Heurísticas Construtivas

Marcone Jamilson Freitas Souza[1,2,3]

Puca Huachi Vaz Penna[1]

[1] Departamento de Computação

[1] Programa de Pós-Graduação em Ciência da Computação

Universidade Federal de Ouro Preto

[2] Programa de Pós-graduação em Modelagem Matemática e Computacional / CEFET-MG

[3] Programa de Pós-graduação em Instrumentação, Controle e Automação de Processos de Mineração / ITV/UFOP

www.decom.ufop.br/prof/marcone, www.decom.ufop.br/puca

E-mail: {marcone,puca}@ufop.edu.br

Souza, Marcone J. F. e Penna, Puca H. V. Heurísticas construtivas. Notas de aula de Técnicas Meta-heurísticas para Otimização Combinatória. Departamento de Computação, Universidade Federal de Ouro Preto, Ouro Preto, 2024. Disponível em www.decom.ufop.br/prof/marcone/Disciplinas/InteligenciaComputacional/HeuristicasConstrutivas.pptx

Heurísticas clássicas

- Classificadas em:
  - Construtivas
    - Consistem em construir uma solução passo a passo, elemento por elemento
  - Refinamento
    - Consistem em efetuar modificações na solução construída, de forma a tentar melhorá-la
- Classificação das heurísticas construtivas
  - Construção aleatória
  - Construção gulosa
  - Construção parcialmente gulosa

Heurística de construção aleatória

- Funcionamento:
  - Constrói uma solução, elemento por elemento
  - A cada passo é adicionado um único elemento candidato na solução parcial
  - O candidato a ser adicionado é escolhido aleatoriamente dentre os elementos candidatos
  - O método se encerra quando todos os elementos candidatos foram analisados

Heurística de construção aleatória

procedimento *ConstrucaoAleatoria*$(g(.), s)$;

$s \leftarrow \emptyset$;

Inicialize o conjunto C de elementos candidatos;

enquanto $(C \neq \emptyset)$ faça

  Aleatoriamente escolha $t \in C$;

  se (t pode ser inserido na solução parcial) então

    $s \leftarrow s \cup \{t\}$; ;

  fim-se;

  $C \leftarrow C \setminus \{t\}$;

fim-enquanto;

Retorne s;

fim *ConstrucaoAleatoria*;

Heurística de construção gulosa

- Funcionamento:
  - Constrói uma solução, elemento por elemento
  - A cada passo é adicionado um único elemento candidato na solução parcial
  - O candidato escolhido é o “melhor” segundo um certo critério
  - O método se encerra quando todos os elementos candidatos foram analisados

Heurística de construção gulosa

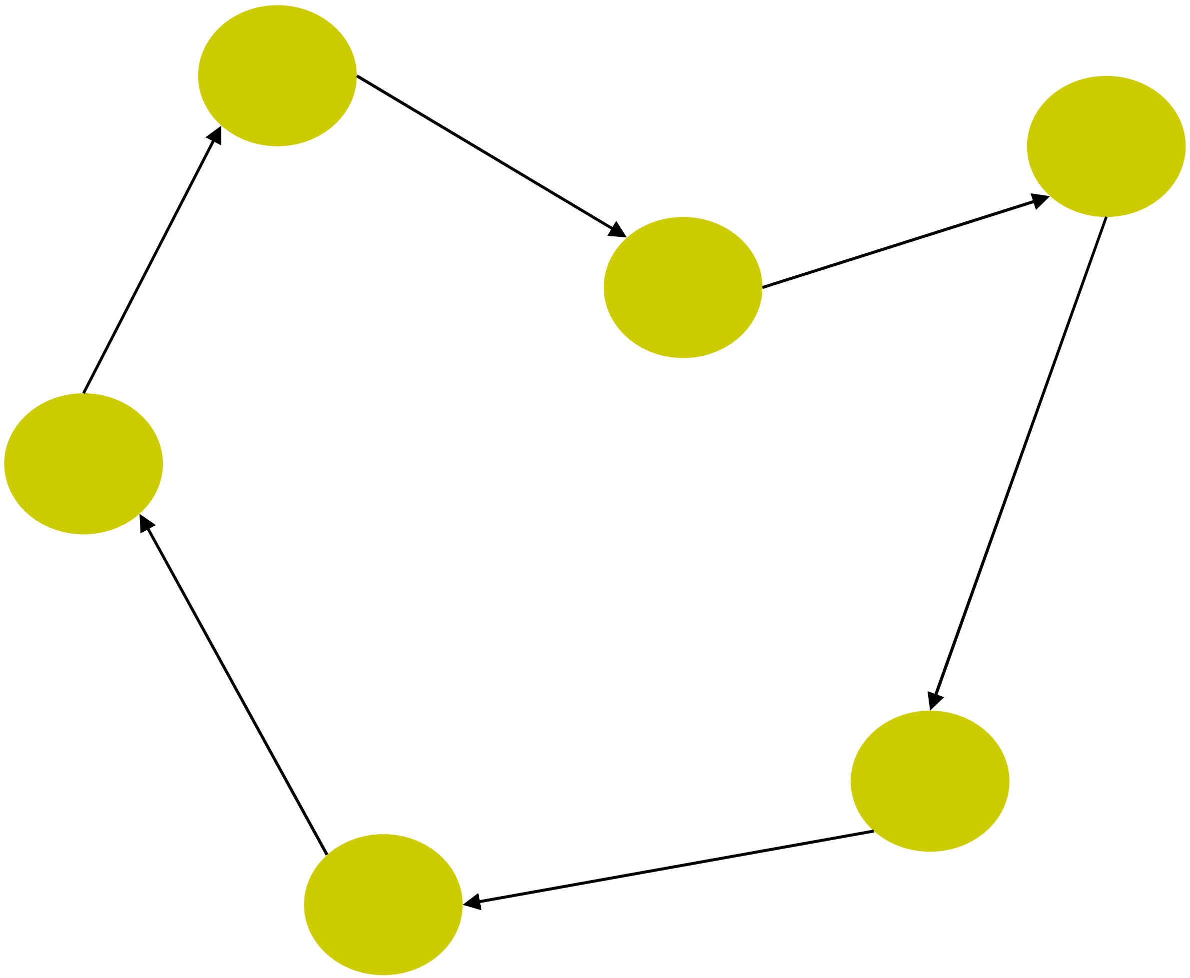
procedimento $ConstrucaoGulosa(g(.), s)$;
    $s \leftarrow \emptyset$;
    Inicialize o conjunto C de elementos candidatos;
    enquanto $(C \neq \emptyset)$ faça
        $g_{melhor} = melhor\{g(t) \mid t \in C\}$;
        Seja t_{melhor} o valor de $t \in C$ associado a g_{melhor};
        se (t_{melhor} pode ser inserido na solução parcial) então
            $s \leftarrow s \cup \{t_{melhor}\}$;
        fim-se;
        $C \leftarrow C \setminus \{t_{melhor}\}$;
    fim-enquanto;
    Retorne s;
fim $ConstrucaoGulosa$;

