Protein Folding Problem on a Grid

Using randomized Metropolis Algorithm

A Thesis Submitted
in Partial Fulfilment of the Requirements
for the Degree of

Master of Technology

by

Anand Babu.N.B.
Roll No. : 12111011

under the guidance of

Prof. Somenath Biswas
Prof. Piyush P Kurur

Department of Computer Science and Engineering

Indian Institute of Technology Kanpur

July, 2014

CERTIFICATE

It is certified that the work contained in this thesis entitled "*Protein Folding Problem on a Grid*", by *Anand Babu.N.B.(Roll No. 12111011)*, has been carried out under my supervision and that this work has not been submitted elsewhere for a degree.

(Dr. Somenath Biswas)
Department of Computer Science and Engineering,
Indian Institute of Technology Kanpur
Kanpur-208016

(Dr. Piyush P Kurur)
Department of Computer Science and Engineering,
Indian Institute of Technology Kanpur
Kanpur-208016

June, 2014

Abstract

In order to study how a protein chain folds rapidly, Sali, Shaknovich and Karplus had used the grid model of folding and used the randomized Metropolis algorithm to find a minimum energy conformation. They conjectured that when the energy gap between the minimum energy and the second minimum energy conformations are high, a polypeptide chain would fold quickly. Our thesis explores the conjecture for the folding problem on 2D grids. Computational results of running the Metropolis algorithm based on contact pairs as well as the one based on energy,are considered here. Also, we find cases where the Metropolis algorithm will not be able to reach the minimum energy conformation. New moves for the Metropolis algorithm are also proposed to tackle the situation with the assumption that compact conformations undergo transformations through other compact conformations.

Dedicated to
my parents,friends and teachers.

Acknowledgement

I would like to express my sincere gratitude towards my thesis supervisors Prof.Somenath Biswas, Prof.Piyush P Kurur and Prof.Nisheeth K.Vishnoi from Microsoft Research India for their constant support and encouragement. I am grateful for their patient guidance and advice in giving a proper direction to my efforts. I would also like to thank the faculty and staff of the Department of CSE for the beautiful academic environment they have created here.

I am indebted to all my friends of the M.Tech 2012 Batch for making these past two years a memorable one for me. I have been fortunate to be with some really interesting, fun people. I especially thank my friends with whom I shared my good times. Their company and their ability to infuse humor into any situation helped me get through some tough times.

Last, but not the least, I would like to thank my parents and other family members for their love, constant support and encouragement. Without their support and patience this work would not have been possible.

Anand Babu.N.B.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

Proteins are biological complex macro molecules that play important role in the body. They are made up of long chains of smaller units, called amino acids. There are 20 different types of amino acids that make up proteins and each of amino acid is coded by a stretch of DNA, called codon, which is of 3-base pairs(nucleotides) in length. DNA is made of 4 nucleotides called ; A(Adenine), T(Thymine), G(Guanine), C(Cystosine). Protein chains differ in their amino acid sequences which in fact are derived from the nucleotide sequence of DNA. Each such protein chain folds to form a unique 3D structure of non covalent bonds interactions such as hydrogen bonding, weak Van der Waal's force, etc. This 3D structure determines its functionality. For understanding the functionality in detail, the 3D structure has to be studied. A chain of amino acids can be in a number of shapes obeying certain structural constraints,these various shapes are called conformations. The number of possible conformations is exponential in the length of the chain. Each conformation has an energy(more particularly, free energy) associated with it. As it is the case with all natural systems, that conformation will be the stable conformation of the chain which has the minimum free energy. A natural protein chain usually forms in an arbitrary initial configuration and then goes to its minimum energy conformation which is its stable form. Such a conformation determines the functionality of the protein.

Protein Folding problem is to determine, from the knowledge of the sequence of amino acids of a protein, its stable lowest energy conformation. The notion of this folding problem came in 1960s. Because of the large number of degrees of freedom for the chain, there are astronomical number of conformations possible. It was observed that even if proteins were folded by sequentially sampling all possible conformations at a rapid rate, still it would take a very long time. Contrary to this, despite having astronomically large number of conformations, natural proteins are known to fold quickly. This paradox was pointed out by Cyrus Levinthal in 1968 and is known as the Levinthal's paradox and he proposed that random search on possible conformations does not occur but it folds through a series of intermediate meta stable states. The problem is very important because solving the protein folding problem will be very useful for manufacturing drugs for many diseases and moreover its the structure that decides the behavior of protein chain. We need a predictive manner in which the proteins fold. It is assumed that proteins fold to its native conformations near their global minimum energy conformation. A simplified version of the problem is folding of a chain on a grid. A grid is a 3D or 2D coordinate space where each chain of the protein occupies a point in the coordinate space. The conformation of proteins on the grid as per the model are self-avoiding paths. Self-avoiding paths are paths on the grid/lattice that do not visit the same coordinate more than once. Any system in the nature tries to be in its most stable state which in fact is the minimum energy state.

In the grid model, the protein chain is thought of as a chain of beads, each bead being an amino acid. Given a protein chain, there are attractions between the amino acids(beads) in the protein molecule. Thereby, each conformation is associated with an energy. For the grid model, energy associated with each conformation is :$E = \frac{1}{2} \sum B_{i,j} C(i,j)$ [3] where $B_{i,j}$ is the interaction energy between i th and j th beads in the chain and $C(i,j)$ is 1 if i and j are nearest neighbors and 0 otherwise ,with $|i-j| > 1$ [5]. The nearest neighbors are those beads in the chain that are less than or equal to unit distance apart and are those which do not have a direct link between them. The energy of the chain is assumed to be only dependent on nearest neighbor contacts and to be independent of other aspects of the conformation. All other interactions always yield to a constant energy contribution and hence discarded. Finding out the minimum energy state among all the possible conformations is essentially an optimization problem. It has been shown that this problem is NP-hard. Sali, Shaknovich and Karplus used a randomized algorithm on 27 monomer protein chains on 3D lattices to find the global minimum energy conformation. The Metropolis algorithm was the randomized algorithm used. It was observed that proteins fold quickly when they have pronounced global minimum energy. Pronounced global minimum energy means the difference in energy between the minimum energy and the second minimum energy is large. On the basis of this observations it has been conjectured that an amino acid chain will fold more rapidly when the difference in energy between the minimum energy state and the second minimum energy state is high. Our work tries to verify the Sali, Shaknovich and Karplus conjecture [1]on 2D lattice. Our work can easily be extended to the 3D case as well. Sali, Shaknovich and Karplus define three transformations which transform one configuration into another. We call such transformations as Monte Carlo moves or simply moves, because these are used by the Metropolis algorithm to move from one configuration to another neighboring configuration in the space of all configurations. In the 2D case, two of these moves are applicable, which we use in our first set of computational experiments. Later, we use a different set of moves which ensures that a compact configuration will move only to another compact configuration. In the 2D case, a compact configuration of n^2 will be termed compact if it is contained in an $n * n$ 2D lattice, and thereby occupying all the n^2 co-ordinates.

1.1 Organization of the Thesis

The rest of this thesis is organized as follows. Chapter 2 explains the Metropolis algorithm,the Monte Carlo moves and introduces the canonical representation of configuration of chains on 2D grids. Chapter 3 presents our experimental results and explains the new moves for the Metropolis algorithm. The conclusion is provided in Chapter 4.

Chapter 2

Canonization of Monte Carlo Moves

This chapter explains the randomized algorithm that was used by Sali, Shaknovich and Karplus which led to their conjecture. We explain the Monte Carlo moves used in it. We define a canonical representation of protein chains to tackle the drifting away problem discussed later in this chapter. Some relevant propositions are also stated and proved here.

