M.Sc. Engg. Thesis

Protein Folding in HP Model on Hexagonal Prism Lattice with Diagonals

by

Dipan Lal Shaw

Submitted to

Department of Computer Science and Engineering

in partial fulfilment of the requirements for the degree of
Master of Science in Computer Science and Engineering

Department of Computer Science and Engineering

Bangladesh University of Engineering and Technology (BUET)

Dhaka 1000

The thesis titled "Protein Folding in HP Model on Hexagonal Prism Lattice with Diagonals", submitted by Dipan Lal Shaw, Roll No. **0413052020**, Session April 2013, to the Department of Computer Science and Engineering, Bangladesh University of Engineering and Technology, has been accepted as satisfactory in partial fulfilment of the requirements for the degree of Master of Science in Computer Science and Engineering and approved as to its style and contents. Examination held on September 6, 2014.

Board of Examiners

1. _____

Dr. M. Sohel Rahman — Chairman

Professor — (Supervisor)

Department of Computer Science and Engineering, BUET, Dhaka.

2. _____

Dr. Mohammad Mahfuzul Islam — Member

Head and Professor — (Ex-Officio)

Department of Computer Science and Engineering, BUET, Dhaka.

3. _____

Dr. M. Kaykobad — Member

Professor

Department of Computer Science and Engineering, BUET, Dhaka.

4. _____

Dr. S. M. Farhad — Member

Assistant Professor

Department of Computer Science and Engineering, BUET, Dhaka.

5. _____

Dr. Muhammad S. Islam — Member

Research Investigator — (External)

Center for Vaccine Science(CVS), ICDDR,B Dhaka-1212.

Candidate's Declaration

This is hereby declared that the work titled "Protein Folding in HP Model on Hexagonal Prism Lattice with Diagonals" is the outcome of research carried out by me under the supervision of Dr. M. Sohel Rahman, in the Department of Computer Science and Engineering, Bangladesh University of Engineering and Technology, Dhaka 1000. It is also declared that this thesis or any part of it has not been submitted elsewhere for the award of any degree or diploma.

Dipan Lal Shaw
Candidate

Acknowledgment

First of all I would like to thank my supervisor, Dr. M. Sohel Rahman, for introducing us to the amazingly interesting and diverse world of protein structure prediction. Without his continuous supervision, guidance and advice it would not have been possible to complete this thesis. I am especially grateful to him for giving us his time whenever we needed, for his encouragement and help at times of disappointment, and always providing continuous support in our effort.

I also want to thank the other members of my thesis committee: Dr. M. Kaykobad, Dr. S. M. Farhad and specially the external member Dr. Muhammad S. Islam for their valuable suggestions.

Last but not the least, I am grateful to my guardians, friends and families for their patience, support and encouragement during this period.

Abstract

The protein folding problem consist in finding the primary structure or native conformation of a protein from its amino acid sequence. It is one of the most studied computational problems in Bioinformatics and Computational Biology. Since, this is NP-hard problem, a number of simplified models have been proposed in literature to capture the essential properties of this problem. An important class of simplified models are known as the lattice models. Lattice models have been proven to be extremely useful tools for reasoning about the complexity of the protein folding problems. Hydrophobic-Polar (HP) model is one of the lattice models where the main force in the folding process is the hydrophobic-hydrophobic force.

In this thesis, we introduce the hexagonal prism lattice with diagonals, that solve some long standing problems of other lattices for protein folding, e.g. parity problem. We present two novel approximation algorithms to solve the protein folding problem in the hexagonal prism lattice with diagonals in HP model. For any given HP string, our first algorithm (Algorithm HelixArrangement) achieves an approximation ratio of 2. Our second algorithm (Algorithm LayerArrangement) achieves a $\frac{9}{7}$ approximation ratio. Furthermore, we incorporate the concept of weighted contact which has biological motivation. Considering weighted contact we analyse our two algorithms as well as a previous algorithm on a different lattice. Finally, we implement our approximation algorithms and conducted experiments on benchmarks datasets.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

The National Center for Biotechnology Information (NCBI 2001) defines bioinformatics as: "Bioinformatics is the field of science in which biology, computer science, and information technology merge into a single discipline. There are three important sub-disciplines within bioinformatics: the development of new algorithms and statistics with which to assess relationships among members of large data sets; the analysis and interpretation of various types of data including nucleotide and amino acid sequences, protein domains, and protein structures; and the development and implementation of tools that enable efficient access and management of different types of information." One of the main areas of bioinformatics is the structural bioinformatics under which structure of biological macromolecules such as protein, RNA, DNA etc. are analysed.

Structural bioinformatics is a branch of bioinformatics which concerns itself with the analysis and prediction of 3D structure of proteins in particular. Different data from micromolecule structures are analysed using computational tools and theoretical frameworks. The data used for analysis are sequence data, sequence alignment data, NMR (Nuclear Magnetic Resonance) data and x-ray crystallographic data. It use various visualization, modeling and prediction tools to analyse and predict the structure, function and behavior of their molecules of interest. One of the goals of structural bioinformatics is to obtain accurate three-dimensional structural models for all known protein families, protein domains or protein folds.

The study of the protein is called proteomics. Finding proteins location, structure and function are the main purpose of this study. The protein folding problem consist in finding

the primary structure or native conformation of a protein from its amino acid sequence.

Proteins are the functional units of life. They are involved in everything from gene expression regulation to defense of an organism. For example, Hemoglobin are used for transportation, Actin and Mysin in muscles are worked for movement, antibodies' function for defence etc. Protein structure leads to its functions. Proteins evolved under selective evolutionary pressure to carry out specific tasks. All these functions are defined largely due to interactions with other molecules. The way a protein interacts with molecules in its environment depends on its three dimensional fold. This fold refers to the overall shape, the surface, active sites, and positioning of key amino acids.

Anfinsens famous experiments in the 1960s stated that, the complex three dimensional structures of the protein molecules are encoded in their amino acid sequences, and the chains autonomously fold under proper conditions. Cracking this code, which is sometimes called "the second part of the genetic code" has been one of the greatest challenges of molecular biology. Although a full understanding of how proteins fold still remains elusive, theoretical and experimental. In the living cell, folding occurs in a complex and crowded environment, often involving helper proteins, and in some cases it can go awry: the protein can misfold, aggregate, or form amyloid fibers. It is increasingly being recognized that misfolded proteins and amyloid formation are the root cause of a number of serious illnesses including several neuro-degenerative diseases. Therefore, the study of protein folding remains a key area of structural bioinfomatics research.

1.1 Motivation

Protein is known as the holy grail of biochemistry and molecular biology. Protein structure prediction is one of the oldest but recalcitrant problem in bioinformatics. The structure of a protein defines what a protein can or cannot do. The distinctive amino acid sequence of proteins allow for the placement of particular chemical groups in specific places in specific places in 3D space. Even a minor modification such as changing one amino acid could change the structure of a protein significantly, thus modifying its function. For example, the sickle cell anemia disease results due to a hemoglobin where the sixth amino acid is changed from

glutamic acid to valine. Protein structures are highly diverse and this diversity the functional diversity of these structures is expanded through interactions with smaller molecules.

The experimental method of determining the native structure of a protein consists in producing a pure solution containing only the protein, followed by, first, crystallizing the protein, and then an x-ray crystallography. A large amount of material of the protein is required for this process and the solution containing the protein must be very pure. But the major limitation of this method is the crystallization process. This step is very time consuming and is limited to a subclass of proteins.

For this reason, many computational techniques have been developed and many simplified models have been introduced to predict protein structures. An important class of simplified models are known as the lattice models. Lattice models have been proven to be extremely useful tools for reasoning about the complexity of the protein folding problems. By sacrificing the atomic details, lattice models can be used to extract essential principles, make predictions, and unify our understanding of many different properties of proteins. Square lattice, triangular lattice, square lattice with diagonal, hexagonal lattice, cubic lattice, face centred cubic lattice etc. are popularly used in literature for protein folding. But these lattices have some drawbacks for protein folding process. A new model that will removes the drawbacks of these lattices might help us to predict protein structure successfully.

Example of some motivation behind predicting protein structure are listed below,

- It can help us understand many diseases causes by disordered protein structures, e.g., Parkinsons disease, Alzheimers disease, Huntingtons disease, BSE (mad cow disease), Cancer, Cystic fibrosis, type II (non-insulin dependent) diabetes etc. These diseases are caused by a specific protein, that misbehaves. In most cases, a defective gene codes cause troublesome protein.

- It also necessary for structure based drug design. Structure based drug design (or direct drug design) relies on knowledge of the three dimensional structure of molecules. A particular drug interacts with a particular protein. Structure based drug design uses the structure of proteins as a basis for designing new drug by applying accepted principles of molecular recognition.

- There are many proteins with similar structure but very different functions. Quite different sequences can adopt the same structure. This fact can useful in identifying

evolutionary relationships. It can, however, identify false relationships as well. Analogous proteins are proteins that have the same function but do not share ancestry. Homologous proteins share ancestry.

- For understanding various biological mechanisms it can help.

  For predicting protein structure many lattice structure had been used in literature. Square lattice and cubic lattice are the mostly studied lattice structure. But in square lattice and cubic lattice it can

1.2 Core Contribution

In this thesis, hexagonal prism lattice with diagonals is introduced. This lattice model removes some of the well known problems of protein folding in other lattices, e.g., parity problem. Also, as will be clear later our proposed lattice model can provide better results.

The main contribution of this thesis are as follows.

1 As mentioned above, we introduce a new lattice model, hexagonal prism lattice with diagonals. Compare to other lattices previously used in literature for protein folding, this new lattice removes drawbacks of other lattices.

2 We present two novel approximation algorithms to solve the protein folding problem in the hexagonal prism lattice with diagonals in HP model. For any given HP string, our first algorithm (Algorithm HelixArrangement) achieves an approximation ratio of 2 for $k > 16$, where k is the total number of H-runs and n is the total number of H. Our second algorithm on hexagonal prism lattice with diagonals (Algorithm LayerArrangement) achieves an approximation ratio of $\frac{9}{7}$ under some parametric constraints. Both algorithms are polynomial in terms of the length of the given HP string.

3 We incorporate the concept of weighted contact which has biological motivation. Considering weighted contact we analyse our two algorithms as well as previous algorithm on a different lattice. In particular we first apply the concept of weighted contact on a previous algorithm (Algorithm ChainArrangement) of Shaw et al.[46]. Considering weighted contact, the Algorithm ChainArrangement provides 1.96-approximation ratio for $k > 8$, where k is number of sequence of Hs in the HP string. This new analysis on hexagonal lattice with diagonal improve the performance of the algorithm.

4 Considering weighted contact, Algorithm HelixArrangement achieves an approximation ratio of 2 for $k > 13$ and Algorithm LayerArrangement achieves 1.45-approximation ratio for $k > 89$, where k is number of sequence of Hs in the HP string.

5 We develop a simple visualization software for the approximation algorithms and tested under standard dataset. This software simulate the Algorithm ChainArrangement, Algorithm HelixArrangemnt and Algorithm LayerArrangement. Protein structure generate from this algorithm along with their contacts are shown in simulation output. The test under standard dataset results similar approximation ratio that theoretically found.

The rest of the chapters are organized as follows. In Chapter 2, we describe the protein folding problem and different types of methods to solve this problem. We also describe HP Models and why it is used in this thesis. In Chapter 3, we present the literature review in the field of protein folding. We also describe different lattices used before for the protein folding problem in this chapter. Chapter 4 presents our main research results on protein folding. We introduce hexagonal prism lattice with diagonals in this chapter and provide two approximation algorithms to solve the problem. The analysis for finding the approximation ratio is also presented here. Chapter 5 presents the analysis using the concept of weighted contact for hexagonal lattice with diagonal. In Chapter 6 the analysis considering weighted contact continues for hexagonal prism lattice with diagonal. In Chapter 7, we describe our Visualization Software and experimental results on this lattice. Finally, in Chapter 8, we conclude our thesis with a brief overview and future research directions.

Chapter 2

Preliminaries

In this chapter, we discuss necessary notion and notations that are necessary to describe the background of this thesis. Here we describe the structure of protein and their functions. We state the protein folding problem and discuss approaches to solve it. We conclude this chapter after describing the HP Model.

2.1 Approximation Algorithms

The algorithm generates approximate solution for optimization problem known as approximation algorithm. These algorithms provide feasible but not necessarily optimal solutions.

Let, C be the cost of a solution found for a problem of size n and C^* be the optimal solution for that problem.
Then we say an algorithm has an approximation ratio of $\rho(n)$ if,
$C/C* \leq \rho(n)$ for minimization problems: the factor by which the actual solution obtained is larger than the optimal solution.
$C*/C \leq \rho(n)$ for maximization problems: the factor by which the optimal solution is larger than the solution obtained.
An algorithm that has an approximation ratio of $\rho(n)$ is called a $\rho(n)$-approximation algorithm.