Melhor = max em um problema de maximização e
Melhor = min se o problema for de minimização

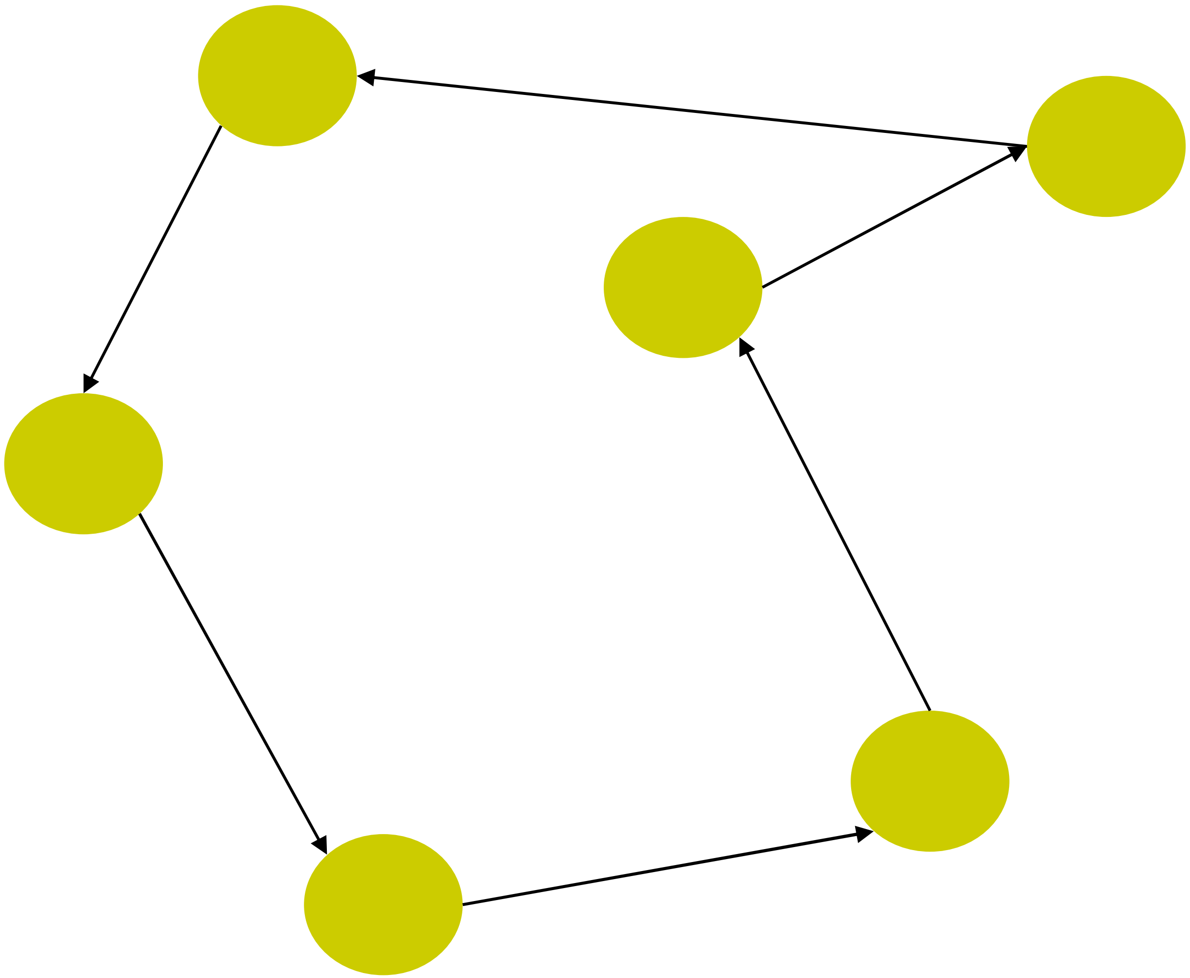
Problema do Caixeiro Viajante

- Dado um conjunto de cidades e uma matriz de distâncias entre elas
- PCV consiste em encontrar uma rota para um Caixeiro Viajante tal que este:
  - parta de uma cidade origem
  - passe por todas as demais cidades uma única vez
  - retorne à cidade origem ao final do percurso
  - percorra a menor distância possível
- Rota conhecida como ciclo hamiltoniano

