

Ant Colony Optimization

Uma abordagem em paralelo

César Eduardo de Souza¹,
Guilherme Diel¹

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

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

Resumo.

1. Introdução

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 1 retrata o passo a passo de como foi implementado este algoritmo.

A Figura 1 apresenta as etapas do algoritmo, que foi implementado da seguinte maneira:

1. Inicialização:

- Temperatura inicial $T_0 = 1000$
- Taxa de resfriamento $\alpha = 0.99$
- Temperatura final $T_f = 0.1$
- Número de iterações por temperatura $N = 1000$
- $SA_{max} = 1, 5$ ou 10 (limite mínimo de temperatura)

2. Iteração por temperatura:

- (a) Gerar nova solução aleatória (modificando uma variável)
- (b) Calcular $\delta E = E_{nova} - E_{atual}$
- (c) Se $\delta E < 0$, aceitar a nova solução

(d) Caso contrário, aceitar com probabilidade:

$$P_{accept} = \exp\left(-\frac{\Delta E}{T_i}\right) \quad (1)$$

3. Resfriamento:

- Atualizar temperatura: $T_{i+1} = \alpha T_i$ eq:resfriamento
- Verificar critério de parada:
 - Se $T_i \leq SA_{max} \rightarrow$ Parar
 - Caso contrário \rightarrow Voltar ao passo 2

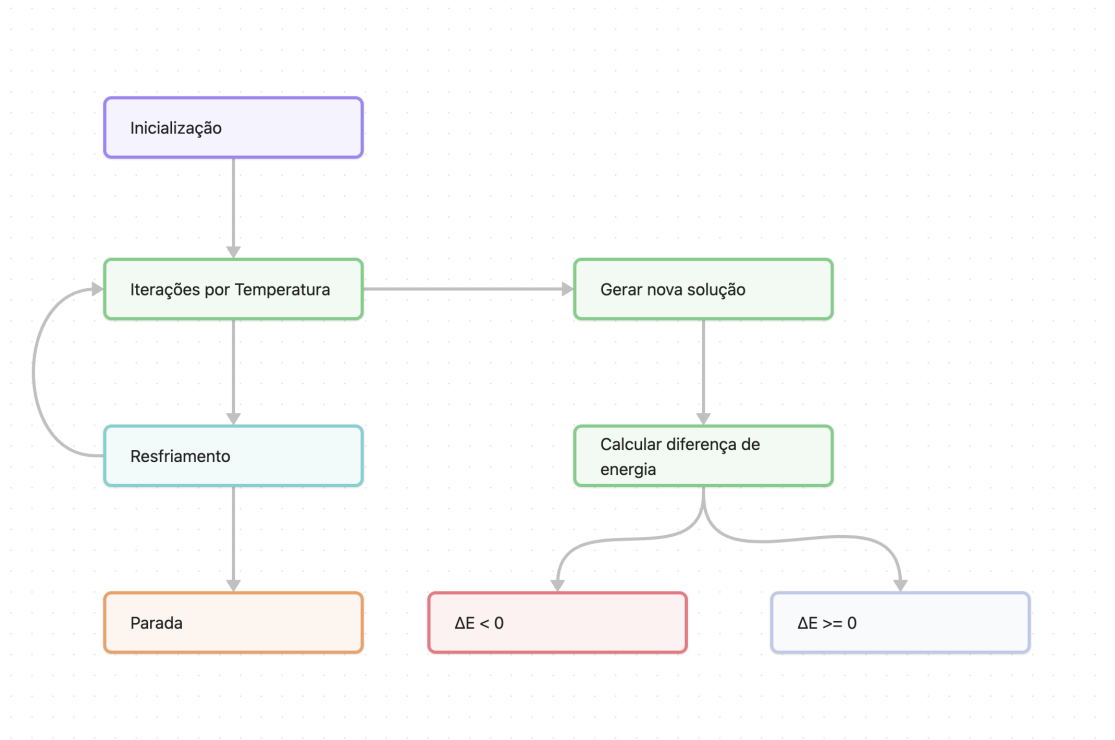


Figura 1. Diagrama do algoritmo de Simulated Annealing

3. Descrição de Experimentos/Simulações e Resultados Obtidos

Foi com a temperatura inicial = 1000 = iterações por temperatura taa 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:

