Teoria dos Grafos

Valeriano A. de Oliveira, Socorro Rangel, Silvio A. de Araujo

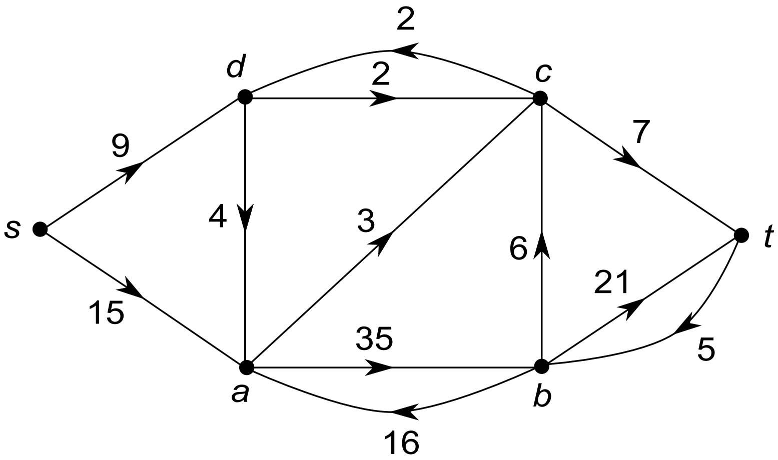
Departamento de Matemática Aplicada

Capítulo 10: Caminho mínimo - Algoritmo de Dijskstra

Preparado a partir do texto:
Rangel, Socorro. Teoria do Grafos, Notas de aula, IBILCE, Unesp, 2002-2013.

Introdução

Considere a rede:



Introdução

Dado dois vértices nesta rede, queremos determinar o menor caminho entre eles.

Uma primeira questão é como representar os valores associados às arestas neste grafo. Isto pode ser feito através da *matriz de pesos*.

Definição

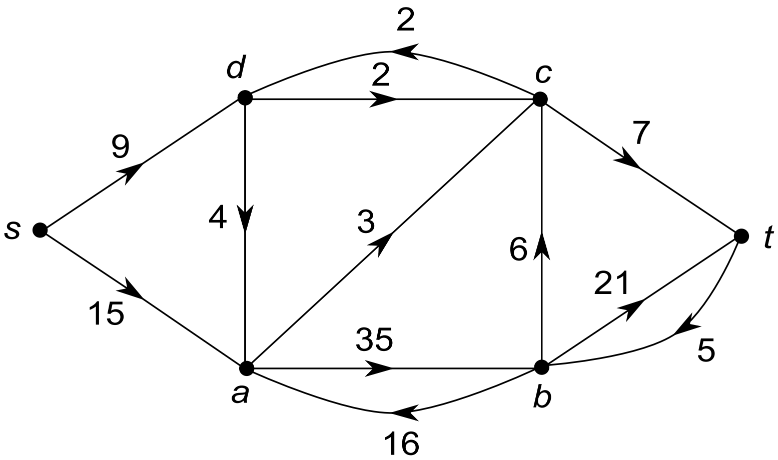
Seja D um digrafo simples cujas arestas possuem "pesos" associados, digamos, a cada aresta (v_i, v_j) *está associado um número real* $w_{ij} \geq 0$ *(que pode representar comprimento, distância, valor, etc).*

Vamos definir $w_{ii} = 0$ *para todo* i *e* $w_{ij} = \infty$ *quando não existe uma aresta entre os vértices* v_i *e* v_j*.*

Assim a ***matriz de pesos*** *é uma matriz* $n \times n$ *definida como* $W = [w_{ij}]$*, onde* n *é o número de vértices.*

Exemplo

Construir a matriz de pesos para a rede abaixo:



Observação

Em geral

- $w_{ij} \neq w_{ji}$ *e*
- $w_{ij} + w_{jk}$ *pode ser menor ou igual a* w_{ik}.

