

## 15º Congresso de Inovação, Ciência e Tecnologia do IFSP - 2024

### USO DO ALGORITMO SIMULATED ANNEALING NO PROBLEMA DO CAIXEIRO VIAJANTE

CAUÊ D. M. TEIXEIRA<sup>1</sup>, BRUNO R. GAMINO<sup>2</sup>

<sup>1</sup> Graduando em Bacharel em Engenharia de Controle e Automação, Bolsista PIBIFSP, IFSP, Campus Suzano, caue.daniel@aluno.ifsp.edu.br.

<sup>2</sup> Doutor em Engenharia Elétrica, Professor, IFSP, Campus Suzano, bruno.rafael@ifsp.edu.br.

Área de conhecimento (Tabela CNPq): 3.08.02.02-4 Programação Linear, Não-Linear, Mista e Dinâmica.

**RESUMO:** O Problema do Caixeiro Viajante (PCV) é um desafio clássico de otimização combinatória em que um vendedor ambulante precisa encontrar a rota mais curta para visitar um conjunto de cidades, passando exatamente uma vez em cada uma, e retornar ao ponto de origem. Para resolvê-lo, uma técnica de otimização probabilística inspirada no processo físico de recozimento de materiais foi utilizada, o *Simulated Annealing*. Ele inicia com uma solução inicial e, ao longo das iterações, aceita movimentos que levam a soluções piores com uma probabilidade decrescente baseada na temperatura do sistema. Isso permite ao algoritmo escapar de mínimos locais e explorar diferentes regiões do espaço de busca. A temperatura é gradualmente reduzida ao longo do tempo, reduzindo a probabilidade de aceitar soluções piores. A solução ótima global foi encontrada em todas as instâncias testadas pelo menos uma vez, atendendo ao objetivo proposto.

**PALAVRAS-CHAVE:** *simulated annealing*; problema do caixeiro viajante; otimização combinatória; metaheurística.

### USE OF THE SIMULATED ANNEALING ALGORITHM IN THE TRAVELING SALESMAN PROBLEM

**ABSTRACT:** The Traveling Salesman Problem (TSP) is a classical combinatorial optimization challenge where a traveling salesman must find the shortest route to visit a set of cities, passing exactly once through each, and return to the starting point. To address this, a probabilistic optimization technique inspired by the physical annealing process of materials was employed—*Simulated Annealing*. It begins with an initial solution and, over iterations, accepts moves that lead to worse solutions with a decreasing probability based on the system's temperature. This allows the algorithm to escape local minima and explore different regions of the search space. The temperature gradually decreases over time, reducing the probability of accepting worse solutions. The global optimal solution was found in all tested instances at least once, meeting the proposed objective.

**KEYWORDS:** simulated annealing, traveling salesman problem, combinatorial optimization; metaheuristic.

### INTRODUÇÃO

O Problema do Caixeiro Viajante (*Traveling Salesman Problem* ou TSP, em inglês) é um problema complexo de otimização combinatória, que busca determinar a rota mais curta para visitar um conjunto de cidades, passando apenas uma vez por cada uma e retornando à cidade inicial. O TSP, além de sua relevância teórica, possui aplicações práticas em logística, design de circuitos e outros.

Dada sua natureza NP-difícil, estratégias heurísticas são cruciais para encontrar soluções aproximadas em tempo razoável. Algumas técnicas amplamente empregadas na literatura para a resolução do TSP incluem os Algoritmos Genéticos (Gonçalves; Oliveira; Rocha, 2020), o *Particle Swarm Optimization* (Wang et al., 2003) e outras metaheurísticas. O *Simulated Annealing* é uma técnica inspirada nos processos de recozimento de metais, em que o material é aquecido e resfriado

gradualmente para atingir um estado de baixa energia. Analogamente, ele emprega um processo de busca que permite a aceitação de movimentos que pioram a solução atual, com o objetivo de escapar de mínimos locais e buscar regiões promissoras do espaço de soluções (Välimäki, 2005).

