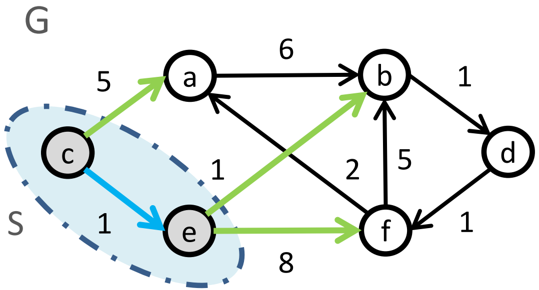
Estimativas de distância

Para aresta (c,a): dist-est[a] = dist[c] + d_{ca} = 0 + 5 = 5

Vértice não explorado   Vértice explorado

Aresta não explorada   Aresta do corte   Aresta do caminho

Método de Dijkstra – Exemplo



	a	b	c	d	e	f
Distância	∞	∞	0	∞	1	∞
Predecessor	∅	∅	∅	∅	c	∅

S = { c, e }

corte(S) = { (c,a), (e,b), (e,f) }

Estimativas de distância

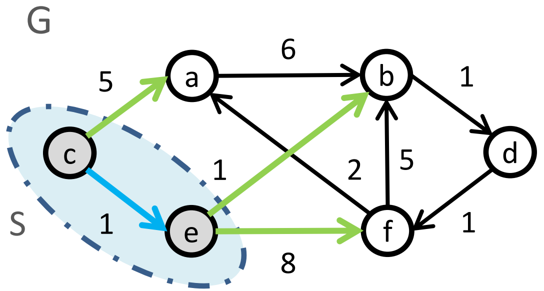
Para aresta (c,a): dist-est[a] = dist[c] + d_{ca} = 0 + 5 = 5

Para aresta (e,b): dist-est[b] = dist[e] + d_{eb} = 1 + 1 = 2

Vértice não explorado Vértice explorado

Aresta não explorada Aresta do corte Aresta do caminho

Método de Dijkstra – Exemplo



	a	b	c	d	e	f
Distância	∞	∞	0	∞	1	∞
Predecessor	∅	∅	∅	∅	c	∅

S = { c, e }

corte(S) = { (c,a), (e,b), (e,f) }

Estimativas de distância

Para aresta (c,a): dist-est[a] = dist[c] + d_{ca} = 0 + 5 = 5

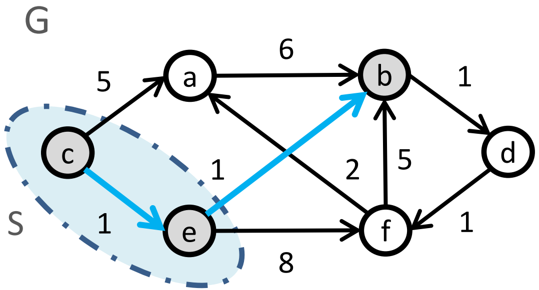
Para aresta (e,b): dist-est[b] = dist[e] + d_{eb} = 1 + 1 = 2

