



- DCC136 - Algoritmos Construtivos

– Heurísticas de Construção –



Conceitos Comuns

- Roteiro
 - Algoritmo de Construção
 - Algoritmo Guloso
 - Algoritmo Guloso randomizado, adaptativo, reativo

Métodos Construtivos

- Problema de otimização combinatória

Definição 1:

- Pode ser definido por um par (S, f) , onde:
- S representa o conjunto de soluções viáveis
- $f : S \rightarrow \mathbb{R}$ a função objetivo a otimizar (função que associa a cada solução $s \in S$ do espaço de busca um valor real que indica seu valor, permitindo uma relação de ordem entre pares de soluções).

– Pag 3

Métodos Construtivos

- Problema de otimização combinatória:

Definição 2:

- Dado um conjunto finito $E = \{1, 2, \dots, n\}$ e uma função de custo $c: 2^E \rightarrow \mathbb{R}$ encontrar $s^* \in F$ tal que $c(s^*) \leq c(s) \forall s \in F$ onde $F \in 2^E$ é o conjunto de soluções viáveis do problema.
- F : Conjunto discreto de soluções com um número finito de elementos.

Métodos Construtivos

- Algoritmo Guloso ou algoritmo construtivo
 - Constroem uma solução de forma iterativa, inserindo a cada passo um elemento no conjunto solução.

Métodos Construtivos

- Algoritmo Guloso ou algoritmo construtivo
 - A partir de uma solução inicial (conjunto vazio), a cada passo, atribui-se valor para uma variável de decisão até que uma solução viável seja construída.
 - Em problemas de otimização, onde a solução pode ser definida pela presença/ausência de elementos, uma solução parcial pode ser vista como um subconjunto de elementos. A cada etapa um elemento é inserido na solução parcial.

Métodos Construtivos

- Algoritmo Guloso ou algoritmo construtivo
 - A partir de uma solução inicial (conjunto vazio), a cada passo, atribui-se valor para uma variável de decisão até que uma solução viável seja construída.
 - Em problemas de otimização, onde a solução pode ser definida pela presença/ausência de elementos, uma solução parcial pode ser vista como um subconjunto de elementos. A cada etapa um elemento é inserido na solução parcial.

Métodos Construtivos

- Algoritmo Guloso ou algoritmo construtivo
 - Selecionar sequencialmente elementos de E , eventualmente descartando alguns, de tal forma que se obtenha ao final uma solução viável, isto é, pertencente a F .

Métodos Construtivos

- Algoritmo Guloso ou algoritmo construtivo
Funcionamento geral:
 - Constrói uma solução, elemento por elemento;
 - A cada passo é adicionado um único elemento candidato;
 - O candidato escolhido é o “melhor” segundo um certo critério (heurística);
 - O método se encerra quando todos os elementos candidatos foram analisados.
-

Métodos Construtivos

- Algoritmo Guloso ou algoritmo construtivo

Exemplo: Problema da Mochila

Seja, então, uma mochila de capacidade $b = 23$ e os 5 objetos da tabela abaixo, com os respectivos pesos e benefícios.

Objeto (j)	1	2	3	4	5
Peso (w_j)	4	5	7	9	6
Benefício (p_j)	2	2	3	4	4

Construamos uma solução para esse problema usando a seguinte idéia: adicionemos à mochila a cada passo, o objeto mais valioso por unidade de peso e que não ultrapasse a capacidade da mochila. Reordenando os objetos de acordo com a relação p_j/w_j , obtemos:

Objeto (j)	5	1	4	3	2
Peso (w_j)	6	4	9	7	5
Benefício (p_j)	4	2	4	3	2
(p_j/w_j)	0.67	0.50	0.44	0.43	0.40

Representemos uma solução s por um vetor binário de n posições.