Por exemplo, na rede anterior,

$$w_{sd} + w_{da} = 9 + 4 \leq 15 = w_{sa}$$

e

$$w_{ab} = 35 \neq 16 = w_{ba}.$$

Algoritmos de caminho mais curto

Trabalho	**Complexidade**
Dijkstra (1959)[1]	$O(n^2)$
Ford (1956), Moore (1957), Bellman (1958)	$O(nm)$
Ford & Fulkerson (1962)	$O(nm)$
Floyd (1962)	$O(n^3)$
Hu (1968)	$O(nm)$
Dial (1969)	$O(m + Cn)$[2]
Pape (1974)	$\Theta(n2^n)$

[1] O algoritmo de Dijkstra é um dos mais eficientes até hoje.

[2] A constante C é um limite superior para o peso das arestas.

Algoritmos de caminho mais curto

Pallottino (1984)	$(n^2 m)$
Glover et al. (1984 e 1985)	$O(mn)$
Kelin & Reif	$O(n^{2/3} \log^{7/3} n(\log n + \log D))^3$
Goldberg & Radzik (1993)	$O(nm)$
Cohen (1996)	$O(\log^4 n)$
Träff & Zaroliagis (1996)	$O((n^{2\varepsilon} + n^{1-\varepsilon}) \log n)^4$
Henzinger et al. (1997)	$O(n^{3/4} L \log n)^5$
Pettie & Ramachandran (2005)	$O(m + n \log \log n)$

Para mais detalhes, ver

M. Goldbarg & E. Goldbarg, Grafos: conceitos, algoritmos e aplicações, Elsevier Campus, 2012.

[3] D é a soma dos pesos das arestas.

[4] $0 < \varepsilon < 1/2$.

[5] L é o valor absoluto do maior comprimento negativo de uma aresta.

Ideia do Algoritmo de Dijkstra

Vamos supor que queremos encontrar o caminho mínimo entre os nós s e t em uma rede dada. A ideia consiste em:

- Rotular os vértices do digrafo.
- A partir do vértice inicial s, proceder em direção ao vértice final t (seguindo as arestas orientadas) rotulando os vértices com as suas distâncias ao vértice s, medidas até aquele momento.
- A cada estágio do algoritmo teremos vértices que possuem rótulos temporários e vértice com rótulos permanentes.
- O rótulo de um vértice v_j é feito permanente quando este rótulo representa a menor distância de s até v_j.
- Começamos associando rótulo permanente igual a zero para o vértice s, e um rótulo temporário igual a ∞ para os outros $n-1$ vértices do grafo.

Ideia do Algoritmo de Dijkstra

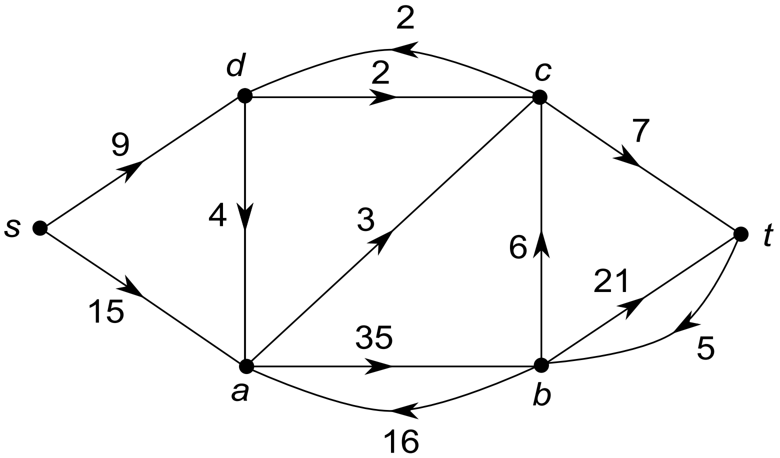
- A cada iteração, um novo vértice recebe um rótulo permanente de acordo com as seguintes regras:
  1. Cada vértice v_j com um rótulo temporário, recebe um novo rótulo temporário dado por:

     $$\min\{\text{rótulo de } v_j, (\text{rótulo de } v_i) + w_{ij}\},$$

     onde v_i é o vértice que recebeu rótulo permanente na iteração anterior e w_{ij} é o valor da aresta entre o vértice v_i e o vértice v_j.
  2. Encontre o menor valor entre os rótulos temporários. Este será o rótulo permanente do respectivo vértice. Em caso de empate selecione qualquer um dos candidatos e atribua rótulo permanente ao escolhido.
- Repetir 1 e 2 até que o vértice destino, t, receba um rótulo permanente.

