

Simulated Annealing aplicado às funções DROPWAVE e RASTRIGIN com Ajuste de Parâmetros

Felipe Barbosa Mourão, Vitor Luis Cagneti

¹Universidade da Região de Joinville (UNIVILLE)
Joinville – SC – Brasil

`felipemourao@univille.br, vitor.cagneti@univille.br`

Abstract. *This study explores the use of the Simulated Annealing algorithm for optimizing the benchmark functions DROPWAVE and RASTRIGIN. The goal was to reach the global optimum by fine-tuning algorithm parameters. Experiments were conducted using a modified open-source framework. The results demonstrate the effectiveness of the algorithm in accurately approaching optimal solutions.*

Resumo. *Este artigo apresenta a aplicação do algoritmo Simulated Annealing às funções de benchmark DROPWAVE e RASTRIGIN, com o objetivo de encontrar o ótimo global por meio de ajustes nos parâmetros do algoritmo. Foram realizados testes utilizando o framework disponível em <https://github.com/thekogami/simulated-annealing>, com modificações específicas para melhorar a performance do algoritmo. Os resultados indicam que as funções puderam ser otimizadas com precisão significativa.*

1. Introdução

O Simulated Annealing (SA) é um algoritmo metaheurístico amplamente utilizado para otimização de funções complexas. Este trabalho tem como objetivo aplicar o SA às funções de benchmark DROPWAVE e RASTRIGIN, conhecidas por sua alta complexidade e múltiplos mínimos locais, a fim de validar a eficácia do algoritmo e suas configurações.

2. Metodologia

2.1. Funções de Benchmark

As duas funções utilizadas neste experimento foram:

- **DROPWAVE:** Função bidimensional com mínimo global em (0, 0), onde $f(x) = \frac{1 + \cos(12\sqrt{x_1^2 + x_2^2})}{0.5(x_1^2 + x_2^2) + 2}$.
- **RASTRIGIN:** Função multimodal de n dimensões, definida como $f(x) = 10n + \sum_{i=1}^n (x_i^2 - 10 \cos(2\pi x_i))$, com mínimo global também em (0, ..., 0).

2.2. Configurações do Algoritmo

O algoritmo foi implementado com base no framework disponível em <https://github.com/thekogami/simulated-annealing>. As seguintes modificações foram realizadas:

- Adição das funções `DropWaveFunction` e `RastriginFunction`.
- Ajustes nos parâmetros do algoritmo: taxa de resfriamento, número de iterações por temperatura, total de iterações e tamanho do vetor de solução.

2.3. Parâmetros Utilizados

Para DROPWAVE:

- Operador: `GaussianOperator(0.3)`
- Fator de resfriamento: 0.97
- Iterações por temperatura: 300
- Limite máximo de iterações: 150000
- Dimensão: 2

Para RASTRIGIN:

- Operador: `GaussianOperator(0.3)`
- Fator de resfriamento: 0.96
- Iterações por temperatura: 150
- Limite máximo de iterações: 150000
- Dimensão: 10

Nota: O número total de iterações realizadas pode ser inferior ao limite configurado, dependendo da taxa de resfriamento e dos critérios de parada.

3. Resultados

3.1. Função DROPWAVE

- **Melhor Solução:** $[-1.662 \times 10^{-4}, -2.993 \times 10^{-4}]$
- **Melhor Fitness:** -0.99999575
- **Total de Iterações:** 208200

