Meta-Heurística Simulated Annealing para resolução do problema 3SAT

Victor Eduardo Requia

[1]Universidade do Estado de Santa Catarina – UDESC
Joinville – Brasil

victorrequia@gmail.com

Resumo. *O problema 3SAT, uma variação do conhecido problema NP-completo SAT, desafia pesquisadores devido à sua complexidade. Enquanto algoritmos exatos têm limitações práticas para grandes instâncias, meta-heurísticas surgem como alternativas viáveis. Neste contexto, este trabalho explora o uso do Simulated Annealing (SA) para resolver o 3SAT, destacando sua eficácia e os desafios na calibração de parâmetros.*

1. Introdução

O problema da satisfatibilidade (SAT) foi o primeiro problema identificado como pertencente à classe de complexidade NP-Completo [Araújo 2023]. Refere-se à tarefa de encontrar uma atribuição de verdade que torne uma expressão booleana arbitrária verdadeira [Spears 1993]. Este problema, possui aplicações práticas relevantes, como verificação de consistência em bases de conhecimento de sistemas especialistas [Nguyen et al. 1985] e síntese de circuitos assíncronos [Gu and Puri 1995].

Para resolver o problema SAT, existem diversos algoritmos, cada um com sua característica, tendo vantagens e desvantagens. Esses métodos de resolução, podem ser aproximadamente classificados em duas categorias: métodos completos e incompletos [Marchiori and Rossi 1999]. Exemplos eficientes da primeira categoria incluem as abordagens baseadas no algoritmo Davis-Putnam [Gu 1997]. Métodos incompletos incluem abordagens baseadas em busca local, bem como abordagens baseadas em computação evolutiva [Back 1998] . Algoritmos genéticos para SAT empregam informações heurísticas na função de aptidão e/ou nos operadores GA (seleção, cruzamento e mutação).

2. Objetivo

Este trabalho, busca uma solução com tempo viável para o problema de satisfatibilidade booleana, mais especificamente uma extensão do SAT, o 3-satisfatibilidade ou, 3SAT. Neste problemas, as fórmulas serão uma conjunção (AND) de cláusulas. Esta é chamada forma normal conjuntiva (FNC). Determinar se uma fórmula nesta forma é satisfazível é ainda um problema NP-completo, onde cada cláusula é limitada a três literais. Isto quer dizer que a expressão tem a forma:

$$(x_1 \vee x_2 \vee \neg x_3) \wedge (\neg x_1 \vee x_3 \vee x_4) \wedge (x_2 \vee \neg x_4 \vee x_5)... \tag{1}$$

Para o objetivo de resolução em tempo viável, foi utilizada a meta-heurística Simulated Annealing (SA). Meta-heurística é uma heurística que pode oferecer uma solução

suficientemente boa para um problema de otimização, espacialmente quando este, tem informações incompletas ou imperfeitas ou até mesmo capacidade computacional limitada.

Meta-heurística parte do princípio que o conjunto possível de soluções seja muito grande para ser completamente enumerado ou explorado de outra forma. Normalmente, as meta-heurísticas são aplicadas a problemas aos quais não se conhece solução algorítmica eficiente, como no problema de satisfatibilidade booleana, problema da mochila, caixeiro viajante, ciclo hamiltoniano entre outros problemas NP-Completos.

A meta-heurística SA, é uma técnica de otimização que se inspira no processo de resfriamento de materiais para encontrar soluções de qualidade em problemas complexos. Neste contexto, o uso do Simulated Annealing permite explorar eficientemente o espaço de busca de soluções para o 3SAT, equilibrando a exploração local e global, a fim de encontrar soluções satisfatórias em um tempo razoável. A principal característica do SA é que ele pode evitar cair em ótimos locais, uma vez que os critérios aceitam probabilisticamente soluções inferiores (em comparação com a atual).

Além disso, a abordagem adotada neste trabalho também se baseia na combinação empírica, visando otimizar a eficiência do algoritmo em encontrar soluções para o 3SAT. Isso envolve a seleção criteriosa de parâmetros, estratégias de vizinhança e critérios de parada que foram adotados para se adequar às características específicas do problema 3SAT. A aplicação dessas técnicas e estratégias busca melhorar a capacidade do Simulated Annealing em encontrar soluções de forma mais rápida, contribuindo para a resolução eficaz deste problema de decisão combinatória.