Exemplo

Vamos aplicar a ideia acima à rede abaixo:



Aplicação do algoritmo: primeira iteração

Inicialização: $rot(s) = 0$ é permanente, $rot(a) = rot(b) = rot(c) = rot(d) = rot(t) = \infty$ são temporários

$it = 1$

Regra 1 – Novo rótulo

Novo $rot(a) = \min\{\infty, 0 + w_{sa}\} = \min\{\infty, 15\} = 15$

Novo $rot(b) = \min\{\infty, 0 + w_{sb}\} = \min\{\infty, \infty\} = \infty$

Novo $rot(c) = \min\{\infty, 0 + w_{sc}\} = \min\{\infty, \infty\} = \infty$

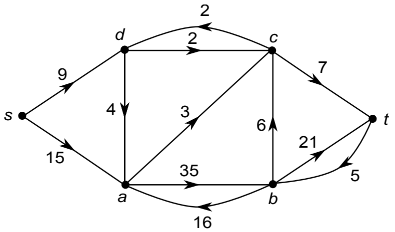
Novo $rot(d) = \min\{\infty, 0 + w_{sd}\} = \min\{\infty, 9\} = 9$

Novo $rot(t) = \min\{\infty, 0 + w_{st}\} = \min\{\infty, \infty\} = \infty$

Regra 2 – Rótulo permanente:

$\min\{15, \infty, \infty, 9, \infty\} = 9 \Rightarrow rot(d)$ torna-se permanente

Aplicação do algoritmo: segunda iteração



$it = 2$

Regra 1 – Novo rótulo

Novo $rot(a) = \min\{15, 9 + w_{da}\} = \min\{15, 9 + 4\} = 13$

Novo $rot(b) = \min\{\infty, 9 + w_{db}\} = \min\{\infty, \infty\} = \infty$

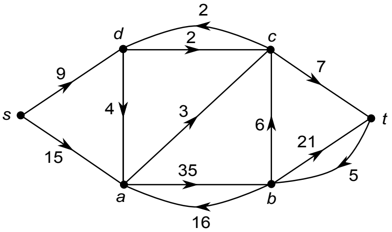
Novo $rot(c) = \min\{\infty, 9 + w_{dc}\} = \min\{\infty, 9 + 2\} = 11$

Novo $rot(t) = \min\{\infty, 9 + w_{dt}\} = \min\{\infty, \infty\} = \infty$

Regra 2 – Rótulo permanente

$\min\{13, \infty, 11, \infty\} = 11 \Rightarrow rot(c)$ torna-se permanente

Aplicação do algoritmo: terceira iteração



$it = 3$

Regra 1 – Novo rótulo

Novo $rot(a) = \min\{13, 11 + w_{ca}\} = \min\{13, 11 + \infty\} = 13$

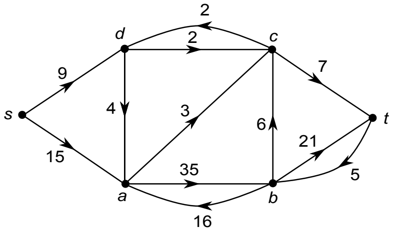
Novo $rot(b) = \min\{\infty, 11 + w_{cb}\} = \min\{\infty, 11 + \infty\} = \infty$

Novo $rot(t) = \min\{\infty, 11 + w_{ct}\} = \min\{\infty, 11 + 7\} = 18$

Regra 2 – Rótulo permanente

$\min\{13, \infty, 18\} = 13 \Rightarrow rot(a)$ torna-se permanente

Aplicação do algoritmo: quarta iteração



$it = 4$

Regra 1 – Novo rótulo

Novo $rot(b) = \min\{\infty, 13 + w_{ab}\} = \min\{\infty, 13 + 35\} = 48$

Novo $rot(t) = \min\{18, 13 + w_{at}\} = \min\{18, 13 + \infty\} = 18$

Regra 2 – Rótulo permanente

$\min\{48, 18\} = 18 \Rightarrow rot(t)$ torna-se permanente

Fim

Podemos parar pois o vértice destino, t, recebeu rótulo permanente. Assim o comprimento do menor caminho entre s e t é igual a 18.

