DCC001 – Análise e Projeto de Algoritmos

Heurísticas

TSP

Material baseado nos slides do Prof. Celso Ribeiro (IC-UFF)

Algoritmos heurísticos

- problema do caixeiro viajante

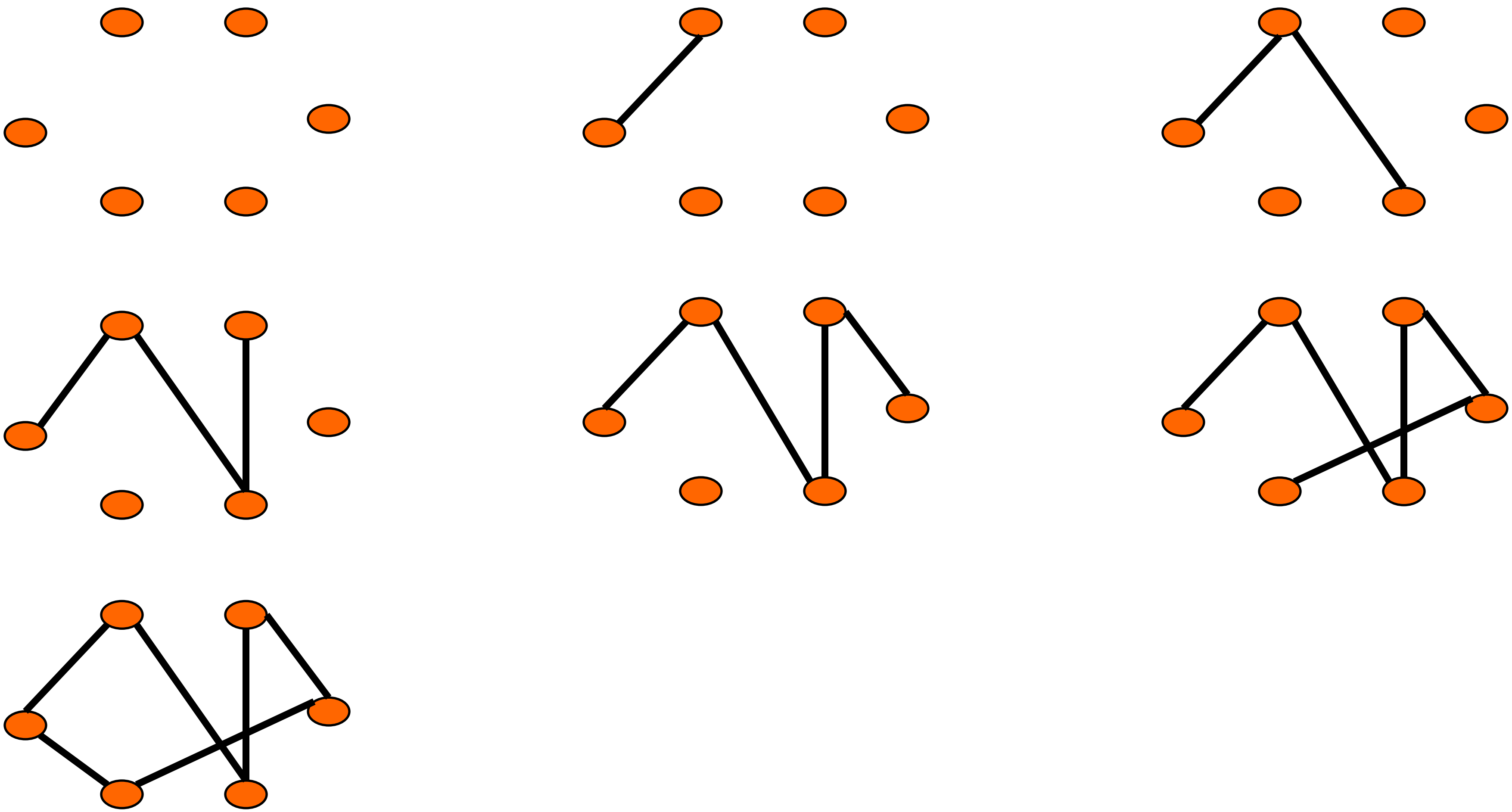
E: conjunto de arestas

F: subconjuntos de arestas que formam um circuito hamiltoniano (isto é, que visita cada cidade exatamente uma única vez)

$c(S) = \Sigma_{e \in S}\, c_e$

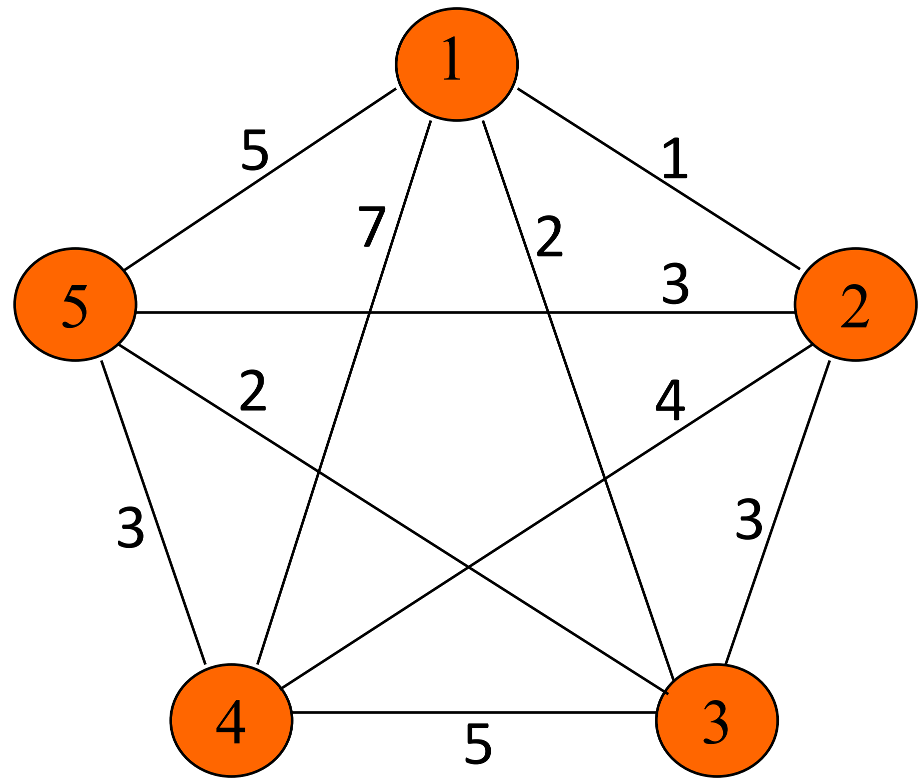
c_e: custo (comprimento, tempo etc) da aresta $e = (i,j)$

Algoritmos construtivos



Problema do caixeiro viajante

Matrix de distâncias c_{ij}
Conjunto de nós N
$|N|=5$



Problema do caixeiro viajante

- Critério do vizinho mais próximo:

Escolher o nó inicial i e fazer $N \leftarrow N-\{i\}$.

Enquanto $N \neq \varnothing$ fazer:

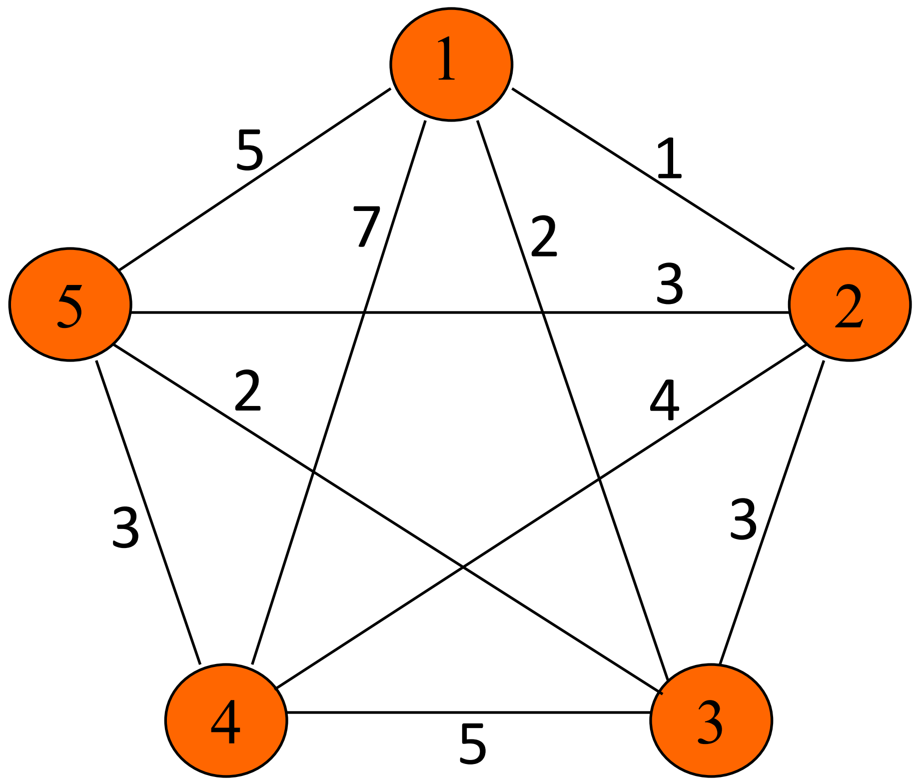
Obter $j \in N$ tal que $c_{i,j} = \min_{k \in N} \{c_{i,k}\}$.

