Tópicos de Análise de Algoritmos

Algoritmos de aproximação

Uma introdução

Sec 11.1 e 11.2 do KT

Sec 2.4 deste livro de algoritmo de aproximação:
`https://www.ime.usp.br/~cris/aprox/`

Problemas de Otimização Combinatória

conjunto de instâncias

para cada instância I,
um conjunto $\mathrm{Sol}(I)$ de soluções viáveis e
uma função $\mathrm{val}(S)$ para cada solução S em $\mathrm{Sol}(I)$

Se $\mathrm{Sol}(I)$ é vazio, I é inviável, caso contrário, I é viável.

Problema de minimização: Dada uma instância I viável, encontrar uma solução S em $\mathrm{Sol}(I)$ tal que $\mathrm{val}(S)$ é mínimo.

Problema de maximização: Dada uma instância I viável, encontrar uma solução S em $\mathrm{Sol}(I)$ tal que $\mathrm{val}(S)$ é máximo.

$\mathrm{opt}(I)$: valor ótimo (valor de uma solução ótima)

Algoritmos de aproximação

Algoritmo A é de aproximação se
é polinomial e existe $\alpha > 0$ tal que

$$\mathrm{val}(A(I)) \leq \alpha \cdot \mathrm{opt}\,(I)$$

para toda instância I do problema (de minimização).

A é uma α-aproximação e

α é a razão de aproximação (ou garantia de performance).

$\alpha > 1$ para problemas de minimização

Para problemas de maximização,
a desigualdade é invertida e $\alpha < 1$.

Objetivo: α tão perto de 1 quanto possível

Problema dos k-centros

Dados: grafo completo $G = (V, E)$, inteiro $k > 0$ e distâncias d_{ij} para cada i e j em V tais que $d_{ii} = 0$ para todo i, $d_{ij} = d_{ji}$ para todo i e j, e $d_{ij} \leq d_{i\ell} + d_{\ell j}$.

Problema dos k-centros

Dados: grafo completo $G = (V, E)$, inteiro $k > 0$ e distâncias d_{ij} para cada i e j em V tais que $d_{ii} = 0$ para todo i, $d_{ij} = d_{ji}$ para todo i e j, e $d_{ij} \leq d_{i\ell} + d_{\ell j}$.

S: conjunto não vazio de vértices

$d(i, S) := \min\{d_{ij} : j \in S\}$

raio de S: $\max\{d(i, S) : i \in V\}$

Problema dos k-centros

Dados: grafo completo $G = (V, E)$, inteiro $k > 0$ e distâncias d_{ij} para cada i e j em V tais que $d_{ii} = 0$ para todo i, $d_{ij} = d_{ji}$ para todo i e j, e $d_{ij} \leq d_{i\ell} + d_{\ell j}$.

S: conjunto não vazio de vértices

$d(i, S) := \min\{d_{ij} : j \in S\}$

raio de S: $\max\{d(i, S) : i \in V\}$

Objetivo: encontrar conjunto S com k vértices de V e raio mínimo.

Problema dos k-centros

Dados: grafo completo $G = (V, E)$, inteiro $k > 0$ e distâncias d_{ij} para cada i e j em V tais que $d_{ii} = 0$ para todo i, $d_{ij} = d_{ji}$ para todo i e j, e $d_{ij} \leq d_{i\ell} + d_{\ell j}$.

S: conjunto não vazio de vértices

$d(i, S) := \min\{d_{ij} : j \in S\}$

raio de S: $\max\{d(i, S) : i \in V\}$

Objetivo: encontrar conjunto S com k vértices de V e raio mínimo.

Vamos mostrar uma 2-aproximação para o problema.

Se soubéssemos o valor ótimo...

Vamos supor que $V = \{1, \ldots, n\}$ e seja r o valor ótimo.

INFORMADO (n, k, d)

```
S ← ∅        D ← V
enquanto D ≠ ∅ faça
      seja s um elemento arbitrário de D
      S ← S ∪ {s}
      remova de D os vértices a distância até 2r de s
devolva S
```

Se soubéssemos o valor ótimo...

Vamos supor que $V = \{1, \ldots, n\}$ e seja r o valor ótimo.

INFORMADO (n, k, d)

```
S ← ∅        D ← V
enquanto D ≠ ∅ faça
        seja s um elemento arbitrário de D
        S ← S ∪ {s}
        remova de D os vértices a distância até 2r de s
devolva S
```

O algoritmo acima usa no máximo k centros.

Esboço da prova:

Se soubéssemos o valor ótimo...

Vamos supor que $V = \{1, \ldots, n\}$ e seja r o valor ótimo.

INFORMADO (n, k, d)

```
S ← ∅        D ← V
enquanto D ≠ ∅ faça
    seja s um elemento arbitrário de D
    S ← S ∪ {s}
    remova de D os vértices a distância até 2r de s
devolva S
```

O algoritmo acima usa no máximo k centros.

Esboço da prova: Cada novo elemento incluído em S leva consigo todo o cluster que o contém no ótimo.

Se soubéssemos o valor ótimo...

Vamos supor que $V = \{1, \ldots, n\}$ e seja r o valor ótimo.

INFORMADO (n, k, d)

```
S ← ∅        D ← V
enquanto D ≠ ∅ faça
        seja s um elemento arbitrário de D
        S ← S ∪ {s}
        remova de D os vértices a distância até 2r de s
devolva S
```

O algoritmo acima usa no máximo k centros.

Esboço da prova: Cada novo elemento incluído em S leva consigo todo o cluster que o contém no ótimo.

Faça uma busca binária pelo valor ótimo.

Mas não precisa do valor ótimo...

Vamos supor que $V = \{1, \ldots, n\}$.

```
GULOSO (n, k, d)
    S ← {1}
    enquanto |S| < k faça
        j ← arg max{d(ℓ, S) : ℓ ∈ V}        ▷ mais distante de S
        S ← S ∪ {j}
    devolva S
```

Mas não precisa do valor ótimo...

Vamos supor que $V = \{1, \ldots, n\}$.