2.2 Protein

The name protein is derived from the Greek word "protos", meaning "primary". Of all the molecules found in living organisms, proteins are the most important. They are the

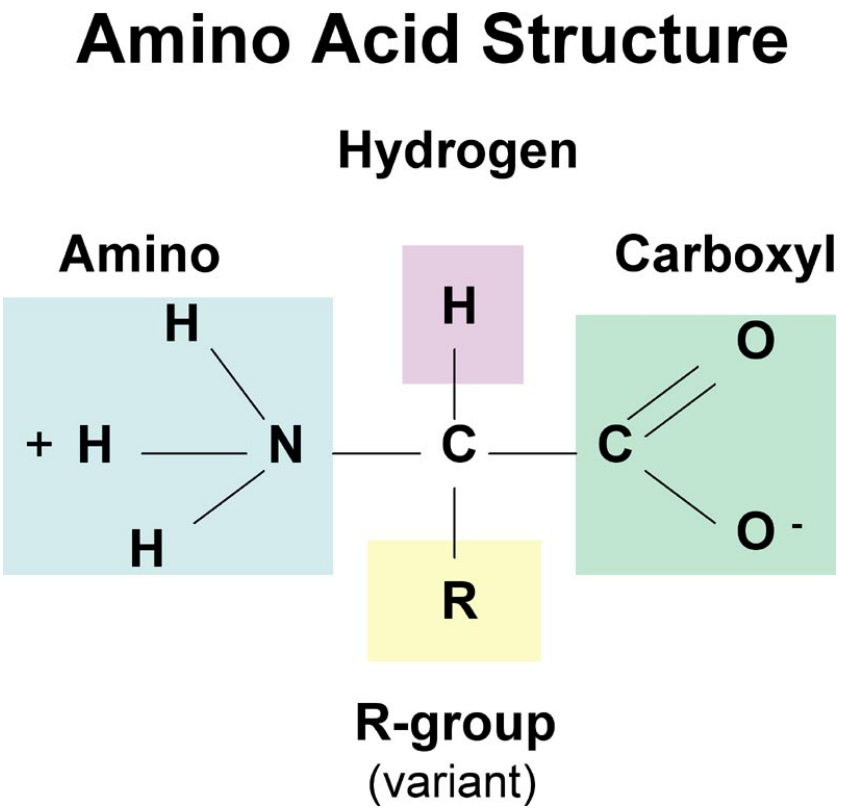

Figure 2.1: Generic Amino Acid Structure

biological workhorses. They carry out vital functions in every cell. They are used to support the skeleton, move muscles, control senses, defend against infections, digest food and process emotions. Proteins can be different in shapes and sizes. For example, they can be round, long, hard or elastic. Its complex shapes include various folds, loops, and curves. More than half of the dry weight of a cell is due to proteins. They have a range of indispensable roles; for example, enzymes, the bio-catalysts that carry out crucial biochemical reactions in every cell. Otherwise it would be too slow to sustain life.

There are more than 100,000 different types of proteins exists in our body. What is remarkable is that all of these are produced from a set of only 20 building blocks, known as amino acids. All amino acids have the same basic structure an alpha carbon $C\alpha$, an amino group, a carboxyl group, a hydrogen atom and a side-chain, known as R (See Fig. 2.1). The R group distinguishes one amino acid from another. Furthermore, the side chain is responsible for the specific chemical properties of the amino acid. The two simplest amino acids are glycine and alanine (please see Fig. 2.2). Depending on the nature of the side-chain, an amino acid can be hydrophilic (water-attracting) or hydrophobic (water-

repelling), acidic or basic. The diversity in the side-chain properties give proteins their different characteristics. 19 of the 20 common amino acids have a chiral carbon atom. Gly does not. Mirror image pairs of amino acids are designated by L (levo) and D (dextro). Proteins are assembled from L amino acids. Only a few D amino acids occur in nature. Almost all sugars have a D conformation. Threonine and isoleucine have 2 chiral carbons each, thus producing 4 possible stereoisomers each. Isomers depend on the position of the 4 group around the chiral center. Amino acids are L or D depending on the position of the amino group.

Table 2.2 presents the list of 20 amino acids in proteins with their three-letter and one-letter codes. Based on this table the amino acids can be classified into several groups. Charged or neutral amino acid. Polar or non-polar amino acid. Charged amino acid shown in table is subclass of polar amino acids. Charged amino acids can be classified in two categories, acidic or negatively charged amino acids and basic or positively charged amino acids.

Amino acids are joined together in proteins by peptide bonds. A peptide bond forms between the carboxyl group of one amino acid and the amino group of the adjacent amino acid (See Fig. 2.3). These chemical bondings aid in holding the protein together and giving it its shape. There are two general classes of protein molecules: globular proteins and fibrous proteins. Globular proteins are generally compact, soluble, and spherical in shape. Fibrous proteins are typically elongated and insoluble. Globular and fibrous proteins may exhibit four types of protein structure. These structure types are called primary, secondary, tertiary, and quaternary structures. The sequence of amino acids in a protein defines its primary structures. Secondary structure refers to the coiling or folding of a polypeptide chain that gives the protein its 3-D shape. Tertiary Structure refers to the comprehensive 3-D structure of the polypeptide chain of a protein. Quaternary Structure refers to the structure of a protein that is formed by interactions between multiple polypeptide chains (See Fig. 2.4).

2.3 Protein Folding Problem

Protein folding refers to the complex spontaneous assembly of proteins. It concerned with how it gets its native structure. Protein folding is a spontaneous, ordered and reversible process.

Table 2.1: List of 20 amino acids with their three-letter and one-letter codes

Amino Acids			
Characteristic	Name	3 Letter code	1 Letter code
Charged	Arginine	Arg	R
	Lysine	Lys	K
	Aspartic acid	Asp	D
	Glutamic acid	Glu	E
Hydrophilic or polar	Glutamine	Gln	Q
	Asparagine	Asn	N
	Histidine	His	H
	Serine	Ser	S
	Threonine	Thr	T
	Tyrosine	Tyr	Y
	Cysteine	Cys	C
	Methionine	Met	M
	Tryptophan	Trp	W
Hydrophobic or non-polar	Alanine	Ala	A
	Isoleucine	Ile	I
	Leucine	Leu	L
	Phenylalanine	Phe	F
	Valine	Val	V
	Proline	Pro	P
	Glycine	Gly	G

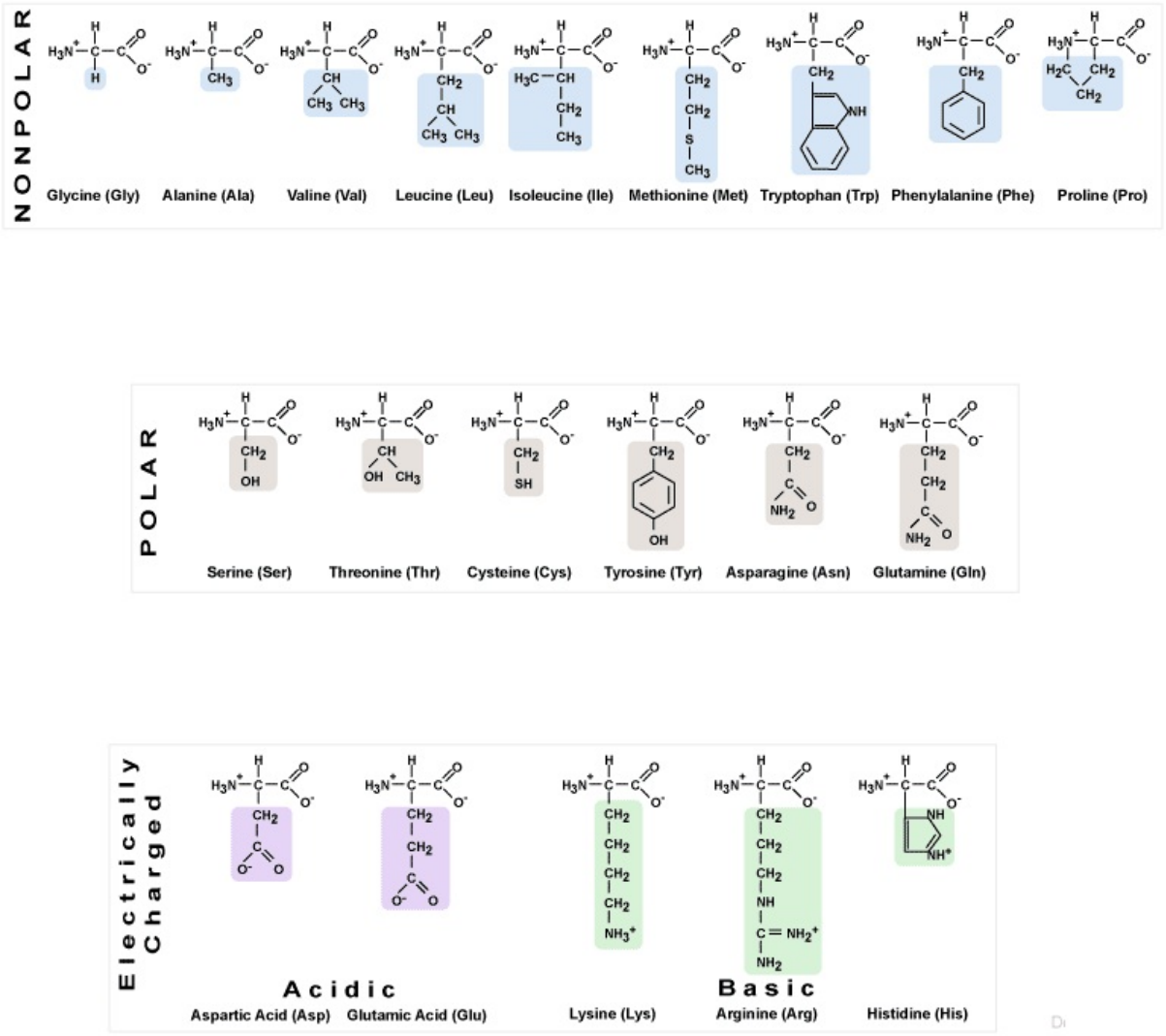

Figure 2.2: Structure of all amino acid

According to K.A. Dill[14] the protein-folding problem is concerned with three broad questions:

(i) What is the physical code by which an amino acid sequence dictates a protein's native structure?

(ii) How can proteins fold so fast?

(iii) Can we devise a computer algorithm to predict protein structures from their sequences?

It is believed that some dominant forces are the major contributor for protein folding.

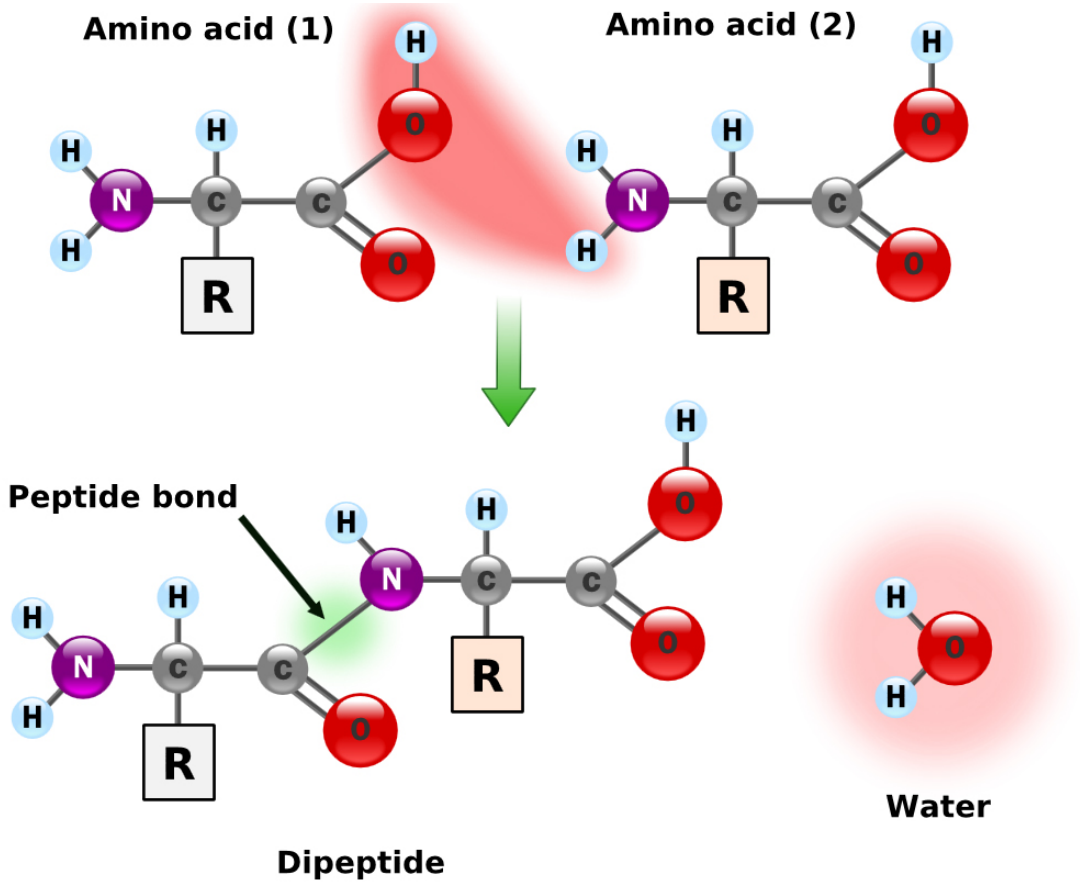

Figure 2.3: Peptide bond

Dominant forces for protein folding are given below,

- Hydrogen bonds
- Induced dipole effects
- Conformational entropy
- Van der waals interaction
- Dissolvent property
- Dipole interactions
- Hydrophobic effect
- Salt bridges

