

# Simulated Annealing

## Uma resolução para SAT-3

César Eduardo de Souza<sup>1</sup>,  
Guilherme Diel<sup>1</sup>

<sup>1</sup>Departamento de Ciência da Computação  
Universidade do Estado de Santa Catarina (UDESC) – Joinville, SC – Brazil

{cesar.souza, guilherme.diel}@edu.udesc.br

**Resumo.** Algoritmos heurísticos se alicerçam em técnicas de busca de solução satisfatória. Um deles é o **Simulated Annealing**, que se baseia no processo de resfriamento de metais. Este algoritmo é capaz de resolver problemas NP-Hard e NP-Completo, como o problema SAT-3, que consiste em determinar se existe uma atribuição de valores lógicos que satisfaça uma expressão booleana. Neste trabalho, apresentamos uma implementação do **Simulated Annealing** para resolver o problema SAT-3, com resultados satisfatórios e consistentes, principalmente para entradas maiores. Além disso, discutimos possíveis aplicações futuras e comparações com outras abordagens.

### 1. Introdução

Foi no século XX que se iniciou a busca pela resolução de problemas NP, NP-Hard e NP-Completo usando ferramentas computacionais. Estas, portanto, foram, em sua maioria, desenvolvidos com base em algoritmos heurísticos – alicerçados em técnicas de busca de solução não necessariamente ótimas, mas sim satisfatórias. – Sendo assim, alguns destes tornaram-se mais disseminados na literatura científica, como a **Busca Gulosa**, **Algoritmo A\***, **Subida de Encosta** e **Simulated Annealing** [Kokash nd].

O método de **Simulated Annealing** teve sua lógica concebida a partir do método de anelização de materiais, Metropolis, desenvolvido por Gibbs em 1953 [Metropolis et al. 1953]. Baseado na fabricação de anéis, este método usa como base o fato de que, quanto mais quente está o material, maior se torna a facilidade de moldá-lo. Sob o mesmo ponto de vista, este método consiste em uma sequência de temperaturas decrescentes em que, quanto maior a temperatura atual, mais aleatorizadas são as otimizações geradas pelo algoritmo, sendo que, quando a temperatura chegar a um certo ponto, idealizando a otimização do resultado conforme decresce a temperatura, até que esta se torne mínima [Bertsimas and Tsitsiklis 1993].

Um problema muito conhecido e discutido na literatura, que é capaz de ser resolvido por algoritmos heurísticos, é o problema da satisfabilidade (SAT), que consiste em, dado um conjunto de cláusulas disjuntivas na forma normal conjuntiva, determinar se existe uma atribuição de valores lógicos (*verdadeiro* ou *falso*) às variáveis envolvidas que satisfaça toda a expressão. Cada cláusula é composta por uma disjunção (operador lógico  $\vee$ ) de literais (variáveis ou suas negações), e a fórmula booleana global é uma conjunção (operador lógico  $\wedge$ ) dessas cláusulas. Formalmente:

$$(A \vee B \vee C \vee \dots) \wedge (D \vee E \vee F \vee \dots) \wedge \dots \quad (1)$$

Ao longo deste relatório, será abordada uma proposta de implementação de do **Simulated Annealing** para resolução de uma variante do SAT, conhecido como SAT-3, que consiste em três variáveis por cláusula no problema SAT.

Este relatório está organizado da seguinte maneira: a seção 2 apresenta estratégias utilizadas, descrições, justificativas de escolhas, fórmulas utilizadas e descrições. Em seguida, na seção 3 são abordadas descrições dos experimentos, configurações utilizadas e descrições dos resultados obtidos. Outrossim, na seção 4 expõem-se considerações sobre os resultados obtidos e análises críticas sobre os mesmos. Por fim, na seção 5 mostram-se considerações sobre o trabalho desenvolvido e identificação de direcionamentos futuros na pesquisa.