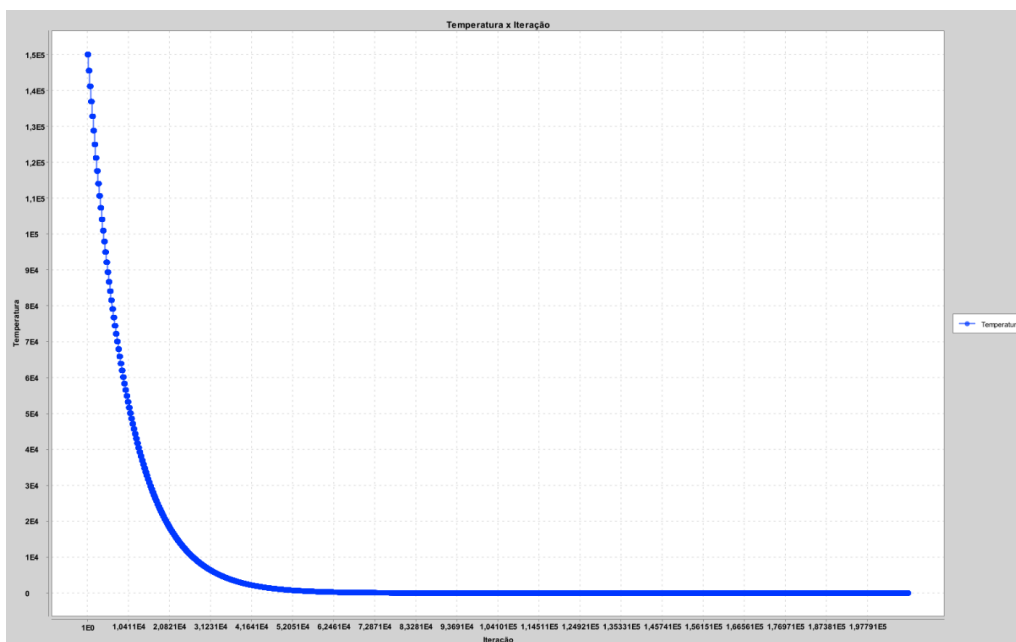


Figure 1. Gráfico de temperatura durante a execução - DROPWAVE

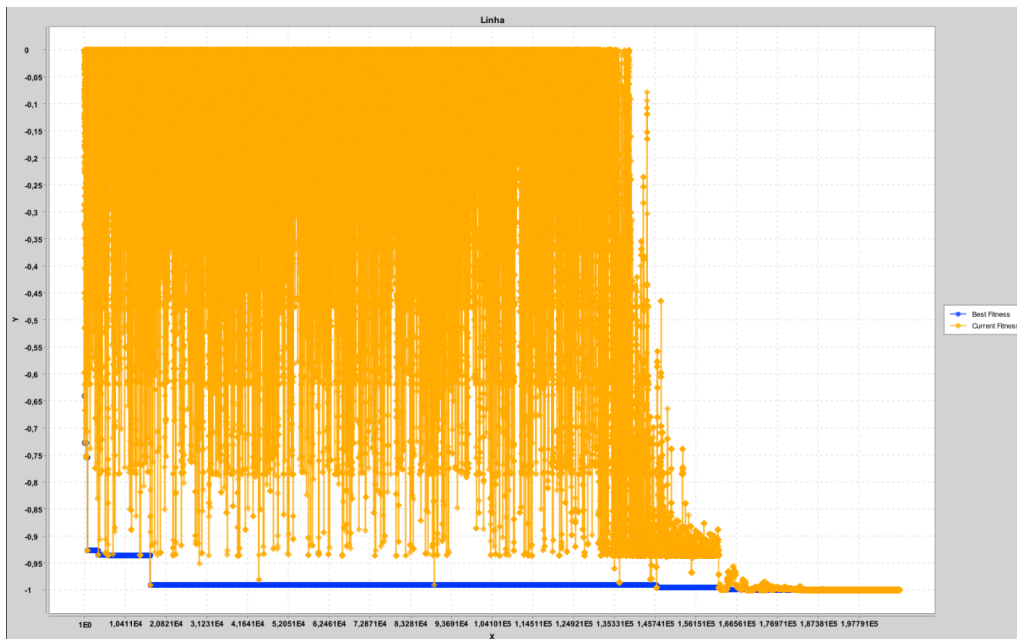


Figure 2. Convergência da melhor solução - DROPWAVE

3.2. Função RASTRIGIN

- **Melhor Solução:** Vetor de 10 dimensões com componentes $x_i \in [-0.01, 0.02]$
- **Melhor Fitness:** 0.0857
- **Total de Iterações:** 77700

