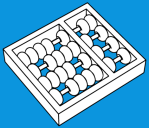
Caminhos mínimos

MC558 - Projeto e Análise de Algoritmos II

11

09/24



Santiago Valdés Ravelo
https://ic.unicamp.br/~santiago/
ravelo@unicamp.br

"Às vezes, a coisa mais produtiva que você pode fazer é relaxar."

Mark Black

Caminhos mínimos com uma origem

Problema do Caminho Mínimo

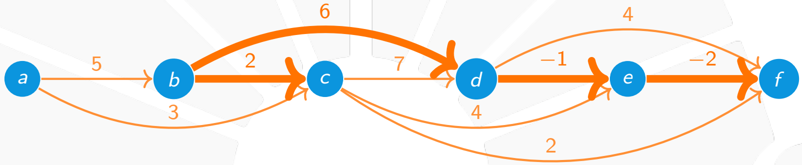
Considere um par (G, w) em que:

- G é um grafo direcionado.
- w associa um peso $w(u, v)$ para cada aresta $(\mathbf{u}, \mathbf{v})$.

Estamos interessados nos seguintes problemas:

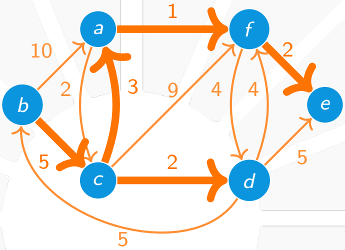
1. **Problema do caminho mínimo entre dois vértices:**
   Dados **s** e **t**, encontrar um caminho de peso mínimo de **s** a **t**.
2. **Problema dos caminhos mínimos com mesma origem:**
   Dado **s**, encontrar um caminho de peso mínimo de **s** a **v** para **TODO** vértice **v** de G.

Exemplo: grafo direcionado acíclico



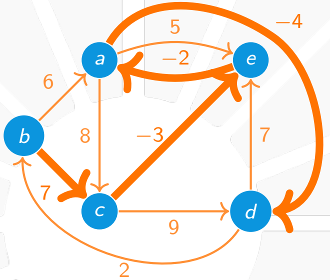
v	**a**	**b**	**c**	**d**	**e**	**f**
dist(**b**, v)	∞	0	2	6	5	3

Exemplo: grafo direcionado sem arestas negativas



v	**a**	**b**	**c**	**d**	**e**	**f**
dist(**b**, v)	8	0	5	7	11	9

Exemplo: grafo direcionado com arestas negativas



v	**a**	**b**	**c**	**d**	**e**
dist(**b**, v)	2	0	7	−2	4

Representando caminhos mínimos

A saída é similar à da busca em largura a partir de **s**:

- Para cada **v** $\in$ **V[G]**, associamos um **PREDECESSOR** π[**v**].
- O vetor π induz uma **ÁRVORE DE CAMINHOS MÍNIMOS** com raiz em **s**.
- Um caminho de **s** a **v** na árvore é um caminho mínimo de **s** a **v** no grafo G.

Definição

Problema (Caminhos mínimos com mesma origem)

Entrada: *Um grafo direcionado* $G = (\boldsymbol{V}, \boldsymbol{E})$, *com peso* w *nas arestas, sem ciclos de peso negativo e um vértice de origem* $\boldsymbol{s}$.

Saída: *Um vetor* d *com* $d[\boldsymbol{v}] = \mathrm{dist}(\boldsymbol{s},\boldsymbol{v})$ *para* $\boldsymbol{v} \in \boldsymbol{V}$ *e um vetor* π *definindo uma* ***ÁRVORE DE CAMINHOS MÍNIMOS***.

Subestrutura ótima de caminhos mínimos

Teorema

Seja (G, w) um grafo direcionado ***SEM CICLOS NEGATIVOS*** *e seja*

$$P = (\mathbf{v_1}, \mathbf{v_2}, \dots, \mathbf{v_k})$$

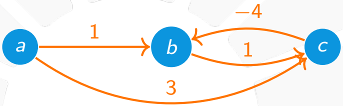
um caminho mínimo de $\mathbf{v_1}$ a $\mathbf{v_k}$. Então para quaisquer índices i, j com $1 \leq i \leq j \leq k$, o subcaminho

$$P_{ij} = (\mathbf{v_i}, \mathbf{v_{i+1}}, \dots, \mathbf{v_j})$$

é um caminho mínimo de $\mathbf{v_i}$ a $\mathbf{v_j}$.