Métodos Construtivos

- Algoritmo Guloso ou algoritmo construtivo

Passo 1 : Adicionemos, primeiramente, o objeto 5, que tem a maior relação p_j/w_j

$$s = (00001)^t$$

$$f(s) = 4$$

$$\text{Peso corrente da mochila} = 6 < b = 23$$

Passo 2 : Adicionemos, agora, o objeto 1, que tem a segunda maior relação p_j/w_j

$$s = (10001)^t$$

$$f(s) = 6$$

$$\text{Peso corrente da mochila} = 10 < b = 23$$

Passo 3 : Adicionemos, agora, o objeto 4, que tem a terceira maior relação p_j/w_j

$$s = (10011)^t$$

$$f(s) = 10$$

$$\text{Peso corrente da mochila} = 19 < b = 23$$

Passo 4 : O objeto a ser alocado agora seria o terceiro. No entanto, esta alocação faria superar a capacidade da mochila. Neste caso, devemos tentar alocar o próximo objeto com a maior relação p_j/w_j , que é o objeto 1. Como também a alocação desse objeto faria superar a capacidade da mochila e não há mais objetos candidatos, concluímos que a solução anterior é a solução final, isto é: $s^* = (10011)^t$ com $f(s^*) = 10$.

Métodos Construtivos

- Algoritmo Guloso ou algoritmo construtivo

Algorithm 1.2 Template of a greedy algorithm.

$s = \{\}$; /* Initial solution (null) */

Repeat

$e_i = \text{Local-Heuristic}(E \setminus \{e/e \in s\})$;

If $s \cup e_i \in F$ **Then**

$s = s \cup e_i$;

Until Complete solution found

Métodos Construtivos

- Algoritmo Guloso ou algoritmo construtivo
 - Principais questões

Algorithm 1.2 Template of a greedy algorithm

```
 $s = \{\}$  ; /* Initial solution (n = 0) */  
Repeat  
   $e_i = \text{Local-Heuristic}(E \setminus \{e / e \in s\})$  ;  
  If  $s \cup e_i \in F$  Then  
     $s = s \cup e_i$  ;  
Until Complete solution found
```

Heurística de seleção de elementos

Identificar a solução como um conjunto de elementos

Métodos Construtivos

- Algoritmo Guloso ou algoritmo construtivo
 - Um exemplo para o Problema Caixeiro Viajante
 - E:
 - Heurística Local:

Métodos Construtivos

- Algoritmo Guloso ou algoritmo construtivo
 - Um exemplo para o Problema Caixeiro Viajante
 - E: Conjunto das arestas
 - Heurística Local: Selecionar o vizinho mais próximo

Métodos Construtivos

- Construção de uma solução
 - Problema do Caixeiro Viajante (PCV)
 - Heurísticas locais para o PCV
 - Vizinho mais próximo
 - Inserção mais próxima
 - Inserção mais distante
 - Inserção mais barata
 - Inserção com maior ângulo
 - Algoritmo de economias
 - ...

Métodos Construtivos

- PCV – Vizinho mais próximo
 - Ideia central: Construir uma rota, passo a passo, adicionando à solução corrente a cidade (ainda não visitada) mais próxima da última cidade inserida.

Métodos Construtivos

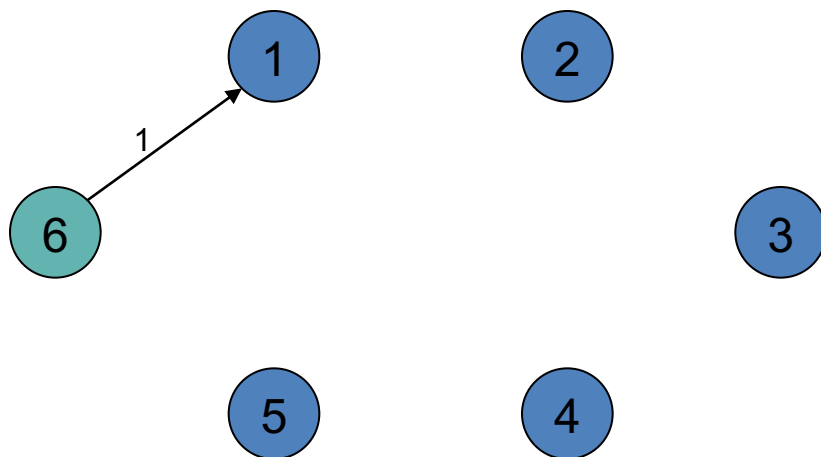
- PCV – Vizinho mais próximo
 - Marcone (UFOP)

