

Algoritmos Aproximados para Problemas NP Completos

Introdução

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

Problemas Exponenciais

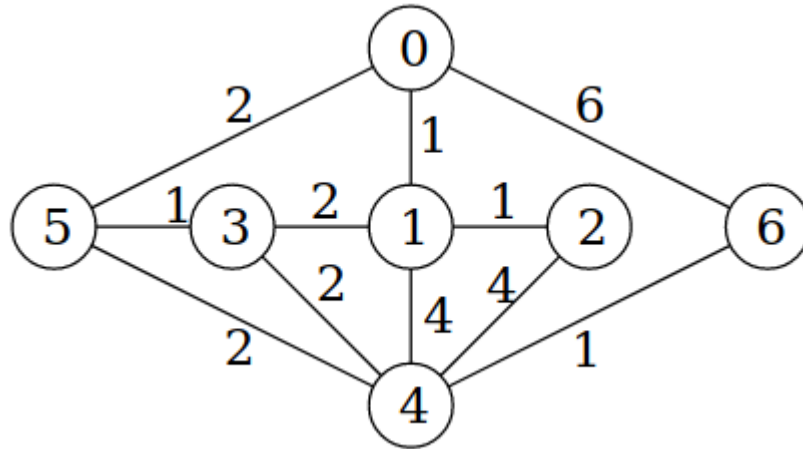
- É desejável resolver instâncias grandes de problemas de otimização em tempo razoável.
 - Para um algoritmo que execute em tempo proporcional a 2^N , não é garantido obter resposta para todos os problemas de tamanho $N \geq 100$.

O que fazer?

- Usar algoritmos exponenciais “eficientes” aplicando técnicas de tentativa e erro.
- Usar algoritmos aproximados.
 - Acham uma resposta que pode não ser a solução ótima, mas é garantido ser próxima dela.
- Concentrar no caso médio.
 - Buscar algoritmos melhores que outros neste quesito e que funcionem bem para as entradas de dados que ocorrem usualmente na prática.

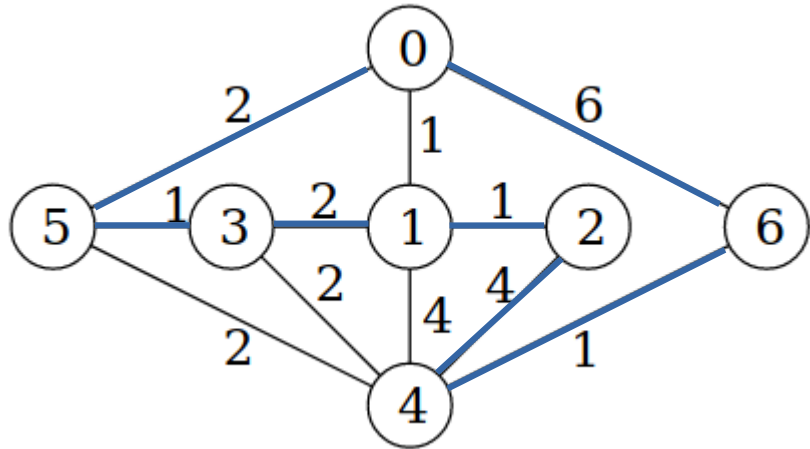
Ciclo Hamiltoniano

- Caminho que permite passar por todos os vértices de um grafo G , não repetindo nenhum.



