

# Introdução aos Problemas de Roteirização e Programação de Veículos

PNV-2450

**André Bergsten Mendes**



# HEURÍSTICA DE INSERÇÃO DE SOLOMON (**VRPTW**)

# Exercício

- Para o problema de roteirização com janela de tempo indicado na tabela, gere uma solução por meio da Heurística de Inserção de Solomon. Adote os valores que julgar conveniente para  $\alpha_1$ ,  $\alpha_2$ ,  $\mu$  e  $\lambda$ .

# Exercício

- 7 clientes (os clientes 0 e 8 referem-se à base)
- A operação de distribuição inicia-se às 7hs, com os veículos já carregados, e é encerrada às 18hs
- Há, no máximo, 3 veículos homogêneos, com capacidade igual a 50 unidades de carga, cada
- As colunas “a” e “b” referem-se aos limites inferior e superior da janela de tempo para chegada aos clientes
- A coluna “Serviço” indica o tempo de atendimento do veículo junto a cada cliente

# Exercício

Cliente	Demanda (unidades)	a (h)	b (h)	Serviço (h)
0	0	7	18	0
1	10	8	8,5	1
2	7	11,5	13	1
3	13	12	14	1
4	19	9	10	1
5	26	15	15,5	1
6	9	14	15	1
7	11	16	17	1
8	0	7	18	0

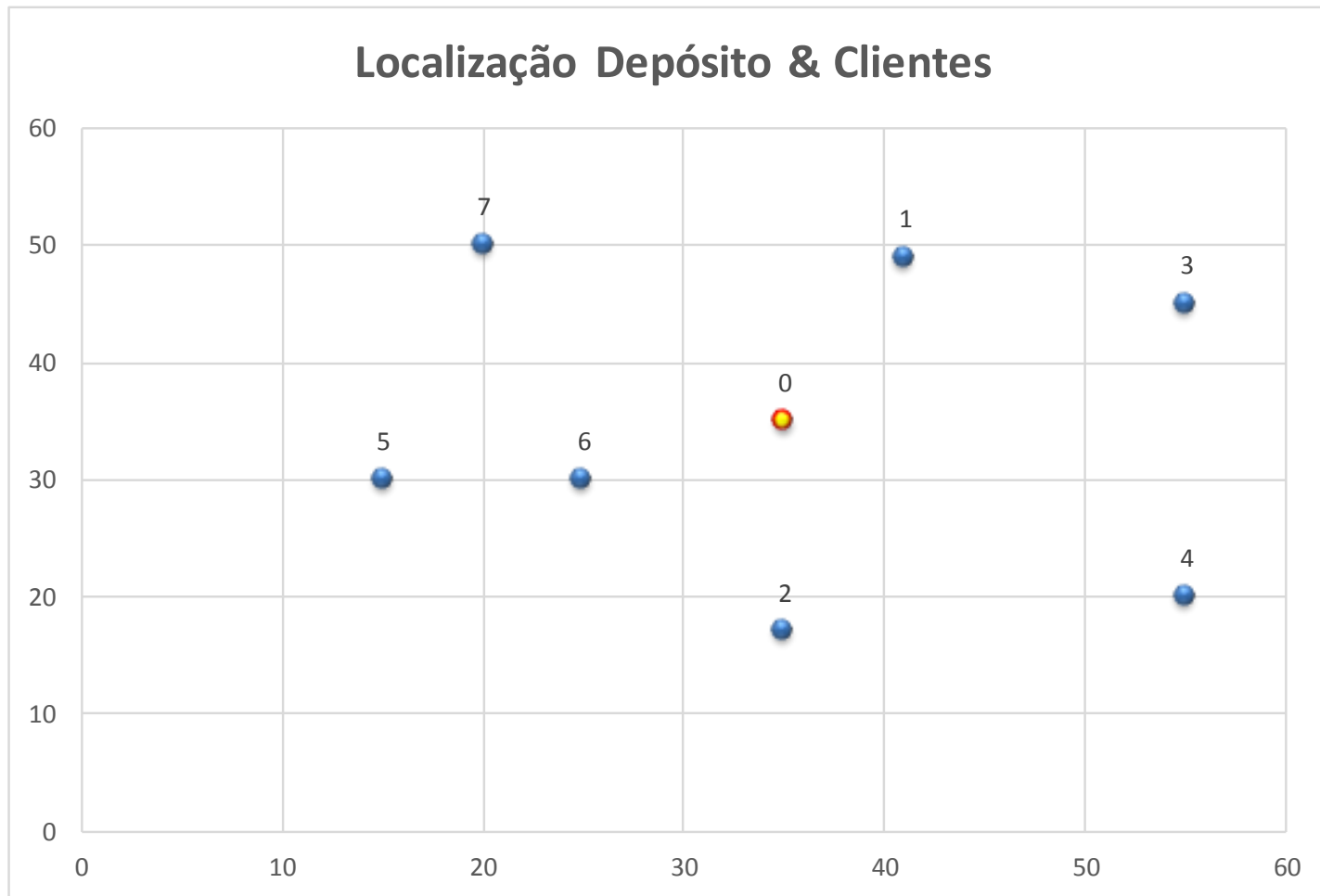
# Matriz de Distância (km)

	0	1	2	3	4	5	6	7	8
0	0	15,2	18	22,3	25	20,6	11,1	21,2	0
1	15,2	0	32,5	14,5	32,2	32,2	24,8	21	15,2
2	18	32,5	0	34,4	20,2	23,8	16,4	36,2	18
3	22,3	14,5	34,4	0	25	42,7	33,5	35,3	22,3
4	25	32,2	20,2	25	0	41,2	31,6	46	25
5	20,6	32,2	23,8	42,7	41,2	0	10	20,6	20,6
6	11,1	24,8	16,4	33,5	31,6	10	0	20,6	11,1
7	21,2	21	36,2	35,3	46	20,6	20,6	0	21,2
8	0	15,2	18	22,3	25	20,6	11,1	21,2	0