Para aresta (e,f): dist-est[f] = dist[e] + d_{ef} = 1 + 8 = 9

Vértice não explorado   Vértice explorado

Aresta não explorada   Aresta do corte   Aresta do caminho

Método de Dijkstra – Exemplo



	a	b	c	d	e	f
Distância	∞	2	0	∞	1	∞
Predecessor	∅	e	∅	∅	c	∅

S = { c, e }

corte(S) = { (c,a), (e,b), (e,f) }

Estimativas de distância

Para aresta (c,a): dist-est[a] = dist[c] + d_{ca} = 0 + 5 = 5

Para aresta (e,b): dist-est[b] = dist[e] + d_{eb} = 1 + 1 = 2

Para aresta (e,f): dist-est[f] = dist[e] + d_{ef} = 1 + 8 = 9

Vértice não explorado

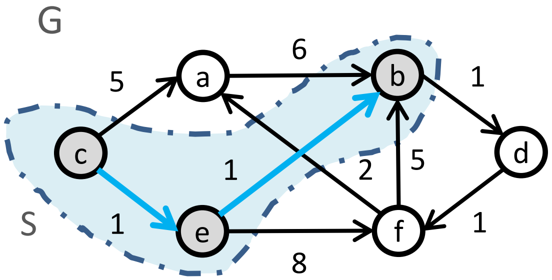
Vértice explorado

Aresta não explorada

Aresta do corte

Aresta do caminho

Método de Dijkstra – Exemplo



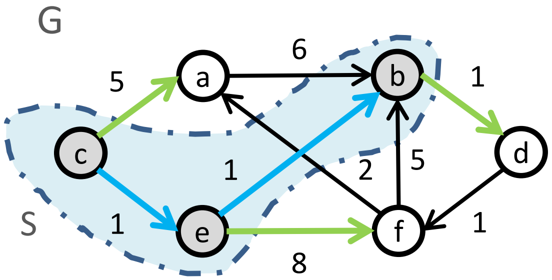
	a	b	c	d	e	f
Distância	∞	2	0	∞	1	∞
Predecessor	∅	e	∅	∅	c	∅

S = { c, e, b }

Vértice não explorado   Vértice explorado

Aresta não explorada   Aresta do corte   Aresta do caminho

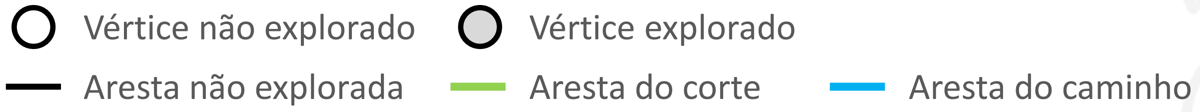
Método de Dijkstra – Exemplo



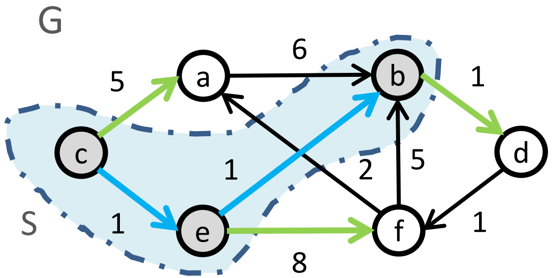
	a	b	c	d	e	f
Distância	∞	2	0	∞	1	∞
Predecessor	∅	e	∅	∅	c	∅