PCV – Vizinho mais Próximo

Exemplo - Passo 1

Cid.	1	2	3	4	5	6
1	0	2	1	4	9	1
2	2	0	5	9	7	2
3	1	5	0	3	8	6
4	4	9	3	0	2	6
5	9	7	8	2	0	2
6	1	2	6	6	2	0

i	j	d_{ij}
6	1	1
6	2	2
6	3	6
6	4	6
6	5	2



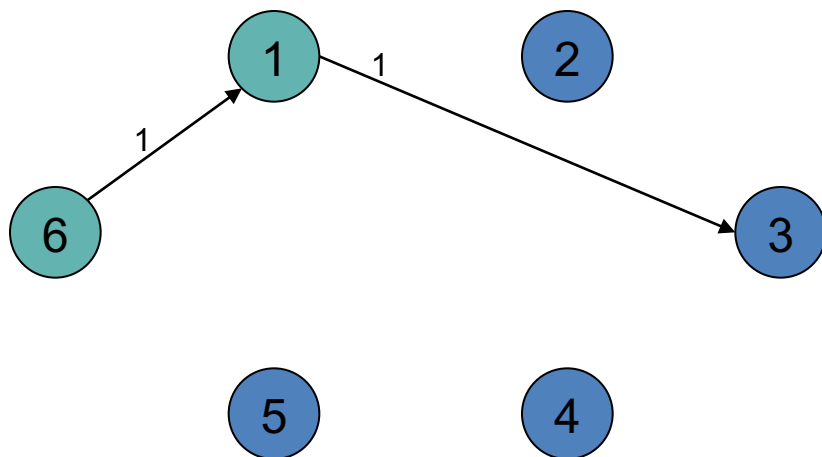
• Distância Total = 1

PCV – Vizinho mais Próximo

Exemplo - Passo 2

Cid.	1	2	3	4	5	6
1	0	2	1	4	9	1
2	2	0	5	9	7	2
3	1	5	0	3	8	6
4	4	9	3	0	2	6
5	9	7	8	2	0	2
6	1	2	6	6	2	0

i	j	d_{ij}
1	2	2
1	3	1
1	4	4
1	5	9



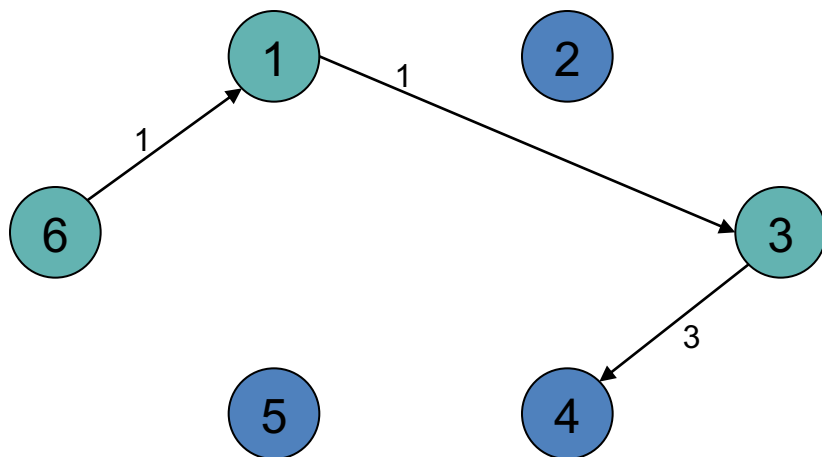
• Distância Total = $1 + 1 = 2$

PCV – Vizinho mais Próximo

Exemplo - Passo 3

Cid.	1	2	3	4	5	6
1	0	2	1	4	9	1
2	2	0	5	9	7	2
3	1	5	0	3	8	6
4	4	9	3	0	2	6
5	9	7	8	2	0	2
6	1	2	6	6	2	0