Ciclo Hamiltoniano

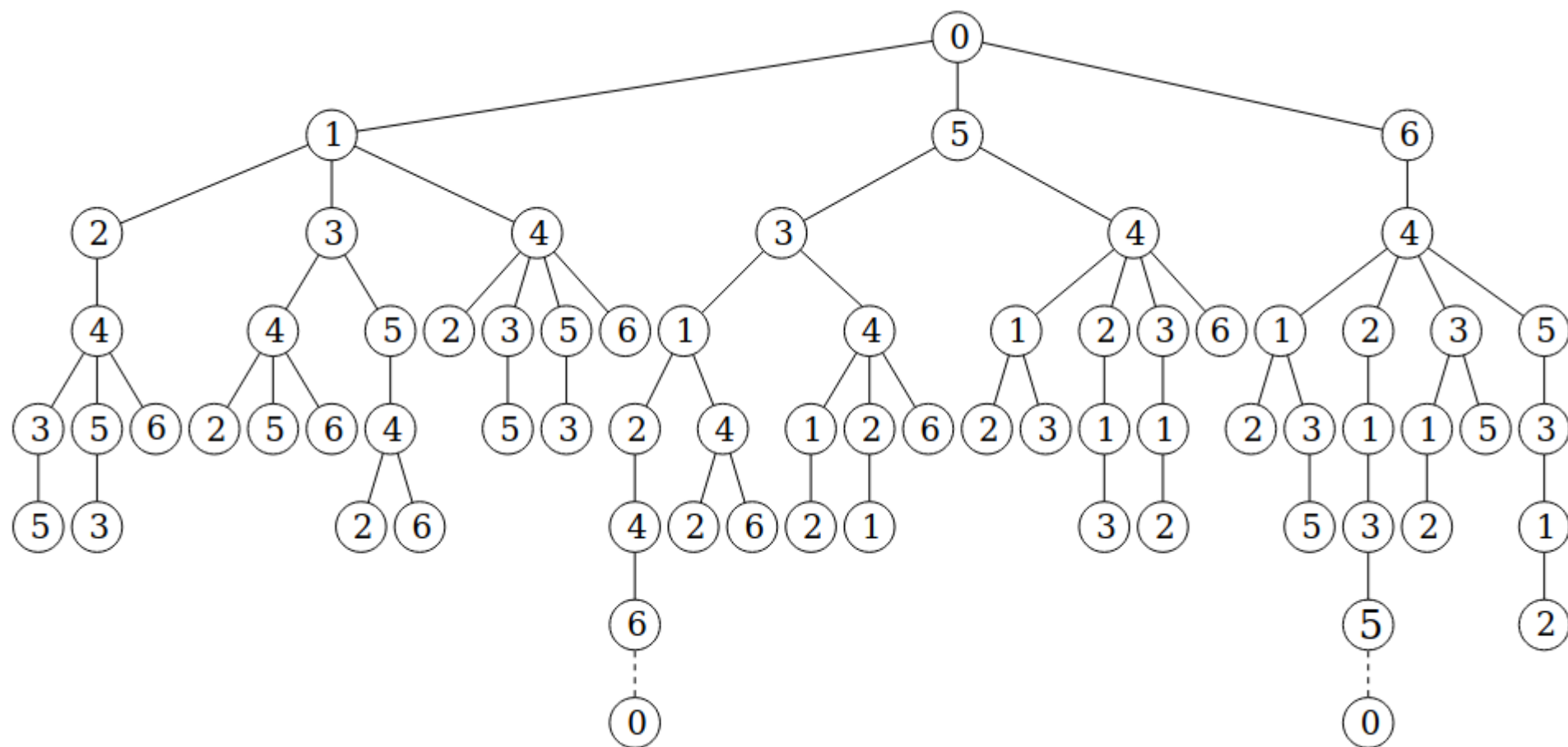
- A rigor, o melhor algoritmo conhecido resolve o problema tentando todos os caminhos possíveis.



Existem duas respostas:

