PRINCIPAIS PROBLEMAS DE OTIMIZAÇÃO EM REDES

DCE692 - Pesquisa Operacional

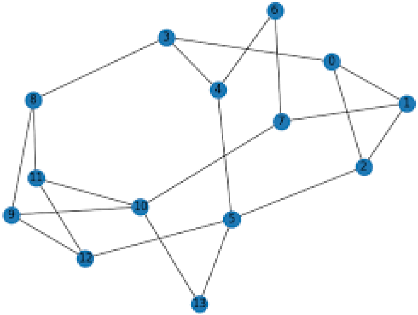
Atualizado em: 16 de outubro de 2023

Iago Carvalho

Departamento de Ciência da Computação

Problemas em redes são aqueles que podem ser representados como uma rede

- Conjunto de elementos
  - Nós
  - Vértices
- Conexões entre os elementos
  - Arcos
  - Arestas



PRINCIPAIS PROBLEMS EM REDES

Existem diversos problemas de otimização em redes

- Alguns podem ser modelados utilizando programação linear
- Outros utilizam programação inteira
  - São problemas NP-Completos

Aqui nós vamos falar de alguns deles

1. Fluxo em redes
   - Fluxo máximo
   - Fluxo de custo mínimo
2. **Caminho mínimo**
3. **Árvore Geradora Mínima**

CAMINHO MÍNIMO

CAMINHO EM GRAFOS

Estamos interessados em encontrar o caminho entre dois vértices de um grafo

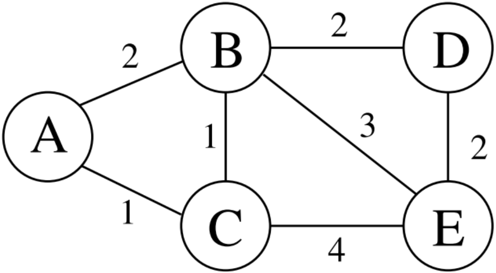
- Diversas aplicações práticas

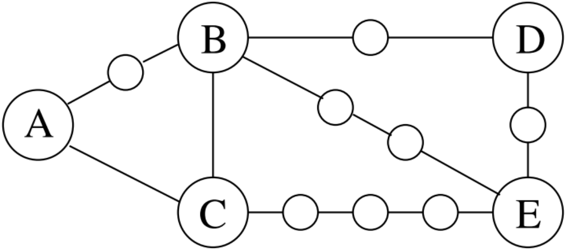
Em grafos não ponderados, podemos utilizar os algoritmos de busca em largura ou de busca em profundidade

- Busca em largura = percurso em níveis
- Busca em profundidade = percurso em pré-ordem

Entretanto, estes dois algoritmos não lidam com arestas (ou arcos) ponderados

GRAFO NÃO-PONDERADO VS PONDERADO





DEFINIÇÃO DO PROBLEMA DO CAMINHO MÍNIMO

O problema do caminho mínimo é definido sobre um grafo não-orientado ponderado $G = (V, E)$

- Ou sobre um grafo orientado e ponderado $G = (V, A)$

Existe uma função w que mapeia cada aresta (ou arco) a um valor real que simboliza o peso da aresta (ou arco)

$$w : E \mapsto \mathbb{R} \quad \text{ou} \quad w : A \mapsto \mathbb{R}$$

DEFINIÇÃO DO PROBLEMA DO CAMINHO MÍNIMO

O peso do caminho p

$$p = < v_0, v_1, \ldots, v_k >$$

é igual ao somatório dos pesos de suas arestas (ou arcos)

$$W(p) = \sum_{i=1}^{k} w(v_{i-1}, v_i)$$

DEFINIÇÃO DO PROBLEMA DO CAMINHO MÍNIMO

O caminho mais curto entre dois vértices $u, v \in V$ pode ser definido como

$$\delta(u,v) = \begin{cases} \min \quad w(p) : u \overset{p}{\sim} v, & \text{se existe caminho entre } u \text{ e } v \\ \infty, & \text{caso contrario} \end{cases}$$

Ou seja, o caminho mais curto entre u e v é

$$w(p) = \delta(u,v)$$

SUBESTRUTURA ÓTIMA DO CAMINHO MÍNIMO

Seja $p =< v_0, v_1, \ldots, v_k >$ o caminho mínimo entre os vértices v_0 e v_k

- $w(p) = \delta(v_0, v_k)$

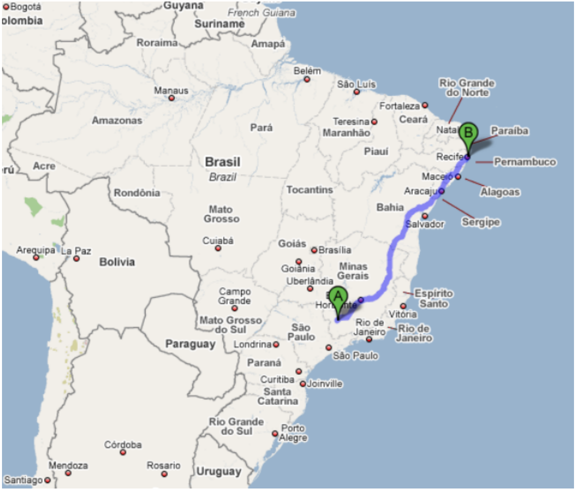
Para quaisquer i e j tais que $1 \leq i < j \leq k$