2.1 Protein Model

In the grid model of Sali. Shakhnovich and Karplus, the energy of a protein chain depends only on nearest neighbor contacts. It is independent of the other aspects of the chain conformation. Two beads which are non-adjacent in the chain are said to be in contact if these are unit distance apart in the placement of the chain in the grid. Beads adjacent in the chain are not considered to be in contact. The energy function [2]of the chain is :
$E = \sum_{i<j} B_{ij}\delta(r_i - r_j)$
where B_{ij} is the interaction energy between bead i and bead j located at positions r_i and r_j respectively. $\delta(r_i - r_j)$ is 1 if beads are in contact and 0 otherwise. For the study of a model with preconceived biases , the interaction parameters B_{ij} are

obtained from Gaussian distribution with mean B_0 and standard deviation σ_B [2] i.e

$P(B_{ij}) = \frac{1}{\sqrt{2\pi\sigma_B}} e^{-\frac{1}{2}(\frac{B_{ij}-B_0}{\sigma_B})^2}$

2.2 Metropolis Algorithm

The Metropolis algorithm [2] [4]is a randomized algorithm that runs a Markov chain. For our problem the set of conformations is its state space. The algorithm starts with an arbitrary initial conformation. For each conformation there is a set of neighborhood conformations associated with it. These neighborhood conformations are obtained by the moves. Each move happens with some probability. Suppose E be the energy of the present conformation and E' be the energy of the next conformation that the chain can move to.If the new energy is less than the free energy of the previous conformation, the transition is favored with probability 1. Else, if the energy is higher, then the transition happens with a probability of $e^{-\delta E}$ where $\delta E = E' - E$. Energy for conformations have been drawn from a normal distribution with mean $\mu = -2$ and standard deviation $\sigma = 1$ as defined earlier.

The algorithm starts with an arbitrary conformation. Monte Carlo moves are applied to the conformation and list of neighborhood conformations are obtained. Among these neighborhood conformations, we replace all spatially equivalent conformations with a single conformation that appears first in the lexicographical order among all their permutations. We continue replacing the conformations in the neighborhood list until no pair of equivalent conformations is present in the list. A random conformation is selected from among the list and depending upon the difference in free energy of the initial conformation and the one selected, the algorithm moves to the new conformation with a certain probability or else stays at the same conformation.

2.3 Monte Carlo Moves

Monte Carlo moves [2] are the local moves/transformations that are applied to a protein chain on a grid to get different conformations. There are 3 Monte Carlo moves allowed for a 3D lattice. They are the following :-

2.3.1 Monte Carlo Move 1

This move is applied at the end of a chain. It can take one of the 5 possible positions if those lattice points are not already occupied in the grid. The penultimate node's position in the lattice remains the same. So it is a one bead move.

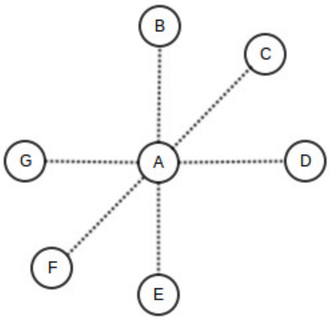

Figure 2.1: Monte Carlo Move 1

2.3.2 Monte Carlo Move 2

This move can be applied if the beads at positions i-1,i,i+1 in the chain are right angled at i in the lattice and if the lattice point diagonally opposite to i is not occupied. Compared to the first move its not just an end chain move, but it is also a one bead move.

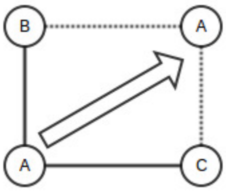

Figure 2.2: Monte Carlo Move 2

2.3.3 Monte Carlo Move 3

This move, known as the Crank Shaft move, can be applied when the beads i,i+1,i+2,i+3 in the chain forms a crank shaft. The crank can be rotated 90 degree in the clockwise or in the anticlockwise direction. This move is a 2 bead move as two of the beads change its co-ordinates in the lattice.

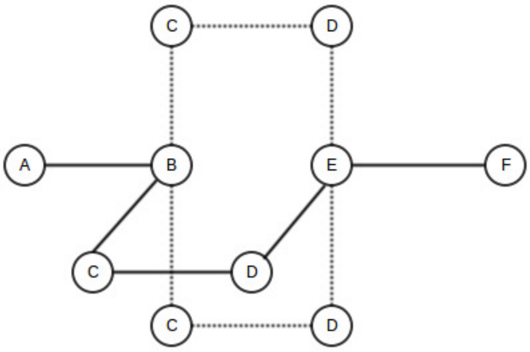

Figure 2.3: Monte Carlo Move 3

2.4 Drifting away Problem

Applying the above Monte Carlo moves to conformations, it can happen that a conformation drift away in the infinite grid in any direction yielding different spatial arrangements/orientations for the same conformation. A simple example is given below:- It can be clearly observed from the above transitions that the initial confor-

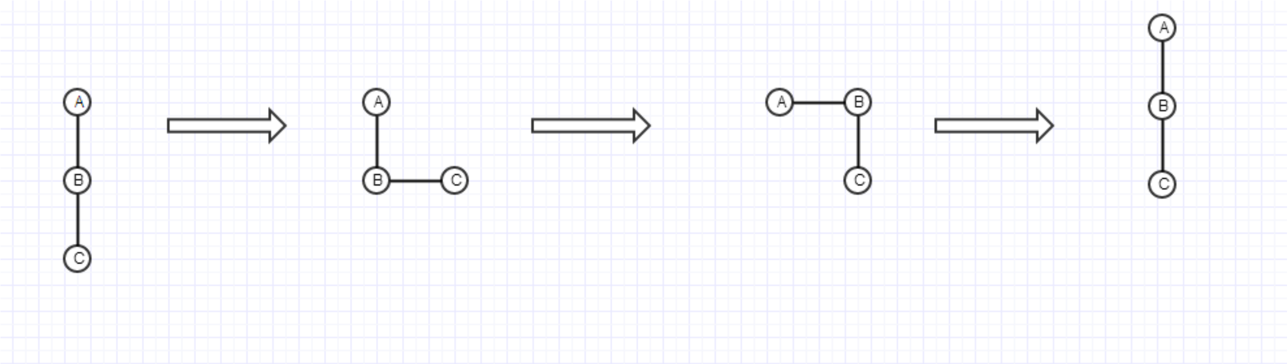

Figure 2.4: Sequence of transformations

mation has now changed its position in the lattice after the two transitions. It has shifted towards the positive Y-axis by a unit which in fact is the same conformation as the initial one. In order to tackle this drifting away problem on the infinite grid, a canonical representation has been introduced.

2.5 Canonical Representation

A specific conformation of protein has different spatial orientations due to the symmetry of the lattice. Each of these different spatial orientations can be specified by a permutation. Each conformation is represented using the directions of the link between the beads in the chain starting from some end. The direction representations used here are L-left,R-Right,U-up,D-down,I-inside,O-outside. A representation for the following 9 bead chain starting from 'A' goes like this : *rrddiror*

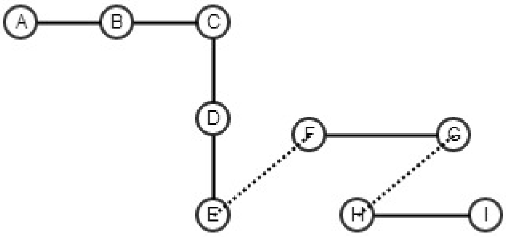

Figure 2.5: Canonical representation

By the symmetry of the 2D square lattice there are 8 different permutations possible. Permutations are bijections from the set S of directions,L,R,U,D to itself. The different permutations possible for a 2D protein chains is as follows :-

$$\pi_0 = \begin{pmatrix} L & R & U & D \\ R & L & D & U \end{pmatrix}$$

$$\pi_1 = \begin{pmatrix} L & R & U & D \\ R & L & U & D \end{pmatrix}$$

$$\pi_2 = \begin{pmatrix} L & R & U & D \\ L & R & D & U \end{pmatrix}$$

$$\pi_3 = \begin{pmatrix} L & R & U & D \\ L & R & U & D \end{pmatrix} = e(the\ \ identity\ \ permutation)$$

$$\pi_4 = \begin{pmatrix} L & R & U & D \\ U & D & L & R \end{pmatrix}$$

$$\pi_5 = \begin{pmatrix} L & R & U & D \\ U & D & R & L \end{pmatrix}$$

$$\pi_6 = \begin{pmatrix} L & R & U & D \\ D & U & L & R \end{pmatrix}$$

$$\pi_7 = \begin{pmatrix} L & R & U & D \\ D & U & R & L \end{pmatrix}$$

These permutations form a group under composition. The composition table is as follows:-

