Projeto e Análise de Algoritmos

Parte 6: Problemas NP-Completos

Eduardo Freire Nakamura
nakamura@dcc.ufam.edu.br

Departamento de Ciência da Computação (DCC)
Instituto de Ciências Exatas (ICE)
Universidade Federal do Amazonas (UFAM)

Introdução

- Problemas estudados até agora
  - Algoritmos polinomiais
  - Entrada de tamanho *n*
  - Tempo é no pior caso $O(n^k)$

- Todo problema pode ser resolvido em tempo polinomial?

- E se dispormos de tempo infinito?

O problema da parada

"Dada uma descrição de um programa e uma entrada finita, decida se o programa termina de rodar ou rodará indefinidamente, dada essa entrada"

Alan Turing (1912-1941)
provou em 1936 que não existe nenhum algoritmo que resolva este problema

Classes de problemas

- Classe de problemas NP-Difícil
  - Não existe solução
- Classe de problemas NP
  - Existe solução não polinomial conhecida
  - Toda solução pode ser verificada em tempo polinomial

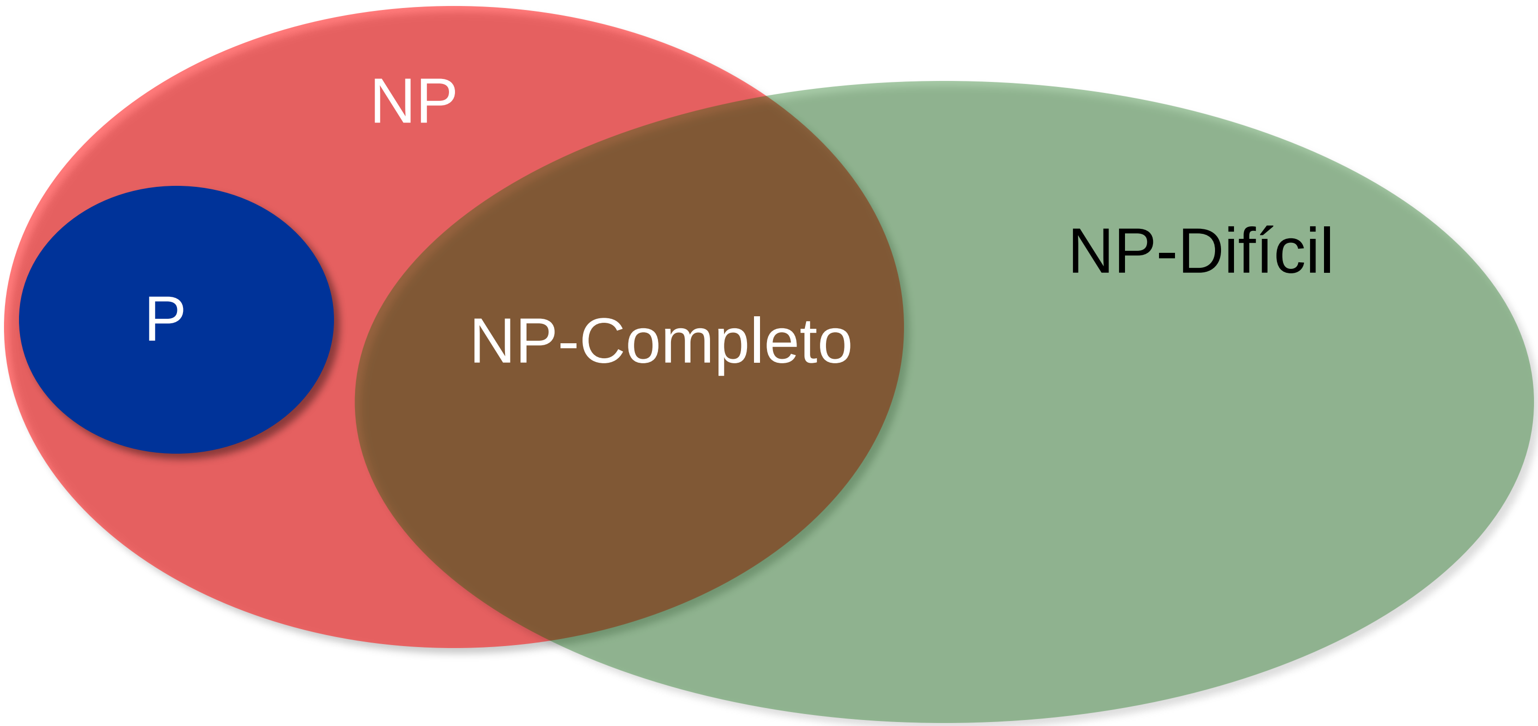
Classes de problemas

- Classe de problemas P
  - Existe solução em tempo polinomial conhecida

- Classe de problemas NP-Completo
  - Existe soluções conhecidas
  - Entretanto...
    - Até hoje ninguém encontrou um algoritmo polinomial
    - Até hoje ninguém provou que tal algoritmo não existe

Classe de problemas

Suposto relacionamento entre classes: P ≠ NP (não há provas)

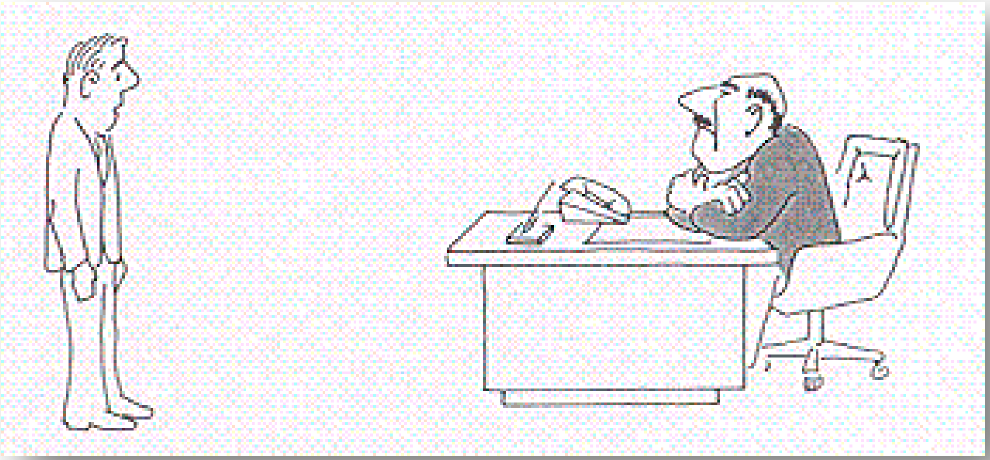


P = NP ou P ≠ NP?

