Ciclo Hamiltoniano
e
Problema do Caixeiro Viajante

Problema do Ciclo Hamiltoniano

Dado um grafo *G* = (*V*,*E*), um ciclo H que passa por todos os vértices de *G* é dito Hamiltoniano.

***G* é dito Hamiltoniano se possui ciclo Hamiltoniano.**

Problema: Dado *G* saber se *G* é Hamiltoniano.

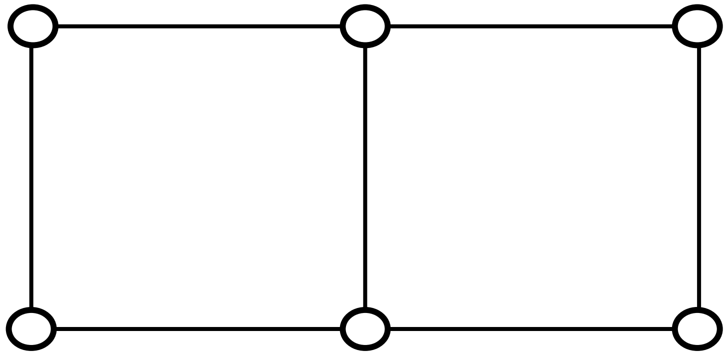
Problema do Caminho Hamiltoniano

Dado um grafo $G = (V,E)$, um caminho C que passa por todos os vértices de G é dito semi-Hamiltoniano.

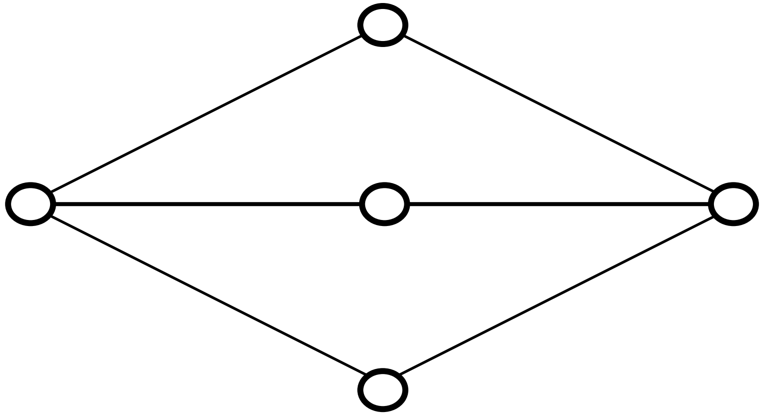
G **é dito semi-Hamiltoniano se possui caminho Hamiltoniano.**

Problema: Dado *G* saber se *G* é semi-Hamiltoniano.

Ciclo Hamiltoniano



Grafo Hamiltoniano



Grafo semi-Hamiltoniano

Ciclo Hamiltoniano

Os problemas de decisão:

***G* é hamiltoniano?**

***G* é semi-hamiltoniano?**

são
NP-completos

Ciclo Hamiltoniano

Condição suficiente

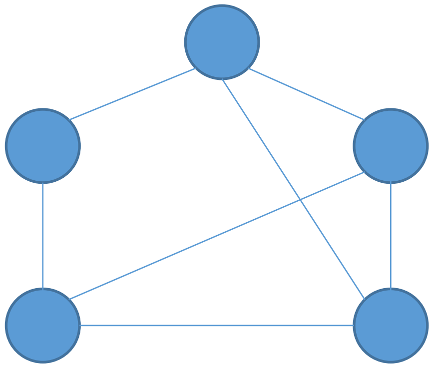
Teorema de Chvátal (1972)

Se um grafo simples G de ordem $n \geq 3$ possui sequência de graus $d_1 \leq d_2 \leq \ldots \leq d_n$, tal que para todo k, $1 \leq k < n/2$, $d_{n-k} \geq n-k$ ou $d_k > k$, então G é hamiltoniano.

Ciclo Hamiltoniano

Teorema de Chvátal (1972)

Se um grafo simples G de ordem $n \geq 3$ possui sequência de graus $d_1 \leq d_2 \leq \ldots \leq d_n$, tal que para todo k, $1 \leq k < n/2$, $d_{n-k} \geq n-k$ ou $d_k > k$, então G é hamiltoniano.



Sequência de graus: 2, 3, 3, 3, 3

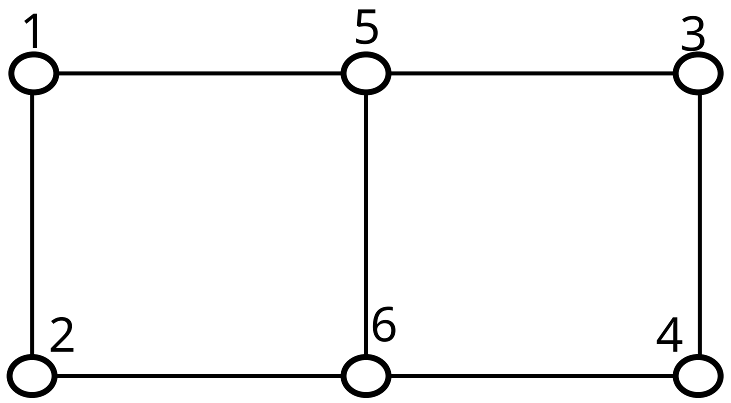
Cumpre a condição do teorema.