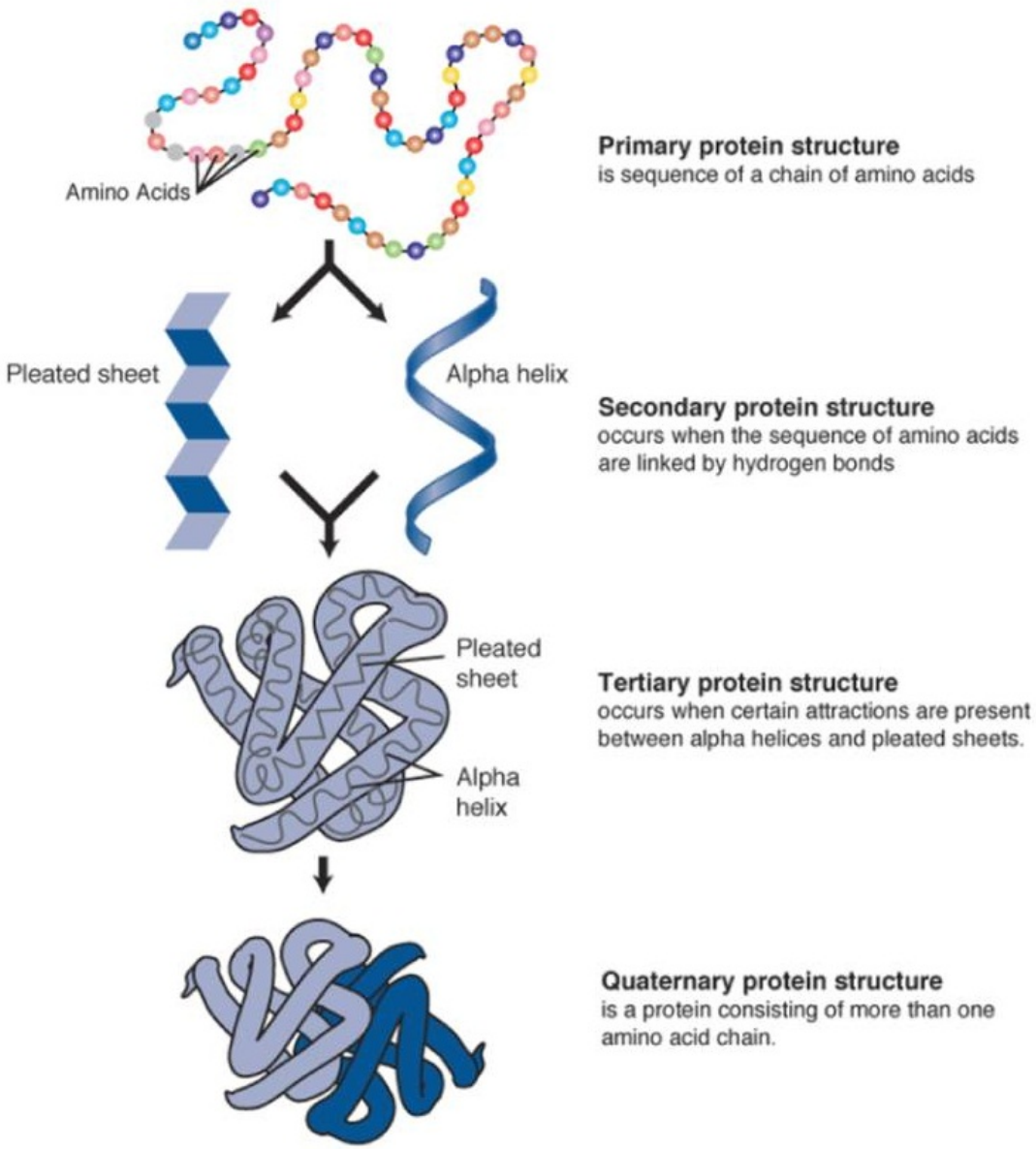

Figure 2.4: Four levels of protein structure

The protein folding problem is a optimization problem can be defined as below:

Input : A sequence of amino acids.

Output : Conformation or primary structure of protein.

Goal : Maximization. Maximize the amount of dominant force in a conformation.

2.4 Different Types of Methods

There are many different computational methods of finding a protein's native structure. Despite the quantity of work on this problem over the past 30 years, no truly accurate computational methods exists to predict the 3-dimensional structure from the amino acid sequence. Here we briefly describe the methods of finding protein structures, namely, homology, threading, ab initio techniques.

2.4.1 Homology Method

In homology modeling, the amino acid sequence of a novel protein P is aligned against sequences of proteins Q, whose tertiary structure is available in the Protein Data Bank (PDB) [6]. Regions of P aligned to regions of Q are assumed to have the same fold, while non-aligned regions are modeled by interconnecting loops. Homology modeling is also referred to as comparative protein modeling or knowledge-based modeling. Examples of comparative modeling software include SWISS-MODEL, developed by T. Schwede et al.[4], MODELER developed by the Sali Lab [30] etc.. Comparative modeling relies on the assumption that evolutionarily related (homologous) proteins retain high sequence identity and adopt the same fold.

2.4.2 Fold Recognition or Threading Method

Fold recognition, though known to be NP-complete [37, 36, 51], is a promising structure prediction approach. Fold recognition methods detect folds that can be used for structural modeling with homology at the sequence level. The principle of fold recognition is the identification of folds that are compatible with a given query sequence i.e. instead of sequences being used to predict folds, the folds are fitted to the sequence. This method can be described in four steps,

1. Step 1: Searching for known folds
2. Step 2: Scoring folds
3. Step 3: Identifying candidates that best fit the sequence
4. Step 4: Aligning the query and the best-scoring proteins

Once such a template has been identified, the remainder of the process is the same as comparative modeling.

Impressive results have been obtained in Skolnick Lab program I-TASSER [58] with web server [60]. It is the best-ranked structure predictions in the blind test CASP-7 (Critical Assessment of Techniques for Protein Structure Prediction) in 2006. Success of threading hinges on two things: energetics and the search strategy, i.e., usually Monte-Carlo or some type of branch and bound algorithm. Energetics defines how the PDB is relatively saturated and contains occurrences of almost all protein folds. Search srategy usually used for protein foldings are Monte-Carlo, some type of branch and bound algorithms etc. According to a study of Zhang and Skolnick [61], the PDB is currently sufficiently saturated to permit adequate threading approaches. But it does not give sufficient accuracy required for drug design.

2.4.3 Ab Inito Technique

Despite advances in comparative modeling and threading, there is an interest in ab initio protein structure prediction, since this is the only method that attempts to understand protein folding from basic principles. It apply the search strategy with a physics-based energy function. Comparative modeling and threading depends on finding a suitable template structure. In the absence of a suitable structure, ab initio prediction is the only method.

A typical procedure can be describe as below,

1. Step 1: Define a mathematical representation of a polypeptide chain and the surrounding solvent. That means, define an energy function that accurately represents the physiochemical properties of proteins.
2. Step 2: Use an algorithm to search for a chain conformation which possesses the minimum free energy.

The problem with ab initio methods is that even short polypeptide chains can fold into a potentially infinite number of structures.

Well known software for ab inito technique are CHARMM [11], Amber [18] and variant Molsoft ICM [1]. The Molsoft ICM uses internal coordinates (dihedral angle space) and local optimization for protein docking and protein-ligand interactions. Other ab initio methods include the Baker Lab program Rosetta [10]. Rosetta benchmarked in [22] with comparable accuracy as the Skolnick Lab program I-TASSER [58]. Search strategies of ab initio methods include molecular dynamics simulation, Metropolis Monte-Carlo (Rosetta [10]), Monte-Carlo with replica exchange (I-TASSER [35]), branch-and-bound (ASTROFOLD [34]), integer linear programming (ASTROFOLD [34]), Monte-Carlo with simulated annealing, evolutionary algorithms, genetic algorithms etc.

2.5 HP Model

Finding complete optimal structure of protein is too hard. Hence, many approximation solutions are given by using a simplified, abstract lattice structure. Lattice structure like square lattice, triangular lattice, hexagonal lattice etc. are popularly used in literature. In this section, we talk about the widely used HP model for protein folding. This model is introduced by K.A. Dill, in 1985 [13].

In HP model, there are only two types of beads. H represents the hydrophobic or non-polar beads and P is referred to polar or hydrophilic ones. These beads are randomly distributed in the lattices. In this model it is assumed that the main force in the folding process is the hydrophobic-hydrophobic force, so H-H contacts are the main forces in this model. Two hydrophobic atoms create contacts if they are topological neighbours.

The input to the protein folding problem is a finite string p over the alphabet $\{P,\ H\}$ where $p\ = \{P\}^* b_1 \{P\}^+ b_2 \{P\}^+ ... \{P\}^+ b_k \{P\}^*$. Here $b_i \in \{H\}^+\ for\ 1 \leq i \leq k$ and let $n = \sum_{i=1}^{k} |b_i|$. Here, H denotes non-polar and P denotes polar amino acids respectively. Often, in what follows, the input string in our problem will be refer to as an HP string. An H-run in an HP string denotes the consecutive H's and a P-run denotes consecutive P's. So, the total number of H-runs is k and total number of H is n. An H-run of even (odd) length is said to be an even H-run (odd H-run). We will now define the valid embeddings and conformation of a protein into lattice.

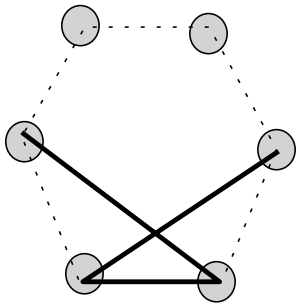

Figure 2.5: Crossing between binding edges; this situation is forbidden in a valid conformation.

Definition 1. *Let $p = p_1 \ldots p_t$ be an HP string of length t and let $G = (V,\ E)$ be a lattice. An embedding of p into G is a mapping function f: $\{1, \ldots, t\} \to V$ from the positions of the string to the vertices of the lattice. It assigns adjacent positions in p to adjacent vertices in G, $(f(i), f(i+1)) \in E$ for all $1 \leq i \leq t-1$. The edges $(f(i), f(i+1)) \in E$ for $1 \leq i \leq t-1$ are called **binding edges**. An embedding of p into G is called a conformation, if no two binding edges cross each other (see Fig. 2.5). This idea is called self-avoiding walk.* □

An embedding is a **self-avoiding walk** inside the grid. A walk in a graph is self-avoiding if their is no edge crossings in total walk.

Since, only hydrophobic- hydrophobic contacts are the energy source and all other interactions, namely, hydrophobic-polar, polar-polar and the interaction of solvent with any of those kinds are considered as neutral.

$$TotalEnergy, E = \sum_{(i!=j\ \&\ p_i,p_j \in H)}\ Contact(p_i, p_j)$$

For optimal embedding our main goal is to maximize the energy or other wisely we can say molecular stability. That means energy will be maximized if contacts are maximized.

Chapter 3

Literature Review

3.1 Introduction

Determining the structure of a protein is far from trivial. A protein with just five amino acids, could fold into 100 billion structures. Hence the lattice structures are introduced for presenting the simplified views of protein folding process by simplifying on following dimensions:

- Reduction of the level of detail at which protein sequences are represented
- Classification of the amino acids into classes
- Discretization of the conformational space
- Considering a simplified energy function

In the lattice models, an energy value is associated with every conformation taking into account particular neighbourhood relationships of the amino acids on the lattice. Consequently, given a lattice model L and a sequence s, the PSP (Protein Structure Prediction) problem is to find a conformation of s in L with minimal energy.

3.2 The HP Model Under Different Lattices

In 1998, P. Crescenzi et al. proved that protein folding problem in HP model is NP-hard for the 2D square lattice by reducing the Hamiltonian cycle problem to this problem [12]. In the same year Berger and Leighton proved that this computational problem was NP-Complete [5]. Hence the number of approximation algorithms increased using simplified

lattice structures over decades. Here we described some approximation algorithms of HP model on different lattices:

3.2.1 Square Lattice

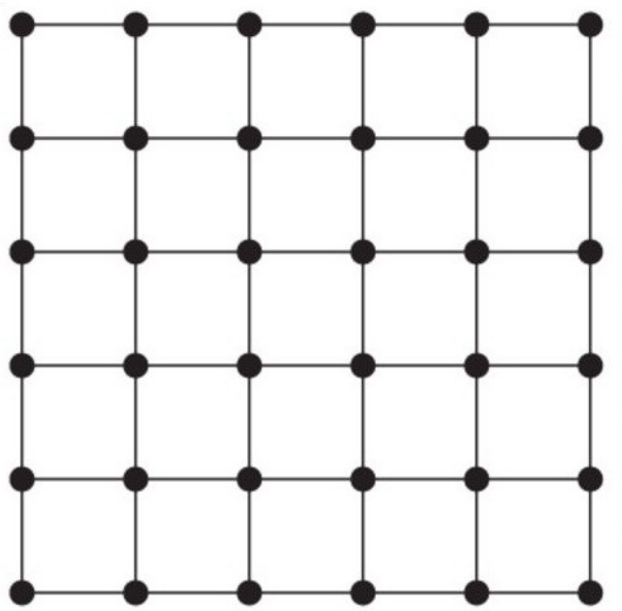

Figure 3.1: Square Lattice

The square lattice can be imagined as the set of points $\mathbb{Z}^2$ where two points, $x = (x_1, y_1)$ and $y = (x_2, y_2)$ are adjacent if $|x_1 - x_2| + |y_1 - y_2| = 1$. That means, x and y are adjacent if and only if the sum of the differences between the first and second coordinates of x and y is 1 (see Fig. 3.1).