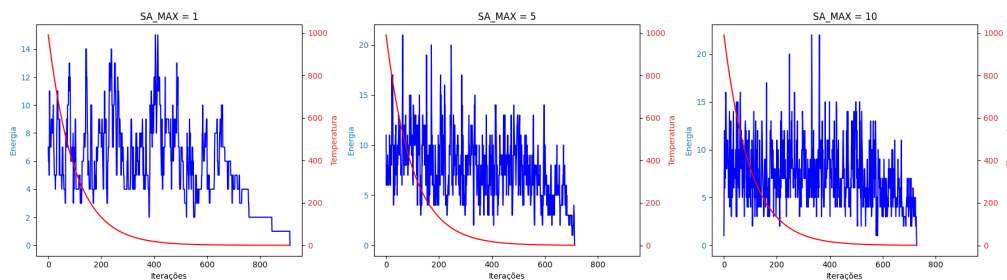


Figura 2. Convergência para 20 entradas

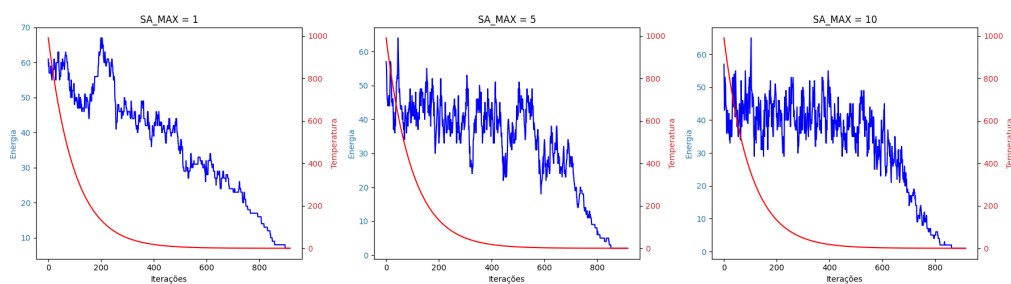


Figura 3. Convergência para 100 entradas

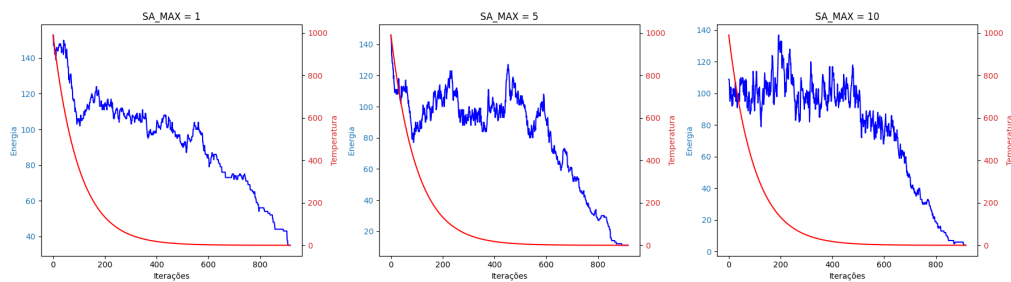


Figura 4. Convergência para 250 entradas

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