Problema do Caixeiro Viajante



Problema do Caixeiro Viajante



Formulação Matemática para o Problema do Caixeiro Viajante

- Dados de entrada:
  - Cidades: Conjunto de cidades
  - d_{ij} = distância entre as cidades i e j
- Variáveis de decisão:
  - x_{ij} = 1 se a aresta (i,j) será usada; 0, caso contrário
  - f_{ij} = quantidade de fluxo de i para j
- Função objetivo:

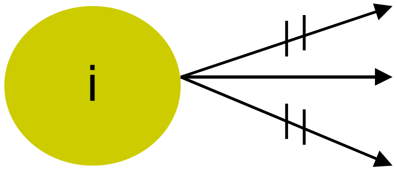
$$\min \sum_{i \in Cidades} \sum_{j \in Cidades} d_{ij} x_{ij}$$

Formulação Matemática para o Problema do Caixeiro Viajante

Restrições:

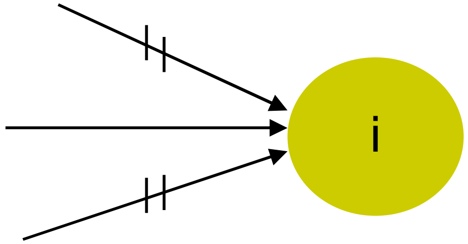
1. De cada cidade i só sai uma aresta:

$$\sum_{j \in Cidades} x_{ij} = 1 \quad \forall i \in Cidades$$



2. A cada cidade j só chega uma aresta:

$$\sum_{i \in Cidades} x_{ij} = 1 \quad \forall j \in Cidades$$



Formulação Matemática para o Problema do Caixeiro Viajante

3. O fluxo que chega a uma cidade *k* menos o que sai é igual a uma unidade:

$$\sum_{i \in Cidades} f_{ik} - \sum_{j \in Cidades} f_{kj} = 1 \;\; \forall\, k \in Cidades \mid k \neq 1$$

4. O fluxo máximo em cada aresta é igual a *n*-1, sendo *n* o número de cidades:

$$f_{ij} \leq (n-1)x_{ij} \quad \forall i \in Cidades, \forall j \in Cidades$$

5. Integralidade e não negatividade:

$$f_{ij} \geq 0 \quad \forall i \in Cidades, \forall j \in Cidades$$

$$x_{ij} \in \{0,1\} \quad \forall i \in Cidades, \forall j \in Cidades$$

Problema do Caixeiro Viajante Complexidade

- PCV é da classe NP-difícil: ainda não existem algoritmos exatos que o resolva em tempo polinomial
- No pior caso é necessário analisar toda a árvore de busca
- À medida que *n* cresce, o tempo de execução para obter a solução ótima pode ser proibitivo
- Inviável resolvê-lo exclusivamente por métodos exatos
- Resolvido por métodos heurísticos

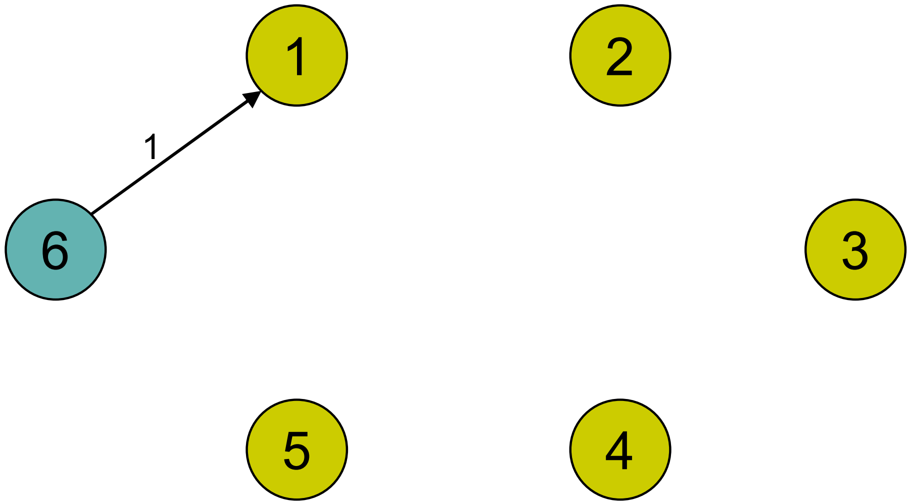
Heurísticas construtivas para o Problema do Caixeiro Viajante

- Vizinho mais próximo
  - Idéia central:
    - Construir uma rota passo a passo, adicionando à solução corrente a cidade mais próxima (ainda não visitada) da última cidade inserida;
    - Encerrar o procedimento construtivo quando todas as cidades forem visitadas. Neste caso, fazer a ligação da última cidade visitada à primeira.