Como recuperar o caminho?

A partir do vértice destino t, verificamos o vértice com rótulo permanente usado na obtenção do rótulo de t.

No nosso exemplo, o vértice c.

Repetimos o processo a partir de c até alcançar o vértice inicial s.

Assim temos t, c, d, s.

O caminho é obtido invertendo a ordem obtida: s, d, c, t.

Implementação

Em uma possível implementação deste algoritmo vamos precisar armazenar as seguintes informações:

- Indicação se um vértice v_k possui rótulo permanente ou temporário.
- Guardar a menor distância entre o vértice inicial s e o vértice v_k.
- Guardar o vértice com rótulo permanente que deu origem a um novo rótulo (importante para recuperar o caminho).

Assim vamos podemos definir a seguinte estrutura de dados:

1. Entrada de dados: matriz de pesos W $(O(n^2))$.
2. A dificuldade é distinguir, a cada iteração, os vértices com rótulos permanentes e os vértices com rótulo temporário. Utilizamos um vetor lógico (ou binário) n-dimensional:

$$\text{final}(v) = \begin{cases} \texttt{True} & \text{se o rótulo do vértice } v \text{ é permanente,} \\ \texttt{False} & \text{se o rótulo do vértice } v \text{ é temporário.} \end{cases}$$

Implementação

3. Precisamos também de um vetor n-dimensional para guardar as distâncias acumuladas do vértice inicial s aos outros vértices v_i. Vamos chamar este vetor de dist.
4. Como recuperar o caminho? Sabemos que a menor distância será dada por $\text{dist}(t)$. Mas qual é este caminho?

   Cada vez que o rótulo de um vértice é modificado precisamos saber a partir de que vértice foi calculado o novo rótulo. Mantendo um vetor n-dimensional pred tal que

   - $\text{pred}(v)$ indica o vértice com rótulo permanente que deu origem ao rótulo do vértice v,
   - e se v for o vértice inicial, então $\text{pred}(s) = -1$;

   temos que o menor caminho é dado por:

   $$s, \text{pred}(\text{pred}(\cdots)), \ldots, \text{pred}(\text{pred}(t)), \text{pred}(t), t.$$

Implementação

Considere um digrafo $D(V, A)$, com n vértices e sua matriz de pesos $W = [w_{ij}]$, $n \times n$.

Queremos encontrar o menor caminho entre o vértice s e o vértice t no digrafo D.

Defina os vetores:

- final(i) – indica se o vértice v_i recebeu rótulo permanente (potencial) ou não;
- dist(i) – indica a distância acumulada do vértice inicial s até o vértice v_i;
- pred(i) – indica o vértice com rótulo permanente que deu origem ao rótulo do vértice v_i.

Algoritmo de Dijkstra

O algoritmo a seguir pode ser encontrado em [M. Syslo, N. Deo, J.S. Kowallk, Discrete Optimization Algorithms].

Inicialização

para todo $v \in V$ **faça**

início

$\text{dist}(v) = \infty$
$\text{final}(v) = \texttt{falso}$
$\text{pred}(v) = -1$

fim

$\text{dist}(s) = 0$

$\text{final}(s) = \texttt{verdadeiro}$

$recente = s$

Algoritmo de Dijkstra cont.

Iteração Principal

enquanto final(t) = `falso` **faça**

    início

        para todo vértice v_j t.q. exista a aresta $(recente, v_j)$ e t.q. final(j) = `falso` **faça**

            início (atualização dos rótulos)

                rotulo = dist$(recente) + w_{recente,j}$

                (o rótulo de v_j é modificado se houver um caminho menor de s a v_j através do vértice recente)

                se rotulo $<$ dist(j) **então**

                    início

                        dist(j) = rotulo

                        pred(j) = *recente*

                    fim

            fim

    (encontre o vértice com menor rótulo temporário)

    Seja y o vértice com menor rótulo temporário, tal que dist$(y) \neq \infty$

        início (o rótulo do vértice y se torna permanente)

            final(y) = `verdadeiro`

            $recente = y$

        fim

    fim

Exemplo

Rastrear o algoritmo de Dijkstra usando o digrafo com vértices s, a, b, c, d, t cuja matriz de pesos associada é:

$$W = \begin{bmatrix} 0 & 7 & 4 & 9 & 7 & \infty \\ 7 & 0 & 1 & \infty & \infty & 6 \\ 4 & 1 & 0 & 3 & \infty & \infty \\ 9 & \infty & 3 & 0 & 1 & 3 \\ 7 & \infty & \infty & 1 & 0 & 5 \\ \infty & 6 & \infty & 3 & 5 & 0 \end{bmatrix}.$$

Exemplo cont.