Subestrutura ótima de caminhos mínimos

A subestrutura ótima **NÃO** vale se o grafo tiver **CICLOS NEGATIVOS**:



- (**a**, **b**, **c**) é um caminho mínimo de **a** a **c** com peso $1 + 1 = 2$.
- Mas, (**a,b**) **NÃO** é um caminho mínimo de **a** a **b**.
- (**a,c,b**) é um caminho mínimo de **a** a **b** com peso $3 - 4 = -1$.

Algoritmos baseados em relaxação

Veremos algoritmos com as seguintes características:

1. Inicializam *d* e π com uma sub-rotina INITIALIZE-SINGLE-SOURCE.
2. Alteram *d* e π apenas com uma sub-rotina RELAX.

Esses algoritmos mantêm algumas invariantes:

- Existe um caminho de **s** a **v** com peso *d*[**v**].
- Esse caminho pode ser recuperado por meio π.
- Assim, *d*[**v**] é sempre **MAIOR OU IGUAL** a dist(**s,v**).
- Queremos que no final valha *d*[**v**] = dist(**s,v**).

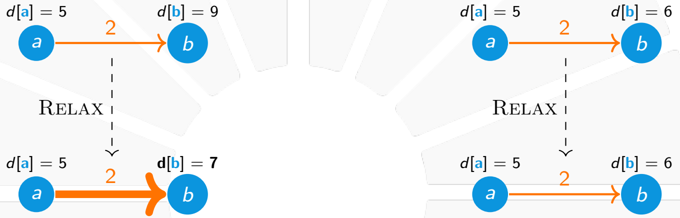
Inicialização

Algoritmo: $\textsc{Initialize-Single-Source}(G, s)$

```
para cada v ∈ V[G]
    d[v] ← ∞
    π[v] ← NIL
d[s] ← 0
```

Relaxação

Tenta melhorar a estimativa $d[v]$ examinando **(u,v)**.



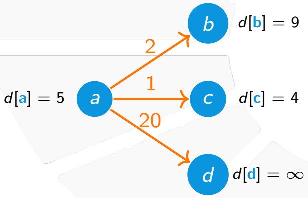
Algoritmo: $\textsc{Relax}(u, v, w)$

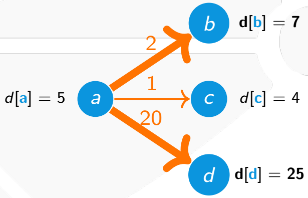
---

se $d[v] > d[u] + w(u, v)$
  $d[v] \leftarrow d[u] + w(u, v)$
  $\pi[v] \leftarrow u$

Relaxação dos vizinhos

Em cada iteração o algoritmo seleciona um vértice **u** e para cada vizinho **v** de **u** aplica $\textsc{Relax}(\mathbf{u},\mathbf{v}, w)$.





Casos do problema de caminhos mínimos

Veremos três algoritmos baseados em relaxação para tipos de subcasos diferentes do Problema de caminhos mínimos:

1. **APLICAÇÃO DE ORDENAÇÃO TOPOLÓGICA**: G é acíclico.
2. **ALGORITMO DE DIJKSTRA**: (G, w) não tem arestas de peso negativo.
3. **ALGORITMO DE BELLMAN-FORD**: (G, w) pode ter arestas de peso negativo, mas não contém ciclos negativos.

Caminhos mínimos em grafos acíclicos

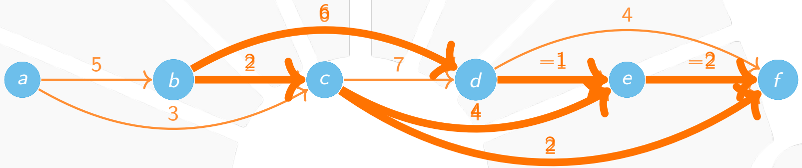
Caminhos mínimos em grafos acíclicos

Problema

Entrada: *Um grafo direcionado acíclico* $G = (\mathbf{V}, \mathbf{E})$, *uma função de peso* w *nas arestas e um vértice origem* s.

Saída: *Um vetor* d *com* $d[\mathbf{v}] = \operatorname{dist}(\mathbf{s},\mathbf{v})$ *para* $\mathbf{v} \in \mathbf{V}$ *e um vetor* π *definindo uma* ***ÁRVORE DE CAMINHOS MÍNIMOS***.