## 2. Metodologia de Desenvolvimento

O método de **Simulated Annealing** consiste em:

1. Para uma temperatura  $T_i$ , realizar  $N$  vezes:
  - Realizar uma perturbação aleatória no estado atual  $estado_0$ , gerando um novo  $estado_i$
  - Calcular a variação de energia  $\Delta E = E(estado_i) - E(estado_0)$
  - Se  $\Delta E < 0$  (melhora na energia), aceitar a transição ( $estado_i \rightarrow estado_0$ )
  - Caso contrário, aceitar a transição com probabilidade  $P_{accept}(T_i)$
2. Critério de parada:
  - Se  $T_i \leq T_f$  (temperatura final) ou o sistema atingir convergência (e.g.,  $\Delta E \approx 0$  por  $k$  iterações consecutivas)
  - Retornar o  $estado_0$  como solução
  - Caso contrário, reduzir a temperatura ( $T_i \leftarrow \alpha T_i$ , com  $0 < \alpha < 1$ ) e retornar ao Passo 1

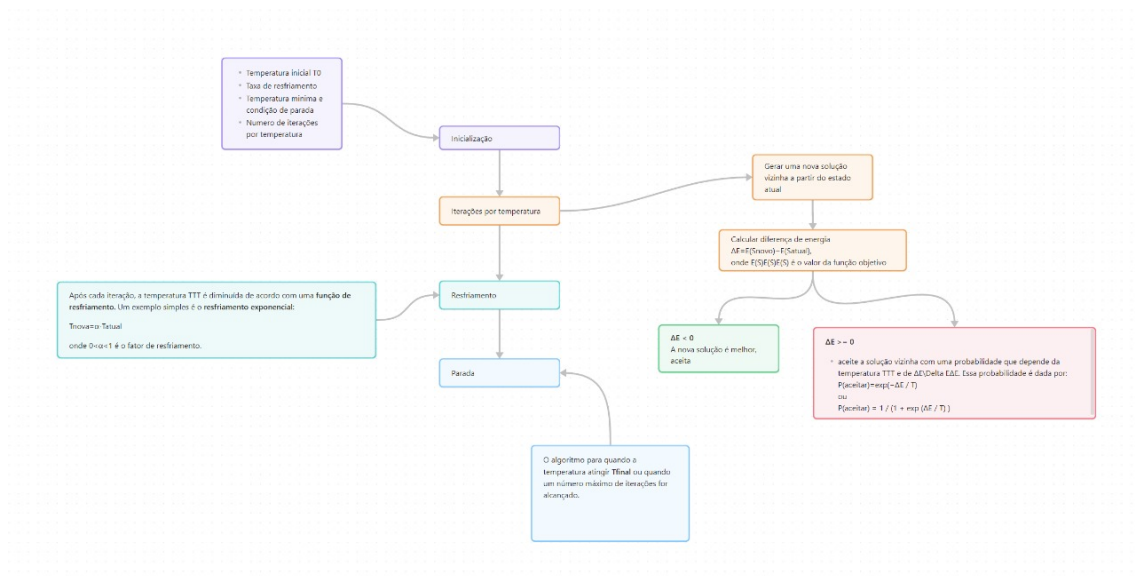
A aplicação do método de **Simulated Annealing** para a otimização do problema do SAT-3 foi realizada por meio da linguagem de programação *Python*, junto com a biblioteca *Numpy*. A Figura 7 retrata o passo a passo de como foi implementado este algoritmo. O passo 1 (inicialização) consiste em realizar a inicialização do sistema:

- Temperatura inicial  $T_0 = 1000$
- Taxa de resfriamento  $\alpha = 0.99$   $Temperatura\ final T_f = 0.1$
- Número de iterações por temperatura  $N = 1000$

No segundo passo (iteração por temperatura) da Figura 7 a fórmula de probabilidade de aceitação de estados com pior energia que foi adotada, foi:

$$P_{accept} = \exp\left(-\frac{\Delta E}{T_i}\right) \quad (2)$$

O terceiro passo (resfriamento) corresponde ao passo 1 do código de 2.