Sua execução é baseada em uma função de custo, que avalia a qualidade de uma solução, e em um parâmetro chamado temperatura, que controla a probabilidade de aceitar uma solução pior. Inicialmente, a temperatura é alta, permitindo uma exploração mais ampla do espaço de soluções e facilitando a superação de obstáculos subótimos. À medida que o algoritmo avança, a temperatura é gradualmente reduzida, o que reduz a aceitação de soluções piores e promove a convergência para uma solução ótima (Kirkpatrick; Gelatt; Vecchi, 1983; Van Laarhoven; Aarts, 2013).

Este trabalho tem como objetivo a utilização do *Simulated Annealing* como forma de solucionar o Problema do Caixeiro Viajante.

## MATERIAL E MÉTODOS

Os dados das instâncias escolhidas para o trabalho foram extraídos de TSPLIB (2018). Para testar o alcance do algoritmo e sua eficácia tanto em instâncias pequenas quanto em grandes, as instâncias “eil51” (426) e “pcb442” (50778) foram escolhidas. Além disso, duas instâncias intermediárias também foram utilizadas, “eil76” (538) e “kroa100” (21282).

Os dados foram estruturados em uma matriz de coordenadas, na primeira linha estão a latitude e longitude da cidade 1 e assim por diante. A distância entre as cidades é calculada de acordo com a distância euclidiana, utilizando o Teorema de Pitágoras. O *software* em que o algoritmo foi desenvolvido e implementado é o GNU Octave, um programa livre utilizado para o cálculo numérico. Na Figura 1 é apresentado o fluxograma que representa a sequência dos passos realizados durante a execução do algoritmo *Simulated Annealing* que foi implementado.

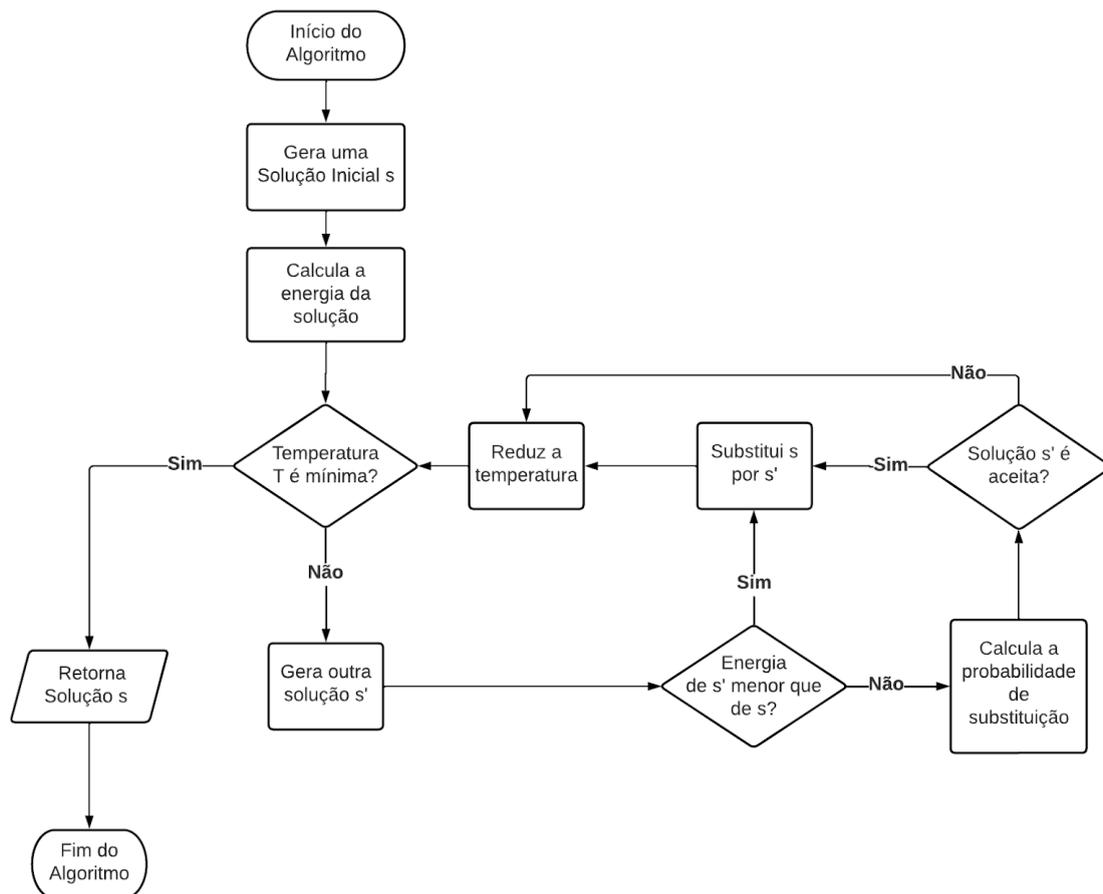


FIGURA 1. Fluxograma do *Simulated Annealing*.