Exemplo



	a	b	c	d	e	f
d	∞	0	∞ 2	∞ 6	∞ 6 5	∞ 4 3

Algoritmo

Algoritmo: DAG-SHORTEST-PATHS(G, w, s)

1 ordene topologicamente os vértices de G
2 INITIALIZE-SINGLE-SOURCE(G, s)
3 **para cada** *u na ordem topológica*
4   **para cada** $v \in Adj[u]$
5   RELAX(u, v, w)
6 **devolva** d, π

Linha(s)	Tempo total
1	$O(V + E)$
2	$O(V)$
3-5	$O(V + E)$

Complexidade de DAG-SHORTEST-PATHS: $O(V + E)$.

Correção

- Há diversas estratégias para demonstrar que Dag-Shortest-Paths esteja correto.
- Vamos demonstrar algumas propriedades que valem porque o algoritmo é baseado em relaxação.
- Essas propriedades também serão úteis para analisar os outros algoritmos depois.

Propriedade de algoritmos baseados em relaxação

Ao longo de um algoritmo baseado em relaxação sempre vale:

- **Limite superior:** Vale $d[\mathbf{v}] \geq \text{dist}(\mathbf{s},\mathbf{v})$ e, tão logo $d[\mathbf{v}]$ alcança $\text{dist}(\mathbf{s},\mathbf{v})$, nunca mais muda.
- **Inexistência de caminho:** Se não existe caminho de **s** a **v**, então $d[\mathbf{v}] = \infty$.
- **Subgrafo de predecessores:** Se $d[\mathbf{v}] < \infty$, então o subgrafo dos predecessores induzido por π é um caminho de peso $d[\mathbf{v}]$.
- **Convergência:** Se p é um caminho mínimo de **s** até **v** terminando com a aresta **(u,v)** e $d[\mathbf{u}] = \text{dist}(\mathbf{s},\mathbf{u})$, então ao relaxar **(u,v)**, $d[\mathbf{v}] = \text{dist}(\mathbf{s},\mathbf{v})$, que nunca mais muda.
- **Relaxamento de caminho:** Se $p = (\mathbf{v}_0, \mathbf{v}_1, \ldots, \mathbf{v}_k)$ é um caminho mínimo de $\mathbf{s} = \mathbf{v}_0$ a $\mathbf{v}_k$ e relaxamos as arestas de p na ordem $(\mathbf{v}_0, \mathbf{v}_1),\ (\mathbf{v}_1, \mathbf{v}_2),\ \ldots,\ (\mathbf{v}_{k-1}, \mathbf{v}_k)$, então $d[\mathbf{v}_k] = \text{dist}(\mathbf{s}, \mathbf{v}_k)$.

  A propriedade vale mesmo se tivermos realizado quaisquer outras relaxações durante a execução.

Correção de DAG-SHORTEST-PATHS

- Seja $\mathbf{v}$ um vértice e suponha que $p = (\mathbf{v_0}, \mathbf{v_1}, \ldots, \mathbf{v_k})$ é um caminho mínimo de $\mathbf{s} = \mathbf{v_0}$ a $\mathbf{v} = \mathbf{v_k}$.
- Como $\mathbf{v_0}$, $\mathbf{v_1}$, $\ldots$, $\mathbf{v_k}$ aparecem em ordem na ordenação topológica, as arestas $(\mathbf{v_0}, \mathbf{v_1})$, $\ldots$, $(\mathbf{v_{k-1}}, \mathbf{v_k})$ são relaxadas em ordem.
- Logo, pela propriedade do **Relaxamento de caminho**, o algoritmo computa corretamente $d[\mathbf{v}] = \mathrm{dist}(\mathbf{s},\mathbf{v})$ para cada $\mathbf{v} \in \mathbf{V}$.

Também é fácil ver que π define uma árvore de caminhos mínimos. Por quê?

Perguntas

Vamos fazer alguns exercícios?

Exercício 1

Como se resolve o problema de encontrar um caminho de peso **MÁXIMO** de **s** a **t** em um grafo direcionado acíclico (G, w)?

Exercício 2

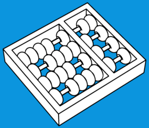
Como se resolve o problema do caminho mínimo de **s** a **t** em **TEMPO LINEAR** para um grafo direcionado em que todas as arestas têm o mesmo peso $C > 0$?

Caminhos mínimos

MC558 - Projeto e Análise de Algoritmos II

11

09/24



Santiago Valdés Ravelo
https://ic.unicamp.br/~santiago/
ravelo@unicamp.br

UNICAMP