$N \leftarrow N-\{j\}$

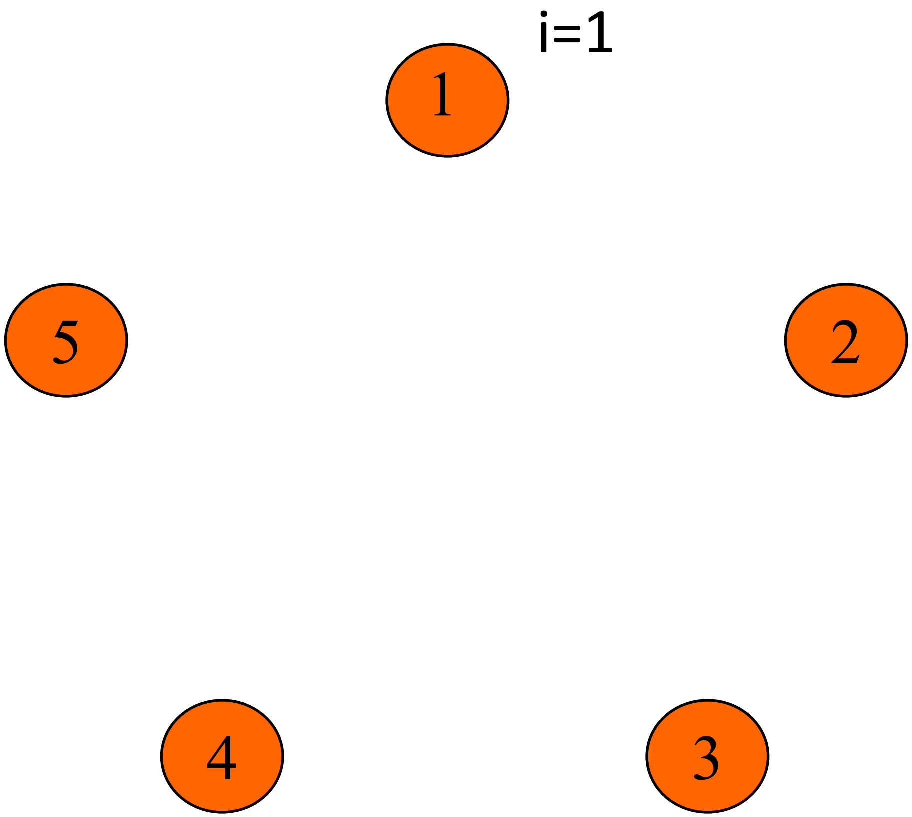
$i \leftarrow j$

Fim-enquanto

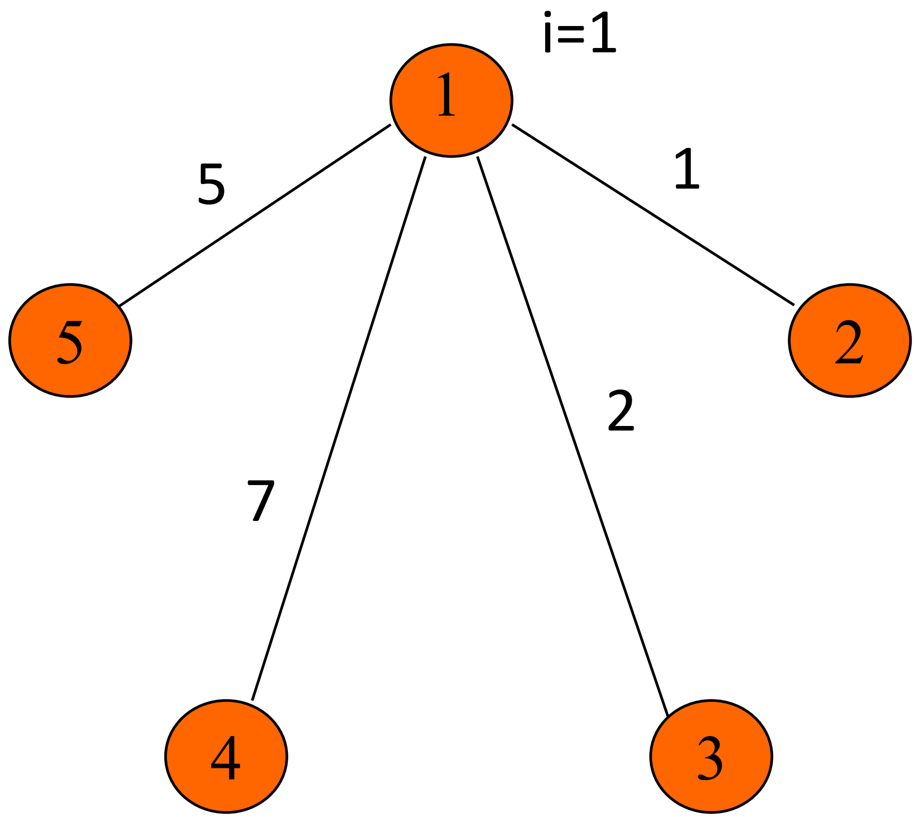
Problema do caixeiro viajante



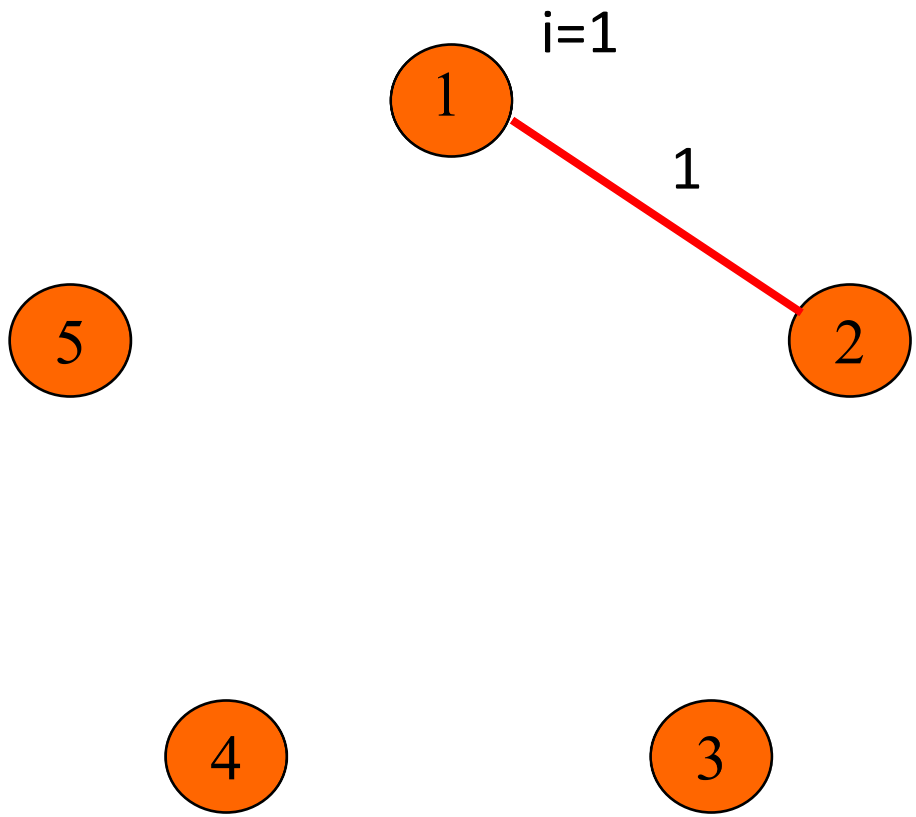
Problema do caixeiro viajante



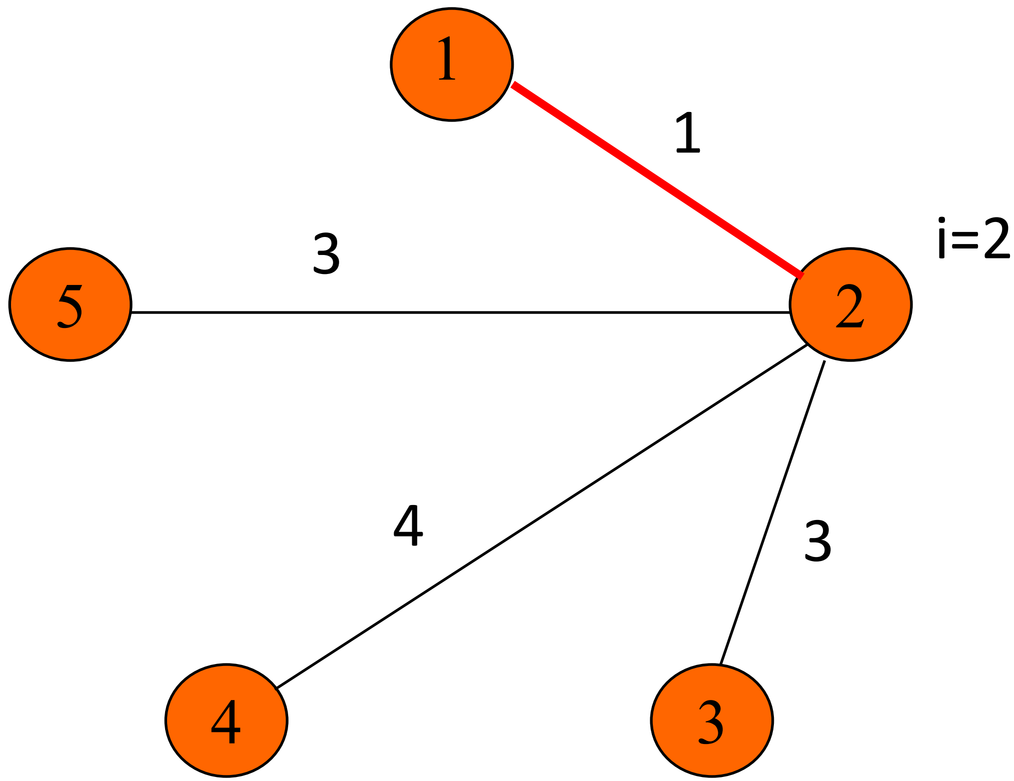
Problema do caixeiro viajante



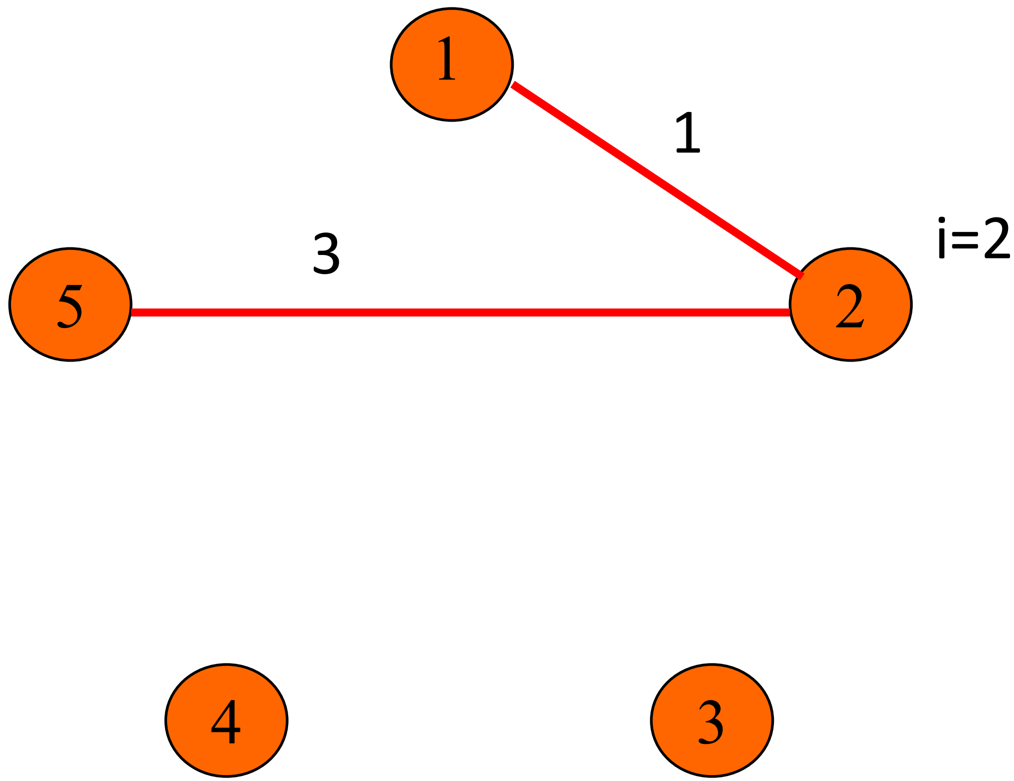
Problema do caixeiro viajante



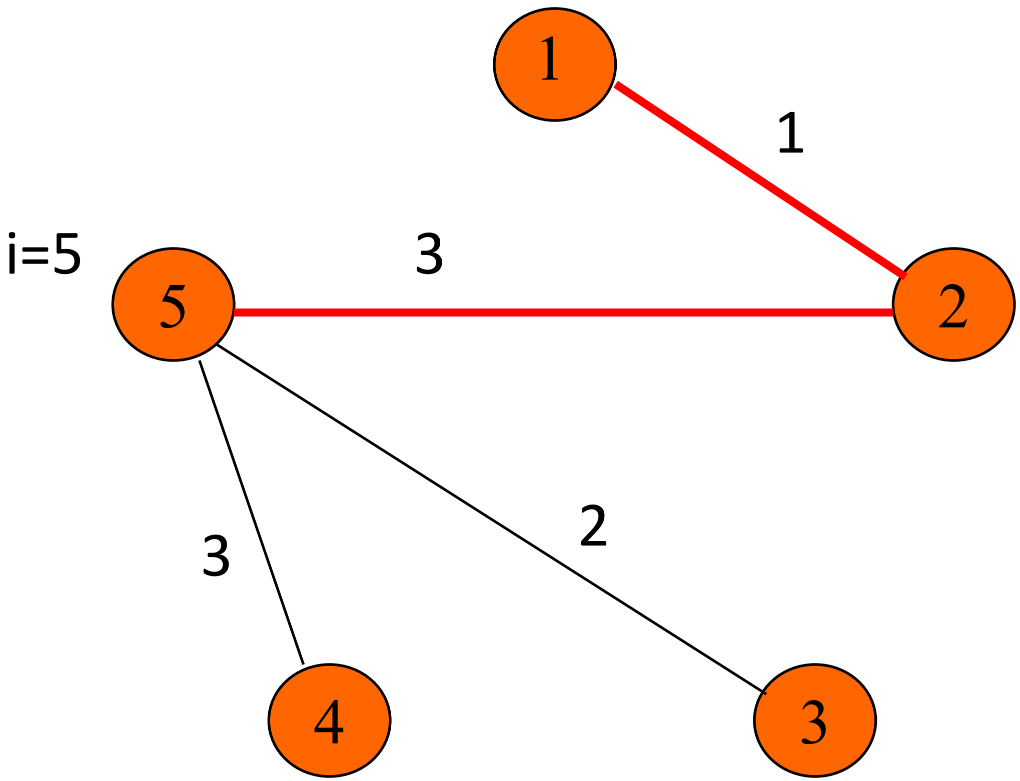
Problema do caixeiro viajante



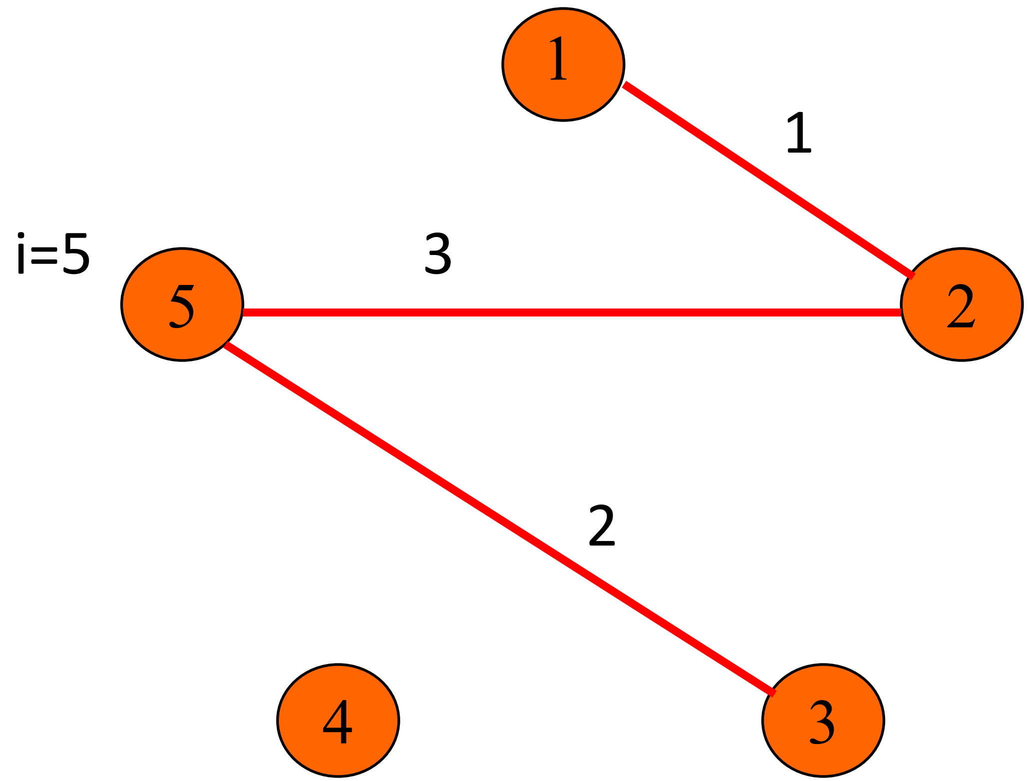
Problema do caixeiro viajante



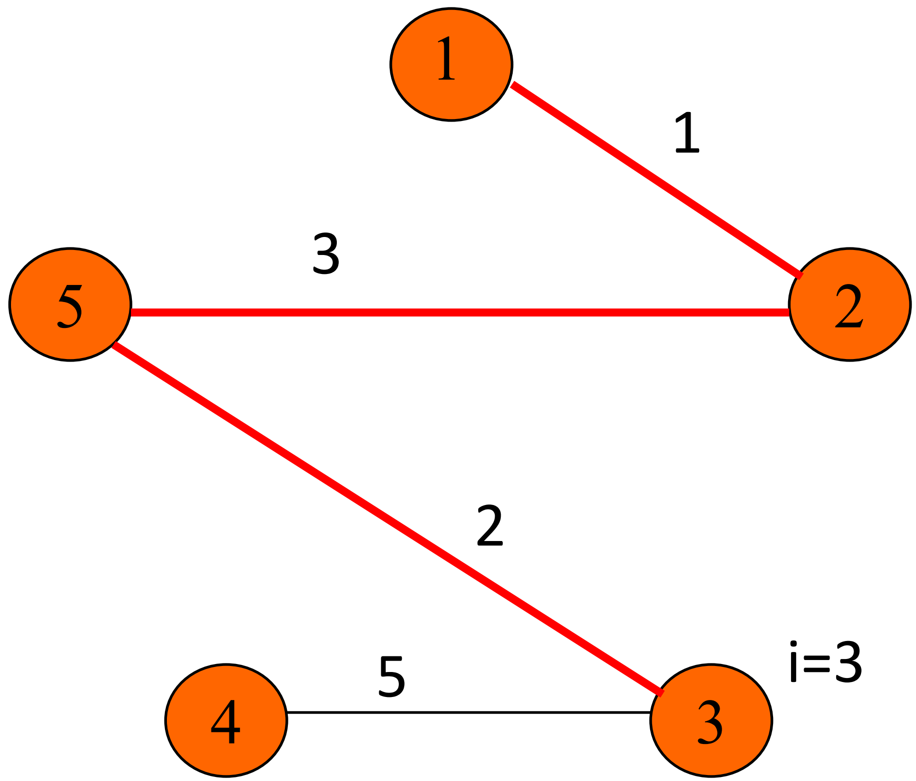
Problema do caixeiro viajante



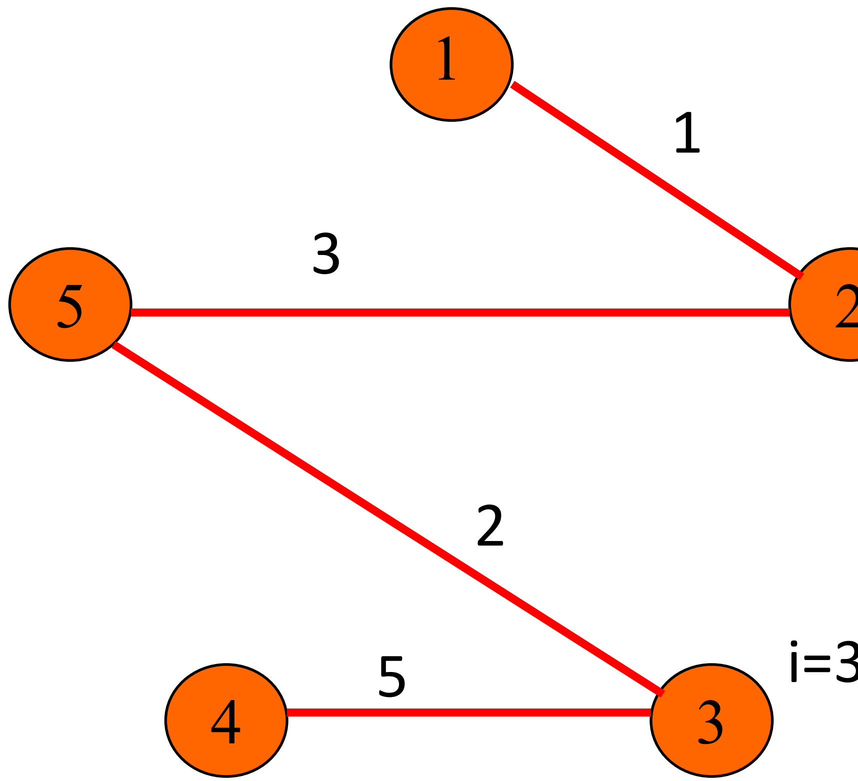
Problema do caixeiro viajante



Problema do caixeiro viajante

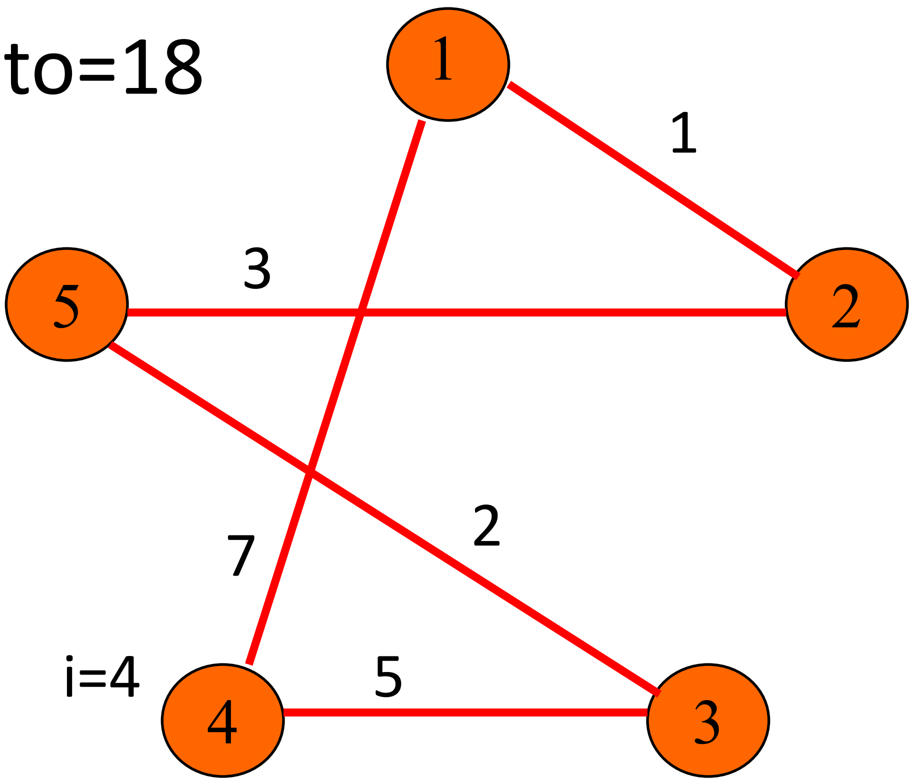


Problema do caixeiro viajante



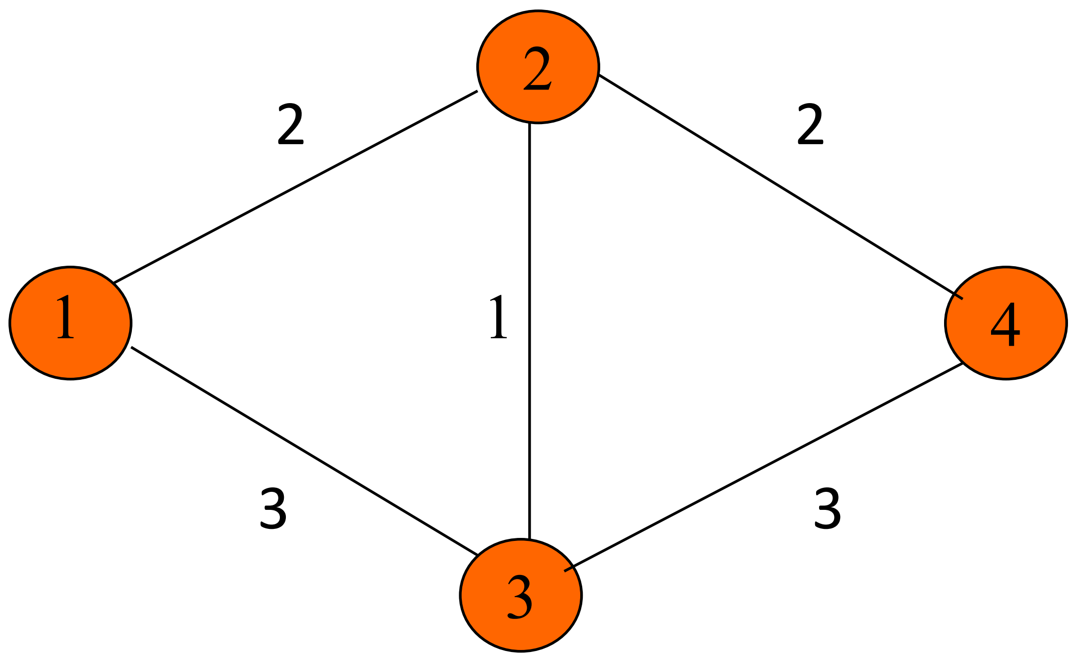
Problema do caixeiro viajante

comprimento=18



Problema do caixeiro viajante

- Algoritmos construtivos simples podem falhar mesmo para casos muito simples:



