Algoritmos Aproximados para Problemas NP Completos

Parte 1

Introdução

- Problemas intratáveis ou difíceis são comuns na natureza e nas áreas do conhecimento.
  - “Fáceis” → resolvidos por algoritmos polinomiais
    - Busca Binária: O(logn), Pesquisa Sequencial: O(n), Ordenação por Merge Sorte: O(nlogn)
  - “Difíceis” → resolvidos por algoritmos exponenciais

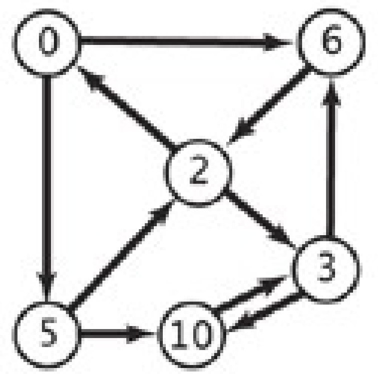
Grafos

- Vértices e arcos (arestas)
- Grafo dirigido
  - Arco “começa” e “termina” em um vértice
  - Notação: v-w
    - Arco sai de v e entra em w

Grafos

- Maneira de representar um grafo

0-5 0-6 2-0 2-3 3-6 3-10 4-1 5-2 5-10

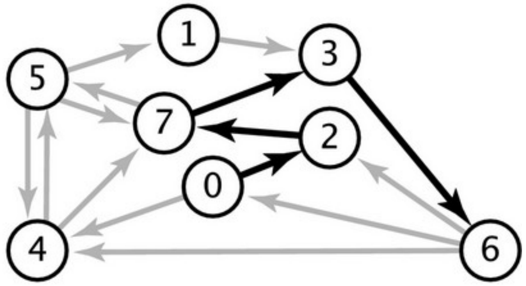


Grafos

- Passeio em grafo
  - Um passeio (= walk) em um grafo é uma sequência de vértices
    - se v e w são vértices consecutivos na sequência então v-w é um arco do grafo.

Grafos

- Caminho em grafo
  - Um passeio sem arcos repetidos
- Por exemplo, 0-2-7-3-6 é um caminho simples no grafo da figura.



Problema do Caminho Mínimo

- Dado um vértice **s** de um grafo com custos **positivos** nos arcos, encontrar uma árvore de caminhos baratos com raiz **s** no grafo.
  - Problema foi descoberto por Edsger W. Dijkstra em 1959

Algoritmo de Dijkstra

- A franja (= fringe) de uma subárvore radicada T de G é o conjunto de todos os arcos do grafo que têm ponta inicial em T e ponta final fora de T.
- Leque de saída do conjunto de vértices de T.

Algoritmo de Dijkstra

- O processo iterativo consiste no seguinte: enquanto a franja de T não estiver vazia,
  - escolha, na franja de T, um arco x-y que minimize dist[x] + c_{xy} ,
  - acrescente o arco x-y e o vértice y a T
  - faça dist[y] = dist[x] + c_{xy} .
- c_{xy} é o custo do arco x-y

Exemplo

- Grafo com custos a seguir (raiz em 0):

0-1	0-2	1-3	1-4	2-3	2-4	3-1	3-5	4-5
10	20	70	80	50	60	0	10	10

Exemplo

- Grafo com custos a seguir (raiz em 0):

0-1	0-2	1-3	1-4	2-3	2-4	3-1	3-5	4-5
10	20	70	80	50	60	0	10	10

	dist[]						
T	0	1	2	3	4	5	franja
0	0	*	*	*	*	*	0-1 0-2
0 1	0	10	*	*	*	*	0-2 1-3 1-4
0 1 2	0	10	20	*	*	*	1-3 1-4 2-3 2-4
0 1 2 3	0	10	20	70	*	*	1-4 2-4 3-5
0 1 2 3 4	0	10	20	70	80	*	3-5 4-5
0 1 2 3 4 5	0	10	20	70	80	80	

0-1	0-2
10	20

2-3	2-4	3-5
50	60	10

Exercício

- Grafo com custos a seguir (raiz em 0):

0-2	0-3	0-4	2-4	3-4	3-5	4-1	4-5	5-1	1-2
70	50	30	10	10	20	0	30	50	30

Exercício

- Grafo com custos a seguir (raiz em 0):

0-2	0-3	0-4	2-4	3-4	3-5	4-1	4-5	5-1	1-2
70	50	30	10	10	20	0	30	50	30

T	dist[]	franja
0	0 * * * * *	0-2 0-3 0-4
0 4	0 * * * 30 *	0-2 0-3 4-1 4-5
0 4 1	0 30 * * 30 *	0-2 0-3 4-5 1-2
0 4 1 3	0 30 * 50 30 *	0-2 4-5 1-2 3-5
0 4 1 3 5	0 30 * 50 30 60	0-2 1-2
0 4 1 3 5 2	0 30 60 50 30 60	

0-3	0-4
50	30

4-1	4-5	1-2
0	30	30

Algoritmo de Dijkstra

- Implementação ingênua
  - Cada iteração recalcula a franja
- Possível otimizar o algoritmo proposto
  - Vértices maduros x imaturos
  - Fila priorizada implementada por um heap

Algoritmo de Dijkstra

- Análise do algoritmo
  - Versão apresentada: $O(V^2)$
  - Versão melhorada: $O(A \log V)$

Problemas Fáceis x Difíceis

- Considere um grafo com peso positivo nas arestas, dois vértices i e j e um inteiro k>0.
  - Existe um caminho de i até j com peso <=k?

- Fácil ou Difícil?

Problemas Fáceis x Difíceis

- Considere um grafo com peso positivo nas arestas, dois vértices i e j e um inteiro k>0.
  - Existe um caminho de i até j com peso >k?

- Fácil ou Difícil

Problemas Fáceis x Difíceis

- Considere um grafo com peso positivo nas arestas, dois vértices i e j e um inteiro k>0.
  - Existe um caminho de i até j com peso >k?

- Fácil ou **Difícil**
  - Não existe algoritmo eficiente. É equivalente ao PCV em termos de complexidade.