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

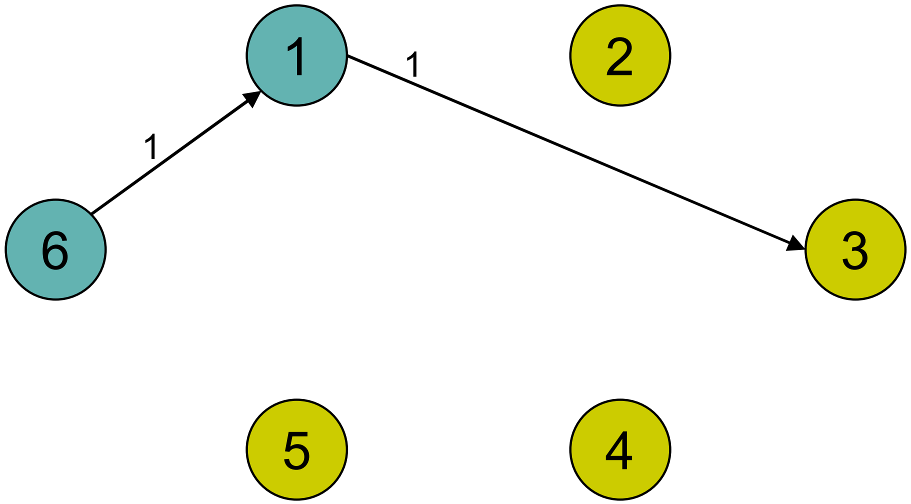


- **Distância Total =** 1

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

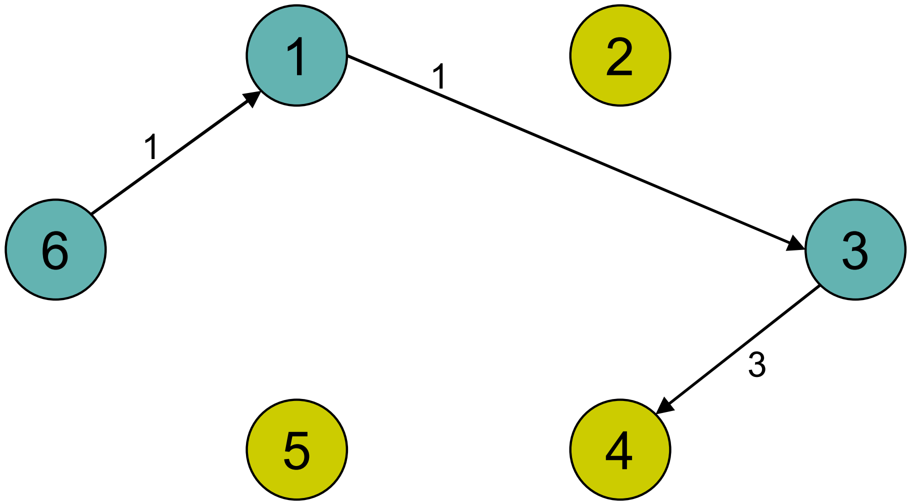


- **Distância Total =** 1 + 1 = 2

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

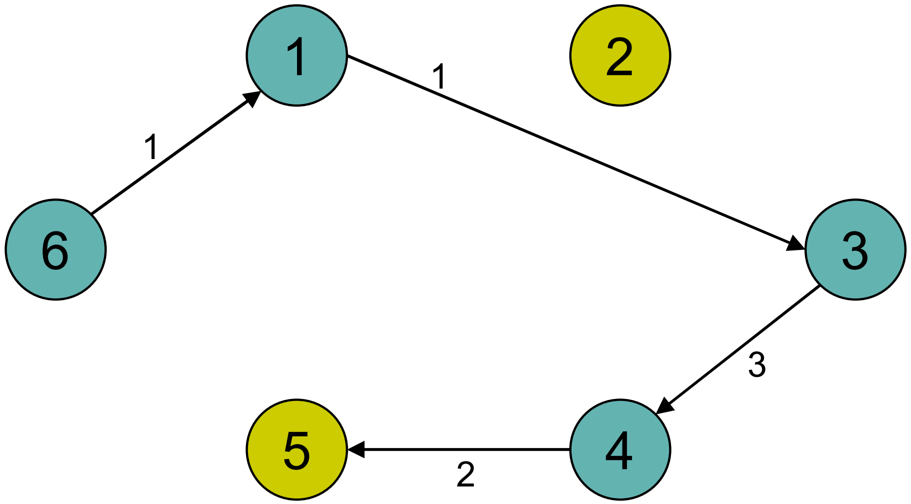


- **Distância Total =** 2 + 3 = 5

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

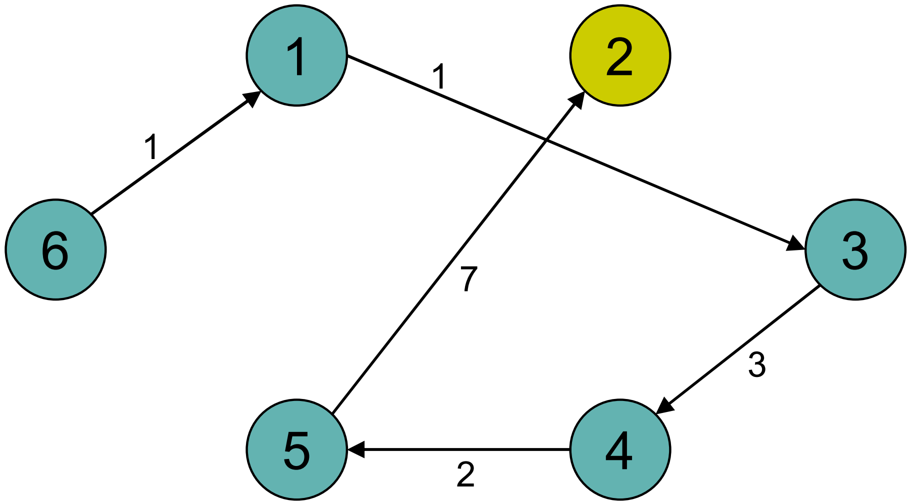


- **Distância Total =** 5 + 2 = 7

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

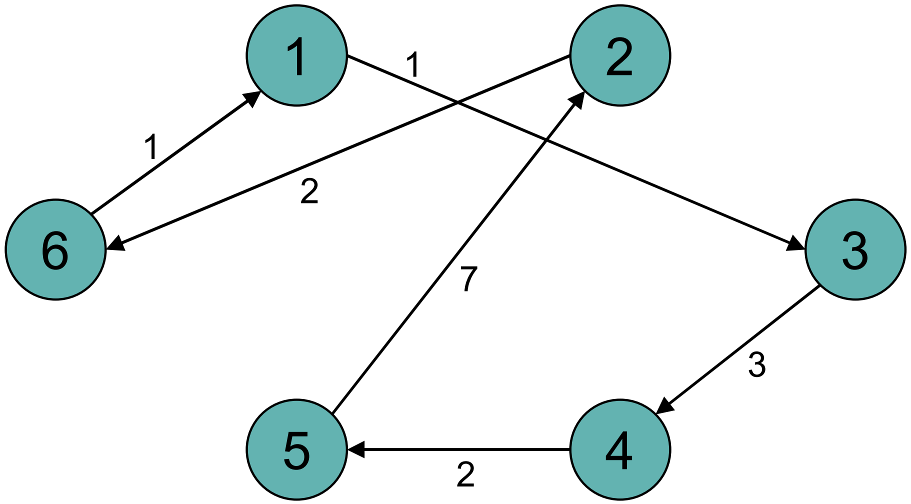
i	j	d_{ij}
5	**2**	7



- **Distância Total =** 7 + 7 = 14

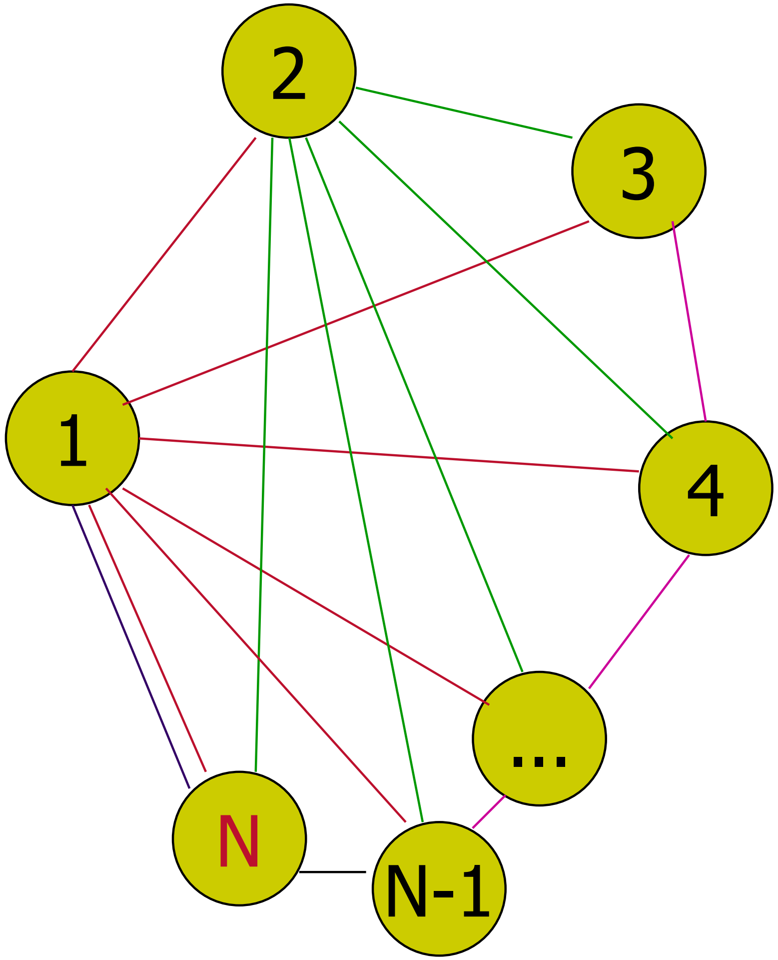
PCV – Vizinho mais Próximo
Exemplo – Passo final: “Inserção forçada”

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

Complexidade da heurística construtiva do vizinho mais próximo aplicada ao PCV



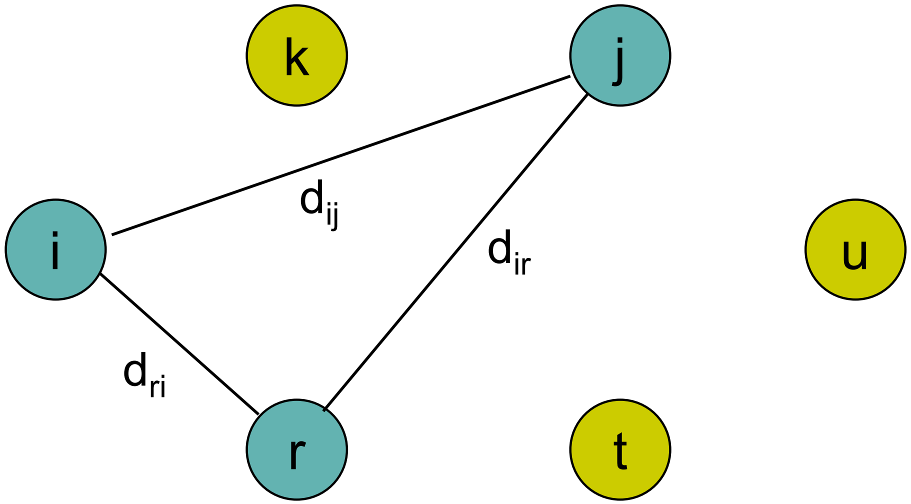
Iteração	Número de avaliações
1	N-1
2	N-2
...	...
N-1	1
N	1
Total	1 + N(N-1)/2

Heurística da Inserção Mais Barata para o Problema do Caixeiro Viajante

- Inserção mais barata
  - Idéia central:
    - Construir uma rota passo a passo, partindo de rota inicial envolvendo 3 cidades (obtidas por um método qualquer) 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;
    - Encerrar o método quando todas as cidades forem visitadas.

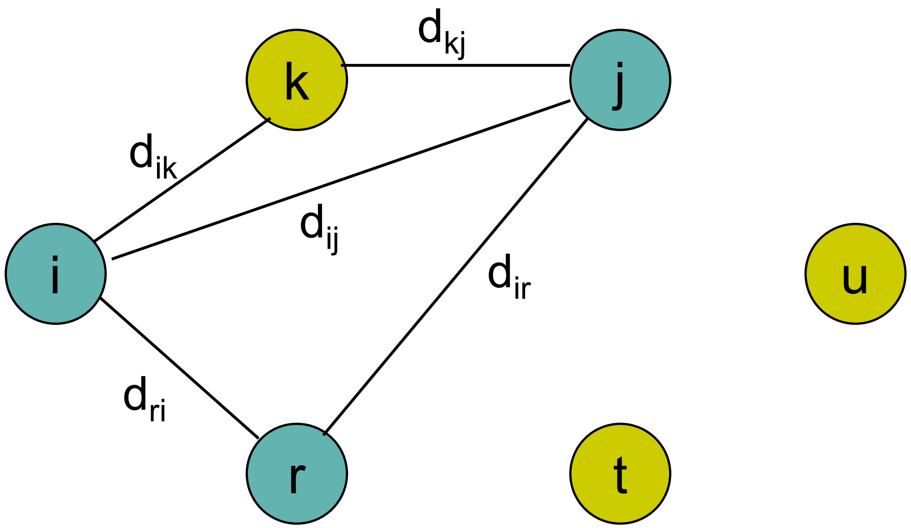
Heurística da Inserção Mais Barata para o Problema do Caixeiro Viajante

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



Heurística da Inserção Mais Barata para o Problema do Caixeiro Viajante

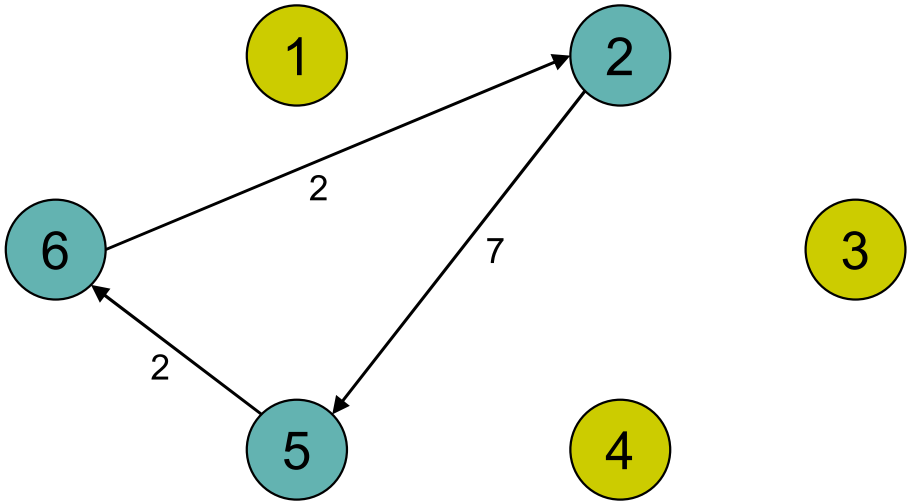
Cid.	1	2	3	4	5	6
1	0	2	1	4	9	1
2	2	0	5	9	7	2
3	1	5	0	3	8	6
4	4	9	3	0	2	6
5	9	7	8	2	0	2
6	1	2	6	6	2	0