# Tempo de Deslocamento (h)

	0	1	2	3	4	5	6	7	8
0	0	0,7	0,9	1,1	1,2	1	0,5	1	0
1	0,7	0	1,6	0,7	1,6	1,6	1,2	1	0,7
2	0,9	1,6	0	1,7	1	1,1	0,8	1,8	0,9
3	1,1	0,7	1,7	0	1,2	2,1	1,6	1,7	1,1
4	1,2	1,6	1	1,2	0	2	1,5	2,3	1,2
5	1	1,6	1,1	2,1	2	0	0,5	1	1
6	0,5	1,2	0,8	1,6	1,5	0,5	0	1	0,5
7	1	1	1,8	1,7	2,3	1	1	0	1
8	0	0,7	0,9	1,1	1,2	1	0,5	1	0

# Heurística de Inserção de Solomon





# Heurística de Inserção de Solomon


- Inicialização da primeira rota: cliente 1, pois possui janela de tempo mais apertada

Cliente	Demanda (unidades)	a (h)	b (h)	Serviço (h)
0	0	7	18	0
1	10	8	8,5	1
2	7	11,5	13	1
3	13	12	14	1
4	19	9	10	1
5	26	15	15,5	1
6	9	14	15	1
7	11	16	17	1
8	0	7	18	0

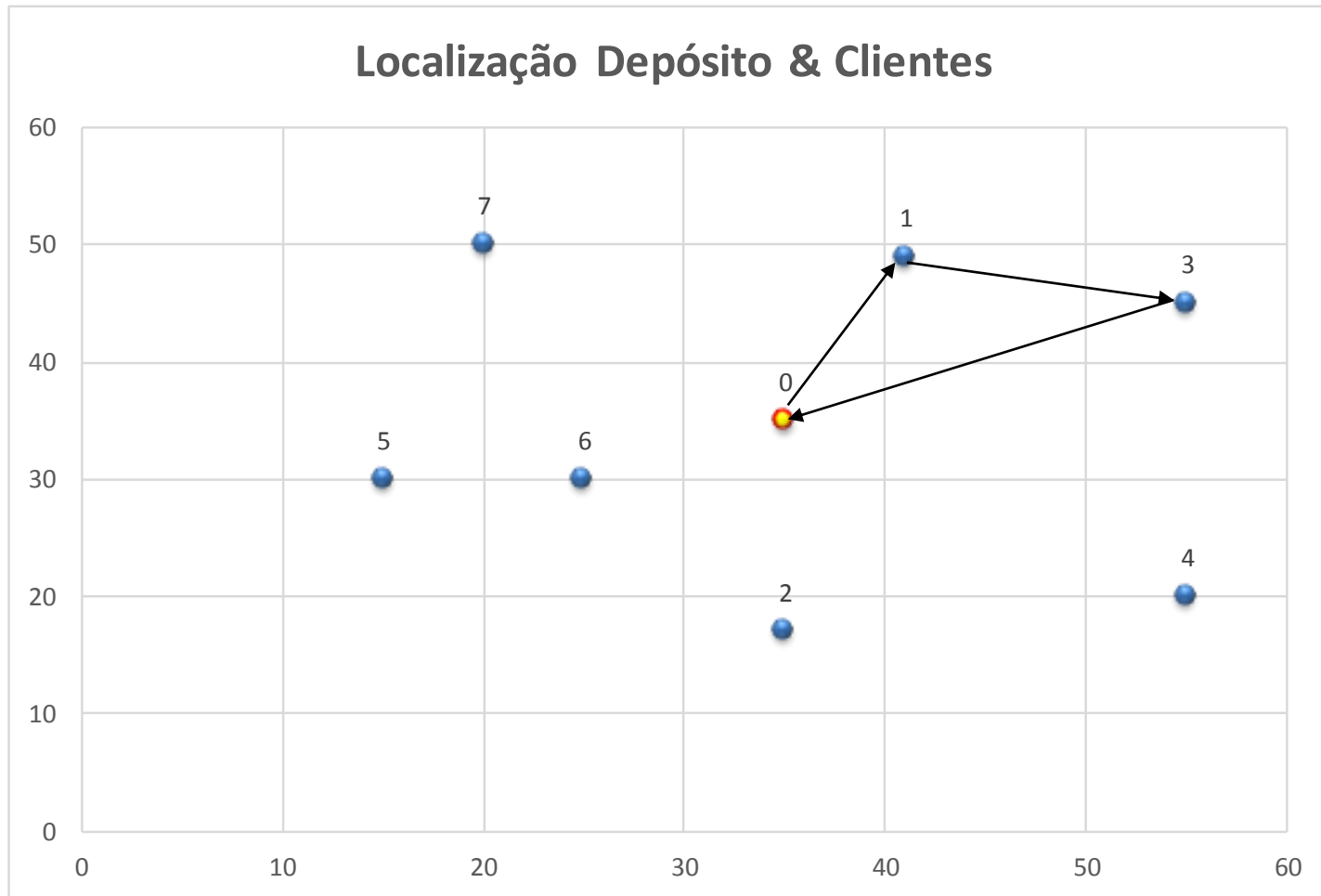
# Heurística de Inserção de Solomon

- Possíveis inserções:

	Dem	Ac.	Inst.	Cheg.	Janela	c1	c2		
0	1	2	8	17	10,6	11,5	13,0	35,3	-17,3
0	1	3	8	23	9,7	12,0	14,0	21,6	0,7
0	1	4	8	29	10,6	9,0	10,0		
0	1	5	8	36	10,6	15,0	15,5	37,6	-17,0
0	1	6	8	19	10,2	14,0	15,0	20,7	-9,6
0	1	7	8	21	10,0	16,0	17,0	27,0	-5,8