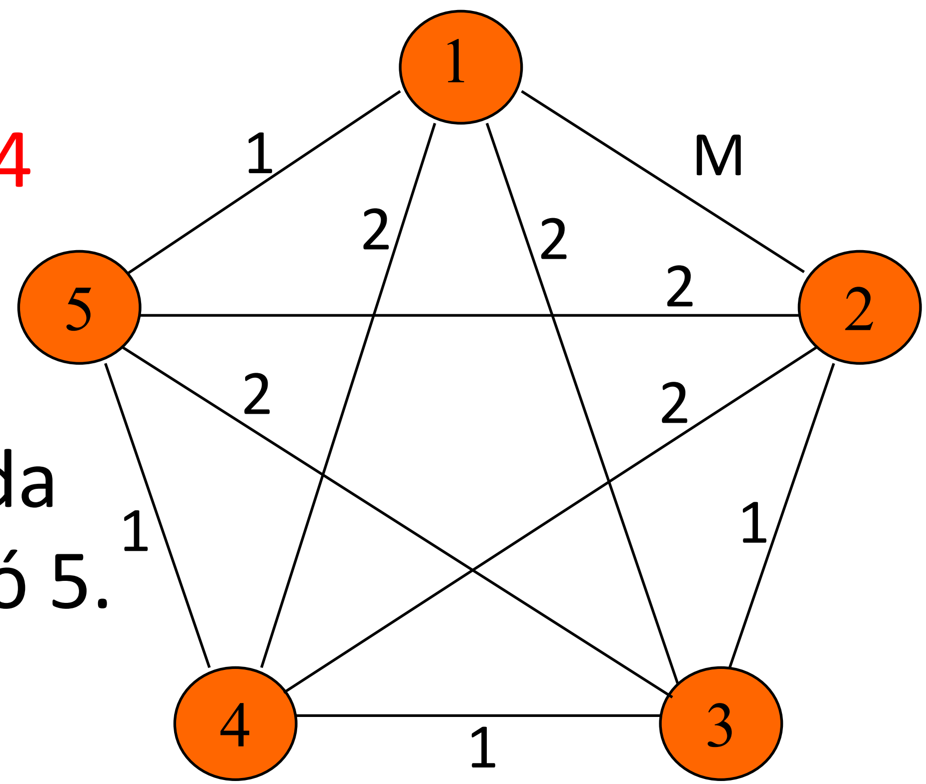
Problema do caixeiro viajante

- Podem encontrar soluções arbitrariamente ruins:

Heurística (1-5-4-3-2-1): M+4

Ótimo (1-5-2-3-4-1): 7

A solução ótima é encontrada se o algoritmo começa do nó 5.



Problema do caixeiro viajante

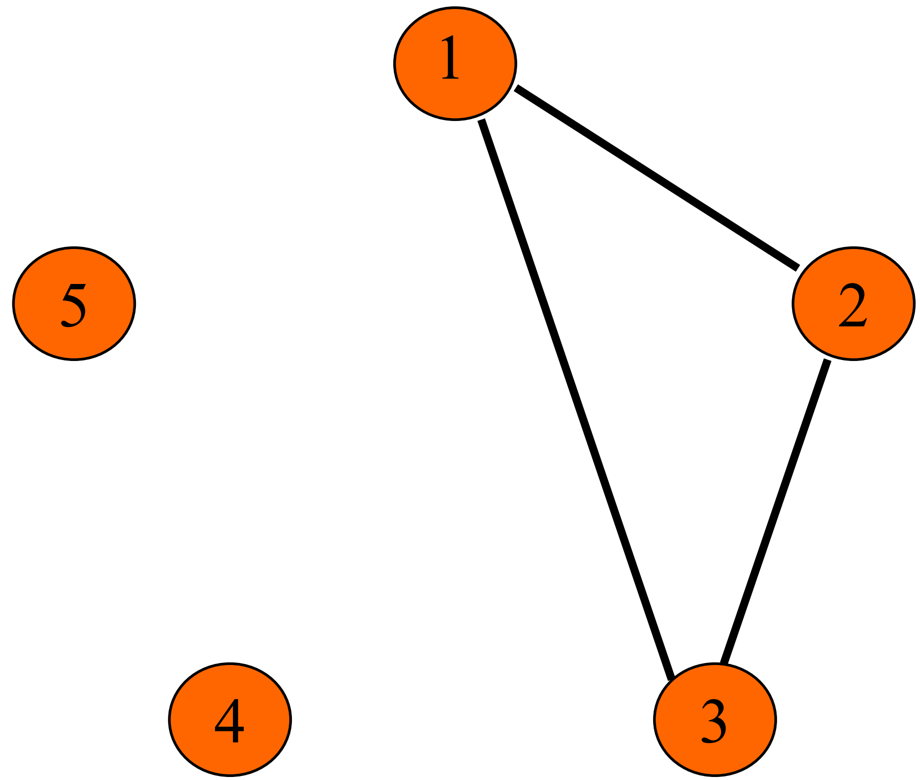
- Extensões e melhorias simples:
  - Repetir a aplicação do algoritmo a partir de cada nó do grafo e selecionar a melhor solução obtida.
    - Melhores soluções, mas tempo de processamento multiplicado por n.
  - A cada iteração selecionar a aresta mais curta a partir de alguma das extremidades em aberto do circuito, e não apenas a partir da última extremidade inserida: solução de comprimento 15 (tempos multiplicados por dois).
  - Critérios mais elaborados para (1) seleção do novo nó incorporado ao circuito a cada iteração e para (2) seleção da posição onde ele entra no circuito: algoritmo baseado no crescimento de um circuito até completá-lo.

Problema do caixeiro viajante

- Escolha do novo nó a ser incorporado ao circuito a cada iteração:
  - Selecionar o nó k fora do circuito parcial corrente, cuja aresta de menor comprimento que o liga a este circuito parcial corrente é mínima $\rightarrow$ algoritmo de inserção mais próxima
  - Selecionar o nó k fora do circuito parcial corrente, cuja aresta de menor comprimento que o liga a este circuito parcial corrente é máxima $\rightarrow$ algoritmo de inserção mais afastada

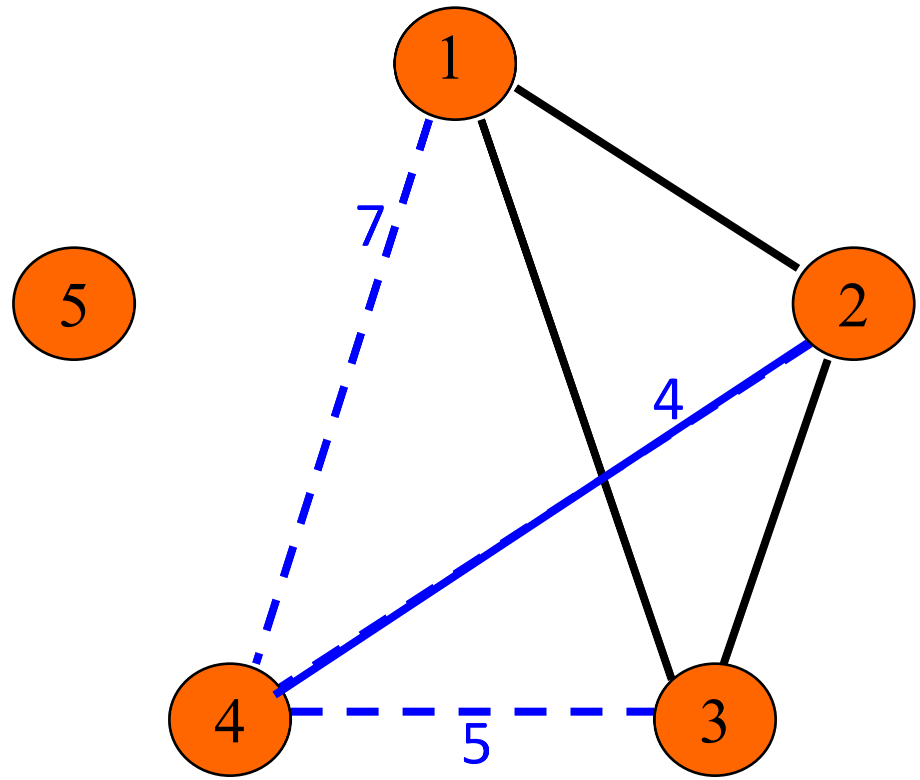
Problema do caixeiro viajante

- Escolha do novo nó a ser incorporado ao circuito a cada iteração pela inserção mais próxima
- Seleção do nó:



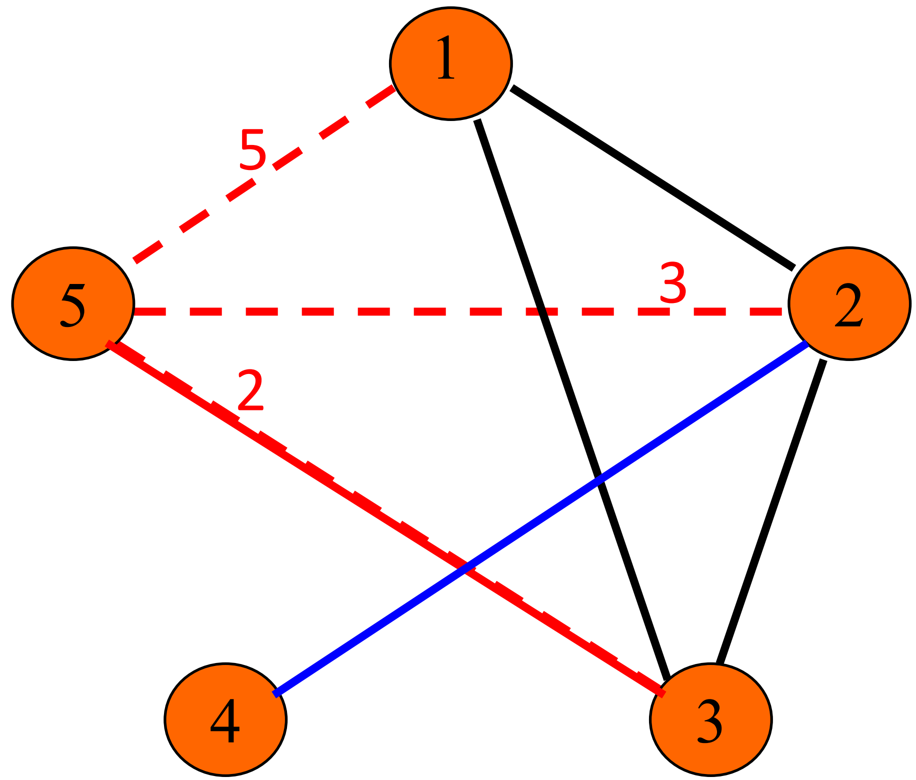
Problema do caixeiro viajante

- Escolha do novo nó a ser incorporado ao circuito a cada iteração pela inserção mais próxima
- Seleção do nó:
  - Distância mínima do nó 4: 4



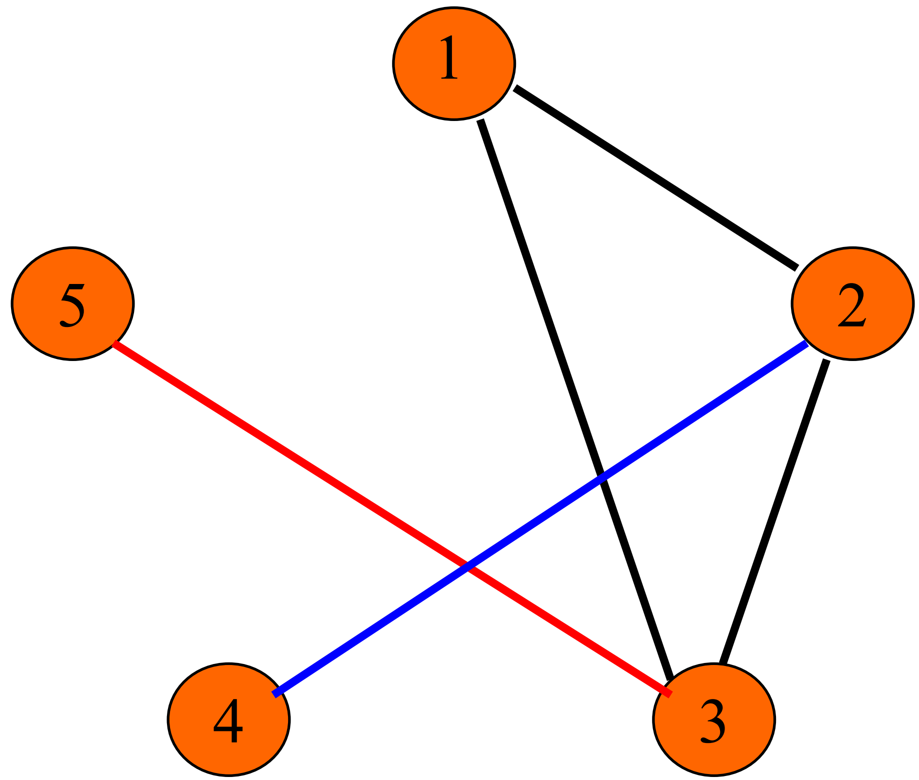
Problema do caixeiro viajante

- Escolha do novo nó a ser incorporado ao circuito a cada iteração pela inserção mais próxima
- Seleção do nó:
  - Distância mínima do nó 4: 4
  - Distância mínima do nó 5: 2



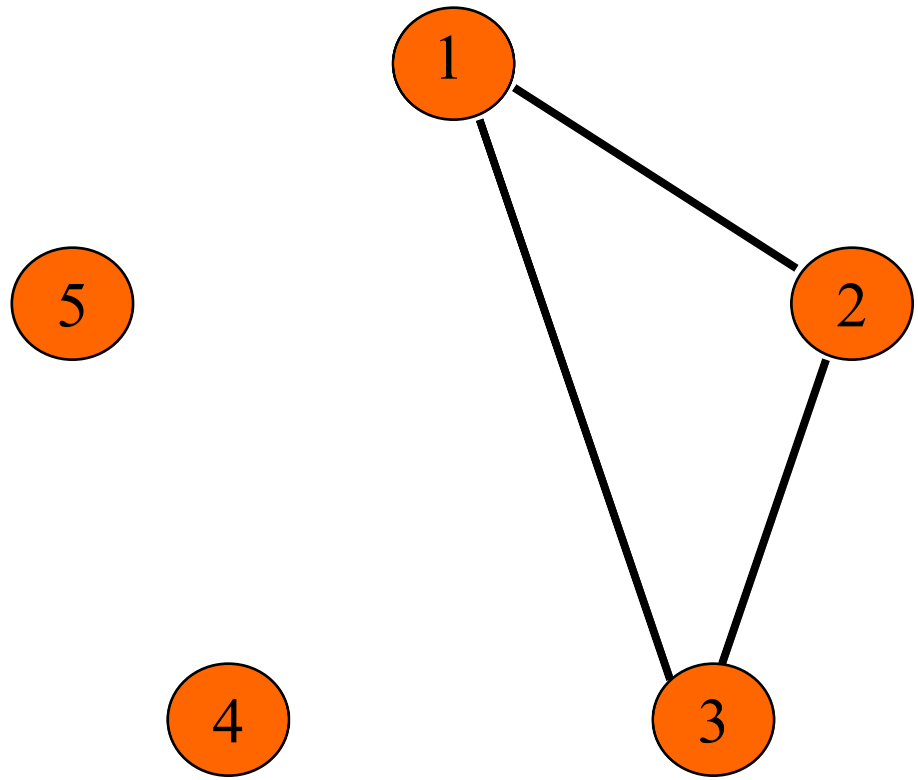
Problema do caixeiro viajante

- Escolha do novo nó a ser incorporado ao circuito a cada iteração pela inserção mais próxima
- Seleção do nó:
  - Distância mínima do nó 4: 4
  - Distância mínima do nó 5: 2
  - Nó selecionado: 5



