Evaluating the Efficacy of Decryption Methods for Substitution Ciphers Using Markov Chains of Various Orders

YAGMUR DELAL CENGIZ

This research investigates the effectiveness of different decryption methods—Hill Climbing, Simulated Annealing, and Metropolis-Hastings Markov Chain Monte Carlo (MCMC)—in deciphering substitution ciphers using Markov chains of varying orders. The study evaluates these methods across first to seventh-order Markov chains, assessing their accuracy and processing time. The results indicate that while Hill Climbing and Simulated Annealing showed limited success, MCMC demonstrated superior performance in certain cases. Specifically, MCMC outperformed the other methods in higher-order contexts, particularly in the fifth-order Markov chain scenario. Despite some improvements with increased context, the effectiveness of all methods varied significantly, with Simulated Annealing excelling in specific orders and MCMC consistently showing the highest accuracy in more complex contexts. These findings highlight the nuanced impact of Markov chain order on decryption efficacy and suggest avenues for future research in optimizing these methods for better performance in cryptographic applications. The program used for this research can be reached on Github

Keywords: Decryption methods, Substitution ciphers, Markov chains, Hill Climbing, Simulated Annealing, Metropolis-Hastings MCMC, Cryptography, Accuracy, Processing time.

1. Introduction

Substitution ciphers have a long history in cryptography, often being one of the first methods introduced to those new to the field. Despite their conceptual simplicity, substitution ciphers can present significant challenges in cryptanalysis, particularly when the cipher text is lengthy and lacks any discernible patterns. In a substitution cipher, each letter in the plaintext is replaced by a corresponding letter in the ciphertext according to a fixed system. The challenge for a cryptanalyst is to discover this fixed system, which can be especially difficult when the key length is the same as the length of the alphabet being used [12]. This study delves into the application of Markov chains in the cryptanalysis of substitution ciphers. Markov chains, which are mathematical models that undergo transitions from one state to another in a chain-like process, are well-suited for modeling the statistical properties of natural languages. To address these challenges, this study explores the use of statistical models, particularly Markov chains, in combination with heuristic optimization techniques such as simulated annealing, hill climbing, and Markov Chain Monte Carlo (MCMC).[13]. We aim to contribute to the field not only in understanding the vulnerabilities of substitution ciphers but also in the broader implications for cryptography as a whole. As digital communication becomes increasingly integral to everyday life, the need for robust cryptographic systems is more critical than ever. The principles learned from analyzing basic ciphers can be applied to more complex systems, thereby enhancing our ability to protect sensitive information in a variety of real-world applications, from secure online banking to private messaging.

The novelty of our approach is investigating how the order of the Markov chain affects the decryption process. Specifically, we ask: *How do the order of the Markov chain and the choice of optimization algorithm (hill climbing, simulated annealing, MCMC) impact the effectiveness of substitution cipher decryption?* The order of a Markov chain refers to the number of previous states that the next state depends on. A first-order Markov chain assumes that each character in a text depends only on the preceding character, while a higher-order Markov chain assumes that each character may depend on multiple preceding characters. This higher-order dependency could potentially capture more complex language structures, thus improving the accuracy of decryption [1]. Our hypothesis in this research is that as the order of the Markov chain increases, the decryption accuracy will also increase, up to a point. However, this improvement may come at the cost of increased computational complexity, as higher-order Markov chains require the estimation and storage of a larger number of parameters. Thus, the study also aims to find an optimal balance between the order of the Markov chain and the computational resources required for the cryptanalysis. In this paper, we investigate the effectiveness of different methods implemented with Markov chains of first-order to seventh-order in the cryptanalysis of substitution ciphers. Each order of Markov chain captures different levels of statistical dependency between characters in the plaintext, potentially offering varying degrees of accuracy in breaking the cipher. Furthermore, the study conducts a comparative analysis of these three algorithms to decrypt substitution ciphers. While hill climbing and simulated annealing are designed to navigate the vast search space of possible decryptions, MCMC offers a probabilistic approach to explore the solution space by sampling from a distribution, thus providing another perspective on how to achieve global optimization. By conducting a comparative analysis, we aim to identify the most effective approach for decrypting ciphertext encrypted using a substitution cipher. The results of this research have the potential to inform both historical cryptanalysis efforts and the development of future cryptographic algorithms.

The remainder of this paper is structured as follows: Section 2 presents a literature review, summarizing the key contributions in the field of cryptanalysis using Markov chains. Section 3 describes the methodology, including the specific algorithms and data sets used in the study. Section 4 provides an analysis of the results, while Section 5 offers a mathematical estimation of the effectiveness of different Markov chain orders. Finally, Section 6 concludes the paper with a discussion of the findings and their implications for cryptographic research.

2. Literature Review

The basic idea of substituting one letter for another can be traced back to classical times, with the Caesar cipher being one of the most well-known examples [15]. However, the simplicity of substitution ciphers makes them vulnerable to various cryptanalytic attacks, especially when frequency analysis is applied. Markov chains, introduced by Andrey Markov in the early 20th century, have become a powerful tool in cryptanalysis due to their ability to model the probabilistic nature of language. The use of Markov chains in cryptanalysis, particularly for substitution ciphers, has been explored in various studies. For instance, Sinkov [17] provides an early discussion of using statistical methods, including Markov chains, to break classical ciphers. Subsequent research by Norvig [18] and others has shown that higher-order Markov models can significantly improve the accuracy of cryptanalysis by better capturing the dependencies between letters. Recent advances have focused on comparing the effectiveness of different orders of Markov chains in cryptanalysis. Studies have demonstrated that while first-order Markov chains capture basic frequency information, higher-order models can better exploit contextual information, leading to more accurate decryption results [19]. Additionally, research has shown that the computational complexity increases with the order of the Markov chain, making it crucial to balance accuracy and efficiency in practical applications [20].