GULOSO (n, k, d)

$S \leftarrow \{1\}$
enquanto $|S| < k$ **faça**
    $j \leftarrow \arg\max\{d(\ell, S) : \ell \in V\}$ ▷ mais distante de S
    $S \leftarrow S \cup \{j\}$
devolva S

Teorema:
GULOSO é uma 2-aproximação para o problema dos k-centros.

Esboço da prova:

Mas não precisa do valor ótimo...

Vamos supor que $V = \{1, \ldots, n\}$.

GULOSO (n, k, d)

$S \leftarrow \{1\}$
enquanto $|S| < k$ **faça**
    $j \leftarrow \arg\max\{d(\ell, S) : \ell \in V\}$ $\triangleright$ mais distante de S
    $S \leftarrow S \cup \{j\}$
devolva S

Teorema:
GULOSO é uma 2-aproximação para o problema dos k-centros.

Esboço da prova: É um caso particular do anterior.
Especificamente, o j escolhido aqui poderia ser
um s do algoritmo anterior em cada iteração.

Problema dos k-centros

Teorema: Se existe uma α-aproximação para o problema dos k-centros com $\alpha < 2$, então P $=$ NP.

Problema dos k-centros

Teorema: Se existe uma α-aproximação para o problema dos k-centros com $\alpha < 2$, então $\mathrm{P} = \mathrm{NP}$.

Conjunto dominante: conjunto S de vértices tal que todo vértice do grafo ou está em S ou é adjacente a um vértice de S.

Problema dos k-centros

Teorema: Se existe uma α-aproximação para
o problema dos k-centros com $\alpha < 2$, então $\mathsf{P} = \mathsf{NP}$.

Conjunto dominante: conjunto S de vértices tal que
todo vértice do grafo ou está em S ou é adjacente a um vértice de S.

Problema: Dado um grafo G e um inteiro $k \geq 0$,
existe um conjunto dominante em G de tamanho no máximo k?

Problema dos k-centros

Teorema: Se existe uma α-aproximação para o problema dos k-centros com $\alpha < 2$, então $\mathsf{P} = \mathsf{NP}$.

Conjunto dominante: conjunto S de vértices tal que todo vértice do grafo ou está em S ou é adjacente a um vértice de S.

Problema: Dado um grafo G e um inteiro $k \geq 0$, existe um conjunto dominante em G de tamanho no máximo k?

Este problema é NP-completo.

Problema dos k-centros

Teorema: Se existe uma α-aproximação para
o problema dos k-centros com $\alpha < 2$, então $P = NP$.

Conjunto dominante: conjunto S de vértices tal que
todo vértice do grafo ou está em S ou é adjacente a um vértice de S.

Problema: Dado um grafo G e um inteiro $k \geq 0$,
existe um conjunto dominante em G de tamanho no máximo k?

Este problema é NP-completo.

Esboço da prova do teorema: Redução do conjunto dominante.

Problema dos k-centros

Teorema: Se existe uma α-aproximação para o problema dos k-centros com $\alpha < 2$, então $\mathrm{P} = \mathrm{NP}$.

Conjunto dominante: conjunto S de vértices tal que todo vértice do grafo ou está em S ou é adjacente a um vértice de S.

Problema: Dado um grafo G e um inteiro $k \geq 0$, existe um conjunto dominante em G de tamanho no máximo k?

Este problema é NP-completo.

Esboço da prova do teorema: Redução do conjunto dominante.

Dados G e k, defina $d_{ij} = 1$ se i e j são vizinhos em G, e $d_{ij} = 2$ caso contrário.

Problema dos k-centros

Teorema: Se existe uma α-aproximação para
o problema dos k-centros com $\alpha < 2$, então P $=$ NP.

Conjunto dominante: conjunto S de vértices tal que
todo vértice do grafo ou está em S ou é adjacente a um vértice de S.

Problema: Dado um grafo G e um inteiro $k \geq 0$,
existe um conjunto dominante em G de tamanho no máximo k?

Este problema é NP-completo.

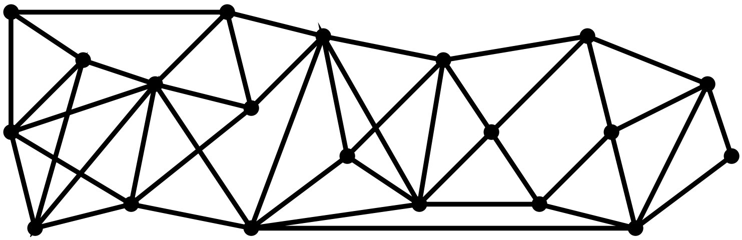
Esboço da prova do teorema: Redução do conjunto dominante.

Dados G e k, defina $d_{ij} = 1$ se i e j são vizinhos em G,
e $d_{ij} = 2$ caso contrário.

Há conjunto dominante com k vértices sse a α-aproximação com
$\alpha < 2$ aplicada a $(V(G), k, d)$ encontra um tal conjunto dominante.

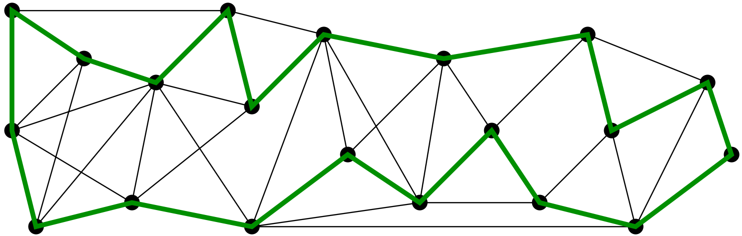
Circuito Hamiltoniano

G: grafo conexo



Circuito Hamiltoniano

G: grafo conexo



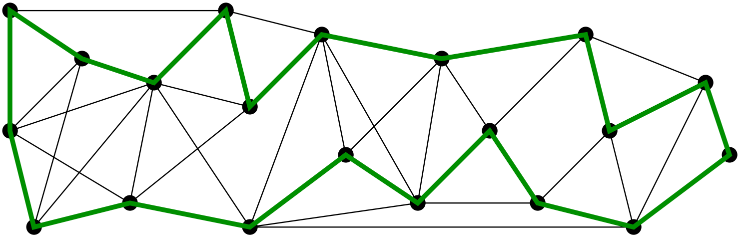
Circuito Hamiltoniano: circuito que passa por todos os vértices

Problema do Caixeiro Viajante

Dados

grafo completo G

comprimento ℓ_{ij} da aresta $ij \quad (ij \in E_G)$

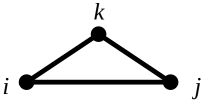


Problema (TSP): Dados G e ℓ, encontrar circuito Hamiltoniano C em G de comprimento $\ell(C)$ mínimo.