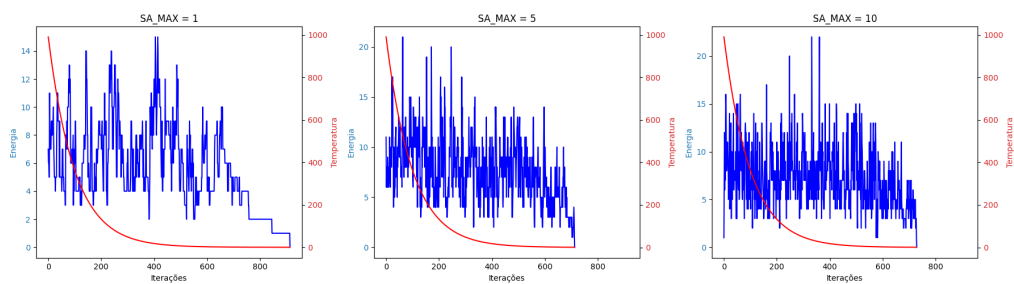


**Figura 1. Diagrama do algoritmo de Simulated Annealing**

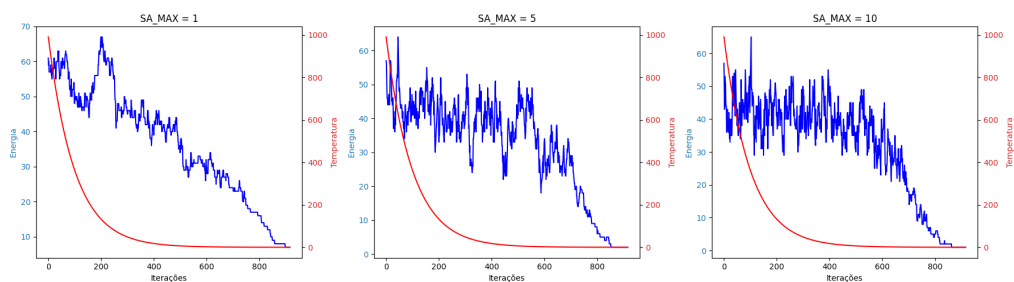
### 3. Descrição de Experimentos/Simulações e Resultados Obtidos

Foi com a temperatura inicial = 1000 = iterações por temperatura taxa de resfriamento = 0.99

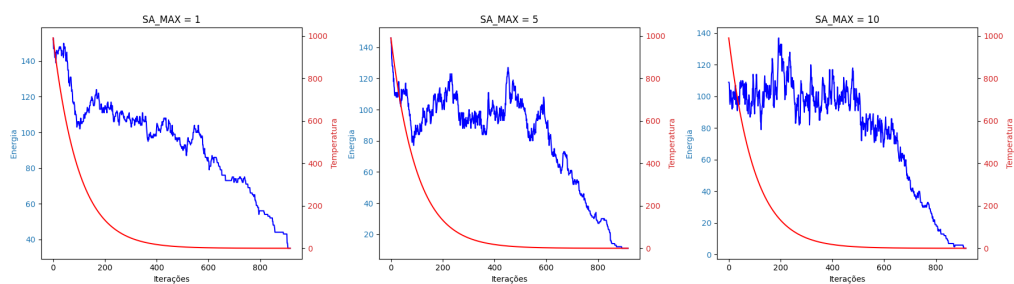
Nestas configurações foram obtidos resultados para bases de SAT-3 de 20, 100 e 250 entradas como os seguintes gráficos de convergência:



**Figura 2. Convergência para 20 entradas**



**Figura 3. Convergência para 100 entradas**



**Figura 4. Convergência para 250 entradas**

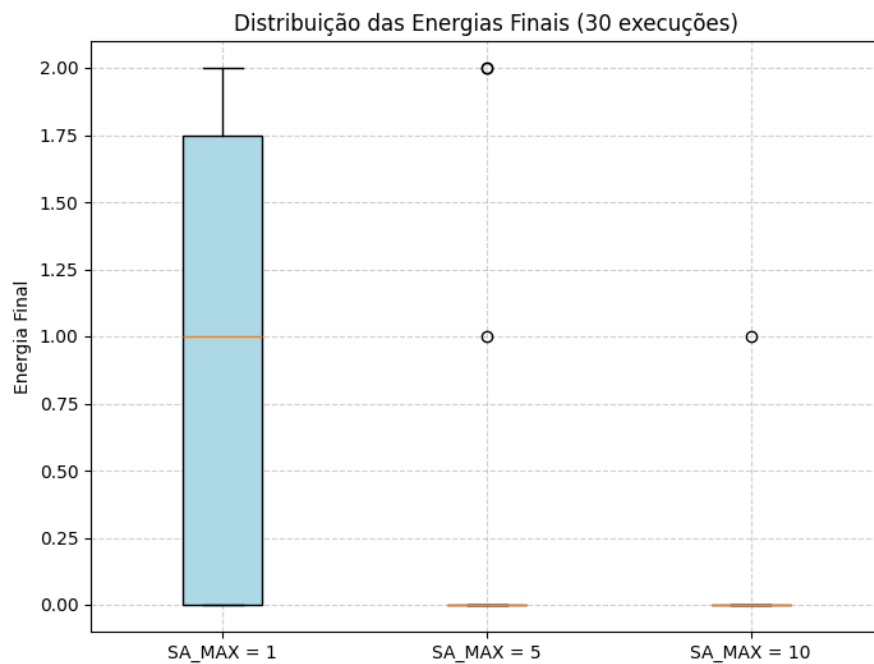
Além disso, é possível verificar a seguinte tabela com média e desvio padrão de 30 execuções do experimento, apontados pelos consecutivos boxplots.