S = { c, e, b }

corte(S) = { (c,a), (e,f), (b,d) }



Método de Dijkstra – Exemplo



S = { c, e, b }

corte(S) = { (c,a), (e,f), (b,d) }

	a	b	c	d	e	f
Distância	∞	2	0	∞	1	∞
Predecessor	∅	e	∅	∅	c	∅

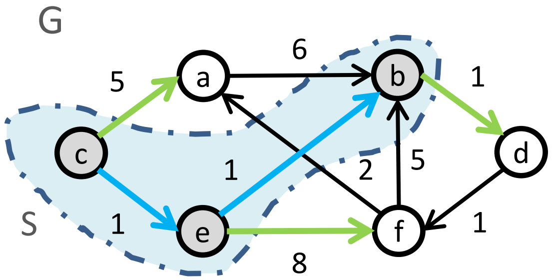
Estimativas de distância

Para aresta (c,a): dist-est[a] = dist[c] + d_{ca} = 0 + 5 = 5

Vértice não explorado   Vértice explorado

Aresta não explorada   Aresta do corte   Aresta do caminho

Método de Dijkstra – Exemplo



	a	b	c	d	e	f
Distância	∞	2	0	∞	1	∞
Predecessor	∅	e	∅	∅	c	∅

S = { c, e, b }

corte(S) = { (c,a), (e,f), (b,d) }

Estimativas de distância

Para aresta (c,a): dist-est[a] = dist[c] + d_{ca} = 0 + 5 = 5

Para aresta (e,f): dist-est[f] = dist[e] + d_{ef} = 1 + 8 = 9

Vértice não explorado

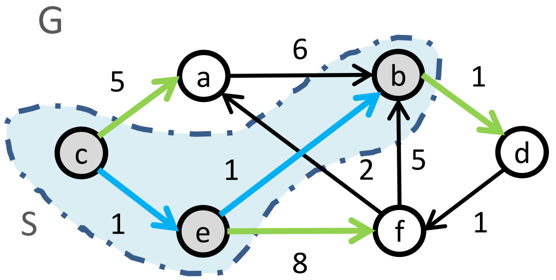
Vértice explorado

Aresta não explorada

Aresta do corte

Aresta do caminho

Método de Dijkstra – Exemplo



S = { c, e, b }

corte(S) = { (c,a), (e,f), (b,d) }

	a	b	c	d	e	f
Distância	∞	2	0	∞	1	∞
Predecessor	∅	e	∅	∅	c	∅

Estimativas de distância

Para aresta (c,a): dist-est[a] = dist[c] + d_{ca} = 0 + 5 = 5

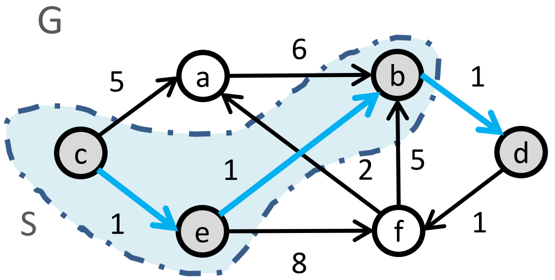
Para aresta (e,f): dist-est[f] = dist[e] + d_{ef} = 1 + 8 = 9

Para aresta (b,d): dist-est[d] = dist[b] + d_{bd} = 2 + 1 = 3

Vértice não explorado Vértice explorado

Aresta não explorada Aresta do corte Aresta do caminho

Método de Dijkstra – Exemplo



	a	b	c	d	e	f
Distância	∞	2	0	3	1	∞
Predecessor	$\varnothing$	e	$\varnothing$	b	c	$\varnothing$

S = { c, e, b }

corte(S) = { (c,a), (e,f), (b,d) }

Estimativas de distância

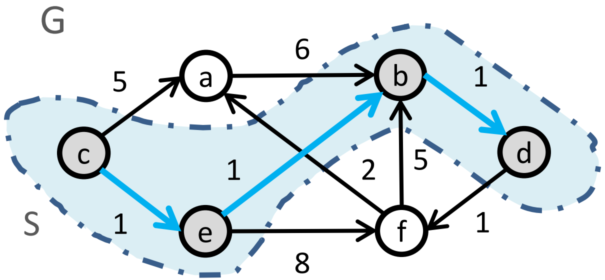
Para aresta (c,a): dist-est[a] = dist[c] + d_{ca} = 0 + 5 = 5

Para aresta (e,f): dist-est[f] = dist[e] + d_{ef} = 1 + 8 = 9

Para aresta (b,d): dist-est[d] = dist[b] + d_{bd} = 2 + 1 = 3

Vértice não explorado
Vértice explorado
Aresta não explorada
Aresta do corte
Aresta do caminho

Método de Dijkstra – Exemplo



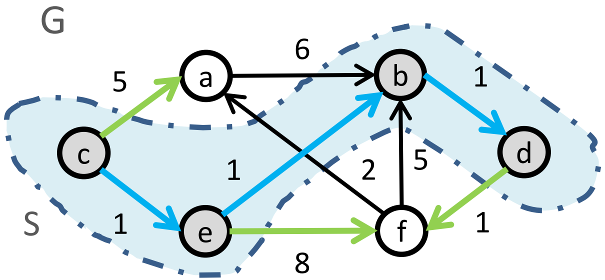
	a	b	c	d	e	f
Distância	∞	2	0	3	1	∞
Predecessor	∅	e	∅	b	c	∅