Também é objetivo do trabalho, comparar o algoritmo SA desenvolvido com o método aleatório de resolução (busca aleatória), para verificar a eficiência da meta-heurística utilizada no trabalho.

3. Metodologia de Desenvolvimento

Para atingir o objetivo do trabalho, é importante descrevermos inicialmente o ambiente do sistema, para ajudar na implementação do algoritmo.

- **Observável** Parcialmente
- **Determinístico / Estocástico:** Estocástico
- **Episódico / Sequencial:** Sequencial
- **Estático / Dinâmico:** Estático
- **Discreto / Contínuo:** Misto
- **Multiagente / Único Agente:** Único Agente

Será feita uma série de testes empíricos, utilizando a implementação do pseudocódigo baseado no artigo de [Spears 1993] para aplicação do SA no problema 3SAT.

Existem três instâncias de problemas que serão analisadas, comparando o número de iterações por cláusulas resolvidas, a primeira possui 20 variáveis e 91 cláusulas, a segunda 100 variáveis e 430 cláusulas e a última 250 variáveis e 1065 cláusulas. Os algoritmos (SA e Busca aleatória) serão executado 10 vezes, com objetivo de garantir melhor precisão nos resultados em termos probabilístico.

Alguns parâmetros do SA são apresentados a seguir, como forma de facilitar o entendimento da meta-heurística.

Temperatura inicial: ao utilizar uma temperatura inicial muito baixa, o algoritmo SA poderá ter dificuldades de sair dos mínimos locais, uma vez que a possibilidade de aceite de soluções ruins também será baixa. Já ao usar uma temperatura muito alta, a busca será próxima da aleatória.

Taxa de variação de temperatura: se a variação for muito rápida, possivelmente não se atingirá uma boa solução. Todavia, se a mudança de temperatura for lenta, o tempo de execução do algoritmo será alto.

Solução Inicial: para que o algoritmo inicie a busca, é necessária uma solução inicial. Esta solução pode ser aleatória ou seguir alguma lógica.

Vizinhança: o algoritmo de têmpera simulada deve ser capaz de procurar em todo o espaço de soluções. Se os passos forem muito grandes, o algoritmo se comporta como uma busca aleatória, enquanto que passos muito pequenos têm maior dificuldade de escapar de ótimos locais.

Reaquecimento: quando o algoritmo SA atinge baixas temperaturas, este terá dificuldades de sair dos mínimos locais. Para evitar isso, utiliza-se o reaquecimento para elevar novamente a temperatura e escapar dos mínimos locais.

Critérios de Parada: o SA deve parar sua busca quando atingir o mínimo de energia ou um certo número de iterações (utilizado como critério nesse trabalho).

4. Modelagem do problema

A função objetivo a ser otimizada no problema é minimizar o número de cláusulas falsas. A solução ótima é o conjunto de variáveis com cláusulas falsas sendo igual a 0.

5. Configurações do algoritmo

O algoritmo possui diversas etapas de execução, apontadas e descritas a seguir:

1. Primeiro é realizada a leitura dos dados em um arquivo de texto.
2. Durante a leitura, as cláusulas são separadas em um conjunto de vetores de vetores.
3. Será gerada uma solução inicial aleatória.
4. A solução aleatória, será o valor da solução inicial.
5. Os itens a seguir serão repetidos 250 mil vezes:
    (a) Gerar um vizinho dessa solução inicial.
    (b) Comparar o valor do vizinho com o valor da solução inicial.
    (c) Caso o valor do vizinho seja menor que o valor da solução, o vizinho assume a solução.
    (d) Caso o valor do vizinho seja menor que o melhor valor, o valor do vizinho assume o melhor valor.
    (e) Caso contrário, com base na temperatura, o algoritmo determina se vizinho assume ou não.
6. Retorna o melhor valor.

6. Estratégias e justificativas

Como apresentado na sessão de Metodologia de Desenvolvimento, alguns parâmeros do SA devem ser ajustados para melhorar o tempo de busca de uma solução ou até mesmo achar uma solução razoável. Nessa sessão, alguns desses parâmetros serão apresentados assim como as justificativas.

