O Problema do Caixeiro Viajante

Leandro G. M. Alvim

Agenda

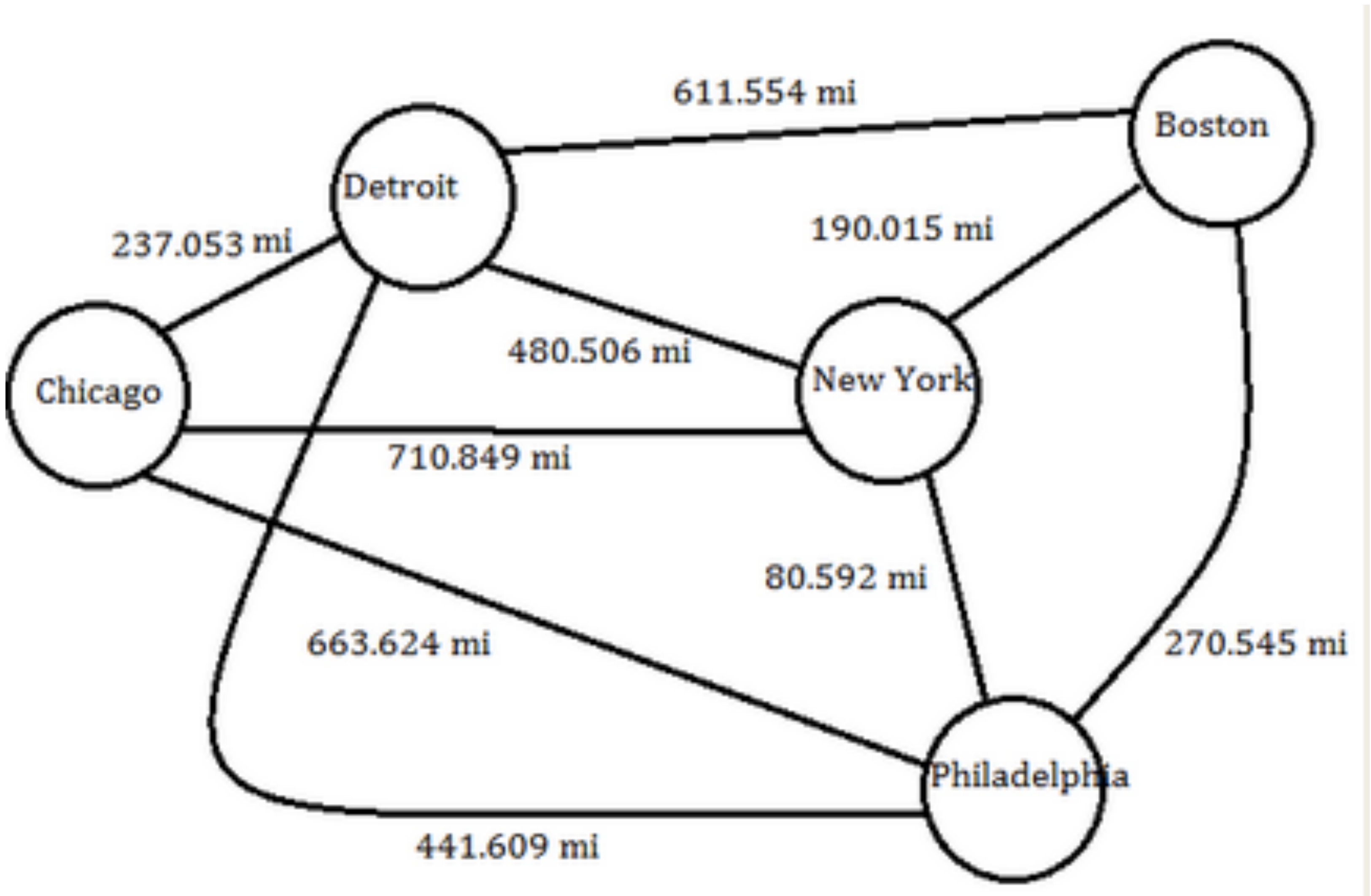
- Definição do Problema
- Complexidade
- História
- Aplicações
- Abordagens
- Referências

Problema

- Seja G=(V,E) um grafo tal que V representa um conjunto de cidades e E um conjunto dos custos entre cidades, queremos encontrar a menor rota que passa por cada cidade uma única vez e retorne a sua origem.