**Tabela 1. Média e Desvio Padrão dos Resultados Obtidos**

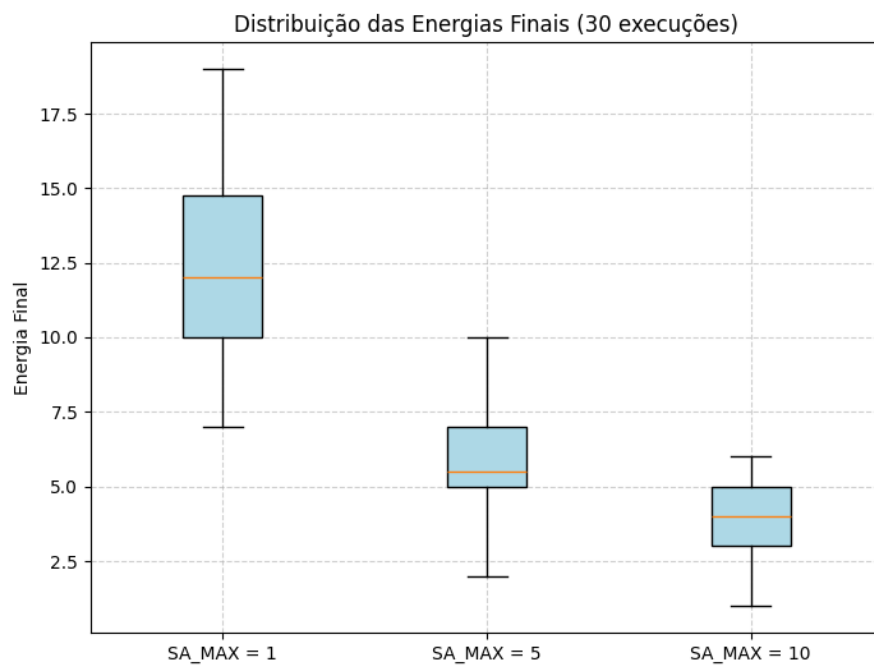
SAMAX	Número de Entradas	Média	Desvio Padrão
1	20	0.80	0.85
5	20	0.17	0.53
10	20	0.03	0.18
1	100	12.43	3.09
5	100	5.67	1.56
10	100	3.93	1.48
1	250	43.53	6.25
5	250	16.10	3.29
10	250	9.93	2.48

**Tabela 2. Média e Desvio Padrão do Histórico dos Resultados Obtidos**

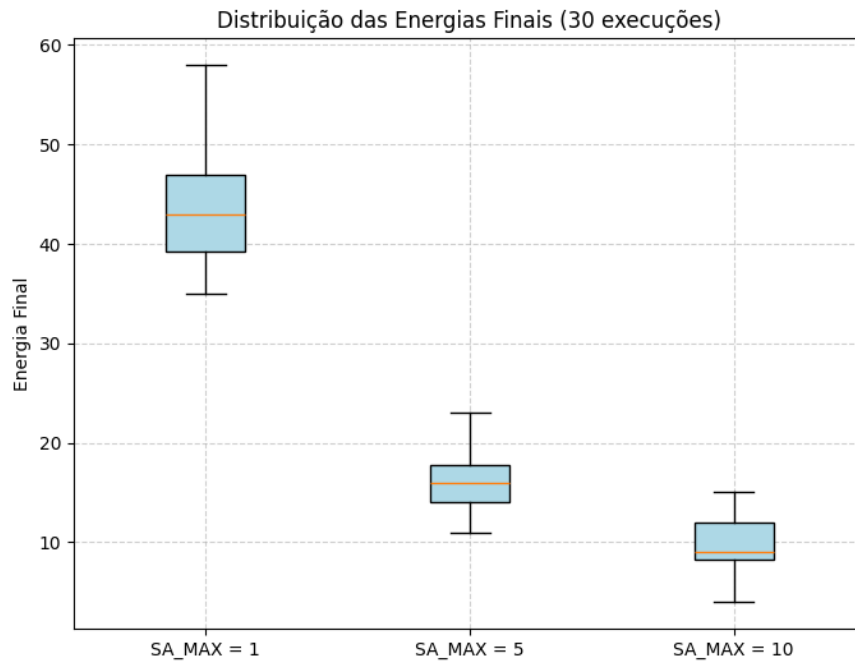
SAMAX	Número de Entradas	Média	Desvio Padrão
1	20	6.50	3.47
5	20	7.18	3.39
10	20	7.34	3.22
1	100	35.14	11.74
5	100	31.41	13.39
10	100	30.70	14.16
1	250	90.75	23.46
5	250	78.44	30.06
10	250	75.82	32.76