# Heurística de Inserção de Solomon



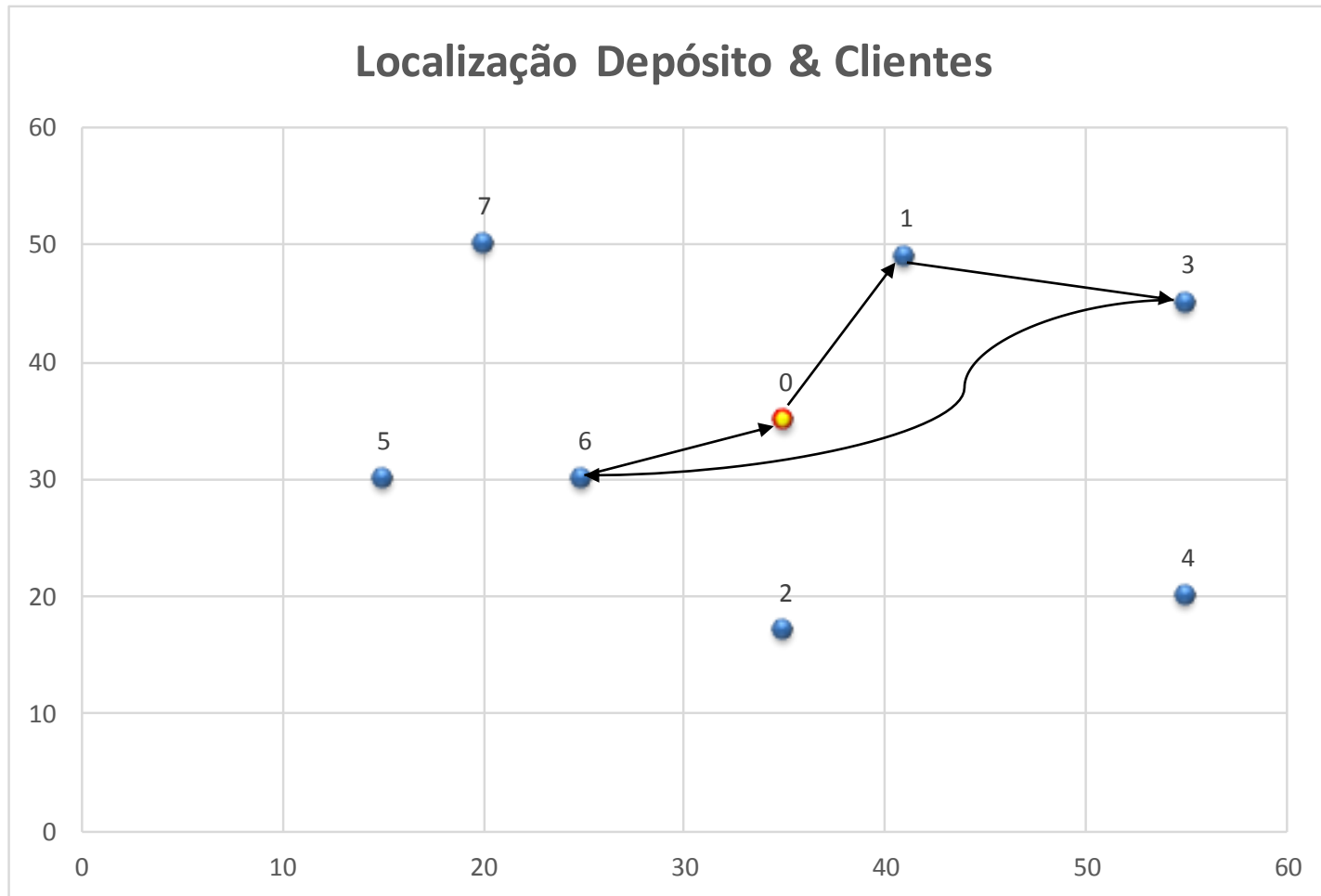
# Heurística de Inserção de Solomon

## ■ Possíveis inserções:

	0	1	2	3	8	Dem	Ac.	Inst.	Cheg.	Janela	c1	c2	
	0	1	2	3	8	30			10,6	11,5	13,0		
	0	1	3	2	8	30			14,7	11,5	13,0		
	0	1	4	3	8	42			10,6	9,0	10,0		
	0	1	3	5	8	49			15,1	15,0	15,5	41,0	-20,4
	0	1	3	6	8	32			14,6	14,0	15,0	22,3	-11,2
	0	1	3	7	8	34			14,7	16,0	17,0	34,2	-13,0



# Heurística de Inserção de Solomon



# Heurística de Inserção de Solomon


- Inicialização da segunda rota: cliente 4, pois possui janela de tempo mais apertada

Cliente	Demanda (unid)	a	b	Serviço (h)
0	0	7,0	18,0	0
1	10	8,0	8,5	1
2	7	11,5	13,0	1
3	13	12,0	14,0	1
4	19	9,0	10,0	1
5	26	15,0	15,5	1
6	9	14,0	15,0	1
7	11	16,0	17,0	1
8	0	7,0	18,0	0