Variantes do Caixeiro Viajante

TSP métrico

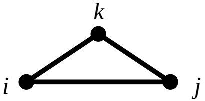
- grafo completo
- função comprimento ℓ satisfaz
  desigualdade triangular: $\ell_{ij} \leq \ell_{ik} + \ell_{kj} \quad \forall i, j, k \in V_G$



Variantes do Caixeiro Viajante

TSP métrico

- grafo completo
- função comprimento ℓ satisfaz
  desigualdade triangular: $\ell_{ij} \leq \ell_{ik} + \ell_{kj} \quad \forall i, j, k \in V_G$

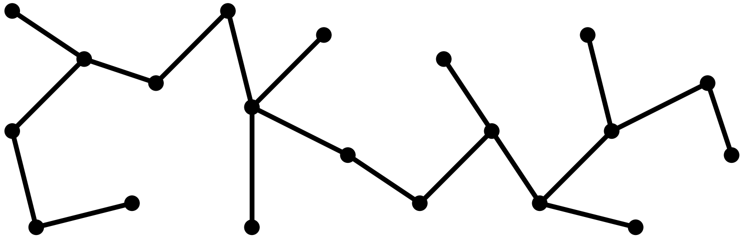


Resultados Conhecidos:

- NP-difícil mesmo se $\ell_e \in \{1, 2\}$ para toda aresta e [GJ79]
- Difícil de aproximar [SG76]
- $3/2$-aproximação para o caso métrico [C76]
- PTAS para o caso Euclideano [A98,M99]

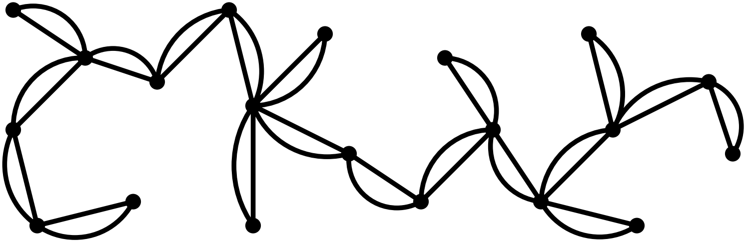
2-aproximação p/o TSP Métrico

(1) Árvore geradora de comprimento mínimo (polinomial)



2-aproximação p/o TSP Métrico

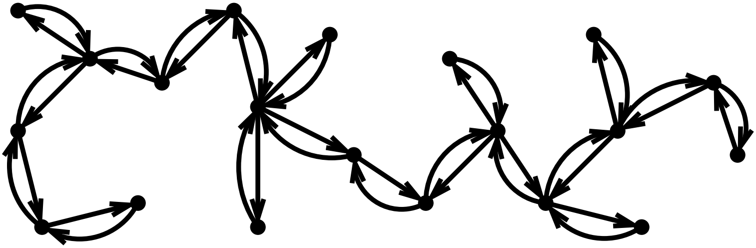
(1) Árvore geradora de comprimento mínimo (polinomial)
(2) Dobre as arestas da árvore (todos os vértices tem grau par)



2-aproximação p/o TSP Métrico

(1) Árvore geradora de comprimento mínimo (polinomial)
(2) Dobre as arestas da árvore (todos os vértices tem grau par)
(3) Obtenha ciclo Euleriano P (polinomial)

ciclo Euleriano: passeio fechado que passa exatamente uma vez por cada aresta

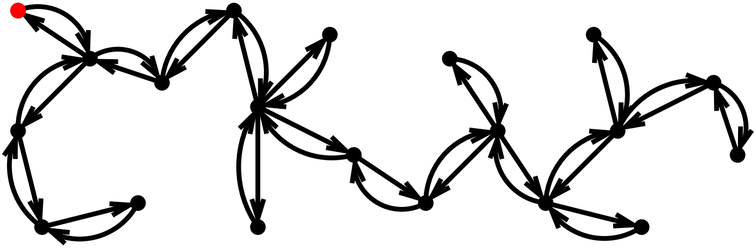


2-aproximação p/o TSP Métrico

(1) Árvore geradora de comprimento mínimo (polinomial)
(2) Dobre as arestas da árvore (todos os vértices tem grau par)
(3) Obtenha ciclo Euleriano P (polinomial)

ciclo Euleriano: passeio fechado que passa exatamente uma vez por cada aresta

(4) Obtenha de P circuito Hamiltoniano C

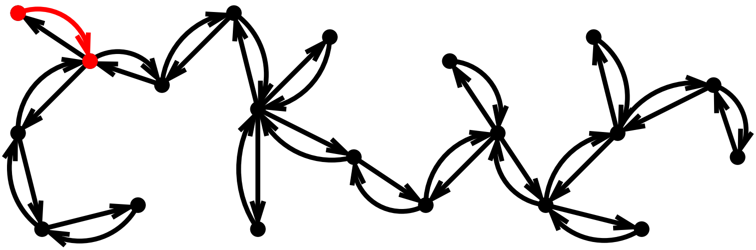


2-aproximação p/o TSP Métrico

(1) Árvore geradora de comprimento mínimo (polinomial)
(2) Dobre as arestas da árvore (todos os vértices tem grau par)
(3) Obtenha ciclo Euleriano P (polinomial)

ciclo Euleriano: passeio fechado que passa exatamente uma vez por cada aresta

(4) Obtenha de P circuito Hamiltoniano C

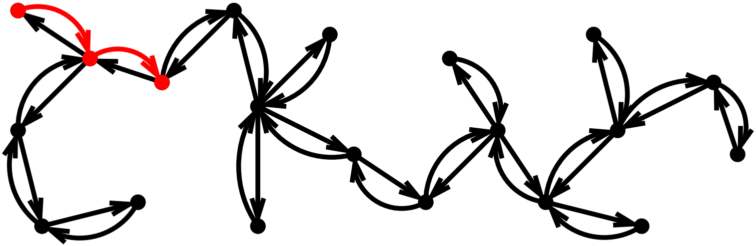


2-aproximação p/o TSP Métrico

(1) Árvore geradora de comprimento mínimo (polinomial)
(2) Dobre as arestas da árvore (todos os vértices tem grau par)
(3) Obtenha ciclo Euleriano P (polinomial)