i	j	d_{ij}
3	2	5
3	4	3
3	5	8



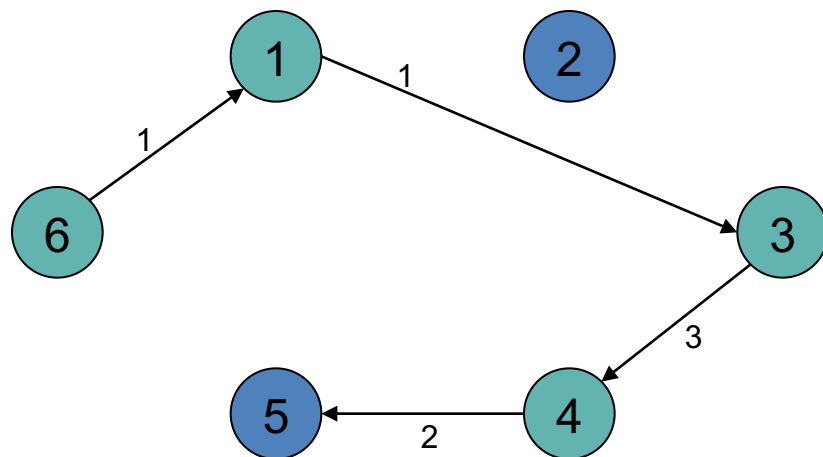
• Distância Total = $2 + 3 = 5$

PCV – Vizinho mais Próximo

Exemplo - Passo 4

Cid.	1	2	3	4	5	6
1	0	2	1	4	9	1
2	2	0	5	9	7	2
3	1	5	0	3	8	6
4	4	9	3	0	2	6
5	9	7	8	2	0	2
6	1	2	6	6	2	0

i	j	d_{ij}
4	2	9
4	5	2



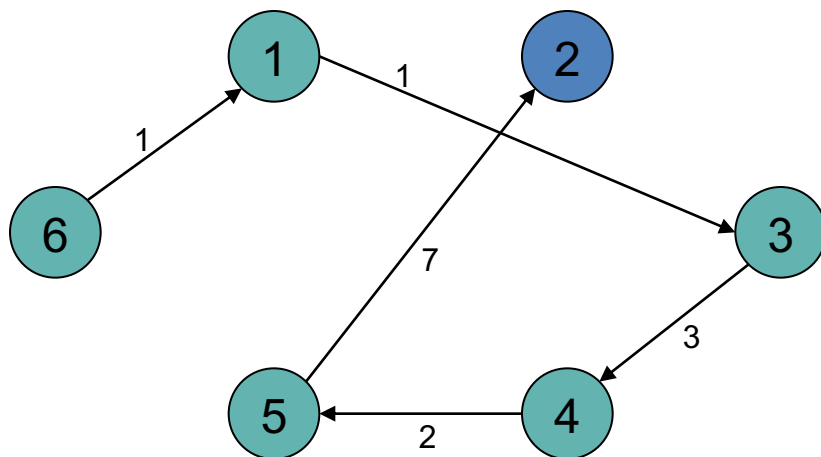
• Distância Total = 5 + 2 = 7

PCV – Vizinho mais Próximo

Exemplo - Passo 5

Cid.	1	2	3	4	5	6
1	0	2	1	4	9	1
2	2	0	5	9	7	2
3	1	5	0	3	8	6
4	4	9	3	0	2	6
5	9	7	8	2	0	2
6	1	2	6	6	2	0

i	j	d_{ij}
5	2	7

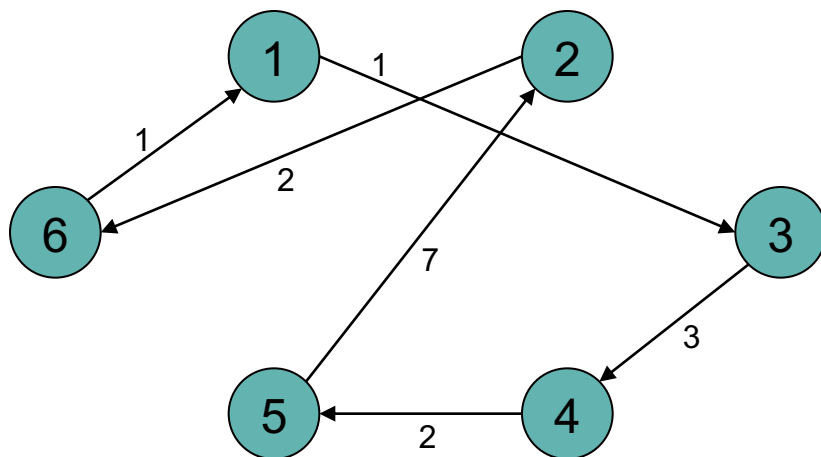


• Distância Total = $7 + 7 = 14$

PCV – Vizinho mais Próximo

Exemplo – Passo final: “Inserção forçada”