Many algorithms have been developed on the 2D square lattice. Hart and Istrail gave an $\frac{1}{4}$-approximation algorithm for the problem on the 2D square lattice [20]. Mauri, Piccolboni, and Pavesi [40] give different approximation algorithm, also having the same approximation ratio of $\frac{1}{4}$; but they argue that it works better in practice. Later on, Newman [43] improved the approximation ratio to $\frac{1}{3}$ considering the conformation as a folded loop. A work on the

square lattice with side chains by Berger and Lighton achieves an approximation ratio of $\frac{1}{12}$ [5].

3.2.2 Cubic Lattice

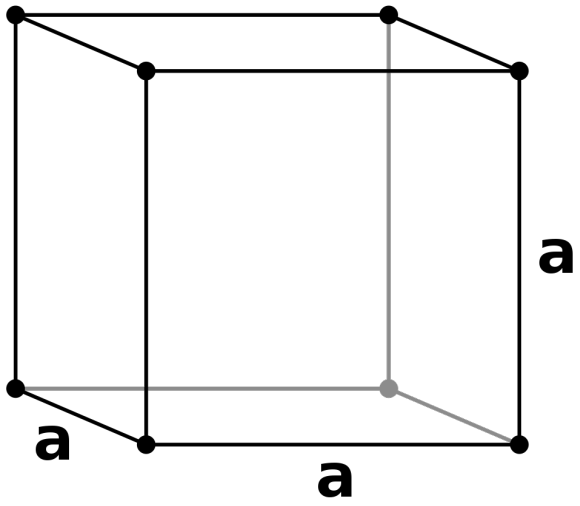

Figure 3.2: Cubic Lattice

The cubic lattice is the extension of the square lattice into three dimensions. We can express the set of points of the cubic lattice as $\mathbb{Z}^3$ with (0;0;0) as the origin. Here, two points x and y are adjacent if and only if the sum of the differences of their coordinates is 1 or $|x_1 - x_2| + |y_1 - y_2| + |z_1 - z_2| = 1$ (see Fig. 3.2).

A $\frac{3}{8}$-approximation algorithm for the problem on the cubic lattice was given by Hart and Istrail [20]. Later Newman and Ruhl improved this based on different geometric idea. They achieved an improved approximation ratio of .37501 [44] shows that $\frac{3}{8}$ is not the best approximation ratio guaranteed before. A similar work of side chain cubic lattice gives an algorithm with approximation ratio $\frac{4}{10}$[21]. An algorithm implemented in cubic lattice with diagonals introduced by Bockenhauer and Bongartz [8], gave $\frac{5}{8}$ approximation ratio.

3.2.3 Triangular lattice

The triangular lattice can be represented as the vertices of an infinite tessellation of equilateral triangles (see Fig. 3.3). A significant drawback of the square lattice and the cubic lattice is the parity problem. Parity problem can be defined as if two residues are at even

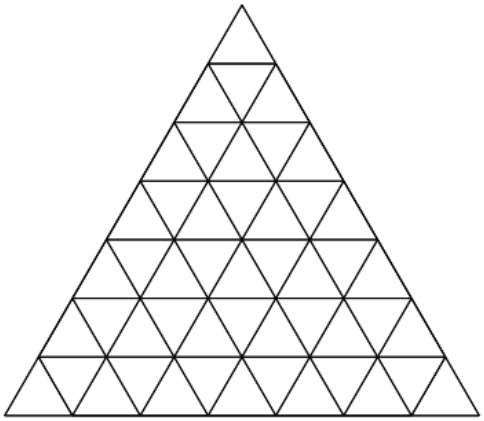

Figure 3.3: Triangular Lattice

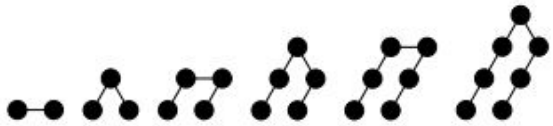

Figure 3.4: Any bead sequence can be implemented using triangular lattice

distance from one another in the sequence then they cannot be in topological contact with one another when the protein is embedded in the lattice. Agarwala et al. first suggest triangular lattice is more suitable to remove this parity problem [2]. They give an $\frac{1}{2}$ and an $\frac{6}{11}$ approximation algorithm using a better upper bound. Unlike the square lattice graph, the triangular lattice doesn't have the parity problem. As shown in the sequence in Fig. 3.4, we can force the endpoints of any 'bead sequence' of arbitrary length to be adjacent in the triangular lattice graph thereby eliminating the parity problem. In this lattice, Islam and Rahman gave an algorithm with an expected approximation ratio of $1 - \frac{2logn}{n-1}$ for $n \geq 6$, where n is the total number of H in a given HP string [27, 26].

3.2.4 Face centered cubic lattice or FCC lattice

FCC lattice is a more generalized 3 dimensional version of the triangular lattice proposed for protein folding by Agarwala et al [2] (see Fig. 3.5). He gives an approximation algorithm

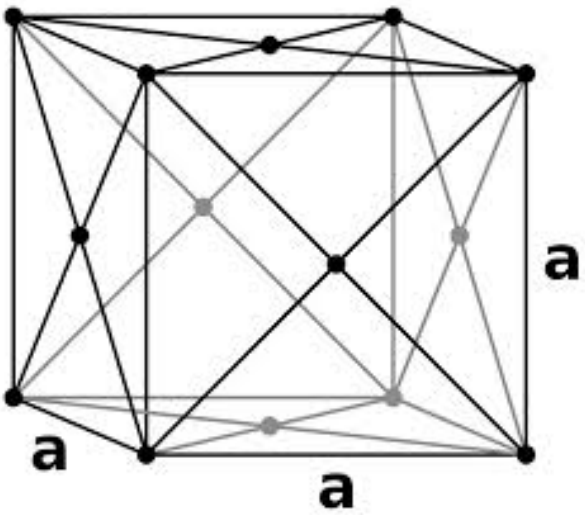

Figure 3.5: Face centered cubic lattice or FCC lattice

with ratio $\frac{3}{5}$. Heun [23] present algorithms with ratios $\frac{59}{70}$ and $\frac{37}{42}$. The second algorithm was designed for a natural subclass of proteins, which covers more than 99.5% of all sequenced proteins.

3.2.5 Hexagonal Lattice or Honeycomb Lattice

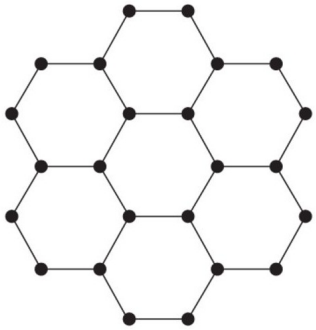

Figure 3.6: Hexagonal Lattice

2D hexagonal lattice (see Fig. 3.6) is a biologically meaningful alternative to the standard square lattice. To alleviate the problem of sharp turn Jiang and Zhu gave an approximation algorithm of ratio $\frac{1}{6}$ [29].

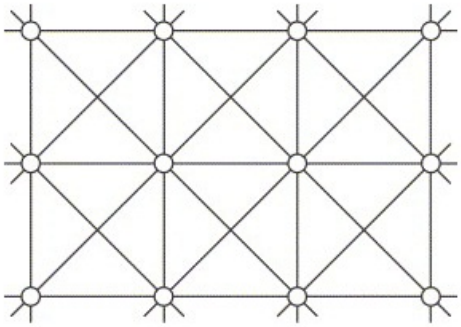

Figure 3.7: Square Lattice with Diagonals

3.2.6 Square Lattice with Diagonals

To remove the parity problem of the square lattice Bockenhauer and Bongartz introduce square lattice with diagonal [8] (see Fig. 3.7). They define square lattice with diagonals as an infinite graph $L_{2D} = (V, E)$ with vertex set $V = \mathbb{Z}^2$ and edge set $E = (x, x')|x, x' \in \mathbb{Z}^2, |x - x'|_2 \leq \sqrt{2}$, where $|.|_2$ denotes the Euclidean norm. They achieve an approximation ratio of $\frac{26}{15}$ in this lattice.

3.2.7 Hexagonal lattice with diagonals

The hexagonal lattice has the parity problem. Shaw et al. [46] remove this problem by introducing diagonals into hexagonal lattice and gave two approximation algorithms for protein folding on hexagonal lattice with diagonals (see Fig. 5.1). Their first algorithm is a $\frac{5}{3}$ approximation algorithm, which is based on the strategy of partitioning the entire protein sequence into two pieces. The next algorithm is also based on partitioning approaches and improves upon the first algorithm, achieving an approximation ration of $\frac{5}{4}$ [46].

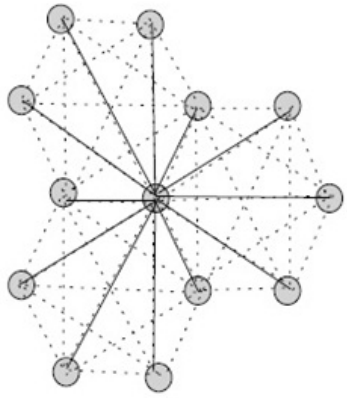

Figure 3.8: The Hexagonal Lattice with Diagonals

3.3 Heuristics approaches for protein folding problem

A number of heuristics and meta-heuristic techniques have also been applied to tackle the protein folding problem in the literature. A genetic algorithm for the protein folding problem in the HP model in 2D square lattice was proposed in [57]. In [24, 25], a hybrid genetic algorithm was presented for the HP model in 2D triangular lattice and 3D FCC lattice.

Different naturally inspired optimization algorithm has successfully applied for protein folding problem. Lin et al. [39] proposed an efficient hybrid Taguchi genetic algorithm. It combines genetic algorithm, Taguchi method, and particle swarm optimization, in order to enhance the performance of predicting protein structure. Zhang et al. [62] investigated the bacterial chemotaxis optimization (BCO) on the 2D lattice model. He compared BCO with standard genetic algorithm, immune genetic algorithm, and artificial immune system for various chain lengths. His result showed that the BCO (bacterial chemotaxis optimization) has the highest successful rate. Lately, several optimization heuristics inspired by bee colonies have been proposed. The two main approaches are the evolutionary algorithms and the foraging algorithms. The evolutionary approach was initially proposed by [54] and was based on the mating of bee drones with a queen bee. The foraging approach was proposed

simultaneously in [45] and [32] under the inspiration of honey bees. Dervis Karaboga claims that artificial bee colony (ABC) [32] performs better than genetic algorithm, differential evolution and particle swarm optimization. The reason is, normal global optimization techniques conduct only one search operation in one iteration, for example the particle swarm optimization carries out global search at the beginning stage and local search in the ending stage. On the other hand, ABC (artificial bee colony) conducts both global search and local search in each iteration. As a result the probability of finding the optimal is significantly increase. Authors of [48, 49, 50, 55] conducted their research on protein folding using ant colony optimization which was proposed by [7, 16, 15].

The authors in [38] first proposed the *pull move set* for the rectangular lattices, which was used in the HP model under a variety of local search methods, such as tabu search. They also showed the completeness and reversibility of the pull move set for the rectangular grid lattices. In [9, 56, 28], the authors extended the idea of the *pull move set* in the local search approach for finding an optimal embedding in the 2D triangular grid and the FCC lattice in 3D. Other local search approaches such as simulated annealing [19, 52, 3] used extensively in literature for protein folding problem.

Chapter 4

Protein Folding in the Hexagonal Prism Lattice with Diagonals

In this chapter, we introduce and define hexagonal prism lattice model with diagonals. Then, give two approximation algorithms for protein folding on this lattice. Our first algorithm leads us to a helix like structure which is commonly found in protein structure. Our first algorithm achieves and approximation ratio of 2. Our next algorithm is based on layer topology which improves the approximation ratio to $\frac{9}{7}$.

4.1 Definitions

In this section we formally define the hexagonal prism lattice with diagonals.