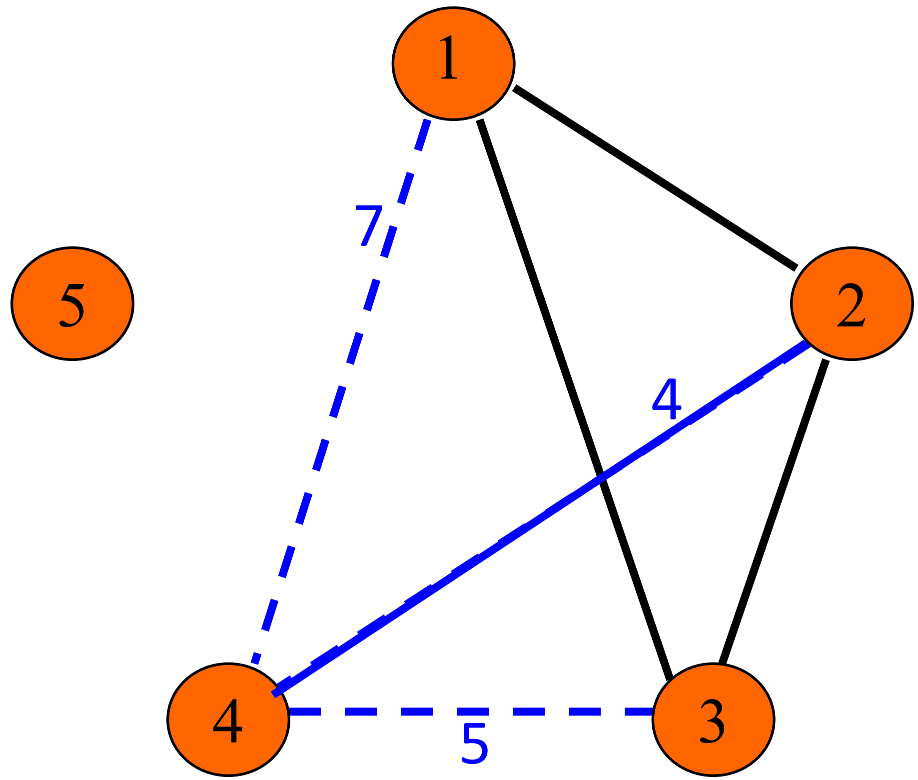
Problema do caixeiro viajante

- Escolha do novo nó a ser incorporado ao circuito a cada iteração pela inserção mais afastada
- Seleção do nó:



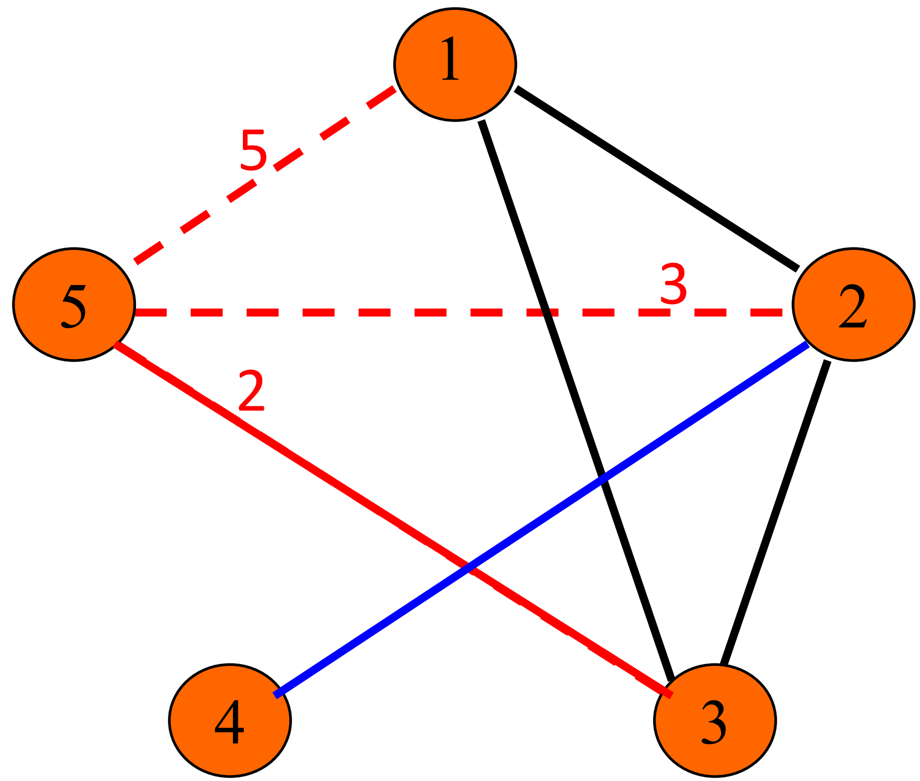
Problema do caixeiro viajante

- Escolha do novo nó a ser incorporado ao circuito a cada iteração pela inserção mais afastada
- Seleção do nó:
  - Distância mínima do nó 4: 4



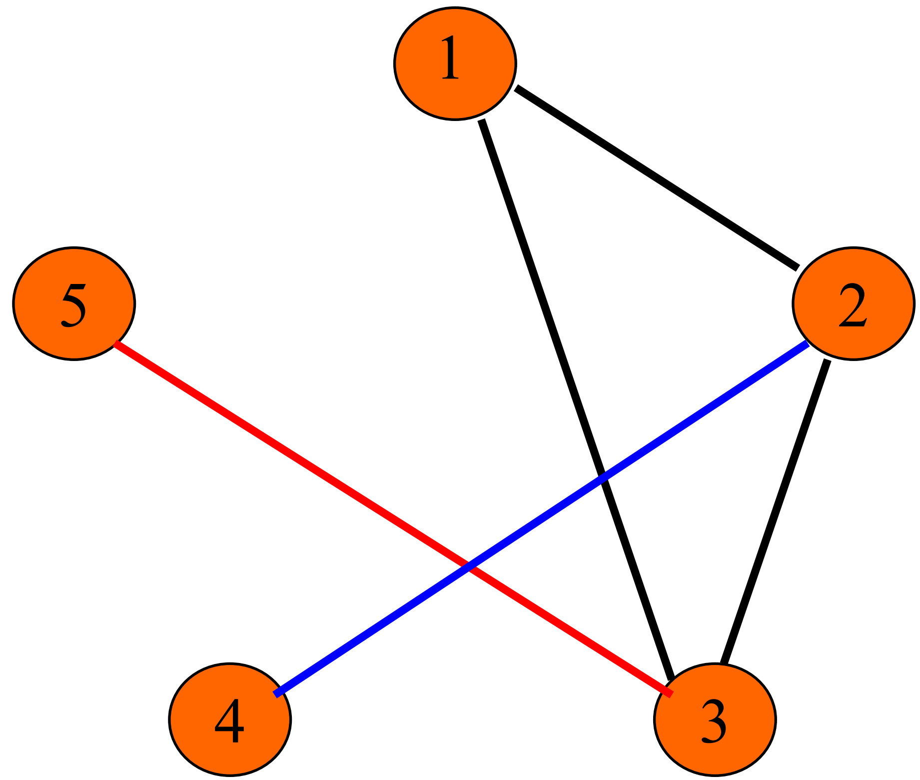
Problema do caixeiro viajante

- Escolha do novo nó a ser incorporado ao circuito a cada iteração pela inserção mais afastada
- Seleção do nó:
  - Distância mínima do nó 4: 4
  - Distância mínima do nó 5: 2



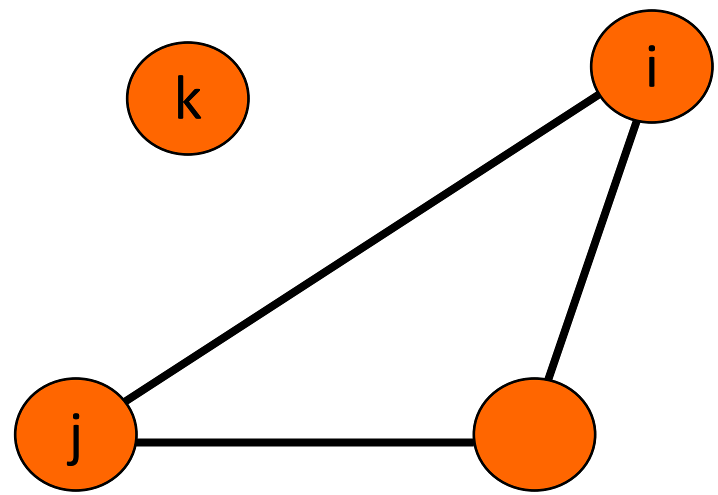
Problema do caixeiro viajante

- Escolha do novo nó a ser incorporado ao circuito a cada iteração pela inserção mais afastada
- Seleção do nó:
  - Distância mínima do nó 4: 4
  - Distância mínima do nó 5: 2
  - Nó selecionado: 4



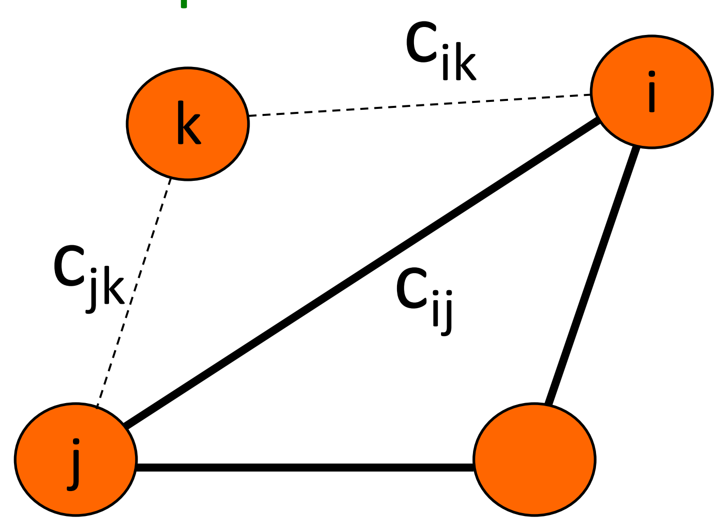
Problema do caixeiro viajante

- Escolha da posição onde o nó selecionado entra no circuito:
  - Suponha a entrada do nó k entre o nó i e o nó j.
  - Devem ser inseridas as areastas (i,k) e (j,k) no circuito parcial corrente, substituindo a aresta (i,j).
  - A variação no comprimento do circuito parcial corrente é dada por $\Delta_{ij}(k) = c_{ik} + c_{jk} - c_{ij}$.



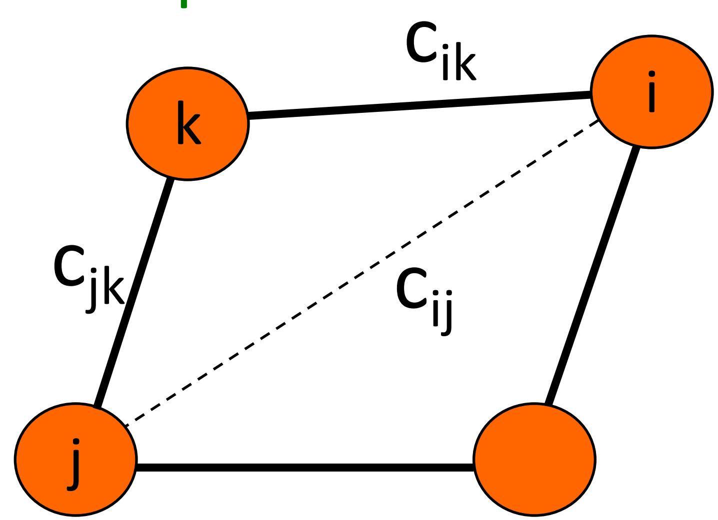
Problema do caixeiro viajante

- Escolha da posição onde o nó selecionado entra no circuito:
  - Suponha a entrada do nó k entre o nó i e o nó j.
  - Devem ser inseridas as areastas (i,k) e (j,k) no circuito parcial corrente, substituindo a aresta (i,j).
  - A variação no comprimento do circuito parcial corrente é dada por $\Delta_{ij}(k) = c_{ik} + c_{jk} - c_{ij}$.



Problema do caixeiro viajante

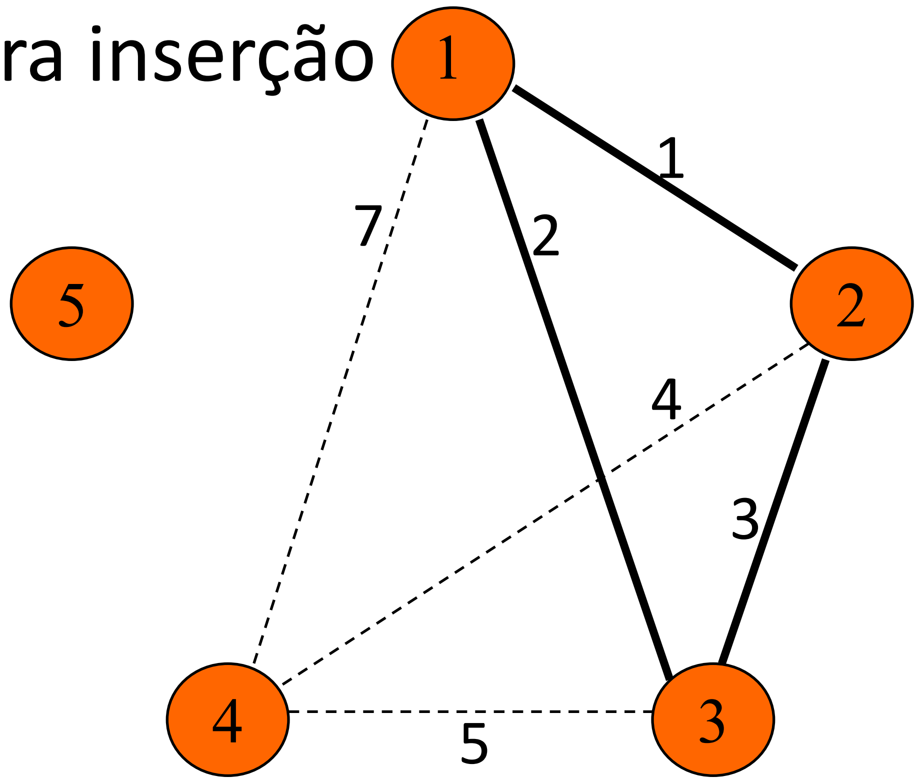
- Escolha da posição onde o nó selecionado entra no circuito:
  - Suponha a entrada do nó k entre o nó i e o nó j.
  - Devem ser inseridas as areastas (i,k) e (j,k) no circuito parcial corrente, substituindo a aresta (i,j).
  - A variação no comprimento do circuito parcial corrente é dada por $\Delta_{ij}(k) = c_{ik} + c_{jk} - c_{ij}$.



Problema do caixeiro viajante

- Escolha da posição onde o nó selecionado entra no circuito parcial corrente
- Exemplo: nó 4 escolhido para inserção

$\Delta_{12}(4) = 7+4-1 = 10$
$\Delta_{13}(4) = 7+5-2 = 10$
$\Delta_{23}(4) = 5+4-3 = 6$



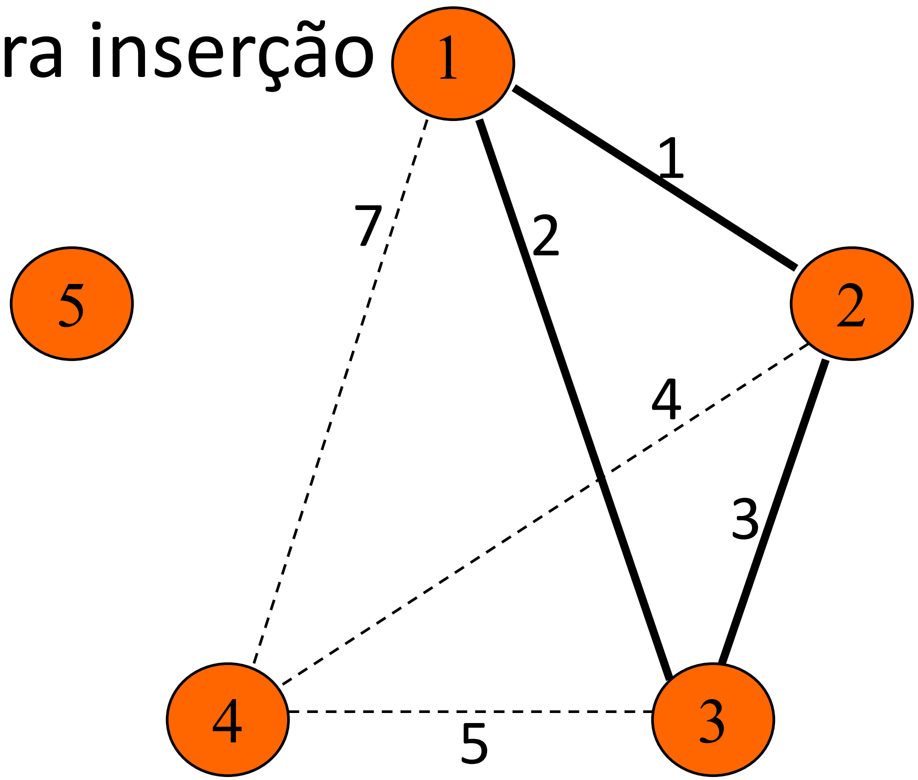
Problema do caixeiro viajante

- Escolha da posição onde o nó selecionado entra no circuito parcial corrente
- Exemplo: nó 4 escolhido para inserção