Problema

- Encontrar o circuito Hamiltoniano mais curto



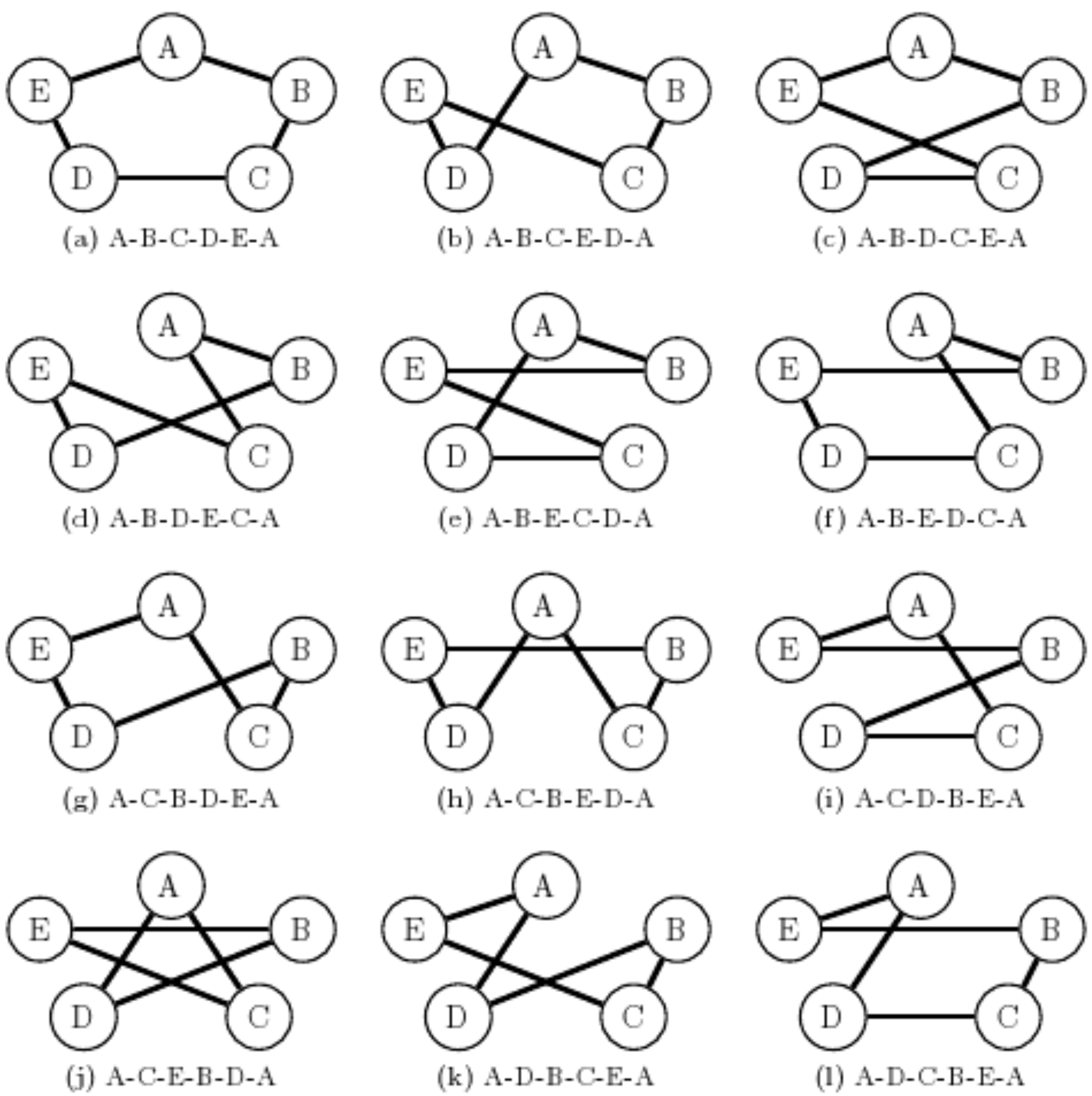
Problema

- Seja V={v1,..vn} um conjunto de vértices, A={(r,s):r,s e V} um conjunto de arestas e drs o custo associado a (r,s)
    - sTSP: drs=dsr
    - aTSP: drs<>dsr
    - mTSP: vários caixeiros partem e terminam em um mesmo vértice

Complexidade

- Algoritmo Força Bruta

$(V\text{-}1)!/2$



(a) $cost(ABCDEA) = 3 + 5 + 2 + 5 + 6 = 21$

(b) $cost(ABCEDA) = 3 + 5 + 3 + 5 + 5 = 21$

(c) $cost(ABDCEA) = 3 + 1 + 2 + 3 + 6 = \mathbf{15}$

(d) $cost(ABDECA) = 3 + 1 + 5 + 3 + 4 = 16$

(e) $cost(ABECDA) = 3 + 4 + 3 + 2 + 5 = 17$

(f) $cost(ABEDCA) = 3 + 4 + 5 + 2 + 4 = 18$

(g) $cost(ACBDEA) = 4 + 5 + 1 + 5 + 6 = 21$

(h) $cost(ACBEDA) = 4 + 5 + 4 + 5 + 5 = 23$

(i) $cost(ACDBEA) = 4 + 2 + 1 + 4 + 6 = 17$

(j) $cost(ACEBDA) = 4 + 3 + 4 + 1 + 5 = 17$

(k) $cost(ADBCEA) = 5 + 1 + 5 + 3 + 6 = 20$

(l) $cost(ADCBEA) = 5 + 2 + 5 + 4 + 6 = 22$

Complexidade

- Algoritmo Força Bruta O(V!)

Cidades	Tamanho
5	12
7	360
10	181.440
15	43.589.145.600
20	6,08226E+16

Inviável!!!

História

- Definição (Hamilton e Kirkman, ~1800)
- Icosian Game (Hamilton, 1857)

História

- Forma geral (Karl Menger, 1930)
- Definição do nome Traveling Salesman Problem (TSP) (Whitney)
- 1950-1960 Europa, EUA
  - 49-cidades - Solução ótima por Programação inteira (Dantzig, Fulkerson, Johnson e RAND corp.)
- NP-Completo (Karp, 1972)

História

- 1970-1980
  - 2392-cidades - Cutting-Planes e Branch-and-Bound (Grotschel, et al)
- Implementação - Concorde (Applegate et al, 1990)
- TSPLIB (Reinelt, 1991)
- 33810-cidades do TSLIB (Cook et. al, 2005)

História

- Resultados Concorde (Applegate et al., 2006)

N	Type	Sample size	Mean CPU seconds
100	random	10000	0.7
500	random	10000	50.2
1000	random	1000	601.6
2000	random	1000	14065.6
2500	random	1000	53737.9

Aplicações

- Roteamento de veículos
- Conectar componentes em uma placa mãe (Lenstra e Kan, 1974)