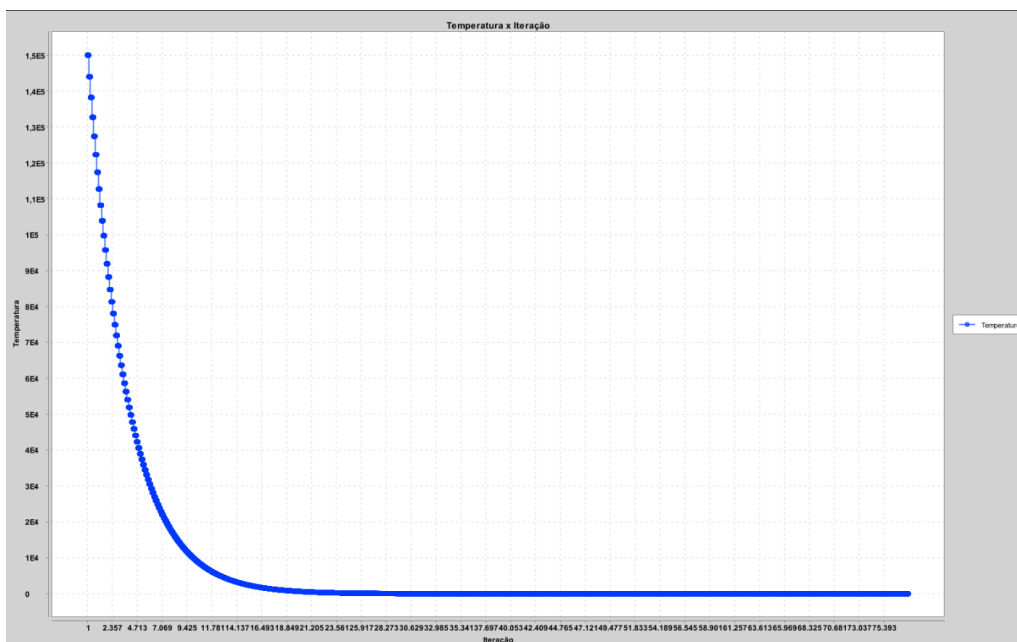


Figure 3. Gráfico de temperatura durante a execução - RASTRIGIN

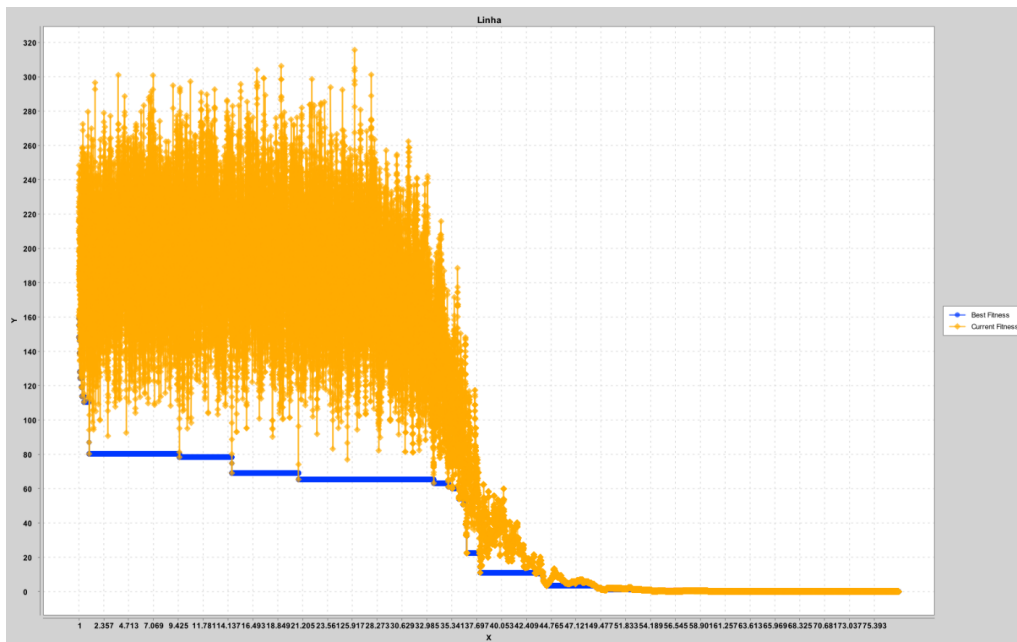


Figure 4. Convergência da melhor solução - RASTRIGIN

4. Conclusão

O algoritmo de Simulated Annealing mostrou-se eficaz na otimização das funções DROP-WAVE e RASTRIGIN. Através do ajuste dos parâmetros como taxa de resfriamento, número de iterações por temperatura e o tamanho do vetor de entrada, foi possível alcançar soluções próximas ao ótimo global, com baixo custo computacional.

As modificações simples no código e a adaptação de funções permitiram estender a aplicabilidade do framework original, mantendo sua estrutura modular e clara.

Referências

- Surjanovic, S. & Bingham, D. Virtual Library of Simulation Experiments: Test Functions and Datasets. <https://www.sfu.ca/~ssurjano/optimization.html>
- Kirkpatrick, S., Gelatt, C. D., & Vecchi, M. P. (1983). Optimization by Simulated Annealing. *Science*, 220(4598), 671–680.
- Projeto base: <https://github.com/thekogami/simulated-annealing>