Markov Chains in Cryptanalysis

Markov chains have been instrumental in the development of statistical methods for cryptanalysis. Shannon's foundational work on information theory laid the groundwork for applying Markov models to cryptographic analysis. Shannon's seminal paper demonstrated that statistical models of text, such as first-order Markov chains, could be used to break simple substitution ciphers [12]. Several research further expanded and analyzed the use of Markov Chains for cryptanalysis. Chen et al. (2012) investigated the use of Markov Chain Monte Carlo (MCMC) methods to attack classical ciphers [2]. Karlof et al. (2003) showed that Hidden Markov Models can be used to cryptanalyze randomized side channel countermeasures [3].

Heuristic Optimization Techniques in Cryptanalysis

Hill Climbing

Hill climbing is a fundamental heuristic optimization technique that involves iteratively improving a candidate solution by making local changes and selecting the change that yields the greatest improvement in the objective function. In the context of substitution ciphers, the objective function often measures the alignment of the decrypted text with expected linguistic patterns. Research by Kaeding et al. (2019) demonstrated an application of hill climbing to monoalphabetic substitution ciphers[4]. It has been noted that while hill climbing could quickly find good solutions, it was prone to getting stuck in local optima . To address this, research proposed hybrid approaches that integrated hill climbing with other optimization techniques.

Simulated Annealing

Simulated annealing has been applied to cryptanalysis of various ciphers, often in comparison with other optimization techniques. For simplified-DES, Tabu Search outperformed simulated annealing in a cipher-text only attack scenario[5]. However, simulated annealing has shown promise in automated cryptanalysis of mono-alphabetic substitution ciphers, with proven convergence and potential for more complex block ciphers[6]. In a comparative study on transposition ciphers, genetic algorithms, Tabu Search, and simulated annealing were evaluated for their effectiveness in automated attacks [7]. These studies demonstrate the applicability of simulated annealing and other evolutionary computation techniques to cryptanalysis, addressing NP-hard combinatorial problems in cipher breaking. The research suggests that while simulated annealing can be effective, its performance may vary depending on the specific cipher and attack scenario, highlighting the importance of comparative analyses in cryptanalysis research.

Markov Chain Monte Carlo (MCMC) in Cryptanalysis

Metropolis-Hastings Algorithm

The Metropolis-Hastings algorithm is a powerful Markov Chain Monte Carlo (MCMC) method used to simulate multivariate distributions and approximate complex probability functions [10]. MCMC techniques generate samples by running a Markov Chain with a stationary distribution that follows the input function, utilizing repeated random sampling to exploit the law of large numbers. The Metropolis-Hastings algorithm has diverse applications, including optimization tasks, solving non-deterministic polynomial time problems, and cryptographic decoding [11]. In cryptanalysis, MCMC methods have been successfully applied to break classical ciphers, including simple substitution, transposition, and substitution-plus-transposition ciphers, even for key lengths up to 40 [14]. The algorithm's versatility is demonstrated in various implementations, such as acceptance-rejection sampling and block-at-a-time scans, with the Gibbs sampler being a special case of the Metropolis-Hastings algorithm. [10, 16]

3. Mathematical Foundations

3.1. *Definition 1: Substitution Cipher*

A substitution cipher is a method of encryption by which units of plaintext are replaced with ciphertext according to a fixed system; the ”units” may be single letters, pairs of letters, triplets of letters, and so forth. In a simple substitution cipher, the ciphertext alphabet is a permutation of the plaintext alphabet.

Definition 2: Markov Chains

A **Markov chain** is a stochastic process characterized by the Markov property, which states that the future state of the process depends only on the present state and not on the sequence of events that preceded it. Formally, let $\{X_n\}_{n\geq 0}$ be a discrete-time Markov chain with state space S. The process is said to satisfy the Markov property if:

$$P(X_{n+1} = x \mid X_n = x_n, X_{n-1} = x_{n-1}, \ldots, X_0 = x_0) = P(X_{n+1} = x \mid X_n = x_n)$$

for all $x, x_n, x_{n-1}, \ldots, x_0 \in S$ and $n \geq 0$.

Transition Probability Matrix
The behavior of a Markov chain is often described using the **transition probability matrix** P, where $P_{ij} = P(X_{n+1} = j \mid X_n = i)$ represents the probability of transitioning from state i to state j. The matrix P is stochastic, meaning that each row sums to 1:

$$\sum_{j\in S} P_{ij} = 1 \quad \text{for all } i \in S.$$

Stationary Distribution
A Markov chain is said to have a **stationary distribution** π if:

$$\pi_j = \sum_{i\in S} \pi_i P_{ij}$$

for all $j \in S$, where π is a probability distribution satisfying:

$$\sum_{i\in S} \pi_i = 1 \quad \text{and} \quad \pi_i \geq 0 \text{ for all } i \in S.$$

The stationary distribution π remains unchanged as the Markov chain evolves over time.