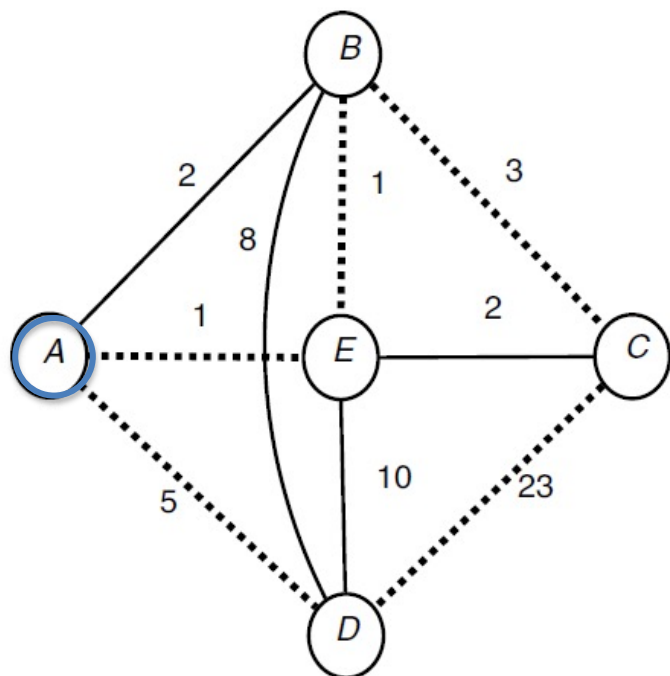
Cid.	1	2	3	4	5	6
1	0	2	1	4	9	1
2	2	0	5	9	7	2
3	1	5	0	3	8	6
4	4	9	3	0	2	6
5	9	7	8	2	0	2
6	1	2	6	6	2	0



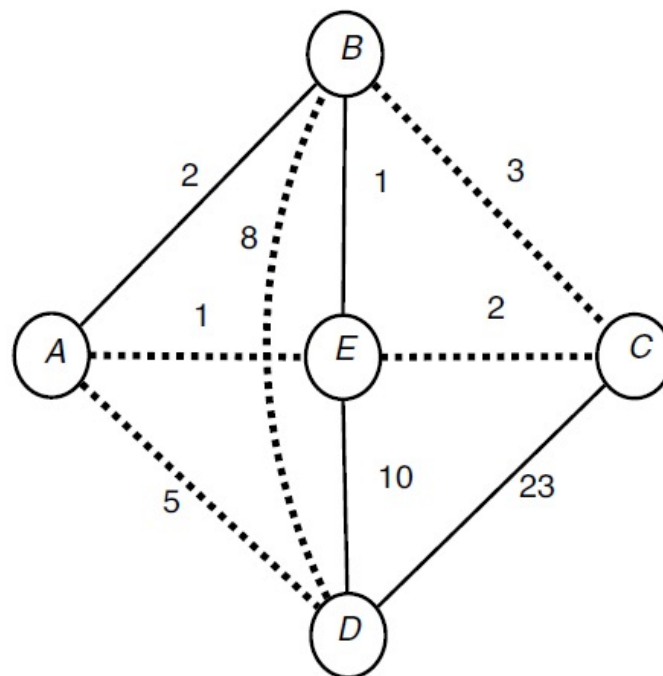
• Distância Total = $14 + 2 = 16$

Métodos Construtivos

- Algoritmo Guloso ou algoritmo construtivo
 - Exemplo: PCV



Greedy final solution : A – E – B – C – D – A
with cost = 33



Better solution : A – E – C – B – D – A
with cost = 19

10

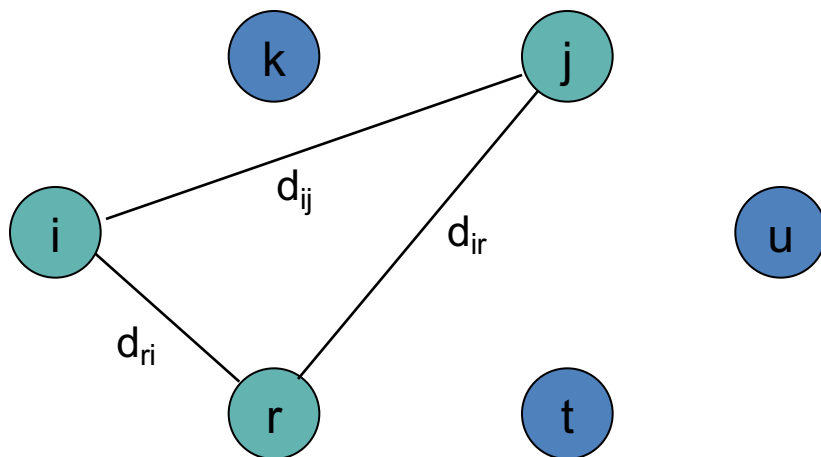
Métodos Construtivos

- PCV – Inserção mais barata
 - Ideia central: Construir uma rota passo a passo, partindo de rota inicial envolvendo 3 cidades e adicionar a cada passo, a cidade k (ainda não visitada) entre a ligação (i, j) de cidades já visitadas, cujo custo de inserção seja o mais barato