$*$	π_0	π_1	π_2	π_3	π_4	π_5	π_6	π_7
π_0	π_3	π_2	π_1	π_0	π_7	π_6	π_5	π_4
π_1	π_2	π_3	π_0	π_1	π_6	π_7	π_4	π_5
π_2	π_1	π_0	π_3	π_2	π_5	π_4	π_7	π_6
π_3	π_0	π_1	π_2	π_3	π_4	π_5	π_6	π_7
π_4	π_7	π_5	π_6	π_4	π_3	π_1	π_2	π_0
π_5	π_6	π_4	π_7	π_5	π_2	π_0	π_3	π_1
π_6	π_5	π_7	π_4	π_6	π_1	π_3	π_0	π_2
π_7	π_4	π_6	π_5	π_7	π_0	π_2	π_1	π_0

Table 2.1: Composition table

Identity element is :-

$a * e = e * a = a$

$e = \pi_3$

Inverses of the permutations are as follows:-

$\pi_0^{-1} = \pi_0$

$\pi_1^{-1} = \pi_1$

$\pi_2^{-1} = \pi_2$

$\pi_3^{-1} = \pi_3$

$\pi_4^{-1} = \pi_4$

$\pi_5^{-1} = \pi_6$

$\pi_6^{-1} = \pi_5$

$\pi_7^{-1} = \pi_7$

Proposition 2.5.1. *For a specific 2D protein chain there can be a maximum of 8 different spatial arrangements.*

Proof. All the permutations possible are given above. □

Proposition 2.5.2. *In case of 3D protein chain there can be a maximum of 48 different spatial arrangements.*

Proof. Similar kind of permutations discussed for 2D protein chains can be applied with two more extra directions(I-inside the plane,O-outside the plane) in case of 3D proteins. Because of the 2 additional directions the permutations are obviously higher than 2D which when iterated yields a maximum of 48 different permutations. □

Proposition 2.5.3. *If a configuration C' is reachable from any arbitrary configuration C through the Monte Carlo moves, then it is possible to reach from any permutation of the configuration $\pi(C)$ to the corresponding permutation $\pi(C')$ with the same number of Monte Carlo moves/transformations.*

Proof. Suppose the initial configuration be C,Each Monte Carlo move yields new configurations at each steps say $C_1, C_2, C_3, ..., C_n$.Applying a given permutation to the configurations,let the configurations obtained be $\pi(C_1), \pi(C_1), \pi(C_2), \pi(C_3), ..., \pi(C_n)$. Consider $C_1, C_2, C_3, ..., C_n$ as one single configuration D,with combined number of nodes =$(|C_1| + |C_2| + |C_3| + ... + |C_n|)$ and $\pi(C_1), \pi(C_1), \pi(C_2), \pi(C_3), ..., \pi(C_n)$

as another single configuration D'. Applying permutation $\pi(D)$ will yield D' because it's the same as applying permutation π to each of C_1, C_2, C_3, ..., C_n to yield $\pi(C_1), \pi(C_1), \pi(C_2), \pi(C_3)$, ..., $\pi(C_n)$.

Now it requires to prove that the transformation from $\pi(C_1)$ to $\pi(C_2)$ is valid. If a lattice point X is unoccupied for a configuraion C then the lattice point is unoccupied for configuration $\pi(C)$.Hence the monte carlo move from $\pi(C_1)$ to $\pi(C_2)$ is also valid since the corresponding monte carlo move from C_1 to C_2 is valid. Therefore it takes exact same number of monte carlo moves for any permutation of a specific configuration C to some other configuration C' and for any permutation $\pi(C)$ to $\pi(C')$.

Hence all different permutations of a specific configuration which infact are the various spatial arrangements of the same configuration is considered to be equivalent configurations. □

2.6 Recovering a compact folding from contact points

We are concerned with amino acid chains numbered 1,2,...,n where n = k^2 for some k. We are given a set of pairs of 1,2,....,n which are supposed to be representing the set of contact pairs of a compact folding of the chain on to the k*k grid. A compact folding of a n-length chain is nothing but a Hamiltonian path on the square grid graph of size $\sqrt{n} * \sqrt{n}$. A pair i,j is said to be in contact in a compact folding if $|i - j| > 1$ and i and j are adjacent in the square grid graph. The goal is to either determine the folding up to the symmetries of the plane or to conclude that there is no folding which can give rise to the given set of contact points. We present an algorithm for this problem. A consequence of this algorithm is that the set of contact pairs determines a unique compact folding (up to symmetries of the plane) if it exists.

We give an algorithm to recover the compact folding if one exists and prove that none exists if our algorithm fails. Assume that the vertices of the grid are $\{(i, j) : 1 \leq i, j \leq k\}$. Let C be the set of contact pairs that is given. For $1 \leq i \leq n$, let C_i be the elements in the chain that are in contact with i. Note that there are the following set of cases that can occur:

1. An element i is in the interior of the grid in which case it has 4 neighbors. Unless i = 1 or n, two of the four neighbors are $i - 1$ and $i + 1$ and i is in contact with two elements.

2. i = 1 or n and is in the interior in which case it has exactly 3 contact pairs.

3. i is not one of 1 or n and is on the boundary but not one of the four corners in which case i is involved in exactly one contact pair.

4. i = 1 or n and is on the boundary but not one of the four corners in which case i is involved in exactly two contact pairs.

5. i is not one of 1 or n and is one of the four corners in which case i is involved in no contact pairs.

6. i is one of 1 or n and is one of the four corners in which case i is involved in exactly one contact pair.

The algorithm starts by guessing the element $\sigma_{1,1}$ at (1,1). It can then look at $C_{\sigma 1,1}$ to figure out the two elements to be put at (1, 2) and (2, 1). Up to symmetry, it can place either element at either of the two places. Thus, assuming that the guess $\sigma_{1,1}$ was correct, we have figured out the elements at (1, 1), (1, 2) and (2, 1). The next step is to figure out the element at (2, 2). To do this, first check if there is a j $\in C_{\sigma_{1,2}} \cap C_{\sigma_{2,1}}$. If there is such a j, then we let $\sigma_{2,2} = j$. Else, we know that exactly one of $\sigma_{1,2}$ or $\sigma_{2,1}$ has an adjacent element at (2, 2) which should be among the contact pairs with the other. If such an element exist, we assign it $\sigma_{2,2}$, else we output that there is no feasible compact folding which achieves the given set of contact pairs. Next we determine the elements at the positions (1, 3) and (3, 1). But this is easy as well; for instance, to figure out what to place at (1, 3) we use the fact that we know $\sigma_{1,1}$, $\sigma_{1,2}$ and $\sigma_{2,2}$. Thus, by looking at the $C_{\sigma_{1,2}}$, and using the cases discussed above, we can uniquely figure out $\sigma 1, 3$ or declare that there is no compact formulation consistent with the given set of contact pairs. A similar argument allows us to figure out $\sigma_{3,1}$. We can inductively continue this argument by assuming that we know $\sigma_{i,j}$ for all i, j such that $i + j \leq s$ and figure out $\sigma_{i,j}$ for $i + j = s + 1$.

Chapter 3

Experimental Results

This chapter explains the results obtained by the Metropolis simulations on a 2D protein chain with suitable graphs. It also provides statistics regarding number of conformations on different length chains and their energy densities. The cases where the Metropolis algorithm with the Sali et.al.'s local moves fail and the newly proposed global moves are also explained here in this chapter.

Proposition 3.0.1. *Number of contact points is maximum in a square lattice.*

Proof. Suppose the lattice is a m*n grid.

Let N be the numer of contact pairs.

$$\begin{aligned} N &= (m-1)*n+(n-1)*m-(m*n-1) && (3.1)\\ &= (m*n)-n+(m*n)-m-(m*n)+1 && (3.2)\\ &= (m*n)-(m+n)+1 && (3.3)\end{aligned}$$

Let m*n=k

Equation(3.3) becomes:

$$N \quad = \quad k - (m + \frac{k}{m}) + 1 \tag{3.4}$$

Differentiating w.r.t. m

$$-1 + \frac{k}{m^2} = 0 \tag{3.4}$$

$$k = m^2 \tag{3.5}$$

$$m = \sqrt{k} \tag{3.6}$$

$$n = \sqrt{k} \tag{3.7}$$

Therefore the lattice must be of $\sqrt{k} * \sqrt{k}$ square grid so that the number of contact pairs are maximum. □

Proposition 3.0.2. *All self-avoiding paths cannot be reached through the Monte carlo moves.*

Proof: For proving this we describe a counter example.
Self avoiding path(2D): ***d****dluuurrrdddluu*

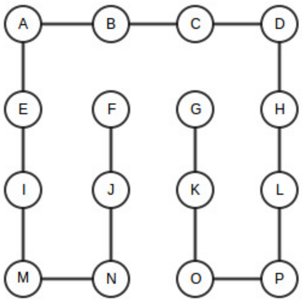

Figure 3.1: Unreachable conformation 2D

Neither of the two Monte Carlo moves allowed for a 2D lattice can be applied on the conformation depicted above. Consider the above configuration to be C and suppose C is reachable from another configuration C' using a Monte Carlo move which is not possible because if such a configuration exists then inverse of the Monte Carlo step can be applied to C to obtain C'.

3.1 Metropolis Algorithm based on number of contact pairs

In this experiment the Metropolis algorithm is run on chains with different number of nodes until the conformation with maximum number of contact pairs is reached. Maximum number of contact pairs for a chain with k^2 nodes is $(k-1)^2$ from proposition[3.0.1]. Maximum number of contact pairs for other nodes are computed by checking all conformations possible for a fixed length chain. A graph is plotted with the Metropolis iterations against the length of chain(number of nodes). For each of the Metropolis iteration, values shown in the graph are average over 50 Metropolis runs.

1. Starts from an initial configuration (a straight chain).

2. Monte Carlo moves are applied onto the initial conformation and neighborhood conformations are obtained.

3. Among the neighborhood conformations list replace all spatially equivalent conformations with a single conformation which appears first in the lexicographical order among all of their permutations.

4. Continue doing Step 3 until no more equivalent conformations for any of the conformations are left behind in the neighborhood list of the initial conformation.

5. Select a random conformation from the neighborhood list.

6. Calculate the number of contact pairs of the initial configuration N , and the randomly selected configuration N'.

7. If $N' >$ N then the random conformation is taken as the next conformation and repeat the whole procedure from Step 1 until the maximum number of contact pairs are reached.

8. Else if $N' \leq$ N then the procedure moves to the new randomly chosen conformation with a probability $e^{-\delta N}$.

9. Else none of the Monte Carlo steps are selected and the whole procedure repeats from Step 1.

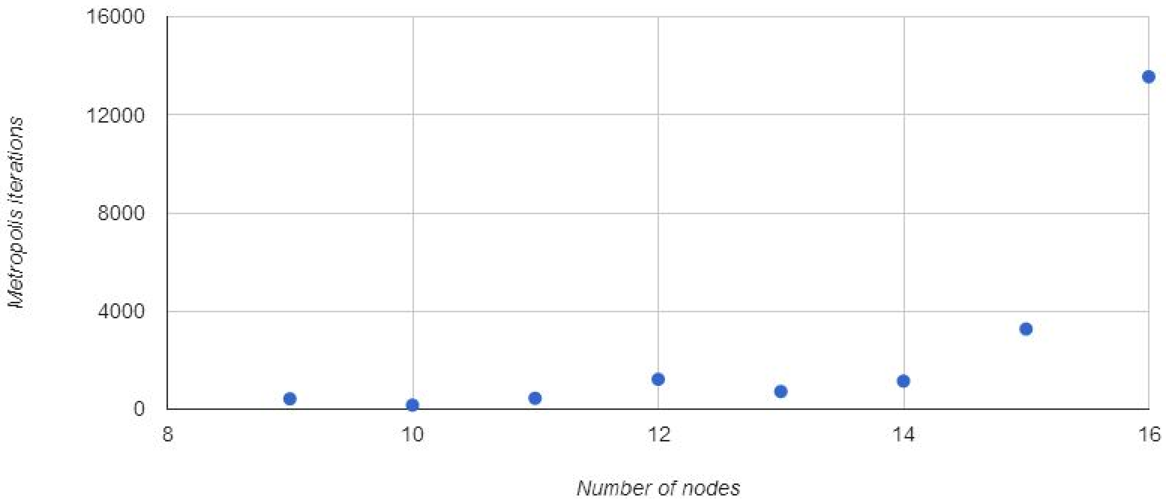

Figure 3.2: Metropolis results

Observations:When the number of nodes N , has fewer numbers of the form $m * n$ that make up N , then it seems to have lesser Metropolis iterations. This may be because the number of different lattices in which the conformations fit to are less.

3.2 Statistics of conformations for different length chains

Here we have experimentally calculated the number of conformations for fixed length chains and categorized them based on the number of contact pairs they have. Shown below is a table for conformations with number of nodes ranging from 9 to 16. Row indicates the number of contact pairs and column indicate the number of nodes of the chain.

	0	1	2	3	4	5	6	7	8	9
9	2244	2032	1072	528	40					
10	5324	5376	3400	1384	784					
11	12668	14224	9832	4608	2384	384				
12	29940	36976	27600	15552	6424	3552	248			
13	71012	95504	77000	45744	22640	10096	2936			
14	167468	243536	211736	133888	76304	29776	15912	2880		
15	396172	619168	572560	387616	226376	109200	45240	16976	1136	
16	932628	1559168	1534512	1107568	676856	364512	149864	69296	21640	552

Table 3.1: Number of contact pairs vs Number of nodes in the chain

All the conformations with different contact pairs for a fixed length chain adds upto the total conformations possible. Shown below is the total number of 2D conformations possible for a fixed length chain with the number of nodes of the chain ranging from 9 to 16.

	Total Conformations
9	5916
10	16268
11	44100
12	120292
13	324932
14	881500
15	2374444
16	6416596

Table 3.2: Total Conformations

This table show that the Metropolis algorithm of section 3.1 is far superior to a sequential search. For example, for a 16 node chain the search space has 6,416,596 elements with 552 conformations. It can be seen that the algorithm samples only about 14000 points to reach an optimum configuration.

3.3 Non-isomorphic compact conformations

Non-isomorphic conformations are conformations that are not spatially equivalent to other conformations. Among 552 compact conformations of a 16 node chain on a $4 * 4$ grid only 69 of them are non-equivalent to each other. All these 69 non-isomorphic compact conformations of a 16 node chain on a $4 * 4$ grid with all of their 9 nearest neighboring pairs are shown below. This serve as an example for Section 2.5 Recovering a compact folding from contact points whose immediate consequence is a one-one mapping between the compact folding and the set of contact pairs.