3.2. *Theorem 1: Convergence of Markov Chains*

For a Markov Chain with a finite state space that is irreducible and aperiodic, the chain converges to a unique stationary distribution, regardless of the initial state. This theorem ensures that the long-term behavior of the chain can be described by a stationary distribution, which is critical for analyzing the statistical properties of ciphertext generated by substitution ciphers.

Heuristic Optimization Algorithms

Hill Climbing
Hill climbing is a local search algorithm that continuously moves towards the direction of increasing value of the objective function. Mathematically, let $f : \mathbb{R}^n \to \mathbb{R}$ be the objective function, and let x^* be the current state. At each step, hill climbing evaluates neighboring states x' and selects the state x' that maximizes f:

$$x^{(t+1)} = \arg\max_{x' \in N(x^t)} f(x'),$$

where $N(x^t)$ represents the neighborhood of x^t. The algorithm stops when no better neighbors are found, potentially getting stuck in local optima.

Algorithm 1 Hill Climbing Algorithm

```
Data: Initial solution $x$
Result: Optimal solution
while not terminated do
    Generate neighboring solutions of $x$  Select the best neighbor $x^*$  if $f(x^*) > f(x)$ then
        $x \leftarrow x^*$
```

Simulated Annealing
Simulated annealing is an optimization technique inspired by the annealing process in metallurgy. It allows occasional moves to worse states to escape local optima. Formally, the probability of moving from state x to state x' is given by:

$$P(x \to x') = \min\left(1, \exp\left(\frac{f(x) - f(x')}{T}\right)\right),$$

where T is the temperature parameter that decreases over time. As $T \to 0$, simulated annealing converges to a solution similar to hill climbing.

Algorithm 2 Simulated Annealing Algorithm

```
Data: Initial solution $x$, temperature $T$
Result: Optimal solution
while not terminated do
    Generate a neighbor solution $x'$  Calculate the cost difference $\Delta E = f(x') - f(x)$  if $\Delta E < 0$ then
        $x \leftarrow x'$
    else
        $x \leftarrow x'$ with probability $\exp\left(\frac{-\Delta E}{T}\right)$
    Update temperature $T$ according to the cooling schedule
```

Lastly, MCMC methods are used to sample from complex probability distributions. The Metropolis-Hastings algorithm is a popular MCMC technique.

Metropolis-Hastings Algorithm

The Metropolis-Hastings algorithm generates a sequence of samples $\{X^n\}$ from a target distribution π. Let $q(x' \mid x)$ be the proposal distribution from which candidate states are sampled. The algorithm proceeds as follows:

Algorithm 3 Metropolis-Hastings Algorithm

```
Data: Initial state $x_0$, number of iterations $N$
Result: Sampled states
$x \leftarrow x_0$  for $i \leftarrow 1$ to $N$ do
    Propose a new state $x^*$ from a proposal distribution $q(x \mid x_{i-1})$  Calculate the acceptance ratio:
        $\alpha = \min\left(1, \frac{p(x^*)q(x_{i-1} \mid x^*)}{p(x_{i-1})q(x^* \mid x_{i-1})}\right)$
     Sample a uniform random variable $u \sim \text{Uniform}(0,1)$  if $u < \alpha$ then
        $x_i \leftarrow x^*$
    else
        $x_i \leftarrow x_{i-1}$
```

Convergence and Ergodicity
For MCMC to produce samples from the target distribution, the Markov chain must be **ergodic**, meaning it must be irreducible (every state can be reached from any other state) and aperiodic (the chain does not cycle through states in a fixed pattern). Under these conditions, the chain will converge to the stationary distribution π as the number of iterations approaches infinity.

4. Methodology

4.1. *Corpus Selection and Preprocessing*

The first stage of our methodology involves selecting and preprocessing a text corpus for training the Markov chain models. The corpus used in this study is loaded from a text file named `corpus_training.txt`. The preprocessing steps include converting the entire corpus to lowercase and removing any characters that do not belong to the English alphabet. This cleaning process is vital for ensuring that the Markov chain models are trained on a consistent set of textual patterns, as it eliminates noise such as punctuation, spaces, and line breaks that could otherwise disrupt the statistical modeling of language structures.

The preprocessing can be mathematically described as a function $P : S \to T$, where S is the original text corpus and T is the cleaned corpus:

$$T = P(S) = \text{Lowercase}(\text{RemoveNonAlphabetic}(S))$$

This ensures that only sequences of letters are retained, enabling the model to focus purely on letter transitions.

4.2. *Markov Chain Model Training*

Next, we train Markov chain models for various orders, ranging from 1 to 7. The order of a Markov chain defines how many previous characters are considered when predicting the next character in the sequence. A first-order Markov chain uses only the immediately preceding character, while a higher-order Markov chain considers longer sequences of preceding characters.

For each order n, we define the transition probabilities $P(X_t|X_{t-1},\dots,X_{t-n})$ using the following process:

1. For each sequence of n characters in the corpus, count the occurrences of each possible subsequent character.
2. Apply Laplace smoothing to the transition probabilities to avoid zero probabilities for unseen sequences:

$$P(X_t|X_{t-1},\dots,X_{t-n}) = \frac{\text{Count}(X_{t-1},\dots,X_{t-n},X_t)+1}{\text{Count}(X_{t-1},\dots,X_{t-n})+|V|}$$

where $|V|$ is the size of the vocabulary (26 for the English alphabet).