O código inicia, após a determinação dos parâmetros pré-estabelecidos, com a geração aleatória de uma única solução inicial e prossegue com o cálculo da energia consumida por ela. Neste momento, é checado se a temperatura atual é menor ou igual à temperatura mínima determinada pelos parâmetros

iniciais e, caso seja, a solução corrente é elegida como solução final. Se a resposta for negativa, uma nova solução é gerada a partir da solução corrente e é feito o cálculo da probabilidade de substituição da solução corrente pela nova solução, baseado na temperatura atual. Após isso a temperatura atual é diminuída. Este processo é repetido em *looping* até que a solução ótima seja encontrada ou que a temperatura atinja o valor mínimo.

Para a elaboração do algoritmo final, alguns métodos de geração de perturbação na solução atual foram implementados e testados, assim como duas formas diferentes de aceitação da solução vizinha. A primeira se baseia na combinação de um fator de aleatoriedade com a diferença de energia entre a solução vizinha e a solução atual, demonstrada na Equação (1):

$$rand < \exp\left(-\frac{\Delta E}{T}\right). \quad (1)$$

Na Equação (1), “*rand*” representa um número aleatório uniformemente distribuído entre 0 e 1, “*exp(...)*” uma função exponencial, “ $\Delta E$ ” a diferença de energia entre a solução vizinha e a solução atual e “*T*” a temperatura atual.

Se o número aleatório gerado for menor que a exponencial da variação de energia dividida pela temperatura, a solução vizinha é aceita. Isso reflete a natureza probabilística do *Simulated Annealing*, onde a aceitação de soluções subótimas é mais provável em temperaturas mais altas e torna-se menos provável à medida que a temperatura diminui ao longo do tempo (Aarts; Korst, 1989).

A outra abordagem de aceitação da solução vizinha foi baseada na comparação direta entre as energias das soluções, mas usando uma função de probabilidade que não depende da aleatoriedade, vinculado apenas à temperatura atual (Van Laarhoven; Aarts, 2013). A fórmula utilizada é apresentada na Equação (2):

$$\text{Probabilidade de Aceitação} = \frac{1}{1 + \exp\left(\frac{\Delta E}{T}\right)}. \quad (2)$$

Na Equação (2), tem-se que:

1. Se  $\Delta E \leq 0$ , a solução vizinha é sempre aceita;
2. Se  $\Delta E > 0$ , a probabilidade de aceitar a solução pior diminui à medida que a diferença de energia aumenta, mas nunca atinge zero.

Na fase de perturbação do *Simulated Annealing*, a eficácia na exploração do espaço de soluções depende significativamente dos operadores utilizados. Quatro operadores de perturbação distintos foram escolhidos para aprimorar a diversificação e a qualidade das soluções durante a execução do algoritmo. Cada um desempenha um papel específico na busca por soluções de alta qualidade.

O operador de inversão busca introduzir variedade na solução atual ao inverter a ordem de uma parte aleatória do caminho. O processo começa com a escolha aleatória de dois pontos no caminho. Em seguida, a ordem dos nós entre esses pontos é invertida, incluindo os próprios pontos. Finalmente, a solução é atualizada com a nova configuração invertida (Cormen et al., 2009).

O operador de *Swap* consiste na troca de dois vetores da solução. Dois pares de pontos são escolhidos e arranjados de forma que não entrem em conflito uns com os outros, então os vetores que estão entre eles são trocados de lugar. Essa perturbação permite uma rápida mudança da solução atual, facilitando a exploração de vizinhanças e evitando a estagnação em ótimos locais (Kirkpatrick; Gelatt; Vecchi, 1983).

