

# Relatório de Experimento: PRVC com Simulated Annealing

Avaliação de Funções de Mudança de Vizinhança

Caio Miguel G. Darzi      Felipe dos Santos Moreira  
Vitor de Assis Ramos

6 de dezembro de 2025

## 1 Introdução

Este experimento avalia o desempenho do **Simulated Annealing (SA)** aplicado ao **Problema de Roteamento de Veículos com Capacidade (PRVC)** sob diferentes funções de mudança de vizinhança (movimentos).

### Objetivo do experimento

Queremos entender **como cada tipo de movimento afeta a qualidade final da solução**. Em particular, investigamos:

- **Qualidade final média** obtida por cada movimento;
- **Estabilidade** o quanto houve de variancia nos resultados;
- **Em quais instâncias cada movimento se destaca**;
- **Trade-off** entre explorar soluções novas e refinar soluções existentes.

### Por que a vizinhança importa no SA?

No SA, a busca acontece por meio de “saltos” no espaço de soluções. Esses saltos são definidos pela função de movimento. Assim, **mudar a vizinhança muda completamente a dinâmica do algoritmo**, pois altera:

- **O tamanho do passo** (pequeno → intensificação; grande → diversificação);
- **A parte da solução que é modificada** (intra-rota vs inter-rotas);
- **A chance de escapar de ótimos locais**.

## 2 Movimentos (Vizinhanças) Testados

Neste trabalho, testamos quatro vizinhanças clássicas no PRVC:

## 2.1 relocateMove

Remove um cliente de sua posição atual e o insere em outra posição (na mesma rota ou em rota diferente). É um movimento com bom poder de reorganização entre rotas, ajudando a redistribuir clientes e cargas.

## 2.2 exchangeMove

Troca dois clientes entre posições/rotas. Esse movimento altera duas partes da solução de uma vez, podendo melhorar simultaneamente duas rotas.

## 2.3 orOptMove

Move uma cadeia curta de clientes consecutivos (tipicamente 2–3) para outra posição. É focado em refinar a ordem dentro das rotas, com passos menores.

## 2.4 swap2Opt

Movimento do tipo 2-opt/swap intra-rota, normalmente revertendo segmentos para reduzir cruzamentos. Melhora a geometria interna da rota, mas altera pouco a distribuição global de clientes.

# 3 Metodologia Experimental

## 3.1 Instâncias e repetições

Foram utilizadas 10 instâncias clássicas do conjunto CMT:

CMT1, CMT2, ..., CMT10.

Para cada instância:

- Executamos o SA **30 vezes** (para reduzir variância);
- Repetimos isso para cada uma das **4 vizinhanças**.

Total de execuções:

$$4 \times 10 \times 30 = 1200.$$

## 3.2 Métricas analisadas

Para cada instância e vizinhança registramos:

- **Custo final médio** (qualidade);
- **Desvio padrão** (robustez);
- **Tempo médio** (custo computacional).

## 4 Resultados

### 4.1 Resumo global

Tabela 1: Médias globais considerando todas as instâncias.

Move	Custo médio final	Desvio padrão global	Melhoria média	Tempo médio (s)
exchangeMove	1550.22	682.05	46.40%	0.0496
relocateMove	1558.80	686.85	45.58%	0.0531
orOptMove	2030.93	834.94	29.35%	0.0604
swap2Opt	2144.85	903.86	25.70%	0.0508

Leitura direta:

- **exchangeMove** e **relocateMove** são os melhores em custo final.
- **orOptMove** e **swap2Opt** têm desempenho inferior, apesar de estáveis.
- O tempo médio é semelhante, logo o fator dominante é a vizinhança.

### 4.2 Resultados por instância

Tabela 2: Custo médio final por instância (30 execuções). Menor é melhor.

Instância	exchangeMove	relocateMove	orOptMove	swap2Opt
CMT1	<b>752.94</b>	877.89	917.23	958.73
CMT2	<b>1096.78</b>	1108.85	1703.88	1736.64
CMT3	1332.66	<b>1215.59</b>	1662.73	1772.40
CMT4	1919.49	<b>1898.72</b>	2512.52	2678.67
CMT5	<b>2682.36</b>	2708.09	3355.95	3598.18
CMT6	<b>746.75</b>	837.72	919.15	966.21
CMT7	<b>1087.56</b>	1103.22	1705.70	1731.50
CMT8	1308.09	<b>1194.93</b>	1657.72	1763.57
CMT9	1907.68	<b>1898.30</b>	2519.51	2670.61
CMT10	<b>2667.94</b>	2744.65	3354.94	3571.94

### 4.3 Quem se destaca

- **exchangeMove** é o melhor em 6/10 instâncias (CMT1,2,5,6,7,10).
- **relocateMove** é o melhor em 4/10 instâncias (CMT3,4,8,9).
- **orOptMove** e **swap2Opt** não foram os melhores em nenhuma instância.

## 4.4 Estabilidade (Desvio padrão)

Tabela 3: Desvio padrão do custo final (30 execuções). Menor indica maior robustez.

Instância	exchangeMove	relocateMove	orOptMove	swap2Opt
CMT1	37.78	165.50	<b>7.32</b>	31.58
CMT2	43.62	107.15	<b>5.95</b>	18.52
CMT3	67.59	39.62	<b>24.51</b>	40.36
CMT4	74.79	84.46	<b>24.44</b>	49.14
CMT5	74.62	75.46	<b>37.59</b>	50.32

**Resumo:**

- **orOptMove** é o mais estável em todas as instâncias.
- **exchangeMove** tem boa estabilidade com excelente custo final.
- **relocateMove** tem custo ótimo, mas variância alta em CMT1 e CMT2.
- **swap2Opt** é estável, mas converge para soluções piores.

## 5 Conclusão

Os resultados mostram que a escolha de vizinhança é decisiva no SA para PRVC. As vizinhanças **exchangeMove** e **relocateMove** exploram melhor o espaço de busca, obtendo os menores custos finais. Já **orOptMove** e **swap2Opt** são mais intensificadoras, convergindo de forma estável, porém em ótimos locais de qualidade inferior.

Uma estratégia promissora, sugerida pelos dados, é usar múltiplas vizinhanças no SA:

- **Alta temperatura:** exchangeMove + relocateMove (exploração).
- **Baixa temperatura:** orOptMove + swap2Opt (polimento).