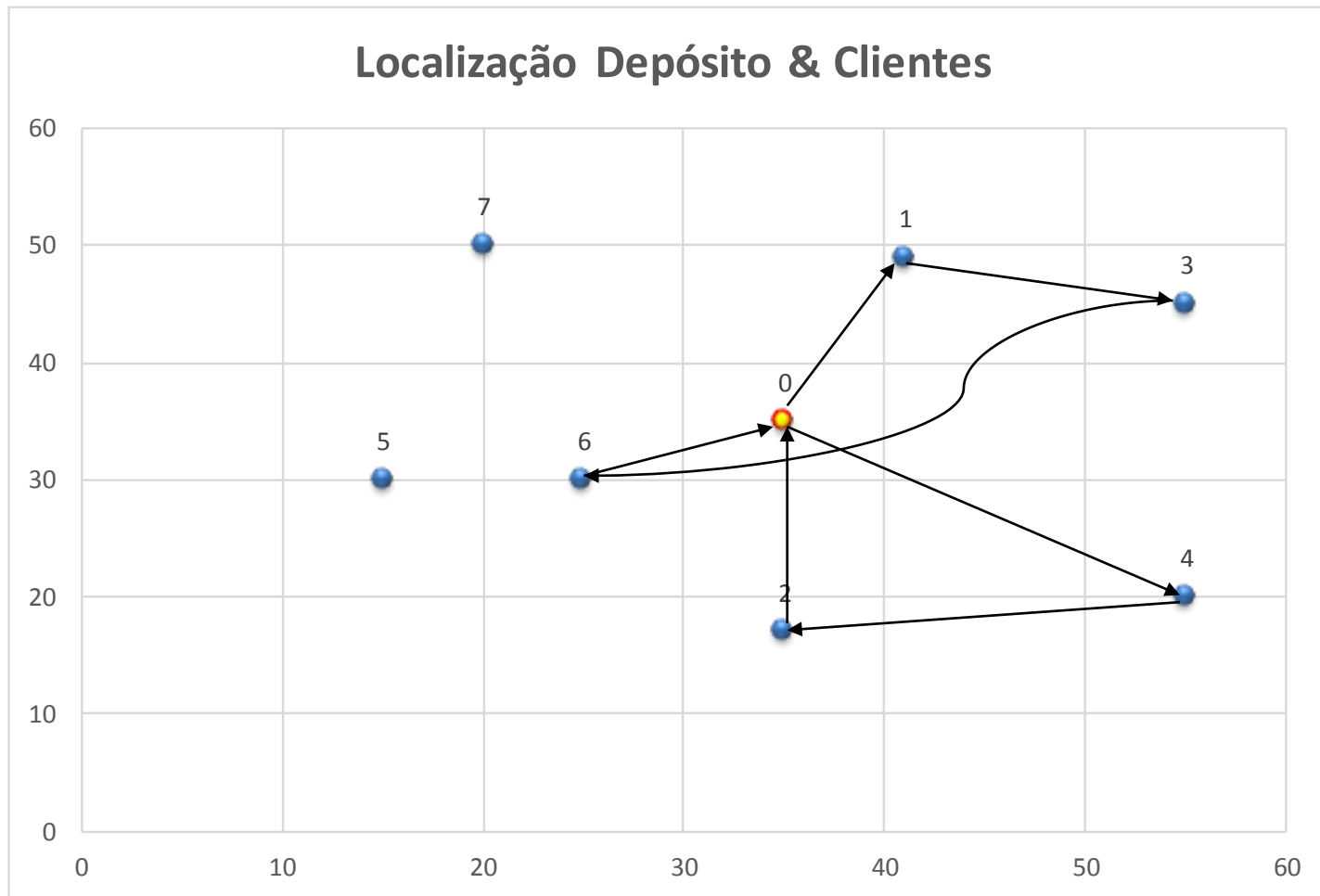
# Heurística de Inserção de Solomon

## ■ Possíveis inserções:

	Dem	Ac.	Inst.	Cheg.	Janela	c1	c2
0	4	2	8	26	11,5 13,0	13,2	4,8
0	4	5	8	45	15,0 15,5	36,8	-16,2
0	4	7	8	30	16,0 17,0	42,2	-21,0



