

Algoritmo Aproximação

Prof. Anderson Almeida Ferreira

[DPV]9.2

[ZIV]9.2.2 e 9.2.3

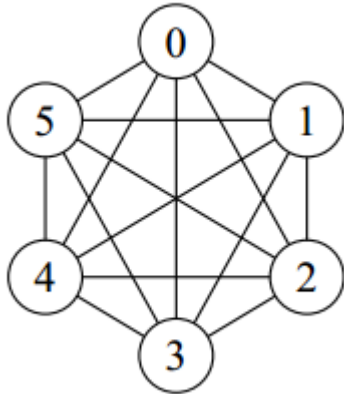
Heurísticas para Problemas NP-Completo

- Heurística: algoritmo que pode produzir um bom resultado (ou até a solução ótima), mas 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.
- Aspecto negativo: embora todas as arestas escolhidas sejam localmente mínimas, a aresta final pode ser bastante longa.

Heurística para o PCV



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

- Caminho ótimo para esta instância: 0 1 2 5 3 4 0 (comprimento 58).
- Para a heurística do vizinho mais próximo, se iniciarmos pelo vértice 0, o vértice mais próximo é o 1 com distância 3.
- A partir do 1, o mais próximo é o 2, a partir do 2 o mais próximo é o 4, a partir do 4 o mais próximo é o 3, a partir do 3 restam o 5 e o 0.
- O comprimento do caminho 0 1 2 4 3 5 0 é 60.

Heurística para o PCV

- Embora o algoritmo do vizinho mais próximo não encontre a solução ótima, a obtida está bem próxima do ótimo.
- Entretanto, é possível encontrar instâncias em que a solução obtida pode ser muito ruim.
- Pode mesmo ser arbitrariamente ruim, uma vez que a aresta final pode ser muito longa.
- É possível achar um algoritmo que garanta encontrar uma solução que seja razoavelmente boa no pior caso, desde que a classe de instâncias consideradas seja restrita.

Algoritmos Aproximados para Problemas NP-Completo

- Para projetar algoritmos polinomiais para “resolver” um problema de otimização NP-completo é necessário relaxar o significado de resolver.
- Removemos a exigência de que o algoritmo tenha sempre de obter a solução ótima.
- Procuramos algoritmos eficientes que não garantem obter a solução ótima, mas sempre obtêm uma próxima da ótima.
- Tal solução, com valor próximo da ótima, é chamada de solução aproximada.
- Um algoritmo aproximado para um problema Π é um algoritmo que gera soluções aproximadas para Π .
- Para ser útil, é importante obter um limite para a razão entre a solução ótima e a produzida pelo algoritmo aproximado.

Medindo a Qualidade da Aproximação

- O comportamento de algoritmos aproximados quanto a qualidade dos resultados (não o tempo para obtê-los) tem de ser monitorado.
- Seja I uma instância de um problema Π e seja $S^*(I)$ o valor da solução ótima para I .
- Um algoritmo aproximado gera uma solução possível para I cujo valor $S(I)$ é maior (pior) do que o valor ótimo $S^*(I)$.
- Dependendo do problema, a solução a ser obtida pode minimizar ou maximizar $S(I)$.
- Para o PCV, podemos estar interessados em um algoritmo aproximado que minimize $S(I)$: obtém o valor mais próximo possível de $S^*(I)$.
- No caso de o algoritmo aproximado obter a solução ótima, então $S(I) = S^*(I)$.

Algoritmos Aproximados - Definição

- Um algoritmo aproximado para um problema Π é um algoritmo polinomial que produz uma solução $S(I)$ para uma instância I de Π .
- O comportamento do algoritmo A é descrito pela **razão de aproximação**
 $R_A(I) = S(I) / S^*(I)$,
que representa um problema de minimização
- No caso de um problema de maximização, a razão é invertida.
- Em ambos os casos, $R_A(I) \geq 1$.