0 5 3 1 2 4 6 0
0 6 4 2 1 3 5 0

Ciclo Hamiltoniano

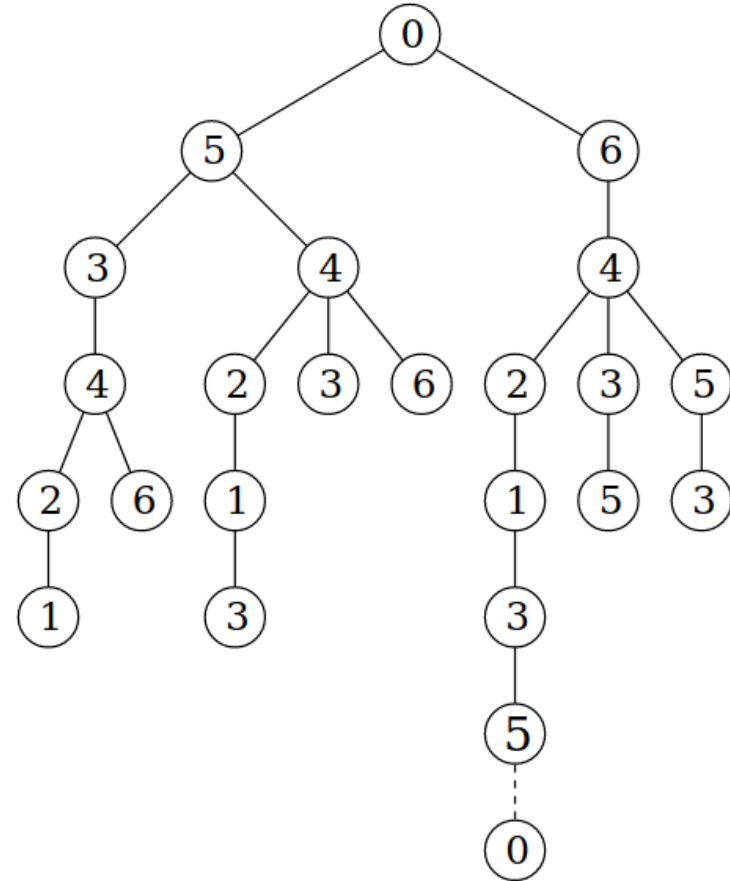
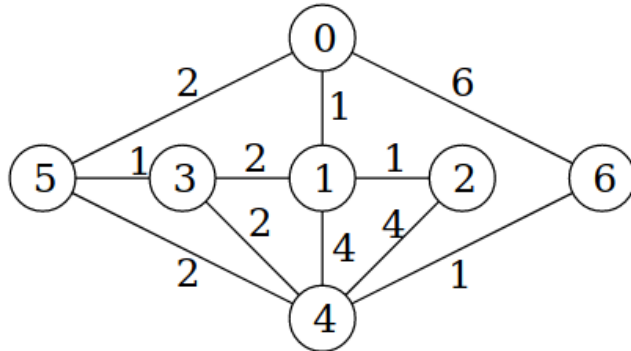


Ciclo Hamiltoniano

- Diminuir número de chamadas fazendo “**poda**” na árvore de caminhamento.
 - No exemplo anterior, cada ciclo é obtido duas vezes, caminhando em ambas as direções.

Ciclo Hamiltoniano

- Insistindo que o nó 2 apareça antes do 0 e do 1, não precisamos chamar Visita para o nó 1 a não ser que o nó 2 já esteja no caminho.

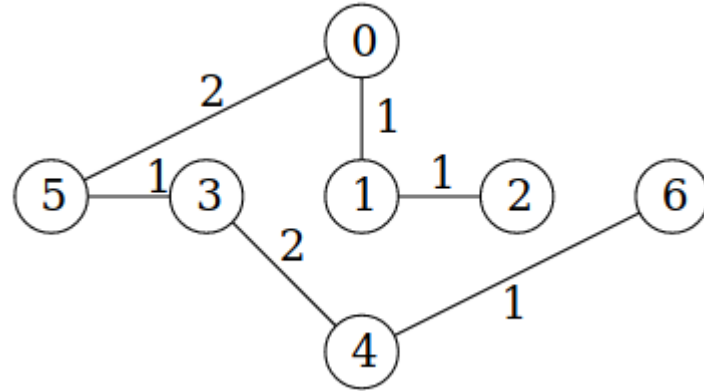
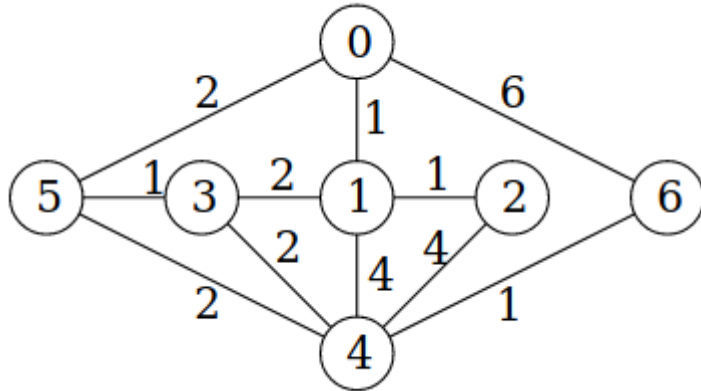