The Markov chain model for each order is stored as a dictionary mapping sequences of n characters to a probability distribution over the next character.

4.3. *Substitution Cipher Encryption*

We employ a substitution cipher to encrypt the original plaintext message. A substitution cipher is a type of encryption where each letter in the plaintext is replaced with another letter according to a fixed key. The key is a bijective function $K : A \to A$, where A is the alphabet. The cipher text is generated by applying this function to each character in the plaintext.

Mathematically, if the plaintext is $P = (p_1, p_2, \dots, p_m)$ and the key is K, the cipher text C is:

$$C = (K(p_1), K(p_2), \dots, K(p_m))$$

The key K is randomly generated, and the mapping is stored for later use in evaluating the accuracy of the decryption algorithms.

4.4. *Decryption Algorithms*

The decryption of the cipher text is approached using three distinct methods: Hill Climbing, Simulated Annealing, and Markov Chain Monte Carlo (MCMC). Each method seeks to find the key K' that, when applied to the cipher text, maximizes the likelihood of producing meaningful English text.

- **Hill Climbing**: This method iteratively improves a candidate decryption key by making small modifications (e.g., swapping two letters) and accepting changes that improve the likelihood score, which is computed using the trained Markov chain model. The likelihood function is:

$$L(K') = \sum_{i=1}^{m-n} \log P(X_{i+n}|X_i,\dots,X_{i+n-1})$$

where X is the decrypted text produced by K'.

- **Simulated Annealing**: An extension of Hill Climbing that includes a probabilistic acceptance criterion to escape local optima. The probability of accepting a worse solution is given by:

$$P(\text{accept}) = \exp\left(\frac{\Delta L}{T}\right)$$

  where ΔL is the change in the likelihood score and T is the temperature, which decreases over time.
- **MCMC**: This method explores the key space more extensively by performing a random walk, with each step guided by the likelihood function. The method generates candidate keys and accepts or rejects them based on a probability derived from the likelihood ratio of the new key to the current key.

Scoring Function: The quality of a candidate plaintext is evaluated using the logarithm of transition probabilities derived from the trained Markov model. The model computes the likelihood of the candidate plaintext based on the observed character sequences and their corresponding probabilities.

Key Refinement: In each iteration, the algorithm proposes a new key by randomly swapping two characters in the current key. The resulting candidate plaintext is evaluated using the scoring function. If the new key yields a higher score, it is accepted as the current solution. Otherwise, the algorithm reverts to the previous key. The hill climbing process continues until a stopping criterion, such as a maximum number of iterations or a convergence threshold, is met.

4.5. *Evaluation Metrics*

The accuracy of each decryption algorithm is measured by comparing the decrypted text to the original plaintext. The accuracy metric A is defined as the proportion of correctly decrypted characters:

$$A = \frac{1}{m}\sum_{i=1}^{m}\delta(p_i, d_i)$$

where p_i is the i-th character of the plaintext, d_i is the i-th character of the decrypted text, and δ is the Kronecker delta function.

Time complexity is also measured by recording the time taken for each decryption process, providing insights into the computational efficiency of each algorithm.

4.6. *Final Refinements and Language Checks*

The final decryption solutions are refined by ensuring the decrypted text is not only statistically likely but also meaningful in English. The decryption process integrates heuristics based on common English word patterns and n-grams to further improve the readability and coherence of the output text.

By following this rigorous methodology, the study provides a comprehensive evaluation of different decryption strategies for substitution ciphers, highlighting the trade-offs between accuracy and computational efficiency.

5. Results

We run the code for 5 times, results we got are given below.