Definition 2. *The three-dimensional hexagonal prism lattice with diagonals is an infinite graph $G = (V, E)$ in the Euclidian Space with vertex set $V = R^3$ and edge set $E = \{(x, x') | x, x' \in R^3 \ , |x - x'| \leq 2\}$, where $|.|$ denotes the Euclidean norm. The hexagonal prism lattice is composed by stacking multiple two-dimensional hexagonal lattices with diagonals on top of each other. On a hexagonal prism lattice with diagonals each two-dimensional hexagonal lattice with diagonals is called a layer. The edges connecting the two layers are called layer edges. An edge $e \equiv (x, x') \in E$ is a non-diagonal edge or a non-diagonal layer edge iff $|x - x'| = 1$; otherwise it is a diagonal edge or diagonal-layer edge.* □

We use the well known notion of neighbourhood or adjacency from graph theory: two

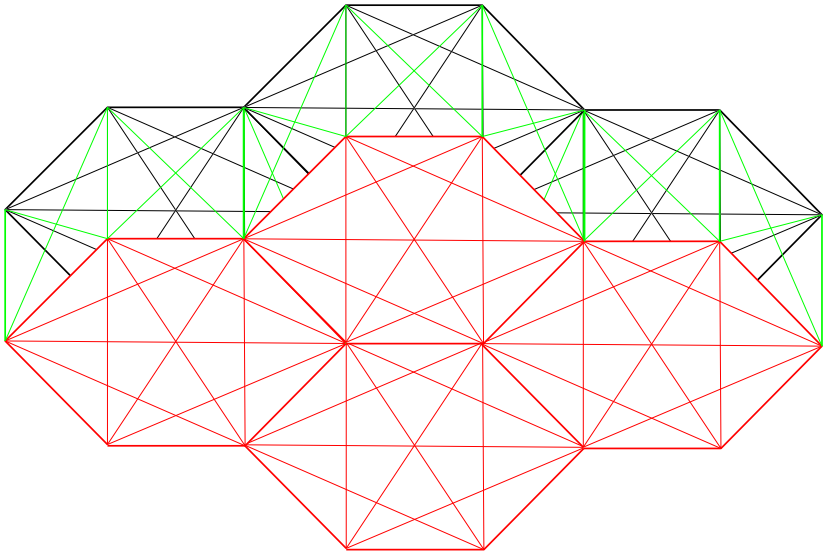

Figure 4.1: A hexagonal prism lattice with diagonals. Different layers are indicated using black and red color. Connecting edges between layers are indicated using green color.

vertices are adjacent/neighbour to each other if they are connected through an edge. In this connection, the difference between the usual hexagonal prism model and our propose model lies in the fact that a vertex in the former has 5 neighbours, whereas in the latter it has additional 15 neighbours, i.e., a total of 20 neighbours (see Fig. 4.1).

In a conformation, a vertex occupied by an H (P) will often be referred to as an H-vertex (a P-vertex). Fig. 4.2 shows an example of a conformation. Edges coloured blue are binding edges and all other edges between residues are non-binding edges. Throughout the paper, the H-vertices are indicated by filled circle and the P-vertices are indicated by blank circles.

Definition 3. *Given a conformation ϕ, an edge (x, x') of G is called a* ***contact edge****, if it is not a binding edge, but there exist $i, j \in \{1, \ldots, t\}$ such that $f(i) = x, f(j) = x'$, and $p_i = p_j = H$. The vertices of the lattice which are not occupied by an H or a P are called* ***unused vertices****. A binding edge connecting an H with a P is called an* ***alternating edge****.* ***Loss edge*** *is a non-binding edge incident to an H that is not a contact edge (see Fig. 4.3).* □

Now, we define the neighbourhood of an edge in the lattice.

Definition 4. *Let $e = (x, y)$ be any edge in G. We define the neighbourhood $N(e)$ of e as the intersection of the neighbours of its endpoints x and y.* □

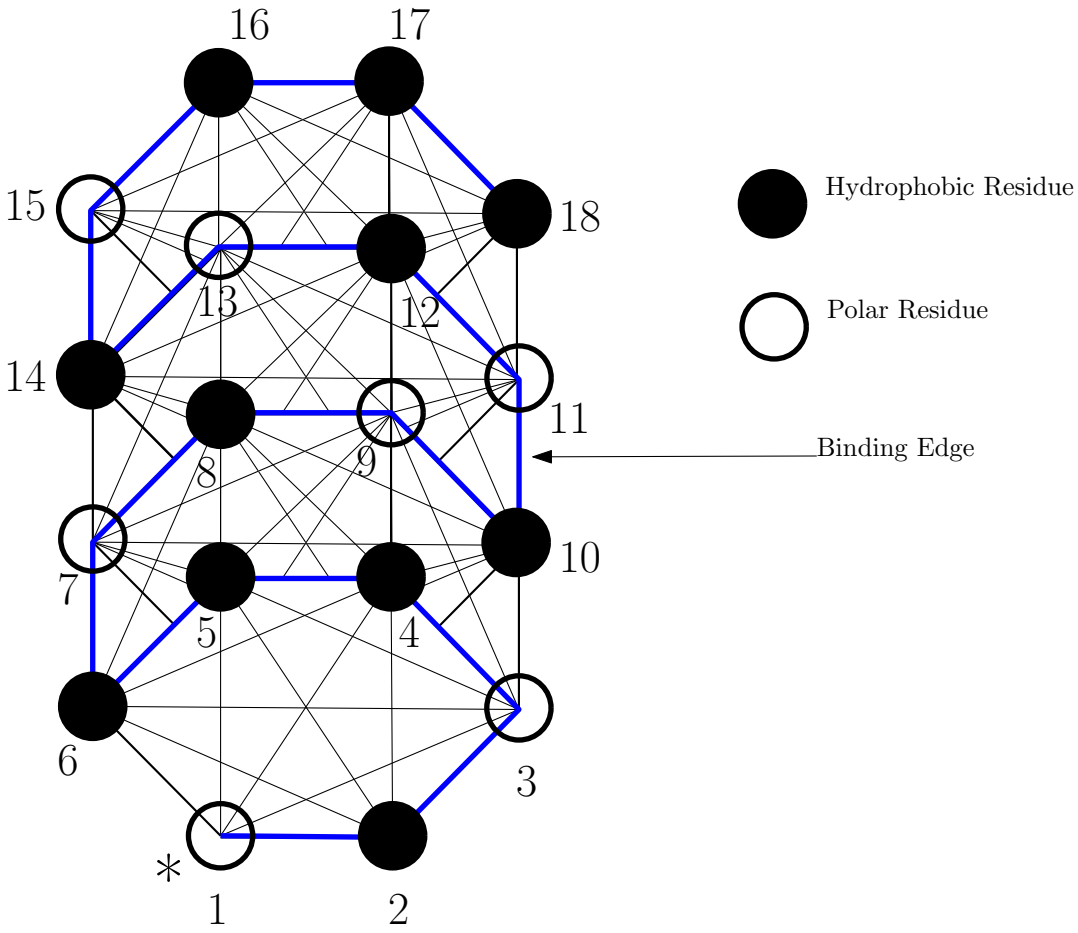

Figure 4.2: Conformation of PHPHHHPHPHPHPHPHHH on the lattice. * indicates the start symbol. The numbers in the figure is the position of beads in the string.

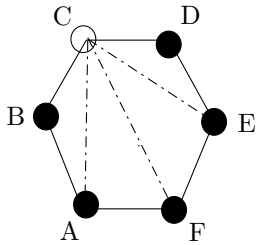

Figure 4.3: (C,D) and (B,C) are alternating edges; (A,C), (C,F) and (C,E) are loss edges.

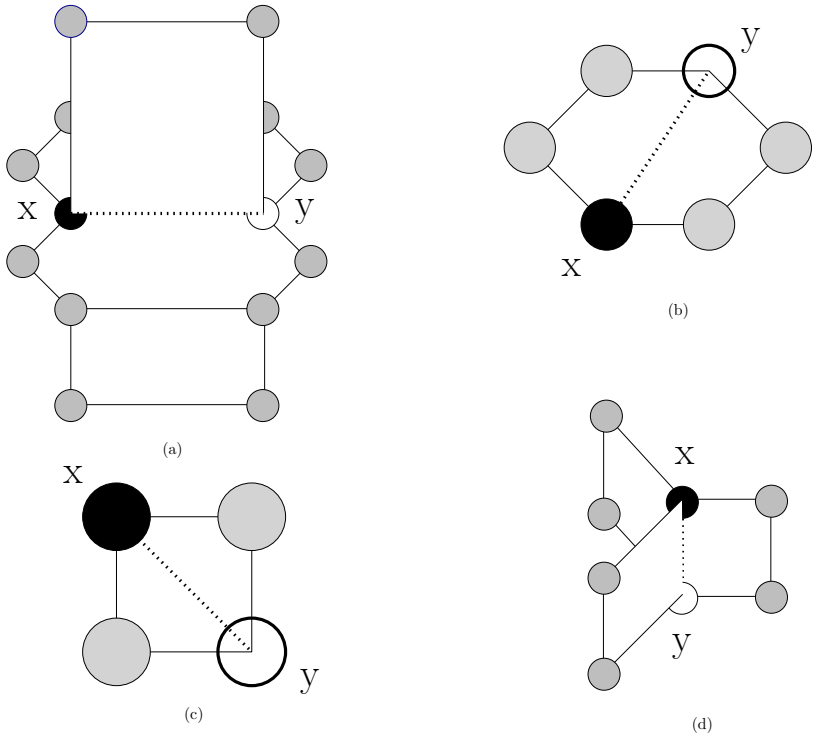

Figure 4.4: (a) 12 neighbours of the non-diagonal edge (x, y) (b) 4 neighbours of the diagonal edge (x, y) (c) 2 neighbours of the layer-diagonal edge (x, y) (d) 6 neighbours of the layer non-diagonal edge (x, y).

4.2 Our Approaches

4.2.1 Upper Bound

We will deduce a bound based on a simple counting argument: we will count the number of neighbours of a vertex in the lattice. We start with the following useful lemmas.

Lemma 1. *Let p be an HP string and $G = (V,\ E)$ is a hexagonal lattice with diagonals. If p has a conformation in G, then any H in p can have at most 18 contact edges.*

Proof: Every vertex in the lattice G has exactly 20 neighbours comprising 3 non-diagonal neighbours, 9 diagonal neighbours in one layer, 4 neighbours from the upper layer and 4 neighbours from lower layer (see Fig. 4.1). In this conformation, every H-vertex has exactly two binding edges. Hence 18 edges remain, which could potentially be contact edges. And hence the result follows. □

Lemma 2. *Let p be an input string for the problem and ϕ be a conformation of p. Let e $=(x, y)$ be a loss edge with respect to ϕ. Then there are at most four alternating edges in $N(e)$.*

Proof: From Fig. 4.4 if e is a non-diagonal edge, then $N(e)$ contains 12 vertices; if e is a diagonal edge, then $N(e)$ contains 4 vertices; if e is a layer-diagonal edge, then $N(e)$ contains 2 vertices; if e is a layer non-diagonal edge, then $N(e)$ contains 6 vertices. Again, each of x and y can be incident to at most two binding edges. So, there are at most four binding edges in $N(e)$. It follows immediately that there can be at most four alternating edges adjacent to e. □

Now we are ready to present the upper bound.

Lemma 3. *For a given HP string p, the the total number of contacts in a conformation ϕ is at most $18n - \frac{1}{2}k$, where k is the total number of H-runs and n is the total number of H.*

Proof : From Lemma 1, we know that the number of contacts is at most $18n$. In a confirmation one loss edge incident to H means that it would lose one contact edge. In what follows we will show that there will be at least $\frac{1}{2}k$ loss edges in ϕ. Since every H-run is preceded and followed by a total of two alternating edges, it is sufficient to prove that, for each alternating edge in ϕ for p, we have $\frac{1}{4}$ loss edge on average.
From Lemma 2 we know that, for every loss edge there will be at most four alternating edges in its neighbourhood. Alternatively, we can say that, for every four alternating edges there will be at least one loss edge, assuming that the alternating edges are in the neighbourhood of that loss edge. Clearly, if the alternating edges are not within the neighbourhood then the number of loss edges will increase. So, for every alternating edge there will be at least $\frac{1}{4}$ loss edge. There are a total of $2k$ alternating edges. So, the total number of loss edges will be, $\frac{1}{4} \times 2 \times k = \frac{1}{2}k$. Hence, the result follows. □

4.2.2 Algorithms and lower bounds

In this section, we present two novel approximation algorithms for the problem.

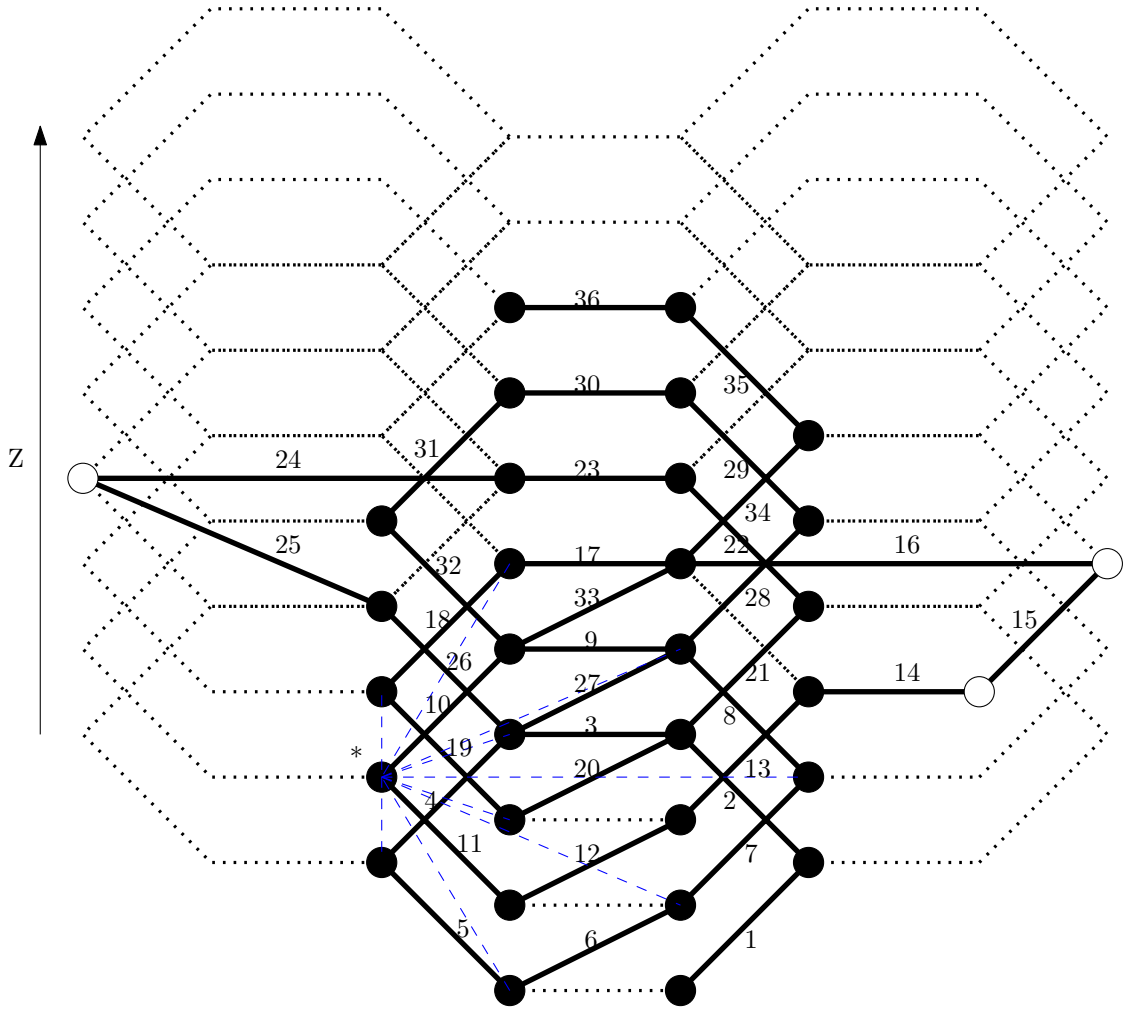

Figure 4.5: Folding of HP string $H^{14}P^2H^8P^1H^{11}$ by Algorithm HelixArrangement. Dotted black lines represent the lattice, solid lines represent the binding edges of the protein, blue dashed lines show 9 contacts of a H (identified by *). Binding edges are numbered sequentially. z indicates the direction of side layers of the upper layer.

4.2.2.1 Algorithm HelixArrangement

The idea of the first algorithm is to arrange all H's of the input string in a helix structure. The main difference between this new helix structure and conventional helix structure is that arrange P's of the input string outside of the main helix structure and put H only in the main helix structure. Fig. 4.5 shows the way we arrange H's and P's.

Algorithm HelixArrangement

Input: An HP string p.

1. Arrange the H's as follows:
   (a) Starting from a layer arrange the first six H's in a hexagon. Let, called this base hexagon.
   (b) Using the layer diagonal edge climb up to the upper layer. In this layer arrange the next six H's in a hexagon which is parallel to the base hexagon.
   (c) repeat Step (b) until the end of the string p. The hexagon where the process ends, let called that top hexagon.
2. Intermediate P-runs are arranged in the outer side of the hexagon in a layer (see Fig. 4.5)

4.2.2.2 Approximation ratio for Algorithm HelixArrangement

Except for the H's of the base hexagon and top hexagon an H can achieve at least 9 contacts as follows. An H from its layer achieves 3 contacts, from its immediate upper layer 3 contacts and from its immediate lower layer 3 contacts. H's of the base hexagon miss the contacts from the lower layer and H's of the top hexagon miss the contacts from the upper layer. So, there is in total 12 H in base hexagon and top hexagon combined which miss a total of $12 * 3$ *or* 36 contacts. Note that, it is possible that top hexagon is not filled with 6 H's. But it does not change any computation, because there is still 6 H's in the top hexagon and the immediate lower layer hexagon of top hexagon, which miss 3 contacts each.

Now, if we consider the P's arrangement, we will achieve two contacts for every alternating edge. If there is k alternating edges we will achieve $2k$ contacts.

So, for n H's total number of contacts ($\mathcal{C}$) can be achieved as follows:

$\mathcal{C} \geq 9n - 36 + 2k$

Hence we get the following approximation ratio A_1:

$$A_1 = \frac{18n - \frac{1}{2}k}{(9n - 36 + 2k)} \tag{4.1}$$

From Equation 4.1 it can be seen that for large n, A_1 tends to reach $\frac{18}{9}$ *or* 2. So we compute the value of k so that our approximation ratio is at most 2 as shown below.

$\frac{18n - \frac{k}{2}}{(9n-36+2k)} \leq \frac{18}{9}$

$\Rightarrow 81k \geq 18 \times 30 \times 2$

$\Rightarrow k \geq \frac{48}{3} \approx 16$

So, if the total number of H-runs is greater than 16, then Algorithm HelixArrangement will achieve an approximation ratio of 2. This can be summarized in Theorem later.

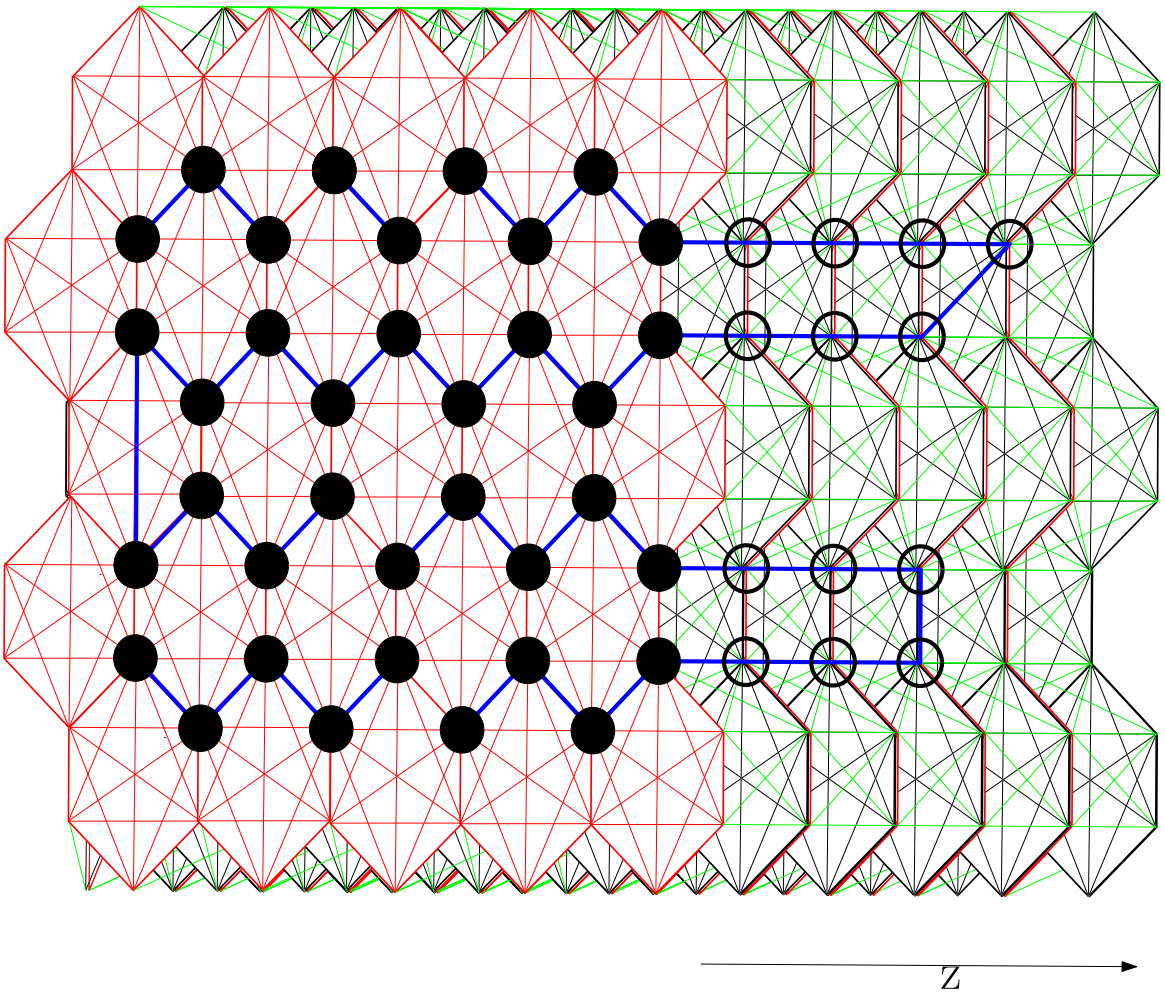

Figure 4.6: Folding of HP string $H^9P^6H^{18}P^7H^9$ by Algorithm LayerArrangement only in the Upper layer. Z indicates the direction of side layers of Upper layer

Theorem 1. *For any given HP string, Algorithm HelixArrangement gives a* 2 *approximation ratio for* $k > 16$*, where* k *is the total number of H-runs and* n *is the total number of H.*
□

4.2.2.3 Algorithm LayerArrangement

The idea of the second algorithm is to arrange all H's occurring in the input string along the two layers. We arrange the H's in the prefix of the string up to the $\lfloor \frac{n}{2} \rfloor$-th H on the upper layer and arrange the rest of those on the lower layer. In a layer, H-runs are arranged in a spiral manner. Then we arrange the P's between the H's outside these two layers. The arrangements of the P-runs outside the two layers are shown in Fig. 4.6. Within a layer the arrangement is done in chains (see Fig. 4.6). The arrangement in the upper (lower) layer can be further divided into nine regions, namely, the left region, the right region, the up region, the down region, the inside-left region, the inside-right region, the inside-up region, the inside-down region and the middle region (see Fig. 4.7).

Algorithm LayerArrangement
Input: An HP string p.

1. Set $f = \lfloor \frac{n}{2} \rfloor$.

2. Suppose F denotes the position in p after the f-th H. Denote by *pref* $F(p)$ the prefix of p up to position F and by *suff* $F(p)$ the suffix, that starts right after it. Now,

   (a) Arrange the H's in *pref* $F(p)$ in the upper layer as follows:

      i. Let, i and j are two integers that divide m_1 with reminder 0, such that the $|i - j|$ is minimal for all i and j. Let, $r = min(i, j)$, which is number of the chains in a layer. Let $s = \lfloor \frac{f}{r} \rfloor$, which is the number of residues in a chain. Suppose, S_1, S_2, S_3... denote the position in p after the s-th,$2s$-th,$3s$-th... H respectively. Denote, $S_i(p) = p_{S_{i-1}}...p_{S_i-1}$ for $i = 1, 2, 3....$ Here S_0 is starting position.

      ii. Now arrange $S_i(p)$ in chain one by one from top to bottom for $i = 1, 2, 3....$

      iii. Intermediate P-runs are arranged in the upper-side layers of the upper layer (see Fig. 4.6)

   (b) Arrange the H's in *suff* $F(p)$ along the lower layer following the same strategy spelled out in Step 2(a); intermediate P-runs are arranged in the lower-side layer of the lower layer (see Fig. 4.6).

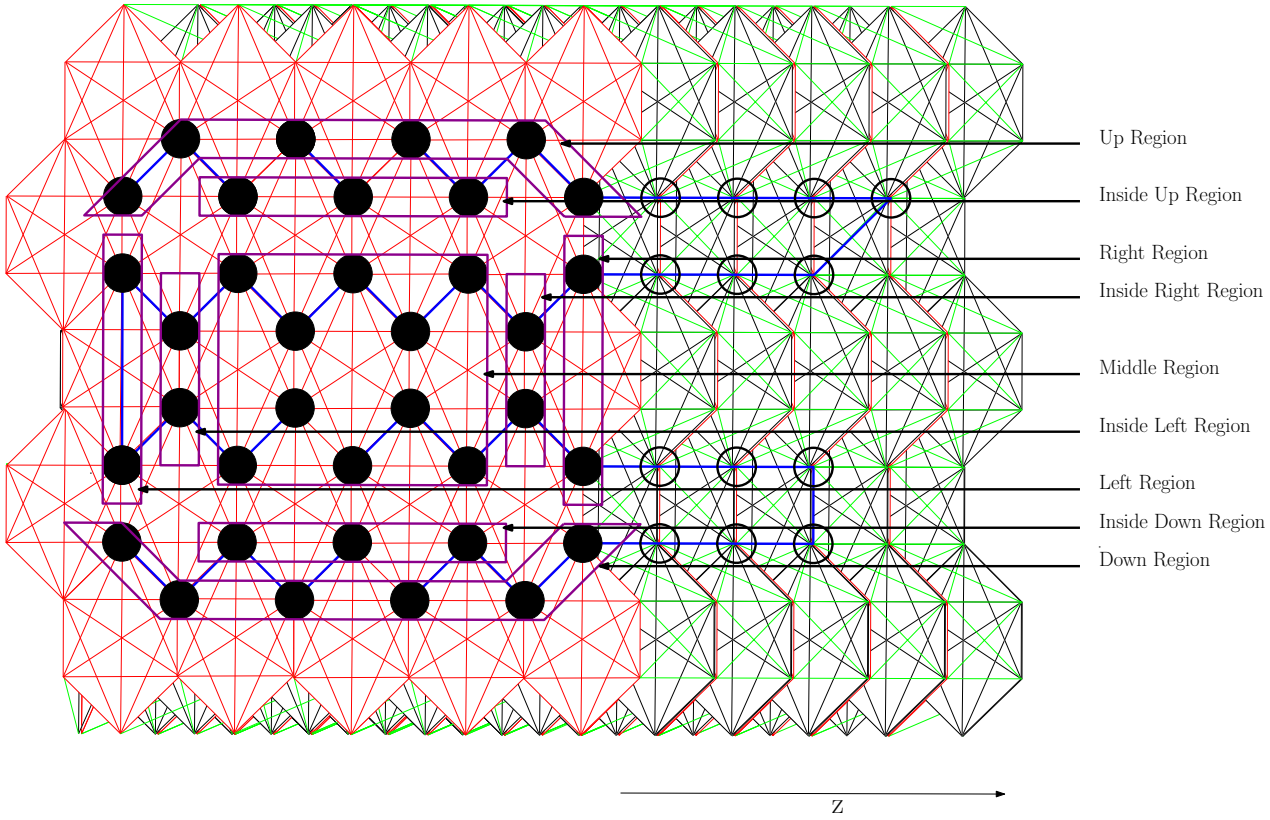

Figure 4.7: Divided into 9 region. They are up region, inside up region, right region, inside right region, middle region, inside left region, left region, inside down region, down region.

4.2.2.4 Approximation ratio for Algorithm LayerArrangement

Now we focus on deducing an approximation ratio for Algorithm LayerArrangement. Suppose that $m_1 = \lfloor \frac{n}{2} \rfloor$. So, according to Algorithm LayerArrangement, the upper (lower) layer will contain m_1 (m_1 or $m_1 + 1$) H's. We consider two cases, namely, where m_1 is odd, i.e., $m_1 = 2x + 1$ and m_1 is even, i.e., $m_1 = 2x$, with an integer $x > 0$.

Now, let, i and j are two integers that divide m_1 with remainder 0, such that $|i - j|$ is minimal for all i and j. Let, $r = min(i, j)$, which is the number of the chains in a layer. Now, let, $s = m_1/r$, which is the number of residues in a chain. The chains are arranged spirally in a layer.

In what follows, we will use vw-upper layer (vw-lower layer) to denote a particular region of the upper (lower) layer. So, vw could be one of the 9 options, namely, lR (left region), rR (right region), uR (up region), dR (down region), i_lR (inside-left region), i_rR (inside-right

region), i_uR (inside-up region), i_dR (inside-down region) and mR (middle region). We also use ϕ_{LA} to refer to the conformation given by Algorithm LayerArrangement.

The analysis here will be easy to understand with the help of Fig. 4.7. In ϕ_{LA}, every vertex in the lR-up layer and rR-up layer has at least 8 contacts. Every vertex in the i_lR-upper layer and the i_rR-upper layer has at least 12 contacts. For each of lR-upper layer, rR-upper layer, i_lR-upper layer and the i_rR-upper layer, there are $r-2$ such vertices (see Fig. 4.7). Every vertex in the uR-upper layer and the dR-upper layer has at least 6 contacts. There are $\frac{s+3}{2}$ such vertices for each of the uR-upper layer and the dR-upper layer. Every vertex in the i_uR-upper layer and the i_dR-upper layer has at least 11 contacts. There are $(\frac{s-3}{2})$ such vertices for each of the i_uR-upper layer and the i_dR-upper layer. So there remain $(rs-2r-2s-4)$ vertices in the upper layer which is arranged in mR-upper layer, where every vertex achieves 14 contacts.

So, the total number of contacts ($\mathcal{C}$) of all the vertices of the upper layer can be computed as follows:

$$\mathcal{C} \geq 2\times 8\times (r-2)+2\times 12\times (r-2)+2\times 6\times \tfrac{s+3}{2}+2\times 11\times (\tfrac{s-3}{2})+14\times (2x-2r-2s-4)$$
$$\Rightarrow \mathcal{C} \geq 16r-32+24r-48+6s+18+11s-33+14sr-28r-28s-56$$
$$\Rightarrow \mathcal{C} \geq 14sr+12r-11s-151$$
$$\Rightarrow \mathcal{C} \geq 14m_1+12r-11s-151$$
$$\Rightarrow \mathcal{C} \geq 7n+12r-11s-151$$

Since the upper layer is symmetric to the lower layer, both layer will have the same number of vertices if $n=2m_1$. So all the vertices of the lower layer will also have at least $\mathcal{C}$ contacts. So the total number of contacts will be at least $2\mathcal{C}$ or $14n+24r-22s-302$.

If $n=2m_1+1$, then let $n_1=n-1$. This n_1 vertices will have at least $14n_1+24r-22s-302$ contacts. The remaining vertex will have at least 2 contacts. So the total number of contacts will be at least $14(n-1)+24r-22s-302+2$ or $14n+24r-22s-314$. So, combining the two cases, we get that the total number of contacts is at least $14n+24r-22s-314$. Now we need to take the alternating edges into our consideration. For every alternating edge we get two extra contacts for the two vertices (each having one). So, for n H's and k alternating edges we get a total of at least $14n+24r-22s-314+2k$ contacts. Hence we

get the following approximation ratio A_2:

$$A_2 = \frac{18n - \frac{1}{2}k}{(14n + 24r - 22s - 314 + 2k)} \tag{4.2}$$

From Equation 4.2 it can be seen that for large n, A_2 tends to reach $\frac{18}{14}$. So we compute the value of k so that our approximation ratio is at most $\frac{18}{14}$ as shown below.

$\frac{18n-\frac{k}{2}}{(14n+24r-22s-314+2k)} \leq \frac{18}{14}$

$\Rightarrow 14 \times 18n - \frac{k}{2} \leq \frac{18}{(14n+24r-22s-314+2k)}$

$\Rightarrow 252n - 7k \leq 252n + 432r - 396s - (314 \times 18) + 36k$

$\Rightarrow 43k \geq 36(11s - 12r) + (314 \times 18) \quad \Rightarrow k \geq \frac{36(11s-12r)+(314\times18)}{43}$

Now, from this case if $11s = 12r$, $k \geq \frac{(314\times18)}{43} \approx 132$

So, if the total number of H-runs is greater than 132, then Algorithm LayerArrangement will achieve an approximation ratio of $\frac{18}{14}$ or $\frac{9}{7}$ for $11s = 12r$.

Note that, the value of k is dependent on n and the HP string. We now deduce the expected value of k for a given HP string. This problem can be mapped into the problem of *Integer Partitioning* as defined below. Notably, similar mapping has recently been utilized in [26][27][46] for deriving an expected approximation ratio of some other algorithms.

Problem 1. Given an integer Y, the problem of Integer Partitioning aims to provide all possible ways of writing Y, as a sum of positive integers.

Note that the ways that differ only in the order of their summands are considered to be the same partition. A summand in a partition is called a part. Now, if we consider n as the input of Problem 1 (i.e., Y) then each length of H-runs can be viewed as parts of the partition. So if we find the expected number of partitions we in turn get the expected value of k. Kessler and Livingston [33] showed that to get an integer partition of an integer Y, expected number of required parts is:

$$\sqrt{\frac{3Y}{2\pi}} \times (\log Y + 2\gamma - 2\log\sqrt{\frac{\pi}{6}}),$$

where γ is the famous Euler's constant. For our problem $Y = n$. If we denote $E[P]$ as the expected number of H-runs then,

$$E[P] = \sqrt{\frac{6}{\pi}} \times \sqrt{n} \times (\frac{1}{2}\log n + \gamma - \log\sqrt{\frac{\pi}{6}}).$$

Now, as $(\frac{1}{2}\log n + \gamma - \log\sqrt{\frac{\pi}{6}}) \leq (\sqrt{\frac{2\pi}{3}} \times \frac{1}{2}\log n)$ for $n \geq 5$, we can say that

$$E[P] \leq \sqrt{n} \times \log n.$$

So the expected value of k is less than or equal to $\sqrt{n} \times \log n$ which implies that $\sqrt{n} \times \log n \geq 132$ or $n \geq 500$. Now, if $11s > 12r$, the lower bound of k increases, as a result the expected lower bound of n will increase. On the other side, if $11s < 12r$, expected lower bound of n will decrease. The above findings are summarized in the following theorems.

Theorem 2. *For any given HP string, Algorithm LayerArrangement achieves an approximation ratio of $\frac{9}{7}$ for $k > 132$, where k is the total number of H-runs, $11s = 12r$ and $n = 2rs$ is the total number of H.* □

Theorem 3. *For any given HP string, Algorithm LayerArrangement is expected to achieve an approximation ratio of $\frac{9}{7}$ for $n \geq 500$ and $11s = 12r$ where, $n = 2rs$ is the total number of H.* □

4.3 Discussion and Conclusions

One vertex in the SC (Simple Cubic) lattice has 6 neighbours, and in FCC (Face Centered Cubic) or BCC (Body Centered Cubic) lattice it has 14 neighbours. On the other hand, one vertex of the hexagonal prism lattice with diagonals have 20 neighbours which property leads us to find better approximation ratio for our algorithm for protein folding. On the other hand this lattice model removes some well known problems of protein folding in SC lattice, e.g., parity problem. Considering such properties of this lattice we believe that more algorithms could be developed in this lattice. Also heuristics algorithms can be applied on this lattice, which can lead us to better result.

Chapter 5

Weighted Contact Analysis in Hexagonal Lattice with Diagonals

In this chapter, we extend the hexagonal lattice model with diagonals with the concept of weighted contacts. This lattice model was first proposed by Shaw et al. [47]. In [47] they gave an upper bound for the total number contact edges and an approximation algorithm having an approximation ratio of $\frac{10}{3}$, which is based on the strategy of partitioning the entire protein sequence into two pieces. Our new approach for analysis improve the ratio to 1.96.

5.1 Review of the lattice structure of [46, 47]

In this section, we briefly review the notions and notations to describe the hexagonal lattice model with diagonals introduced in [46, 47] (See Fig. 5.1).

Definition 5. *The two-dimensional hexagonal lattice with diagonals is an infinite graph* $G = (V, E)$ *in the Euclidian Space with vertex set* $V = R^2$ *and edge set* $E = \{(x, x')|x, x' \in R^2 \ , |x - x'| \leq 2\}$*, where* $|.|$ *denotes the Euclidean norm. An edge* $e \equiv (x, x') \in E$ *is a non diagonal edge iff* $|x - x'| = 1$*; otherwise it is a diagonal edge. We distinguish between 2 types of diagonal edges, one having length less than 2 (i.e.,* $|x - x'| < 2$*) and the other having length equal to 2 (i.e.,* $|x - x'| = 2$*). The former is refered to as the small diagonal edge and the latter as the big diagonal edge.*

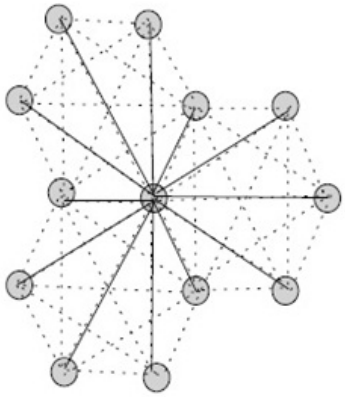

Figure 5.1: The Hexagonal Lattice with Diagonals

5.2 Concept of weighted contact

Further refinement of this model concerns the intensity of the chemical forces along the different types of edges. The chemical binding force directly depends on the distance between the two adjacent amino acids. The greater the distance the smaller the chemical binding force. The three fundamental noncovalent chemical binding forces are, electrostatic interactions, hydrogen bonds, and van-der-waals interactions. They differ in geometry, strength, and specificity. Furthermore, these bonds are greatly affected in different ways by the presence of water. According to *Coulomb's* law, electrostatic interaction between two atoms depends on the electric charges on atoms and inversely proportional to their distance.

In our assumption, the chemical binding force between the two amino acids which are connected via a big diagonal edge is smaller than that between the two amino acids which are connected via small diagonal edge. We introduce two additional parameters α_1 and α_2 $(0 \leq \alpha_1 \leq \alpha_2 \leq 1)$ to measure the loss of binding power relative to the binding power given by non-diagonal edges. Here we will count weighted contact of 1 for each non-diagonal contact edge, weighted contact of α_1 for big diagonal contact edge, weighted contact of α_2 for small diagonal contact edge.