Abordagens

- Soluções Exatas
- Soluções Aproximadas
  - Gap de x% da solução ótima

Soluções Exatas

- sTSP Programação Inteira (Applegate et. al, 2003)

Minimize

$$\sum_{i<j} c_{ij} x_{ij}$$

Subject to

$$\sum_{i<k} x_{ik} + \sum_{j>k} x_{kj} = 2 \qquad (k \in V)$$

$$\sum_{i,j \in S} x_{ij} \leq |S| - 1 \qquad (S \subset V, 3 \leq |S| \leq n-3)$$

$$x_{ij} = 0 \text{ or } 1 \qquad (i,j) \in E$$

Soluções Exatas

- aTSP Programação Inteira (Dantzig et al., 1954)

Minimize

$$\sum c_{ij} x_{ij}$$

Subject to

$$\sum_{j=1}^{n} x_{ij} = 1 \qquad (i \in V, i \neq j)$$

$$\sum_{i=1}^{n} x_{ij} = 1 \qquad (j \in V, j \neq i)$$

$$\sum_{i,j \in S} x_{ij} \leq |S| - 1 \qquad (S \subset V, 2 \leq |S| \leq n-2)$$

$$x_{ij} = 0 \text{ or } 1 \qquad (i,j) \in A$$

Soluções Aproximadas

- sTSP Construtivo (~10-15% ótimo)
  - Vizinho mais próximo
  - Guloso
  - Inserção
  - Rota MST
  - Christofides

Assumimos que

$c_{ij} <= c_{ik} + c_{kj}$

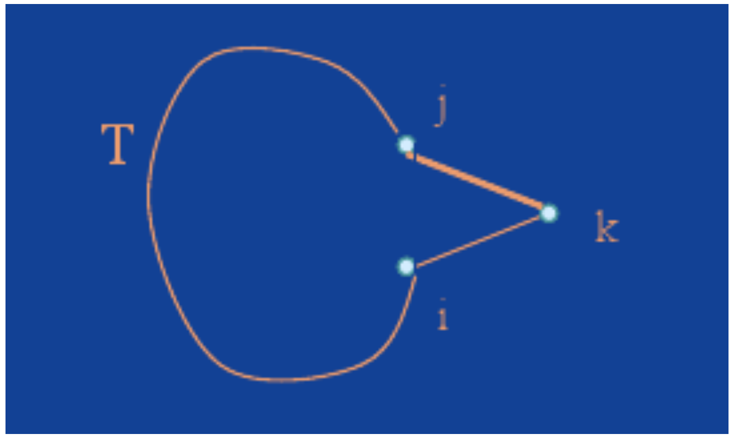
Soluções Aproximadas

- Vizinho mais próximo
  - Escolha uma cidade
  - Vá para a cidade mais próxima que não foi visitada
  - Se existirem cidades a serem visitadas, vá para o passo 1 senão volte retorne para a cidade inicial

Soluções Aproximadas

- Guloso
  - Ordene todas as arestas
  - Escolha a menor das arestas (Não gera ciclo < N e grau do vértice igual a 1)
  - Se número de arestas igual a N então pare. Caso contrário volte para o passo 2.

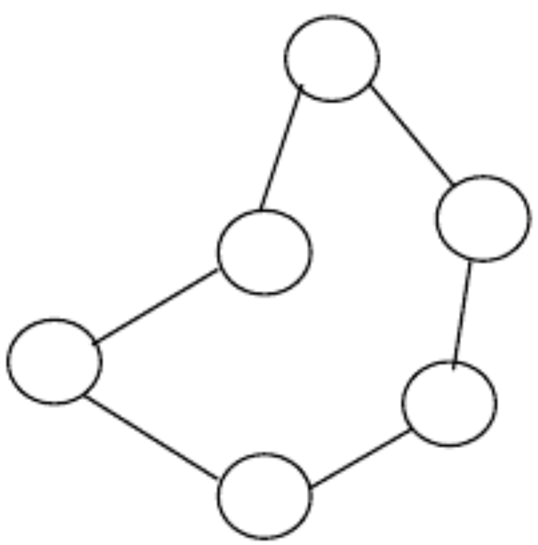
Soluções Aproximadas



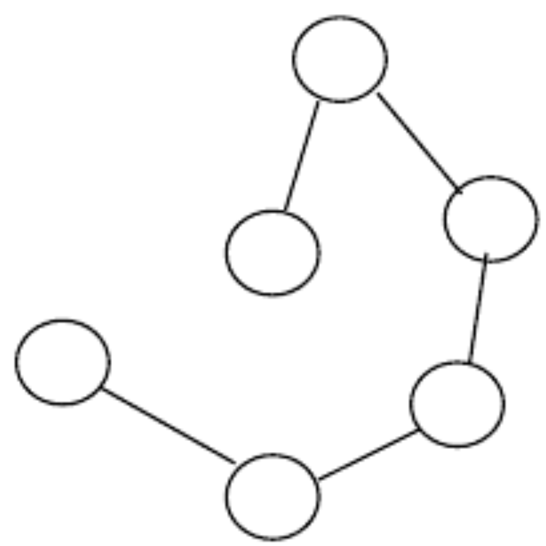
- Inserção
  - Gere uma subrota S
  - Escolha uma cidade k fora da subrota S
  - Conecte k a S de forma que apresente custo mínimo dentre todas as opções de S
  - Repita o passo 2 até que não existam mais cidades restantes.