- *Millennium Prize Problems* (www.claymath.org/millennium)
  - Dez problemas clássicos
  - US$ 1 milhão para quem resolver

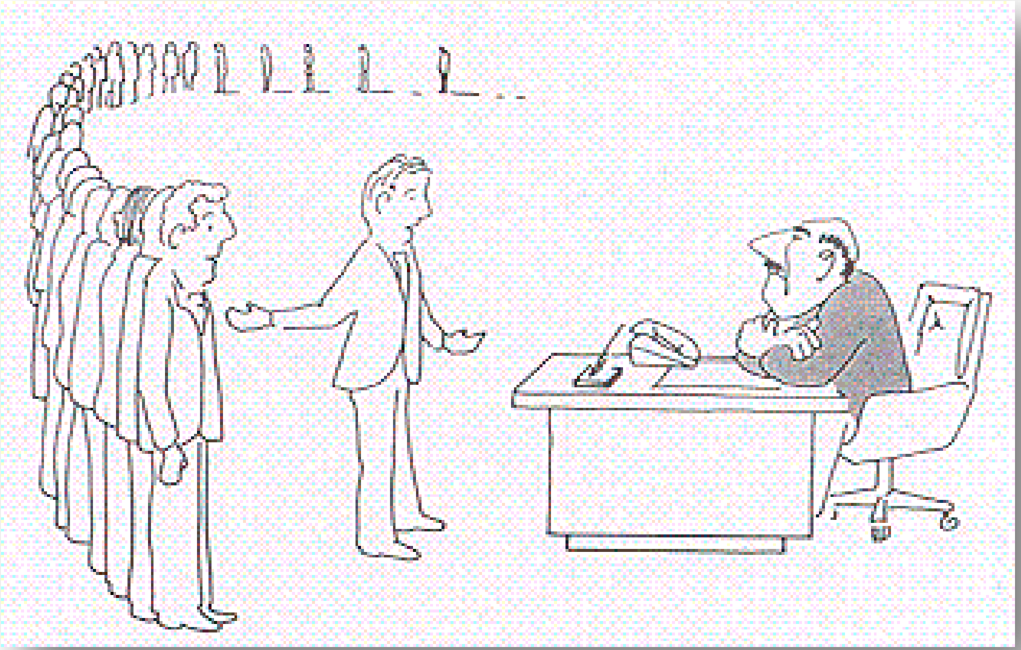
- Somente um foi resolvido
  - Conjectura de Poincaré (1904)
  - 2002 (prova)
  - 2006 (Medalha Fields)
  - 2010 (Millenium Prize)
  - Dr. Grigori Yakovlevich Perelman

Por que é importante?



Eu não encontro nenhuma solução eficiente,
acho que sou meio limitado...

Por que é importante?



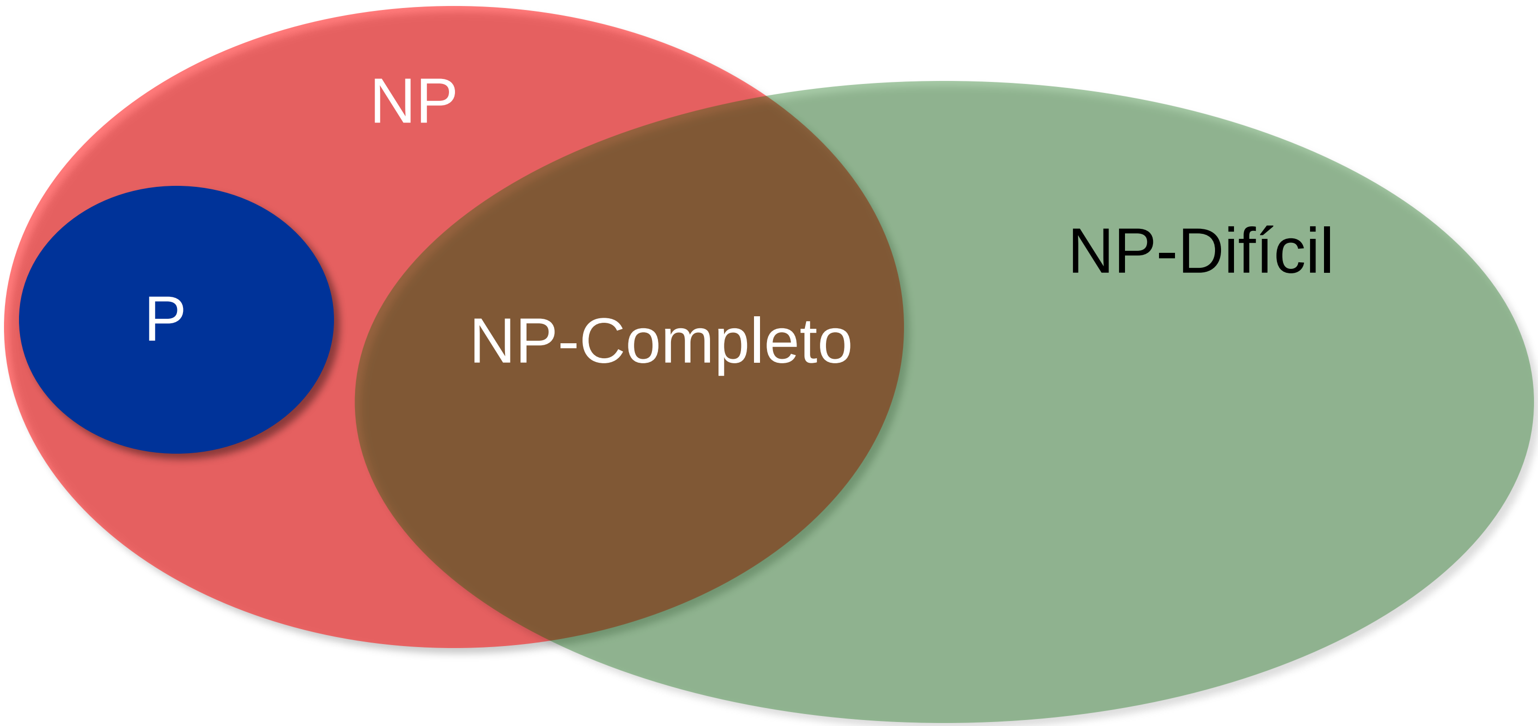
Eu não encontro nenhuma solução eficiente,
mas nenhum destes cientistas famosos consegue também...

Um aspecto intrigante

- Vários problemas NP-completos são muito semelhantes a problemas polinomiais, diferindo em detalhes
- Um exemplo
  - Caminho mínimo: algoritmos polinomiais $O(V \times E)$
  - Caminho máximo: NP-Completo (mesmo se todos os pesos das arestas for 1)

Como provar NP-completo?

- Veja o que é NP-Completo:



Como provar NP-completo?

- Para provar que um problema **X** é NP-completo é necessário seguir os seguintes passos
  - Mostre que **X** está na classe NP
    - Para isso, apresente um algoritmo que verifique em tempo polinomial se uma solução é válida
  - Mostre que **X** está na classe NP-Difícil
    - Para isso, apresente uma redução polinomial que transforme instâncias de um problema **Y** (conhecidamente NP-Difícil) no problema **X**

O problema base NP-completo

- Satisfabilidade (SAT)

- Dada uma expressão booleana
  - Conectivos AND ($\wedge$), OR ($\vee$), NOT ($\neg$)
  - Variáveis
  - Parênteses

$$(a_1 \vee \neg a_2) \wedge (a_1 \vee a_2 \vee a_3) \wedge (\neg a_1 \vee a_3) \wedge ....$$

- Pergunta-se
  - Existe alguma atribuição de valores a estas variáveis que torne a expressão VERDADEIRA?