Algumas observações foram levadas em conta:

- **Objetivo do algoritmo**: A escolha entre maximização e minimização depende da natureza do problema. No contexto do problema 3SAT, o objetivo é minimizar o número de cláusulas falsas. Portanto, a minimização foi escolhida, pois facilita o desenvolvimento e a interpretação dos resultados.
- **Geração de vizinho**: A estratégia de geração de vizinhos é crucial para o desempenho do SA. Uma abordagem comum é perturbar ligeiramente a solução atual para obter um vizinho. A perturbação pode ser feita invertendo o valor de uma variável aleatória ou trocando os valores de duas variáveis. A escolha da estratégia de vizinhança deve garantir que o espaço de soluções seja adequadamente explorado.
- **Número de iterações**: O número ideal de iterações depende da complexidade do problema e da capacidade computacional disponível. Embora 250.000 iterações tenham sido sugeridas inicialmente, observou-se que um número menor de iterações foi suficiente para obter bons resultados em todos os casos testados. Isso sugere que o algoritmo foi eficiente em explorar o espaço de soluções.
- **Perturbações**: Durante os experimentos, foi observado que um maior número de perturbações resulta em uma busca mais global. Isso é consistente com a natureza do SA, onde perturbações frequentes podem ajudar o algoritmo a escapar de mínimos locais e explorar mais amplamente o espaço de soluções.
- **Parâmetro *Smax***: O valor de *Smax* influencia a taxa de resfriamento do algoritmo. Um valor de 5 foi encontrado como o mais eficaz nos testes, indicando que essa taxa de resfriamento permitiu ao algoritmo equilibrar adequadamente a exploração local e global.

Ao ajustar esses parâmetros, foi possível garantir que o SA funcionasse de maneira eficaz para o problema 3SAT, resultando em soluções de alta qualidade em um tempo computacional razoável. Nos subcapítulos a seguir, será discutido as estratégias mais relevantes.

6.1. Configuração de Temperatura

A temperatura é um dos parâmetros mais críticos no algoritmo Simulated Annealing (SA). Ela desempenha um papel fundamental na determinação da probabilidade de aceitar soluções piores do que a solução atual. A configuração adequada da temperatura e de sua taxa de decaimento pode influenciar significativamente a eficácia do algoritmo.

- **Temperatura Inicial**: A temperatura inicial determina o ponto de partida do algoritmo. Uma temperatura inicial alta permite que o algoritmo aceite quase todas as novas soluções no início, garantindo uma exploração ampla do espaço de soluções. Isso ajuda o algoritmo a evitar ficar preso em mínimos locais no início da busca. No entanto, uma temperatura inicial muito alta pode fazer com que o algoritmo se comporte de maneira quase aleatória, enquanto uma temperatura muito baixa pode restringir a exploração.

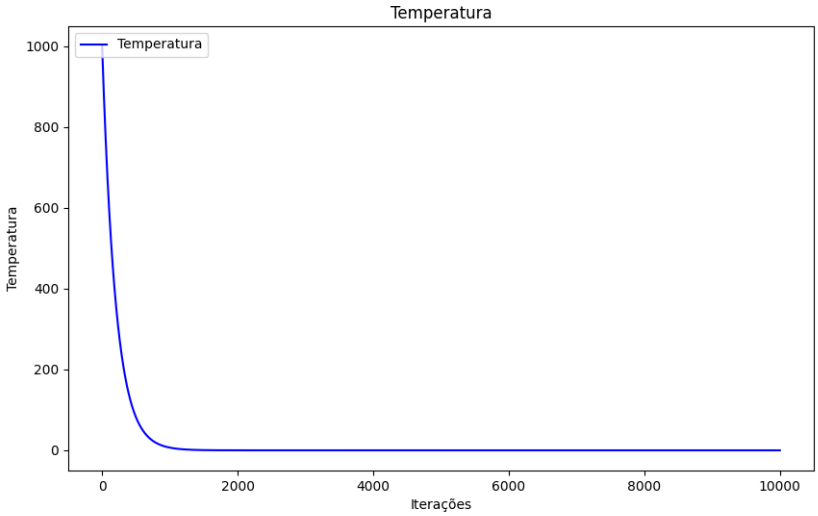

Figura 1. Comportamento da queda de temperatura.