Soluções Aproximadas

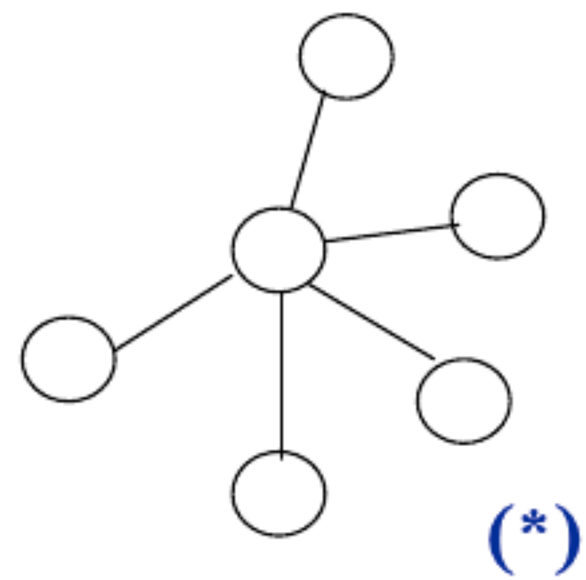
- Árvore Geradora Mínima



sol. ótima



> árvore da sol. ótima



>= árvore ger. mínima

Soluções Aproximadas

- Árvore Geradora Mínima

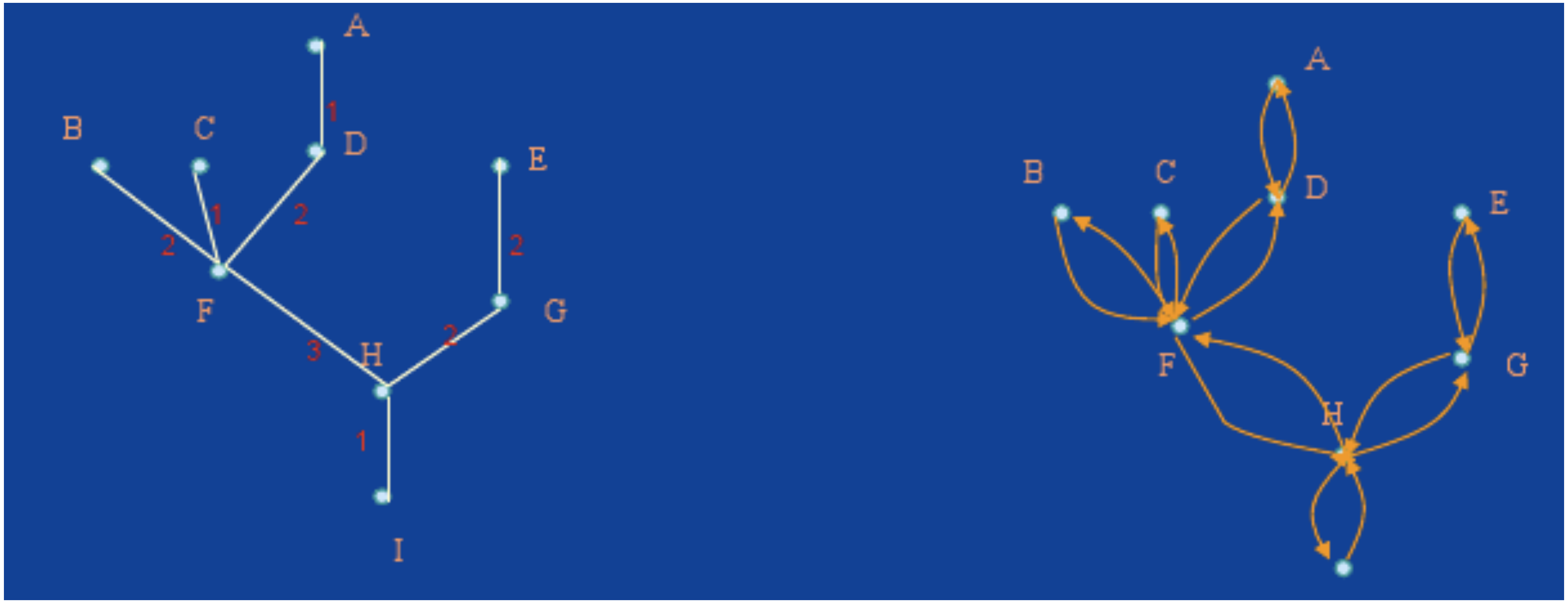
Claim 3.1.2 *The weight of the MST M is less than OPT, the weight of the TSP solution T.*

Proof:

- Take T and remove an edge e. T is now a spanning tree.
- Because M is the MST, $w(M) \leq w(T-e) = w(T) - w(e) = OPT - w(e)$
- $\forall e : w(e) > 0$
- Therefore, $w(M) < OPT$.

Soluções Aproximadas

- Árvore Geradora Mínima (limite inferior)