PCV – Inserção mais barata

Exemplo - Passo 0

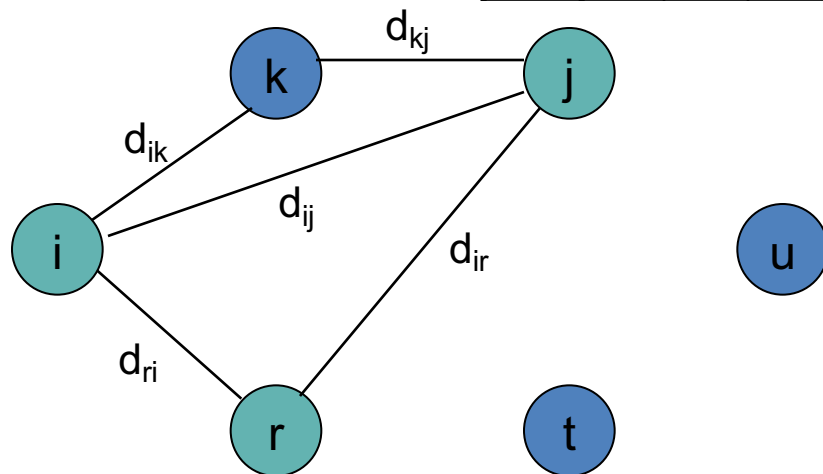
Cid.	1	2	3	4	5	6
1	0	2	1	4	9	1
2	2	0	5	9	7	2
3	1	5	0	3	8	6
4	4	9	3	0	2	6
5	9	7	8	2	0	2
6	1	2	6	6	2	0



PCV – Inserção mais barata

Exemplo - Passo 1

Cid.	1	2	3	4	5	6
1	0	2	1	4	9	1
2	2	0	5	9	7	2
3	1	5	0	3	8	6
4	4	9	3	0	2	6
5	9	7	8	2	0	2
6	1	2	6	6	2	0

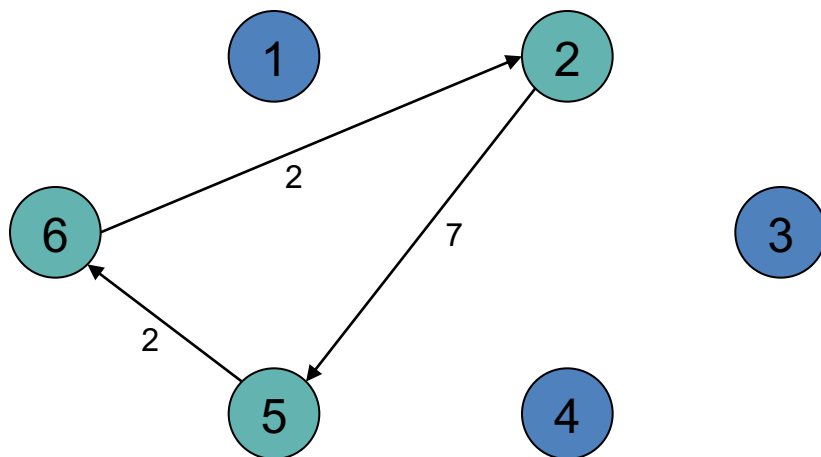


$$\text{Custo da inserção} = d_{ik} + d_{kj} - d_{ij}$$

PCV – Inserção mais barata

Exemplo - Passo 1

Cid.	1	2	3	4	5	6
1	0	2	1	4	9	1
2	2	0	5	9	7	2
3	1	5	0	3	8	6
4	4	9	3	0	2	6
5	9	7	8	2	0	2
6	1	2	6	6	2	0