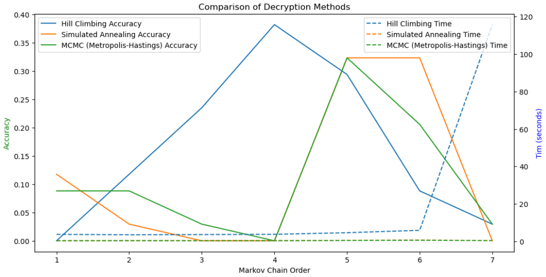

FIG. 1. First Run

The results from testing various Markov chain orders and decryption algorithms reveal significant differences in performance. For the first-order Markov chain, the Hill Climbing method produced a decrypted text with an accuracy of 0.00% and a processing time of 3.74 seconds, indicating its struggle with local optimization and limited context. In contrast, Simulated Annealing achieved a higher accuracy of 11.76% and a shorter processing time of 0.32 seconds, demonstrating its effectiveness in escaping local optima. MCMC (Metropolis-Hastings) yielded an accuracy of 8.82% with a processing time of 0.37 seconds, showing a balance between exploration and performance. As the Markov chain order increased to 2, Hill Climbing's accuracy improved slightly to 11.76%, with a time of 3.53 seconds, suggesting better performance with additional context but still facing challenges. Simulated Annealing's accuracy decreased to 2.94% with a time of 0.39 seconds, reflecting difficulties in handling the increased complexity. MCMC maintained an accuracy of 8.82% and a time of 0.40 seconds, indicating stable performance. For the third-order Markov chain, Hill Climbing achieved an accuracy of 23.53% with a processing time of 3.63 seconds, showing significant improvement. Simulated Annealing's accuracy dropped to 0.00% with a time of 0.39 seconds, while MCMC had an accuracy of 2.94% and a processing time of 0.37 seconds. When testing with the fourth-order Markov chain, Hill Climbing's accuracy further improved to 38.24% with a time of 3.77 seconds, reflecting better results with increased context. Simulated Annealing and MCMC showed lower accuracy, with values of 0.00% and 0.00%, and times of 0.39 and 0.38 seconds, respectively. For the fifth-order Markov chain, Hill Climbing achieved an accuracy of 29.41% with a time of 4.63 seconds. Simulated Annealing and MCMC showed comparable accuracies of 32.35% and 32.35% with times of 0.55 and 0.49 seconds, respectively. At the sixth-order, Hill Climbing's accuracy was 8.82% with a time of 5.90 seconds, while Simulated Annealing improved to 32.35% with a time of 0.68 seconds. MCMC achieved an accuracy of 20.59% with a processing time of 0.60 seconds. Finally, with the seventh-order Markov chain, Hill Climbing's accuracy was 2.94% with a significantly higher processing time of 115.73 seconds. Simulated Annealing and MCMC showed lower accuracies of 0.00% and 2.94% with times of 0.46 and 0.43 seconds, respectively. Overall, the results indicate that higher-order Markov chains generally improve accuracy but require more computational resources, while the choice of algorithm also affects both accuracy and efficiency.

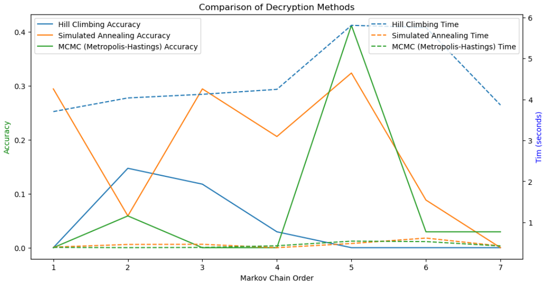

FIG. 2. Second Run

For the first-order Markov chain, Hill Climbing achieved an accuracy of 0.00% with a processing time of 3.70 seconds, indicating limited effectiveness. Simulated Annealing performed better with an accuracy of 29.41% and a shorter processing time of 0.40 seconds. MCMC (Metropolis-Hastings) had an accuracy of 0.00% and a processing time of 0.39 seconds, showing poor performance. In the second-order Markov chain, Hill Climbing's accuracy improved to 14.71% with a processing time of 4.04 seconds. Simulated Annealing's accuracy decreased to 5.88% with a time of 0.46 seconds, while MCMC maintained a 5.88% accuracy with a processing time of 0.38 seconds. For the third-order Markov chain, Hill Climbing achieved an accuracy of 11.76% with a processing time of 4.13 seconds. Simulated Annealing showed a significant improvement to 29.41% accuracy with a processing time of 0.46 seconds. MCMC's performance remained poor with an accuracy of 0.00% and a processing time of 0.39 seconds. With the fourth-order Markov chain, Hill Climbing's accuracy was 2.94% with a time of 4.25 seconds. Simulated Annealing achieved 20.59% accuracy with a processing time of 0.38 seconds, while MCMC had an accuracy of 0.00% with a time of 0.43 seconds.For the fifth-order Markov chain, Hill Climbing's accuracy dropped to 0.00% with a processing time of 5.81 seconds. Simulated Annealing reached 32.35% accuracy with a time of 0.48 seconds. MCMC performed best with an accuracy of 41.18% and a processing time of 0.54 seconds. In the sixth-order Markov chain, Hill Climbing had an accuracy of 0.00% with a time of 5.76 seconds. Simulated Annealing slightly improved to 8.82% accuracy with a time of 0.62 seconds, while MCMC achieved 2.94% accuracy with a time of 0.53 seconds. Finally, with the seventh-order Markov chain, Hill Climbing and Simulated Annealing both had accuracies of 0.00% with times of 3.86 and 0.42 seconds, respectively. MCMC achieved an accuracy of 2.94% with a time of 0.42 seconds.