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

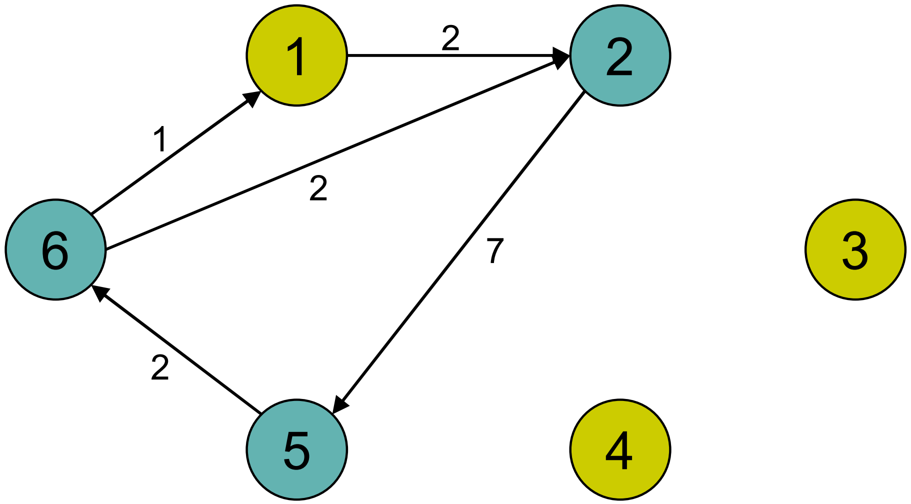


- **Distância Total =** 11

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

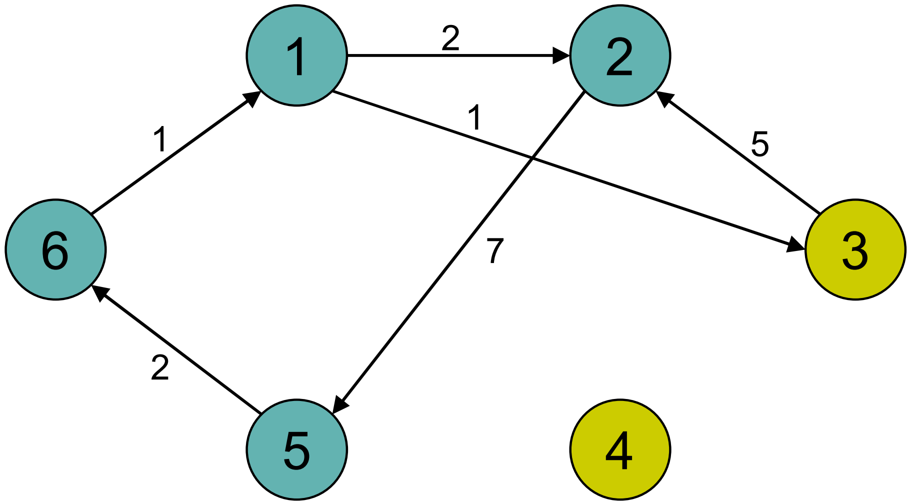


- **Distância Total =** 11 + 1 = 12

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**	$2 + 6 - 2 = 6$

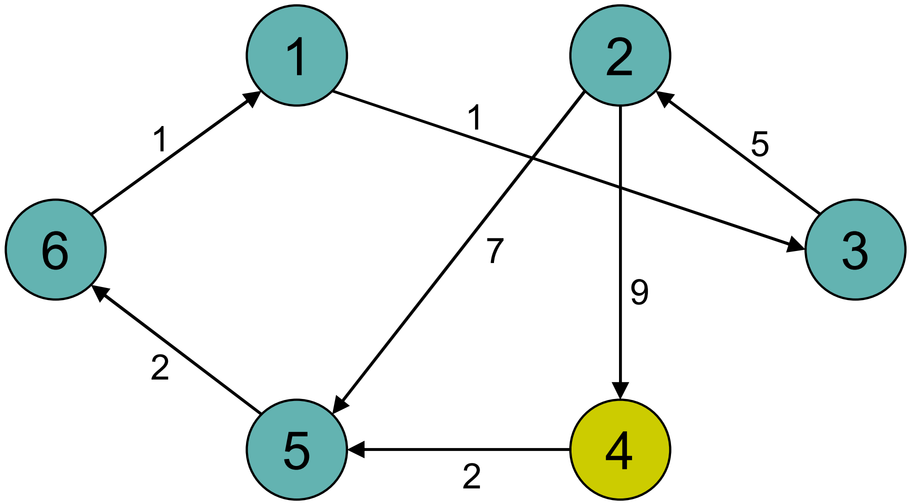


- **Distância Total =** 12 + 4 = 16

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

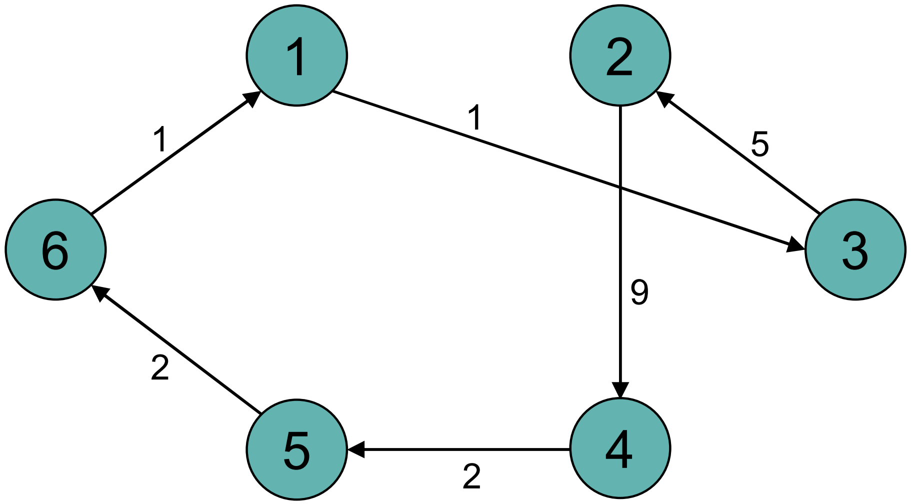
i	k	j	$d_{ik} + d_{kj} - d_{ij}$
6	**4**	**1**	6 + 4 – 1 = 9
1	**4**	**3**	4 + 3 – 1 = 6
3	**4**	**2**	3 + 9 – 5 = 7
2	**4**	**5**	9 + 2 – 7 = 4
5	**4**	**6**	2 + 6 – 2 = 6