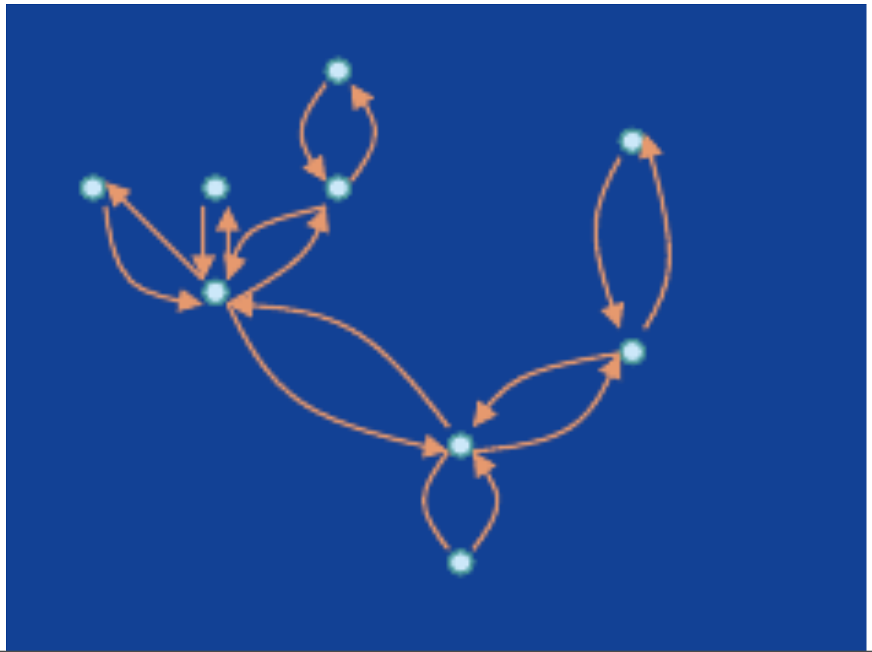
MST

Duplicar MST

2MST <=2opt(TSP)

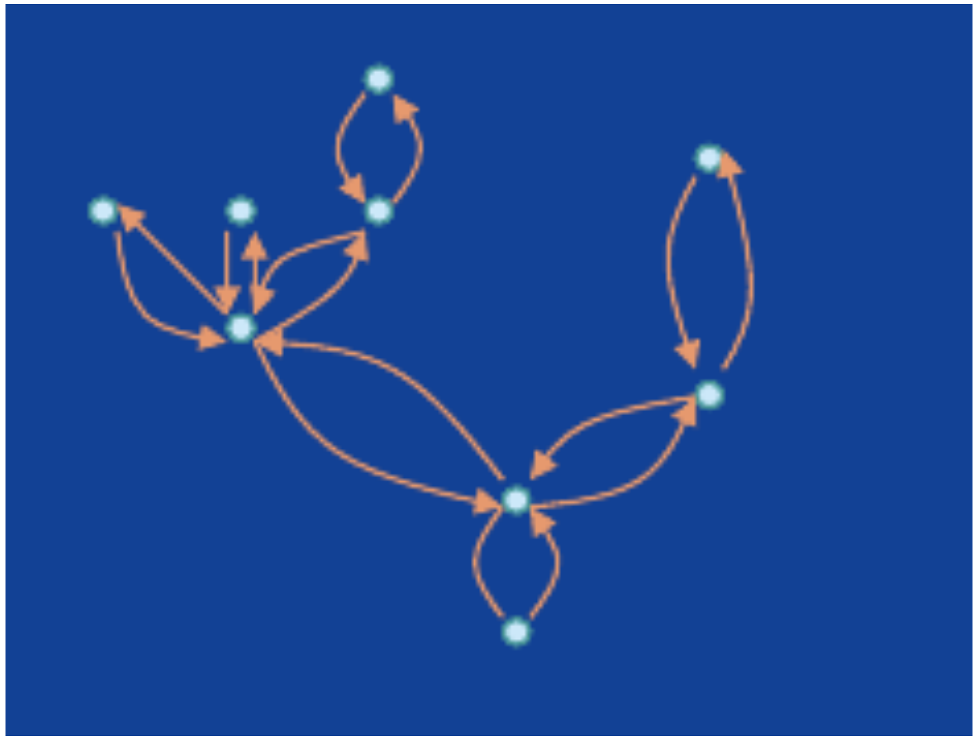
Soluções Aproximadas

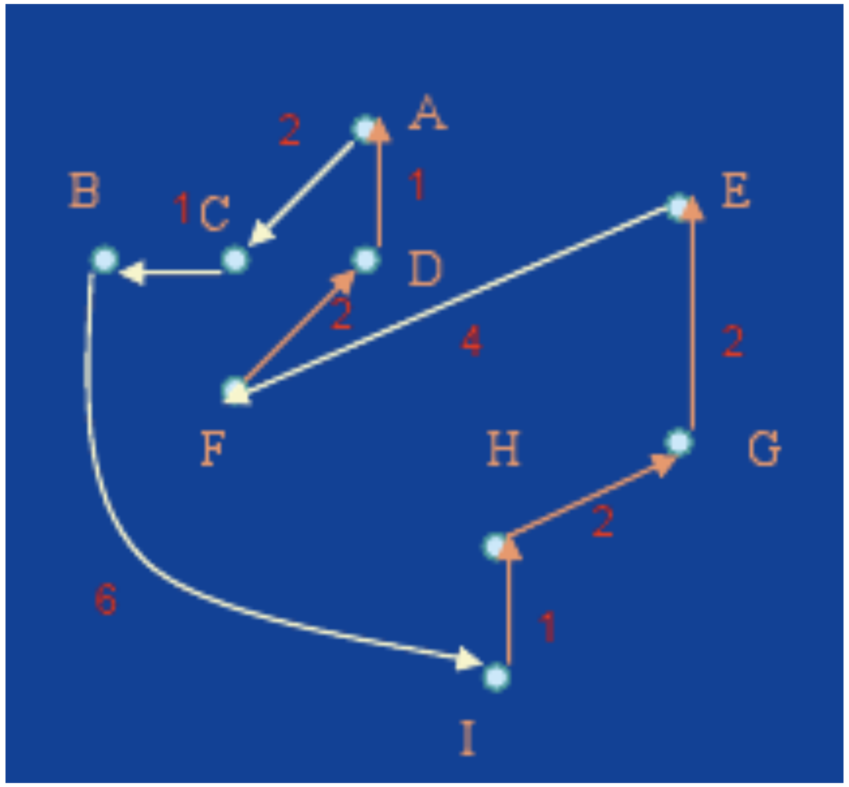
- Árvore Geradora Mínima (limite inferior)
  - Busca em profundidade
    - Ao invés de voltar ao nó pai, vá para o próximo filho não visitado