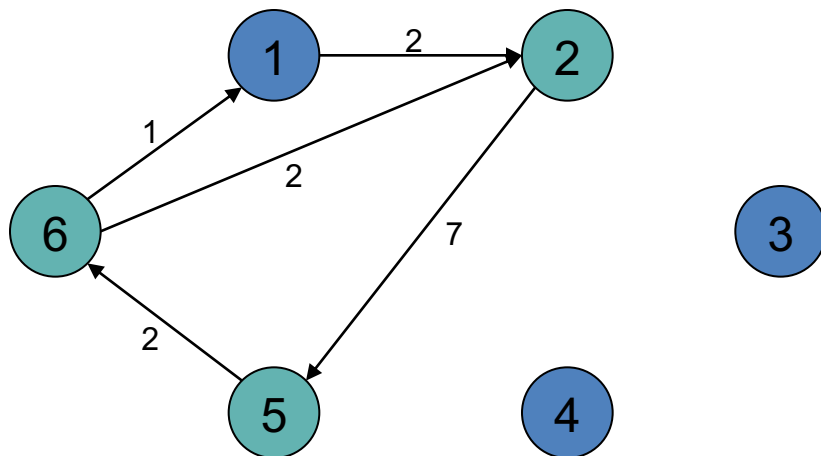


• Distância Total = 11

PCV – Inserção mais barata

Exemplo - Passo 2

Cid.	1	2	3	4	5	6
1	0	2	1	4	9	1
2	2	0	5	9	7	2
3	1	5	0	3	8	6
4	4	9	3	0	2	6
5	9	7	8	2	0	2
6	1	2	6	6	2	0



i	k	j	$d_{ik} + d_{kj} - d_{ij}$
6	1	2	$1 + 2 - 2 = 1$
6	3	2	$6 + 5 - 2 = 9$
6	4	2	$6 + 9 - 2 = 3$
2	1	5	$2 + 9 - 7 = 4$
2	3	5	$5 + 8 - 7 = 6$
2	4	5	$9 + 2 - 7 = 4$
5	1	6	$9 + 1 - 2 = 8$
5	3	6	$8 + 6 - 2 = 12$
5	4	6	$2 + 6 - 2 = 6$

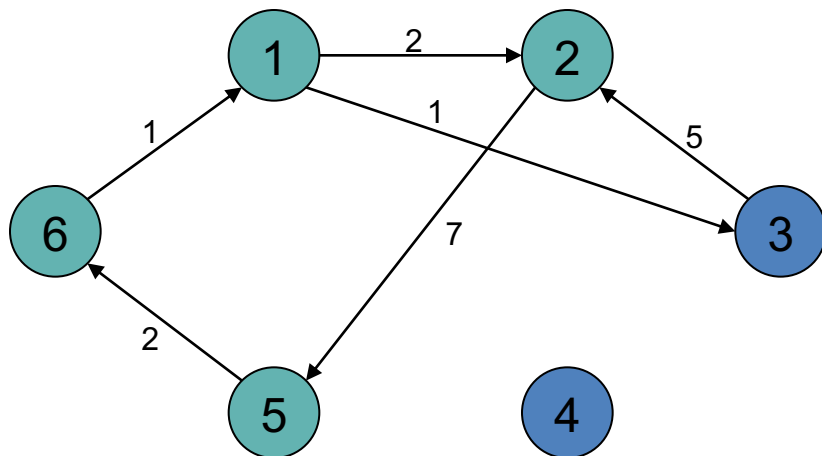
• Distância Total = 11 + 1 = 12

PCV – Inserção mais barata

Exemplo - Passo 3

Cid.	1	2	3	4	5	6
1	0	2	1	4	9	1
2	2	0	5	9	7	2
3	1	5	0	3	8	6
4	4	9	3	0	2	6
5	9	7	8	2	0	2
6	1	2	6	6	2	0

i	k	j	$d_{ik} + d_{kj} - d_{ij}$
6	3	1	$6 + 1 - 1 = 6$
6	4	1	$6 + 4 - 1 = 9$
1	3	2	$1 + 5 - 2 = 4$
1	4	2	$4 + 9 - 2 = 11$
2	3	5	$5 + 8 - 7 = 6$
2	4	5	$9 + 2 - 7 = 4$
5	3	6	$8 + 6 - 2 = 12$
5	4	6	$2 + 6 - 2 = 6$