- Seja $p_{ij} =< v_i, v_{i+1}, \ldots, v_j >$ o subcaminho entre i e j em p

Temos que $p_{ij} = \delta(v_i, v_j)$

- Isto é, o caminho mínimo entre v_i e v_j é um subcaminho de p

PROBLEMA DO CAMINHO MÍNIMO



Seja $G = (N, A)$ um grafo

- O conjunto N são os vértices
  - Vértice inicial $s \in N$ e vértice final $t \in N$
- O conjunto A são os arcos entre dois vértices

Todo arco a_{ij} possui um custo c_{ij}

O problema é definido utilizando variáveis $x_{ij} > 0$

- Existe uma variável x_{ij} para todo arco $a_{ij} \in A$
- $x_{ij} = 1$ se o arco faz parte do caminho mínimo entre s e t
- $x_{ij} = 0$ caso contrário

CAMINHO MÍNIMO POR PROGRAMAÇÃO LINEAR

A função objetivo do modelo tenta minimizar o custo do caminho entre s e t

- Isto é equivalente a minimizar o custo das arestas escolhidas

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

CAMINHO MÍNIMO POR PROGRAMAÇÃO LINEAR

Temos que garantir a continuidade do caminho

- Deve existir um único fluxo contínuo entre s e t

Para isto, devemos garantir 3 coisas

1. Sai uma unidade de fluxo do vértice s
2. Chega uma unidade de fluxo no vértice t
3. Tudo que chega tem que sair nos outros vértices

$$\sum_{(i,k)\in A} x_{ik} - \sum_{(k,j)\in A} x_{kj} = \begin{cases} 1, & \text{se } k = s \\ -1, & \text{se } k = t \\ 0, & \text{caso contrário} \end{cases} \quad ,\forall k \in V$$

Por fim, devemos dizer que todas as variáveis são reais e positivas

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

O modelo completo de programação linear do caminho mínimo pode ser descrito como

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

$$\sum_{(i,k)\in A} x_{ik} - \sum_{(k,j)\in A} x_{kj} = \begin{cases} 1, & \text{se } k = s \\ -1, & \text{se } k = t \\ 0, & \text{caso contrário} \end{cases} \quad ,\forall k \in V$$

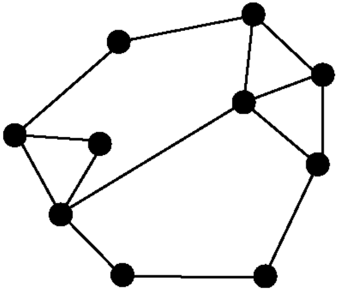
$$x_{ij} > 0 \qquad (i,j) \in A$$

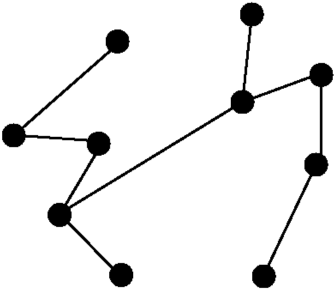
ÁRVORE GERADORA MÍNIMA

Sega $G = (V, E)$ um grafo não-orientado e ponderado

Uma árvore $T = (V, E')$ é um subgrafo induzido de G tal que

- T é conexo
- $|E'| = |V| - 1$





ÁRVORE GERADORA MÍNIMA

Uma árvore geradora mínima (AGM) é a árvore de menor custo total

Assim como no problema do caminho mínimo, considere que w que mapeia cada aresta a um valor real que simboliza o peso da aresta

$$w : E \mapsto \mathbb{R}$$

O peso de uma árvore geradora T é definido como $\sum_{e \in E'} w_e$

Existe um número exponencial de diferentes árvores geradoras em um grafo

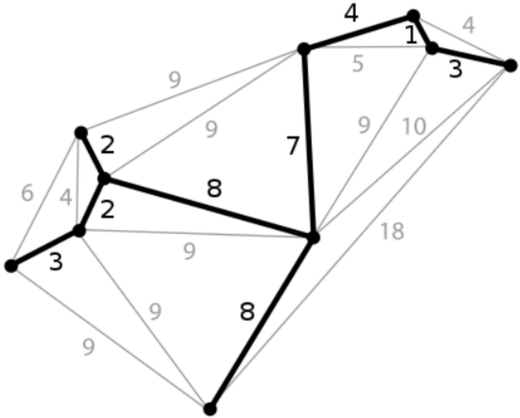
- Para ser mais preciso, existem n^{n-2} diferentes AGMs
- Enumerar todas elas é um problema $\#p$-Completo

