30509, Computer Programming Assignment

Jakob Lutz, 3253325

January 2025

Introduction

The 3-SAT problem is a well-known NP-complete problem in computer science. In this project we will analyze the performance of the Simulated Annealing algorithm as a solver for the 3-SAT problem. We will start by evaluating the effectiveness of Simulated Annealing for a varying numbers of clauses. We will then investigate some general properties of the 3-SAT problem by examining how the probability of solving random instances depends on the number of clauses and variables. The aim of this report is to provide insights into the effectiveness and limitations of Simulated Annealing and the universal behavior of 3-SAT under different clause-to-variable ratios.

1 Performance of Simulated Annealing scheme

The performance of Simulated Annealing as a solver for the 3-SAT problem depends heavily on the parameters we choose for the annealing process. Therefore, it is crucial to choose proper values for the following parameters:

Parameter	**Description**
Monte Carlo Steps (`mcmc_steps`)	This parameter controls the number of Monte Carlo Steps evaluated at each temperature. A higher value increases accuracy but also computational cost.
Annealing Steps (`anneal_steps`)	This parameter determines how gradually the temperature decreases. More steps allow the system to equilibrate at each temperature, which increases accuracy but also computational cost.
Initial Temperature ($\beta_0 = 1/T_0$)	This parameter determines the randomness at the start of the process. A lower β_0 (higher temperature) encourages exploration at the beginning.
Final Temperature ($\beta_1 = 1/T_1$)	This parameter controls the selectivity at the end of the annealing process. A higher β_1 (lower temperature) focuses on exploitation and fine-tuning.
Random Seed (`seed`)	This parameter ensures reproducibility.

Table 1: Parameters for Simulated Annealing

Optimal Parameters for $M = 200$

In order to determine the optimal parameters for solving the 3-SAT problem using Simulated Annealing with $M = 200$ and $N = 200$, we need to find the right balance between exploration and exploitation during the optimization process. I was able to find solutions consistently with **500** Monte Carlo Steps and **10** annealing steps. Furthermore, I chose the value **1.0** for β_0 in order to allow the algorithm to explore a wide variety of configurations early in the annealing process. I chose the value **10.0** for β_1 to make sure that the algorithm focuses on exploitation near the end. For the random seed I chose the number **42** since according to "The Hitchhiker's Guide to the Galaxy" 42 is the "Answer to the Ultimate Question of Life, the Universe, and Everything".

Acceptance rate for $M = 200$

The algorithm successfully converged to a solution with a final cost of 0. Figure 1 shows the evolution of the acceptance rate during the annealing process. At $\beta = 1.0$, the acceptance rate is high (0.786). This allows the algorithm to explore a wide variety of configurations early in the annealing process. The initial cost decreases from 23 to 5 in the first step. The acceptance rate declines as β increases and the temperature decreases. This indicates a shift from exploration to exploitation, which is a key characteristic of the Simulated Annealing algorithm. At $\beta = 3.25$, the algorithm finds an optimal configuration with a cost of 0. This means that all clauses are satisfied and we have solved the 3-SAT problem. After finding a solution, the acceptance rate stabilizes between 0.608 and 0.664 for higher values of β.

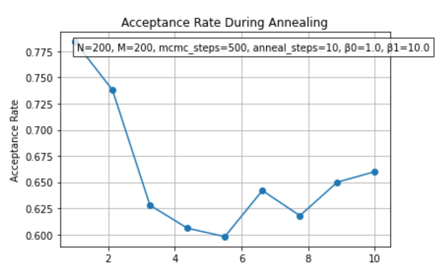

Figure 1: Acceptance Rate Evolution for $M = 200$

Acceptance rate for $M = 400$ and $M = 600$

For $M = 400$ and $M = 600$ the same parameters used before still yield an optimal solution with a final cost of 0. The following two plots show the evolution of the acceptance rate during the annealing process for $M = 400$ (Figure 2) and $M = 600$ (Figure 3). For $M = 400$, the acceptance rate starts high (70.6% at $\beta = 1.0$), which leads to a fast exploration. The optimal solution is found quickly at $\beta = 3.25$. For $M = 600$, the acceptance rate starts lower (60.6% at $\beta = 1.0$). Due to increased constraints, the algorithm requires more iterations for effective exploration. The solution is only found at the penultimate annealing step with $\beta = 10.0$.

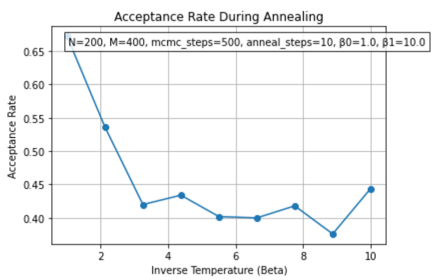

Figure 2: Acceptance Rate Evolution for $M = 400$

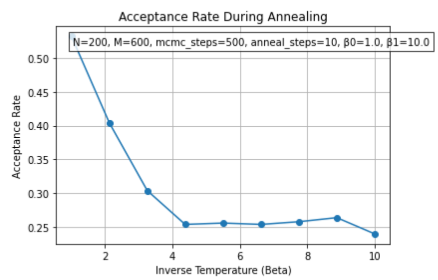

Figure 3: Acceptance Rate Evolution for $M = 600$

Acceptance rate for $M = 800$ and $M = 1000$

With the current parameters, Simulated Annealing is not always able to find a solution as M increases. For $M = 800$, the algorithm performs well with only 2 unsatisfied clauses. For $M = 1000$ however, the final cost is higher with 11 unsatisfied clauses. The optimization problem becomes increasingly more challenging as M increases. Because of the increased constraints it is harder for Simulated Annealing to escape local minima. In order to be able to also solve larger problems, we need to adjust our parameters. Therefore, I chose to increase the Monte Carlo steps to 1000, increase the annealing steps to 20 and reduce β_0 to 0.5. Now, I was able to find a solution for $M = 800$ and a configuration for $M = 1000$ with only 8 unsatisfied clauses. When looking at the acceptance rate for $M = 800$ (Figure 4) and $M = 1000$ (Figure 5), we can see that the acceptance rates start high at 0.7 and 0.6 respectively. Then the acceptance rates decreases gradually down to 0.1. For $M = 800$ the optimal solution is found after 12 annealing steps at $\beta = 6.31$, whereas for $M = 1000$ the algorithm gets stuck at a local minimum with 8 unsatisfied clauses.

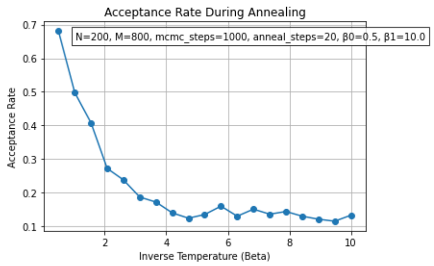

Figure 4: Acceptance Rate Evolution for $M = 800$

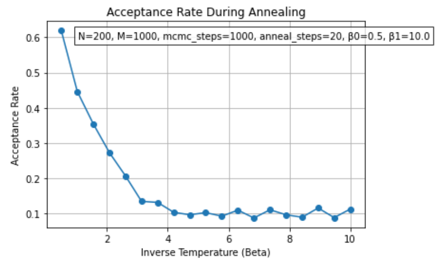

Figure 5: Acceptance Rate Evolution for $M = 1000$

2 General 3-SAT properties

For the second part of the project, we are doing some empirical tests to find some general properties of the 3-SAT problem. The empirical probability of solving a random 3-SAT instance using Simulated Annealing is defined as:

$$P(N, M) = \frac{\text{Number of Solved Problem Instances}}{\text{Total Number of Problem Instances}}$$

where N is the number of variables and M is the number of clauses.

Empirical test for $N = 200$

In this experiment, we fix $N = 200$ and compute $P(N, M)$ for increasing values of M, specifically $M \in \{400, 500, 600, 700, 800, 900, 1000\}$. For each value of M, we repeat the experiment over $n = 30$ random instances. When choosing values for the parameters, it is important to find the right balance between computational cost and accuracy. Therefore, I chose the same parameters as at the beginning (see explanation on page 2):

- 500 Monte Carlo steps
- 10 annealing steps
- $\beta_0 = 1.0$
- $\beta_1 = 10.0$

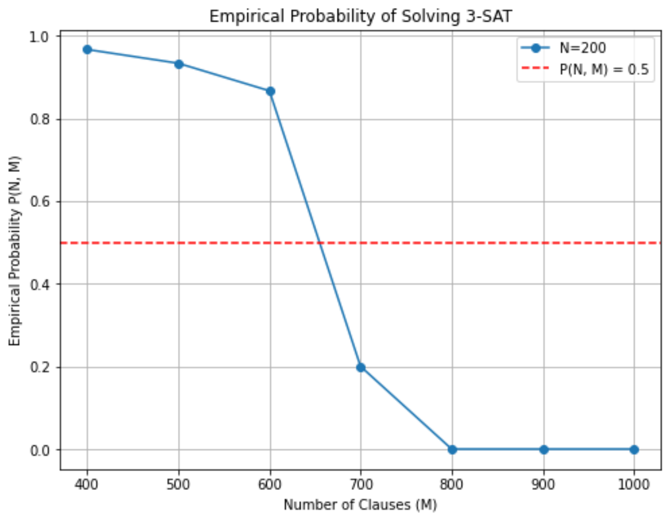

Figure 6: Empirical test for $N = 200$

From this plot we can see that as the number of clauses (M) increases, the empirical probability ($P(N, M)$) decreases. This means that solving the 3-SAT problem becomes harder with increasing number of clauses.

The dotted red line in the graph shows the algorithmic threshold $M_{\text{Alg}}(N)$. The algorithmic threshold $M_{\text{Alg}}(N)$ is the number of clauses, at which the empirical probability of solving a clause with Simulated Annealing equals $\frac{1}{2}$. For $N = 200$ the algorithmic threshold $M_{\text{Alg}}(200) = 655$. At this critical point we observe a sharp decrease in the empirical probability. This means that solving a 3-SAT problem transitions from being very likely to being very unlikely. Below this threshold the Simulated Annealing algorithm is very likely to find a solution. Beyond this threshold however, the clause density is too high for Simulated Annealing to reliably solve the problem.

Empirical test for $N \in \{300, 400, 500, 600\}$

To investigate whether the algorithmic threshold $M_{\text{Alg}}(N)$ depends on the choice of N, we repeat the analysis for different values of N, specifically $N \in \{300, 400, 500, 600\}$. For each of these values of N, we compute the probability $P(N, M)$ for increasing values of $M \in \{400, 500, 600, 700, 800, 900, 1000\}$ and determine the algorithmic threshold $M_{\text{Alg}}(N)$. I kept the same parameters as before. Although this is computationally costly, I wanted to ensure accurate results. After some time we get the following results (Figure 7):