ciclo Euleriano: passeio fechado que passa exatamente uma vez por cada aresta

(4) Obtenha de P circuito Hamiltoniano C

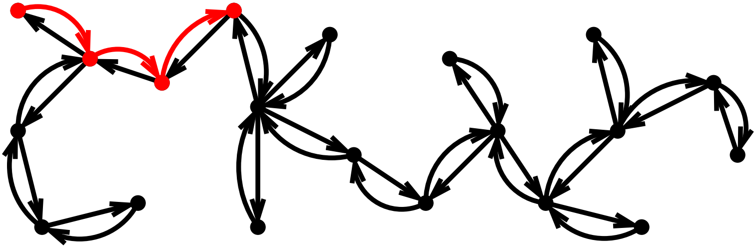


2-aproximação p/o TSP Métrico

(1) Árvore geradora de comprimento mínimo (polinomial)
(2) Dobre as arestas da árvore (todos os vértices tem grau par)
(3) Obtenha ciclo Euleriano P (polinomial)

ciclo Euleriano: passeio fechado que passa exatamente uma vez por cada aresta

(4) Obtenha de P circuito Hamiltoniano C

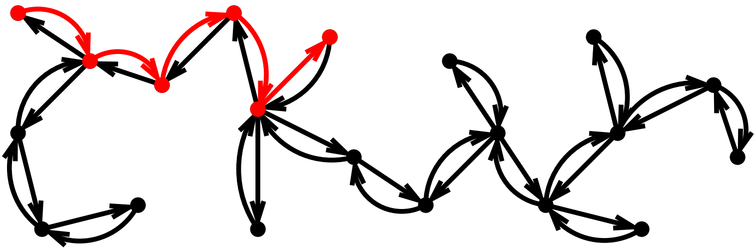


2-aproximação p/o TSP Métrico

(1) Árvore geradora de comprimento mínimo (polinomial)
(2) Dobre as arestas da árvore (todos os vértices tem grau par)
(3) Obtenha ciclo Euleriano P (polinomial)

ciclo Euleriano: passeio fechado que passa exatamente uma vez por cada aresta

(4) Obtenha de P circuito Hamiltoniano C

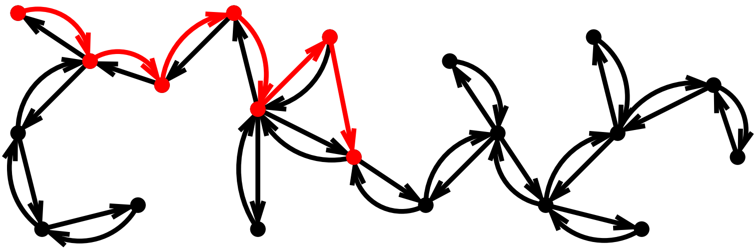


2-aproximação p/o TSP Métrico

(1) Árvore geradora de comprimento mínimo (polinomial)
(2) Dobre as arestas da árvore (todos os vértices tem grau par)
(3) Obtenha ciclo Euleriano P (polinomial)

ciclo Euleriano: passeio fechado que passa exatamente uma vez por cada aresta

(4) Obtenha de P circuito Hamiltoniano C

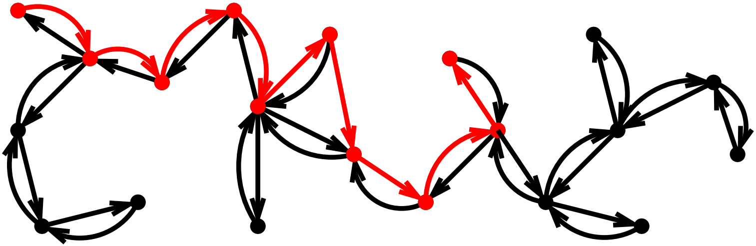


2-aproximação p/o TSP Métrico

(1) Árvore geradora de comprimento mínimo (polinomial)
(2) Dobre as arestas da árvore (todos os vértices tem grau par)
(3) Obtenha ciclo Euleriano P (polinomial)

ciclo Euleriano: passeio fechado que passa exatamente uma vez por cada aresta

(4) Obtenha de P circuito Hamiltoniano C

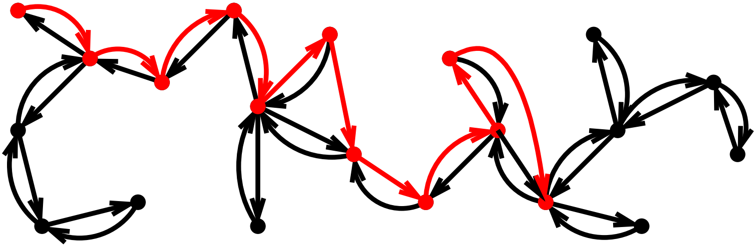


2-aproximação p/o TSP Métrico

(1) Árvore geradora de comprimento mínimo (polinomial)
(2) Dobre as arestas da árvore (todos os vértices tem grau par)
(3) Obtenha ciclo Euleriano P (polinomial)

ciclo Euleriano: passeio fechado que passa exatamente uma vez por cada aresta

(4) Obtenha de P circuito Hamiltoniano C

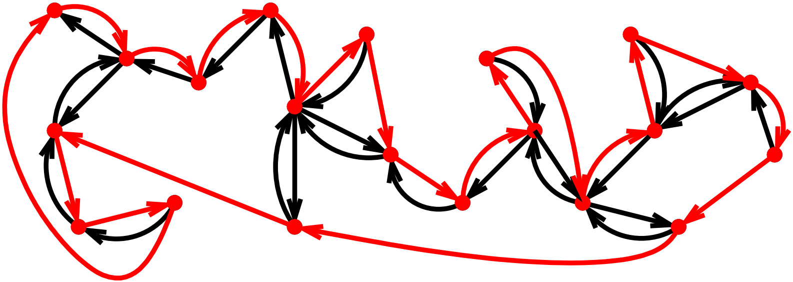


2-aproximação p/o TSP Métrico

(1) Árvore geradora de comprimento mínimo (polinomial)
(2) Dobre as arestas da árvore (todos os vértices tem grau par)
(3) Obtenha ciclo Euleriano P (polinomial)

ciclo Euleriano: passeio fechado que passa exatamente uma vez por cada aresta

(4) Obtenha de P circuito Hamiltoniano C

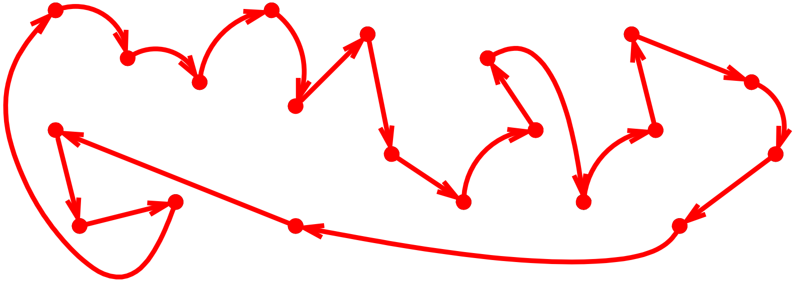


2-aproximação p/o TSP Métrico

(1) Árvore geradora de comprimento mínimo (polinomial)
(2) Dobre as arestas da árvore (todos os vértices tem grau par)
(3) Obtenha ciclo Euleriano P (polinomial)

ciclo Euleriano: passeio fechado que passa exatamente uma vez por cada aresta

(4) Obtenha de P circuito Hamiltoniano C
(5) Devolva C



2-Aproximação p/o TSP Métrico

Algoritmo TSPM-Rosenkrantz-Stearn-Lewis (G, ℓ)

$T \leftarrow \text{mst}(G, \ell)$

$T' \leftarrow T + T$

$P \leftarrow \text{Euler}(T')$

$C \leftarrow \text{atalho}(P)$

devolve C

2-Aproximação p/o TSP Métrico

Algoritmo TSPM-Rosenkrantz-Stearn-Lewis (G, ℓ)

$T \leftarrow \text{mst}(G, \ell)$

$T' \leftarrow T + T$

$P \leftarrow \text{Euler}(T')$

$C \leftarrow \text{atalho}(P)$

devolve C

Tempo de execução: $\mathrm{O}(n^2)$

n: o número de vértices de G

2-Aproximação p/o TSP Métrico

Delimitação inferior: $\text{opt} \geq \ell(T)$

onde T é árvore geradora de comprimento mínimo

2-Aproximação p/o TSP Métrico

Delimitação inferior: $\text{opt} \geq \ell(T)$

onde T é árvore geradora de comprimento mínimo

Prova:

C^*: circuito Hamiltoniano de comprimento mínimo

e: aresta de C^*

$C^* - e$ é árvore geradora

2-Aproximação p/o TSP Métrico

Delimitação inferior: $\text{opt} \geq \ell(T)$

onde T é árvore geradora de comprimento mínimo

Prova:

C^*: circuito Hamiltoniano de comprimento mínimo

e: aresta de C^*

$C^* - e$ é árvore geradora

$$\text{opt} = \ell(C^*) \geq \ell(C^* - e) \geq \ell(T). \quad \square$$

2-Aproximação p/o TSP Métrico

Delimitação inferior: $\mathrm{opt} \geq \ell(T)$

onde T é árvore geradora de comprimento mínimo

Prova:

C^*: circuito Hamiltoniano de comprimento mínimo

e: aresta de C^*

$C^* - e$ é árvore geradora

$$\mathrm{opt} = \ell(C^*) \geq \ell(C^* - e) \geq \ell(T). \quad \square$$

Teorema: TSPM-Rosenkrantz-Stearn-Lewis é uma 2-aproximação polinomial para o TSP métrico.

2-Aproximação p/o TSP Métrico

Delimitação inferior: $\text{opt} \geq \ell(T)$

onde T é árvore geradora de comprimento mínimo

Prova:

C^*: circuito Hamiltoniano de comprimento mínimo

e: aresta de C^*

$C^* - e$ é árvore geradora

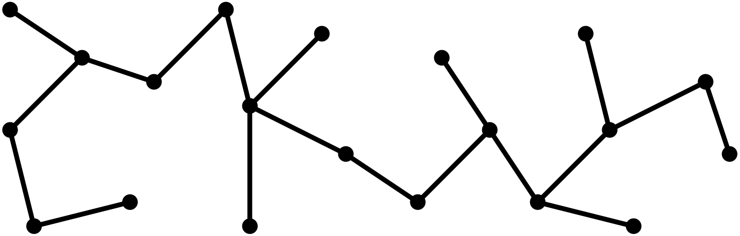
$$\text{opt} = \ell(C^*) \geq \ell(C^* - e) \geq \ell(T). \quad \square$$

Teorema: TSPM-Rosenkrantz-Stearn-Lewis é uma 2-aproximação polinomial para o TSP métrico.

Prova: $\ell(C) \leq \ell(P) = \ell(T') = 2\,\ell(T) \leq 2\,\text{opt}. \quad \square$

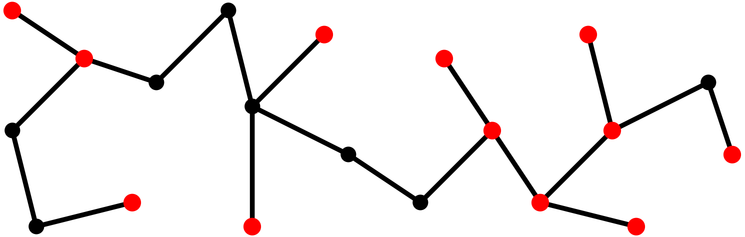
Algoritmo de Christofides

(1) Árvore geradora de comprimento mínimo



Algoritmo de Christofides

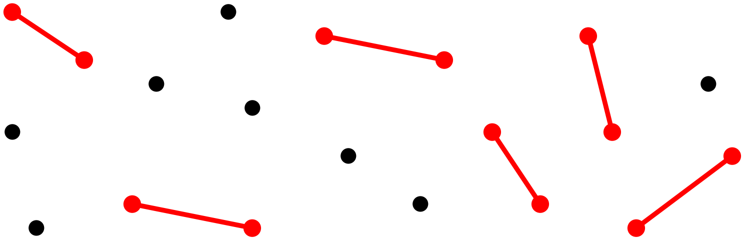
(1) Árvore geradora de comprimento mínimo



vértices de grau ímpar

Algoritmo de Christofides

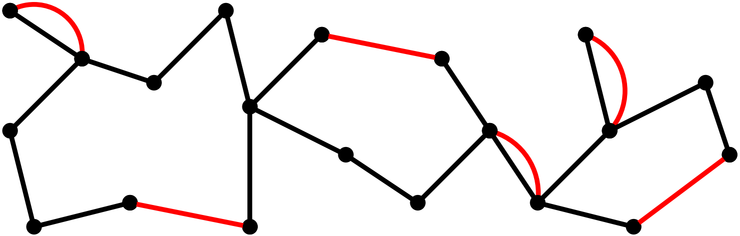
(1) Árvore geradora de comprimento mínimo



(2) Emparelhamento perfeito de comprimento mínimo
no subgrafo induzido pelos vértices de grau ímpar (polinomial)

Algoritmo de Christofides

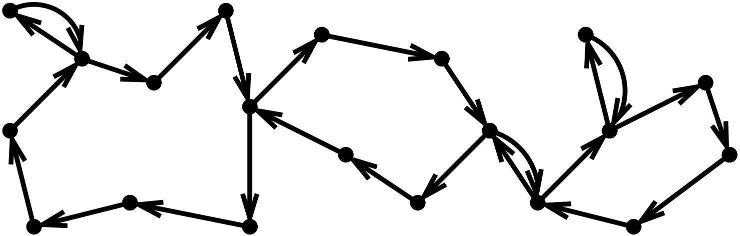
(1) Árvore geradora de comprimento mínimo



(2) Emparelhamento perfeito de comprimento mínimo
no subgrafo induzido pelos vértices de grau ímpar (polinomial)
(3) Junte os dois obtendo T' Euleriano

Algoritmo de Christofides

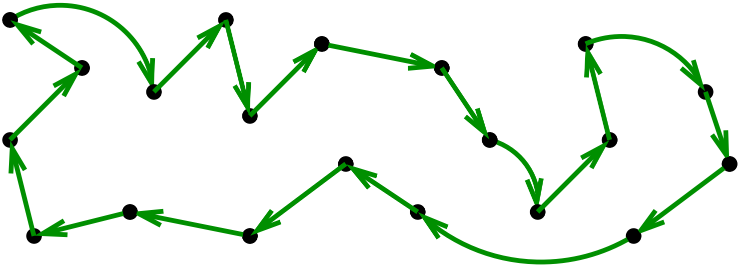
(1) Árvore geradora de comprimento mínimo



(2) Emparelhamento perfeito de comprimento mínimo
no subgrafo induzido pelos vértices de grau ímpar (polinomial)
(3) Junte os dois obtendo T' Euleriano
(4) Obtenha ciclo Euleriano P de T'

Algoritmo de Christofides

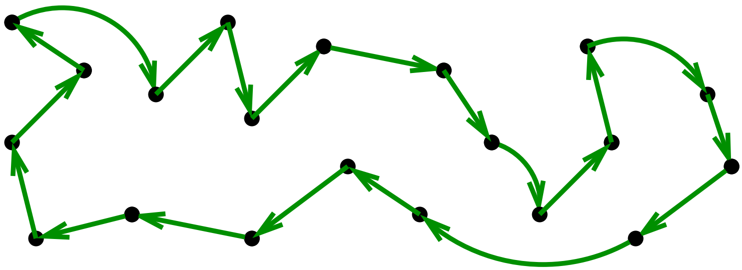
(1) Árvore geradora de comprimento mínimo



(2) Emparelhamento perfeito de comprimento mínimo
no subgrafo induzido pelos vértices de grau ímpar (polinomial)
(3) Junte os dois obtendo T' Euleriano
(4) Obtenha ciclo Euleriano P de T'
(5) Obtenha de P circuito Hamiltoniano C

Algoritmo de Christofides

(1) Árvore geradora de comprimento mínimo



(2) Emparelhamento perfeito de comprimento mínimo
no subgrafo induzido pelos vértices de grau ímpar (polinomial)
(3) Junte os dois obtendo T' Euleriano
(4) Obtenha ciclo Euleriano P de T'
(5) Obtenha de P circuito Hamiltoniano C
(6) Devolva C

Algoritmo de Christofides

Algoritmo TSPM-Christofides (G, ℓ)

$T \leftarrow \text{mst}(G, \ell)$
$I \leftarrow$ conjunto de vértices de grau ímpar de T
$M \leftarrow \text{Edmonds}(G[I], \ell)$
$T' \leftarrow T + M$
$P \leftarrow \text{Euler}(T')$
$C \leftarrow \text{atalho}(P)$
devolve C

($G[I]$: subgrafo de G induzido por I)

Algoritmo de Christofides

Algoritmo TSPM-Christofides (G, ℓ)

$T \leftarrow \text{mst}(G, \ell)$
$I \leftarrow$ conjunto de vértices de grau ímpar de T
$M \leftarrow \text{Edmonds}(G[I], \ell)$
$T' \leftarrow T + M$
$P \leftarrow \text{Euler}(T')$
$C \leftarrow \text{atalho}(P)$
devolve C

($G[I]$: subgrafo de G induzido por I)

Tempo de execução: n^3
n: o número de vértices de G

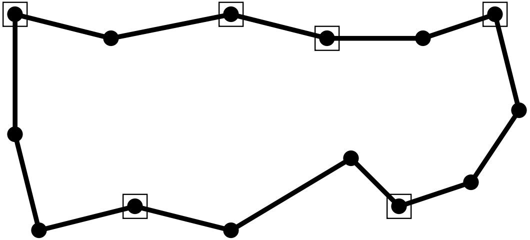
Algoritmo de Christofides: análise

Uma segunda delimitação inferior: $\text{opt} \ \geq \ 2\,\ell(M)$
onde M é e. p. de comprimento mínimo em $G[I]$