Algoritmos Aproximados para o PCV

- Seja $G = (V, A)$ um grafo não direcionado, completo, especificado por um par (N, d) .
- N é o conjunto de vértices do grafo (cidades), e d é uma função distância que mapeia as arestas em números reais, onde d satisfaz:
 1. $d(i, j) = d(j, i) \forall i, j \in N$,
 2. $d(i, j) > 0 \forall i, j \in N$,
 3. $d(i, j) + d(j, k) \geq d(i, k) \forall i, j, k \in N$
- 1ª propriedade: a distância da cidade i até outra adjacente j é igual à de j até i .
- Quando isso não acontece, temos o problema conhecido como PCV Assimétrico
- 2ª propriedade: apenas distâncias positivas.
- 3ª propriedade: desigualdade triangular. A distância de i até j somada com a de j até k deve ser maior do que a distância de i até k .

Algoritmos Aproximados para o PCV

- Quando o problema exige distâncias não restritas à desigualdade triangular, basta adicionar uma constante k a cada distância.
- Exemplo: as três distâncias envolvidas são 2, 3 e 10, que não obedecem à desigualdade triangular pois $2 + 3 < 10$. Adicionando $k = 10$ às três distâncias obtendo 12, 13 e 20, que agora satisfazem a desigualdade triangular.
- O problema alterado terá a mesma solução ótima que o problema anterior, apenas com o comprimento da rota ótima diferindo de $n \times k$.

Limite Inferior para a Solução do PCV a Partir da AGM

- A partir da AGM, podemos derivar o limite inferior para o PCV.
- Considere uma aresta (x_1, x_2) do caminho ótimo do PCV. Remova a aresta e ache um caminho iniciando em x_1 e terminando em x_2 .
- Ao retirar uma aresta do caminho ótimo, temos uma árvore geradora que consiste de um caminho que visita todas as cidades.
- Logo, o caminho ótimo para o PCV é necessariamente maior do que o comprimento da AGM.
- O limite inferior para o custo deste caminho é a AGM.
- Logo, $O_{\text{otimo}_{\text{PCV}}} > \text{AGM}$.

Limite Superior de Aproximação para o PCV

- A desigualdade triangular permite utilizar a AGM para obter um limite superior para a razão de aproximação com relação ao comprimento do caminho ótimo.
- Vamos considerar um algoritmo que visita todas as cidades, mas pode usar somente as arestas da AGM.
- Uma possibilidade é iniciar em um vértice folha e usar a seguinte estratégia:
 - Se houver aresta ainda não visitada saindo do vértice corrente, siga aquela aresta para um novo vértice.
 - Se todas as arestas a partir do vértice corrente tiverem sido visitadas, volte para o vértice adjacente pela aresta pela qual o vértice corrente foi inicialmente alcançado.
 - Termine quando retornar ao vértice inicial.

Limite Superior de Aproximação para o PCV - Busca em Profundidade

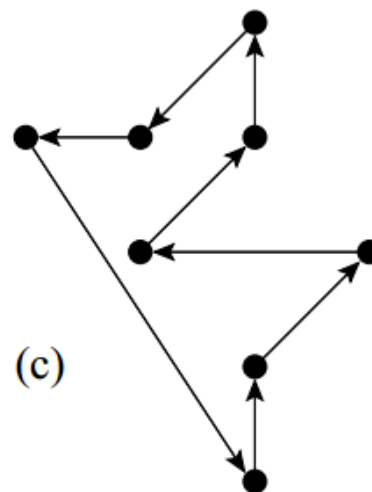
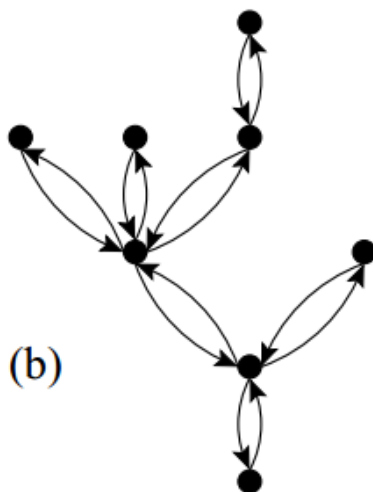
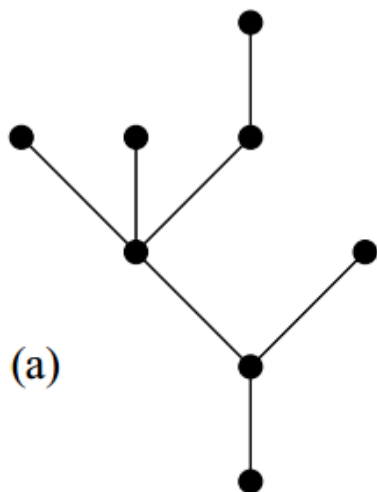
- O algoritmo descrito anteriormente é a Busca em Profundidade aplicada à AGM.
- Verifica-se que:
 - o algoritmo visita todos os vértices.
 - nenhuma aresta é visitada mais do que duas vezes.
- Obtém um caminho que visita todas as cidades cujo custo é menor ou igual a duas vezes o custo da árvore geradora mínima.
- Como o caminho ótimo é maior do que o custo da AGM, então o caminho obtido é no máximo duas vezes o custo do caminho ótimo.
 $\text{Caminho}_{\text{PCV}} < 2\text{Otimo}_{\text{PCV}}$.
- Restrição: algumas cidades são visitadas mais de uma vez.
- Para contornar o problema, usamos a desigualdade triangular.

Limite Superior de Aproximação para o PCV - Desigualdade Triangular

- Introduzimos curto-circuitos que nunca aumentam o comprimento total do caminho.
- 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.
- A rota direta não é maior do que a anterior indireta, em razão da desigualdade triangular

Limite Superior de Aproximação para o PCV - Desigualdade Triangular

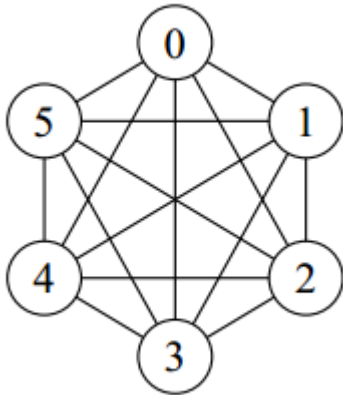
- Se todas as cidades tiverem sido visitadas, volte para o ponto de partida.



- O algoritmo constrói um caminho solução para o PCV porque cada cidade é visitada apenas uma vez, exceto a cidade de partida.

Exercício

- Qual o clico encontrado para o grafo a seguir, usando o algoritmo de aproximação visto



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

Cobertura de Vértices

- Entrada: Um grafo não direcionado $G=(V,E)$
- Saída: Um subconjunto de vértices $S \subseteq V$ que toca todas as arestas.
- Objetivo: Minimizar $|S|$

Cobertura de vértices

- Aproximação baseada em emparelhamento (um subconjunto de arestas que não tem nenhum vértice em comum).
- Um emparelhamento é *maximal* se nenhuma nova aresta pode ser adicionada a ele.
- Geração de emparelhamentos maximais:
 - Repetidamente selecione arestas disjuntas das já selecionadas, até que isso não seja mais possível.

Relação entre Emparelhamento e Cobertura de Vértices

- Qualquer cobertura de vértices em G deve ser tão grande quanto o número de arestas em qualquer emparelhamento em G
- Emparelhamento fornece uma cota inferior para a Cobertura de Vértices
- Seja S um conjunto que contenha ambas as extremidades de cada aresta em um emparelhamento maximal M .
 - S é uma cobertura de vértices
 - A cobertura S tem $2|M|$ vértices.

Algoritmo de aproximação para a cobertura de vértices

- Algoritmo
 - Encontre um emparelhamento maximal $M \subseteq E$
 - Retorne $S = \{\text{todas as extremidades de arestas em } M\}$
- Cobertura de vértices tem tamanho no máximo duas vezes o ótimo (razão de aproximação 2).

Exemplo