O problema da árvore geradora mínima consiste em encontrar a AGM de menor peso dentre todas as possíveis árvores

- Problema pode ser resolvido em tempo polinomial
- Subestrutura ótima

Em outras palavras, consiste em encontrar a combinação de arestas cuja AGM resultante tenha o menor peso possível

ÁRVORE GERADORA MÍNIMA



APLICAÇÕES

Diversos problemas práticos são modelados como AGMs

- Tabela de roteamento em redes de computadores
- Modelar uma rede de comunicações
- Análise de clusters
- Desenho de circuitos VLSI (microprocessadores)
- Taxonomia
- Problema base para diversos outros algoritmos e problemas em grafos

ÁRVORE GERADORA MÍNIMA POR PROGRAMAÇÃO LINEAR

Seja $G = (N, E)$ um grafo

- O conjunto N são os vértices e E são as arestas

A formulação mais simples para AGM utiliza arcos

- Para toda aresta $e_k \in E$ que conecta os vértices $i \in N$ e $j \in N$
- Definimos um par de variáveis a_{ij} e a_{ji}
- Definimos um conjunto de arcos A

Todo arco a_{ij} possui um custo c_{ij}

- Custo é reflexivo: $a_{ij} = a_{ji}$

ÁRVORE GERADORA MÍNIMA POR PROGRAMAÇÃO LINEAR

O problema é definido utilizando variáveis $x_{ij} > 0$

- Existe uma variável x_{ij} para todo arco a_{ij}
- $x_{ij} = 1$ se o arco faz parte da árvore geradora mínima
- $x_{ij} = 0$ caso contrário

Além disso, variáveis de fluxo $y_{ij}^{p} > 0$ adicionais são utilizadas

- Existem p variáveis y_{ij} para todo arco a_{ij}
- y_{ijp} determina a quantidade de fluxo entre os vértices $i \in N$ e $j \in N$

A função objetivo do modelo minimiza o custo da árvore geradora

- Isto é equivalente a minimizar o custo das arestas escolhidas

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

ÁRVORE GERADORA MÍNIMA POR PROGRAMAÇÃO LINEAR

Temos que garantir a conectividade da árvore

- Deve existir um único fluxo contínuo entre um vértice especial r e todos os outros vértices do grafo

Para isto, devemos garantir 2 coisas

1. Saem $|V|-1$ unidades de fluxo do vértice r
2. Chega uma unidade de fluxo em cada outro vértice

Isto implica que existem $|V|-1$ caminhos na árvore

- Existe um caminho de r para todos os outros vértices

$$\sum_{(i,k)\in A} y_{ik}^{p} - \sum_{(k,j)\in A} y_{kj}^{p} = \begin{cases} 1, & \text{se } k = r \\ -1, & \text{se } k = p \\ 0, & \text{caso contrário} \end{cases} \quad \forall k \in V,\ p \in V \setminus \{r\}$$

ÁRVORE GERADORA MÍNIMA POR PROGRAMAÇÃO LINEAR

Devemos conectar as variáveis x e y do modelo de tal forma que x_{ij} assume um valor sempre maior ou igual a y_{ij}^p, para todo $p \in V \setminus \{r\}$

- Fazemos isto pois $\sum_{p \in V} y_{ij}^p$ pode ser maior que 1
- Entretanto, x_{ij} nunca será maior do que 1

$$x_{ij} \geq y_{ij}^p, \qquad \forall (i,j) \in A, \quad p \in V \setminus \{r\}$$

Também devemos garantir uma árvore só pode ter $|V| - 1$ arestas

$$\sum_{(i,j) \in A} x_{ij} \leq |V| - 1$$

Por fim, devemos dizer que todas as variáveis são reais e positivas

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

$$y_{ij}^{p} \geq 0, \qquad \forall (i,j) \in A, \quad p \in V \setminus \{r\}$$

ÁRVORE GERADORA MÍNIMA POR PROGRAMAÇÃO LINEAR

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

$$\sum_{(i,j)\in A} x_{ij} \leq |V| - 1$$

$$x_{ij} \geq y_{ij}^{p}, \qquad \forall (i,j) \in A, \quad p \in V \setminus \{r\}$$

$$\sum_{(i,k)\in A} y_{ik}^{p} - \sum_{(k,j)\in A} y_{kj}^{p} = \begin{cases} 1, & \text{se } k = r \\ -1, & \text{se } k = p \\ 0, & \text{caso contrário} \end{cases} \quad \forall k \in V,\ p \in V \setminus \{r\}$$

$$x_{ij} \geq 0, \quad y_{ij}^{p} \geq 0, \qquad \forall (i,j) \in A, \quad p \in V \setminus \{r\}$$

PRÓXIMA AULA:
EXERCÍCIOS