$\Delta_{12}(4) = 7+4-1 = 10$

$\Delta_{13}(4) = 7+5-2 = 10$

$\Delta_{23}(4) = 5+4-3 = 6$

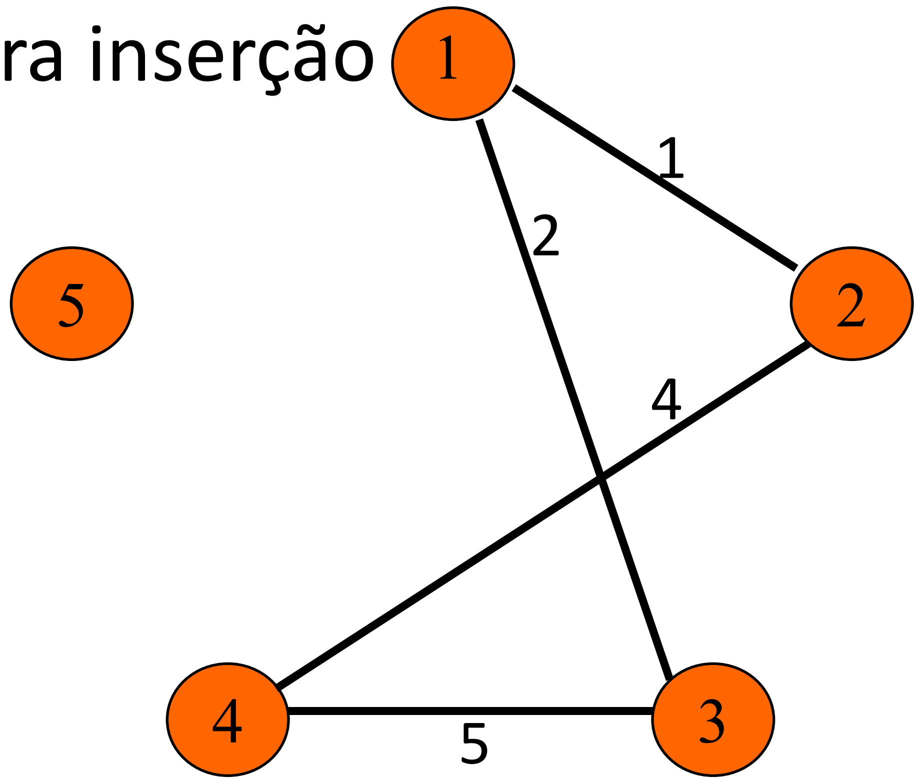


- Inserção mais próxima: entre os nós 2 e 3

Problema do caixeiro viajante

- Escolha da posição onde o nó selecionado entra no circuito parcial corrente
- Exemplo: nó 4 escolhido para inserção

$\Delta_{12}(4) = 7+4-1 = 10$
$\Delta_{13}(4) = 7+5-2 = 10$
$\Delta_{23}(4) = 5+4-3 = 6$



- Inserção mais próxima: entre os nós 2 e 3

Problema do caixeiro viajante

- Algoritmo de inserção mais próxima:

Escolher o nó inicial i, inicializar um circuito apenas com o nó i e fazer $N \leftarrow N-\{i\}$.

Enquanto $N \neq \emptyset$ fazer:

Encontrar o vértice k fora do circuito corrente cuja aresta de menor comprimento que o liga a ele é mínima.

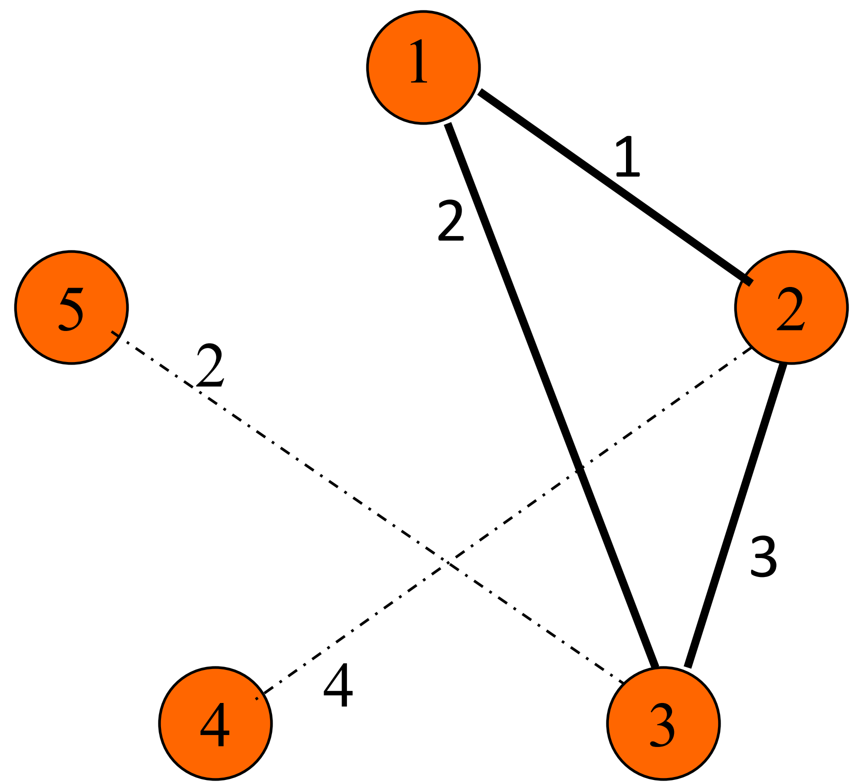
Encontrar o par de arestas (i,k) e (j,k) que ligam o vértice k ao ciclo minimizando $\Delta_{ij}(k) = c_{ik} + c_{jk} - c_{ij}$.

Inserir as arestas (i,k) e (j,k) e retirar a aresta (i,j).

Fazer $N \leftarrow N-\{k\}$.

Fim-enquanto

Problema do caixeiro viajante

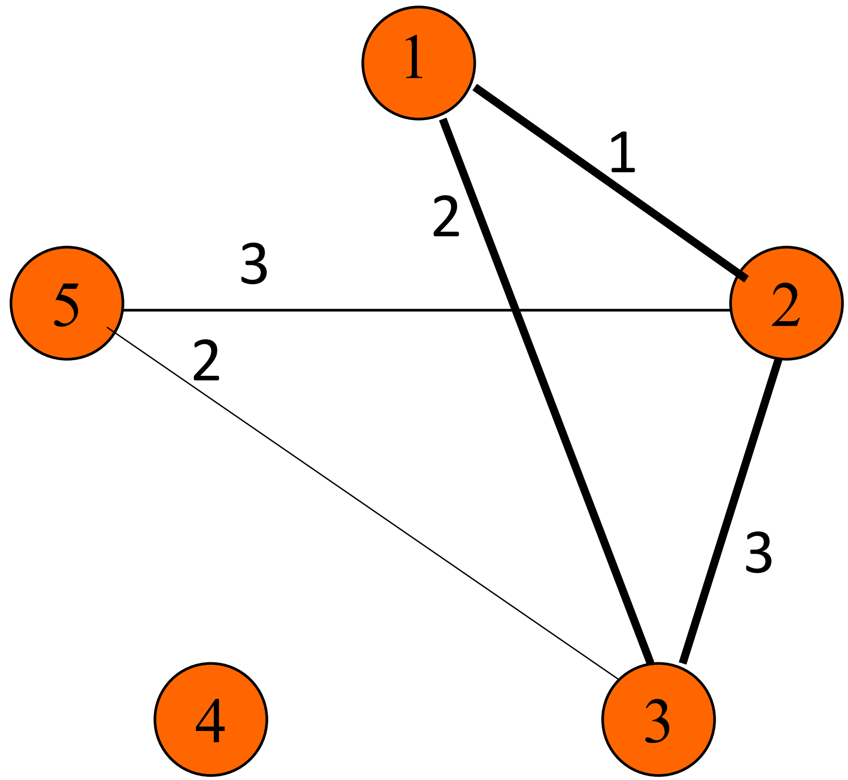


Problema do caixeiro viajante

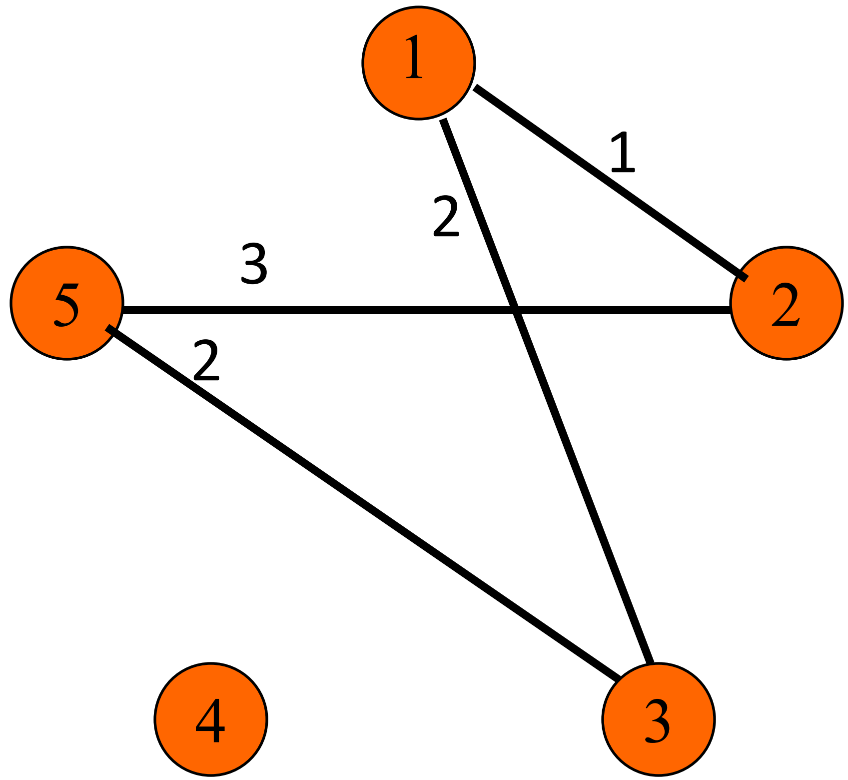
$\Delta_{12}(5) = 5+3-1 = 7$

$\Delta_{23}(5) = 2+3-3 = 2$

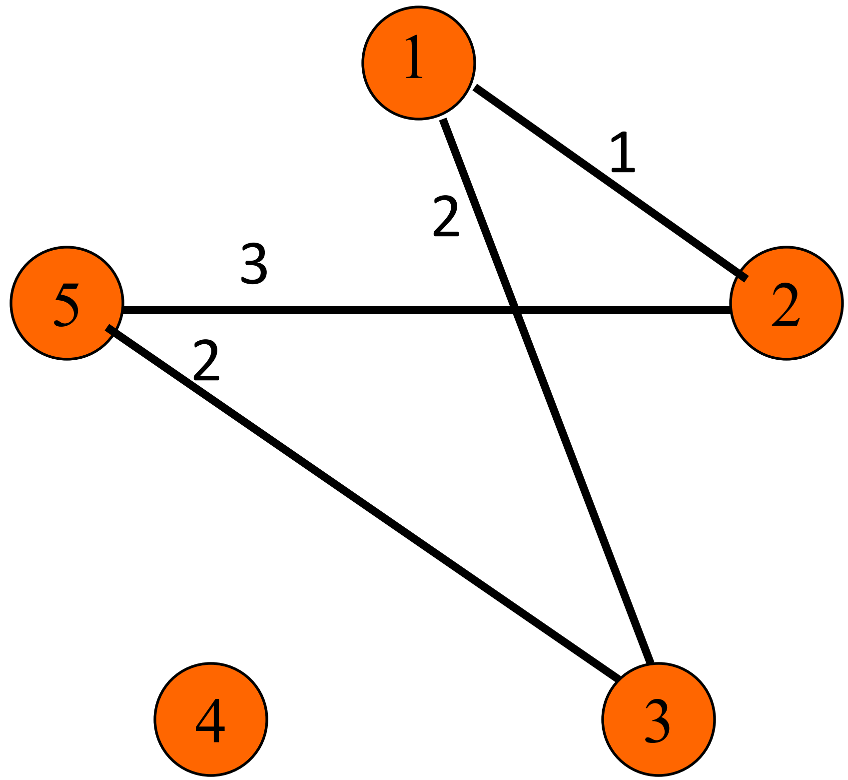
$\Delta_{13}(5) = 2+5-2 = 5$



Problema do caixeiro viajante



Problema do caixeiro viajante



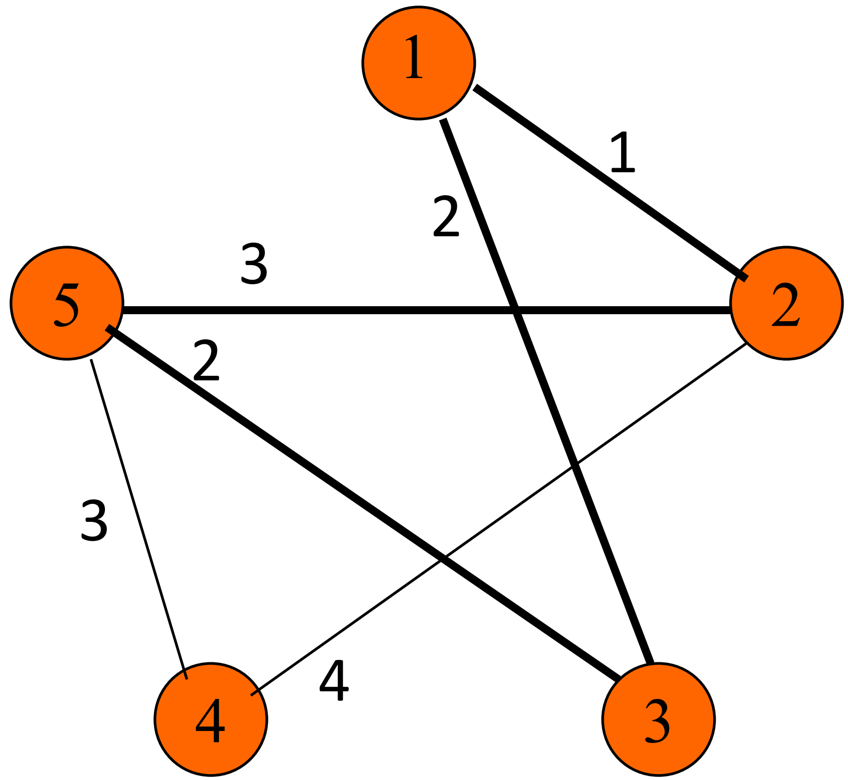
$\Delta_{12}(4) = 7+4-1 = 10$

$\Delta_{13}(4) = 7+5-2 = 10$

$\Delta_{35}(4) = 5+3-2 = 6$

$\Delta_{25}(4) = 4+3-3 = 4$

Problema do caixeiro viajante



$\Delta_{12}(4) = 7+4-1 = 10$

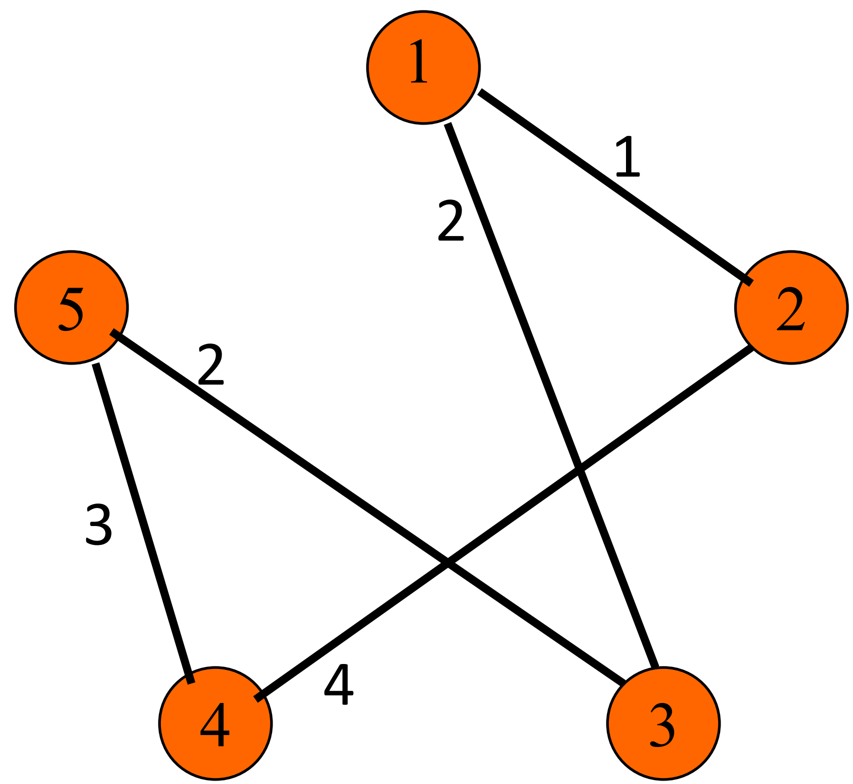
$\Delta_{13}(4) = 7+5-2 = 10$

$\Delta_{35}(4) = 5+3-2 = 6$

$\Delta_{25}(4) = 4+3-3 = 4$

Problema do caixeiro viajante

comprimento = 12



Problema do caixeiro viajante

- Comparação: na prática, o método de inserção mais afastada alcança melhores resultados do que o de inserção mais próxima.
- Melhoria simples: método de inserção mais barata
  - Por que separar em dois passos (1) a seleção do novo nó incorporado ao circuito a cada iteração e (2) a seleção da posição onde ele entra no circuito?
  - Fazer a escolha da melhor combinação em conjunto.
  - Melhores soluções, mas tempos de processamento maiores (cerca de n vezes maiores).