Inicialização

dist $= [\infty \quad \infty \quad \infty \quad \infty \quad \infty \quad \infty]$

final $= [\texttt{falso falso falso falso falso falso}]$

pred $= [-1 \quad -1 \quad -1 \quad -1 \quad -1 \quad -1]$

$\texttt{dist}(s) = 0$

$\text{final}(s) = \texttt{verdadeiro}$

$recente = s$

it = 1

Vértices v tais que existe a aresta $(recente, v)$ e tais que $\text{final}(v) = \texttt{falso}$: $\quad a, b, c, d$

Exemplo cont.

$\mathbf{v} = \mathbf{a}$:
rotulo $= \text{dist}(\mathit{recente}) + w_{\mathit{recente},a} = 0 + 7 < \infty = \text{dist(a)}$

$\text{dist}(a) = 7$
$\text{pred}(a) = s$

$\mathbf{v} = \mathbf{b}$:
rotulo $= \text{dist}(\mathit{recente}) + w_{\mathit{recente},b} = 0 + 4 < \infty = \text{dist(b)}$

$\text{dist}(b) = 4$
$\text{pred}(b) = s$

$\mathbf{v} = \mathbf{c}$:
rotulo $= \text{dist}(\mathit{recente}) + w_{\mathit{recente},c} = 0 + 9 < \infty = \text{dist(c)}$

$\text{dist}(c) = 9$
$\text{pred}(c) = s$

$\mathbf{v} = \mathbf{d}$:
rotulo $= \text{dist}(\mathit{recente}) + w_{\mathit{recente},d} = 0 + 7 < \infty = \text{dist(d)}$

$\text{dist}(d) = 7$
$\text{pred}(d) = s$

Exemplo cont.

Vértice y com menor rótulo temporário t.q. $\text{dist}(y) \neq \infty$: b

$\text{final}(b) = \texttt{verdadeiro}$

$recente = b$

$\text{it} = 2$

Vértices v tais que existe a aresta $(recente, v)$ e tais que $\text{final}(v) = \texttt{falso}$: a, c

$\text{v} = \text{a}$:

$\text{rotulo} = \text{dist}(recente) + w_{recente,a} = 4 + 1 < 7 = \text{dist(a)}$

$\text{dist}(a) = 5$

$\text{pred}(a) = b$

$\text{v} = \text{c}$:

$\text{rotulo} = \text{dist}(recente) + w_{recente,c} = 4 + 3 < 9 = \text{dist(c)}$

$\text{dist}(c) = 7$

$\text{pred}(c) = b$

Exemplo cont.

Vértice y com menor rótulo temporário t.q. $\text{dist}(y) \neq \infty$: b

$\text{final}(b) = \texttt{verdadeiro}$

$recente = b$

it = 2

Vértices v tais que existe a aresta $(recente, v)$ e tais que $\text{final}(v) = \texttt{falso}$: a, c

v = a:

$\text{rotulo} = \text{dist}(recente) + w_{recente,a} = 4 + 1 < 7 = \text{dist(a)}$

$\text{dist}(a) = 5$

$\text{pred}(a) = b$

v = c:

$\text{rotulo} = \text{dist}(recente) + w_{recente,c} = 4 + 3 < 9 = \text{dist(c)}$

$\text{dist}(c) = 7$

$\text{pred}(c) = b$

Exemplo cont.