Uma variação comum

- 3SAT
  - Satisfabilidade de 3 literais por grupo
  - Continua sendo NP-Completo

- Disjunção de conjunções ou conjunção de disjunções, agrupadas 3 a 3

$$(a_1 \vee \neg a_4 \vee a_3) \wedge (a_1 \vee a_2 \vee a_3) \wedge (\neg a_1 \vee a_2 \vee a_3) \wedge \ldots.$$

Satisfabilidade

- Dada uma atribuição de valores às variáveis
  - Podemos verificar em tempo polinomial se ela é verdadeira
  - Basta substituir os valores das variáveis e executar as operações lógicas

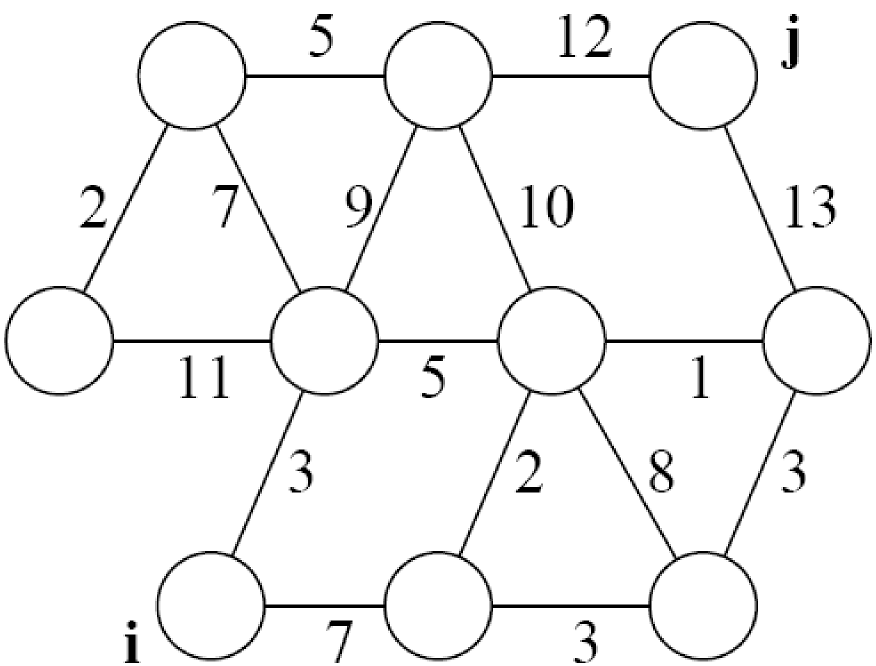
- Algoritmo
  - Existe apenas um algoritmo
  - Testar todas as possibilidades de atribuições de valores para todas as variáveis e verificar todas a procura de uma que satisfaça a expressão booleana
  - Qual é o custo deste algoritmo?

Problema do Caixeiro Viajante

- Dados
  - Constante *k*, conjunto de cidades $C = \{c_1, c_2, ..., c_n\}$ e uma distância $d(c_i; c_j)$ para cada par de cidades $c_i, c_j \in C$
- Questão
  - Existe um “roteiro” para todas as cidades em *C* cujo comprimento total seja menor ou igual a *k*?

Caminho em um grafo

- Considere um grafo com peso nas arestas, dois vértices *i*, *j* e um inteiro *k* > 0

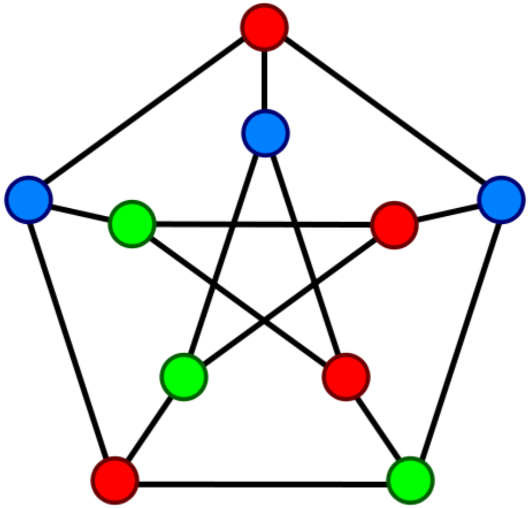


- Fácil
  - Existe um caminho de i até j com peso ≤ k?
  - Dijkstra
- Difícil
  - Existe um caminho de i até j com peso ≥ k?

Coloração de um Grafo

- Em um grafo G = (V, E), mapear c : V → S, sendo S um conjunto finito de cores tal que se <v,w> $\in$ E então c(v) $\neq$ c(w) (vértices adjacentes possuem cores distintas)

- Dados G e um inteiro positivo k, existe uma coloração de G usando k cores?
  - Fácil k = 2
  - Difícil k > 2



Coloração de um Grafo - Horários

- Em uma aplicação modelada como um problema de coloração de vértices, os vértices de mesma cor representam indivíduos que não conflitam entre si

- Atribuição de freqüências de rádio
- Separação de produtos explosivos
- Agendamento de cursos na universidade

Atribuição de frequencias de rádio

- Os vértices representam as estações de rádio
- Duas estações são vizinhas quando suas áreas de transmissão se sobrepõem, resultado em interferência se elas usassem a mesma freqüência
- Cada cor contém estações que podem receber a mesma freqüência

Separação de produtos explosivos

- Os vértices são produtos químicos necessários em algum processo de produção
- Existe uma aresta ligando cada par de produtos que podem explodir, se colocados lado a lado
- O número mínimo de cores representa o número mínimo de compartimentos para guardar estes produtos em segurança

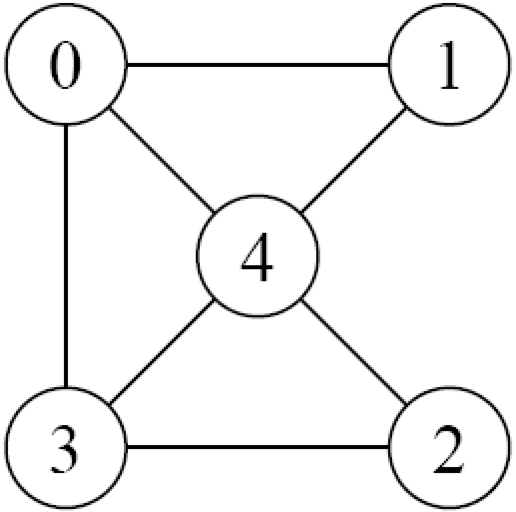
Agendamento de horário

- Os vértices representam os cursos de uma universidade
- Dois cursos são adjacentes se um aluno se matricula para ambos os cursos
- O número mínimo de cores representa o número mínimo de horários necessários para acomodar os cursos

Ciclo de Hamilton

- Ciclo de Hamilton
  - Ciclo que passa por todos os vértices uma única vez

- Caminho de Hamilton
  - Caminho que passa por todos os vértices uma única vez



- Exemplo de ciclo de Hamilton
  - 0 1 4 2 3 0

- Exemplo de caminho de Hamilton:
  - 0 1 4 2 3

Ciclo de Hamilton

- Existe um ciclo de Hamilton no grafo G?

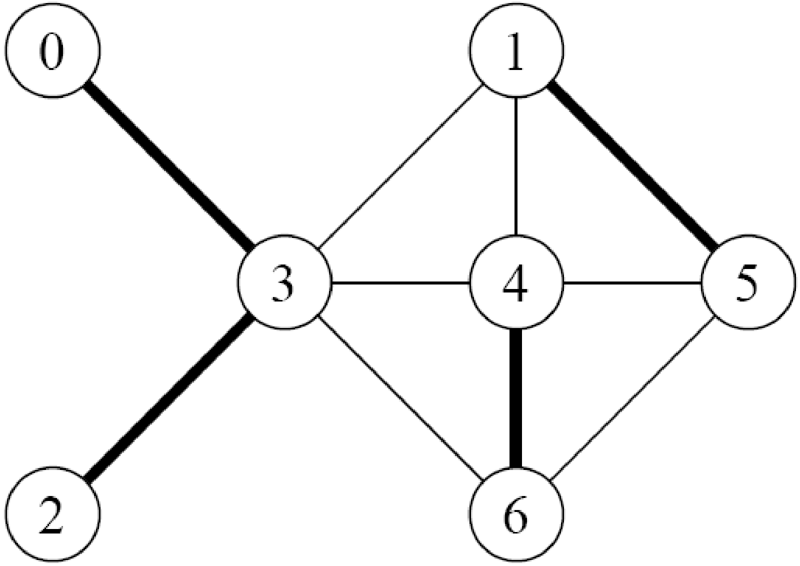
- Fácil
  - Grafos com grau máximo ≤ 2 (vértices com no máximo duas arestas incidentes)

- Difícil
  - Grafos com grau > 2

- Caso especial do PCV
  - Pares de vértices com uma aresta entre eles tem distância 1 e pares de vértices sem aresta entre eles têm distância infinita

Cobertura de arestas e vértices

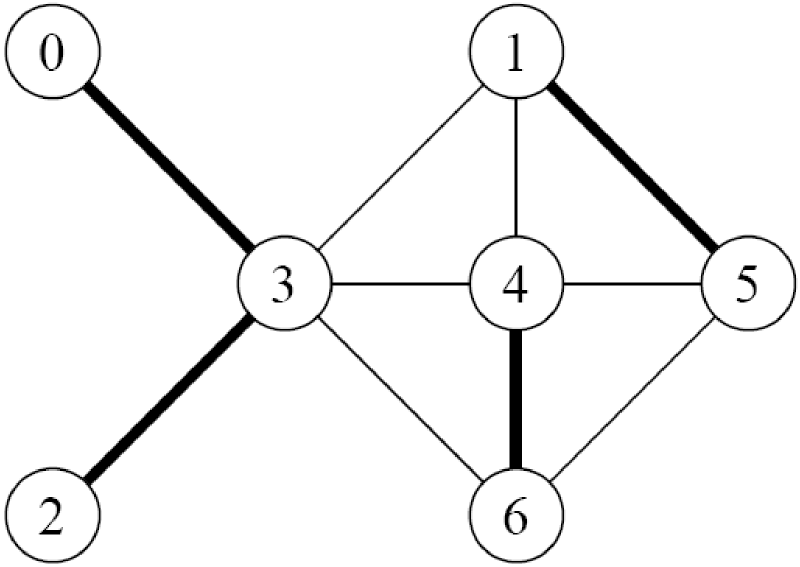
- Uma cobertura de arestas de um grafo G = (V, E) é um subconjunto E’ $\subset$ E de k arestas tal que todo v $\in$ V é parte de pelo menos uma aresta de E



- Uma resposta para k=4
  - E’ = {<0,3>; <2,3>; <1,5>; <4,6>}

Cobertura de arestas e vértices

- Uma cobertura de vértices é um subconjunto V’ ∈ V tal que se <u,v> ∈ E, então u ∈ V’ ou v ∈ V’, isto é, cada aresta do grafo é incidente em um dos vértices de V’

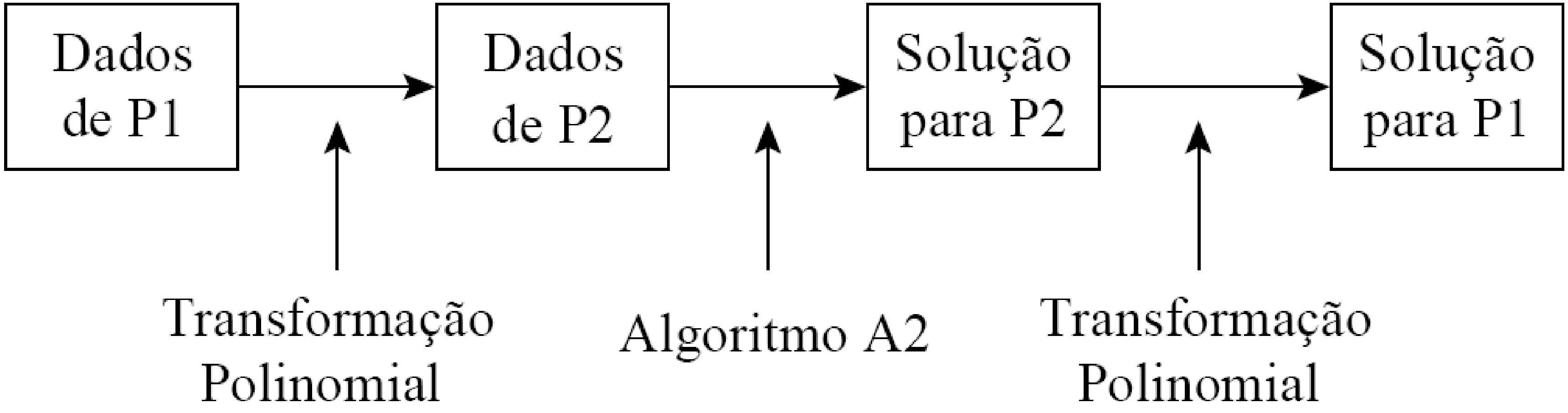


- Resposta para o exemplo
  - V’ = {3, 4, 5} , para k = 3
- Fácil
  - ∃ uma cobertura de arestas ≤ k?
- Difícil
  - ∃ uma cobertura de vétices ≤ k?

Transformação polinomial

- Sejam P_1 e P_2 dois problemas de decisão
  - Suponha que um algoritmo A_2 resolva P_2
- Se for possível transformar P_1 em P_2 e a solução de P_2 em solução de P_1
  - Então A_2 pode ser utilizado para resolver P_1
- Se pudermos realizar as transformações nos dois sentidos em tempo polinomial
  - Então P_1 é polinomialmente transformável em P_2

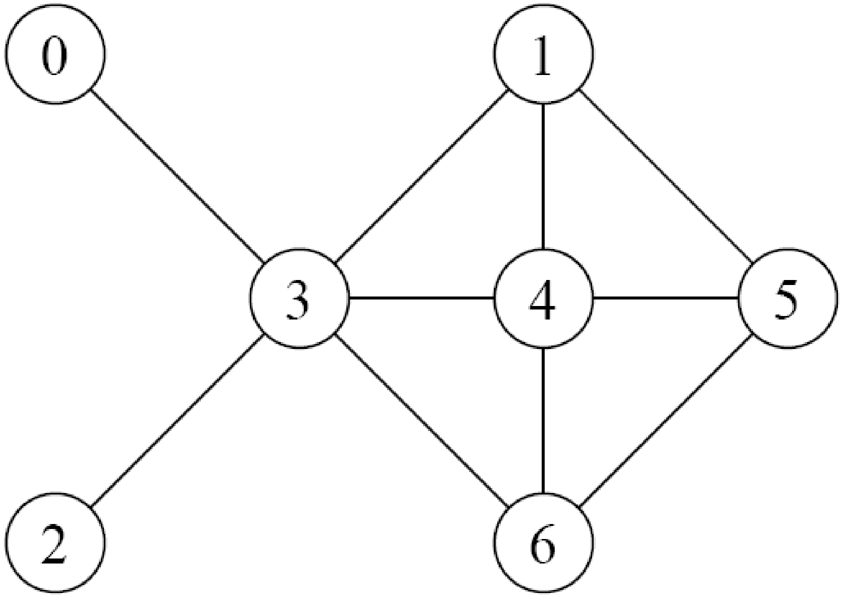
Transformação polinomial



- Para mostrar um exemplo de transformação polinomial, definiremos clique de um grafo e conjunto independente de vértices de um grafo

Conjunto independente de vértices

- O conjunto independente de vértices de um grafo G=(V,E) é constituído do subconjunto V’ $\in$ V , tal que v, w $\in$ V $\Rightarrow$ $\langle v,w \rangle \notin$ E



- Todo par de vértices de V’ é não adjacente
  - V’ é um subgrafo totalmente desconectado

Aplicações

- Em problemas de dispersão é necessário encontrar grandes conjuntos independentes de vértices
  - Procura-se um conjunto de pontos mutuamente separados

- Exemplo
  - Identificar localizações para instalação de franquias
  - Duas localizações não podem estar perto o suficiente para competirem entre si

Aplicações

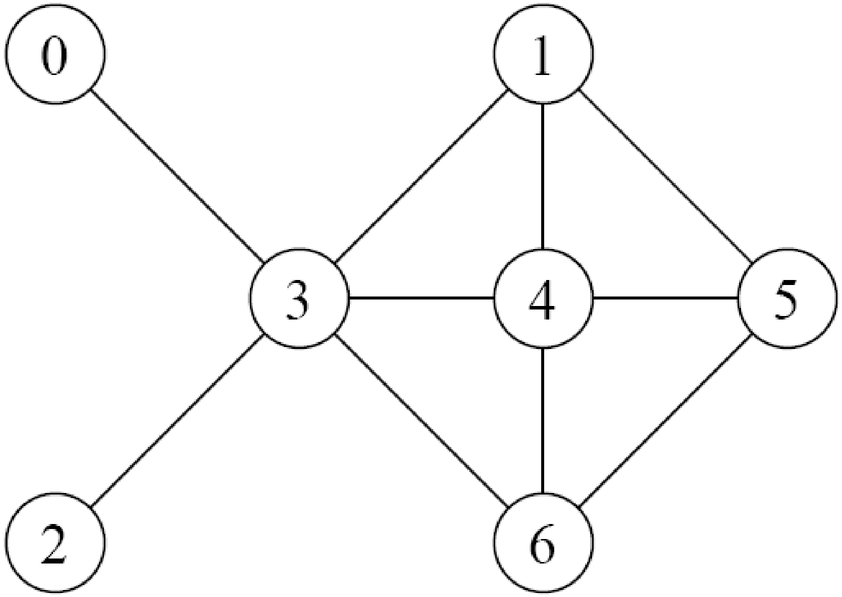
- Solução
  - O maior conjunto independente fornece o maior número de franquias que podem ser concedidas sem prejudicar as vendas

- Em geral, conjuntos independentes evitam conflitos entre elementos

Clique de um grafo

- Clique de um grafo G = (V,E) é constituído do subconjunto V' $\in$ V , tal que v, w $\in$ V' $\Rightarrow$ <v,w> $\in$ E

- Todo par de vértices de V' é adjacente (V' é um subgrafo completo)



- Exemplo de cardinalidade 3
  - V'={1, 3, 4}

Aplicação

- Identificar agrupamentos de objetos relacionados
  - Encontrar grandes cliques em grafos

- Exemplo
  - Empresa de fabricação de peças por meio de injeção plástica que fornece para diversas outras empresas montadoras
  - É preciso identificar os clientes que adquirem os mesmos produtos, para negociar prazos de entrega comuns e assim aumentar o tamanho dos lotes produzidos

Aplicação

- Exemplo (cont.)
  - Construir um grafo com cada vértice representando um cliente e ligar com uma aresta os que adquirem os mesmos produtos
  - Um clique no grafo representa o conjunto de clientes que adquirem os mesmos produtos

A transformação polinomial

- Considere P_1 o problema clique e P_2 o problema conjunto independente de vértices

- Transformação polinomial
  1. Obter o grafo complementar G'
  2. G possui um clique de tamanho $\geq$ k se, e somente se, G' possui um conjunto independente de vértices $\geq$ k

A transformação polinomial

- Se existe um algoritmo polinomial que resolva o conjunto independente, ele pode ser utilizado para resolver o clique

- Clique é redutível ao Conjunto independente de vétices

Outro exemplo

- Transformação polinomial do Caixeiro Viajante para o ciclo de Hamilton
  1. Para as cidades, use os vértices
  2. Para as distâncias, use 1 se existir um arco no grafo original e 2 se não existir

- Use o PCV para achar um roteiro ≤ V
  - O roteiro é o ciclo de Hamilton

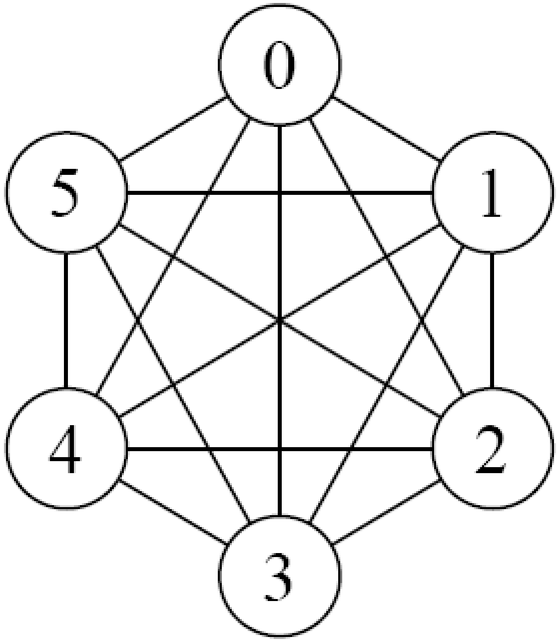
Heurísticas

- Algoritmo que pode produzir um bom resultado (ou até a solução ótima)
  - Pode também não obter solução ou obter uma distante da ótima
- Pode haver instâncias em que uma heurística nunca vai encontrar uma solução

Heurística para o PCV

- Algoritmo do vizinho mais próximo, heurística gulosa simples
  1. Inicie com um vértice arbitrário
  2. Procure o vértice mais próximo do último vértice adicionado que não esteja no caminho e adicione ao caminho a aresta que liga esses dois vértices
  3. Quando todos os vértices estiverem no caminho, adicione uma aresta conectando o vértice inicial e o último vértice adicionado

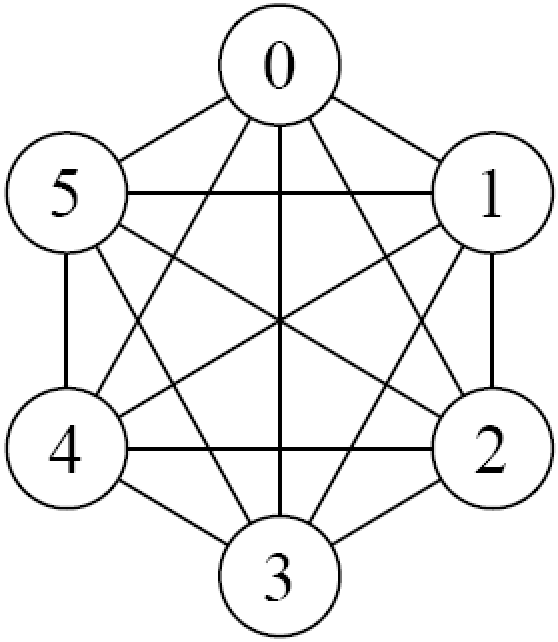
Heurística para o PCV



	0	1	2	3	4	5
0	x	3	10	11	7	25
1	3	x	8	12	9	26
2	10	8	x	9	4	20
3	11	12	9	x	5	15
4	7	9	4	5	x	18
5	25	26	20	15	18	x

Caminho ótimo: 0-1-2-5-3-4-0
Custo: 3+8+20+15+5+7 = 58

Heurística para o PCV

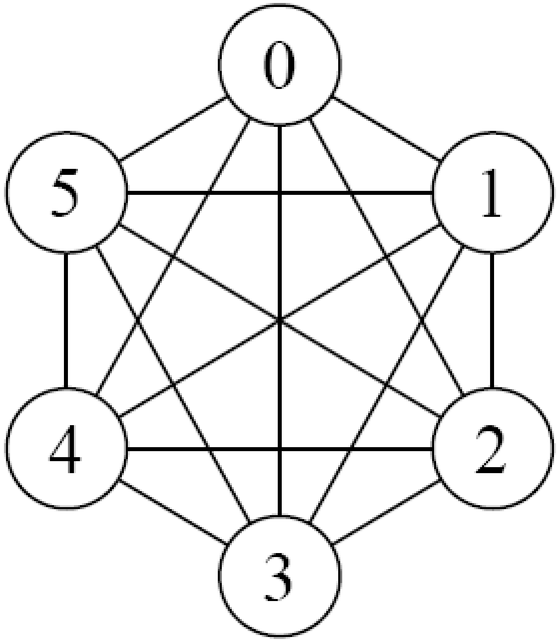


	0	1	2	3	4	5
0	x	3	10	11	7	25
1	3	x	8	12	9	26
2	10	8	x	9	4	20
3	11	12	9	x	5	15
4	7	9	4	5	x	18
5	25	26	20	15	18	x

Caminho vizinho: 0
Custo:

Caminho ótimo: 0-1-2-5-3-4-0
Custo: 3+8+20+15+5+7 = 58

Heurística para o PCV

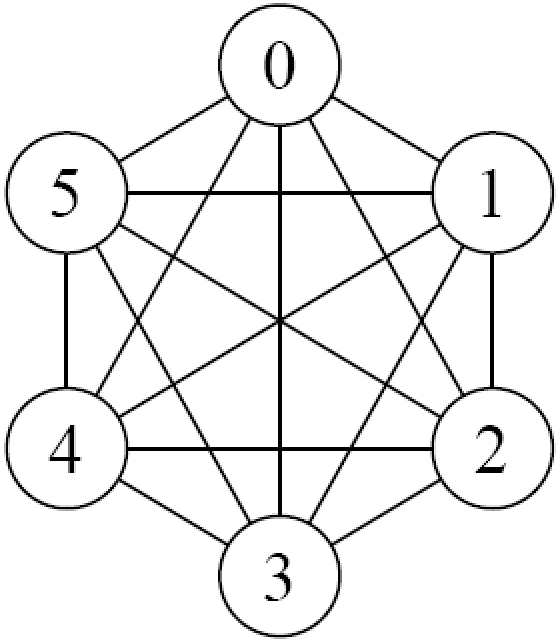


	0	1	2	3	4	5
0	x	3	10	11	7	25
1	3	x	8	12	9	26
2	10	8	x	9	4	20
3	11	12	9	x	5	15
4	7	9	4	5	x	18
5	25	26	20	15	18	x

Caminho vizinho: 0-1
Custo: 3

Caminho ótimo: 0-1-2-5-3-4-0
Custo: 3+8+20+15+5+7 = 58

Heurística para o PCV

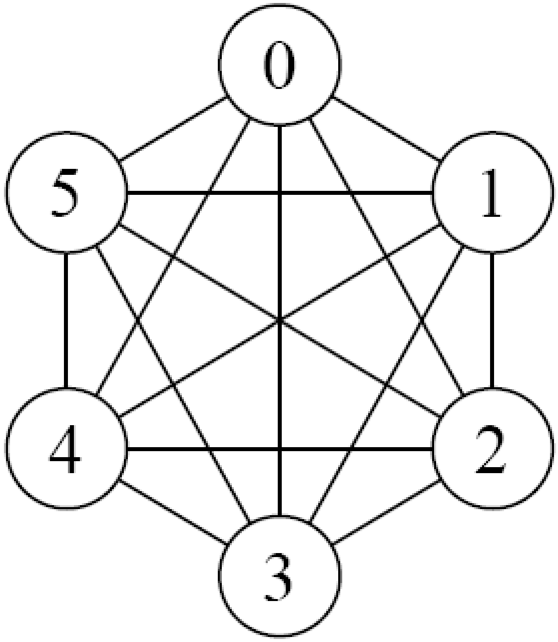


	0	1	2	3	4	5
0	x	3	10	11	7	25
1	3	x	8	12	9	26
2	10	8	x	9	4	20
3	11	12	9	x	5	15
4	7	9	4	5	x	18
5	25	26	20	15	18	x

Caminho vizinho: 0-1
Custo: 3

Caminho ótimo: 0-1-2-5-3-4-0
Custo: 3+8+20+15+5+7 = 58

Heurística para o PCV

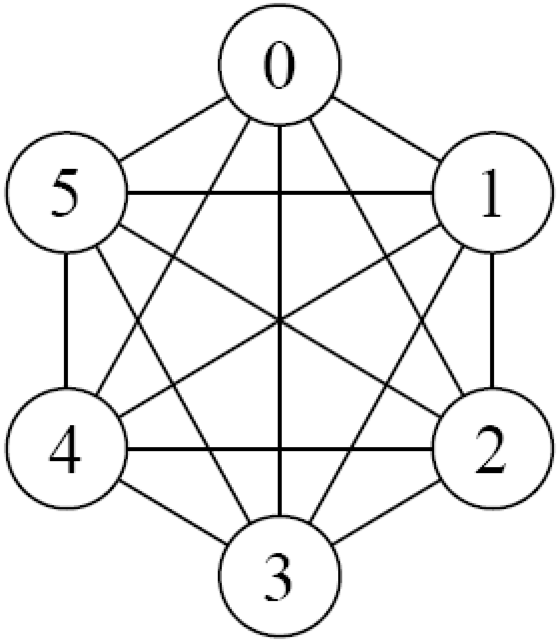


	0	1	2	3	4	5
0	x	3	10	11	7	25
1	3	x	8	12	9	26
2	10	8	x	9	4	20
3	11	12	9	x	5	15
4	7	9	4	5	x	18
5	25	26	20	15	18	x

Caminho vizinho: 0-1-2
Custo: 3+8

Caminho ótimo: 0-1-2-5-3-4-0
Custo: 3+8+20+15+5+7 = 58

Heurística para o PCV

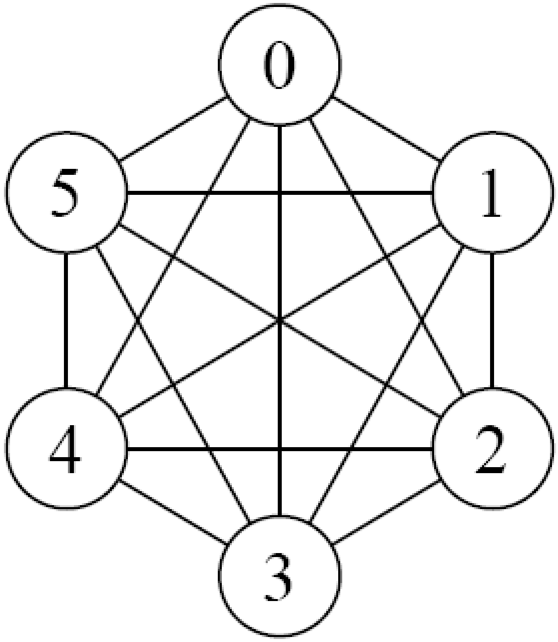


	0	1	2	3	4	5
0	x	3	10	11	7	25
1	3	x	8	12	9	26
2	10	8	x	9	4	20
3	11	12	9	x	5	15
4	7	9	4	5	x	18
5	25	26	20	15	18	x

Caminho vizinho: 0-1-2
Custo: 3+8

Caminho ótimo: 0-1-2-5-3-4-0
Custo: 3+8+20+15+5+7 = 58

Heurística para o PCV

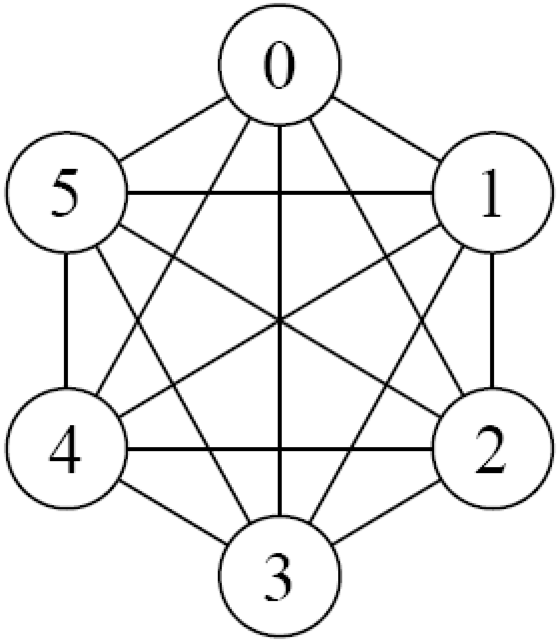


	0	1	2	3	4	5
0	x	3	10	11	7	25
1	3	x	8	12	9	26
2	10	8	x	9	4	20
3	11	12	9	x	5	15
4	7	9	4	5	x	18
5	25	26	20	15	18	x

Caminho vizinho: 0-1-2-4
Custo: 3+8+4

Caminho ótimo: 0-1-2-5-3-4-0
Custo: 3+8+20+15+5+7 = 58

Heurística para o PCV

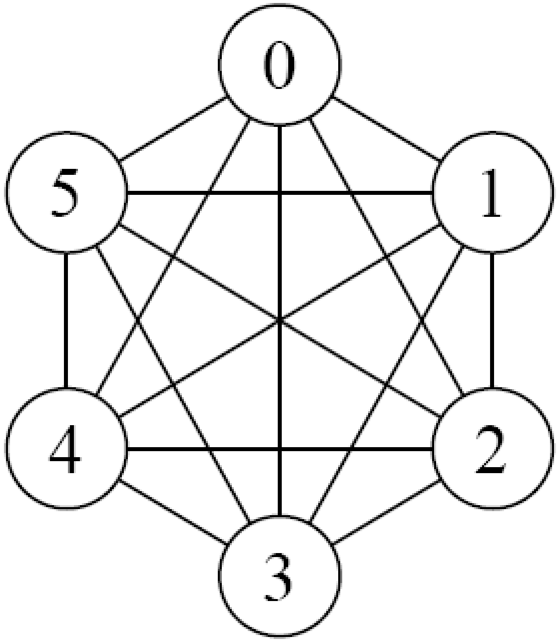


	0	1	2	3	4	5
0	x	3	10	11	7	25
1	3	x	8	12	9	26
2	10	8	x	9	4	20
3	11	12	9	x	5	15
4	7	9	4	5	x	18
5	25	26	20	15	18	x