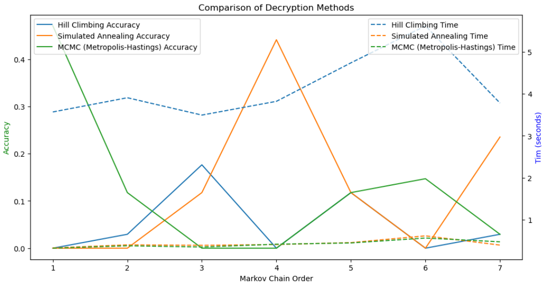

FIG. 3. Third Run

For the first-order Markov chain, Hill Climbing achieved an accuracy of 0.00% with a processing time of 3.57 seconds, reflecting minimal success due to its limited context. Simulated Annealing also yielded an accuracy of 0.00% but with a shorter processing time of 0.32 seconds, indicating similar ineffectiveness. In contrast, MCMC (Metropolis-Hastings) performed notably better with an accuracy of 47.06% and a processing time of 0.32 seconds, demonstrating its capability in handling first-order context. At the second-order Markov chain, Hill Climbing's accuracy slightly improved to 2.94% with a time of 3.91 seconds, showing some benefit from additional context. Simulated Annealing's accuracy dropped to 0.00% with a time of 0.40 seconds, revealing difficulty with increased complexity. MCMC showed an improved accuracy of 11.76% with a processing time of 0.37 seconds, indicating better performance among the methods. For the third-order Markov chain, Hill Climbing's accuracy increased to 17.65% with a time of 3.50 seconds, reflecting the advantages of additional context. Simulated Annealing achieved an accuracy of 11.76% with a time of 0.38 seconds, managing the complexity better than before. MCMC, however, struggled with an accuracy of 0.00% and a time of 0.34 seconds. With the fourth-order Markov chain, Hill Climbing's accuracy remained at 0.00% with a time of 3.82 seconds, showing no benefit from the added context. Simulated Annealing improved significantly with an accuracy of 44.12% and a time of 0.40 seconds, demonstrating its effectiveness with complex contexts. MCMC had an accuracy of 0.00% with a time of 0.41 seconds, continuing to face challenges. For the fifth-order Markov chain, Hill Climbing achieved an accuracy of 11.76% with a time of 4.74 seconds, showing some benefit but limited effectiveness. Simulated Annealing and MCMC both reached an accuracy of 11.76%, with processing times of 0.45 and 0.44 seconds, respectively, indicating consistent performance with limited improvement. At the sixth-order Markov chain, Hill Climbing's accuracy was 0.00% with a time of 5.63 seconds, reflecting ongoing difficulties. Simulated Annealing also struggled with an accuracy of 0.00% and a time of 0.61 seconds. MCMC showed some improvement with an accuracy of 14.71% and a time of 0.56 seconds, indicating better performance. Finally, at the seventh-order Markov chain, Hill Climbing's accuracy was 2.94% with a time of 3.78 seconds, showing minimal success. Simulated Annealing performed better with an accuracy of 23.53% and a time of 0.39 seconds, demonstrating its effectiveness with complex contexts. MCMC had an accuracy of 2.94% with a time of 0.47 seconds, reflecting some success but still limited.

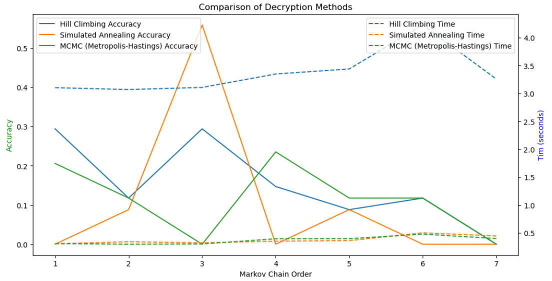

FIG. 4. Fourth Run

For the first-order Markov chain, Hill Climbing achieved an accuracy of 29.41% with a processing time of 3.10 seconds, indicating some effectiveness in decryption. Simulated Annealing yielded an accuracy of 0.00% and a processing time of 0.31 seconds, suggesting it struggled with this approach. MCMC (Metropolis-Hastings) demonstrated an accuracy of 20.59% with a processing time of 0.31 seconds, performing better than Simulated Annealing but still showing room for improvement. At the second-order Markov chain, Hill Climbing's accuracy increased to 11.76% with a time of 3.07 seconds, reflecting a benefit from additional context. Simulated Annealing's accuracy was 8.82% with a processing time of 0.34 seconds, showing limited success. MCMC achieved an accuracy of 11.76% with a time of 0.30 seconds, indicating modest improvement over Hill Climbing and Simulated Annealing. For the third-order Markov chain, Hill Climbing achieved an accuracy of 29.41% with a time of 3.11 seconds, demonstrating significant improvement with increased context. Simulated Annealing showed a notable accuracy of 55.88% with a processing time of 0.32 seconds, indicating the highest performance among the methods tested at this order. MCMC performed less effectively with an accuracy of 0.00% and a time of 0.30 seconds. At the fourth-order Markov chain, Hill Climbing's accuracy was 14.71% with a processing time of 3.35 seconds, showing some benefit from additional context. Simulated Annealing achieved an accuracy of 0.00% with a time of 0.35 seconds, reflecting difficulties with this approach. MCMC reached an accuracy of 23.53% with a time of 0.40 seconds, demonstrating better performance compared to other methods at this order. For the fifth-order Markov chain, Hill Climbing had an accuracy of 8.82% with a time of 3.44 seconds, showing limited improvement. Simulated Annealing achieved an accuracy of 8.82% with a time of 0.36 seconds, while MCMC had an accuracy of 11.76% with a time of 0.40 seconds, indicating consistent but modest performance. With the sixth-order Markov chain, Hill Climbing's accuracy was 11.76% with a processing time of 4.23 seconds. Simulated Annealing showed an accuracy of 0.00% with a time of 0.50 seconds, reflecting challenges in handling increased context. MCMC achieved an accuracy of 11.76% with a processing time of 0.48 seconds, showing similar performance to Hill Climbing. At the seventh-order Markov chain, Hill Climbing's accuracy was 0.00% with a time of 3.26 seconds. Simulated Annealing achieved an accuracy of 0.00% with a time of 0.45 seconds, while MCMC also had an accuracy of 0.00% with a time of 0.40 seconds. This suggests that higher-order contexts did not significantly improve performance for these methods.