Algoritmo Aproximado

- Não precisa encontrar a solução ótima
- Necessário monitorar a solução encontrada
 - Razão de aproximação $R_A(I) = \frac{S(I)}{S^*(I)}$,
 - Limite inferior e superior

Exemplo

- Problema do Caixeiro Viajante (PCV)
- Limite inferior?
 - Árvore geradora mínima (AGM)



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 $\text{dist}[x] + c_{xy}$,
 - acrescente o arco x-y e o vértice y a T
 - faça $\text{dist}[y] = \text{dist}[x] + c_{xy}$.
- 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

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

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

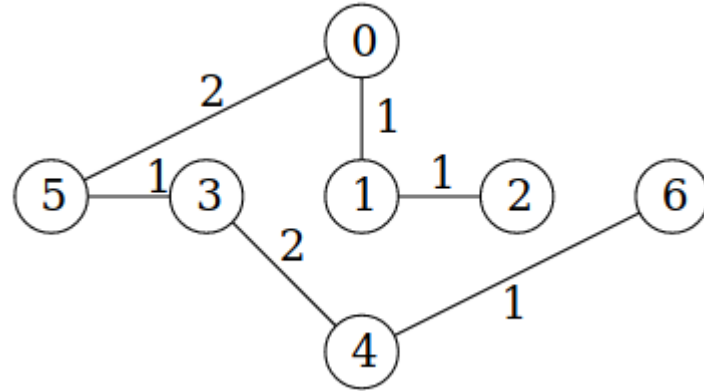
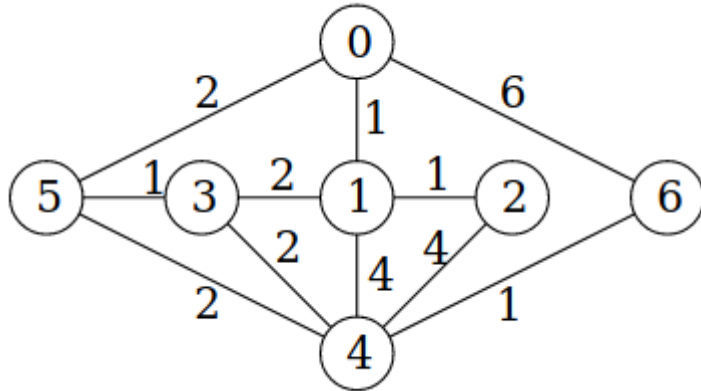
- Grafo com custos a seguir (raiz em 0):

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	0-3	0-4
			50	30
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	4-1 4-5	1-2
0 4 1 3	0 30 * 50 30 *	0-2 4-5 1-2 3-5	0 30	30
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			