Caminho vizinho: 0-1-2-4
Custo: 3+8+4

Caminho ótimo: 0-1-2-5-3-4-0
Custo: 3+8+20+15+5+7 = 58

Heurística para o PCV

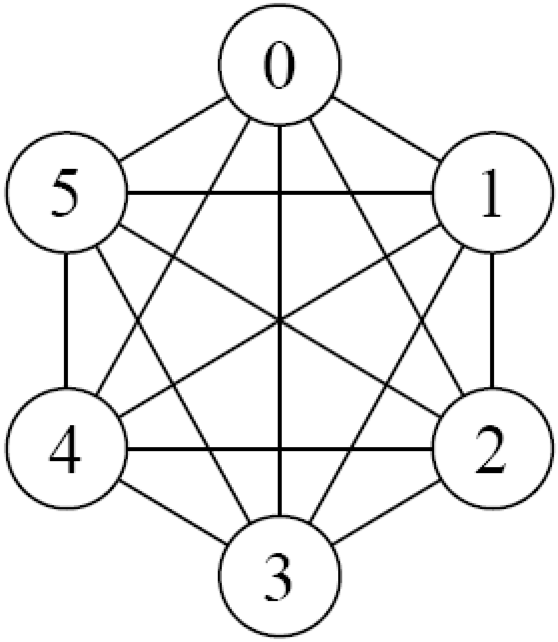


	0	1	2	3	4	5
0	x	3	10	11	7	25
1	3	x	8	12	9	26
2	10	8	x	9	4	20
3	11	12	9	x	5	15
4	7	9	4	5	x	18
5	25	26	20	15	18	x

Caminho vizinho: 0-1-2-4-3
Custo: 3+8+4+5

Caminho ótimo: 0-1-2-5-3-4-0
Custo: 3+8+20+15+5+7 = 58

Heurística para o PCV

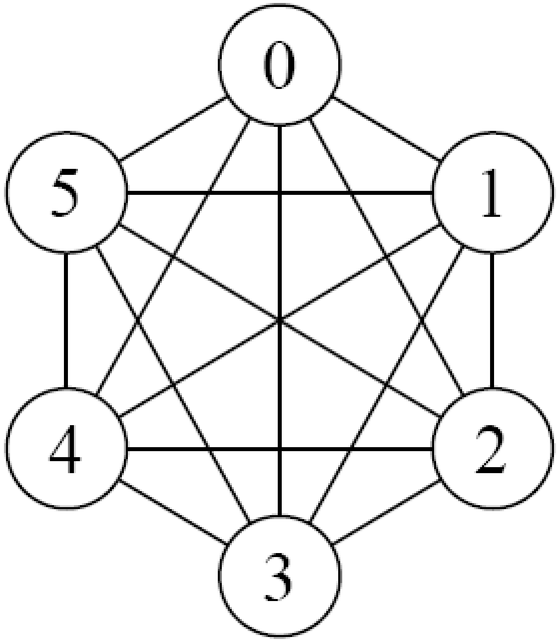


	0	1	2	3	4	5
0	x	3	10	11	7	25
1	3	x	8	12	9	26
2	10	8	x	9	4	20
3	11	12	9	x	5	15
4	7	9	4	5	x	18
5	25	26	20	15	18	x

Caminho vizinho: 0-1-2-4-3
Custo: 3+8+4+5

Caminho ótimo: 0-1-2-5-3-4-0
Custo: 3+8+20+15+5+7 = 58

Heurística para o PCV

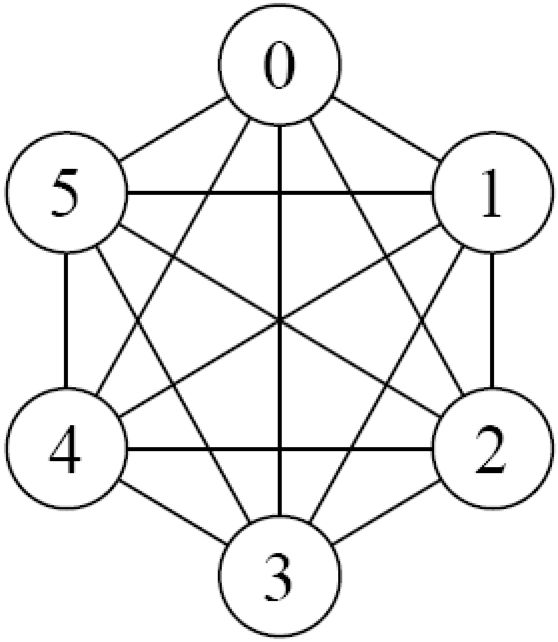


	0	1	2	3	4	5
0	x	3	10	11	7	25
1	3	x	8	12	9	26
2	10	8	x	9	4	20
3	11	12	9	x	5	15
4	7	9	4	5	x	18
5	25	26	20	15	18	x

Caminho vizinho: 0-1-2-4-3-5
Custo: 3+8+4+5+15

Caminho ótimo: 0-1-2-5-3-4-0
Custo: 3+8+20+15+5+7 = 58

Heurística para o PCV

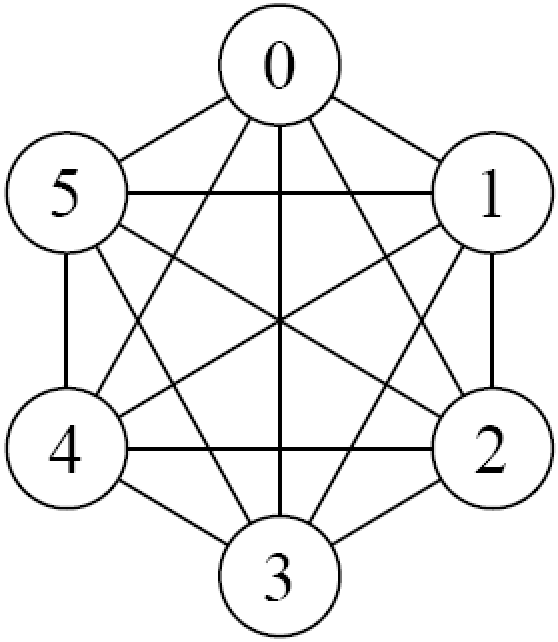


	0	1	2	3	4	5
0	x	3	10	11	7	25
1	3	x	8	12	9	26
2	10	8	x	9	4	20
3	11	12	9	x	5	15
4	7	9	4	5	x	18
5	25	26	20	15	18	x

Caminho vizinho: 0-1-2-4-3-5
Custo: 3+8+4+5+15

Caminho ótimo: 0-1-2-5-3-4-0
Custo: 3+8+20+15+5+7 = 58

Heurística para o PCV



	0	1	2	3	4	5
0	x	3	10	11	7	25
1	3	x	8	12	9	26
2	10	8	x	9	4	20
3	11	12	9	x	5	15
4	7	9	4	5	x	18
5	25	26	20	15	18	x

Caminho vizinho: 0-1-2-4-3-5-0
Custo: 3+8+4+5+15+25=60

Caminho ótimo: 0-1-2-5-3-4-0
Custo: 3+8+20+15+5+7 = 58

Análise

Qual é o custo do vizinho mais próximo?

Exercício

- Escreva a heurística do vizinho mais próximo e faça sua análise de complexidade

- Escreva uma heurística para o problema do ciclo de Hamilton, baseada no vizinho mais próximo e faça sua análise