- **Distância Total =** 16 + 4 = 20

PCV – Inserção mais Barata Exemplo – Solução final

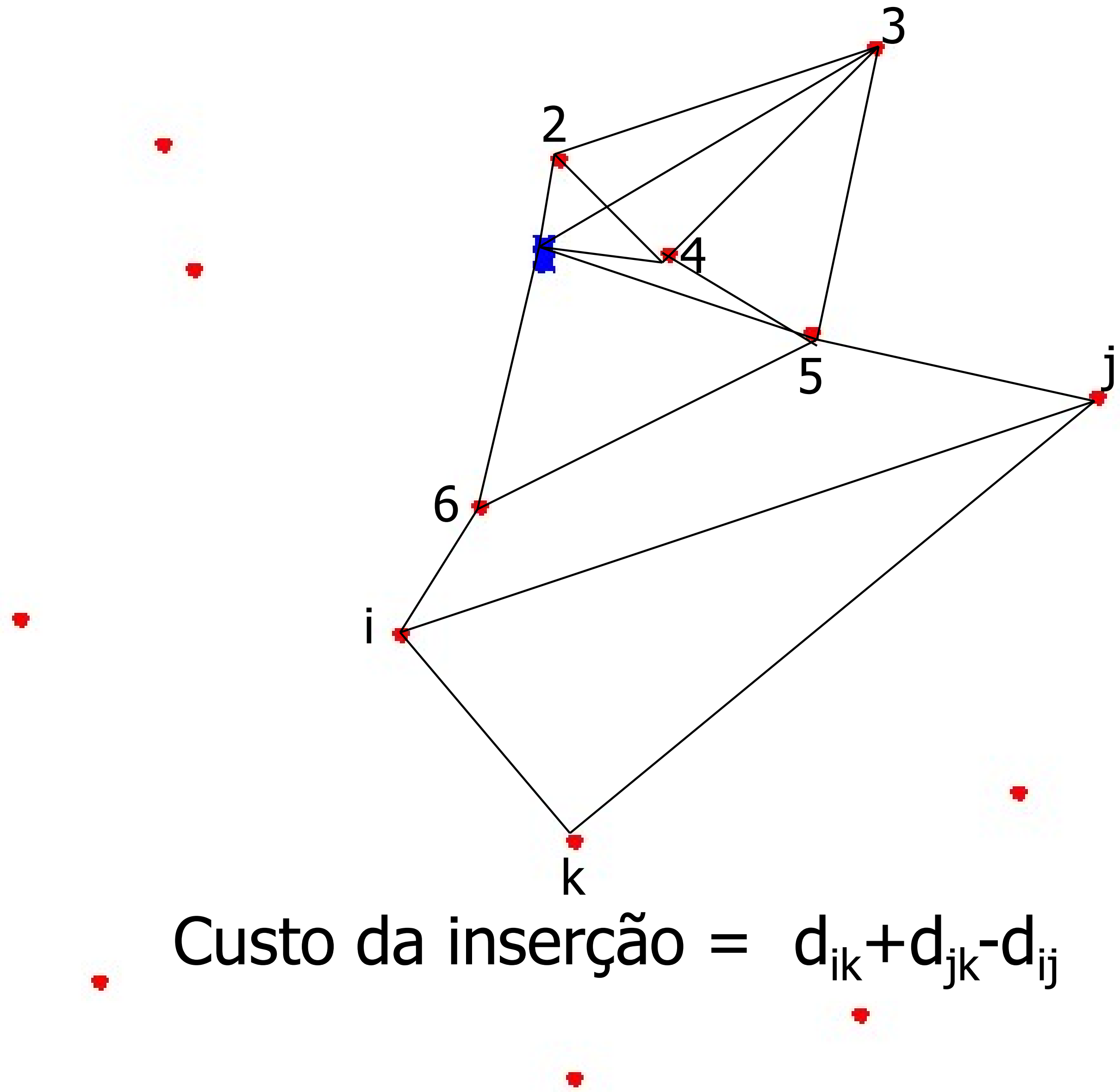
Cid.	1	2	3	4	5	6
1	0	2	1	4	9	1
2	2	0	5	9	7	2
3	1	5	0	3	8	6
4	4	9	3	0	2	6
5	9	7	8	2	0	2
6	1	2	6	6	2	0



- **Distância Total =** 16 + 4 = 20

s = (6 1 3 2 4 5)

Complexidade da heurística construtiva da inserção mais barata aplicada ao PCV



Custo da inserção = $d_{ik}+d_{jk}-d_{ij}$

Complexidade da heurística construtiva da inserção mais barata aplicada ao PCV

Iteração	Número de avaliações
1	3(N - 3)
2	4(N - 4)
...	...
i-2	i(N-i)
...	...
N-3	(N – 1)(N-(N-1))
Total	$\sum_{i=3}^{n-1} i(n-i)$

$$\sum_{i=3}^{n-1} i(n-i) = \frac{1}{6}n^3 - \frac{19}{6}n \quad + 5$$

Soma obtida a partir da aplicação da fórmula da soma dos quadrados dos n primeiros números naturais:

$$S = \frac{(n+1)n(2n+1)}{6}$$

Comparação entre as heurísticas construtivas para o PCV

Exemplo para n = 20 cidades

Método	Complexidade	Tempo (s)
Vizinho mais próximo	$\frac{1}{2}n^2 - n + 1$	$1,8 \times 10^{-6}$
Inserção mais barata	$\frac{1}{6}n^3 - \frac{19}{6}n + 5$	$12,75 \times 10^{-6}$

Supor uma avaliação executada em 10^{-8} segundos

Comparação entre as heurísticas construtivas para o PCV

Exemplo para n = 1000 cidades

Método	Complexidade	Tempo (s)
Vizinho mais próximo	$\frac{1}{2}n^2 - n + 1$	0,005
Inserção mais barata	$\frac{1}{6}n^3 - \frac{19}{6}n \quad + 5$	1,667

Supor uma avaliação executada em 10^{-8} segundos

Comparação entre as heurísticas construtivas para o PCV

Exemplo para n = 10000 cidades

Método	Complexidade	Tempo (s)
Vizinho mais próximo	$\frac{1}{2}n^2 - n + 1$	0,5
Inserção mais barata	$\frac{1}{6}n^3 - \frac{19}{6}n + 5$	1667

Supor uma avaliação executada em 10^{-8} segundos