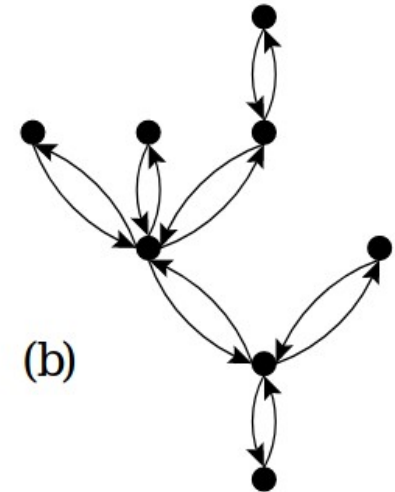
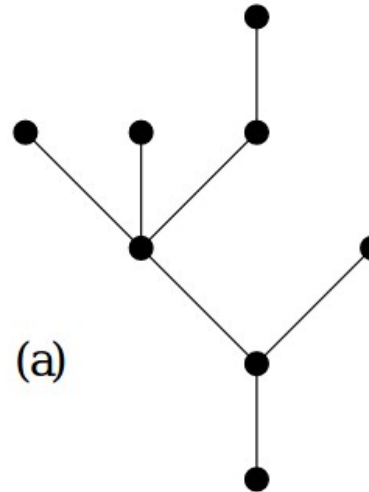
Exemplo

- Problema do Caixeiro Viajante (PCV)
- Limite inferior?
 - Árvore geradora mínima (AGM)



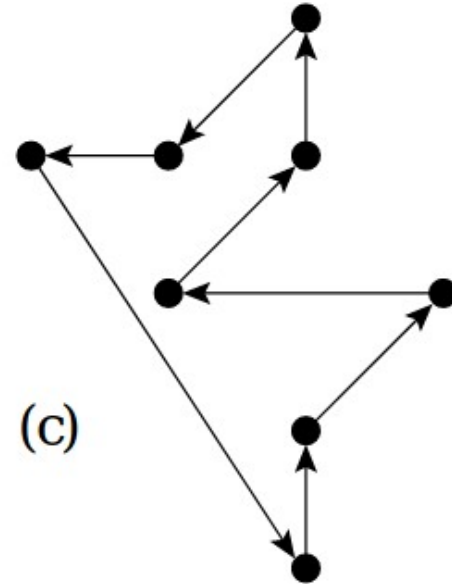
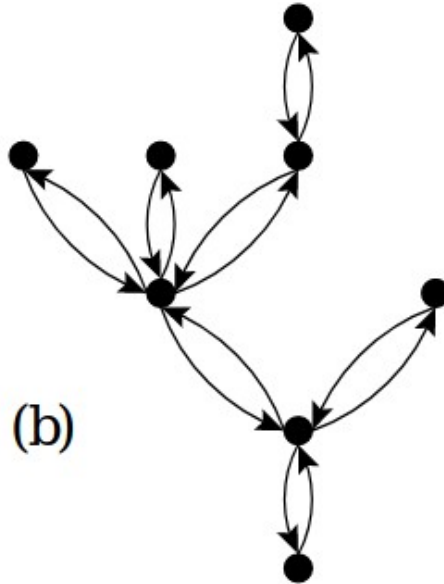
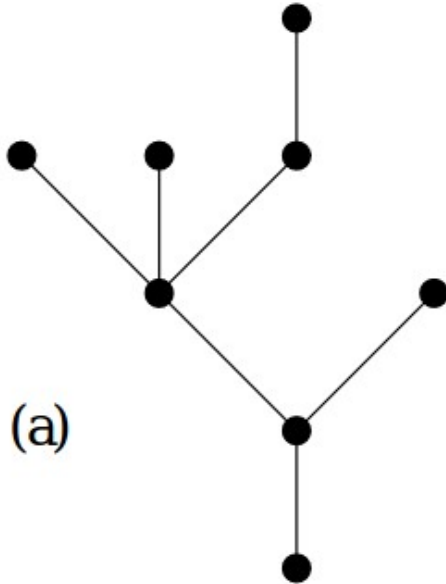
Algoritmo Aproximado

- Problema do Caixeiro Viajante (PCV)
- Limite superior?
 - Busca em Profundidade na AGM
 - Visita todos os vértices
 - Nenhuma aresta é visitada mais que 2 vezes



Algoritmo Aproximado

- Inicie em uma folha da AGM, mas sempre que a busca em profundidade for voltar para uma cidade já visitada, salte para a próxima ainda não visitada.



Algoritmo Aproximado