Problema do caixeiro viajante

- Algoritmo de inserção mais barata:

Escolher o nó inicial i, inicializar um circuito apenas com o nó i e fazer $N \leftarrow N-\{i\}$.

Enquanto $N \neq \varnothing$ fazer:

Encontrar o vértice k fora do circuito corrente e o par de arestas (i,k) e (j,k) que ligam o vértice k ao ciclo minimizando $\Delta_{ij}(k) = c_{ik} + c_{jk} - c_{ij}$.

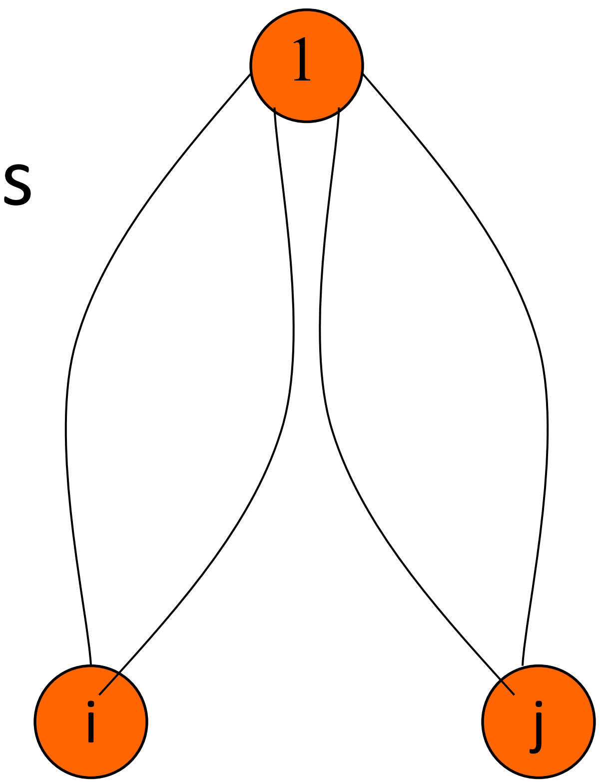
Inserir as arestas (i,k) e (j,k) e retirar a aresta (i,j).

Fazer $N \leftarrow N-\{k\}$.

Fim-enquanto

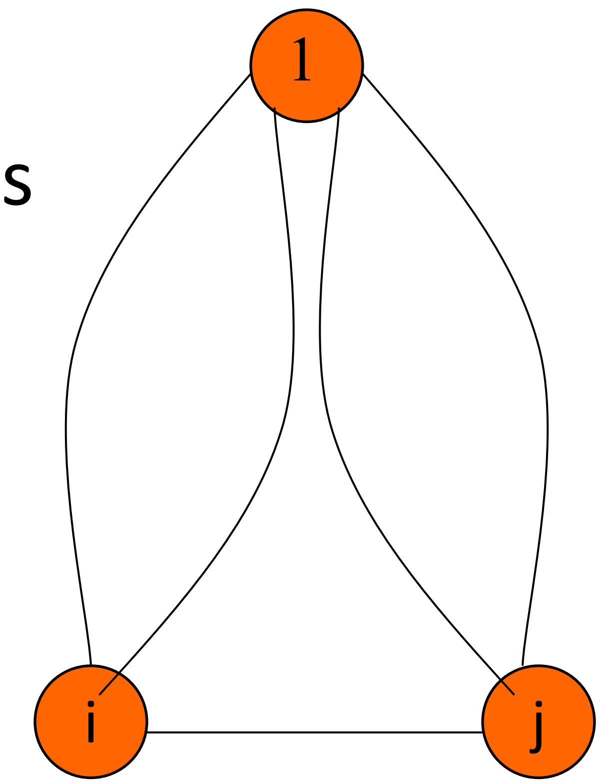
Problema do caixeiro viajante

- Outra idéia diferente: considerar a fusão de subcircuitos
- Considerar dois subcircuitos passando pelo nó 1 e pelos nós i e j.
- Conectá-los diretamente através da aresta (i,j).



Problema do caixeiro viajante

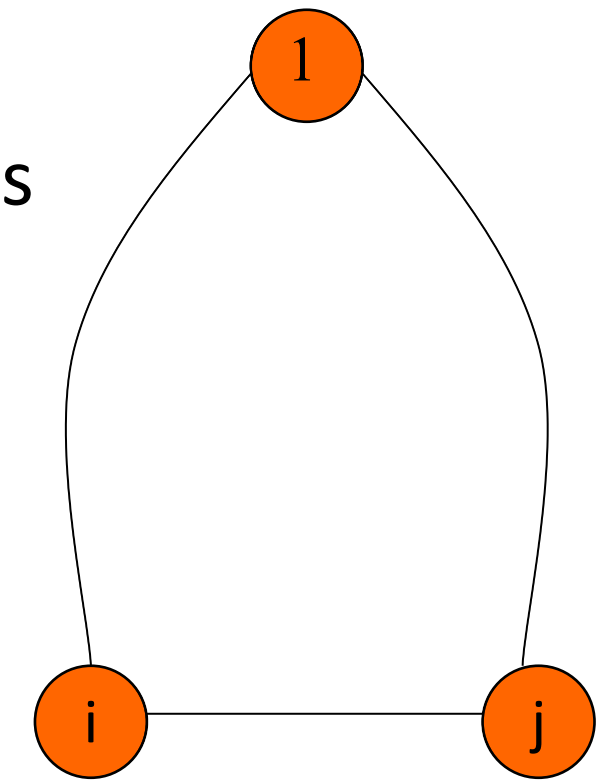
- Outra idéia diferente: considerar a fusão de subcircuitos
- Considerar dois subcircuitos passando pelo nó 1 e pelos nós i e j.
- Conectá-los diretamente através da aresta (i,j).
- Remover as arestas (1,i) e (1,j).



Problema do caixeiro viajante

- Outra idéia diferente: considerar a fusão de subcircuitos
- Considerar dois subcircuitos passando pelo nó 1 e pelos nós i e j.
- Conectá-los diretamente através da aresta (i,j).
- Remover as arestas (1,i) e (1,j).
- Economia realizada:

  $s_{ij} = c_{1i} + c_{1j} - c_{ij}$



Problema do caixeiro viajante

- Algoritmo das economias:

Escolher um nó inicial i (e.g., i = 1).

Construir subcircuitos de comprimento 2 envolvendo o nó inicial (e.g., i = 1) e cada um dos demais nós de N.

Calcular as economias $s_{ij} = c_{1i} + c_{1j} - c_{ij}$ obtidas pela fusão dos subcircuitos contendo i e j e ordená-las em ordem decrescente.

Percorrer a lista de economias e fundir os subcircuitos possíveis: a cada iteração, maximizar a distância economizada sobre a solução anterior, combinando-se dois subcircuitos e substituindo-os por uma nova aresta.

Problema do caixeiro viajante

$s_{45} = 9$

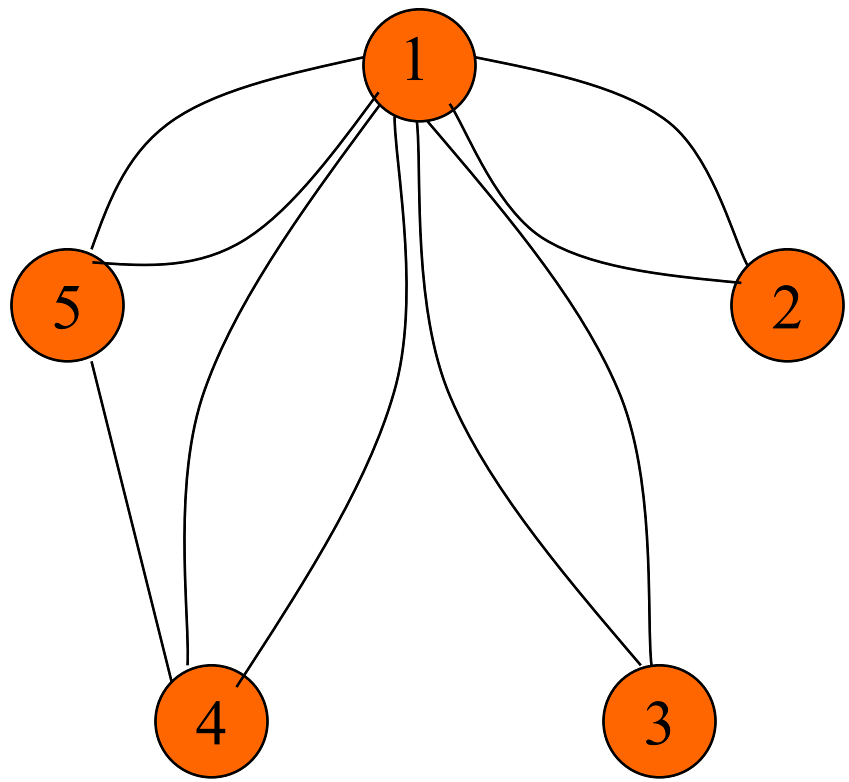
$s_{35} = 5$

$s_{34} = 4$

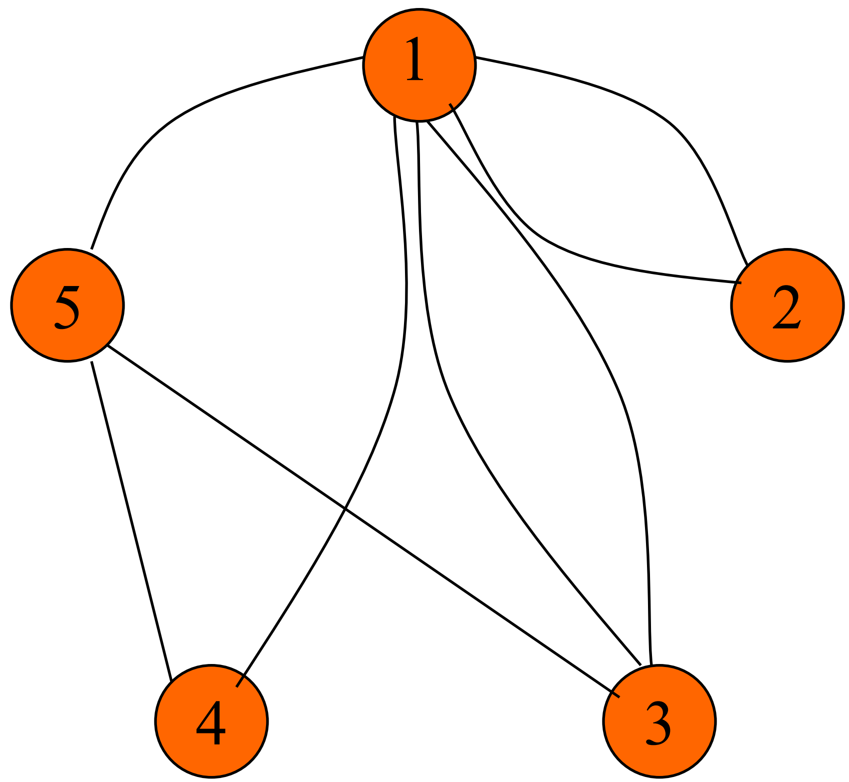
$s_{24} = 4$

$s_{25} = 3$

$s_{23} = 0$



Problema do caixeiro viajante



$s_{45} = 9$

$s_{35} = 5$

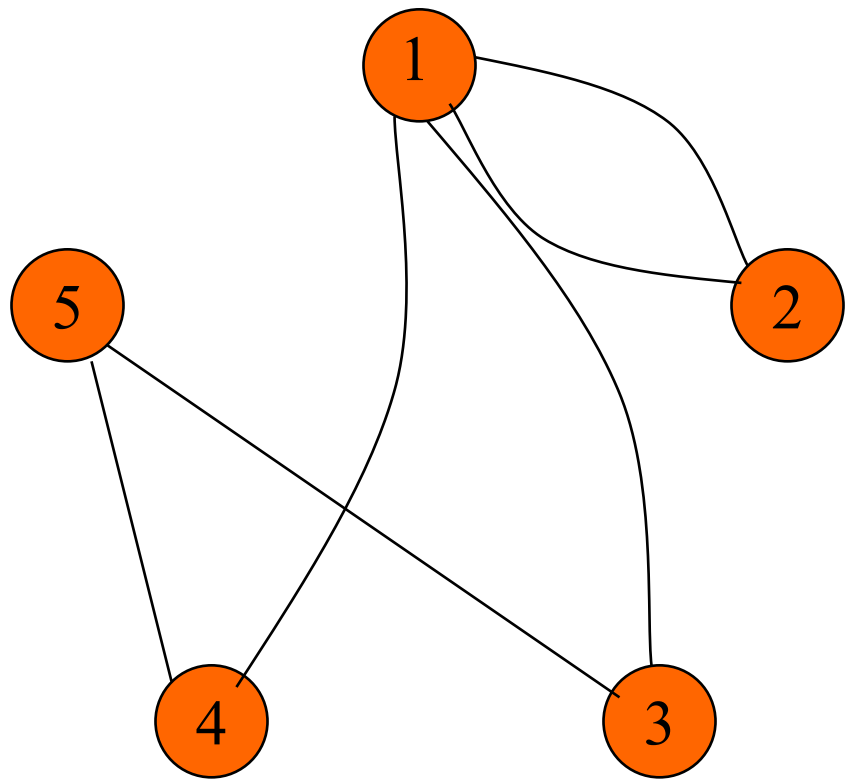
$s_{34} = 4$

$s_{24} = 4$

$s_{25} = 3$

$s_{23} = 0$

Problema do caixeiro viajante



$s_{45} = 9$

$s_{35} = 5$

$s_{34} = 4$

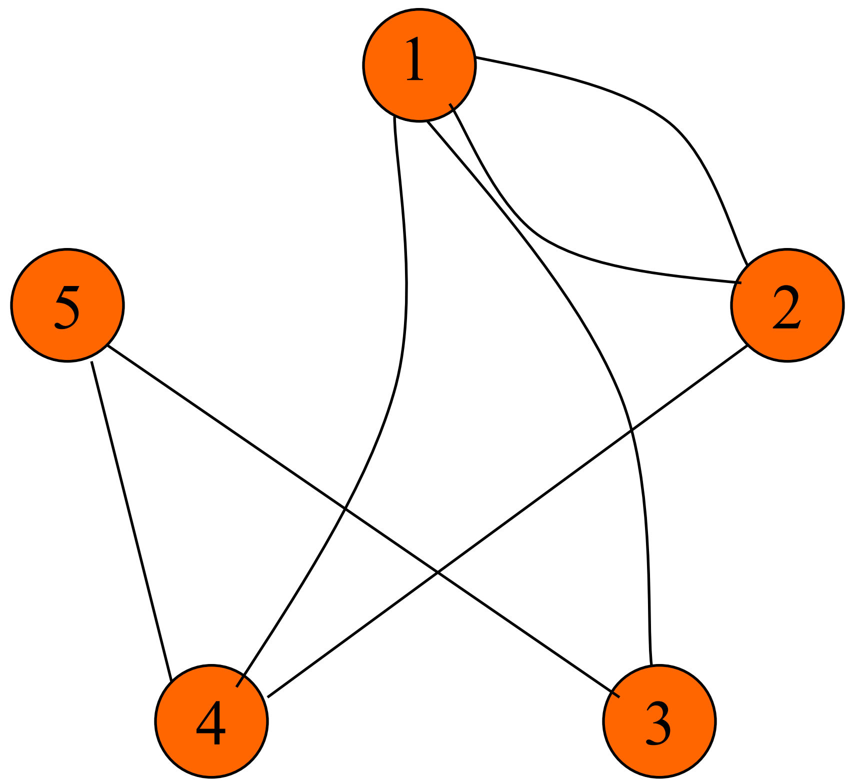
$s_{24} = 4$

$s_{25} = 3$

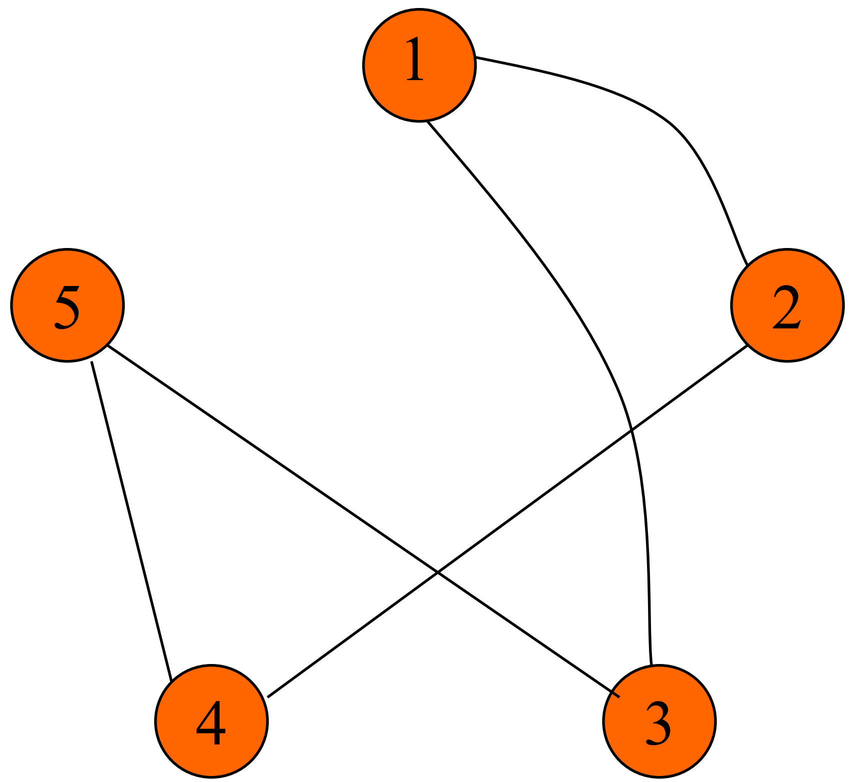
$s_{23} = 0$

Problema do caixeiro viajante

$s_{45} = 9$
$s_{35} = 5$
$s_{34} = 4$
$s_{24} = 4$
$s_{25} = 3$
$s_{23} = 0$



Problema do caixeiro viajante



$s_{45} = 9$
$s_{35} = 5$
$s_{34} = 4$
$s_{24} = 4$
$s_{25} = 3$
$s_{23} = 0$

comprimento = 12

Problema de Steiner em grafos

- Grafo não-orientado G=(V,E)

  V: vértices

  E: arestas

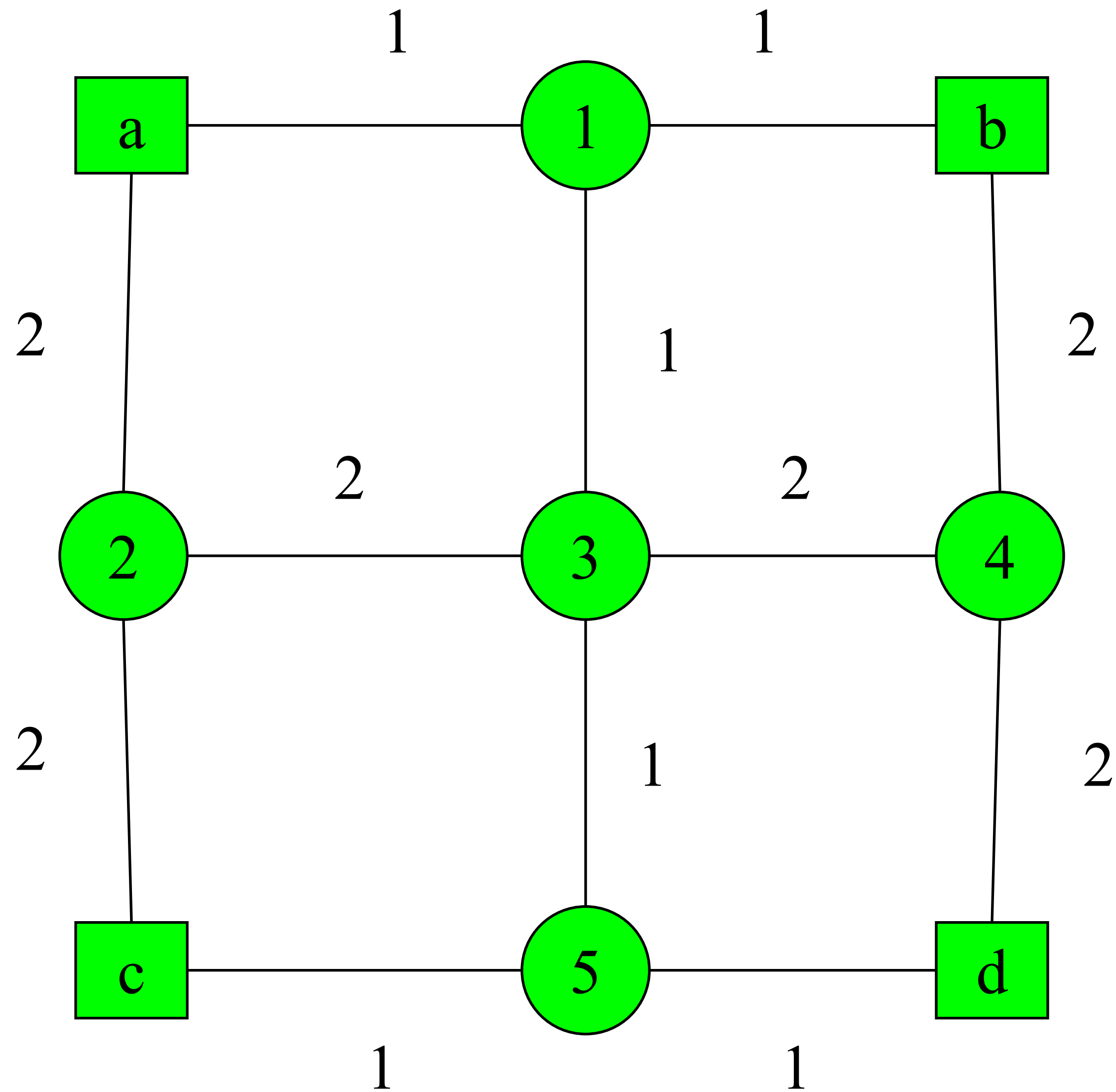
  T: vértices terminais (obrigatórios)

  c_e: peso (positivo) da aresta $e \in E$
- Problema: conectar os nós terminais com custo (peso) mínimo, eventualmente utilizando os demais nós como passagem.

Problema de Steiner em grafos

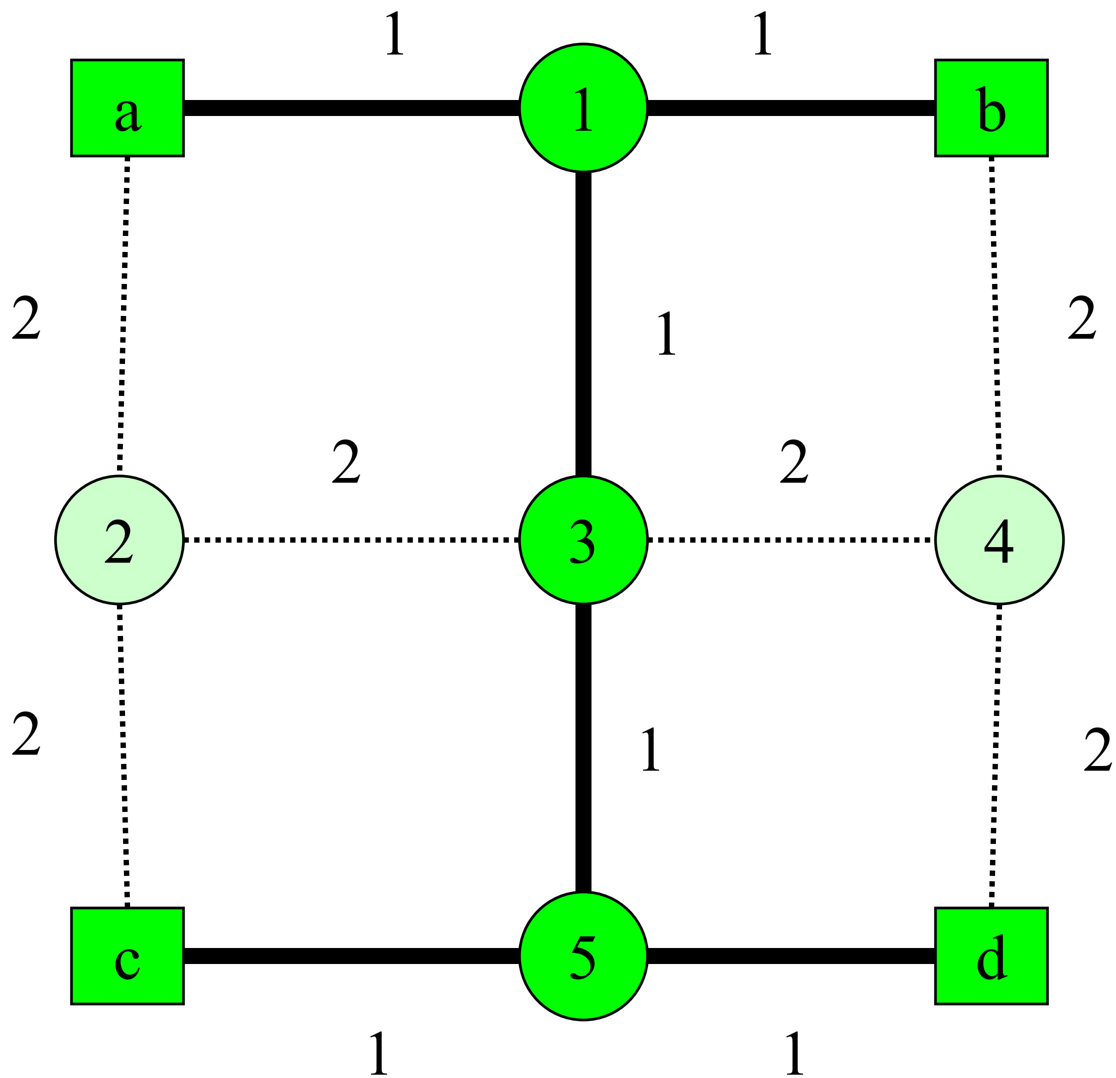
- Vértices de Steiner: vértices opcionais que fazem parte da solução ótima
- Aplicações: projeto de redes de computadores (conectar um conjunto de clientes através de concentradores, cujos locais devem ser determinados), redes de telecomunicações, problema da filogenia em biologia, etc.

Problema de Steiner em grafos



Problema de Steiner em grafos

Mínimo: 6



Problema de Steiner em grafos

- Heurística da rede de distâncias:

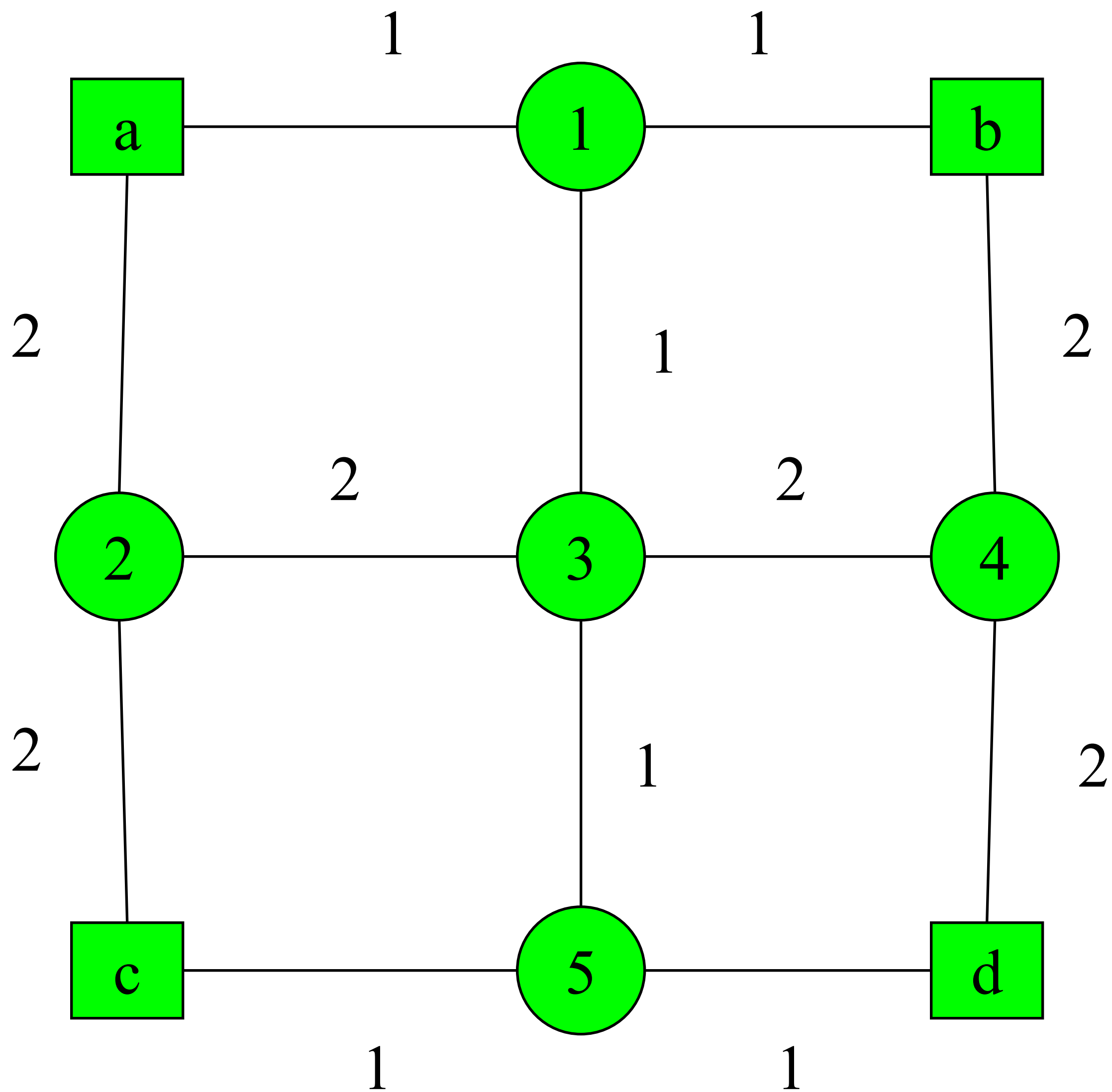
Calcular os caminhos mais curtos entre cada par de terminais do grafo.

Criar a rede de distâncias formada pelos nós obrigatórios e pelas arestas correspondentes aos caminhos mais curtos.

Obter a árvore geradora de peso mínimo dos nós da rede de distâncias.

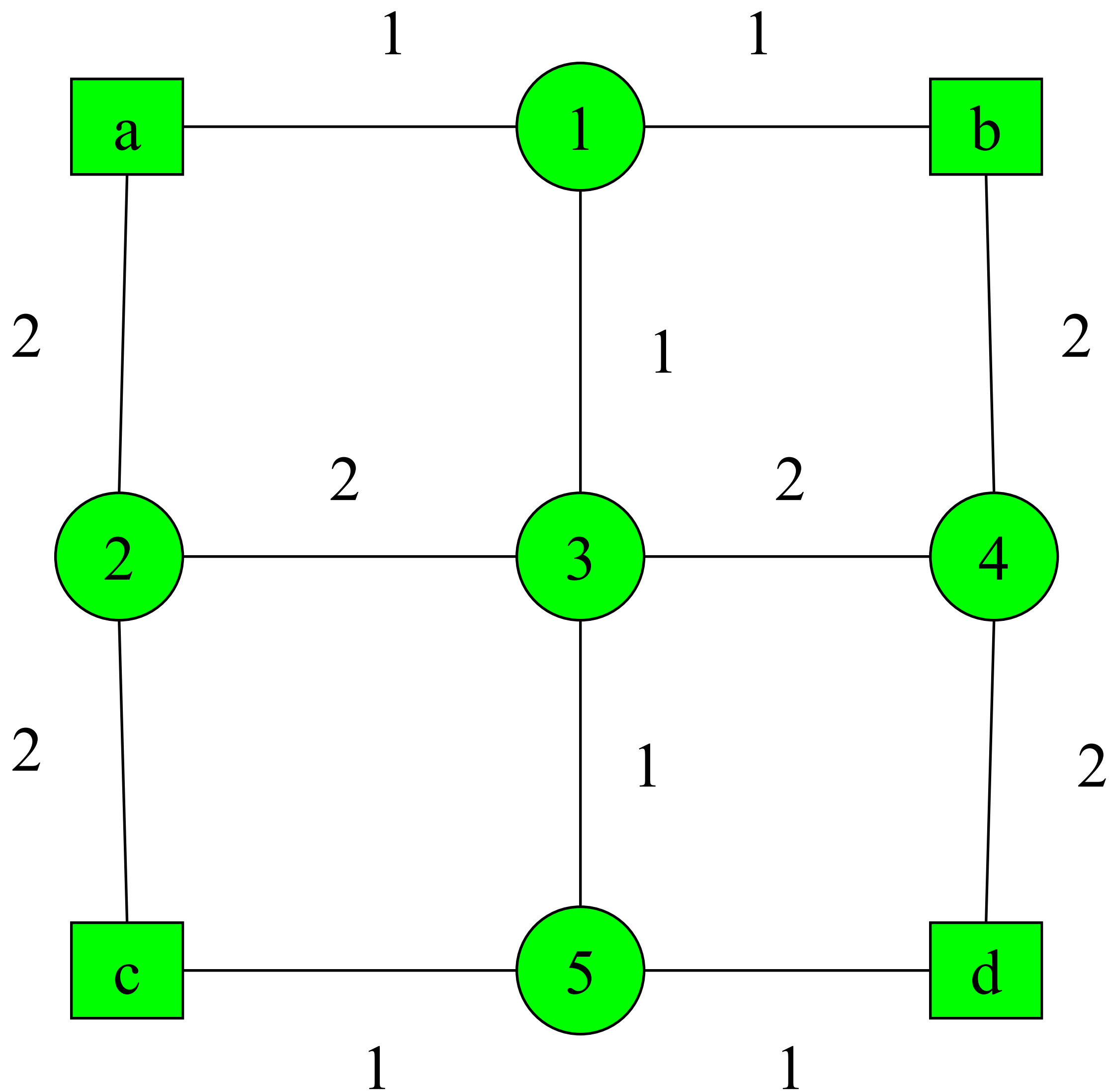
Expandir as arestas da árvore geradora.

Problema de Steiner em grafos



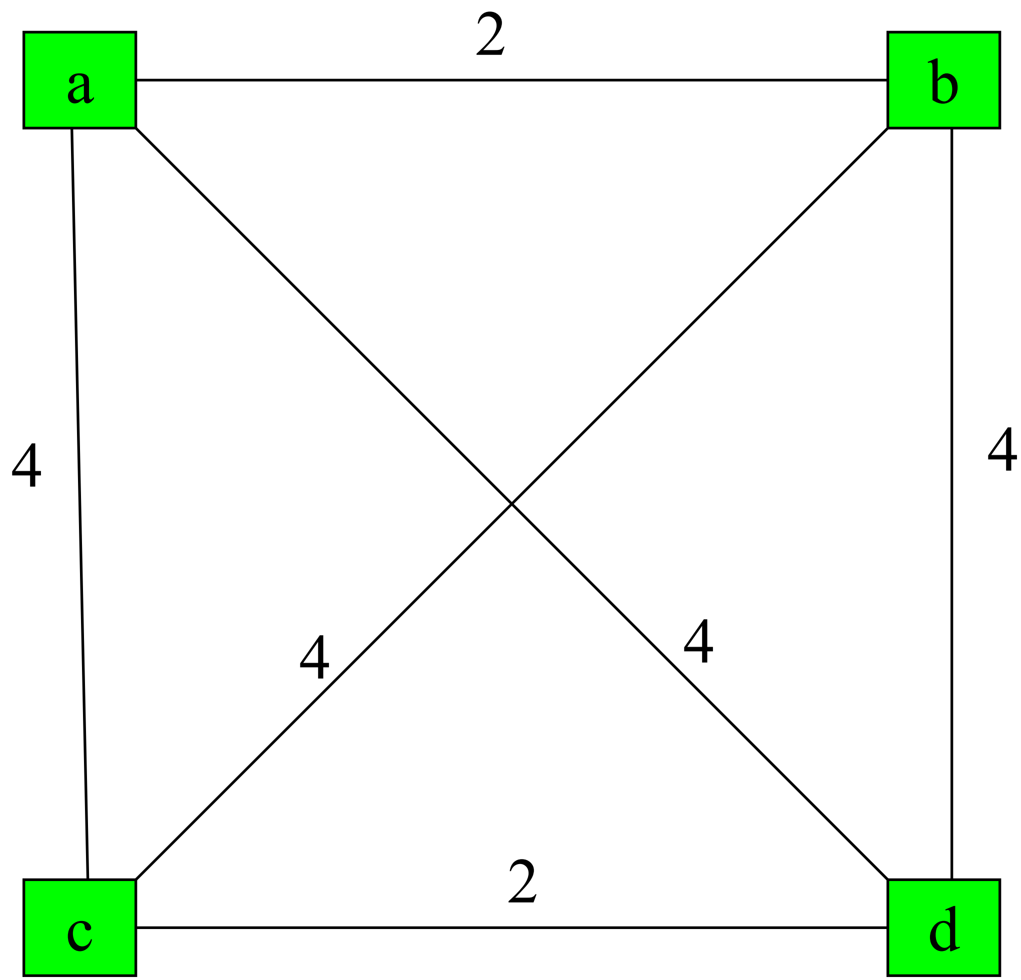
C_{ab}: a,1,b (2)
C_{ac}: a,2,c (4)
C_{ad}: a,1,3,5,d (4)
C_{bc}: b,1,3,5,c (4)
C_{bd}: b,4,d (4)
C_{cd}: c,5,d (2)

Problema de Steiner em grafos



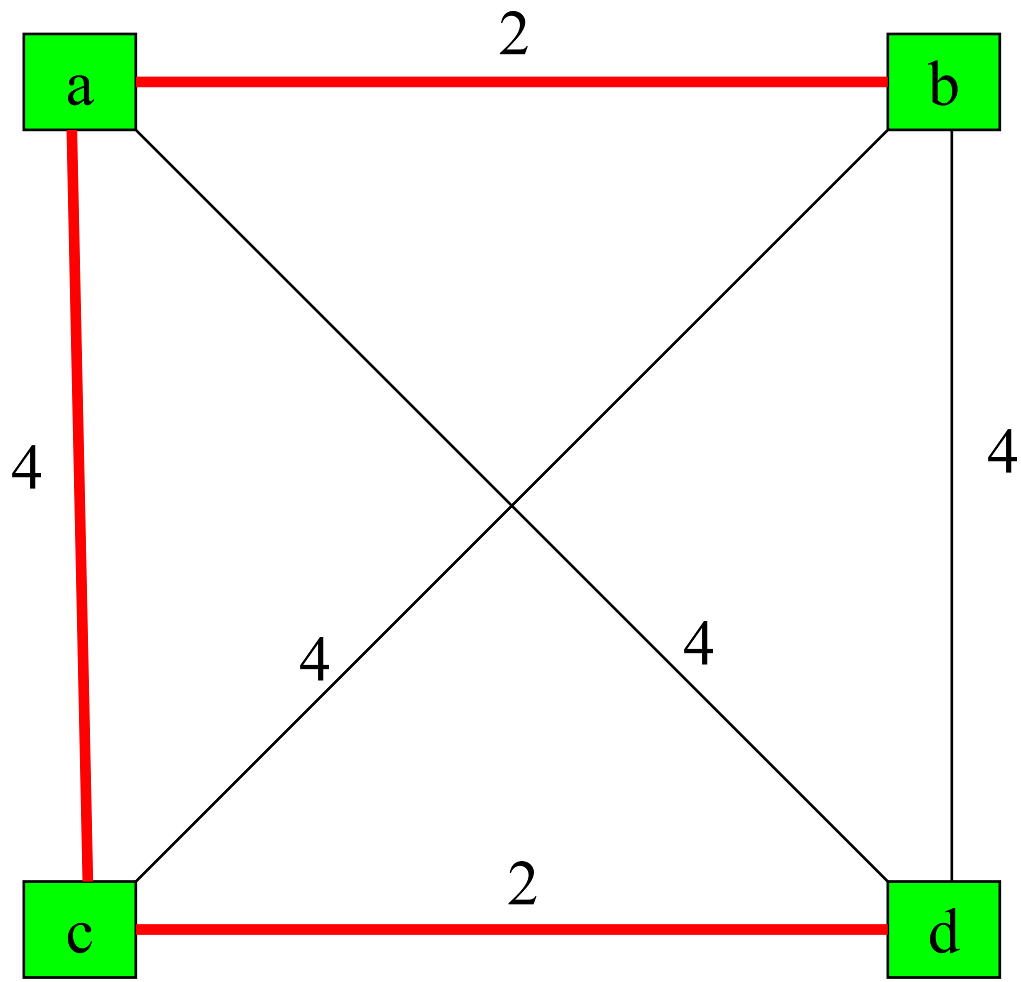
C_{ab}: a,1,b (2)
C_{ac}: a,2,c (4)
C_{ad}: a,1,3,5,d (4)
C_{bc}: b,1,3,5,c (4)
C_{bd}: b,4,d (4)
C_{cd}: c,5,d (2)

Problema de Steiner em grafos



C_{ab}: a,1,b (2)
C_{ac}: a,2,c (4)
C_{ad}: a,1,3,5,d (4)
C_{bc}: b,1,3,5,c (4)
C_{bd}: b,4,d (4)
C_{cd}: c,5,d (2)

Problema de Steiner em grafos



C_{ab}: a,1,b (2)

C_{ac}: a,2,c (4)

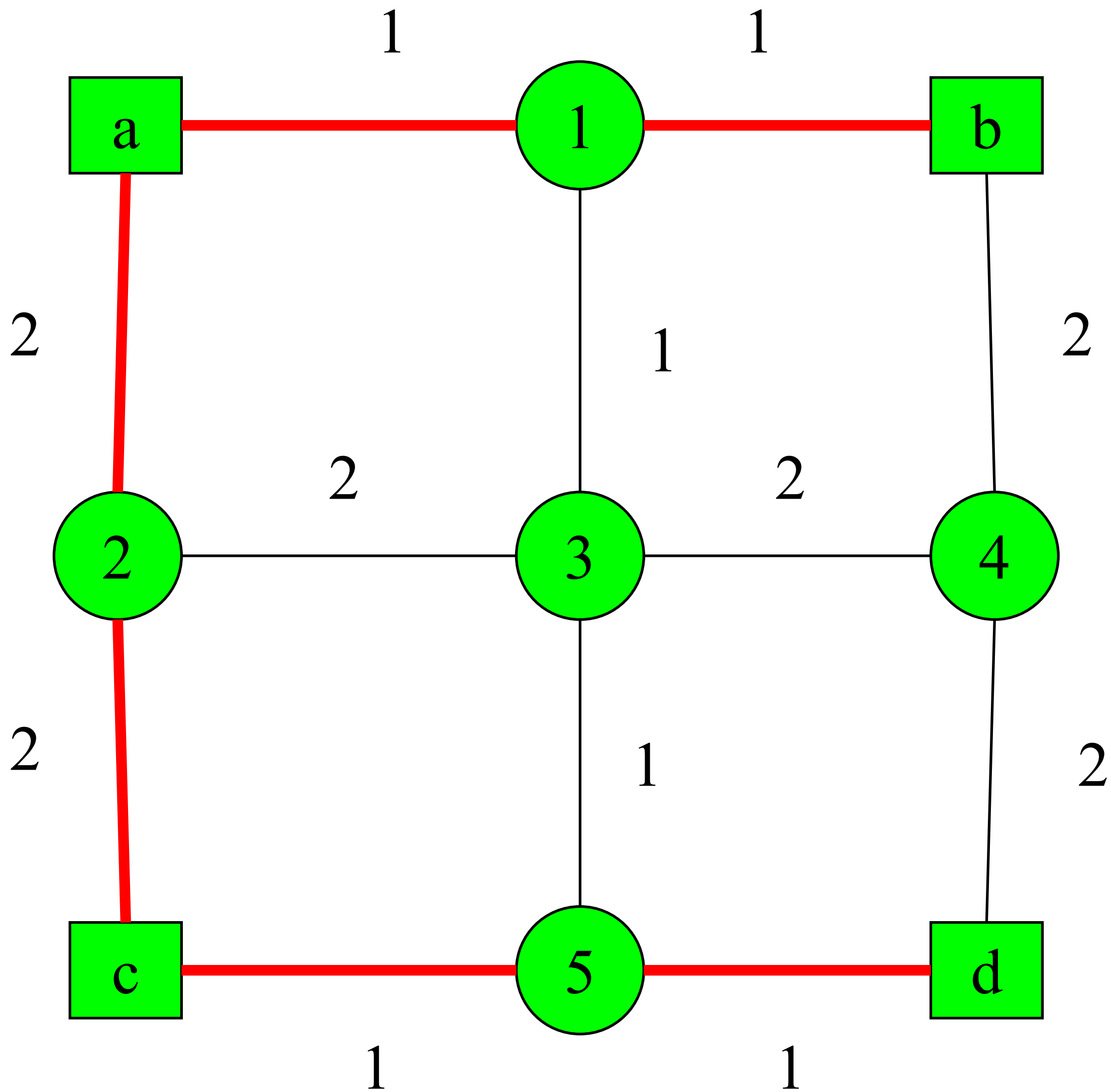
C_{ad}: a,1,3,5,d (4)

C_{bc}: b,1,3,5,c (4)

C_{bd}: b,4,d (4)

C_{cd}: c,5,d (2)

Problema de Steiner em grafos



C_{ab}: a,1,b (2)

C_{ac}: a,2,c (4)

C_{ad}: a,1,3,5,d (4)

C_{bc}: b,1,3,5,c (4)

C_{bd}: b,4,d (4)

C_{cd}: c,5,d (2)

Peso: 8

Problema de Steiner em grafos

- Heurística dos caminhos mais curtos:

Calcular o caminho mais curto de entre cada par de terminais.

Sejam s um nó terminal, Solução $\leftarrow$ {s}, S $\leftarrow$ {s}, k $\leftarrow$ 0.

Enquanto S $\neq$ T fazer:

Obter o terminal s mais próximo de Solução e o caminho correspondente C.

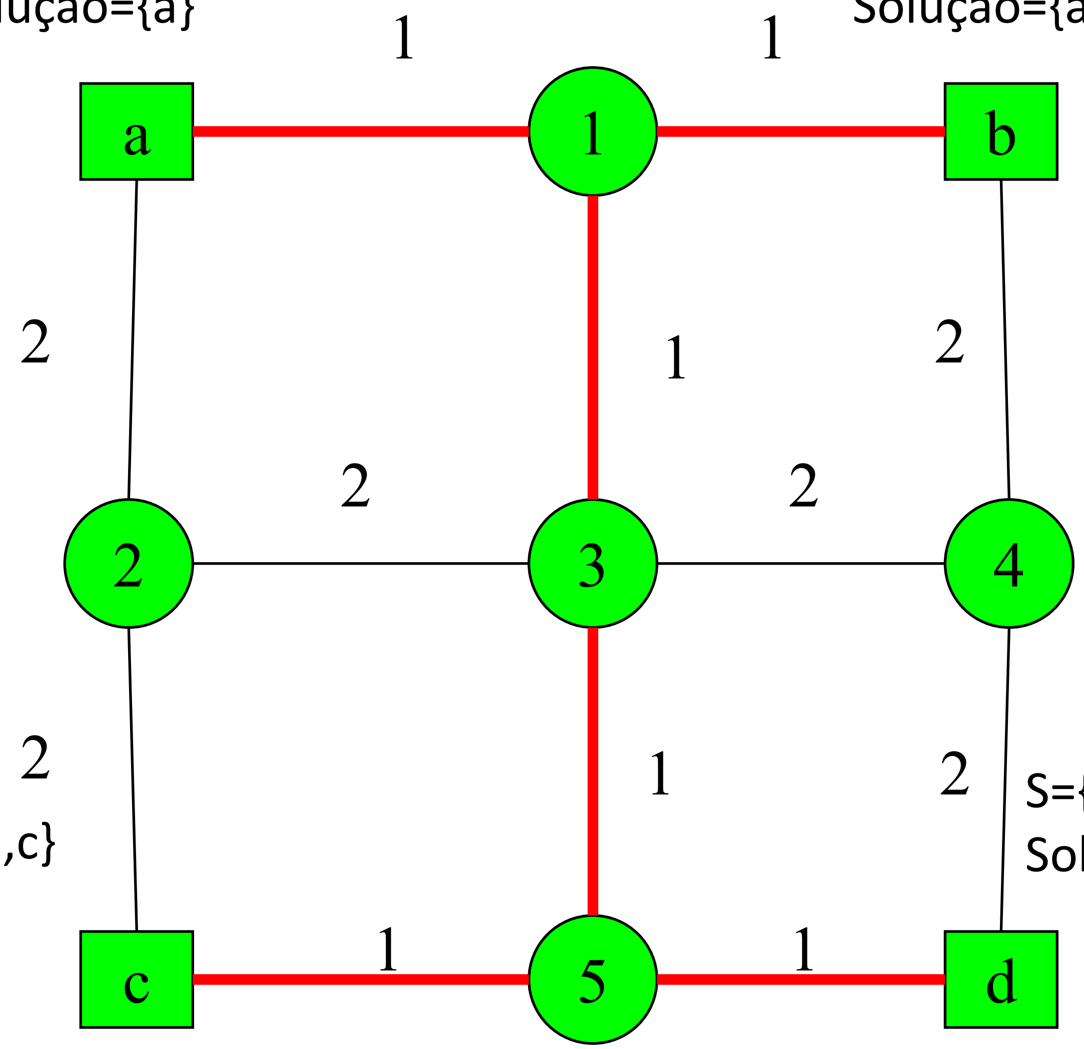
Fazer S $\leftarrow$ S $\cup$ {s} e Solução $\leftarrow$ Solução $\cup$ C.

Fim-enquanto

Problema de Steiner em grafos

S={a}
Solução={a}

S={a,b}
Solução={a,1,b}



Peso: 6

S={a,b,d,c}
Solução={a,1,b,3,5,d,c}

S={a,b,d}
Solução={a,1,b,3,5,d}

Considerações finais

- Algoritmos heurísticos podem não garantir a otimalidade, mas em tempo polinomial podem oferecer soluções para problemas da classe NP-Difícil.

- O conceito de “resolver” o problema é relaxado e considera-se então obter uma boa solução.

Algoritmos gulosos

- Cada elemento que entra na solução, nela permanece até o final.
- Algoritmo guloso para o problema da mochila:
  - Ordenar os itens em ordem decrescente da razão c_j/a_j.
  - Selecionar os itens que cabem na mochila segundo esta ordem.
- Algoritmo do vizinho mais próximo para o PCV
- Cuidado: nem sempre encontram a solução ótima exata, são portanto heurísticas para estes problemas!

Exercício – Avaliação

- Descrição de instâncias e soluções ótimas para o problema do caixeiro viajante:

http://www.iwr.uni-heidelberg.de/groups/comopt/software/TSPLIB95/

- Estudo comparativo de heurísticas para o problema do caixeiro viajante:

http://www.research.att.com/~dsj/chtsp/

- Em particular, página com gráficos comparativos:

http://www.research.att.com/~dsj/chtsp/testform2.html

- Referência: Lawler, Lenstra, Rinnooy Kan e Shmoys (eds.), “The traveling salesman problem”, 1985 (entre outras)