S = { c, e, b, d }

Vértice não explorado Vértice explorado

Aresta não explorada Aresta do corte Aresta do caminho

Método de Dijkstra – Exemplo



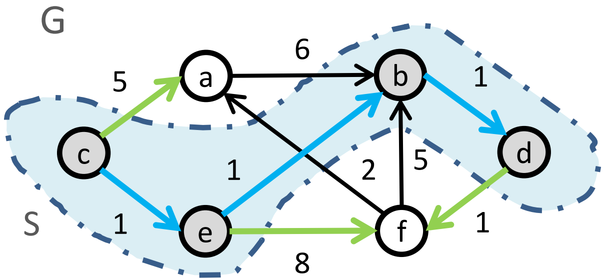
	a	b	c	d	e	f
Distância	∞	2	0	3	1	∞
Predecessor	∅	e	∅	b	c	∅

S = { c, e, b, d }

corte(S) = { (c,a), (e,f), (d,f) }

Vértice não explorado
Vértice explorado
Aresta não explorada
Aresta do corte
Aresta do caminho

Método de Dijkstra – Exemplo



	a	b	c	d	e	f
Distância	∞	2	0	3	1	∞
Predecessor	∅	e	∅	b	c	∅

S = { c, e, b, d }

corte(S) = { (c,a), (e,f), (d,f) }

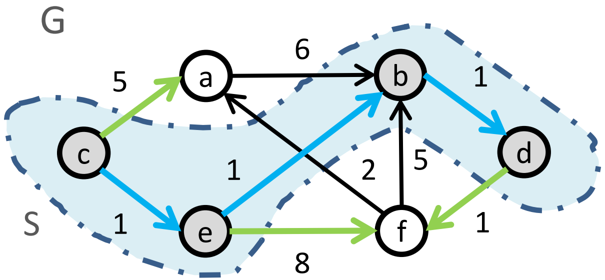
Estimativas de distância

Para aresta (c,a): dist-est[a] = dist[c] + d_{ca} = 0 + 5 = 5

Vértice não explorado Vértice explorado

Aresta não explorada Aresta do corte Aresta do caminho

Método de Dijkstra – Exemplo



	a	b	c	d	e	f
Distância	∞	2	0	3	1	∞
Predecessor	∅	e	∅	b	c	∅

S = { c, e, b, d }

corte(S) = { (c,a), (e,f), (d,f) }

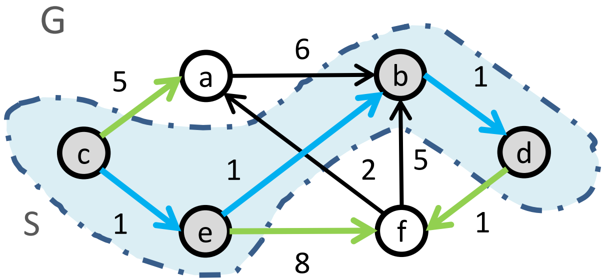
Estimativas de distância

Para aresta (c,a): dist-est[a] = dist[c] + d_{ca} = 0 + 5 = 5

Para aresta (e,f): dist-est[f] = dist[e] + d_{ef} = 1 + 8 = 9

Vértice não explorado
Vértice explorado
Aresta não explorada
Aresta do corte
Aresta do caminho