Soluções Aproximadas

- Rota por Árvore Geradora Mínima





w(AC) < w(CD) + w(DA)

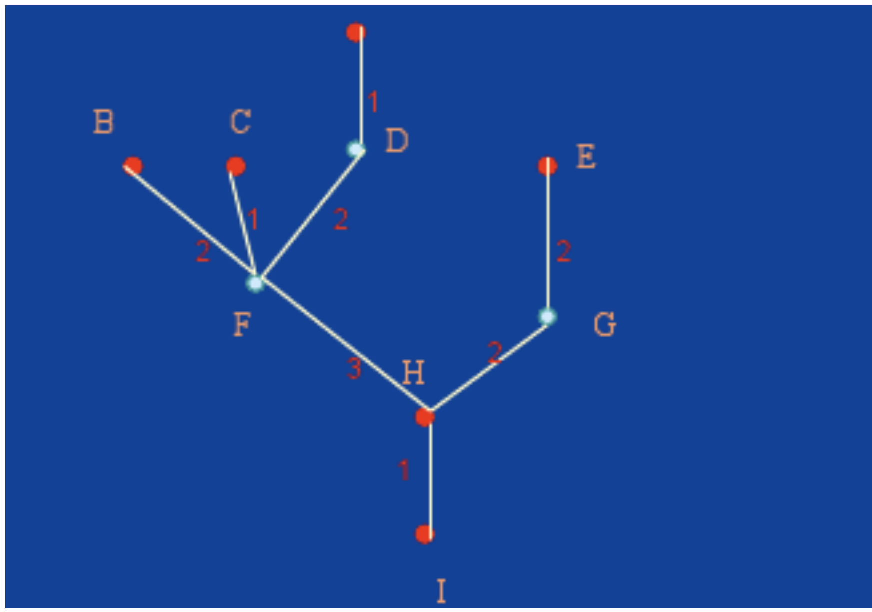
desigualdade triangular

rota(MST) <= 2opt(TSP)

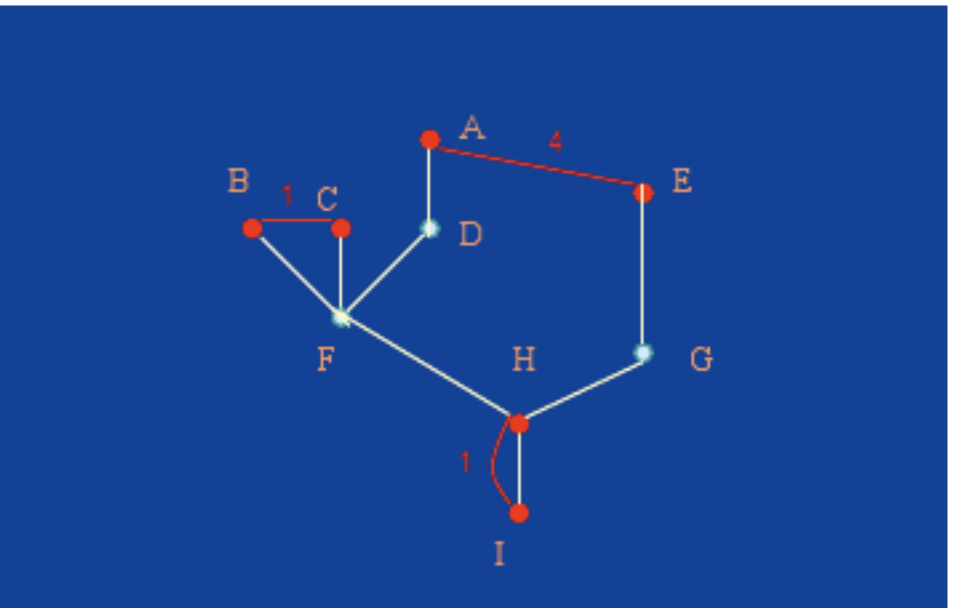
Soluções Aproximadas

- Rota por MST
  - Árvore Geradora Mínima (MST)
  - Duplicar Arestas da MST (G’)
  - Remover visitas redundantes