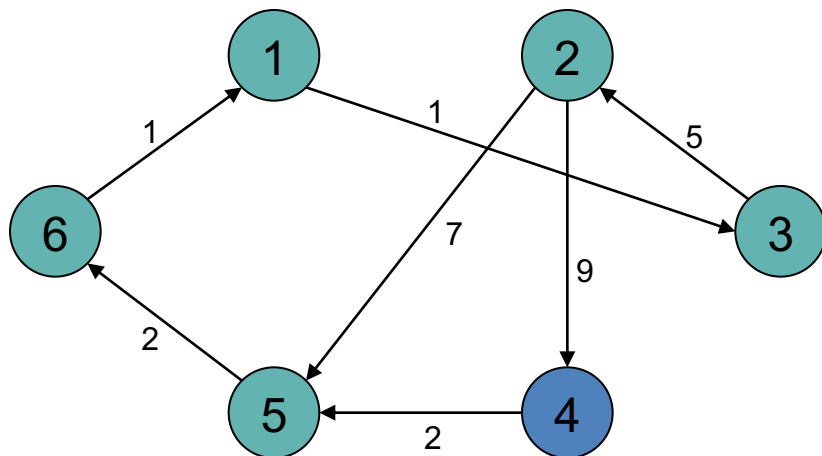
• Distância Total = $12 + 4 = 16$

PCV – Inserção mais barata

Exemplo - Passo final

Cid.	1	2	3	4	5	6
1	0	2	1	4	9	1
2	2	0	5	9	7	2
3	1	5	0	3	8	6
4	4	9	3	0	2	6
5	9	7	8	2	0	2
6	1	2	6	6	2	0

i	k	j	$d_{ik} + d_{kj} - d_{ij}$
6	4	1	$6 + 4 - 1 = 9$
1	4	3	$4 + 3 - 1 = 6$
3	4	2	$3 + 9 - 5 = 7$
2	4	5	$9 + 2 - 7 = 4$
5	4	6	$2 + 6 - 2 = 6$

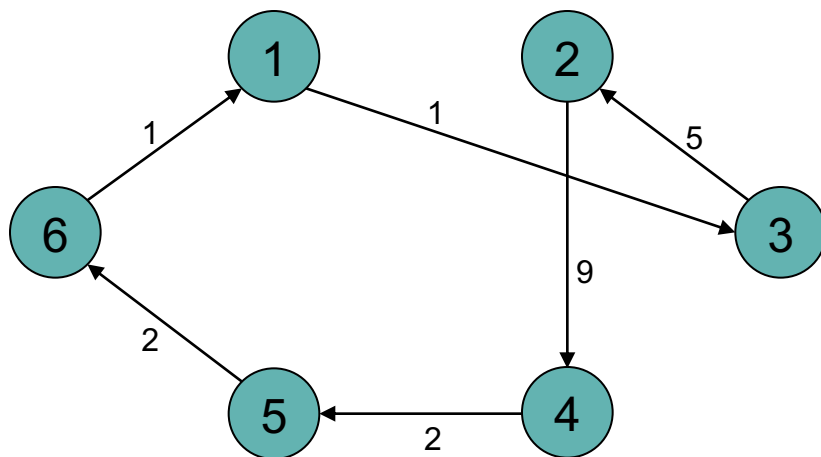


• Distância Total = 16 + 4 = 20

PCV – Inserção mais barata

Exemplo – solução final

Cid.	1	2	3	4	5	6
1	0	2	1	4	9	1
2	2	0	5	9	7	2
3	1	5	0	3	8	6
4	4	9	3	0	2	6
5	9	7	8	2	0	2
6	1	2	6	6	2	0



• Distância Total = $16 + 4 = 20$

$s = (6 \ 1 \ 3 \ 2 \ 4 \ 5)$

Métodos Construtivos

- Como adaptar?
 - Heurística do vizinho mais próximo para o Problema de Roteamento de Veículos com frota homogênea.
 - Heurística da inserção mais barata para o Problema de Roteamento de Veículos com frota homogênea.

Métodos Construtivos

- PRV
 - Estratégia Sequencial x Paralela

Métodos Construtivos

- Árvore de Steiner

Definir:

- E:
- Heurística Local:

Dúvidas

- Perguntas ou comentários?



Luciana Brugiolo Gonçalves
lbrugiolo@ice.ufjf.br

Referências

- EL-GHAZALI TALBI. Metaheuristics: From Design to Implementation, Wiley, 2009.
- Notas de aula do prof. Marcone Jamilson Souza (UFOP)
- Notas de aula da prof^a Liliane Salgado (UFPE).
- Notas de aula do prof Celso Ribeiro (UFF).

TRABALHO

- TRABALHO 1 (continuação):
 - The capacitated team orienteering problem
 - Propor uma estratégia capaz de construir uma solução viável para o CTOP.