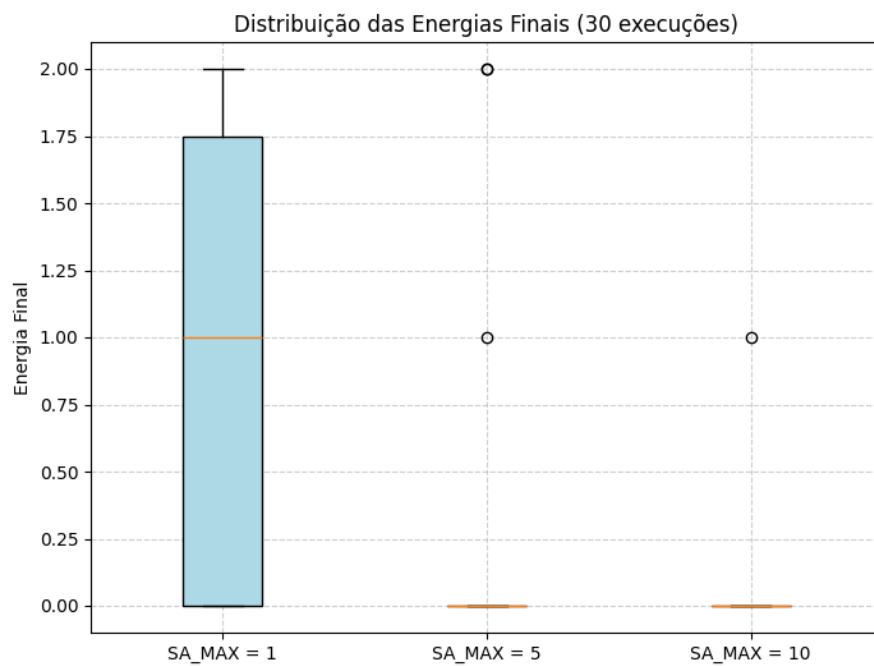


Figura 5. Boxplots para 20 entradas

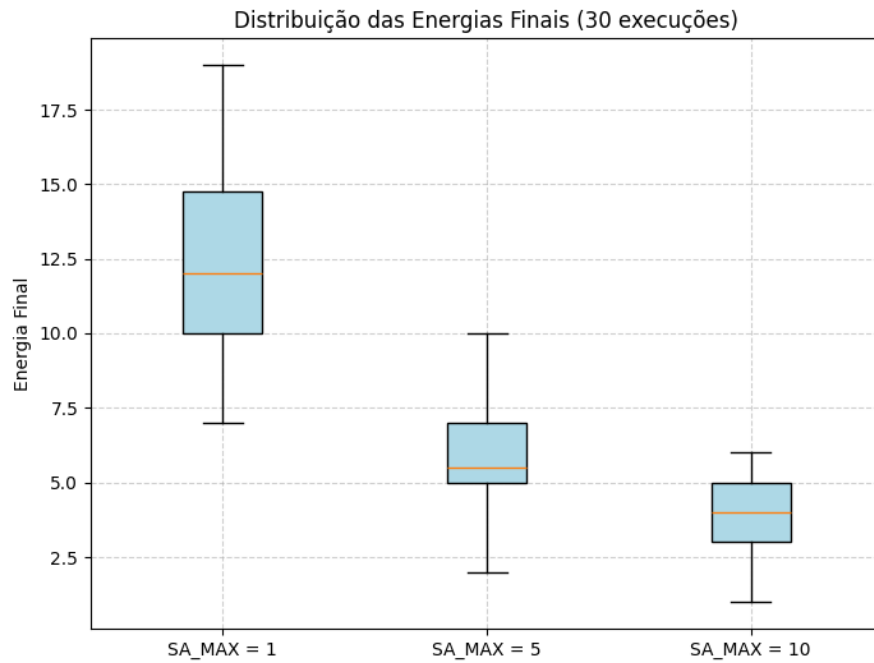


Figura 6. Boxplots para 100 entradas

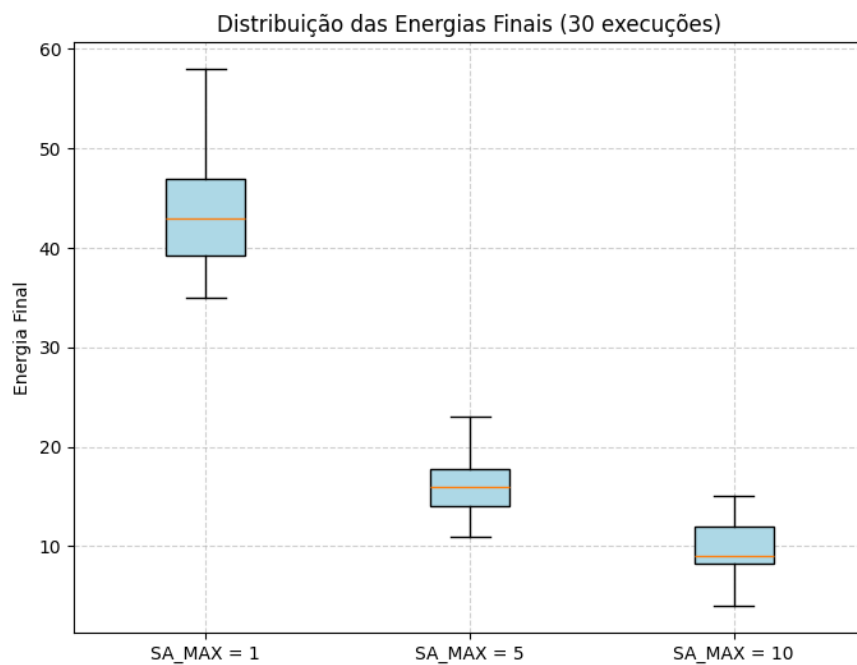


Figura 7. Boxplots para 250 entradas

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.

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.