Método de Dijkstra – Exemplo



	a	b	c	d	e	f
Distância	∞	2	0	3	1	∞
Predecessor	∅	e	∅	b	c	∅

S = { c, e, b, d }

corte(S) = { (c,a), (e,f), (d,f) }

Estimativas de distância

Para aresta (c,a): dist-est[a] = dist[c] + d_{ca} = 0 + 5 = 5

Para aresta (e,f): dist-est[f] = dist[e] + d_{ef} = 1 + 8 = 9

Para aresta (d,f): dist-est[f] = dist[d] + d_{df} = 3 + 1 = 4

Vértice não explorado

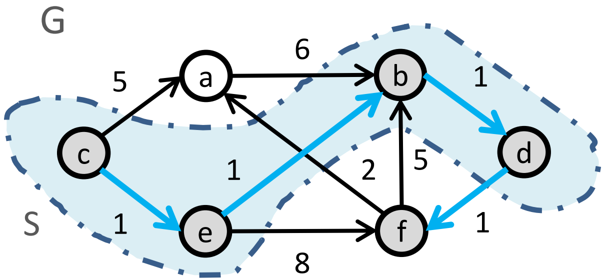
Vértice explorado

Aresta não explorada

Aresta do corte

Aresta do caminho

Método de Dijkstra – Exemplo



	a	b	c	d	e	f
Distância	∞	2	0	3	1	4
Predecessor	∅	e	∅	b	c	d

S = { c, e, b, d }

corte(S) = { (c,a), (e,f), (d,f) }

Estimativas de distância

Para aresta (c,a): dist-est[a] = dist[c] + d_{ca} = 0 + 5 = 5

Para aresta (e,f): dist-est[f] = dist[e] + d_{ef} = 1 + 8 = 9

Para aresta (d,f): dist-est[f] = dist[d] + d_{df} = 3 + 1 = 4

Vértice não explorado

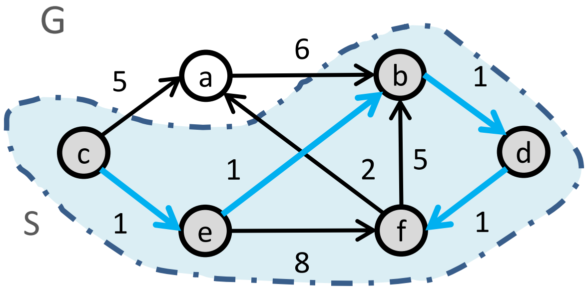
Vértice explorado

Aresta não explorada

Aresta do corte

Aresta do caminho

Método de Dijkstra – Exemplo



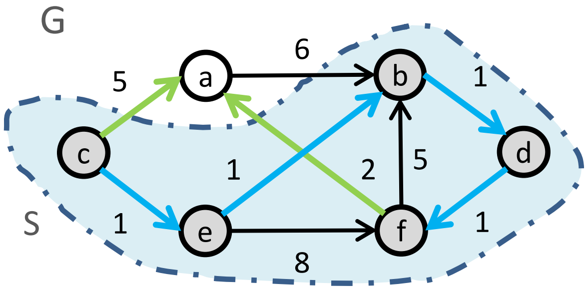
	a	b	c	d	e	f
Distância	∞	2	0	3	1	4
Predecessor	∅	e	∅	b	c	d

S = { c, e, b, d, f }

Vértice não explorado   Vértice explorado

Aresta não explorada   Aresta do corte   Aresta do caminho

Método de Dijkstra – Exemplo



	a	b	c	d	e	f
Distância	∞	2	0	3	1	4
Predecessor	∅	e	∅	b	c	d

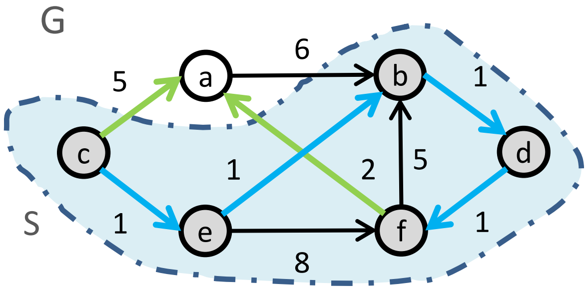
S = { c, e, b, d, f }

corte(S) = { (c,a), (f,a) }

Vértice não explorado Vértice explorado

Aresta não explorada Aresta do corte Aresta do caminho

Método de Dijkstra – Exemplo



	a	b	c	d	e	f
Distância	∞	2	0	3	1	4
Predecessor	$\varnothing$	e	$\varnothing$	b	c	d