Vértice y com menor rótulo temporário t.q. $\text{dist}(y) \neq \infty$: a

$\text{final}(a) = \texttt{verdadeiro}$

$recente = a$

it = 3

Vértices v tais que existe a aresta $(recente, v)$ e tais que $\text{final}(v) = \texttt{falso}$: t

v = t:

$\text{rotulo} = \text{dist}(recente) + w_{recente,t} = 5 + 6 < \infty = \text{dist(t)}$

$\text{dist}(t) = 11$

$\text{pred}(t) = a$

Vértice y com menor rótulo temporário t.q. $\text{dist}(y) \neq \infty$: c ou d

$\text{final}(c) = \texttt{verdadeiro}$

$recente = c$

Exemplo cont.

Vértice y com menor rótulo temporário t.q. $\text{dist}(y) \neq \infty$: a

$\text{final}(a) = \texttt{verdadeiro}$

$recente = a$

it = 3

Vértices v tais que existe a aresta $(recente, v)$ e tais que $\text{final}(v) = \texttt{falso}$: t

v = t:

$\text{rotulo} = \text{dist}(recente) + w_{recente,t} = 5 + 6 < \infty = \text{dist(t)}$

$\text{dist}(t) = 11$

$\text{pred}(t) = a$

Vértice y com menor rótulo temporário t.q. $\text{dist}(y) \neq \infty$: c ou d

$\text{final}(c) = \texttt{verdadeiro}$

$recente = c$

Exemplo cont.

it = 4

Vértices v tais que existe a aresta $(recente, v)$ e tais que final(v) = `falso`: $\quad d, t$

v = d:

rotulo $=$ dist$(recente) + w_{recente,d} = 7 + 1 \not< 7 =$ dist(d)

v = t:

rotulo $=$ dist$(recente) + w_{recente,t} = 7 + 3 < 11 =$ dist(t)

dist$(t) = 10$

pred$(t) = c$

Vértice y com menor rótulo temporário t.q. dist$(y) \neq \infty$: $\quad d$

final(d) = `verdadeiro`

$recente = d$

Exemplo cont.

it = 5

Vértices v tais que existe a aresta $(recente, v)$ e tais que final$(v) =$ `falso`: t

v = t:

rotulo $=$ dist$(recente) + w_{recente,d} = 7 + 5 \not< 10 =$ dist(t)

Vértice y com menor rótulo temporário t.q. dist$(y) \neq \infty$: t

- final$(t) =$ `verdadeiro`
- $recente = t$

Como final$(t) =$ `verdadeiro`, **pare**

Exemplo cont.

Qual é o valor do menor caminho?

$\text{dist}(t) = 10$

Qual é este caminho?

$t, \text{pred}(t)$

$t, c, \text{pred}(c)$

$t, c, b, \text{pred}(b)$

$t, c, b, s, \text{pred}(s)$

$t, c, b, s, -1$

O menor caminho é $\{s, b, c, t\}$

Exemplo cont.

E se quisermos o menor caminho entre s e c? Podemos aproveitar os cálculos anteriores.

Como c possui rótulo permanente, temos que:

$\text{dist}(c) = 7$

E o caminho é:

$c, \text{pred}(c)$

$c, b, \text{pred}(b)$

c, b, s

O menor caminho é então: $\{s, b, c\}$.

Observações

- A complexidade em termos de tempo computacional do algoritmo acima é dada por:

  O laço principal deste algoritmo, no pior caso, é executado $n-1$ vezes. Isto acontece quando o vértice final, t, é o último a receber um rótulo permanente. Para cada execução deste laço, precisamos examinar uma linha da matriz de pesos, e atualizar os vetores dist e pred, ou seja, um tempo proporcional a n. Assim o tempo total é da ordem $2n(n-1)$. A complexidade é $O(n^2)$.

  Observe que independe do número de arestas no grafo.
- É possível, usando o algoritmo de Dijkstra encontrar o menor caminho entre o vértice s e todos os outros vértices do grafo. O que deve ser modificado?

Observações

- Outros Algoritmos para problemas de Caminho Mínimo em grafos podem ser encontrados por exemplo em: [Goldbarg, M. e E. Goldbarg, Grafos: conceitos, algoritmos e aplicações, Elsevier, 2012].
- O algoritmo de Dijskstra funciona apenas se $w_{ij} \geq 0$ para todos i, j. Verifique este fato aplicando o algoritmo à seguinte rede para encontrar o menor caminho entre v_1 e v_3: