Performance of Metropolis Algorithm for the Minimum Weight Code Word Problem

Ajitha Shenoy K B
Department of Computer Science and Engineering
Indian Institute of Technology Kanpur, India.
ajith@cse.iitk.ac.in

Somenath Biswas
Department of Computer Science and Engineering
Indian Institute of Technology Kanpur, India.
sb@cse.iitk.ac.in

Piyush P Kurur
Department of Computer Science and Engineering
Indian Institute of Technology Kanpur, India.
ppk@cse.iitk.ac.in

ABSTRACT

We study the performance of the Metropolis algorithm for the problem of finding a code word of weight less than or equal to M, given a generator matrix of an $[n, k]$-binary linear code. The algorithm uses the set $\mathcal{S}_k$ of all $k \times k$ invertible matrices as its search space where two elements are considered adjacent if one can be obtained from the other via an elementary row operation (i.e by adding one row to another or by swapping two rows.) We prove that the Markov chains associated with the Metropolis algorithm mix rapidly for suitable choices of the temperature parameter T. We ran the Metropolis algorithm for a number of codes and found that the algorithm performed very well in comparison to previously known experimental results.

Categories and Subject Descriptors

G.1.6 [**Optimization**]: Simulated annealing; G.3 [**Probability and Statistics**]: Probabilistic algorithms (including Monte Carlo); H.1.1 [**Systems & Information Theory**]: Information Theory

General Terms

Theory, Algorithms

Keywords

Metropolis Algorithm; Minimum weight code word; search space; conductance; rapid mixing of Markov chain

1. INTRODUCTION

An $[n, k]$ binary linear code is a k-dimensional subspace of the vector space of n-dimensional binary vectors, its code words are the elements of the subspace, and a minimum weight code word of such a code is a non-zero code word with minimum number of 1's. (We provide the formal definitions in the next Section). Such a code can be succinctly presented by providing a basis of the k-dimensional subspace. The minimum weight code word problem requires one to find a minimum weight code word, given a basis of the code. This problem is important for several reasons: a minimum weight code word is a measure of the error correction capability of the code [9], also, codes with large minimum weight code words have applications in diverse areas such as cryptography [18, 17, 16], pseudorandom generators [1, 11].

The problem has been shown to be NP-hard [6], moreover, it remains hard even to obtain a constant factor approximation [10]. It is for this reason that researchers have proposed several probabilistic and heuristic algorithms to find low weight code words, given a basis of a binary linear code. Examples are: GA [4, 13, 5], hill climbing [4, 13, 5], tabu search [8], and ACO [7].

We study in this paper the efficacy of the Metropolis algorithm for solving the problem. Our Metropolis algorithm for an $[n, k]$ code uses the set of all $k \times k$ invertible binary matrices as its search space set. Two such elements are considered neighbours if one can be obtained from the other by an elementary row operation. We prove that the search space graph has large magnification and use this result to prove that the family of Markov chains, as defined by the Metropolis algorithm on an input instance, has a large conductance. It is known that [21] the Metropolis algorithm solves a combinatorial optimization problem like ours in polynomial time if and only if (1) the associated Markov chain family has high conductance (2) the probability of the favorable event in the stationary distribution is high. As our Metropolis algorithm tries to find a code word of weight less than or equal to M, where M is given as an input parameter, the algorithm will be efficient, in view of the conductance result, if the probability p_M of getting a code word of weight M is high in the stationary distribution. A good bound for p_M is difficult to estimate, this quantity closely related to the weight distribution of the binary linear codes. Therefore, to understand how well the Metropolis algorithm works for this problem, we experiment with a few codes. The codes that we use for our experiments are certain BCH codes and full dimensional codes of dimensions 50 and 100. We found that the Metropolis algorithm performed well for several BCH (Bose, Chaudhuri and Hocquenghem) codes [20], even obtaining the minimum in certain cases. For the full dimensional case, where the minimum weight is 1, the algorithm was able to converge quickly to a small weight code word. This compares favorably with previously known experimen-

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
GECCO'14, July 12–16, 2014, Vancouver, BC, Canada.
Copyright 2014 ACM 978-1-4503-2662-9/14/07 ...$15.00.
http://dx.doi.org/10.1145/2576768.2598274 .

tal results on BCH codes which used certain other search heuristics. Details are given in Section 5.

2. PRELIMINARIES

DEFINITION 2.1. ***(Binary Linear Codes)*** *An* $[n, k]$-binary linear code *is a* k*-dimensional vector subspace* C *of the* n*-dimensional vector space* $\mathbf{F}_2^n$ *over the finite field* $\mathbb{F}_2$*. The parameter* n *is called the* length *and* k *the* dimension *of the code* C*.*

A binary linear $[n, k]$-code C can be succinctly described by giving a basis for it. This is typically done by giving a *generator matrix* for the code C.

DEFINITION 2.2. ***(Generator Matrix)*** *A generator matrix for an* $[n, k]$ *linear code* C *is a* $k \times n$ *matrix* G *whose rows form a basis for* C*.*

This is called a generator matrix because a vector is a code word if and only if it is a linear combination of the rows of the generator matrix.

DEFINITION 2.3. ***(Elementary row operations)*** *For a* $k \times n$ *matrix* G *over the field* $\mathbb{F}_2$ *with rows* $\mathbf{g}_r$*,* $1 \leq r \leq k$*, the following are the elementary row operations for distinct* i *and* j*.*

1. $\mathbf{g}_i \leftarrow \mathbf{g}_i + \mathbf{g}_j$ *(the* j*-th row is added to the* i*-th row),*
2. $\mathbf{g}_i \leftrightarrow \mathbf{g}_j$ *(rows* i *and* j *are interchanged).*

DEFINITION 2.4. **(Minimum weight code word)** *For a code* C*, a non-zero vector* $\mathbf{v} \in C$ *of minimum Hamming weight is called a minimum weight code word. The* minimum weight code word *problem is to compute, given a generator matrix* G *of an* $[n, k]$*-code* C*, a minimum weight code word of* C*.*

The decision version of the minimum weight code word problem can be stated as follows.

DEFINITION 2.5. ***(Decision version)*** *Given* G*, a generator matrix for* C *and an integer* M*,* decide *whether there exists a non-zero vector in* C *of weight* M *or less.*

3. SEARCH SPACE

In this section, we define the search space which we use in our Metropolis algorithm. Our search space is similar to the search space defined for shortest lattice vector problem [2, 3]. The algorithm attempts to construct a generator matrix which has a minimum weight code word as one of its rows. It is easy to see that such a generator matrix always exists.

FACT 3.1. *[12] Let* G *be a generator matrix of the* $[n, k]$ *binary linear code* C*, then* G' *is a generator matrix for* C *if and only if there is a* $k \times k$ *invertible matrix* U *such that* $G' = UG$*.*

Consequently, for any input generator matrix G, the set of all matrices UG as U varies over $k \times k$ invertible matrices is precisely the set of all generator matrices of C. Therefore, the search space of our Metropolis algorithm will be the set $\mathcal{S}_k$ of all $k \times k$ invertible matrices and given a generator matrix G of C as input, our goal is to find a matrix U in $\mathcal{S}_k$ such that UG contains a minimum weight code word as one of its rows. A natural neighbourhood structure on the search spaces $\mathcal{S}_k$ can be defined in terms of elementary row operations on matrices in $\mathcal{S}_k$. We now formally define our search space.

DEFINITION 3.2. **(Search Space)** *Given an* $[n, k]$*-code* C *via a generator matrix* G *we define the search space as follows.*

Elements: *The elements of the search space* $\mathcal{S}_k$ *are the* $k \times k$ *invertible matrices* U *over* $GF(2)$*.*

Neighbourhood: *For a matrix* U *in* $\mathcal{S}_k$*, the set* $N(U)$ *of neighbours consists of matrices* V *that can be obtained from* U *by any of the following elementary operations:*

1. *add the* j*-th row to the* i*-th row,*
2. *swap the* i*-th and* j*-th rows,*

where i *and* j *are two distinct rows of* U*. This makes the underlying graph a* D*-regular graph where* D *is* $k(k-1)$*.*

Cost: *For an element* U *of the search space, the cost* $c(U)$ *is defined to be* w^α *where* w *is the minimum of the Hamming weights of rows of* UG *and* α *is a parameter which takes positive values.*

Convention: The elementary row operations that we defined above can be carried out on a matrix A by post multiplying by a suitable elementary matrix say E. We often identify a row operation with the corresponding elementary matrix. Note that $|\mathcal{S}_k|$ equal to number of binary $k \times k$ invertible matrices which is less than 2^{k^2}, the total number of $k \times k$ binary matrices.

We now show that the diameter of the search space graph is bounded by a polynomial in k.

PROPOSITION 3.3. *For any two matrices* U *and* V *in the search space* $\mathcal{S}_k$ *there is a path between them in the search space graph of length at most* k^2*.*

PROOF. The matrix $V^{-1}U$ is invertible as both U and V are. Therefore, $V^{-1}U$ can be transformed to the $k \times k$ identity matrix I_k by Gauss-Jordan elimination which uses a sequence of elementary row operations. This is done in k-stages one for each column. In the i-th stage, we transform the i-th column to the vector $\mathbf{e}_i$ which has a 1 at the i-th position and 0's elsewhere. Each of these stages can be carried out using at most k elementary row operations (more details in the proof of Lemma 3.4) and thus using $\ell \leq k^2$ elementary operations, $V^{-1}U$ gets transformed into I_k. Let $E_1, \ldots, E_\ell$ be the elementary matrices associated with the elementary row operations described above. Consequently, the product $V^{-1}UE_1 \cdots E_\ell$ is the identity matrix I_k. Therefore, V is the product $UE_1 \cdots E_\ell$. The sequence of matrices $U_i = U \cdot \prod_{j=1}^{i} E_i$ for $0 \leq i \leq \ell$ gives the path from matrix $U = U_0$ to $V = U_\ell$. $\square$

We prove next that our search space graph has large magnification. This will be used later to show that the family of Markov chains as defined by our Metropolis algorithm mixes rapidly. We now recall the definition of magnification [23]. Let $G = (V, E)$ be an undirected graph. Let S be a non-empty subset of V and let $\overline{S}$ denote its complement,

i.e. $V \setminus S$. Let $E(S, \overline{S})$ denote the set of edges that go out of S. The magnification $\mu(S)$ is defined as

$$\mu(S) = \frac{|E(S, \overline{S})|}{|S|}.$$

The magnification for the graph G (also called as *edge expansion*), denoted by $\mu(G)$, is the minimum $\mu(S)$ where the minimization is done over all non-empty subsets S of V of cardinality at most $\frac{|V|}{2}$.

LEMMA 3.4. *A search space graph for minimum weight code word problem has magnification at least* $\frac{1}{2}$.

PROOF. We use the *canonical path method* [22] to lower-bound the magnification of the search graph. For this, we first canonize the Gauss-Jordan elimination procedure described in the proof of Proposition 3.3 that transforms an arbitrary $k \times k$ invertible matrix A to identity matrix. The procedure works in k-stages. The i-th stage starts with a matrix A_{i-1} whose r-th column, for any $r < i$ is the vector $\mathbf{e}_r$, the vector which has a 1 at r-th entry and 0 every where else. In the i-th stage, we convert the i-th column into $\mathbf{e}_i$ using elementary row operations. The stage begins with a swap of i-th row if and only if the (i, i)-th entry of A_{i-1} is a 0. In such a case, we choose to swap the i-th row and the j-th row, where j is the smallest integer greater than i such that the (j, i)-th entry is 1. There is always one such j because the matrix A_{i-1} is invertible. Having ensured that the (i, i)-th entry is 1, we convert each 1 in the i-th column, except the (i, i)-th entry to 0 by adding the i-th row. This gives us the matrix A_i. The elimination process ends when i is k and the resulting matrix A_k becomes identity. Let us call the sequence $E_1, \ldots, E_\ell$ of elementary operations used to reach the identity matrix from A in the above process as the *canonical Gauss-Jordan sequence* associated with the matrix A.

We show that canonical Gauss-Jordan sequences satisfy the following properties.

CLAIM 3.4.1. *1. There is a unique canonical Gauss-Jordan sequence associated with a given $k \times k$ invertible matrix-A. Moreover, two distinct matrices A and A' in $\mathcal{S}_k$ will have distinct canonical Gauss-Jordan sequences.*

2. The number of distinct canonical Gauss-Jordan sequences is equal to the cardinality of the search spaces $\mathcal{S}_k$.

3. If $E_1, \ldots, E_\ell$ is the canonical Gauss-Jordan sequence of some $k \times k$ matrix-A, then no two operators E_r and E_s in the sequence are the same.

PROOF. The way in which we have canonize the Gauss-Jordan elimination procedure above, it is clear that there is a unique canonical Gauss-Jordan sequence associated with a given $k \times k$ invertible matrix A. Now suppose that two distinct matrices A and A' in $\mathcal{S}_k$ have the same canonical Gauss-Jordan sequence, i.e $AE_1 \ldots E_\ell = I$ and $A'E_1 \ldots E_\ell = I$ which implies $A = A' = IE_\ell^{-1} \ldots E_1^{-1}$ which is a contradiction. Hence two distinct invertible matrices A and A' will have distinct canonical Gauss-Jordan sequences. This completes the proof of part 1 of our claim. To prove part 2 notice that the number of distinct canonical Gauss-Jordan sequences is equal to the number of invertible matrices in $\mathcal{S}_k$ which is equal to $|\mathcal{S}_k|$. Finally to prove part 3, consider any elementary row operation that occurs in the canonical Gauss-Jordan sequence. Either it is a row addition or a swap of two rows. It is clear that all the row additions are distinct as we add the row i only in the i-th stage of the procedure and that too to distinct rows in the i-th stage. A swap is used only to convert a 0-diagonal entry to 1. Consider such swap between rows i and j where $j > i$. This swap can happen only at the i-th stage and not at the j-th stage because to convert a 0 in the diagonal we use a row that is lower down in the matrix. This completes the proof of part 3. □

We now define the canonical path between two search graph elements U and V as follows: Let $E_1, \ldots, E_\ell$ be the canonical Gauss-Jordan sequence associated with the matrix $V^{-1}U$. Then the canonical path from U to V is the sequence $U = U_0, \ldots, U_\ell = V$ where $U_i = U \cdot \prod_{j=1}^{i} E_i$. Given U and V, by a slight abuse of notation, the canonical Gauss-Jordan sequence associated with $V^{-1}U$ is also called the *canonical Gauss-Jordan sequence associated with the canonical path*[1] from U to V.

Fix two neighbours C and D in the search space $\mathcal{S}_k$. We now estimate the number of canonical paths that go through the edge (C, D).

CLAIM 3.4.2. *For two neighbours C and D in $\mathcal{S}_k$ consider any canonical Gauss-Jordan sequence $E_1, \ldots, E_\ell$ containing the matrix $C^{-1}D$. There is a unique canonical path through the edge (C, D) that has $E_1, \ldots, E_\ell$ as its associated canonical Gauss-Jordan sequence.*

PROOF. Consider any canonical Gauss-Jordan sequence $E_1, \ldots, E_\ell$ such that $E_r = C^{-1}D$ for some $1 \leq r \leq \ell$. By Claim 3.4.1 all operators E_s, $s \neq r$, are different from $C^{-1}D$. Therefore, the only canonical path that passes through the edge (C, D) and has $E_1, \ldots, E_\ell$ as its associated canonical Gauss-Jordan sequence starts at $U = CE_{r-1}^{-1} \ldots E_1^{-1}$ and ends at $V = DE_{r+1} \ldots E_\ell$. □

We have the following consequence of the above claim.

CLAIM 3.4.3. *Let C and D be neighbours in the search space $\mathcal{S}_k$. Then the number of canonical paths passing through the edge (C, D) is bounded by the total number of points in the search space $\mathcal{S}_k$.*

This is because such a canonical path is uniquely determined by its canonical Gauss-Jordan sequence. The number of distinct Gauss-Jordan sequences is equal to the number of $k \times k$ invertible matrices which is the number of elements of the search space.

We now prove the bound on the magnification as given in the statement of the Lemma 3.4. Consider any non-empty subset S such that $|S| \leq \frac{|\mathcal{S}_k|}{2}$. There are $|S| \times |\overline{S}|$ canonical paths that go from S to $\overline{S}$. Each of these paths passes through one of the edges in $E(S, \overline{S})$. As no edge can have more than $|\mathcal{S}_k|$ canonical paths passing through it by Claim 3.4.3, we have $|\mathcal{S}_k| \times |E(S, \overline{S})| \geq |S| \times |\overline{S}|$. As $|\overline{S}| \geq \frac{|\mathcal{S}_k|}{2}$ we have $|\mathcal{S}_k| \times |E(S, \overline{S})| \geq |S| \times \frac{|\mathcal{S}_k|}{2}$. Therefore, the magnification $\mu(S) = \frac{|E(S, \overline{S})|}{|S|}$ is greater than 1/2 for all

[1] We note that two different canonical paths may have the same associated Gauss-Jordan sequences: The canonical paths from U to V and from U' to V' will have the same associated canonical Gauss-Jordan sequence if and only if $U' = AU$ and $V' = AV$ for some invertible $k \times k$ matrix A

S of cardinality at most $|\mathcal{S}_k|/2$. Since magnification μ_k of the search space $\mathcal{S}_k$ is the minimum over all such $\mu(S)$'s, we have $\mu_k \geq \frac{1}{2}$. This completes the proof of Lemma 3.4.

4. MIXING TIME ANALYSIS

We use the Metropolis algorithm for the minimum weight code word problem. On a given input instance G, the Metropolis algorithm runs a Markov chain: the state space of the chain is the set $\mathcal{S}_k$ of $k \times k$ invertible matrices, which is the search space of our problem. Recall that the cost $c(U)$ associated with a search space element U, a matrix, is w^α where w is the minimum Hamming weight of the rows of UG. The Markov chain makes use of this cost function to define a random walk biased towards code words of lower weights. We now define the transition probabilities p_{UV}, the probability of making a transition to V given that the chain is at U.

$$p_{UV} = \begin{cases} 0 & \text{if } U \neq V, V \notin N(U) \\ \frac{1}{2D} \cdot \min\left(1, \exp\left(\frac{c(U)-c(V)}{T}\right)\right) & \text{if } V \in N(U) \\ 1 - \sum_{W \neq U} p_{UW} & \text{if } U = V. \end{cases}$$

In the above definition T stands for the temperature parameter, which remains fixed for the algorithm, and D is the degree of the underlying regular graph $\mathcal{S}_k$. Recall that D is $k(k-1)$. It is well known [19, Chapter: 10.4.1] that the above Markov chain has the stationary distribution given by

$$\pi_U = \frac{\exp\left(\frac{-c(U)}{T}\right)}{\sum_{V \in \mathcal{S}_k} \exp\left(\frac{-c(V)}{T}\right)}.$$

The complete algorithm (Algorithm 1) is given below:

Algorithm 1 Metropolis Algorithm

Input : The generator matrix G of linear code C and an integer M
Output : Matrix U such that UG contains a vector $\mathbf{v}$ with $w_H(\mathbf{v}) \leq M$.
Let U be the starting state in the search space as in Definition 3.2 and $c(U)$ denote its cost.
Set $BestWeight = c(U)$, steps=0
while $BestWeight > M$ and $steps < TSteps$
[$TSteps$ denotes Max. No. of steps specified by user]
do
  Select any one of the neighbour U of V uniformly at random by performing one of the elementary operations as defined in Definition 3.2
  Set $U = V$ with probability

$$\alpha = \frac{1}{2}.\min\left(\frac{\exp\left(\frac{-c(V)}{T}\right)}{\exp\left(\frac{-c(U)}{T}\right)}, 1\right)$$

  if $BestWeight > c(U)$ **then**
    $BestWeight = c(U)$
  end if
  $steps = steps + 1$;
end while

We now show that the above Markov chain has large conductance for an appropriate choice of the temperature parameter T.

Definition 4.1. **(Conductance)[23]** *For any non empty subset S of states in $\mathcal{S}_k$ with non empty complement $\overline{S}$, the* conductance $\phi(S)$ *of S is defined as*

$$\phi(S) = \frac{F_S}{C_S}$$

where

$$C_S = \sum_{U \in S} \pi_U$$

$$F_S = \sum_{U \in S, V \in \overline{S}} p_{UV} \pi_U$$

The conductance ϕ_k *of the Markov chain is defined to be*

$$\phi_k = \min_{S: C_S \leq \frac{1}{2}} \phi(S)$$

It is easy to see that $F_S = F_{\overline{S}}$ for all such sets S. This implies that $\phi(\overline{S}) = \phi(S)\frac{C_S}{1-C_S}$ (since $C_S + C_{\overline{S}} = \sum_{U \in \mathcal{S}_k} \pi_U = 1$ which implies $C_{\overline{S}} = 1 - C_S$), so we may equivalently write

$$\phi_k = \min_S \{\max(\phi(S), \phi(\overline{S}))\}$$

Theorem 4.2. *The conductance ϕ_k of the Markov chain associated with our Metropolis algorithm for solving the minimum code word problem for an $[n, k]$-code satisfies*

$$\phi_k \geq \frac{1}{4D \exp\left(\frac{2(n^\alpha - 1)}{T}\right)},$$

where T is the temperature parameter and α is the exponent used in the cost function. In particular, when $T = \Omega(n^\alpha)$ the conductance is $\Omega\left(\frac{1}{D}\right)$, where D-denotes the number of neighbours for a node in the search graph.

Proof. Consider any non-empty subset S of $\mathcal{S}_k$ such that $C_S \leq \frac{1}{2}$. There are two possibilities: either $|S| \leq \frac{|\mathcal{S}_k|}{2}$ or $|S| > \frac{|\mathcal{S}_k|}{2}$. We handle these cases separately.

First assume that $|S| \leq \frac{|\mathcal{S}_k|}{2}$. The flow out of S is bounded as follows.

$$F_S = \sum_{U \in S, V \in \overline{S}} p_{UV} \pi_U \geq \min(p_{UV}) \min(\pi_U) |E(S, \overline{S})|$$

By Lemma 3.4, our search graph has magnification at least 1/2 and hence $|E(S, \overline{S})| \geq \frac{1}{2}|S|$. As a result we have:

$$F_S \geq \min(p_{UV}) \min(\pi_U) \frac{|S|}{2}. \tag{1}$$

We know that:

$$\pi_U = \frac{\exp\left(\frac{-c(U)}{T}\right)}{\sum_{V \in \mathcal{S}_k} \exp\left(\frac{-c(V)}{T}\right)} = \frac{\exp\left(\frac{-c(U)}{T}\right)}{Z}.$$

where Z is the partition function $\sum_{V \in \mathcal{S}_k} \exp\left(\frac{-c(V)}{T}\right)$. Let $c_{\max}$ and $c_{\min}$ denote the maxima and minima of the cost function $c(.)$ respectively. Notice that $c_{\max} \leq n^\alpha$ and $c_{\min} \geq 1$. Therefore, for any element U of the search space, its stationary probability π_U is bounded above and below as follows.

$$\frac{\exp\left(\frac{-n^\alpha}{T}\right)}{Z} \leq \pi_U \leq \frac{\exp\left(\frac{-1}{T}\right)}{Z} \tag{2}$$

Also, the transition probabilities p_{UV} satisfies

$$p_{UV} \geq \frac{1}{2D\exp\left(\frac{c_{\max}-c_{\min}}{T}\right)} \geq \frac{1}{2D\exp\left(\frac{n^\alpha-1}{T}\right)}. \quad (3)$$

From Equations (1),(2) and (3) we obtain

$$F_S \geq \frac{\exp\left(\frac{-n^\alpha}{T}\right)}{Z} \cdot \frac{1}{2D\exp\left(\frac{n^\alpha-1}{T}\right)} \cdot \frac{|S|}{2} \quad (4)$$

The capacity C_S is bounded as follows:

$$C_S = \sum_{U\in S} \pi_U \leq \max(\pi_U)\cdot|S| = \frac{\exp\left(\frac{-1}{T}\right)}{Z}\cdot|S| \quad (5)$$

Therefore from Equations (4) and (5), the conductance $\phi(S)$ of the subset S is lower bounded as:

$$\phi(S) = \frac{F_S}{C_S} \geq \frac{1}{4D\exp\left(\frac{2(n^\alpha-1)}{T}\right)} \quad (6)$$

Now consider the case when $|S| > \frac{|\mathcal{S}_k|}{2}$. Using Equation (6) for $\overline{S}$, we obtain:

$$\phi(\overline{S}) \geq \frac{1}{4D\exp\left(\frac{2(n^\alpha-1)}{T}\right)}.$$

Since $C_{\overline{S}} \geq \frac{1}{2}$ and $\frac{C_{\overline{S}}}{1-C_{\overline{S}}} \geq 1$, we have:

$$\phi(S) = \frac{C_{\overline{S}}}{1-C_{\overline{S}}}\phi(\overline{S}) \geq \frac{1}{4D\exp\left(\frac{2(n^\alpha-1)}{T}\right)}$$

Thus, we find that for both the cases, *viz.*, $|S| \leq \frac{|\mathcal{S}_k|}{2}$ and $|S| > \frac{|\mathcal{S}_k|}{2}$,

$$\phi(S) \geq \frac{1}{4D\exp\left(\frac{2(n^\alpha-1)}{T}\right)}.$$

As a result, the conductance ϕ_k of the Markov chain is bounded by

$$\phi_k \geq \frac{1}{4D\exp\left(\frac{2(n^\alpha-1)}{T}\right)}.$$

□

We use the above conductance result to show that the Markov chain for the Metropolis algorithm mixes rapidly, i.e., in time polynomial in n for an input $[n,k]$-code. Let $P^t(U_0,.)$ denote the probability distribution obtained by running the Markov chain for t steps starting the chain at U_0. As before π denotes the stationary probability distribution. To define mixing time we need the concept of total variation distance.

Definition 4.3. **Total Variation Distance[15]** *Total variation distance between two probability distributions $P^t(U_0,.)$ and π is defined as:*

$$||P^t(U_0,.)-\pi||_{TV} = \frac{1}{2}\cdot\sum_{V\in\mathcal{S}_k}|P^t(U_0,V)-\pi(V)|$$

The mixing time $t_{\text{mix}}(\varepsilon)$ of the Markov chain is defined as

$$t_{\text{mix}}(\varepsilon) = \min\{t : \Delta(t) \leq \varepsilon\},$$

where $\Delta(t) = \max_{U_0\in\mathcal{S}_k}||P^t(U_0,.)-\pi||_{TV}$ denotes maximal variation distance between $P^t(U_0,.)$ and π as U_0 varies over the elements of the search space $\mathcal{S}_k$[15]. In other words, independent of the choice of the initial state U_0 to start the chain, if we run the Markov chain for $t \geq t_{\text{mix}}(\varepsilon)$ steps, we have the guarantee that in the resulting distribution, the probability $P^t(U_0,U)$ of obtaining the any state U is bounded above and below as follows.

$$\pi_U - \varepsilon \leq P^t(U_0,U) \leq \pi_U + \varepsilon.$$

A lower bound on the conductance ϕ_M of a Markov chain M translates to an upper bound on the mixing time as follows [23, Equation (2.13), page 58].

$$t^M_{\text{mix}}(\varepsilon) \leq \frac{2}{\phi_M^2}\left(\ln\varepsilon^{-1} + \ln\pi^{-1}_{\min}\right). \quad (7)$$

where $\pi_{\min}$ denotes the smallest of the stationary probabilities for the chain M and $t^M_{\text{mix}}(\varepsilon)$ denotes its mixing time. Using Theorem 4.2 we obtain the following bound on mixing time.

Corollary 4.4. *The mixing time $t_{\text{mix}}(\varepsilon)$, of the Markov chain associated with our Metropolis algorithm on an input $[n,k]$-code satisfies:*

$$t_{\text{mix}}(\varepsilon) \leq 32D^2\exp\left(\frac{4(n^\alpha-1)}{T}\right)\cdot(k^2\ln 2 + \frac{n^\alpha-1}{T} + \ln\varepsilon^{-1}),$$

where D is the number of neighbours of any search point which we know to be $k(k-1)$.

In particular, when the temperature parameter T is $\Omega(n^\alpha)$, the mixing time $t_{\text{mix}}(\varepsilon)$ is $O(k^6 + k^4\ln\varepsilon^{-1})$.

Proof. We first derive a bound on the probability $\pi_{\min}$ as follows. For any element U, we have

$$\pi_U = \frac{\exp\left(\frac{-c(U)}{T}\right)}{\sum_{V\in\mathcal{S}_k}\exp\left(\frac{-c(V)}{T}\right)}.$$

Using the bound $1 \leq c(U) \leq n^\alpha$ for the cost function $c(.)$, we obtain the following.

$$\pi_U \geq \frac{\exp\left(\frac{-n^\alpha}{T}\right)}{\sum_{V\in\mathcal{S}_k}\exp\left(\frac{-1}{T}\right)} = \frac{1}{\exp\left(\frac{n^\alpha-1}{T}\right)|\mathcal{S}_k|}.$$

The set $\mathcal{S}_k$ is the set of all $k\times k$ invertible matrices and hence $|\mathcal{S}_k| \leq 2^{k^2}$. Thus:

$$\pi_{\min} \geq \frac{1}{2^{k^2}\exp\left(\frac{n^\alpha-1}{T}\right)}. \quad (8)$$

Therefore, $\ln\pi^{-1}_{\min} \leq k^2\ln 2 + \frac{n^\alpha-1}{T}$.

The result follows from the above, Equation (7), and the bound on conductance given in Theorem 4.2. □

Given the generator matrix G of an $[n,k]$-code C and a bound M, our task is to find a code word of Hamming weight M or less if it exists. Every run of the Metropolis algorithm for $t_{\text{mix}}(\varepsilon)$ steps provides us with a sample U with probability at least $\pi_U - \varepsilon$. By taking the row of minimum weight in UG, we get a sample code word $\mathbf{x}$ of C. Let p_M be the probability that the sample code word $\mathbf{x}$ is of weight less than or equal to M. Then we have:

$$p_M \geq \sum_{U:\text{wt}(UG)\leq M}(\pi_U - \varepsilon),$$

where $\mathrm{wt}(UG)$ denotes the minimum of the Hamming weights of the rows of the matrix UG.

We take S samples $\mathbf{x}_1, \ldots, \mathbf{x}_S$ obtained through S runs of the Metropolis algorithm each for $t_{\mathrm{mix}}(\varepsilon)$ time and choose the one with the least Hamming weight. The probability that we fail to find a code word of weight M or less is upper bounded by $(1-p_M)^S$. Therefore, to obtain the such a code word with probability at least δ, we need $S \geq \frac{\log(1-\delta)}{\log(1-p_M)}$.

Let $N_M(G)$ denote the number of $k \times k$ invertible matrices U such that UG has a row of Hamming weight less than or equal to M. Choosing $\varepsilon = \frac{1}{2} \cdot \pi_{\min}$, we get p_M to be greater than $\frac{1}{2} \cdot N_M(G) \cdot \pi_{\min}$. Further, setting the temperature T to n^α we obtain, the mixing time $t_{\mathrm{mix}}(\varepsilon)$ to be $O(k^6)$ (Corollary 4.4) and $\pi_{\min}$ to be $\frac{1}{e2^{k^2}}$ (Equation (8)). Using these values, we have a bound on the total time $t_{\mathrm{total}}(\delta, M)$ as

$$t_{\mathrm{total}}(\delta, M) = S \cdot t_{\mathrm{mix}}(\varepsilon) = O\left(k^6 \cdot \frac{\log(1-\delta)}{\log(1-p_M)}\right).$$

where p_M is $O\left(\frac{N_M(G)}{2^{k^2}}\right)$.

For a fixed code C if G and G' are two generator matrices of C we have $G' = UG$ for some U in $\mathcal{S}_k$. Therefore, $N_M(G) = N_M(G')$. As a result $N_M(G)$ is an invariant of the code C. We do not have an analytical expression for it. It is closely related to the well studied weight distribution function of the code.

The above discussions show that our algorithm will be able to find a code word of weight M or less in polynomial time if $N_M(G)$ is large at most a polynomial factor away from 2^{k^2}. Since we do not have a closed form expression for $N_M(G)$ for most codes, we run experiments to see how the algorithm performs on typical binary linear codes.

5. EXPERIMENTAL RESULTS

In the previous section, we proved that for the cost function $c(U) = \mathrm{w}^\alpha$, the family of Markov chains associated with the Metropolis algorithm is guaranteed to mix rapidly if we set the temperature T as n^α. To understand the performance as α varies, we performed experiments on two BCH codes BCH(511,58,183) and BCH(511,184,91) and the trivial $[50, 50, 1]$ and $[100, 100, 1]$ codes with α set as 1/2, 1, 2, 3, 5 and 7. We chose the trivial codes because their minimum weight code words are known: they are code words with a single 1. For $BCH(n, k, d)$ codes d stands for the *design distance* (see Section 7.3 in [14]) and it lower bounds the actual minimum weights. The performance is plotted in Figure 1 in which we observe that the performance is best for $\alpha = 1$. With α set to 1, we tested our algorithm on the set of 20 test cases given in six previous publications [4, 13, 8, 5, 7, 24]. The heuristic search algorithms used in these publications are Wallis's GA [4, 13], Askali's GA [4, 5], Tabu-search [4, 13, 8, 5], Hill Climbing [4, 13, 5], Ant colony optimization [7] and simulated annealing[7, 24]. Each of these heuristics use the same set, namely, the set of all length k binary words, as their search space.

We report the comparison of our algorithm against the algorithms cited above in Table 1, where the last two columns report the performance of our algorithm. The last but one column is for the case when our algorithm is run for k^2 steps and the best of 5000 samples is chosen. In the last column, we do the same with 500-steps, taking the best of 2000 samples. The other columns in the Table 1 give the performance of the previously studied algorithms. Based on the result, it can be seen that our Metropolis algorithm outperforms hill climbing, tabu search, Wallis's genetic algorithm and ant colony optimization [4, 13, 8, 5, 7] in all the twenty cases considered. When compared to the Askali's genetic algorithm [5, 4] on the twenty test cases, the performance was same in 9 cases, the genetic algorithm outperforms our algorithm in 4 test cases and our algorithm outperforms the genetic algorithm in 7 test cases.

We also compared the performance of our algorithm with that of the simulated annealing as reported in [7]. The paper [7] considered two BCH codes namely BCH(127, 64, 21) and BCH(255, 91, 51) and obtained code words of weight 27 and 75 respectively. In comparison, our algorithm was able to attain the minimum weight as 21 and 55 respectively.

6. CONCLUSION

In this paper, we studied the performance of the Metropolis algorithm for the minimum weight code word problem for binary linear codes. For an $[n, k]$-code, the algorithm uses a search space consisting of $k \times k$ invertible matrices and two such matrices are considered neighbours if one can be obtained from the other by an elementary row operation. We prove that the magnification of the search graph is large. Since this is the property of the search space, other random search heuristics can also possibly use the same search space profitably. Using the magnification result, we also prove that the Markov chain associated with a problem instance has large conductance, which is a necessary condition for the Metropolis algorithm to preform well on that instance. As simulated annealing (SA) has a close connection to the Metropolis algorithm, it would be instructive to try SA for this problem on our search space. We performed experiments to see how well does our algorithm perform on certain codes as against previously studied heuristics for the problem. The experimental results show that our algorithm performs quite well, it out-performs several previous heuristic algorithms used for this problem.

7. REFERENCES

[1] D. Ahmed and A. Asimi. A pseudo random generator efficient based on the decoding of the rational binary goppa code. *International Journal of Engineering Science and Technology (IJEST)*, 5(2):359–364, 2013.

[2] K. B. Ajitha Shenoy, S. Biswas, and P. P. Kurur. Metropolis algorithm for solving shortest lattice vector problem (svp). In *Hybrid Intelligent Systems (HIS), 2011 11th International Conference on*, pages 442–447, Dec 2011.

[3] K. B. Ajitha Shenoy, S. Biswas, and P. P. Kurur. Search graph formulation and hasting's generalization of metropolis algorithm for solving svp. *International Journal of Computer Information Systems and Industrial Management Applications*, 5:317–325, 2013.

[4] M. Askali, A. Azouaoui, S. Nouh, and M. Belkasmi. On the computing of the minimum distance of linear block codes by heuristic methods. *International Journal of Communications, Network and System Sciences*, 5(11):774–784, November 2012.

[5] M. Askali, S. Nouh, and M. Belkasmi. An efficient method to find the minimum distance of linear block

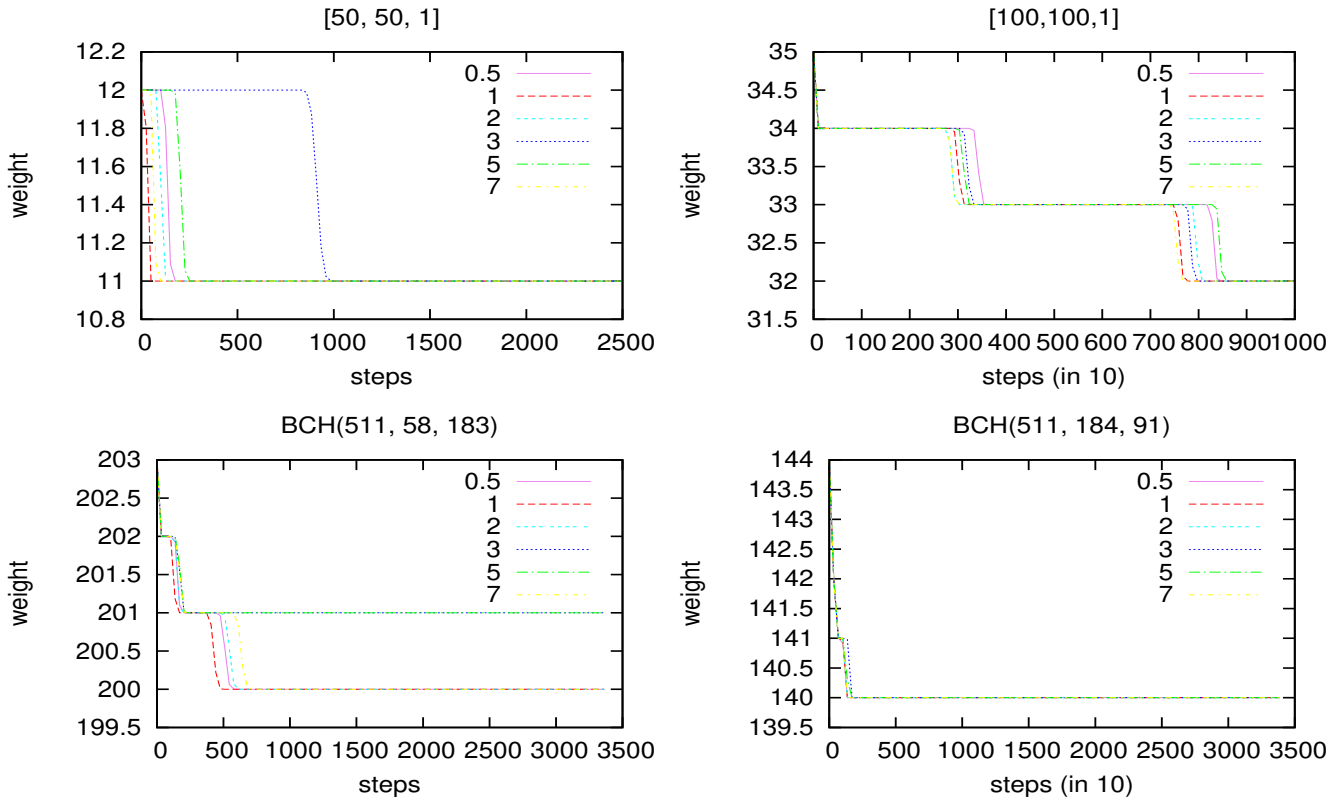

Figure 1: Varying α to choose best α: α set as 0.5, 1, 2, 3, 5 and 7 (k^2 steps, Average is taken over 1000 samples)

Table 1: Minimum Weight Code word found using Our Metropolis Algorithm Vs Different Methods given in papers [4, 13, 8, 5, 7]

Codes BCH $(n, k, d - design)$	Askali's GA [4, 5]	Wallis's GA [4, 13]	Hill Climbing [4, 5, 13]	Tabu Search [8, 4, 5, 13]	Ant Colony Optimization [7]	Metropolis Our Method k^2 steps 5000 Samples	Metropolis Our Method 500 steps 2000 samples
BCH (127, 64, 21)	**21**	**21**	28	24	24	**21**	**21**
BCH (127, 57, 23)	**23**	**23**	28	**23**	24	**23**	**23**
BCH (127, 50, 27)	**27**	**27**	32	31	**27**	**27**	**27**
BCH (255, 71, 59)	**63**	66	79	79	70	**63**	64
BCH (255, 79, 55)	**57**	60	74	64	69	**57**	**57**
BCH (255, 87, 53)	**57**	**57**	70	66	66	**57**	**57**
BCH (255, 91, 51)	**53**	59	72	69	68	54	54
BCH (255, 99, 47)	**51**	55	64	61	62	**51**	52
BCH (255, 107, 45)	**49**	51	64	62	60	50	50
BCH (255, 115, 43)	**45**	50	57	55	58	46	46
BCH (511, 304, 51)	74	79	90	85	–	**73**	74
BCH (511, 286, 55)	84	84	96	92	–	**78**	82
BCH (511, 238, 75)	103	105	118	112	–	102	**99**
BCH (511, 220, 79)	109	111	123	117	–	**108**	**108**
BCH (511, 184, 91)	**111**	128	135	140	–	120	127
BCH (511, 166, 95)	135	137	152	140	–	131	**128**
BCH (511, 121, 117)	155	152	163	163	–	151	**148**
BCH (511, 103, 123)	**160**	164	179	179	–	**160**	**160**
BCH (511, 76, 171)	176	176	195	184	–	**171**	175
BCH (511, 58, 183)	**183**	185	207	199	–	**183**	**183**

codes. In *IEEE International Conference on Multimedia Computing and Signal Processing*, pages 185 – 188. Tangles, May 2012.

[6] E. R. Berlekamp, R. J. McEliece, and H. C. A. van Tilborg. On the inherent intractability of certain coding problems. *IEEE Transactions on Information Theory*, 24(3):384–386, May 1978.

[7] J. A. Bland. Local search optimisation applied to the minimum distance problem. *Advanced Engineering Informatics*, 21(4):391–397, October 2007.

[8] J. A. Bland and A. T. Baylis. A tabu search approach to the minimum distance of error correcting codes. *International Journal of Electronics*, 79(6):829–837, 1995.

[9] R. Bose. *Information Theory Coding and Cryptography*. McGraw-Hill, New Delhi, India, 2008.

[10] I. Dumer, D. Micciancio, and M. Sudan. Hardness of approximating the minimum distance of a linear code. In *Foundations of Computer Science (FOCS), 40th Annual Symposium on*, pages 475–484, October 1999.

[11] J.-B. Fischer and J. Stern. An efficient pseudo-random generator provably as secure as syndrome decoding. In *Proceedings of the 15th Annual International Conference on Theory and Application of Cryptographic Techniques*, EUROCRYPT'96, pages 245–255, Berlin, Heidelberg, 1996.

[12] L. Hogben. *Handbook of Linear Algebra*. Champman and Hall/CRC, Taylor and Francis Group, USA, 2007.

[13] S. K. Houghten and J. L. Wallis. A comparative study of search techniques applied to the minimum distance problem of bch codes. In *In proceedings of sixth IASTED International Conference on Artificial Intelligence and Soft Computing*, pages 164–169. Brock University, May 2002.

[14] W. C. Huffman and V. Pless. *Fundamentals of Error Correcting Codes*. Cambridge University Press, Cambridge, United Kingdom, 2003.

[15] D. A. Levin, Y. Peres, and E. L. Wilmer. *Markov Chains and Mixing Times*. American Mathematical Society, Providence, RI, 2008.

[16] Z. H. Li, T. Xue, and H. Lai. Secret sharing schemes from binary linear codes. *Information Science*, 180(22):2265–2272, 1996.

[17] Z. Liu and X.-W. Wu. On a class of three-weight codes with cryptographic applications. In *Proceedings of 2012 IEEE International Symposium on Information Theory*, MIT, Cambridge, MA, USA, 2012.

[18] J. L. Massey. Some applications of coding theory in cryptography. In *Proceedings of the fourth IMA conference on Cryptography and Coding*, Cirencester, England, 1993.

[19] M. Mitzenmacher and E. Upfal. *Probability and Computing: Randomized Algorithms and Probabilistic Analysis*. Cambridge University Press, New York, USA, 2005.

[20] V. Pless. Bose chaudhuri hocquenghem (bch) codes. *Introduction to the Theory of Error-Correcting Codes, Third Edition*, pages 109–222, 1998.

[21] S. Sanyal, S. Raja, and S. Biswas. Necessary and sufficient conditions for success of the metropolis algorithm for optimization. In *Proceedings of the tenth ACM GECCO'10 Conference on Genetic and Evolutionary Computation*, pages 1417–1424. Portland, OR, USA, July 2010.

[22] A. Sinclair. Improved bounds for mixing rates of markov chains and multicommodity flow. *Combinatorics, Probability and Computing*, 1(4):351–370, 1992.

[23] A. Sinclair. *Algorithms for Ranodom Generation and Counting - A Markov Chain Approach*. Birkhauser Boston, Cambridge, MA 02139, U.S.A., 1993.

[24] M. Zhang and F. Ma. Simulated annealing approach to the minimum distance of error correcting codes. *Internation Journal of Electronics*, 76(3):377–384, 1994.