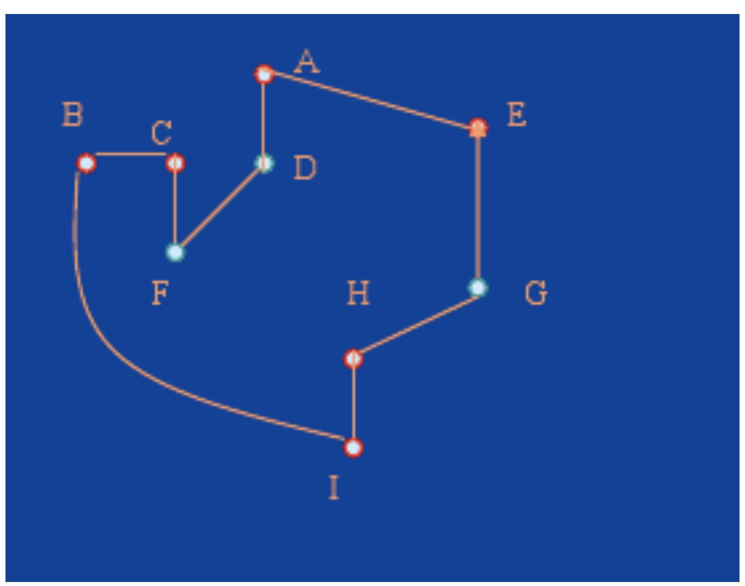
Christofides



MST



Emparelhamento Mínimo



Tour euleriano para TSP

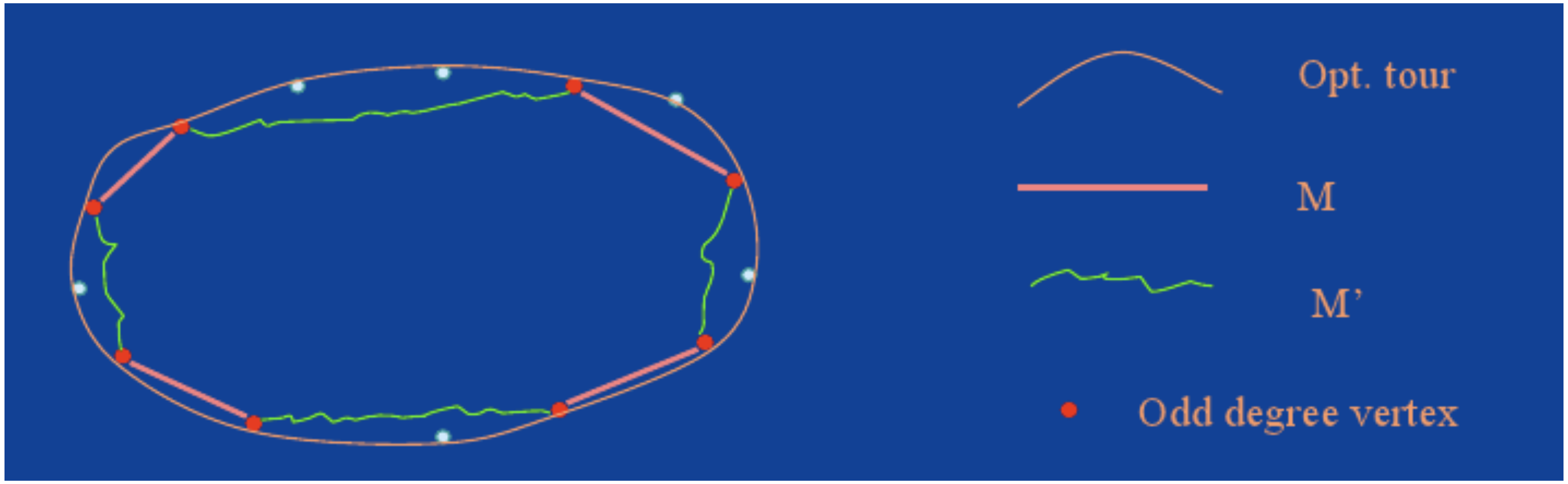
Exemplo gerado no sentido anti-horário

Soluções Aproximadas

- Christofides
  - Árvore Geradora Mínima T
  - Emparelhamento Ponderado Mínimo para os vértices de grau ímpar G’
  - Circuito Euleriano para H = T+G’
    - Evite repetições de cidades

Soluções Aproximadas

- Christofides



w(M) + w(M') <= opt(TSP)

min(w(M) ,w(M')) <= 1/2opt(TSP)

Soluções Aproximadas

- Christofides

$$\min(w(M), w(M')) <= 1/2opt(TSP)$$

$$+$$

$$w(MST) <= opt(TSP)$$

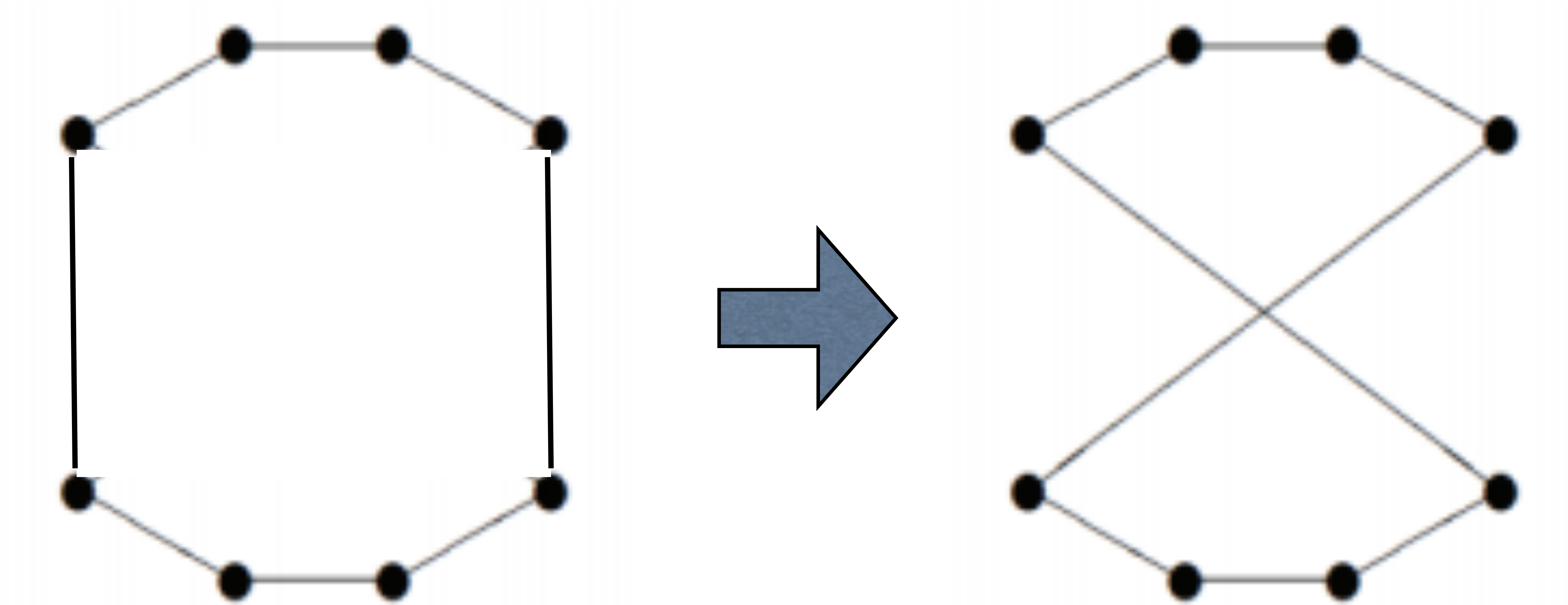
$$w(MST) + \min(w(M), w(M')) <= 3/2opt(TSP)$$

Soluções Aproximadas

- Como podemos melhorar as soluções anteriores a partir Heurísticas (Tour Improvement)
  - Trocas 2-opt, 3-opt, k-opt

Soluções Aproximadas

- 2-opt



custo

custo 2-opt

Soluções Aproximadas

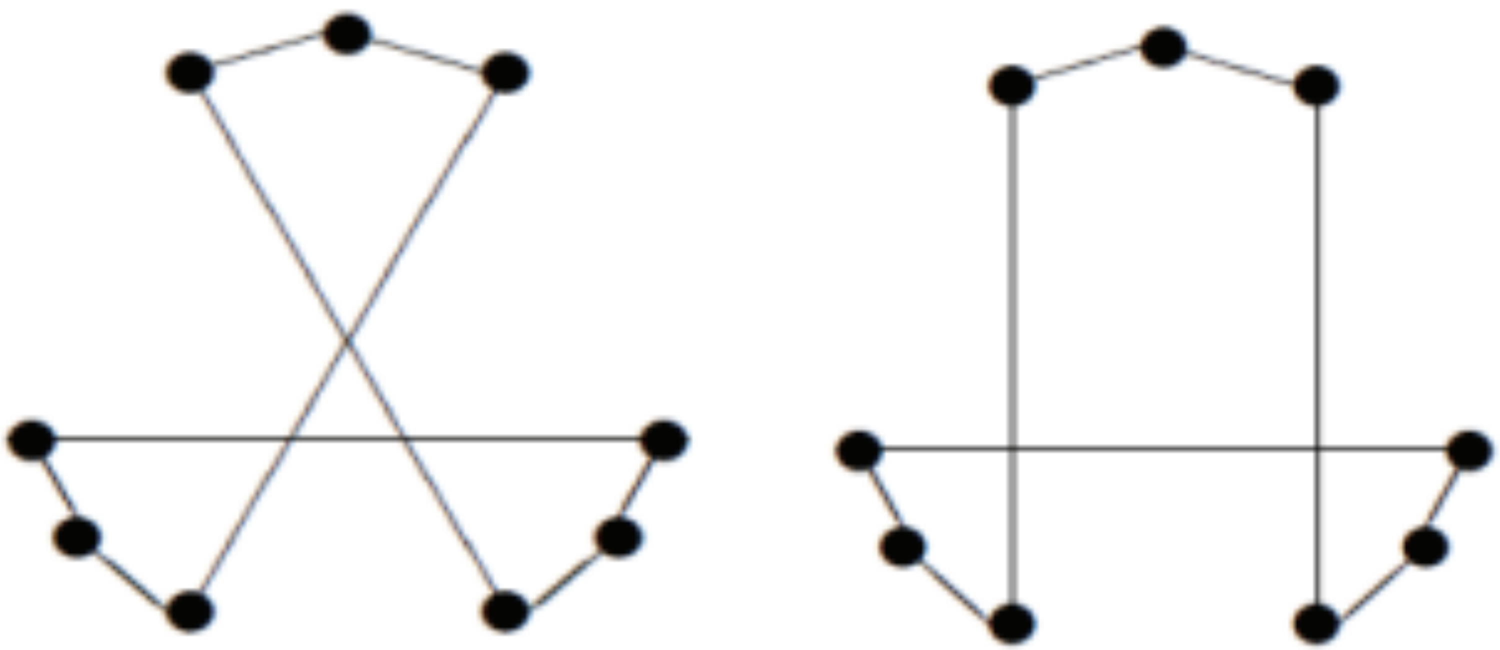
- 2-opt
  - Escolha aleatoriamente duas arestas e remova
  - Se melhora solução, então volte ao passo 1. Caso contrário, pare.

Soluções Aproximadas

- 3-opt
  - Processo semelhante, porém com remoção de três arestas.

Soluções Aproximadas

- 3-opt



custo 3-opt custo 3-opt

Soluções Aproximadas

- Adicionalmente há na literatura
  - Busca Tabu, Algoritmos Genéticos, Simulated Annealing, Colônias de Formiga, ...

Conclusões

- Por mais que não encontremos soluções exatas, podemos encontrar soluções aproximadas (1+e)opt(TSP)
- É interessante fornecer uma garantia de qualidade da solução para problemas práticos

Referências

- Traveling Salesman Problem: An Overview of Applications, Formulations, and Solution Approaches; Rajesh Matai, Surya Prakash Singh2 and Murari Lal Mitta