dddluuuldddluuu	0,7	1,6	2,5	4,11	5,10	6,9	8,15	9,14	10,13
dddluuulldrddlu	0,7	1,6	2,5	4,13	5,12	6,11	8,11	10,15	12,15
dddluuulldrdldr	0,7	1,6	2,5	4,15	5,12	6,11	8,11	10,13	12,15
dddluuulldddruu	0,7	1,6	2,5	4,13	5,14	6,15	8,15	10,15	11,14
dddluulddluuurr	0,15	1,6	2,5	4,9	5,8	6,15	7,12	7,14	8,11
dddlulurulldddr	0,9	1,8	2,5	4,15	5,8	6,13	6,15	7,10	7,12
dddluldluurrull	0,13	1,12	2,5	4,7	5,12	6,9	6,11	10,15	11,14
dddluldluuurrdl	0,13	1,14	2,5	4,7	5,14	6,9	6,15	10,15	12,15
dddluldluuurdru	0,15	1,14	2,5	4,7	5,14	6,9	6,13	10,13	12,15
dddlllurruuldlu	0,11	1,10	2,9	4,9	5,8	7,14	8,13	10,13	12,15
dddlllurruulldr	0,11	1,10	2,9	4,9	5,8	7,14	8,15	10,15	12,15
dddlllurrullurr	0,15	1,10	2,9	4,9	5,8	7,12	8,11	10,15	11,14
dddllluruIurrdd	0,13	1,14	2,15	4,15	5,8	7,10	8,15	9,12	9,14
dddllluurdruull	0,13	1,12	2,11	4,11	5,10	7,10	8,15	9,12	9,14
dddllluuurrddlu	0,11	1,12	2,13	4,13	5,14	7,14	8,15	10,15	12,15

dddllluuurrdldr	0,11	1,12	2,15	4,15	5,14	7,14	8,13	10,13	12,15
dddllluuurddruu	0,15	1,14	2,13	4,13	5,12	7,12	8,11	10,15	11,14
ddluulldrdldrrr	0,5	1,4	2,15	3,10	3,14	4,9	6,9	8,11	10,13
ddllurullddrrr	0,7	1,6	2,15	3,6	3,14	4,11	4,13	5,8	5,10
dlulddrrdllluuu	0,3	1,8	2,5	2,7	4,15	5,14	6,11	6,13	7,10
dlulldrdrrdlllu	0,3	1,10	2,7	2,9	4,7	6,15	8,13	8,15	9,12
dlulldrdldrrrul	0,3	1,14	2,7	2,15	4,7	6,9	8,11	8,15	12,15
dlulldrdldrrurd	0,3	1,14	2,7	2,13	4,7	6,9	8,11	8,13	12,15
dlullddrrrullu	0,3	1,12	2,13	2,15	4,15	6,15	7,14	9,14	10,13
dldrdllluruIurr	0,15	1,4	2,11	2,15	3,6	3,10	7,10	9,12	11,14
dlldrrdllluuurr	0,15	1,6	2,5	2,15	3,12	3,14	4,9	4,11	5,8
dluurullddd luuu	0,3	0,5	2,11	3,10	4,7	4,9	8,15	9,14	10,13
dluurullldrddlu	0,3	0,5	2,13	3,12	4,7	4,11	8,11	10,15	12,15
dluurullldrdldr	0,3	0,5	2,15	3,12	4,7	4,11	8,11	10,13	12,15
dluurullldddruu	0,3	0,5	2,13	3,14	4,7	4,15	8,15	10,15	11,14
dluulddluuurrrd	0,3	0,15	2,7	3,6	4,13	4,15	5,10	5,12	6,9
dluldluurrrulll	0,3	0,11	2,5	3,10	4,7	4,9	8,15	9,14	10,13
dluldluuurrrdll	0,3	0,13	2,5	3,14	4,7	4,15	8,15	10,15	11,14
dluldluuurdrrul	0,3	0,13	2,5	3,12	4,7	4,11	8,11	10,15	12,15
dluldluuurdrurd	0,3	0,15	2,5	3,12	4,7	4,11	8,11	10,13	12,15
dlulurrullldddr	0,3	0,7	2,15	3,6	4,13	4,15	5,10	5,12	6,9
dlllurrurulldlu	0,7	0,9	2,7	3,6	5,14	6,13	8,11	8,13	12,15
dlllurrurullldr	0,7	0,9	2,7	3,6	5,14	6,15	8,11	8,15	12,15
dlllurrullurrrd	0,7	0,15	2,7	3,6	5,10	6,9	8,13	8,15	9,12
dlllurulurrrdld	0,13	0,15	2,15	3,6	5,8	6,15	7,10	7,14	11,14
dllluurdrurulll	0,9	0,11	2,9	3,8	5,8	6,15	7,10	7,14	10,13
dllluuurrrdldlu	0,11	0,13	2,13	3,14	5,14	6,15	8,15	9,12	12,15
dllluuurrrdlldr	0,11	0,15	2,15	3,14	5,14	6,13	8,13	9,12	12,15
dllluuurddrurul	0,11	0,13	2,11	3,10	5,10	6,9	8,15	9,12	12,15

dllluuurddruurd	0,11	0,15	2,11	3,10	5,10	6,9	8,13	9,12	12,15
ddluulddluuurrr	0,5	0,15	1,4	3,8	4,7	5,14	6,11	6,13	7,10
ddluldluuurdrur	0,13	0,15	1,4	3,6	4,13	5,8	5,12	9,12	11,14
ddlllurrullurrr	0,9	0,15	1,8	3,8	4,7	6,11	7,10	9,14	10,13
ddllluuurddruur	0,13	0,15	1,12	3,12	4,11	6,11	7,10	9,14	10,13
dlulddrurdruuu	0,3	0,15	1,10	1,14	2,5	2,9	6,9	8,11	10,13
ddluulddrrruuu	0,5	0,15	1,4	1,14	2,11	2,13	3,8	3,10	4,7
dluurrddruuulll	0,3	0,5	0,7	1,8	4,15	5,14	6,11	6,13	7,10
dluuurdrddruuul	0,3	0,7	0,9	1,10	4,7	6,15	8,13	8,15	9,12
dluuurdrurdddlu	0,3	0,7	0,15	1,14	4,7	6,9	8,11	8,15	12,15
dluuurdrurddldr	0,3	0,7	0,13	1,14	4,7	6,9	8,11	8,13	12,15
dluuurrrdddluul	0,3	0,13	0,15	1,12	4,15	6,15	7,14	9,14	10,13
dluurrrdldrdlll	0,3	0,5	0,9	1,10	1,14	2,15	6,9	8,11	10,13
dldrrrulurullld	0,9	0,13	0,15	1,4	1,8	2,15	5,8	7,10	9,12
ddluuurrdddruuu	0,5	0,7	0,9	1,4	1,10	2,11	8,15	9,14	10,13
ddluuurrrdlddru	0,5	0,7	0,11	1,4	1,12	2,13	8,11	10,15	12,15
ddluuurrrdldrdl	0,5	0,7	0,11	1,4	1,12	2,15	8,11	10,13	12,15
ddluuurrrdddluu	0,5	0,7	0,15	1,4	1,14	2,13	8,15	10,15	11,14
ddllurulurrrddd	0,7	0,11	0,13	1,6	1,14	2,15	3,6	5,8	7,10
dluurrdddllluuu	0,3	0,5	0,7	1,8	1,10	2,11	2,13	3,14	4,15
dluulddrrruuul	0,3	0,13	0,15	1,10	1,12	2,7	2,9	3,6	4,15
dlulurrrdddlllu	0,3	0,7	0,9	1,10	1,12	2,13	2,15	3,6	4,15
dlulddrrruuulll	0,3	0,11	0,13	1,8	1,10	2,5	2,7	3,14	4,15
dlldrrruuulldlu	0,9	0,11	0,13	1,6	1,8	2,5	2,13	3,14	12,15
dlldrrruuullldr	0,9	0,11	0,15	1,6	1,8	2,5	2,15	3,14	12,15

3.4 Stationary Probability Distribution

Stationary probability of a conformation is given by $P_i = \frac{e^{-E_i}}{\sum_{\forall j} e^{-E_j}}$[6]. Here we calculate the stationary probability of the minimum energy conformation with 16 nodes .

This has been done for 1000 different samples. Different samples means the energy between the nodes in each sample is different but drawn from a normal distribution of mean $\mu = -1$ and standard deviation $\sigma = 1$. A plot has been drawn with stationary probability of the minimum energy conformation against energy gap(difference in energy of the lowest energy and the second lowest energy).

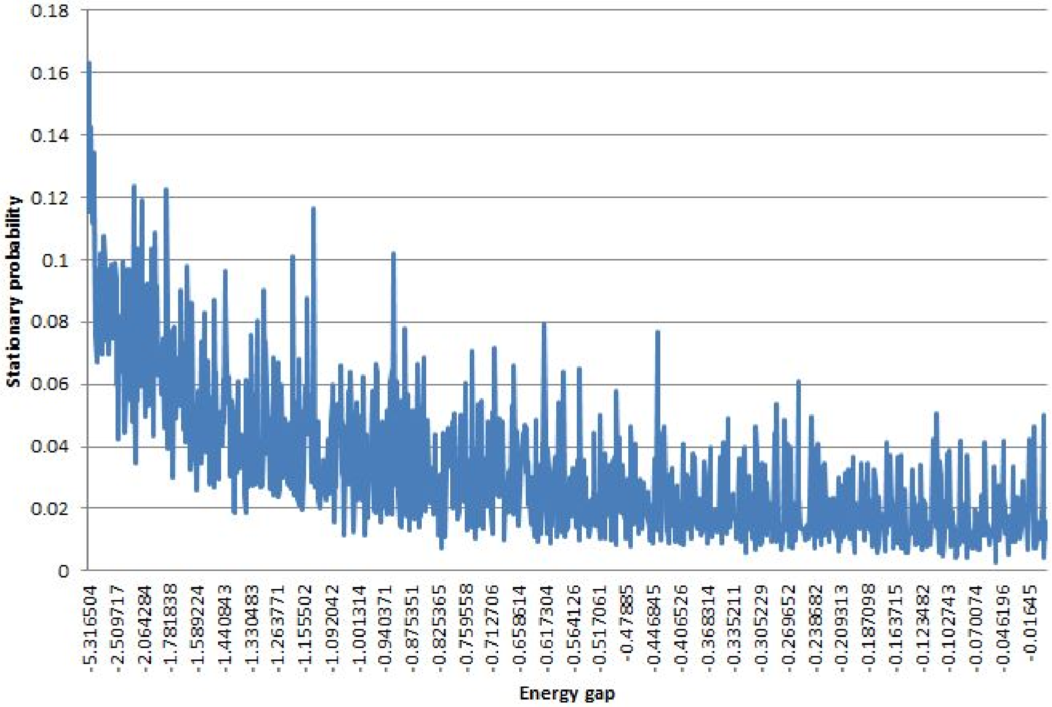

Figure 3.3: Stationary probability distribution

The probability with which minimum energy conformation falls in the range of energy with an interval of 0.5 is given below. The results shown here is from the same 1000 sample described above with normal distribution of mean $\mu = -1$ and standard deviation $\sigma = 1$. Larger the stationary probability of the minimum energy configuration, easier will be for the Metropolis algorithm to find the minimum energy conformation, assuming that number of iterations to come close to the stationary distribution does not vary too much with different samples. Figure 3.3 shows that the stationary probability of the minimum energy conformation tends to increase with the energy gap. Thus, this observation is an evidence in support of Sali, Shaknovich, Karplus conjecture.

Energy (From)	Energy (To)	Probability
-5.5	-5	0.001
-5	-4.5	0
-4.5	-4	0
-4	-3.5	0.004
-3.5	-3	0.003
-3	-2.5	0.025
-2.5	-2	0.03
-2	-1.5	0.069
-1.5	-1	0.149
-1	-0.5	0.265
-0.5	0	0.454

Table 3.3: Probability Table

3.5 Metropolis Algorithm based on energy between monomers

In here the Metropolis algorithm is run until the minimum energy conformation is reached. The energy between each pair of nodes is derived from a normal distribution. The minimum energy conformation is calculated by iterating over all possible conformations. The Metropolis simulations done here is on a chain with 16 nodes. A graph has been plotted with the Metropolis iterations against energy gap(difference in energy between the minimum energy and the second minimum energy). Each Metropolis iteration shown in the graph are the expected values calculated from the average values taken over 50 iterations.

1. Starts from an initial conformation (a straight chain).

2. Monte Carlo moves are applied onto the conformation and neighborhood conformations are obtained.

3. Among the neighborhood conformations list replace all spatially equivalent conformations with a single conformation which appears first in the lexicographical order among all their permutations.

4. Continue doing Step 3 until no more equivalent conformations for any of the conformations are left behind in the neighborhood list.

5. Select a random conformation from the neighborhood list.

6. Calculate the energy of the initial conformation E ,and the randomly selected configuration E'.

7. If $E' < E$ then the random conformation is taken as the initial conformation and repeats the whole procedure from Step 1 until maximum energy conformation is reached.

8. Else if $E' \geq E$ then the procedure moves to the new random configuration with a probability of $e^{-\delta E}$.

9. Else none of the Monte Carlo steps are selected and the whole procedure repeats from Step 1.

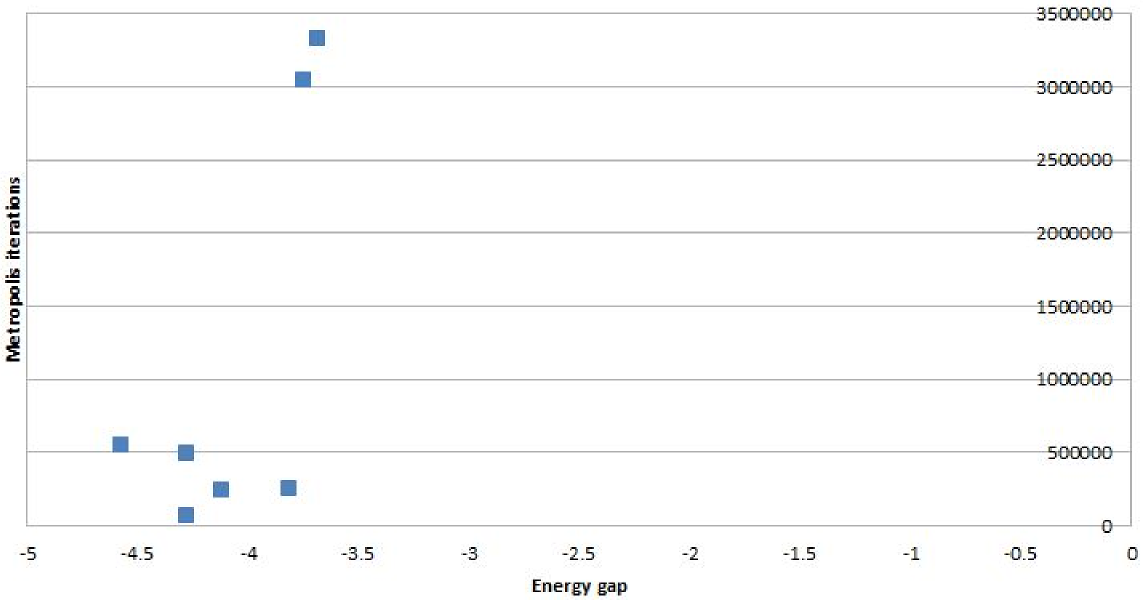

Figure 3.4: Metropolis Expected values

Observations: From the graph it can be inferred that when the difference between lowest energy and second lowest conformation is appreciably large then the Metropolis iterations taken to converge to the minimum energy conformation is less. This clearly verifies what Sali , Shaknovich and Karplus have stated.

3.6 New moves of transformation

'Folding starts with a rapid collapse from a random conformation to a semi-compact globule. Then it proceeds through a slow rate determining search through the semi-compact globules to find the transition state from which it quickly folds to the native state' [1]. With the assumption that once a compact conformation is reached it undergoes transition only through other compact conformations, new global moves on the chain on a grid have been proposed here. New moves are considered because some of the compact conformations cannot be reached by the Metropolis algorithm with the local moves as explained in proposition[3.0.2].

3.6.1 Move 1:

An edge is added between two nodes in contact, each with degree two. Now we need to remove an edge to get a valid compact conformations. Only the new edge added can be removed to get valid compact conformations in this case. So this move will not yield new conformations.

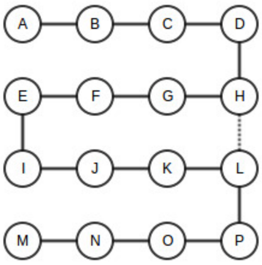

Figure 3.5: New move 1

3.6.2 Move 2:

An edge is added between two nodes in contact, one with degree 1 and the other one with degree 2. Now an edge has to be removed from degree 2 vertex(now the node has degree 3 after adding edge) to yield new conformations in here.

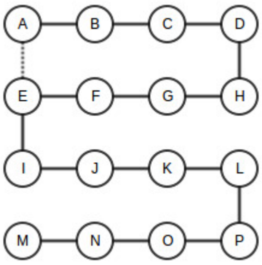

Figure 3.6: New move 2

3.6.3 Move 3:

An edge is added between 2 nodes having degree 1. After adding this edge all nodes have degree 2. Now removing any edge yields a new conformations.

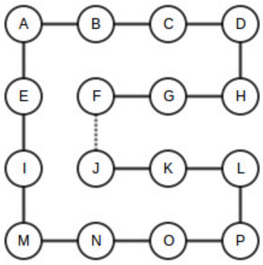

Figure 3.7: New move 3

3.7 Metropolis Algorithm with the new moves

Here the earlier explained Monte Carlo moves in section [2.2] are replaced by the new global moves.

1. Starts from any compact configuration.

2. Add an edge between any two nodes that are unit distance apart on the lattice. There arises 3 cases.

   (1)An edge between a degree 2 vertex and another degree 2 vertex.

   (2)An edge between a degree 2 vertex and a degree 1 vertex.

   (3)An edge between a degree 1 vertex and another degree 1 vertex.

3. If case (1) then

   It is only possible to remove that same previously added edge to get a valid compact configuration. So neighborhood list is not updated.

4. If case (2) then

   It is only possible to remove an edge from a degree 3 node to obtain a valid configurations.All such valid configurations are updated to neighborhood list.

5. If case (3) then

   It is possible to remove any edge already present to give new valid compact configurations.All such valid configurations are updated to neighborhood list.

6. Select a random conformation from the neighborhood list.

7. Calculate the energy of the initial compact conformation E , and the randomly selected conformation E'.

8. If $E' < E$ then the random compact conformation is taken as the initial conformation and repeats the whole procedure from Step 1 until the minimum energy compact conformation is reached.

9. Else if $E' \geq E$ then the procedure moves to the new random conformation with a probability of $e^{-\delta E}$.

10. Else none of the global moves are selected and the whole procedure repeats from Step 1.

Using Monte Carlo moves all compact conformations were not reachable. Since with the new moves all compact conformations can be reached which has not been proved but experimentally verified for $4*4$ grid. A non- trivial example of transition from *rrrdllldrrrdlll* to *ullurrrdddlllur* is as shown below. The dotted lines indicate adding of the new edge which basically is the next move. There can be different sequences of steps in reaching the target conformation from the initial conformation. Here the moves are not local any more. The non-trivial example given below takes 8 transformations to reach the target one.

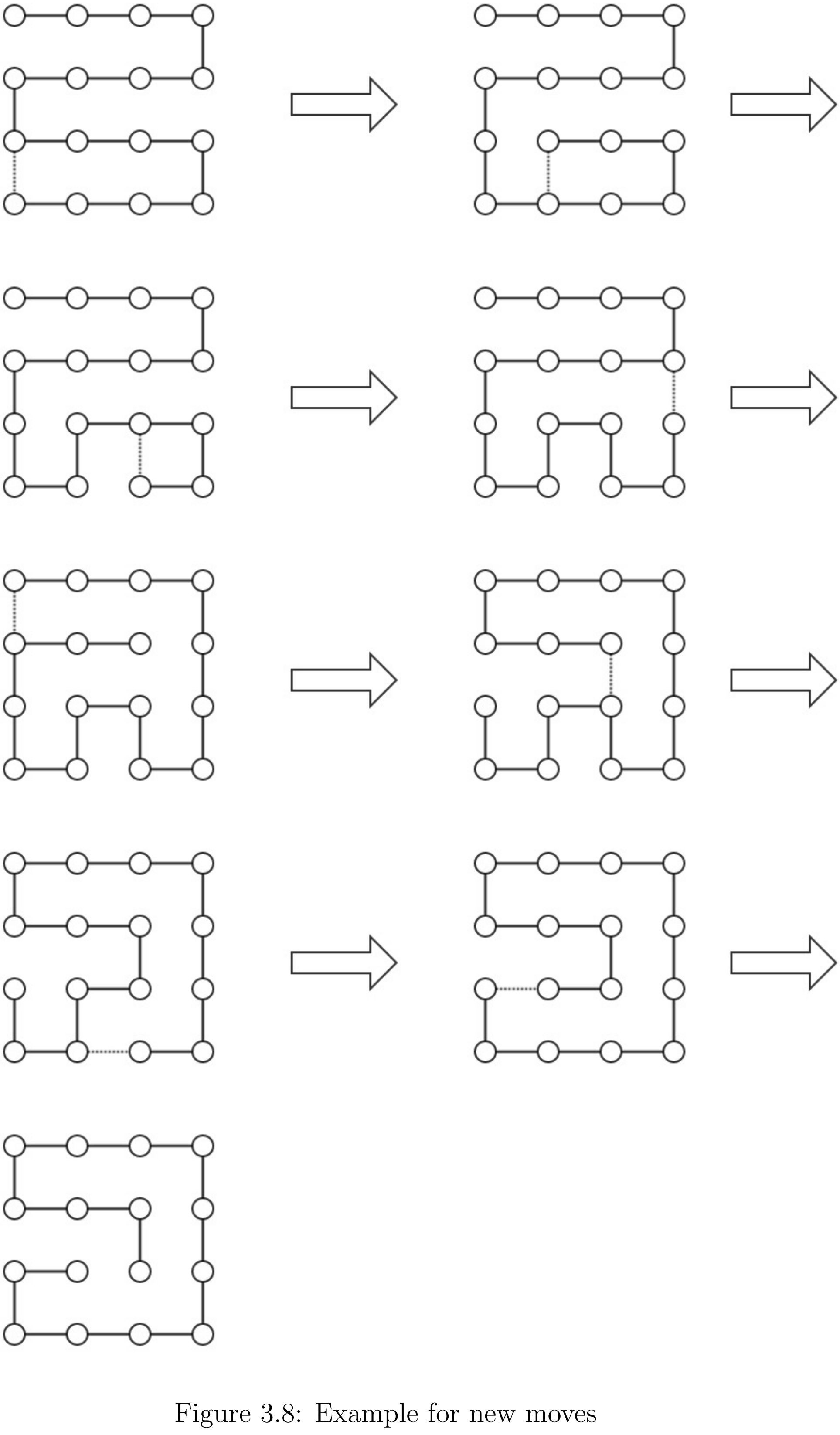

Figure 3.8: Example for new moves

The above non-trivial example gives an intuition that this transitions are pretty faster. Here a graph is plotted between Metropolis iterations against energy gap(difference in minimum energy and the second minimum energy). The results with the new moves on the Metropolis algorithm are shown below. Energy is derived from normal distribution with mean $\mu = -1$ and standard deviation $\sigma = 1$ for a 16 node chain on a $4 * 4$ grid. Each of the Metropolis values depicted in the graph below has been taken over an average of 50 Metropolis runs.

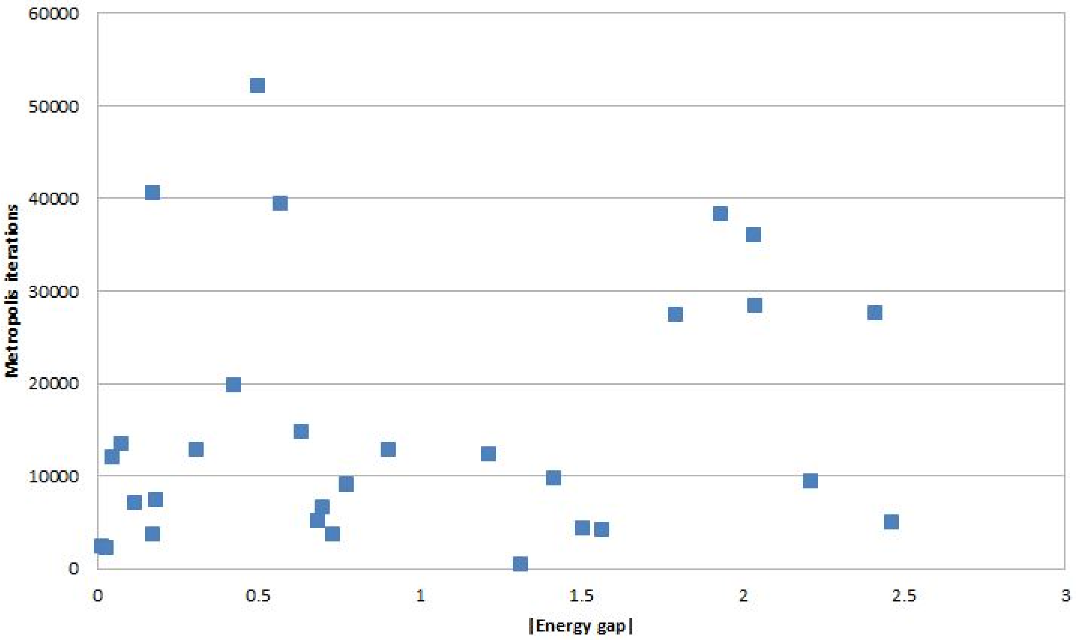

Figure 3.9: Results of Metropolis with new move

Observations: It can be clearly viewed that reaching the minimum energy conformation from an arbitrarily chosen compact conformation through the new global moves is taking very less number of iterations. This can be helpful in finding the minimum energy conformation for large monomer chains in a very short time.

3.8 Energy Density graph

To know how the energy of the conformations of a protein chain of 16 nodes is distributed a graph is plotted with energy against cumulative fraction. Cumulative fraction at an energy $E$is the total fraction of conformation whose energy is greater than E. 6 different samples are obtained each with different energy distributions but all with same mean $\mu = -2$ and standard deviation $\sigma = 1$[1].

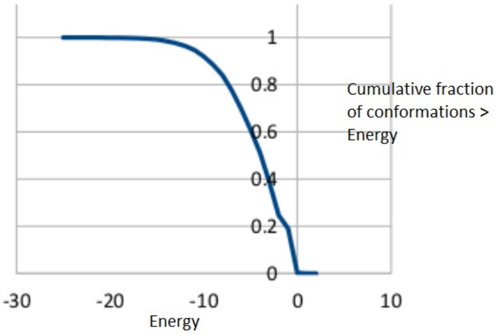

Figure 3.10: Sample 1

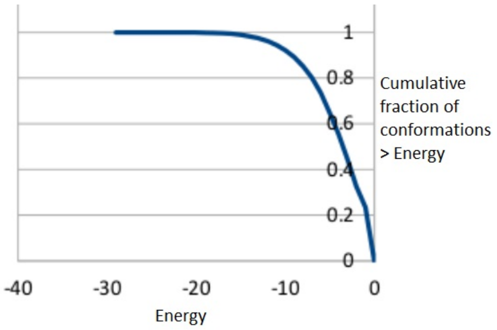

Figure 3.11: Sample 2

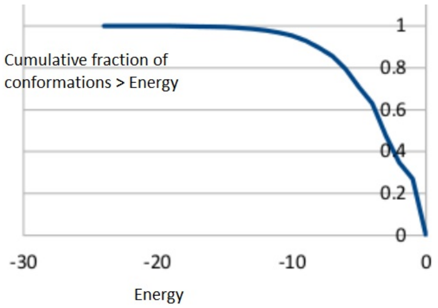

Figure 3.12: Sample 3

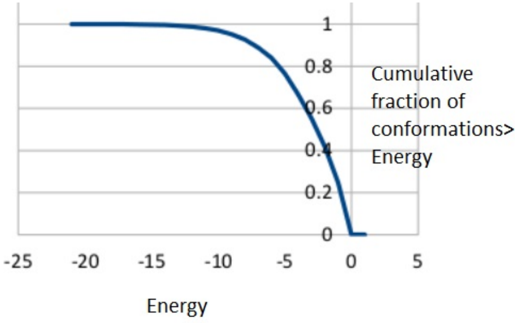

Figure 3.13: Sample 4

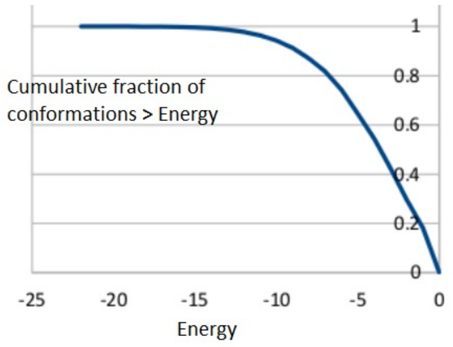

Figure 3.14: Sample 5

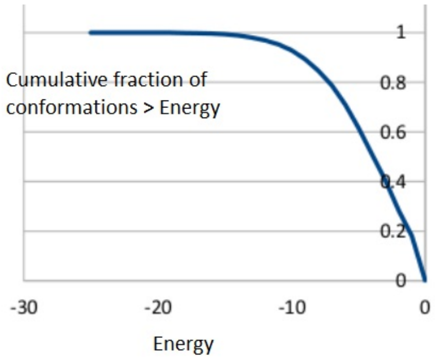

Figure 3.15: Sample 6

Observations: All the 6 graphs plotted below have a general trend that towards lower energy cumulative fraction does not change much. This small change in cumulative fraction towards lower energy indicates that only few conformations are there with low energies. Whereas towards the higher energy the cumulative fraction varies quite appreciably. Most of the conformations are having higher energy in all the graphs plotted.

The Metropolis algorithm based on energy between monomers , covers the conformations with higher energy in less number of iterations. Whereas it takes large number of iterations to cover the lower energy conformations and at last reach the minimum energy conformation. Despite having fewer number of conformations at lower energy the Metropolis takes more iterations and even though there are large number of conformations at higher energy it takes lesser iterations.

Therefore if there is some efficient way of covering lower energy conformations faster , compared to the Metropolis algorithm with the local moves ; then the number of iterations taken to reach the minimum energy conformation reaches a new low. This has been achieved by the Metropolis algorithm with the new global moves proposed in the previous section. It is achieved by using the Metropolis algorithm with the local moves to reach a compact conformation with maximum number of contact pairs and then as the next stage continue with the Metropolis algorithm with the newly proposed global moves.

Chapter 4

Conclusion and Future Work

We have investigated in this thesis how the energy gap between the minimum and second minimum energy conformations effect the convergence time of the protein chain to reach the lowest energy conformation. Our results support the Sali,Shaknovich and Karplus conjecture that it converges faster if the energy gap between the minimum energy state conformation and the second minimum energy state conformation is pronounced has been given here. The results provided here are for 2D monomer chains and can easily be extended to monomer chains in 3D which in fact will provide us with a deeper insight regarding protein folding. We also shows that Metropolis algorithm with the Monte Carlo moves cannot find the minimum energy conformation and thereby would run for a very long time in some cases. It seems that the Metropolis algorithm with our new moves can be used on long 3D protein chains to get the minimum energy conformation in a short time.

Bibliography

[1] Eugene Shakhnovich Andrej Sali and Matrin Karplus. How does a protein fold?

[2] Eugene Shakhnovich Andrej Sali and Matrin Karplus. Kinetics of protein folding,a lattice study of requirements for folding to the native state.

[3] Somenath Biswas and Apurv Nakade. Effect of increasing the energy gap between the two lowest energy states on the mixing time of the metropolis algorithm.

[4] Michael Mitzenmacher and Eli Upfal. Probability and computing randomized algorithms and probabilistic analysis.

[5] Eugene Shakhnovich and Alexander Gutin. Enumeration of all compact conformations of copolymers with random sequence of links.

[6] Raja S Somenath Biswas and Swagato Sanyal. Necessary and sufcient conditions for success of the metropolis algorithm for optimization.