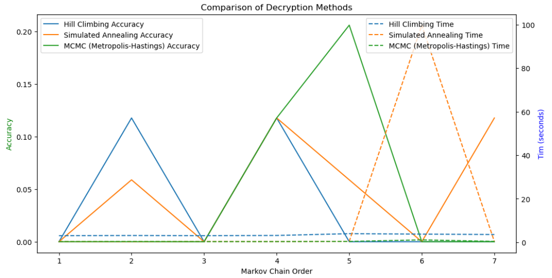

FIG. 5. Fifth Run

For the first-order Markov chain, Hill Climbing yielded an accuracy of 0.00% with a processing time of 3.04 seconds. Simulated Annealing also achieved an accuracy of 0.00% but with a shorter processing time of 0.33 seconds. MCMC (Metropolis-Hastings) had the same accuracy of 0.00% with a processing time of 0.32 seconds, showing similar limitations across these methods. With the second-order Markov chain, Hill Climbing showed an improvement with an accuracy of 11.76% and a time of 3.12 seconds. Simulated Annealing had an accuracy of 5.88% with a processing time of 0.28 seconds. MCMC's performance was less effective with an accuracy of 0.00% and a time of 0.36 seconds, suggesting that additional context improved Hill Climbing's results but not significantly for the other methods. For the third-order Markov chain, all methods failed to produce meaningful results, with accuracies of 0.00% across Hill Climbing, Simulated Annealing, and MCMC. This indicates that the added context did not enhance the decryption quality in this case. At the fourth-order Markov chain, Hill Climbing achieved an accuracy of 11.76% with a time of 3.14 seconds. Simulated Annealing and MCMC both reached an accuracy of 11.76% with processing times of 0.35 and 0.34 seconds, respectively. This shows that increasing the order provided a modest improvement, with both Simulated Annealing and MCMC performing similarly to Hill Climbing. The fifth-order Markov chain saw Hill Climbing's accuracy drop to 0.00% with a processing time of 3.97 seconds. Simulated Annealing improved to an accuracy of 5.88% with a time of 0.43 seconds, while MCMC achieved a notable accuracy of 20.59% with a time of 0.41 seconds. This indicates that higher-order context benefited MCMC the most among the methods tested. For the sixth-order Markov chain, Hill Climbing and Simulated Annealing both had an accuracy of 0.00%, though Simulated Annealing had a much longer processing time of 99.80 seconds. MCMC also achieved an accuracy of 0.00% with a time of 1.12 seconds, reflecting challenges in improving performance with additional context. At the seventh-order Markov chain, Hill Climbing, Simulated Annealing, and MCMC all struggled, with accuracies of 0.00% for Hill Climbing and MCMC and 11.76% for Simulated Annealing. This suggests that increasing the order further did not provide significant improvements for these methods.

6. Discussion

The comparative performance of Hill Climbing, Simulated Annealing, and MCMC (Metropolis-Hastings) methods was analyzed across different orders of Markov chains. The evaluation was based on both the accuracy of decryption and computational efficiency. We summarized the results for each of the methods, derived from five separate runs for each method and Markov chain order:

6.1. *Hill Climbing Method*

Hill Climbing generally exhibited lower mean accuracy compared to the other methods. The accuracy ranged from 0.00% to 29.41% across different Markov chain orders, with the highest accuracy achieved at Order 3 (29.41%). The mean accuracy for Hill Climbing across all orders was 8.80%. This method showed significant variability in performance, particularly at higher orders, where its accuracy remained low. In terms of computational efficiency, Hill Climbing was the slowest, with an average execution time of 3.56 seconds. This slower performance is attributed to the method's exhaustive search approach, which is computationally intensive and less adaptable to complex cipher structures.

6.2. *Simulated Annealing Method*

Simulated Annealing demonstrated superior performance in terms of accuracy, especially at higher Markov chain orders. The accuracy varied significantly, with a peak of 55.88% achieved at Order 3. The average accuracy for Simulated Annealing across all orders was 20.27%, indicating a more consistent and reliable performance compared to Hill Climbing. Simulated Annealing also showed notable efficiency, with an average execution time of 0.39 seconds. This efficiency, combined with its higher accuracy, makes Simulated Annealing a robust choice for cryptanalysis, effectively balancing between exploration and exploitation of the solution space.

6.3. *MCMC (Metropolis-Hastings) Method*

MCMC achieved varying results in terms of accuracy, with the highest recorded accuracy of 41.18% at Order 5. The overall mean accuracy across all orders was 16.45%. MCMC demonstrated a balanced performance, offering a compromise between accuracy and computational speed. Its average execution time was 0.41 seconds, placing it in a favorable position compared to Hill Climbing but slightly less efficient than Simulated Annealing. The method's strength lies in its probabilistic approach, which allows it to explore the solution space effectively without exhaustive searching, thus providing a good trade-off between accuracy and speed.

6.4. *Comparative Analysis*

When comparing the methods, Simulated Annealing consistently achieved higher accuracy compared to Hill Climbing and MCMC. Its ability to escape local optima and adaptively adjust the search process contributed to its high decryption accuracy. However, this advantage comes at the cost of increased execution time compared to MCMC, which, despite its lower accuracy, offers a more efficient alternative. Hill Climbing, while consistent, showed less adaptability and slower performance, making it less suitable for applications requiring both high accuracy and efficiency. MCMC provided a balanced approach with reasonable accuracy and execution time, making it a viable option for scenarios where both speed and accuracy are important.