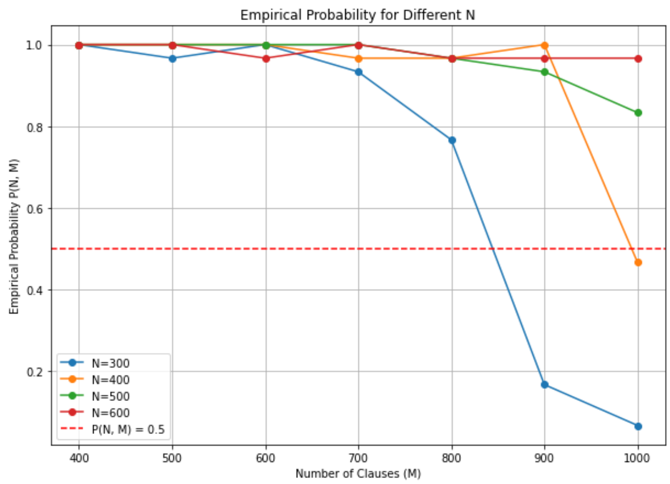

Figure 7: Empirical test for $N \in \{300, 400, 500, 600\}$

This plot illustrates the relationship between the empirical probability $P(N, M)$ of solving 3-SAT instances and the number of clauses M for various values of N ($N \in \{300, 400, 500, 600\}$). The dashed red line represents the algorithmic threshold $P(N, M) = 0.5$. We make the following observations:

- For the value $N = 300$, the empirical probability $P(N, M)$ reaches the threshold $M_{\text{Alg}}(N)$ at $M = 844$. At this point solving a 3-SAT problem transitions from being very likely to being very unlikely.
- For the value $N = 400$, the threshold $M_{\text{Alg}}(N)$ is reached at $M = 980$. Beyond this threshold, the clause density is too high for Simulated Annealing to reliably solve the problem.
- For larger values of N (i.e. $N = 500$ and $N = 600$), Simulated Annealing is able to reliably find solutions to the 3-SAT problem even at higher values of M. Larger instances of N require a higher clause density to reach the critical threshold. This makes sense since increasing the value of N simplifies the problem and therefore the algorithm is still able to find a solution for the 3-SAT problem even with a high number of clauses.

Scaling of M_{Alg} with N

In the final part of this project, we are going to analyze the relationship between the algorithmic threshold M_{Alg} and the number of variables N.

By plotting the empirical probability $P(N, M)$ as a function of the rescaled ratio M/N, we observe an approximately universal behavior for different values of N.

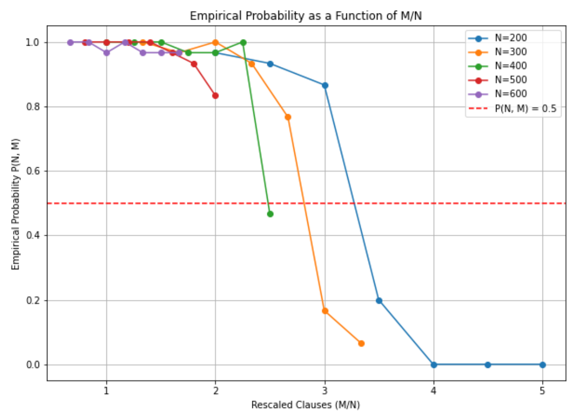

Figure 8: Scaling of M_{Alg} with N

When the number of clauses M is rescaled by the number of variables N, the curves for different values of N ($N \in \{200, 300, 400, 500, 600\}$) approximately collapse into a single curve. This means that the algorithmic threshold M_{Alg} scales approximately linear with N. The critical clause-to-variable ratio, at which the empirical probability $P(N, M) = 0.5$ is approximately $M/N \approx 3$. At this ratio we observe a sharp decrease in the empirical probability of finding an optimal solution. The behavior of $P(N, M)$ as a function of M/N is consistent for different values of N. This means that the clause-to-variable ratio governs the complexity of the problem rather than the absolute values of M and N.

It is interesting to note that the value of the critical clause-to-variable ratio observed in this project ($M/N \approx 3$) differs from the ratio $M/N \approx 4.25$ reported in the literature[1].By using analytical methods and theoretical insights this study was able to find solutions consistently up to a clause-to-variable ratio of 4.25. This finding highlights the limitations of Simulated Annealing in solving SAT problems due to the stochastic nature of the algorithm.

[1] O. Dubois and Y. Boufkhad, "A general upper bound for the satisfiability threshold of random r-sat formulae," *Journal of Algorithms*, vol. 24, pp. 395–420, 1997.