S = { c, e, b, d, f }

corte(S) = { (c,a), (f,a) }

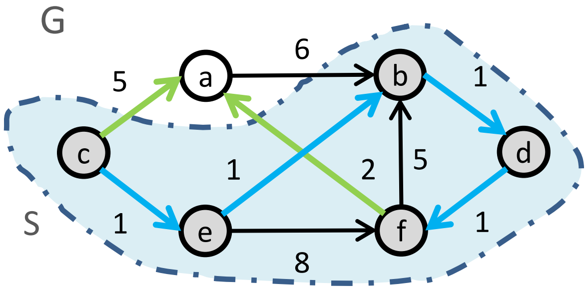
Estimativas de distância

Para aresta (c,a): dist-est[a] = dist[c] + d_{ca} = 0 + 5 = 5

○ Vértice não explorado ● Vértice explorado

— Aresta não explorada — Aresta do corte — Aresta do caminho

Método de Dijkstra – Exemplo



	a	b	c	d	e	f
Distância	∞	2	0	3	1	4
Predecessor	∅	e	∅	b	c	d

S = { c, e, b, d, f }

corte(S) = { (c,a), (f,a) }

Estimativas de distância

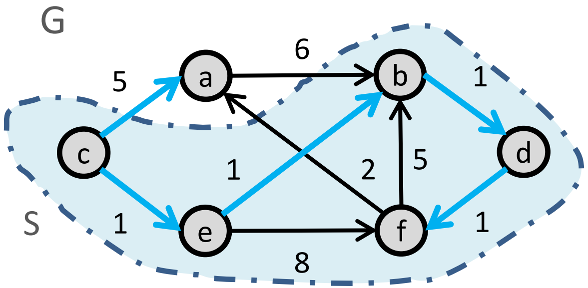
Para aresta (c,a): dist-est[a] = dist[c] + d_{ca} = 0 + 5 = 5

Para aresta (f,a): dist-est[a] = dist[f] + d_{fa} = 4 + 2 = 6

Vértice não explorado | Vértice explorado

Aresta não explorada | Aresta do corte | Aresta do caminho

Método de Dijkstra – Exemplo



	a	b	c	d	e	f
Distância	5	2	0	3	1	4
Predecessor	c	e	∅	b	c	d

S = { c, e, b, d, f }

corte(S) = { (c,a), (f,a) }

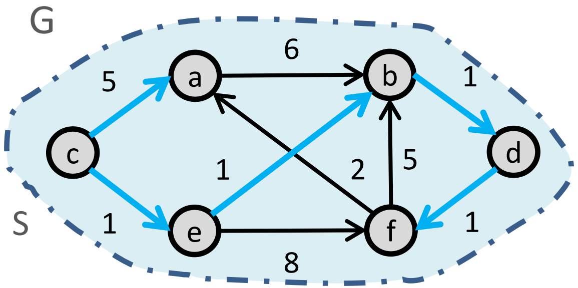
Estimativas de distância

Para aresta (c,a): dist-est[a] = dist[c] + d_{ca} = 0 + 5 = 5

Para aresta (f,a): dist-est[a] = dist[f] + d_{fa} = 4 + 2 = 6

Vértice não explorado
Vértice explorado
Aresta não explorada
Aresta do corte
Aresta do caminho

Método de Dijkstra – Exemplo



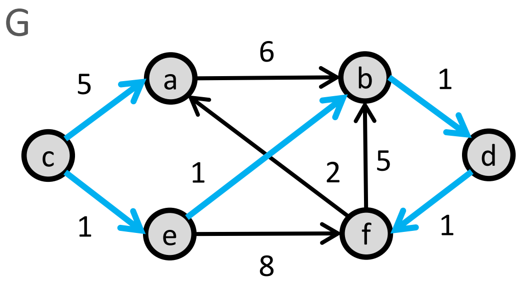
	a	b	c	d	e	f
Distância	5	2	0	3	1	4
Predecessor	c	e	∅	b	c	d