# Heurística de Inserção de Solomon






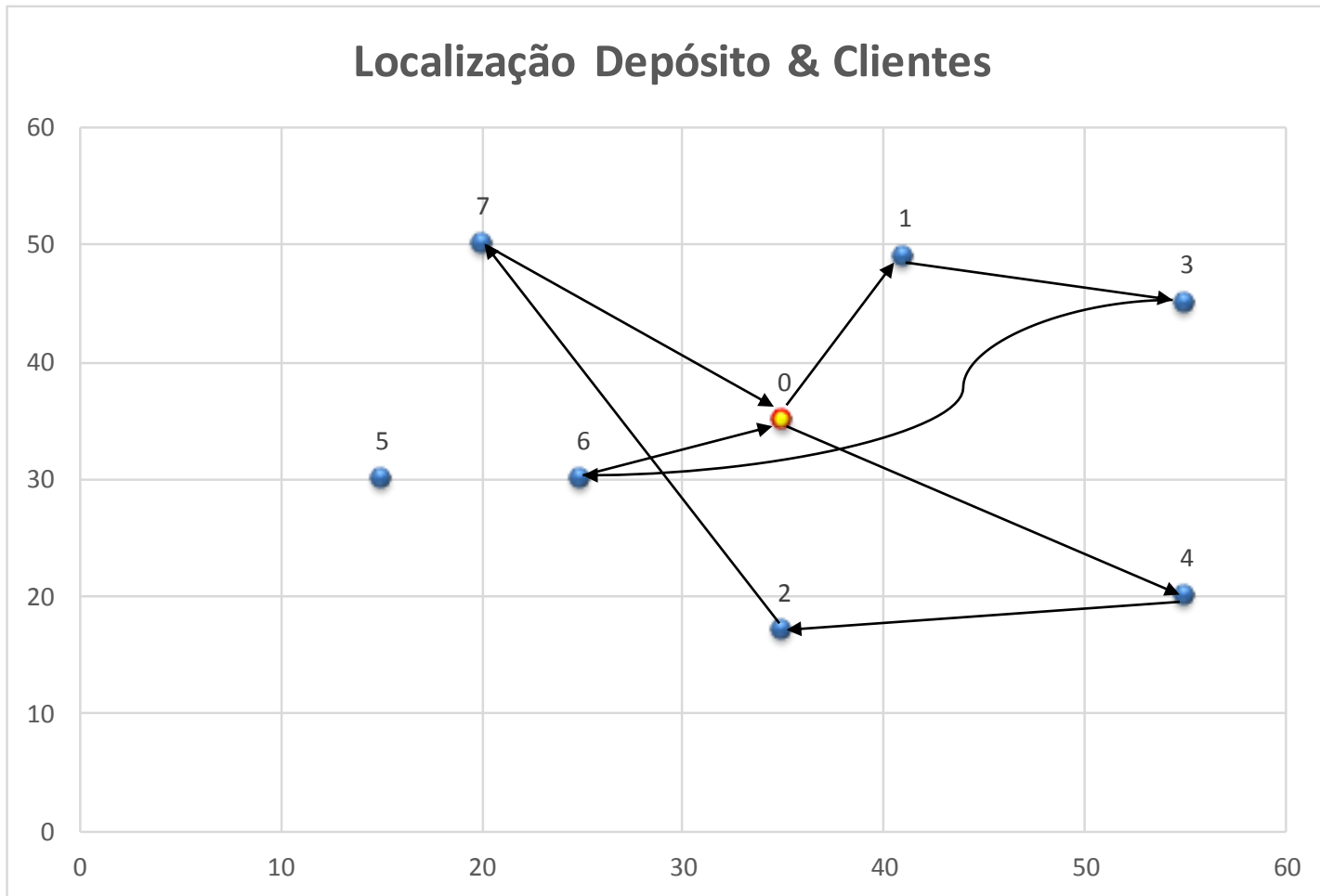
# Heurística de Inserção de Solomon

- Possíveis inserções:

0	4	2	7	8	Dem	Ac.	Inst.	Cheg.	Janela	c1	c2	
0	4	2	5	8		37		13,8	16,0	17,0	54,4	-33,2
						52			15,0	15,5		



# Heurística de Inserção de Solomon

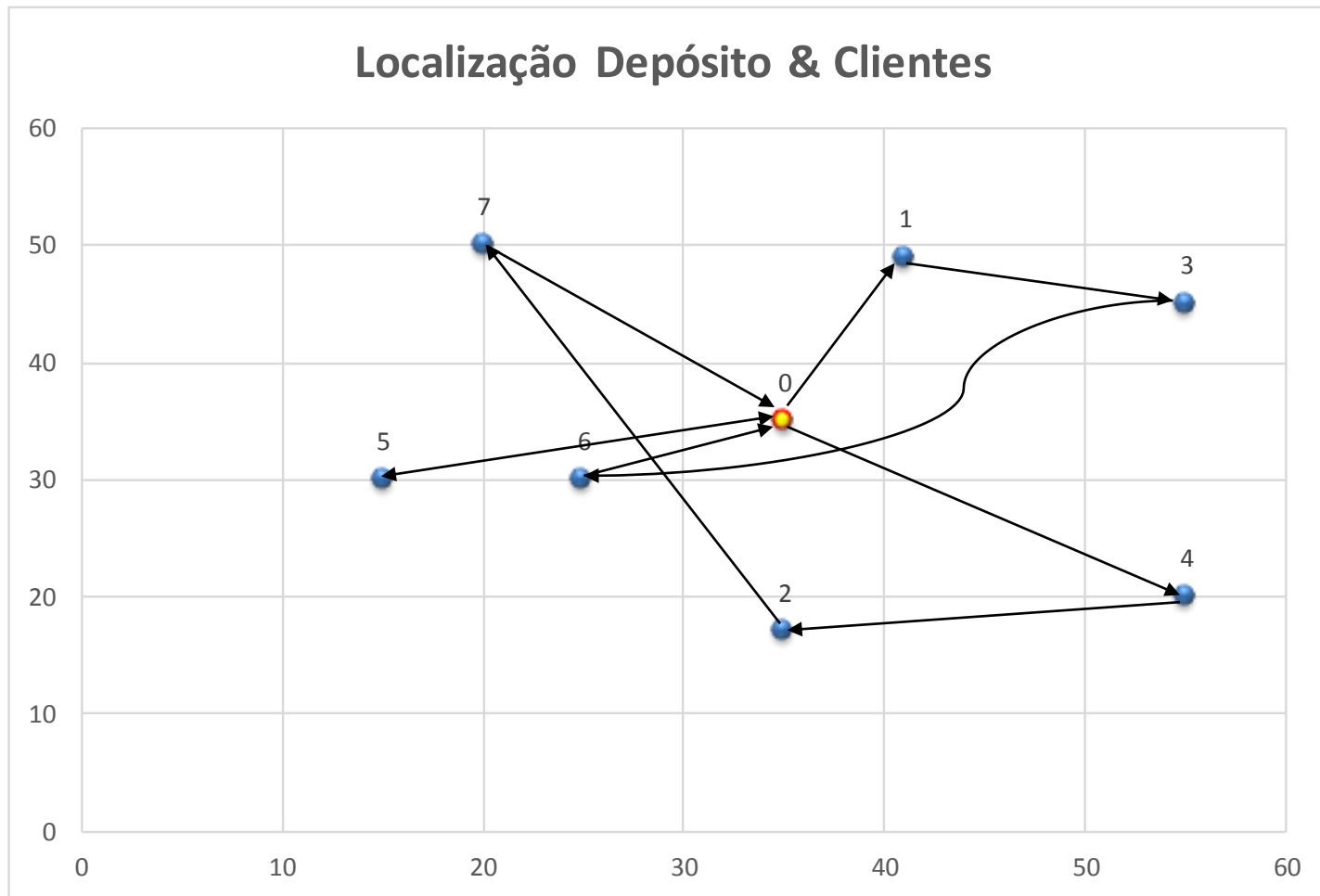


# Heurística de Inserção de Solomon

- Inicialização da terceira rota: cliente 5, o único que sobrou

Cliente	Demanda (unid)	a	b	Serviço (h)
0	0	7,0	18,0	0
1	10	8,0	8,5	1
2	7	11,5	13,0	1
3	13	12,0	14,0	1
4	19	9,0	10,0	1
5	26	15,0	15,5	1
6	9	14,0	15,0	1
7	11	16,0	17,0	1
8	0	7,0	18,0	0