O operador 2-OPT, ou de troca de arestas, foca na melhoria local da solução, escolhendo duas arestas não adjacentes e invertendo a ordem dos vértices entre essas arestas, resultando na formação de um novo circuito. A simplicidade dessa implementação permite a exploração de diversas combinações de trajetórias de forma rápida e eficiente (Van Laarhoven; Aarts, 2013).

O operador de deslocamento, também conhecido como *shift*, implica mover uma subsequência aleatória da solução para uma posição diferente. Inicia-se escolhendo aleatoriamente duas cidades na solução e movendo o vetor que está entre elas para uma posição diferente. A solução é, por fim, atualizada com a nova configuração resultante do deslocamento (Van Laarhoven; Aarts, 2013).

Observa-se nuances distintas em cada uma das abordagens. O operador de inversão destaca-se pela introdução de variabilidade ao inverter aleatoriamente uma parte do caminho, proporcionando uma ampla exploração do espaço de soluções. Em contraste, o operador de *Swap* se destaca por permitir uma mudança abrupta da solução atual, evitando estagnações. O operador 2-OPT, ao focar na

melhoria local pela troca de arestas, destaca-se por sua eficiência em explorar diversas trajetórias. Por fim, o operador de deslocamento, ao mover uma subsequência aleatória, oferece uma estratégia eficaz para diversificar a busca por novas configurações, com mudanças mais leves. Essa diversidade de abordagens demonstra a adaptabilidade do *Simulated Annealing* na otimização do Problema do Caixeiro Viajante, possibilitando a escolha do operador mais adequado em diferentes contextos.

Levando em consideração os testes preliminares, expostos adiante, a combinação utilizada para os testes finais foi: o critério de aceitação vinculado e o operador de perturbação por deslocamento.

Na implementação do algoritmo *Simulated Annealing* realizada, o usuário é livre para definir os parâmetros do algoritmo, como fator de resfriamento, quantidade de iterações e temperatura inicial. Ao final da execução, é retornado o vetor e a função objetivo com a menor distância percorrida. Também é feita uma demonstração gráfica da rota percorrida pelo viajante.

## RESULTADOS E DISCUSSÃO

Para os testes preliminares, foram realizados 300 testes com 2000 iterações para cada operador de perturbação e fórmulas de aceitação de solução vizinha. A Tabela 1 contém a análise estatística dos resultados obtidos com a instância “kroa100”.

TABELA 1. Resultados dos testes preliminares para a instância “kroa100”.

Métodos	Média	Mínimo	Máximo	Amplitude	Desvio Padrão
11	23844	22707	25384	2677	865,37
12	42207,3	37580	49744	12164	3950,52
13	23521,9	22182	24598	2416	954,66
14	23213,5	21908	24070	2162	511,45
21	23276,33	22463	25275	2812	777,66
22	42088,62	39013	44052	5039	3542,3
23	23841,7	22622	25114	2492	949,98
24	22416,66	21681	23037	1356	488,16

Na Tabela 1, a primeira coluna apresenta os modos de aceitação da solução vizinha e os operadores de perturbação, representados pela dezena e pela unidade, respectivamente. O primeiro método de aceitação é o aleatório, e o segundo é o vinculado à temperatura atual. Os operadores são, em ordem, Inversão, *Swap*, 2-OPT e Deslocamento. O valor da função objetivo da solução ótima global encontrada na literatura para a instância “kroa100” é de 21282 (TSPLIB, 2018).

Referindo-se ao método de aceitação, é possível observar um melhor desempenho na média do vinculado em relação a do aleatório. O operador *Swap* apresentou os piores números, tendo resultados com aproximadamente o dobro do valor ótimo global da função objetivo na maioria dos testes realizados, por isso foi descartado. Os operadores por Inversão e 2-OPT obtiveram resultados próximos, mas se demonstraram instáveis em seu desvio padrão.

Por conta destes fatores e por ter apresentado os resultados mais baixos, entre 1,8% e 2,9% acima da solução ótima global e o menor desvio padrão dentre os métodos testados, o operador de Deslocamento se sobressaiu e foi o escolhido para os testes finais.

Com base nisso, os testes finais foram realizados com o método de aceitação vinculado e com o operador de perturbação por Deslocamento. Após efetuar ajustes nos parâmetros do algoritmo implementado, foram realizados 400 testes com 4000 iterações para todas as instâncias. Os resultados são apresentados na Tabela 2.

TABELA 2. Resultados dos testes finais para as instâncias “eil51” (426), “eil76” (538), “kroa100” (21282) e “pcb442” (50778).

Instâncias	Média	Mínimo	Máximo	Amplitude	Desv. Pad.	Taxa Conv.
Eil51	473,16	426	510	84	16,38	77%
Eil76	597,33	538	633	95	47,62	67%
KroA100	22835,75	21282	25094	3812	1252,48	53%
Pcb442	109453,66	50778	128627	77849	39762,71	0,5%

Como visto na Tabela 2, o algoritmo atingiu o valor ótimo global em todos os casos. Para instâncias de até 100 cidades, as taxas de convergência ao ótimo global foram razoáveis e as médias ficaram de 7% a 11% maiores do que o mínimo atingido. Vale ressaltar que o TSP é um problema NP-Difícil com dificuldade fatorial, então quanto maior o número de cidades, maior o número de possibilidades e a dificuldade em resolvê-las. Com isso considerado, o fato do algoritmo ter encontrado o ótimo global na instância “pcb442”, mesmo que em apenas 2 testes, demonstra que o método implementado foi capaz de encontrar soluções de boa qualidade para o TSP.

Demonstrações gráficas também foram emergidas durante o trabalho. Na Figura 2 encontram-se as melhores rotas para todas as instâncias testadas.

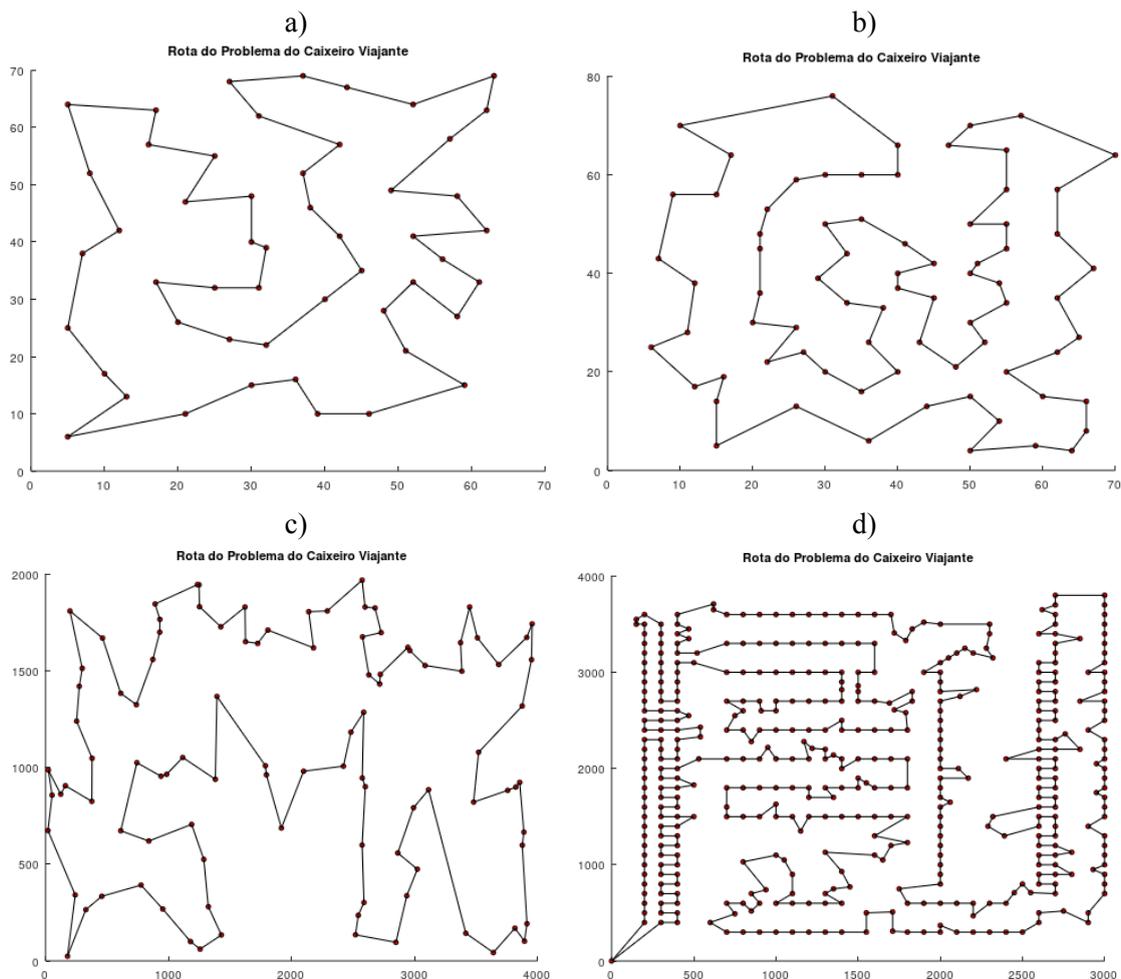


FIGURA 2. Melhores rotas encontradas para as instâncias testadas: a) “eil51”, b) “eil76”, c) “kroa100” e d) “pcb442”.

## CONCLUSÕES

A implementação do *Simulated Annealing* para resolver o Problema do Caixeiro Viajante foi concluída e o objetivo da pesquisa foi atingido, sendo que o valor ótimo global foi encontrado em todas as instâncias testadas, mostrando que o método implementado foi capaz de obter soluções de boa qualidade para o TSP. Para as instâncias menores, as médias e as taxas de convergência foram razoáveis. Para a maior instância, ainda que o método tenha encontrado a solução ótima global nos testes realizados, a média foi alta e os resultados apresentaram grandes variações. Para trabalhos futuros, recomenda-se o enfoque em instâncias com maior quantidade de cidades e/ou trabalhar com instâncias do TSP assimétricas.

## CONTRIBUIÇÕES DOS AUTORES

C.D.M.T contribuiu com a coleta, geração, curadoria e análise dos dados e com a elaboração, codificação e correção do algoritmo. B.R.G contribuiu com os objetivos e a metodologia do trabalho, com a elaboração dos métodos do algoritmo e com a análise dos resultados.

Todos os autores contribuíram com a redação e revisão do trabalho e aprovaram a versão submetida.

## AGRADECIMENTOS

Agradecemos ao Instituto Federal de Educação, Ciência e Tecnologia de São Paulo - Campus Suzano e ao programa PIBIFSP pelo apoio financeiro, tornando possível a realização deste trabalho.

## REFERÊNCIAS

- AARTS, E.; KORST, J. *Simulated Annealing and Boltzmann Machines*. Berlin: Springer, 1989. 285 p.
- CORMEN, T. H.; LEISERSON, C. E.; RIVEST, R. L.; STEIN, C.. **Introduction to Algorithms**. 3 ed. Cambridge, Massachusetts: The MIT Press, 2009. 1312 p.
- GONÇALVES, W.; OLIVEIRA, M. S.; ROCHA, A. R. Algoritmo genético aplicado ao problema de roteamento de veículos: problema do caixeiro viajante no setor varejista. **Cadernos UniFOA**, Volta Redonda, n. 43, p. 11-23, 2020.
- KIRKPATRICK, S.; GELATT, C. D.; VECCHI, M. P. **Optimization by Simulated Annealing**. *Science*, v. 220, n. 4598, p. 671-680, 1983.
- TSPLIB. **University of Heidelberg**, 2018. Disponível em: <<http://comopt.ifl.uni-heidelberg.de/software/TSPLIB95/tsp>>. Acesso em: 12 mai. 2023.
- VÄLIMÄKI, V. Simulated Annealing Algorithms. **IEEE International Symposium on Circuits and Systems**. Kobe: Japan, p. 2685-2688, 2005.
- VAN LAARHOVEN, P. J. M.; AARTS, E. H. L. **Simulated Annealing: Theory and Applications**. Berlin: Springer Science & Business Media, 2013. 187 p.
- WANG, K.; HUANG, L; ZHOU, C; PANG, W. Particle swarm optimization for traveling salesman problem. In: **INTERNATIONAL CONFERENCE ON MACHINE LEARNING AND CYBERNETICS**, 5., 2003, Xi'an. *Proceedings...* Xi'an: ICMLC, 2003. p. 1583-1585.