O grafo é Hamiltoniano

Ciclo Hamiltoniano

Teorema de Chvátal (1972)

Se um grafo simples G de ordem $n \geq 3$ possui sequência de graus $d_1 \leq d_2 \leq \ldots \leq d_n$, tal que para todo k, $1 \leq k < n/2$, $d_{n-k} \geq n-k$ ou $d_k > k$, então G é hamiltoniano.



$n = 6$ e $k = 2$

$d_2 = 2$

$d_4 = 2 \geq 6\text{-}2$ NÃO

Sequência de graus: 2, 2, 2, 2, 3, 3

Não cumpre a condição do teorema.

O grafo é Hamiltoniano

Ciclo Hamiltoniano

Teorema de Dirac, 1952

Um grafo $G = (N, M)$ com $n \geq 3$ e $d(x) \geq n/2$ para todo $x \in N$, é hamiltoniano.

Ciclo Hamiltoniano

Teorema de Ore, 1961

Uma condição suficiente para que um grafo G seja hamiltoniano é que a soma dos graus de cada par de vértices não adjacentes seja no mínimo n.

Problema do Caixeiro Viajante

Problema do Caixeiro Viajante

Dado um grafo ponderado *G* = (*V*,*E*), encontrar um ciclo gerador H cuja soma das arestas seja mínima dentre todos os ciclos geradores de *G*.

*"Dado **N** cidades, achar a caminho mais curto passando por todas as cidades uma única vez."*

Aplicações

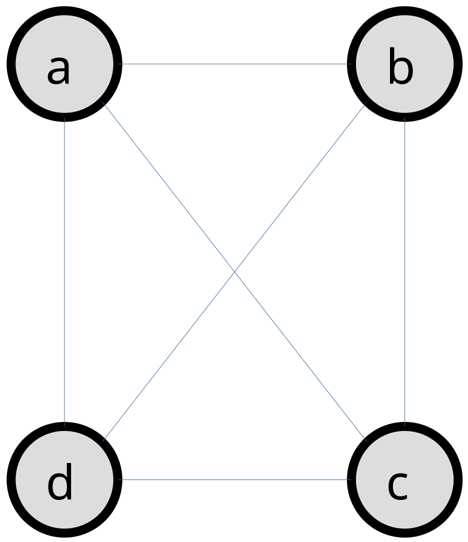
- Programação da rota de ônibus escolares (Angel *et al*., 1972).
- Programação de tripulação (Svestka & Huckfeldt, 1973).
- Fiação de computadores (Lenstra & Rinnooy Kan, 1974).
- Movimentação de material em depósitos (Ratliff & Rosenthal, 1983).
- Programação de transporte entre células de manufatura (Finke & Kusiak , 1985).
- Em grande parte dos problemas de roteamento de veículos (Bodin *et al*.,1983).
- Programação de operações de máquinas em manufatura (Kusiak & Finke, 1987).
- Otimização do movimento de ferramentas de corte (Chauny *et al*., 1987).
- Manutenção de motores de turbina a gás (Plante *et al*., 1987).
- Otimização de perfurações de furos em placas de circuitos impressos (Reinelt, 1989).
- Cristalografia através de raio X (Bland & Shallcross, 1989).
- Na solução de problemas de sequenciamento (Whitley *et al*., 1991).
- Agendamento de entrevistas (Gilbert & Hofstra, 1992).
- Trabalhos administrativos (Laporte *et al.*, 1996).
- Na solução de problemas de programação e distribuição de tarefas em plantas (Salomon *et al.*, 1997).
- Programação de laminação em siderurgia (Tang *et al*., 2000).

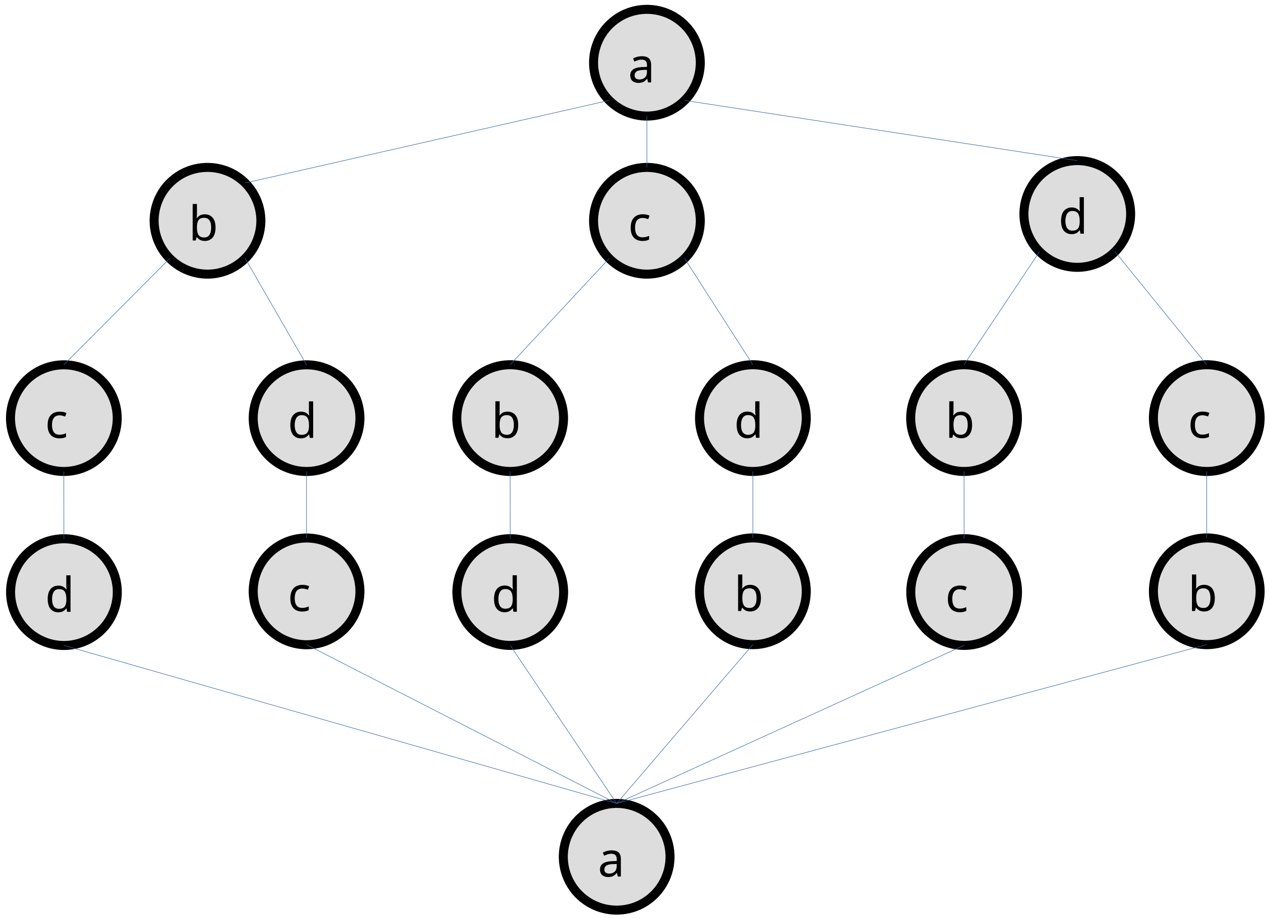
Aplicações

- Programação de operações de monitoramento de satélites (Czerniak, 2000).
- Construção de mapas genéticos e mapeamento de cromossomos (Agarwala *et al*., 2000).
- Programação de Impressão em gráficas ou jornais (Carter & Ragsdale, 2002).
- Planejamento do caminho de robôs cooperativos (Yu *et al*., 2002).
- Posicionamento de satélites para cobertura de comunicação (Saleh & Chelouah, 2004).
- Identificação do Alzheimer (De Vreese *et al*., 2005).
- Predição de propriedades de Proteínas (Johnson & Liu, 2006).
- Classificação de componentes eletrônicos (Goyal, 2006).
- Aplicações reais do TSP em Flowshop (Bagchi *et al*., 2006 e Caricato *et al*., 2007).
- Distribuição de combustível para postos de gasolina (Ismail & Ibrahim, 2008).
- Mamografia (Lupşa *et al.* 2008).
- Programação de laminação a frio (Zhao *et al*., 2011).
- Sistema antimíssil (Gao & Wang, 2011).
- Otimização da busca de planetas extra-solares (Kolemen & Kasdin, 2012).
- Perfuração de peças (Tsai *et al*., 2012)
- Cirurgia Endoscópica (Falcone *et al*., 2013).

Problema do Caixeiro Viajante

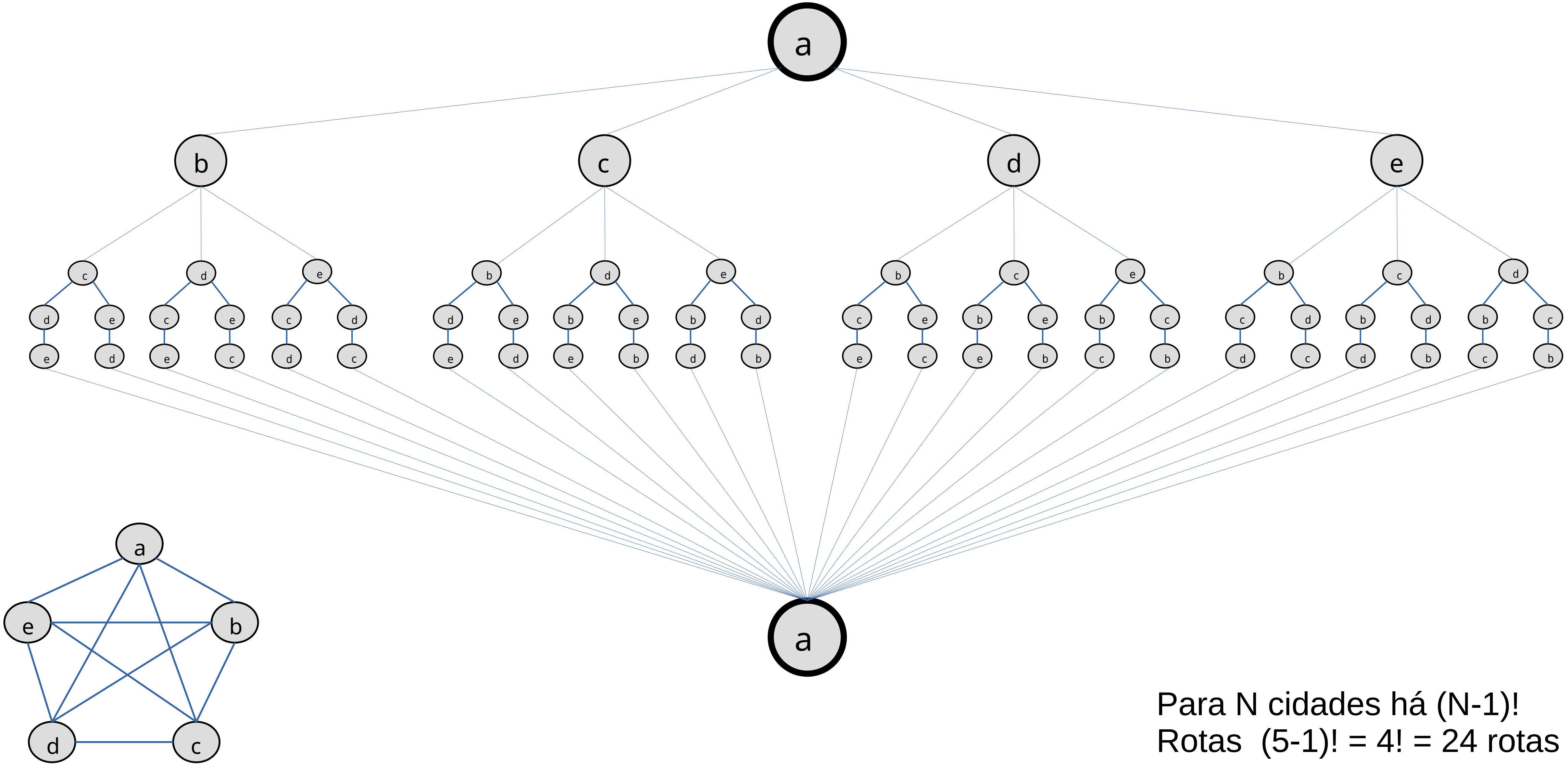
*“Dado **N** cidades, achar a caminho mais curto passando por todas as cidades uma única vez.”*





Para N cidades há (N-1)!
Rotas (4-1)! = 3! = 6 rotas

Problema do Caixeiro Viajante



Para N cidades há (N-1)!
Rotas (5-1)! = 4! = 24 rotas

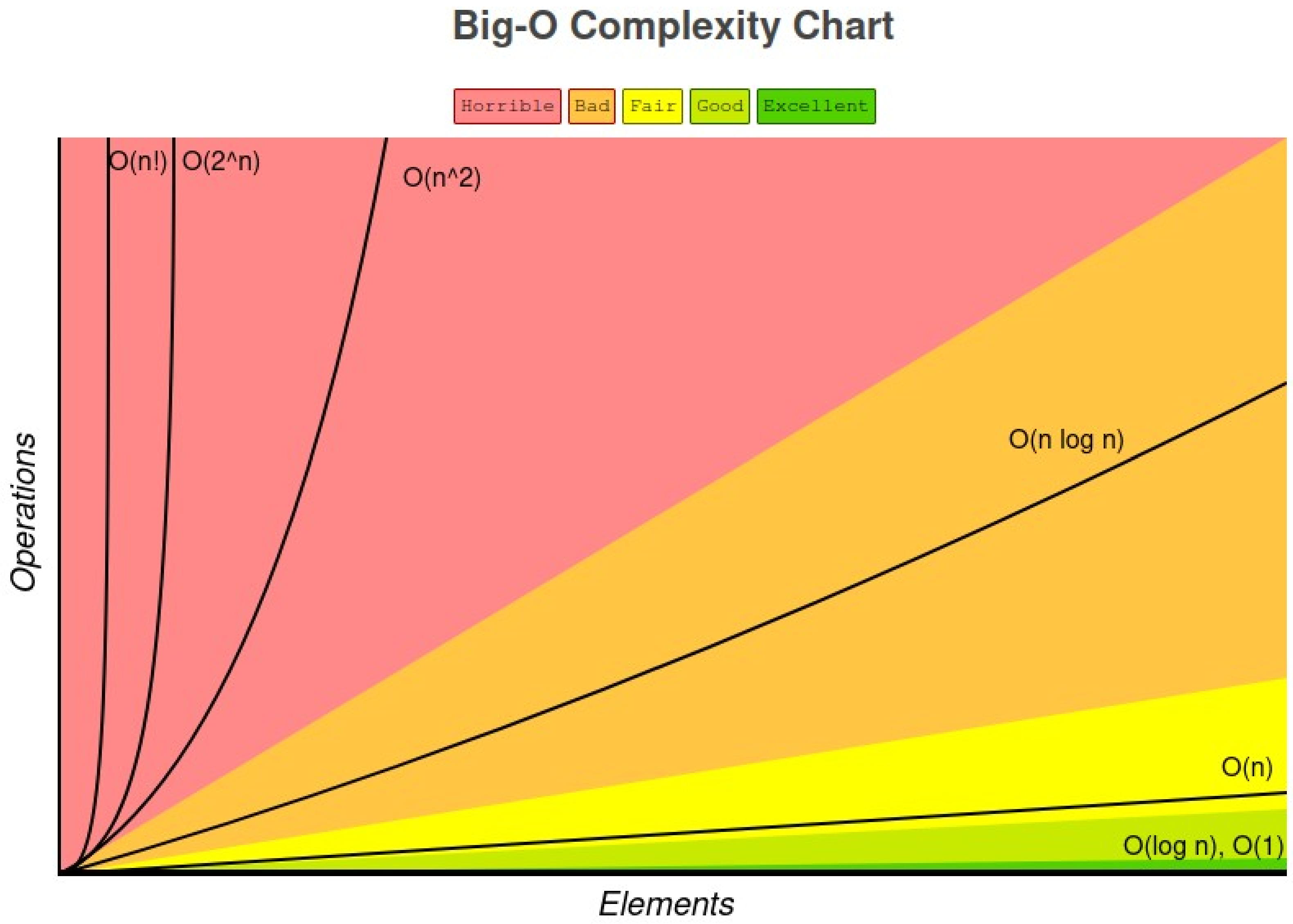
Problema do Caixeiro Viajante

Dado um grafo *G* ponderado em arestas, encontrar o ciclo-hamiltoniano de comprimento mínimo.

Para N cidades há (N-1)! Rotas.

N= 11 --> 3.628.800 rotas.
N= 12 --> 39.916.800 rotas.
N= 26 --> 15.511.210.043.330.985.984.000.000 rotas.

Problema do Caixeiro Viajante



https://www.bigocheatsheet.com/

Problema do Caixeiro Viajante

Dado um grafo *G* ponderado em arestas, encontrar o ciclo-hamiltoniano de comprimento mínimo.

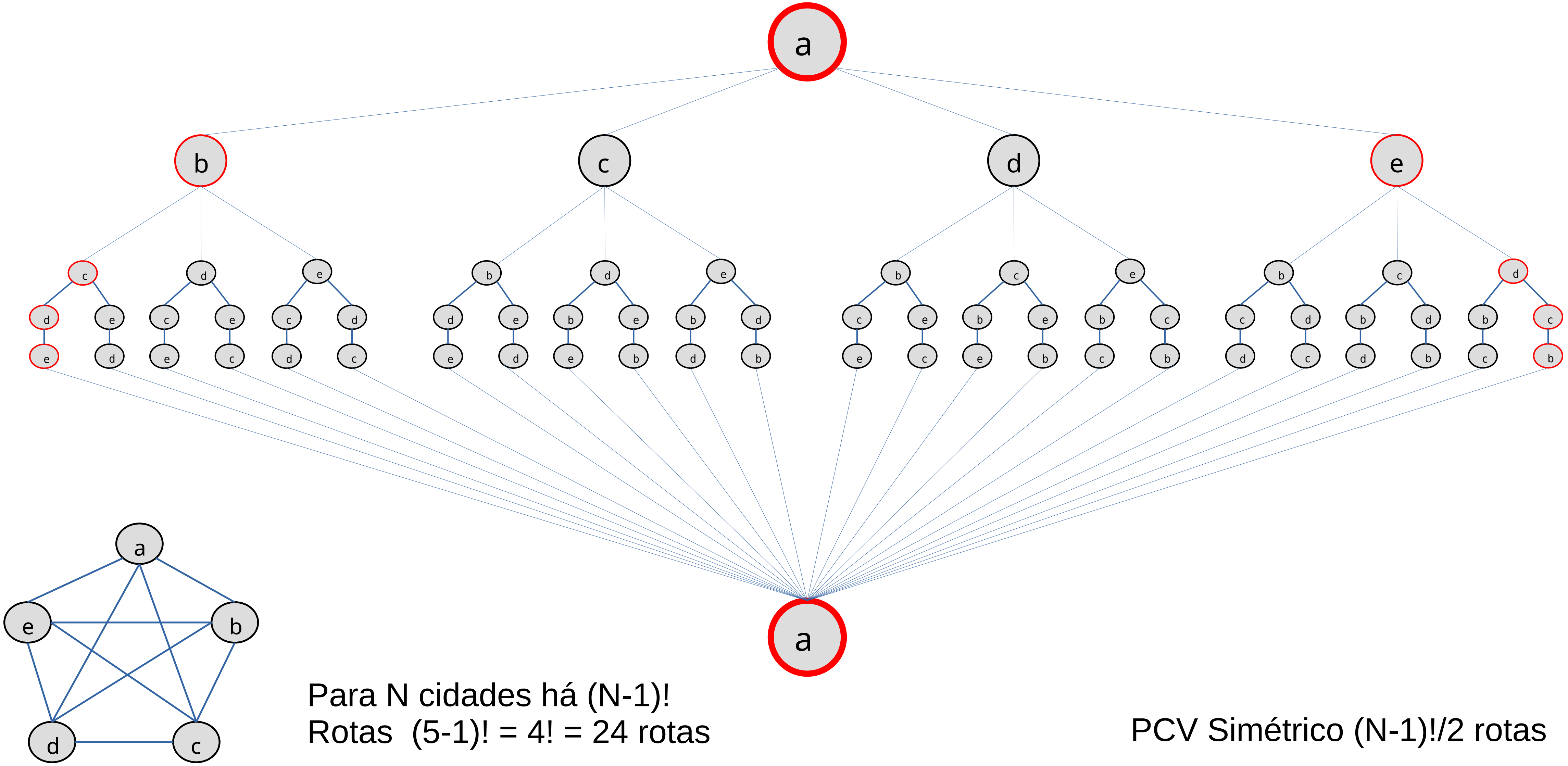
Um problema simétrico com *n* cidades possui (n-1)!/2 soluções.

Se *n*= 20, então existem mais de 10^{18} soluções.

Karp (1975) provou que o problema pertence à classe **NP-difícil**.

Karp, R. M. (1975). On the Computational Complexity of Combinatorial Problems, *Networks* 5:45-68

Problema do Caixeiro Viajante

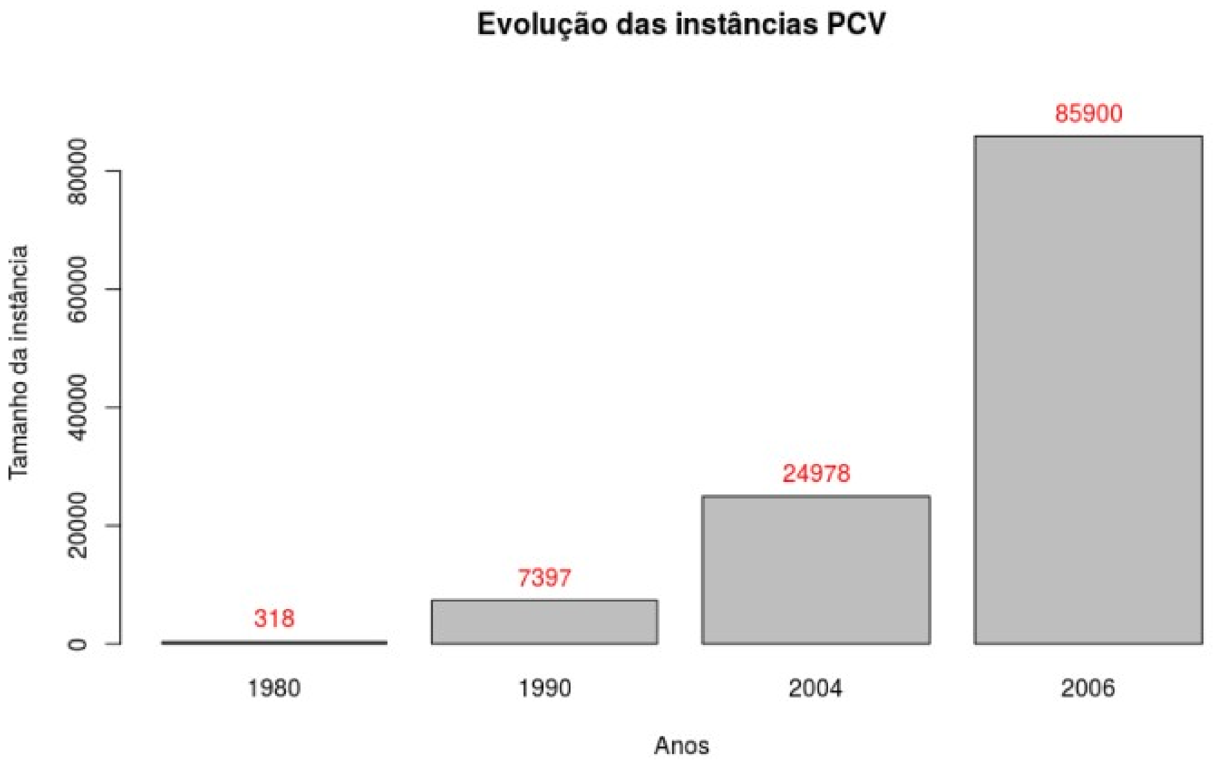


Para N cidades há (N-1)!
Rotas (5-1)! = 4! = 24 rotas

PCV Simétrico (N-1)!/2 rotas

Problema do Caixeiro Viajante

- PCV é NP-Difícil (Grarey & Johnson, 1979)
- Um dos problemas mais pesquisados



Problema do Caixeiro Viajante

Sahni e Gonzalez (1976) provaram que para qualquer $\varepsilon > 0$, existirá um algoritmo ε-aproximado para o Problema do Caixeiro Viajante se e somente se

P = NP

Problema do Caixeiro Viajante

Problema do Caixeiro Viajante

Teorema.
Dada uma constante *c*, se P ≠ NP, então não existe algoritmo *c*-aproximado para o PCV, isto é, não existe algoritmo de aproximação polinomial tal que $Z_H \leq cZ_{ótimo}$.

Algoritmos Exatos

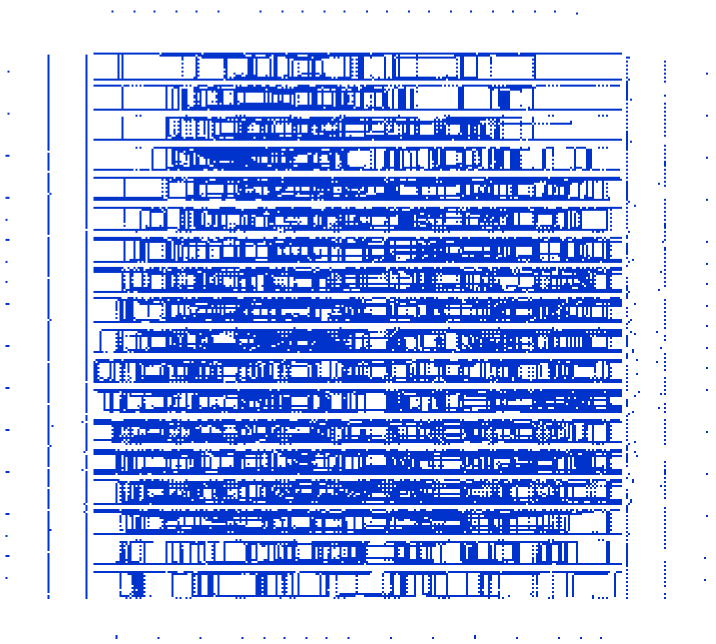
Os algoritmos mais eficientes para o problema são devidos a Padberg e Rinaldi (1991), Grötchell e Holland (1991) e Applegate et al. (1995).

M.W. Padberg, G. Rinaldi, A branch-and-cut algorithm for the resolution of large-scale symmetric traveling salesman problems, SIAM Review 33 (1991) 60-100.

M. Grötchel, O. Holland, Solution of large scale symmetric traveling salesman problems, Mathematical Programming 51 (1991) 141-202.

D. Applegate, R. Bixby, V. Chvàtal, W. Cook, Finding cuts in the TSP (A preliminary report), DIMACS, Tech. Report 95-05, 1995.

Maior instância solucionada



85.900 pontos
Pla85900
(2006) - Concorde

Algoritmos Aproximativos

Vizinho mais Próximo (Guloso)

Belmore e Nemhauser (1988)

1. **Escolher** um vértice inicial
2. **Encontrar** o vértice v_k mais próximo ao último vértice (vértices extremos do caminho) incluído no caminho
3. **Inserir** o vértice v_k após o último vértice (o seu vizinho mais próximo)
4. **Caso** o ciclo formado seja Hamiltoniano pare.
   Caso contrário volte a etapa 2.

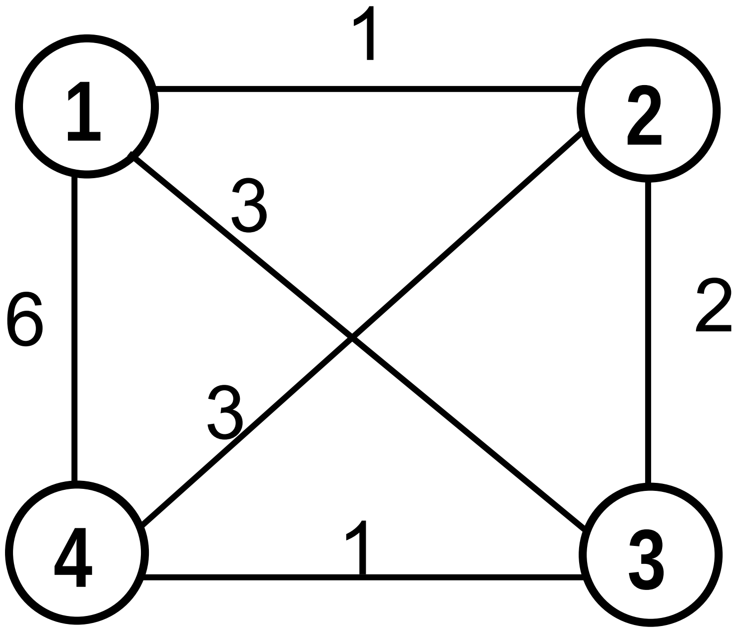
Algoritmos Exatos e Heurísticas

Duas classes de algoritmos são usados para resolver problemas de otimização combinatória:

- Algoritmos exatos.
- Técnicas heurísticas (ou algoritmos aproximados).

Algoritmos Aproximativos

Exemplo 1



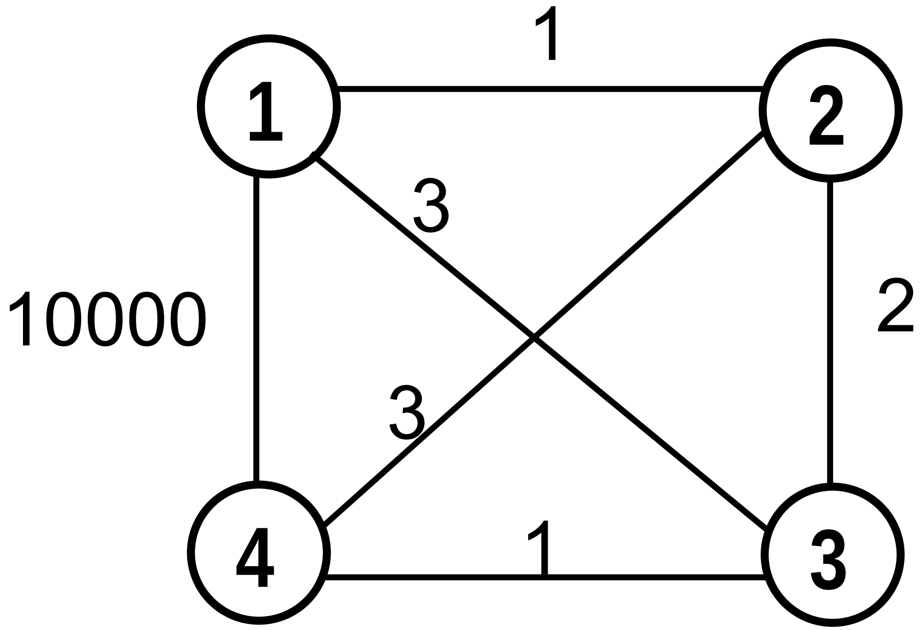
Iniciando no vértice 1
1 – 2 – 3 – 4 – 1
Custo = 10

Melhor solução = 8
1 – 2 – 4 – 3 – 1

10/8 = 1,25 – fator de aproximação

Algoritmos Aproximativos

Exemplo 2



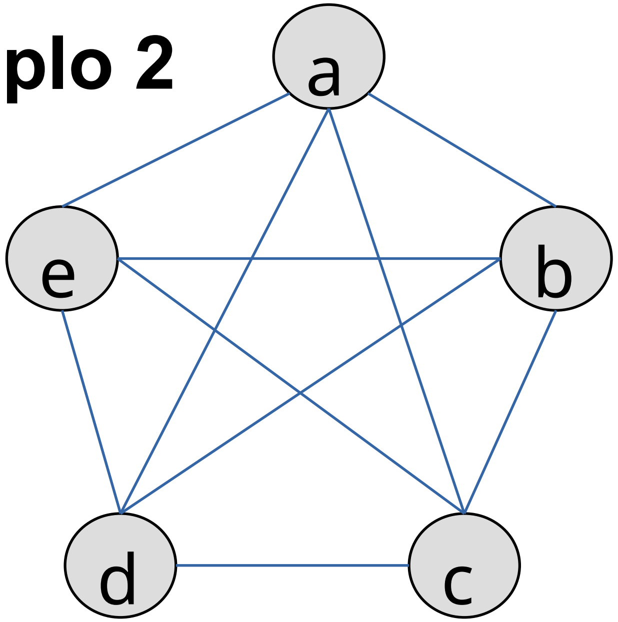
Iniciando no vértice 1
1 – 2 – 3 – 4
Custo = 10004

10004/8 – fator de aproximação

O fator de aproximação pode ser arbitrariamente ruim.
O fator de aproximação tende a infinito.

Algoritmos Aproximativos

Exemplo 2



c_{ij}	a	b	c	d	e
a	-	1	2,2	2	4,1
b	1	-	1,4	2,2	4
c	2,2	1,4	-	2,2	3,2
d	2	2,2	2,2	-	2,2
e	4,1	4	3,2	2,2	-

Algoritmos Aproximativos

Vizinho mais Próximo

Complexidade: $O(n^2)$

A solução gerada depende muito do vértice inicial.

Aplicar a heurística começando em cada vértice e ficar com a melhor solução: $O(n^3)$

Algoritmos Aproximativos

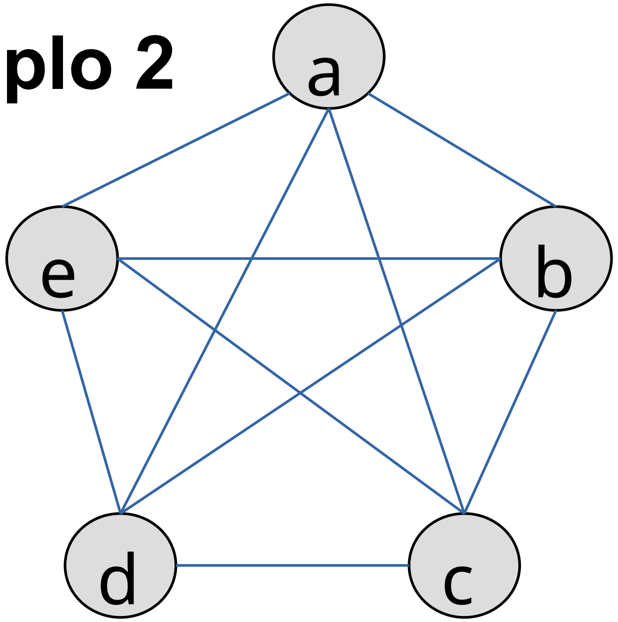
Inserção mais barata

1. **Escolher** um vértice inicial
2. **Iniciar** uma rota com 3 vértices (guloso)
3. **Inserir** uma cidade **k** entre as cidades **i** e **j** tal que esta inserção minimize: $s_{ij} = c_{ik} + c_{kj} - c_{ij}$

4. **Repita** o passo 3 até que todas as cidades sejam visitadas.

Algoritmos Aproximativos

Exemplo 2

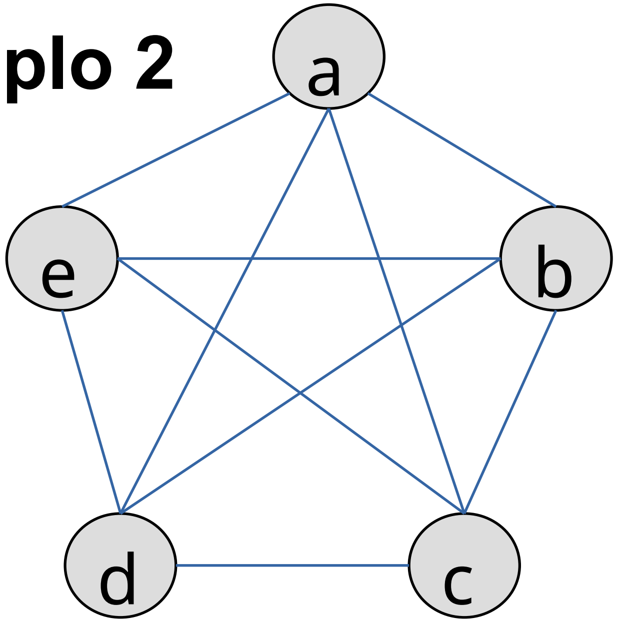


c_{ij}	a	b	c	d	e
a	-	1	2,2	2	4,1
b	1	-	1,4	2,2	4
c	2,2	1,4	-	2,2	3,2
d	2	2,2	2,2	-	2,2
e	4,1	4	3,2	2,2	-

Inserir uma cidade **k** entre as cidades **i** e **j** tal que esta inserção minimize: $s_{ij} = c_{ik} + c_{kj} - c_{ij}$

Algoritmos Aproximativos

Exemplo 2



c_{ij}	**a**	**b**	**c**	**d**	**e**
a	-	1	2,2	2	4,1
b	1	-	1,4	2,2	4
c	2,2	1,4	-	2,2	3,2
d	2	2,2	2,2	-	2,2
e	4,1	4	3,2	2,2	-

Inserir uma cidade **k** entre as cidades **i** e **j** tal que esta inserção minimize: $s_{ij} = c_{ik} + c_{kj} - c_{ij}$