- **Taxa de Decaimento da Temperatura**: A taxa de decaimento controla a velocidade com que a temperatura diminui ao longo das iterações. Uma taxa de decaimento rápida pode fazer com que o algoritmo resfrie muito rapidamente, possivelmente perdendo boas soluções. Por outro lado, uma taxa de decaimento lenta pode resultar em um tempo de execução excessivamente longo e também pode fazer com que o algoritmo fique preso em mínimos locais por muito tempo.
  Neste trabalho, foi analisado as diferentes taxas de decaimento, utilizando as funções disponibilizadas no escopo do trabalho e analisadas no artigo de [Nourani and Andresen 1998]. Concluindo que a melhor taxa de decaimento para esse problema foi dada por:

$$x = \left(1 - \frac{i}{n}\right)^t$$

- **Reaquecimento**: Em algumas variantes do SA, um mecanismo de reaquecimento é introduzido. Quando a temperatura cai abaixo de um certo limite, ela é aumentada novamente para permitir uma nova fase de exploração. Isso pode ajudar o algoritmo a escapar de mínimos locais profundos.
  Neste trabalho, não foi preciso dessa estratégia pois foi possível obter um resultado ótimo ao fim das execuções, sem reaquecimento
- **Critério de Parada Baseado na Temperatura**: Em alguns casos, o algoritmo SA pode usar a temperatura como um critério de parada. Por exemplo, quando a temperatura cai abaixo de um valor mínimo especificado, o algoritmo pode ser interrompido, assumindo que uma solução suficientemente boa foi encontrada.
  Neste trabalho, será usado o número de iterações como critério de parada.

Após serem justificados os parâmetros referentes a temperatura, temos a figura 1, o gráfico de queda da temperatura para os diferentes cenários.

6.2. Configurações de vizinhança

A vizinhança é uma parte fundamental de algoritmos de otimização como o Simulated Annealing. A vizinhança se refere à geração de soluções vizinhas a partir de uma solução

atual. Algumas estratégias para geração de vizinhança são: Flip de Variáveis, Troca de Variáveis e Operações mais complexas.

Nesse trabalho, será utilizada a estratégia de Flip de variável por ser uma estratégia simples porém eficaz. Primeiro, é escolhido de forma aleatória um número discreto de 1 até 8, que representará a quantidade de variáveis a serem alteradas, ou seja, quantas variáveis terão seus valores invertidos de 0 para 1, ou vice-versa, em relação à solução atual.

Após essa determinação, repetimos esse processo um número n aleatório de vezes. O valor de n é escolhido aleatoriamente entre 0 e o número total de variáveis na instância do problema. Esse valor n, determina qual variável será efetivamente modificada em cada iteração.

Para determinar a troca de vizinho, foi utilizada a seguinte função:

$$\Delta = V_{vizinho} - V_{atual} \tag{2}$$

$$aux = e^{-\frac{\Delta}{t_i}} \tag{3}$$

7. Análise dos resultados obtidos

Para a execução dos algortimos, foram feitas algumas médias de tempo para atingir 0 cláusulas não satisfeitas: Os tempos foram em média:

- **20 variáveis**: 7 segundos
- **100 variáveis**: 55 segundos
- **250 variáveis**: 1 minuto e 40 segundos

7.1. 20 variáveis

Para 20 variáveis, tanto o algoritmo SA como RS tiveram sucesso e se mostraram eficientes, por se tratar de instâncias menores, a busca aleatória também foi capaz de resolver. O comportamento pode ser visto no gráfico da figura 2.

O gráfico em demonstra de forma clara que a temperatura influencia diretamente a taxa de aceitação de novos valores. Na medida em que a temperatura cai, o algoritmo aceita menos valores novos, ate que se mantenha na solução ótima.

O resultado das 10 execuções do SA e RS também podem ser observados na figura 2. Todas as execuções chegaram em 0 clausulas não satisfeitas.

Para os 10 experimentos dos algoritmos SA e RS com 20 variáveis a média e o desvio padrão são vistos na tabela 1.