# Heurística de Inserção de Solomon



# Solução Ótima

## Veículo Rota

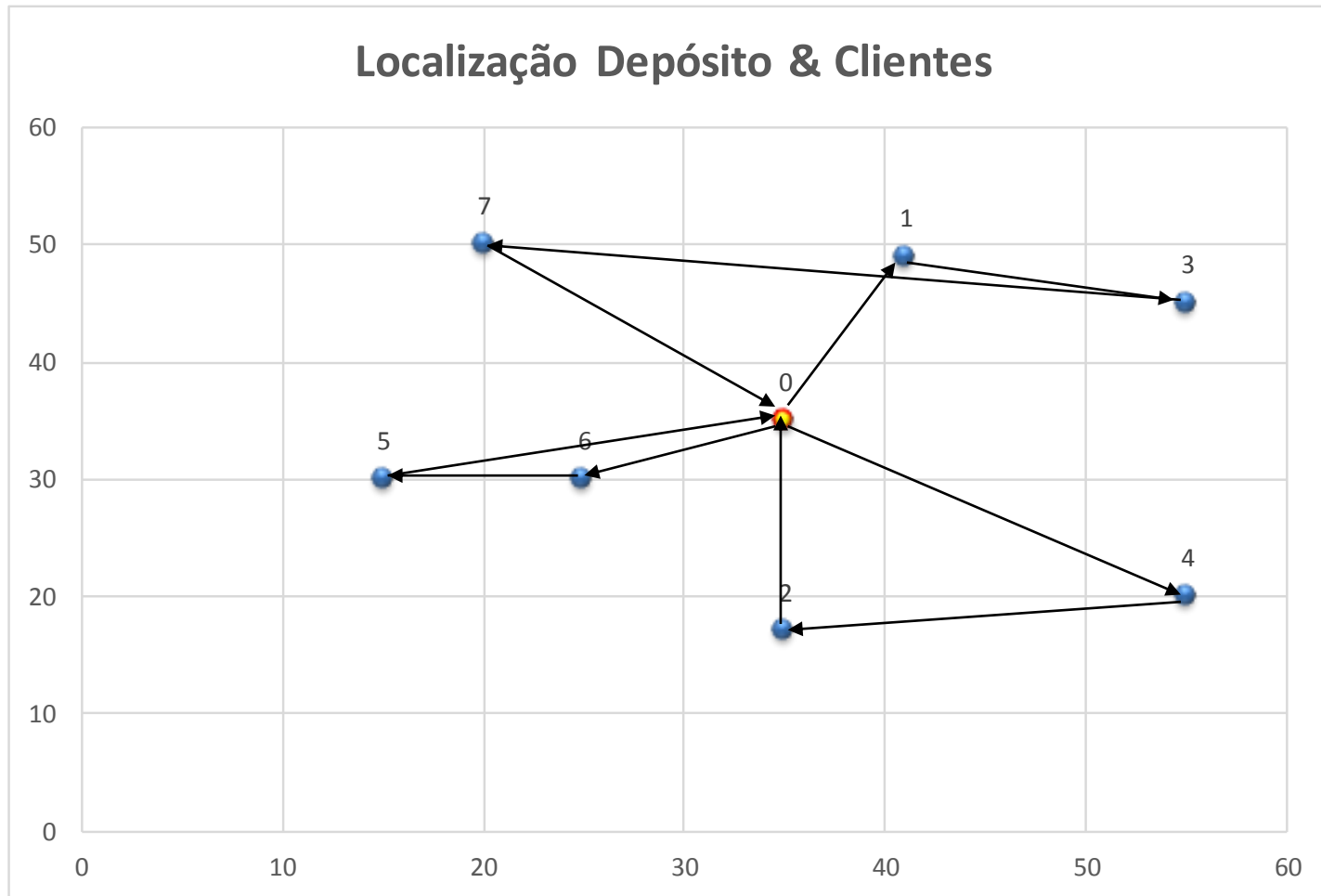
1	0	4	2	8	
2	0	1	3	7	8
3	0	6	5	8	

## Veículo Instantes

1	7	9	11,5	
2	7	8	12	16
3	7	14	15,5	

**Distância total = 191,1**

# Heurística de Inserção de Solomon (Solução ótima)





# GRASP

Feo, T.A., & Resende, M.G.C. (1995). Greedy randomized adaptive search procedures. *Journal of Global Optimization* 6, 109–133.