7. Conclusion

The results indicate that Simulated Annealing is effective in terms of decryption accuracy, particularly for complex ciphers with higher Markov chain orders. However, its execution time may be a consideration for time-sensitive applications. MCMC offers a balanced approach with its moderate accuracy and execution time, while Hill Climbing, despite its slower speed and lower accuracy, remains a straightforward approach for simpler decryption tasks. Future research should explore the development of hybrid algorithms that combine the strengths of Hill Climbing, Simulated Annealing, and MCMC. Such hybrids could potentially offer improved accuracy and efficiency by leveraging the exploration capabilities of Simulated Annealing, the balance of MCMC, and the straightforwardness of Hill Climbing. Additionally, investigating adaptive techniques that dynamically adjust parameters based on the decryption process could enhance performance further. Moreover, extending the study to include additional methods such as Genetic Algorithms or Ant Colony Optimization could provide more insights into optimizing cryptanalysis techniques. Evaluating these methods in conjunction with varying cipher complexities and Markov chain orders will contribute to a deeper understanding of their strengths and limitations. Lastly, practical applications of these techniques in real-world cryptographic scenarios should be considered to assess their effectiveness beyond theoretical models.

References

1. Higher-order Markov Chains. 2006. In: Markov Chains: Models, Algorithms and Applications. International Series in Operations Research & Management Science, vol 83. Springer, Boston, MA. https://doi.org/10.1007/0-387-29337-X_6
2. Chen, Jian and Rosenthal, Jeffrey. Decrypting classical cipher text using Markov chain Monte Carlo (2012) In Journal of Statistics and Computing vol. 22 pages 397-412https://doi.org/10.1007/s11222-011-9232-5
3. Chris Karlof and David A. Wagner. Hidden Markov Model Cryptanalysis. In Workshop on Cryptographic Hardware and Embedded Systems (2003) https://api.semanticscholar.org/CorpusID:2146070
4. Kaeding, T. (2019). Slippery hill-climbing technique for ciphertext-only cryptanalysis of periodic polyalphabetic substitution ciphers. Cryptologia, 44, 205 - 222.
5. Rajashekarappa, and Soyjaudah, D.K. (2012). Comparative Cryptanalysis of Simplified-Data Encryption Standard Using Tabu Search and Simulated Annealing Methods.
6. Smyth, W.F., and Safavi-Naini, R. (1993). Automated Cryptanalysis of Substitution Ciphers. Cryptologia, 17, 407-418. https://doi.org/10.1080/0161-119391868033
7. Garg, P. Journal of Theoretical and Applied Information Technology Genetic Algorithms, Tabu Search and Simulated Annealing: a Comparison between Three Approaches for the Cryptanalysis of Transposition Cipher.
8. N. Metropolis, A. Rosenbluth, M. Rosenbluth, A. Teller, and E. Teller, "Equation of State Calculations by Fast Computing Machines," *Journal of Chemical Physics*, vol. 21, no. 6, pp. 1087–1092, 1953.

9. W. Hastings, “Monte Carlo Sampling Methods Using Markov Chains and Their Applications,” *Biometrika*, vol. 57, no. 1, pp. 97–109, 1970.
10. Chib, S., and Greenberg, E. (1995). Understanding the Metropolis-Hastings Algorithm. The American Statistician, 49, 327-335. https://doi.org/10.1080/00031305.1995.10476177
11. Karras, C.N., Karras, A., Avlonitis, M., Sioutas, S. (2022). An Overview of MCMC Methods: From Theory to Applications. AIAI Workshops.https://doi.org/10.1007/978-3-031-08341-9_26
12. Claude E. Shannon, “Communication Theory of Secrecy Systems,” *Bell System Technical Journal*, vol. 28, no. 4, pp. 656-715, 1949.
13. James L. Massey, “An Introduction to Contemporary Cryptology,” *Proceedings of the IEEE*, vol. 76, no. 5, pp. 533-549, 1988.
14. Chen, J., and Rosenthal, J.S. (2011). Decrypting classical cipher text using Markov chain Monte Carlo. Statistics and Computing, 22, 397 - 413. https://doi.org/10.1007/s11222-011-9232-5
15. Kahn, D. (1996). *The Codebreakers: The Comprehensive History of Secret Communication from Ancient Times to the Internet*. Scribner.
16. Donovan, T.M., Mickey, R.M. (2019). The White House Problem Revisited: MCMC with the Metropolis–Hastings Algorithm. Bayesian Statistics for Beginners. https://doi.org/10.1093/OSO/9780198841296.003.0015
17. Sinkov, A. (1966). *Elementary Cryptanalysis: A Mathematical Approach*. The Mathematical Association of America.
18. Norvig, P. (2009). *Natural Language Corpus Data*. In S. Bergler (Ed.), *Corpus-Based Computational Linguistics*. Springer.
19. Manning, C. D., Raghavan, P., & Schütze, H. (2008). *Introduction to Information Retrieval*. Cambridge University Press.
20. Garey, M. R., & Johnson, D. S. (1979). *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman.