

# Estrutura de Dados II: Simulated Annealing

Vitor Luis Cagneti, Felipe Barbosa Mourão

Bacharelado de Sistemas de Informação  
Universidade da Região de Joinville (UNIVILLE) – Joinville, SC – Brazil

@vitor.cagneti@univille.br, @felipe.mourão@univille.br

**Resumo.** Este trabalho apresenta a aplicação do algoritmo Simulated Annealing (SA) na otimização de duas funções de benchmark: **Drop-Wave** e **Rastrigin**. O objetivo é analisar a capacidade do SA de escapar de mínimos locais e encontrar o ótimo global em superfícies altamente multimodais. O projeto foi desenvolvido em Java, com geração automática de gráficos de temperatura e convergência. As funções foram integradas à arquitetura modular do sistema, e o algoritmo foi ajustado para melhor desempenho. Os resultados demonstram a eficácia do SA em ambientes com alta complexidade de busca.

## 1. Introdução

O problema de otimização global está presente em diversas áreas como ciência de dados, engenharia e inteligência artificial. Algoritmos bioinspirados como o Simulated Annealing destacam-se por sua habilidade em explorar soluções fora do espaço local, utilizando estratégias de resfriamento probabilístico. Neste trabalho, utilizamos o SA para otimizar as funções Drop-Wave e Rastrigin, realizando tuning nos parâmetros e registrando os resultados por meio de gráficos de desempenho.

## 2. Simulated Annealing

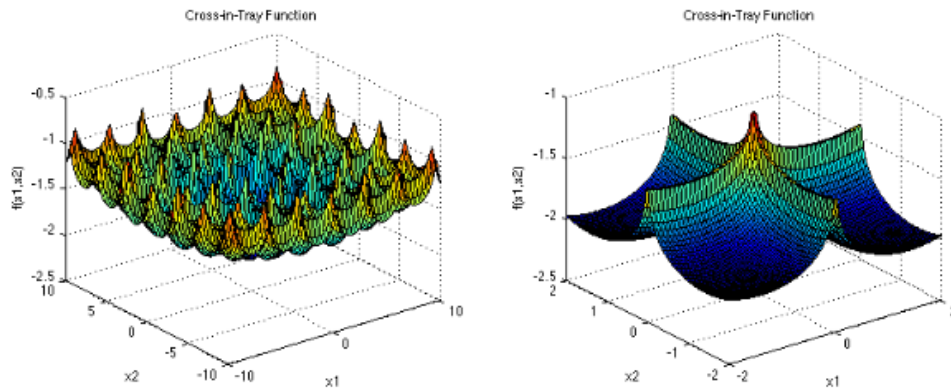
O SA é inspirado no processo de recozimento térmico da metalurgia, onde materiais são aquecidos e resfriados lentamente para minimizar a energia interna. A analogia computacional permite aceitar soluções piores com certa probabilidade, ajudando a escapar de mínimos locais.

A probabilidade de aceitação é controlada por uma função baseada na temperatura:

$$P(\Delta E, T) = \exp\left(-\frac{\Delta E}{T}\right)$$

### 3. Funções Benchmark

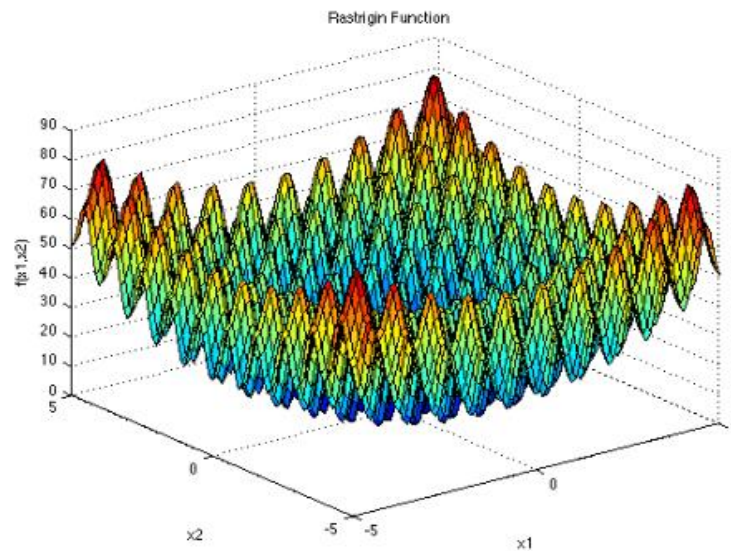
#### a. Cross-in-Tray Function



$$f(\mathbf{x}) = -0.0001 \left( \left| \sin(x_1) \sin(x_2) \exp \left( \left| 100 - \frac{\sqrt{x_1^2 + x_2^2}}{\pi} \right| \right) \right| + 1 \right)^{0.1}$$

Múltiplos ótimos globais, com mínimo de aproximadamente -2.06261.

#### b. Rastrigin Function



$$f(\mathbf{x}) = 10d + \sum_{i=1}^d [x_i^2 - 10 \cos(2\pi x_i)]$$

Valor mínimo global em  $\mathbf{x}=\mathbf{0}$

Valor mínimo teórico: 0.0

Apresenta diversos ótimos locais distribuídos simetricamente.

#### 4. Metodologias Utilizada

- **Linguagem:** Java
- **Projeto:** Modular com interface Function
- **Biblioteca Gráfica:** XChart
- **Funções:** Drop-Wave e Rastrigin implementadas em function/
- **Escolha de função** via FunctionFactory
- **Perturbação:** GaussianOperator com passo de 0.3

#### 5. Parâmetros do SA

Os experimentos foram realizados com os seguintes parâmetros:

- INITIAL\_TEMPERATURE = 1000
- FINAL\_TEMPERATURE = 0.01
- ALPHA = 0.97
- MAX\_ITERATIONS\_PER\_TEMP = 300
- TOTAL\_ITERATIONS = 150000
- PERTURBATION\_RANGE = 0.3

A temperatura é reduzida a cada 500 iterações globais. Adotado mecanismo de resfriamento em degraus, para estabilidade.

#### 6. Modificações Feitas

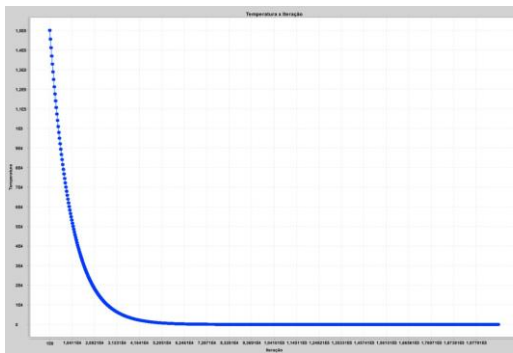
- Adição de novas funções no diretório function/
- Registro das funções no FunctionFactory.java
- Geração automática de gráficos com os logs de iteração e temperatura

#### 7. Resultados

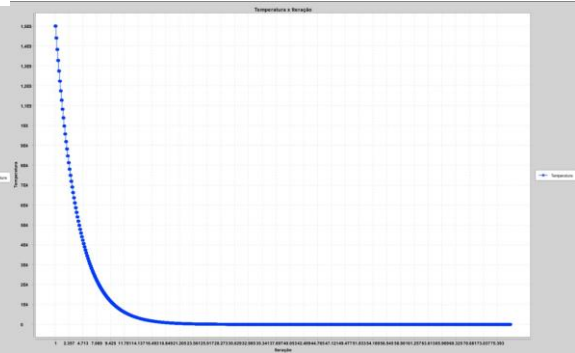
Gráfico de Temperatura: A temperatura apresenta uma queda com pausas regulares, permitindo maior exploração em cada nível térmico.

Gráfico de Convergência: Mostra a evolução da melhor e da solução atual ao longo do tempo. Na Drop-Wave, há convergência estável para valores próximos a -1. Na Rastrigin, o algoritmo oscilou até encontrar o mínimo global (0).

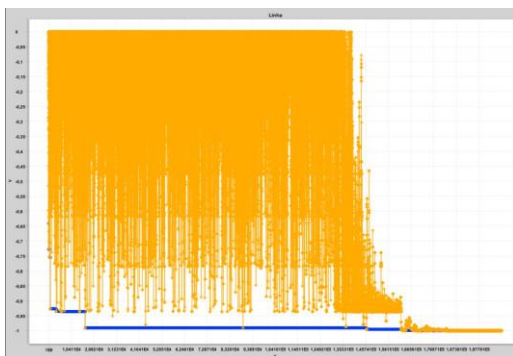
DROPWAVE



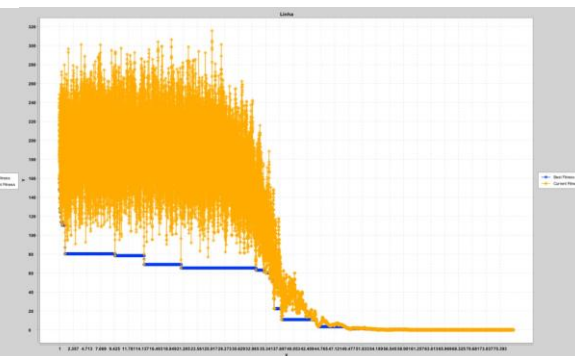
RASTRIGIN



DROPWAVE



RASTRIGIN



## 8. Conclusão

O Simulated Annealing mostrou-se eficiente para funções complexas, especialmente com o uso de estratégias de resfriamento controlado. A função Drop-Wave apresentou convergência rápida e suave, enquanto a Rastrigin exigiu maior exploração por conta de seus múltiplos ótimos locais. A modificação do controle de temperatura contribuiu para uma busca mais robusta e estável.

## 9. Referencias

Kirkpatrick, S., Gelatt, C. D., & Vecchi, M. P. (1983). Optimization by Simulated Annealing. Science.

Surjanovic, S., & Bingham, D. (2013). Virtual Library of Simulation Experiments: Test Functions and Datasets. <https://www.sfu.ca/~ssurjano/optimization.html>

Knowm XChart Library: <https://knowm.org/open-source/xchart/>