**Figura 5. Boxplots para 20 entradas**



**Figura 6. Boxplots para 100 entradas**



**Figura 7. Boxplots para 250 entradas**

Dessa maneira, é possível obter uma visão aprofundada da execução do algoritmo, discutida na seção a seguir.

#### 4. Análise dos resultados obtidos.

Factualmente, torna-se óbvia a observação de que entradas menores produzem resultados limitados, assemelhando-se mais a buscas aleatórias — principalmente com SA\_MAX em 10 — enquanto entradas maiores produzem resultados mais satisfatórios, com uma convergência mais acentuada.

Por outro lado, a convergência para 100 entradas, com SA\_MAX, traz resultados mais satisfatórios, com uma média de 3.93 e desvio padrão de 1.48, enquanto para 250 entradas a média é de 9.93 e desvio padrão de 2.48, como relatado na Tabela 2. Além disso, pode-se observar com os boxplots que, com o aumento do número de entradas, os resultados tornam-se mais homogêneos, relatável pelo desvio padrão, que diminui conforme o número de entradas aumenta. Em suma, os resultados obtidos demonstram que o algoritmo de **Simulated Annealing** é capaz de resolver o problema SAT-3, com uma convergência satisfatória e resultados consistentes, principalmente para grandes entradas, tornando possível quantizar a eficiência do teorema de Gibbs, proposto em 1953.

A partir disso, surgem diversas ideias sobre o trabalho desenvolvido e direcionamentos futuros, abordados na seção a seguir.

#### 5. Conclusões e Trabalhos Futuros

Tendo em vista o trabalho desenvolvido, é possível concluir que o algoritmo de **Simulated Annealing** é uma ferramenta poderosa para a resolução do problema SAT-3, apresentando

resultados satisfatórios e consistentes, principalmente para entradas maiores.

De tal maneira, é possível identificar outras aplicações para a pesquisa em trabalhos futuros, como a aplicação do algoritmo em outros problemas NP-Hard e NP-Completo, como o **Problema do Caixeiro Viajante** ou o **Problema de Otimização de Roteamento**, pode trazer novas perspectivas e resultados interessantes.

Além disso, surge o interesse em comparar esta com outras abordagens como a **Escalada de Encosta** e **Busca Tabu**, para verificar a eficiência e eficácia de cada uma delas em diferentes contextos e problemas. Dessa forma, tais comparações podem contribuir para o avanço do conhecimento na área de algoritmos heurísticos e otimização por meio de inteligência artificial, podendo usar diversas técnicas de análise de desempenho e estatística.

Por fim, a implementação do algoritmo em outras linguagens de programação, como *C++* ou *Java*, pode trazer novas perspectivas e resultados interessantes, além de possibilitar a comparação entre diferentes implementações e otimizações, podendo usar a tal para comparar a eficiência e eficácia de cada uma delas em diferentes contextos e problemas.

## Referências

- Bertsimas, D. and Tsitsiklis, J. (1993). Simulated annealing. *Statistical Science*, 8(1):10–15.
- Kokash, N. (n.d.). An introduction to heuristic algorithms. Department of Informatics and Telecommunications, University of Trento, Italy. [kokash@dit.unitn.it](mailto:kokash@dit.unitn.it).
- Metropolis, N., Rosenbluth, A. W., Rosenbluth, M. N., Teller, A. H., and Teller, E. (1953). Equation of state calculations by fast computing machines. *The journal of chemical physics*, 21(6):1087–1092.