S = { c, e, b, d, f, a }

Vértice não explorado   Vértice explorado

Aresta não explorada   Aresta do corte   Aresta do caminho

Método de Dijkstra – Exemplo



	a	b	c	d	e	f
Distância	5	2	0	3	1	4
Predecessor	c	e	∅	b	c	d

Vértice não explorado   Vértice explorado

Aresta não explorada   Aresta do corte   Aresta do caminho

Potencial

Um potencial para um grafo G é uma numeração dos vértices de G, ou seja, um vetor que associa um número a cada vértice de G.

Em relação a um potencial h[ ], dizemos que um arco (v, w) está **tenso** se $h[w] - h[v] > c$, está **relaxado** se $h[w] - h[v] \leq c$ e está **justo** se $h[w] - h[v] \equiv c$, sendo c o custo do arco.

Em outras palavras, o arco (v, w) está

- <u>tenso</u> se $h[v] + c < h[w]$,
- <u>relaxado</u> se $h[v] + c \geq h[w]$ e
- <u>justo</u> se $h[v] + c \equiv h[w]$.

Potencial × Condições de Minimalidade

Um potencial é relaxado (ou viável) se todos os arcos de G estão relaxados em relação a ele.

Qualquer potencial relaxado h[ ] dá uma cota inferior para as distâncias entre vértices: se P é um caminho de um vértice x a um vértice y então

$$\text{custo}(P) \geq h[y] - h[x],$$

sendo custo(P) o custo do caminho P.

Condição suficiente de minimalidade: Para qualquer caminho P de um vértice x a um vértice y, se existe um potencial relaxado h[ ] tal que h[y] – h[x] = custo(P) então P é mínimo (e portanto a diferença h[y] – h[x] é a distância de x a y).

Potencial × Condições de Minimalidade

Suponha que P seja um caminho 0-1-2-3 então custo(P) = $d_{23} + d_{12} + d_{01}$. Para um potencial relaxado h[ ],

$$custo(P) = d_{23} + d_{12} + d_{01} \geq (h[3] - h[2]) + (h[2] - h[1]) + (h[1] - h[0])$$

$$custo(P) \geq h[3] - h[0]$$

Em particular, se P for um <u>ciclo</u> então custo(P) ≥ 0.

A recíproca da condição anterior é verdadeira se for restrita aos caminhos mínimos que têm uma origem comum (desde que todos os vértices sejam alcançáveis a partir da raiz s), isto é: **Para qualquer vértice s, o vetor das distâncias a partir de s é um potencial relaxado**.

Potencial × Método de Dijkstra

Um vértice do grafo é considerado maduro, se seu conjunto de sucessores já foi examinado, caso contrário, ele e considerado imaturo (todo vértice maduro pertence à árvore de caminhos mínimos).

O método de Dijkstra utiliza um vetor de predecessores pred[ ] que representa uma árvore com raiz s, um potencial dist[ ] associado aos vértices do grafo, e um conjunto de vértices maduros.

No início do processo, todos os vértices são imaturos, dist[s] = 0 para a raiz s, dist[v] = ∞ para todo v diferente de s e os predecessores estão indefinidos.

Potencial × Método de Dijkstra

O processo iterativo consiste no seguinte:

<u>enquanto</u> existir vértice imaturo <u>faça</u>

1. Selecionar o vértice imaturo v que minimiza dist[ ]
2. <u>para</u> cada arco (v, w) que está tenso <u>faça</u>
   i. $\text{dist}[w] \leftarrow \text{dist}[v] + d_{vw}$ // Relaxação da aresta
   ii. $\text{pred}[w] \leftarrow v$
3. Declarar v como maduro