Algoritmo	Desvio Padrão	Média
Simulated Annealing	0	0
Busca Aleatória	0	0

Tabela 1. Desvio padrão e média para diferentes algoritmos

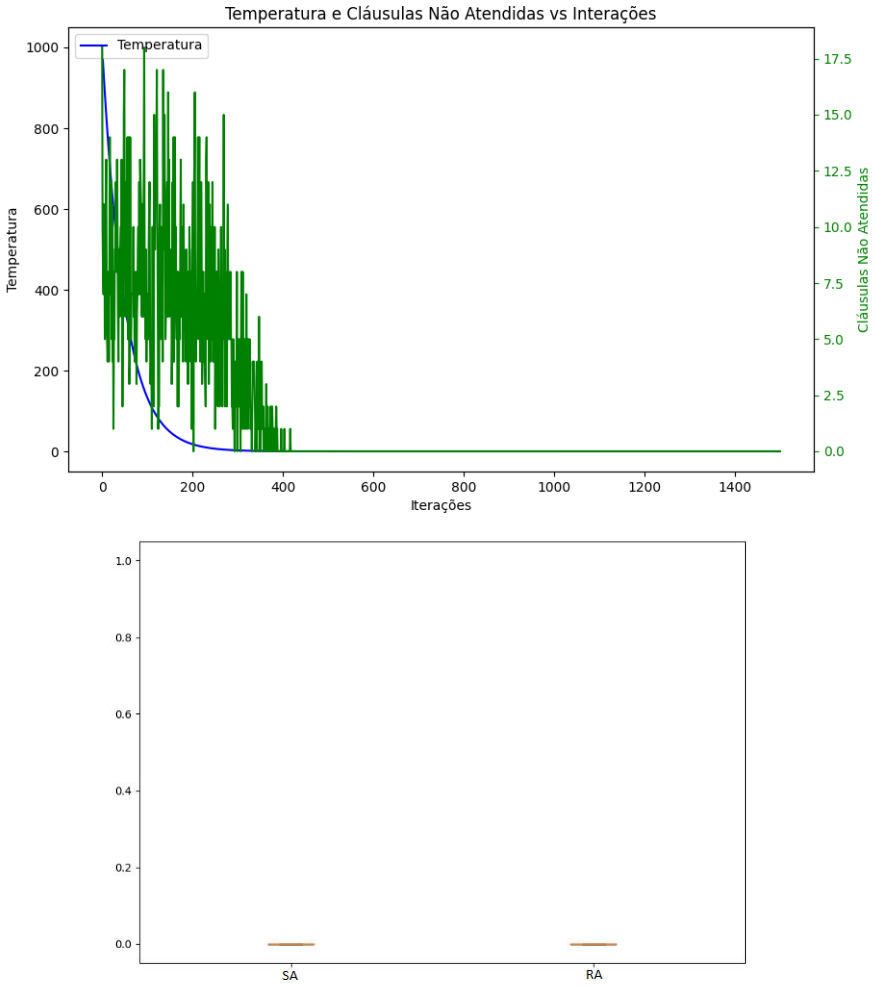

Figura 2. Temperatura, cláusulas não satisfeitas e box plot para 20 variáveis.

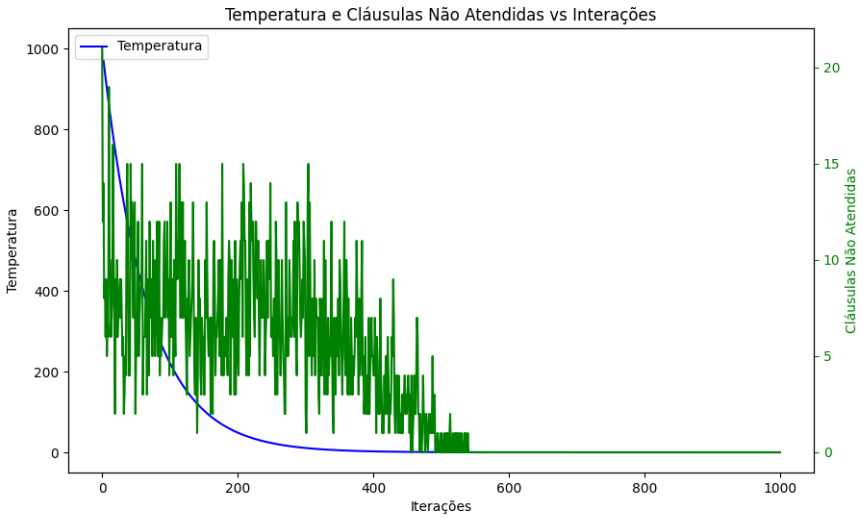

Figura 3. Temperatura e cláusulas não satisfeitas para 100 variáveis.

7.2. 100 variáveis

Para 100 variáveis, o tempo de resolução pelo algoritmo SA aumentou consideravelmente em relação a 20 variáveis e teve muito mais eficácia que o algoritmo de busca aleatória, que não foi capaz de resolver. O comportamento pode ser visto no gráfico da figura 3.

Para os 10 experimentos dos algoritmos SA e RS com 100 variáveis a média e o desvio padrão são vistos na tabela 2.

Algoritmo	Desvio Padrão	Média
Simulated Annealing	0.61375	0.81
Busca Aleatória	2.31548	6.95

Tabela 2. Desvio padrão e média para diferentes algoritmos

7.3. 250 variáveis

Neste cenário o uso de meta-heurística como SA torna-se mais evidente, pois é o problema com mais variáveis e maior dificuldade para satisfazer as cláusulas. Apesar de possuir alguns casos que não foi capaz de encontrar solução ótima, ainda sim teve grande acuratividade e desempenho. O comportamento pode ser visto no gráfico da figura 4.

Para os 10 experimentos dos algoritmos SA e RS com 250 variáveis a média e o desvio padrão são vistos na tabela 3.

Algoritmo	Desvio Padrão	Média
Simulated Annealing	0.66351	1.33
Busca Aleatória	3.71568	18.97

Tabela 3. Desvio padrão e média para diferentes algoritmos

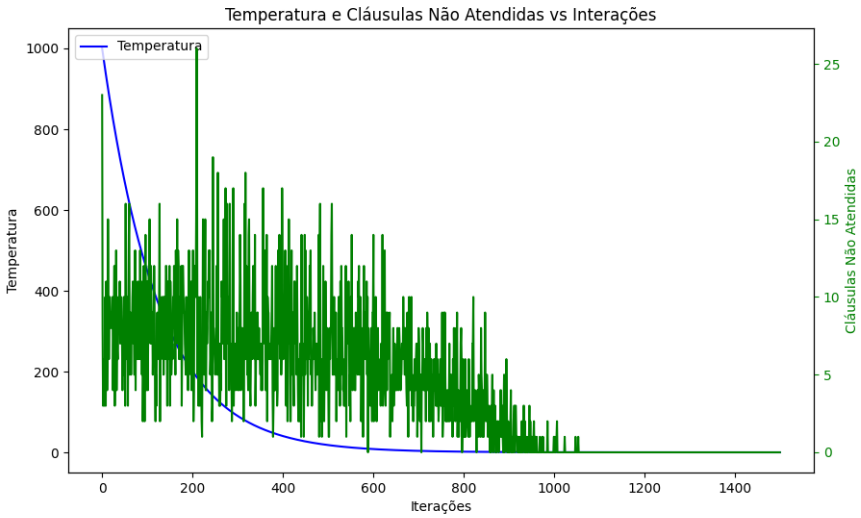

Figura 4. Temperatura e cláusulas não satisfeitas para 250 variáveis.

8. Conclusão

Podemos concluir que o uso da meta-heurística Simulated Annealing, é uma forma muito eficaz para tratar determinadas classes de problemas como 3SAT. O algoritmo SA foi comparado com Random Search e foi possível notar grande diferença, servindo como base para demonstrar a eficiência do SA.

As configurações adequada, principalmente da temperatura e de sua taxa de decaimento é essencial para o sucesso do SA. É comum realizar experimentos preliminares para calibrar esses parâmetros para um problema específico, garantindo que o algoritmo tenha um equilíbrio entre exploração local e exploração global ao longo de sua execução.

Referências

Araújo, M. (2023). Resolução do problema max-sat utilizando têmpera simulada. Acessado em: [Data de Acesso].

Back, e. a. (1998). Computação evolutiva para sat. *Desconhecido*.

Gu, e. a. (1997). Algoritmo davis-putnam. *Desconhecido*.

Gu, J. and Puri, R. (1995). Asynchronous circuit synthesis with boolean satisfiability. *IEEE Transactions on Computer-Aided Design of Integrated Circuits*, 14(8):961–973.

Marchiori, E. and Rossi, C. (1999). A flipping genetic algorithm for hard 3-sat problems. *Desconhecido*.

Nguyen, T., Perkins, W., Laffrey, T., and Pecora, D. (1985). Checking an expert system knowledge base for consistency and completeness. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI 85)*, pages 375–378. Morgan Kaufmann.

Nourani, Y. and Andresen, B. (1998). A comparison of simulated annealing cooling strategies. *Journal of Physics A: Mathematical and General*, 31(41):8373.

Spears, W. M. (1993). Simulated annealing for hard satisfiability problems. *Cliques, Coloring, and Satisfiability*, 26:533–558.