# GRASP

## Greedy Randomized Adaptive Search Procedure

**procedure** ConstructSemiGreedySolution( $\alpha$ )

Initialize solution:  $S \leftarrow \emptyset$ ;

Initialize candidate set:

$C \leftarrow \{s \in E \setminus S \mid S \cup \{s\} \text{ is not infeasible}\}$ ;

**while**  $C \neq \emptyset$  **do**

$g_{\min} \leftarrow \min\{g(s) \mid s \in C\}$

$g_{\max} \leftarrow \max\{g(s) \mid s \in C\}$

$RCL \leftarrow \{s \in C \mid g(s) \leq g_{\min} + \alpha (g_{\max} - g_{\min})\}$

Select  $s \in RCL$ ;

Add  $s$  to solution:  $S \leftarrow S \cup \{s\}$ ;

Update candidate set:

$C \leftarrow \{s \in E \setminus S \mid S \cup \{s\} \text{ is not infeasible}\}$ ;

**end**

**return**  $S$ ;

Fonte: Marti et al. (2013)

$g(s)$  – contribution to the cost of the solution  
under construction; RCL – *Restricted candidate list*



# GRASP

## Greedy Randomized Adaptive Search Procedure

procedure GRASP( $\alpha$ )

$f^* \leftarrow \infty$ ;

**while** stopping criterion not satisfied **do**

Construct feasible randomized semi greedy solution:

$S \leftarrow \text{ConstructSemiGreedySolution}(\alpha)$ ;

Find a locally optimal solution:

$S \leftarrow \text{LocalSearch}(S)$ ;

**if**  $f(S) < f^*$  **then**

$S^* \leftarrow S$ ;

$f^* \leftarrow f(S)$ ;

**end**

**end**

**return**  $S^*$ ;



# VNS

Mladenović, N., & Hansen, P. (1997). Variable neighborhood search.  
*Computers and Operations Research*, 24(11), 1097–1100.

# VNS

## Variable Neighborhood Search

**Algorithm:** Basic VNS

**Input:** The set of neighborhood structures  $N_k$  for  $k=1, \dots, k_{\max}$

```
01 Initialization: Find an initial solution  $x$ ;  
02 repeat  
03    $k \leftarrow 1$ ;  
04   while  $k \leq k_{\max}$  do  
05      $x' \leftarrow$  a random neighbor in  $N_k(x)$  // Shaking  
06      $x'' \leftarrow$  Local Search( $x'$ ) // Local Search  
07     if  $f(x'') < f(x)$  then // Move or Not  
08        $x \leftarrow x''$ ;  $k \leftarrow 1$ ;  
09     else  
10        $k \leftarrow k+1$ ; // Neighborhood Change  
11 until termination condition is met  
12 return  $x$ ;
```

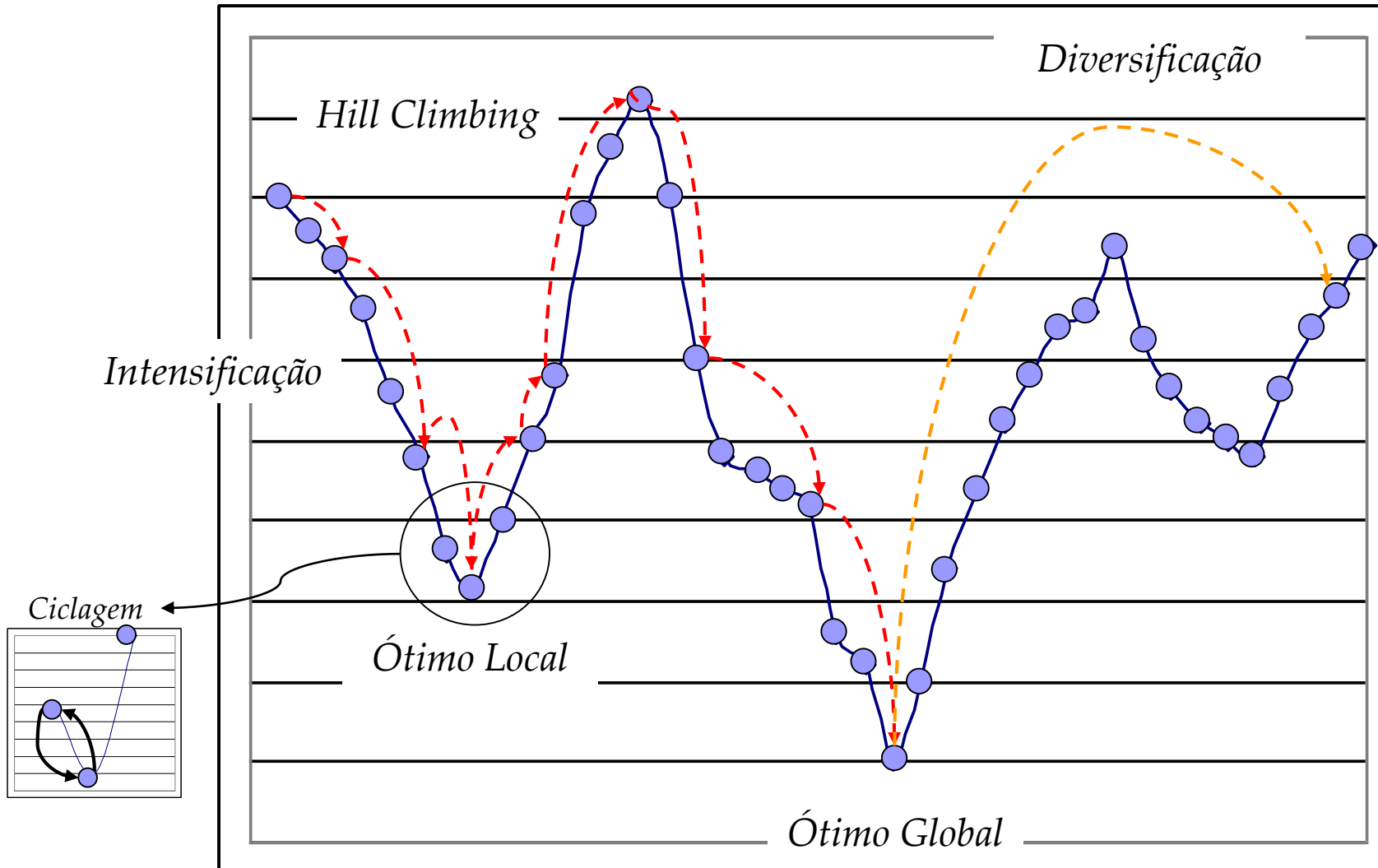


# **BUSCA TABU**

# Busca Tabu

- Proposto por Glover e Laguna
- Heurística iterativa genérica para resolução de problemas de otimização combinatória
- Extensão da Busca Local
- Princípio básico: uma memória força a exploração de novas áreas do espaço de busca. As soluções que foram examinadas recentemente se tornam tabus (proibidas) de serem escolhidas para as próximas soluções