Algoritmo de Christofides: análise

Uma segunda delimitação inferior: $\text{opt} \geq 2\,\ell(M)$
onde M é e. p. de comprimento mínimo em $G[I]$

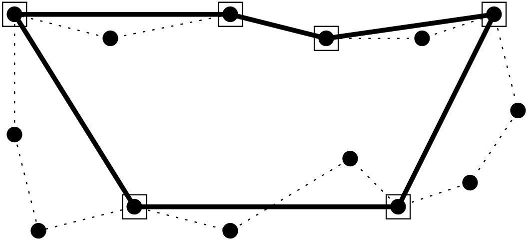
Pr: C^*: circuito Hamiltoniano de comprimento mínimo
I: conjunto de vértices de grau ímpar de T ($|I|$ é par)



Algoritmo de Christofides: análise

Uma segunda delimitação inferior: $\text{opt} \geq 2\,\ell(M)$
onde M é e. p. de comprimento mínimo em $G[I]$

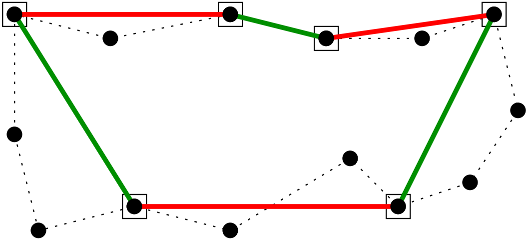
Pr: C^*: circuito Hamiltoniano de comprimento mínimo
I: conjunto de vértices de grau ímpar de T ($|I|$ é par)
C': circuito induzido por C^* em I



Algoritmo de Christofides: análise

Uma segunda delimitação inferior: $\text{opt} \geq 2\,\ell(M)$
onde M é e. p. de comprimento mínimo em $G[I]$

Pr: C^*: circuito Hamiltoniano de comprimento mínimo
I: conjunto de vértices de grau ímpar de T ($|I|$ é par)
C': circuito induzido por C^* em I



C' determina dois emparelhamentos perfeitos em $G[I]$: M_1 e M_2

Algoritmo de Christofides: análise

Uma segunda delimitação inferior: $\mathrm{opt} \geq 2\,\ell(M)$
onde M é e. p. de comprimento mínimo em $G[I]$.

Prova:

$$\begin{aligned} 2\,\ell(M) &\leq \ell(M_1) + \ell(M_2) \\ &= \ell(C') \\ &\leq \ell(C^*) && \text{(pela desigualdade triangular)} \\ &= \mathrm{opt}\,. \end{aligned}$$

$\square$

Algoritmo de Christofides: análise

Teorema: TSPM-Christofides é uma $\frac{3}{2}$-aproximação polinomial para o TSP métrico.

Algoritmo de Christofides: análise

Teorema: TSPM-Christofides é uma $\frac{3}{2}$-aproximação polinomial para o TSP métrico.

Prova:

$$\begin{aligned} \ell(C) &\leq \ell(P) \\ &= \ell(T') \\ &= \ell(T) + \ell(M) \\ &\leq \text{opt} + \frac{1}{2}\,\text{opt} \\ &= \frac{3}{2}\,\text{opt}\,. \end{aligned}$$

□

Algoritmo de Christofides: análise

Teorema: TSPM-Christofides é uma $\frac{3}{2}$-aproximação polinomial para o TSP métrico.

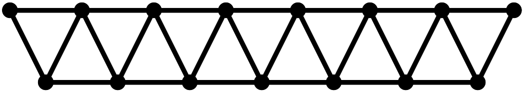
Prova:

$$\begin{aligned} \ell(C) &\leq \ell(P) \\ &= \ell(T') \\ &= \ell(T) + \ell(M) \\ &\leq \mathrm{opt} + \frac{1}{2}\,\mathrm{opt} \\ &= \frac{3}{2}\,\mathrm{opt}\,. \end{aligned}$$

□

Melhor algoritmo de aproximação conhecido para o TSP métrico até 2020, quando foi apresentada uma $(3/2 - \epsilon)$-aproximação.

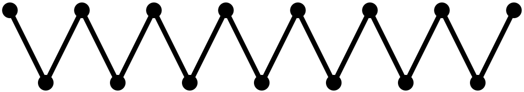
Algoritmo de Christofides: exemplo justo



$2n + 1$ vértices

Algoritmo de Christofides: exemplo justo

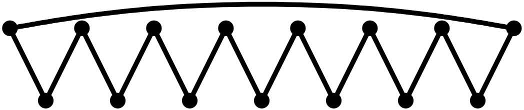
Christofides



$2n + 1$ vértices

Algoritmo de Christofides: exemplo justo

Christofides

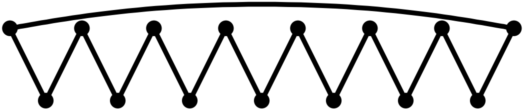


$2n + 1$ vértices

Comprimento: $3n$

Algoritmo de Christofides: exemplo justo

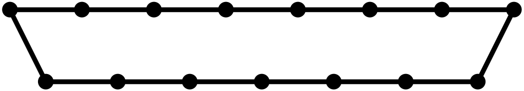
Christofides



$2n + 1$ vértices

Comprimento: $3n$

Circuito ótimo



Comprimento: $2n + 1$

TSP Euclideano

PTAS: esquema de aproximação polinomial

$(1+\epsilon)$-aproximação polinomial para todo $\epsilon > 0$ fixo

Arora'96 e Mitchel'96: PTAS para o TSP Euclideano

Ideia:

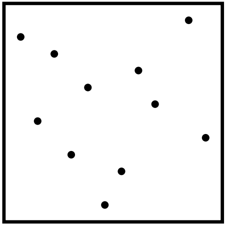
TSP Euclideano

PTAS: esquema de aproximação polinomial

$(1+\epsilon)$-aproximação polinomial para todo $\epsilon > 0$ fixo

Arora'96 e Mitchel'96: PTAS para o TSP Euclideano

Ideia:



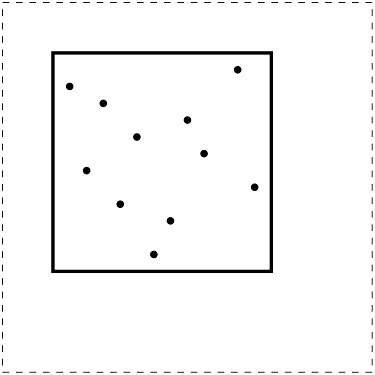
TSP Euclideano

PTAS: esquema de aproximação polinomial

$(1+\epsilon)$-aproximação polinomial para todo $\epsilon > 0$ fixo

Arora'96 e Mitchel'96: PTAS para o TSP Euclideano

Ideia:



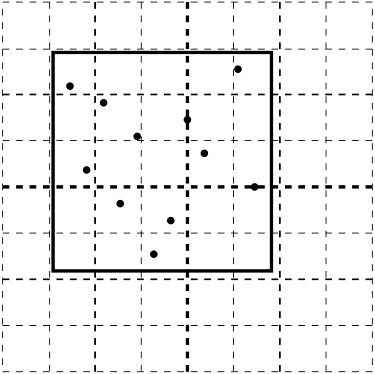
TSP Euclideano

PTAS: esquema de aproximação polinomial

$(1+\epsilon)$-aproximação polinomial para todo $\epsilon > 0$ fixo

Arora'96 e Mitchel'96: PTAS para o TSP Euclideano

Ideia:



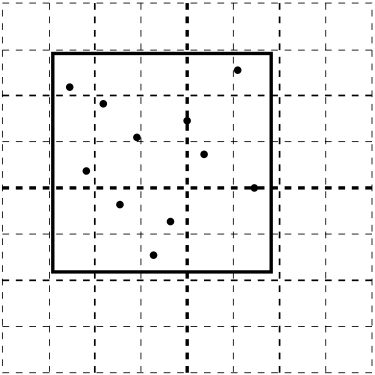
TSP Euclideano

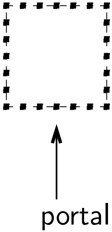
PTAS: esquema de aproximação polinomial

$(1+\epsilon)$-aproximação polinomial para todo $\epsilon > 0$ fixo

Arora'96 e Mitchel'96: PTAS para o TSP Euclideano

Ideia:





TSP Euclideano

PTAS: esquema de aproximação polinomial

$(1+\epsilon)$-aproximação polinomial para todo $\epsilon > 0$ fixo

Arora'96 e Mitchel'96: PTAS para o TSP Euclideano

Ideia:

Circuito que respeita portal:
entra e sai dos quadrados da dissecção através de portais.

Algoritmo: Encontra um circuito Hamiltoniano mais curto que respeita os portais por programação dinâmica.

Tal circuito está tão próximo quando se queira de um TSP tour.

Resultado de Inaproximabilidade

$\alpha : \mathbb{N} \to \mathbb{N}$ função polinomialmente computável

Teorema (Sahni e Gonzalez '76): Se existir $\alpha(n)$-aproximação polinomial p/o TSP(G, ℓ), onde $n := |V_G|$, então P=NP.

Resultado de Inaproximabilidade

$\alpha : \mathbb{N} \to \mathbb{N}$ função polinomialmente computável

Teorema (Sahni e Gonzalez '76): Se existir $\alpha(n)$-aproximação polinomial p/o TSP(G, ℓ), onde $n := |V_G|$, então P=NP.

Problema HC: Dado G, decidir se G tem ou não um circuito Hamiltoniano.

- NP-completo [K72]

Resultado de Inaproximabilidade

Teorema (Sahni e Gonzalez '76): Se existir $\alpha(n)$-aproximação polinomial p/o TSP(G, ℓ), onde $n := |V_G|$, então P=NP.

Prova:

$$\alpha(n)\text{-aproximação polinomial p/o TSP}$$
$$\Downarrow$$
$$\text{algoritmo polinomial p/o HC}$$

Resultado de Inaproximabilidade

Teorema (Sahni e Gonzalez '76): Se existir $\alpha(n)$-aproximação polinomial p/o TSP(G, ℓ), onde $n := |V_G|$, então P=NP.

Prova:

$$\begin{array}{cl} \alpha(n)\text{-aproximação polinomial p/o TSP} & \leftarrow A \\ \Downarrow & \\ \text{algoritmo polinomial p/o HC} & \leftarrow B \end{array}$$

Resultado de Inaproximabilidade

Teorema (Sahni e Gonzalez '76): Se existir $\alpha(n)$-aproximação polinomial p/o TSP(G, ℓ), onde $n := |V_G|$, então P=NP.

Prova:

$$\begin{array}{cl} \alpha(n)\text{-aproximação polinomial p/o TSP} & \leftarrow A \\ \Downarrow & \\ \text{algoritmo polinomial p/o HC} & \leftarrow B \end{array}$$

Algoritmo B (G)

$H \leftarrow$ grafo completo em V_G
para cada e em E_G faça $\ell_e \leftarrow 1$
para cada e em $E_H \setminus E_G$ faça $\ell_e \leftarrow \alpha(|V_G|)|V_G| + 1$
$C \leftarrow A(H, \ell)$
se $\ell(C) \leq \alpha(|V_G|)|V_G|$
então devolva "Sim"
senão devolva "Não"

Resultado de Inaproximabilidade

A polinomial $\Rightarrow$ B polinomial

Resultado de Inaproximabilidade

A polinomial $\Rightarrow$ B polinomial

se existe circuito hamiltoniano em G,
então $\text{opt} = |V_G|$ e

$$I(C) \;\leq\; \alpha(|V_G|)\,\text{opt} \;=\; \alpha(|V_G|)|V_G|.$$

Resultado de Inaproximabilidade

A polinomial $\Rightarrow$ B polinomial

se existe circuito hamiltoniano em G,
então $\mathrm{opt} = |V_G|$ e

$$l(C) \;\leq\; \alpha(|V_G|)\,\mathrm{opt} \;=\; \alpha(|V_G|)|V_G|.$$

se não existe circuito hamiltoniano em G,
então

$$\ell(C) \;\geq\; \alpha(|V_G|)|V_G| + 1$$

pois qq circ. hamilt. usa $e \notin E_G$
(i.e., $\ell_e \;=\; \alpha(|V_G|)|V_G| + 1$).

$\square$

Para completar...