# Busca Tabu



# Busca Tabu

$x_{cur}$  ← solução corrente (= solução inicial);  $x_{opt}$  ← melhor solução (= solução inicial)

diversificação = falso

**enquanto** *critério de parada não for atingido*

**se** diversificação = verdadeiro **então**

**utilizar** *memória de longo prazo*

**senão**

$x' \leftarrow$  melhor solução não-tabu na vizinhança de  $x_{cur}$

$x'' \leftarrow$  melhor solução tabu na vizinhança de  $x_{cur}$

**fim\_se**

**se**  $f(x'') < f(x')$  e  $f(x'') < f(x_{opt})$  **então**

**aplicar** *critério de aspiração*:  $x_{cur} \leftarrow x''$ ;  $x_{opt} \leftarrow x''$

**senão**

**se**  $f(x') < f(x_{opt})$  **então**  $x_{opt} \leftarrow x'$

$x_{cur} \leftarrow x'$

**fim\_se**

**atualizar** *lista tabu(memória de curto prazo), memória de longo prazo, ativação diversificação*

**fim\_enquanto**

# Busca Tabu

## ■ Componentes

1. Operador para geração de vizinhança
2. Vizinhança
3. Memória de curto prazo
4. Memória de longo prazo
5. Critério de aspiração
6. Critério de parada



# Busca Tabu

- **Operador para geração de vizinhança** - A ideia central dos métodos de busca é partir de uma solução viável e realizar sobre a mesma uma modificação que resulte em outra solução viável, denominada de movimento.
- Exemplos: remoção & inserção, swap, etc.

# Busca Tabu

- **Vizinhança** - Dado uma solução inicial e um tipo de operador para geração de vizinhança, existem soluções que podem ser atingidas a partir da solução inicial aplicando sobre esta o operador escolhido. Todas estas soluções formam a vizinhança da solução inicial.

# Busca Tabu

- **Memória de Curto Prazo** – Lista de movimentos proibidos (tabus).
- **Memória de Longo Prazo** – Lista de movimentos mais frequentes; ao utilizar o critério de diversificação, movimentos pouco realizados terão maiores chances de ocorrer.

# Busca Tabu

- **Critério de Aspiração** – Uma solução contendo um movimento tabu torna-se corrente apenas se o movimento gerar uma solução melhor que a solução de referência (a melhor solução já obtida).
- **Critério de Parada** – Número de iterações; número de iterações sem melhoria; número de diversificações; tempo de processamento.

# Busca Tabu

## Matriz de Distâncias

	0	1	2	3	4	5
0	0	3	5	6	7	4
1	5	0	5	7	9	5
2	8	4	0	7	2	6
3	5	6	5	0	5	7
4	3	8	11	6	0	3
5	8	4	3	7	4	0

# Busca Tabu

	1a. Iteração					Distância		
<i>Sol Inicial</i>	<b>0</b>	<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>	<b>5</b>	<b>0</b>	31
	0	2	1	3	4	5	0	32
	0	1	3	2	4	5	0	28
	0	1	2	4	3	5	0	31
	0	1	2	3	5	4	0	29

*Melhora FO*

# Busca Tabu

2a. Iteração

Lista Tabu  $\{(2,3)\}$

Distância

*Sol Corrente = Sol Ótima*

*Solução Tabu*

<b>0</b>	<b>1</b>	<b>3</b>	<b>2</b>	<b>4</b>	<b>5</b>	<b>0</b>	28
0	3	1	2	4	5	0	30
0	1	2	3	4	5	0	31
0	1	3	4	2	5	0	40
0	1	3	2	5	4	0	28

*Melhor Solução  
Não Tabu*

# Busca Tabu

## 3a. Iteração

Lista Tabu  $\{(2,3), (4,5)\}$

Distância

<i>Solução Ótima</i>	<b>0</b>	<b>1</b>	<b>3</b>	<b>2</b>	<b>4</b>	<b>5</b>	<b>0</b>	28
<i>Solução Corrente</i>	<b>0</b>	<b>1</b>	<b>3</b>	<b>2</b>	<b>5</b>	<b>4</b>	<b>0</b>	28
	0	3	1	2	5	4	0	30
<i>Solução Tabu</i>	0	1	2	3	5	4	0	29
	0	1	3	5	2	4	0	25
<i>Solução Tabu</i>	0	1	3	2	4	5	0	28

*Melhora FO*



# Busca Tabu

## 4a. Iteração

Lista Tabu  $\{(2,3), (4,5), (2,5)\}$

Distância

<i>Solução Ótima</i>	<b>0</b>	<b>1</b>	<b>3</b>	<b>5</b>	<b>2</b>	<b>4</b>	<b>0</b>	25
<i>Solução Corrente</i>	<b>0</b>	<b>1</b>	<b>3</b>	<b>5</b>	<b>2</b>	<b>4</b>	<b>0</b>	25
	0	3	1	5	2	4	0	25
	0	1	5	3	2	4	0	25
<i>Solução Tabu</i>	0	1	3	2	5	4	0	28
	0	1	3	5	4	2	0	40

*Melhor Solução  
Não Tabu*