5.3 Analysis with the concept of weighted contact

In this section, we analysis the algorithm of [47, 46] incorporating the concept of weighted contact. Hexagonal lattice with diagonals is used for the algorithm. We start with deducing two upper bounds on the number of possible contacts for any H in the HP string.

5.3.1 An upper bound

We will deduce a bound based on a simple counting argument: we will count the number of neighbours of a vertex in the lattice. We start with the following useful lemmas.

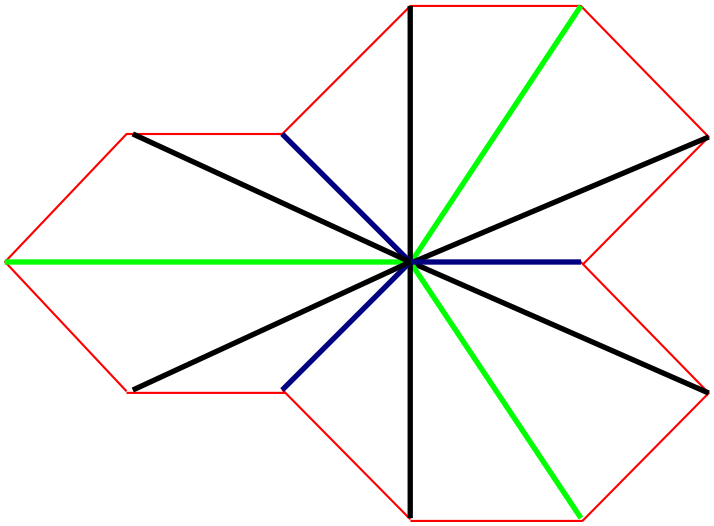

Figure 5.2: Every vertex in the lattice has 12 neighbours comprising 3 non-diagonal neighbours (blue lines), 3 big diagonal neighbours (green lines) and 6 small diagonal neighbours (black lines)

Lemma 4. *Let p be an HP string and $G = (V,\ E)$ be a hexagonal lattice with diagonals. If p has a conformation in G, then the weighted contact of any H in p can be at most $3+\alpha_1+6\alpha_2$.*

Proof: Every vertex in the lattice G has exactly twelve neighbours comprising 3 non-diagonal neighbours, 3 big diagonal neighbours and 6 small diagonal neighbours (See Fig. 5.2). In this conformation, every H-vertex has exactly two binding edges. If the binding edges are big diagonal edges, then we get the minimum loss of weighted contact. Hence 3 non-diagonal edges, 1 big diagonal edge and 6 small diagonal edges cause the maximum weight of contact edges on a vertex. And hence the result follows. □

Lemma 5. *For a given HP string p, the total weighted contact in a conformation ϕ is at most $(3+\alpha_1+6\alpha_2)n-\frac{1}{2}k$, where k is the total number of H-runs and n is the total number of H.*

Proof : From Lemma 4, we know that weighted contact for an H is at most $3+\alpha_1+6\alpha_2$. In a confirmation one loss edge incident to H means that it would lose one contact edge. In what follows we will show that there will be at least $\frac{1}{2}k$ loss edges in ϕ. Since every H-run is preceded and followed by a total of two alternating edges, it is sufficient to prove that, for each alternating edge in ϕ for p, we have $\frac{1}{4}$ loss edge on average.

If e is a non-diagonal edge, then $N(e)$ contain 12 vertices; if e is a diagonal edge, then $N(e)$ contain 4 vertices; if e is a layer-diagonal edge, then $N(e)$ contain 2 vertices; if e is a layer non-diagonal edge, then $N(e)$ contain 6 vertices. Again, each of x and y can be incident to at most two binding edges. So, there are at most four binding edges in $N(e)$. So, for every loss edge there will be at most four alternating edges in its neighbourhood. Alternatively, we can say that, for every four alternating edges there will be at least one loss edge, assuming that the alternating edges are in the neighbourhood of that loss edge. Clearly, if the alternating edges are not within the neighbourhood then the number of loss edges will increase. So, for every alternating edge there will be at least $\frac{1}{4}$ loss edge. There are a total of $2k$ alternating edges. So, the total number of loss edges will be, $\frac{1}{4}\times 2\times k = \frac{1}{2}k$. Hence, the result follows. □

5.3.2 Algorithms lower bounds

In this section, we first briefly review the algorithm of [46, 47], which is called Algorithm ChainArrangement. Then, we deduce the approximation ratio incorporating the concept of weighted contact for Algorithm ChainArrangement developed in [46, 47]. Breifly, the idea of this algorithm is to arrange all H's occurring in the input string along the two chains. The algorithm arrange the H's in the prefix of the string up to the $\lfloor\frac{n}{2}\rfloor$-th H on the left chain and arrange the rest of those on the right one (see Fig. 5.3). The next step of the algorithm is to arrange the P's between H's outside these two chains. The arrangements of the P-runs along the side-arms of the two chains are shown in Fig. 5.3.

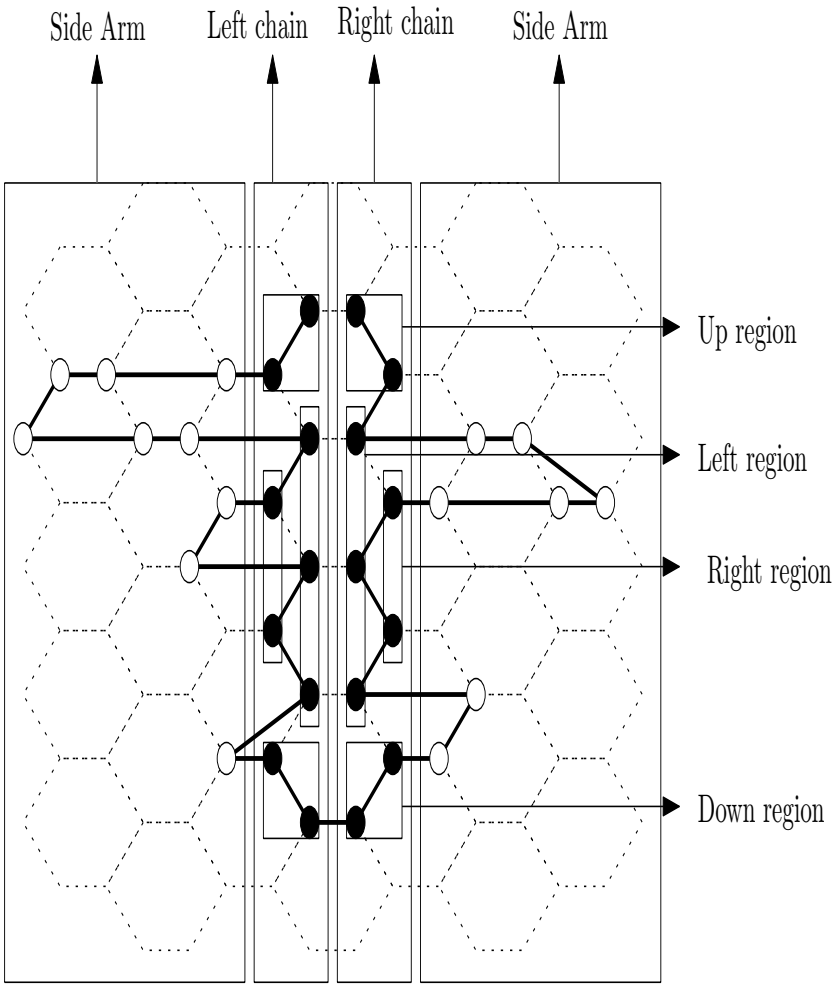

Figure 5.3: Folding of HP string $H^2P^6H^2P^2H^3P^1H^4P^2H^4P^5H^3$ by Algorithm ChainArrangement. The concept of the figure borrowed from [46, 47].

5.3.3 Approximation ratio by weighted contact for Algorithm ChainArrangement

Now we focus on deducing an approximation ratio for Algorithm ChainArrangement using the concept of weighted contact. Suppose that $m_1 = \lfloor \frac{n}{2} \rfloor$. So, according to Algorithm ChainArrangement, the left (right) chain will contain m_1 (m_1 or $m_1 + 1$) H's. We need to consider two cases, namely, where $m_1 = 2x + 1$ and $m_1 = 2x$, with an integer $x > 0$. In what follows, we will use vw-left chain (vw-right chain) to denote a particular region of the left (right) chain. So, vw could be one of the 4 options, namely, lR (left region), rR (right region), uR (up region) and dR (down region). We also use ϕ_{CA} to refer to the conformation given by Algorithm ChainArrangement.

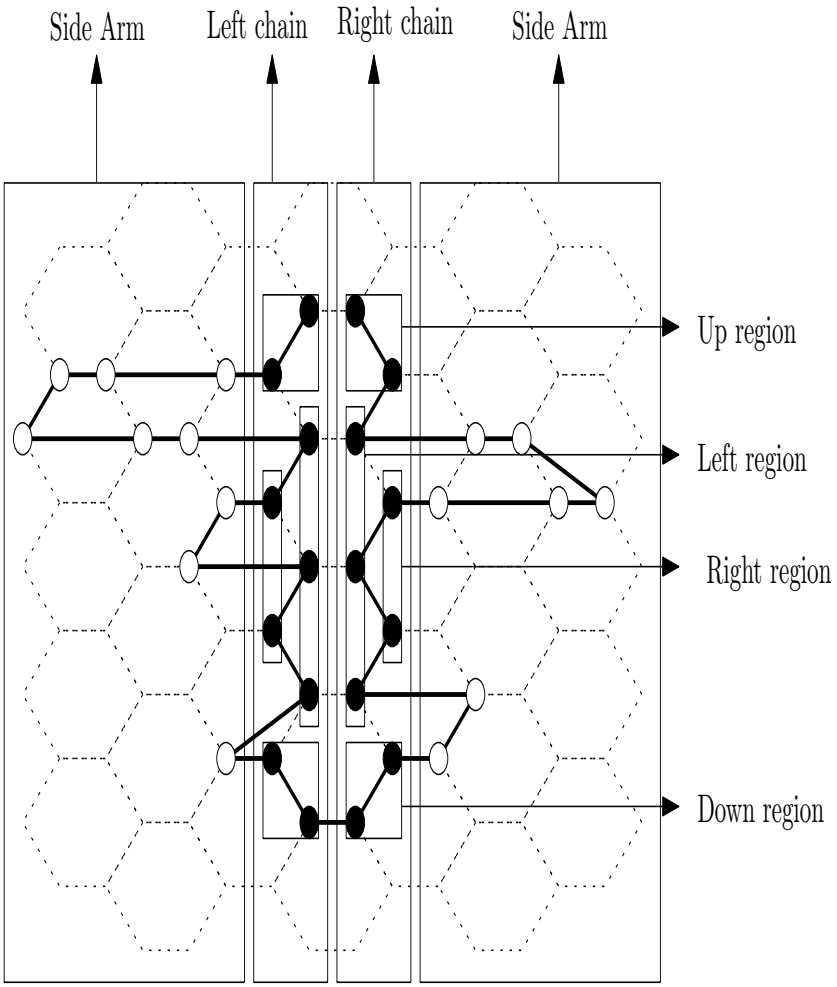

Figure 5.4: Showing different regions of the left chain and the right chain for $m_1 = 2x + 1$. The concept of the figure borrowed from [46, 47]

5.3.3.1 case 1: $m_1 = 2x + 1$

The analysis for this case will be easier to understand with the help of Fig. 5.4 and Table 5.1. First, let, n is even.

So, the total weighted contact ($\mathcal{C}$) of all the vertices in the left chain, can be computed as follows:

$\mathcal{C} \geq (x-2) \times (4\alpha_2 + \alpha_1) + (x-1) \times (4\alpha_2 + 2\alpha_1 + 1) + (2\alpha_2 + \alpha_1 + 1) + (3\alpha_2 + \alpha_1) + (2\alpha_2 + \alpha_1) + (3\alpha_2 + \alpha_1)$

$\Rightarrow \mathcal{C} \geq x(4\alpha_2 + \alpha_1) + x(4\alpha_2 + 2\alpha_1 + 1) - 2(4\alpha_2 + \alpha_1) - (4\alpha_2 + 2\alpha_1 + 1) + (2\alpha_2 + \alpha_1 + 1) + (3\alpha_2 + \alpha_1) + (2\alpha_2 + \alpha_1) + (3\alpha_2 + \alpha_1)$

Table 5.1: Weighted contact for different regions of the left chain, when number of vertex $m_1 = 2x + 1$

Region	nummber of vertex $m_1 = 2x + 1$	non-diagonal edge ($weight = 1$)	small diagonal edge ($weight = \alpha_2$)	big diagonal edge ($weight = \alpha_1$)
lR-left chain	$x - 2$	0	4	1
rR-left chain	$x - 1$	1	4	2
uR-left chain	1	1	2	1
	1	0	3	1
dR-left chain	1	0	2	1
	1	0	3	1

$\Rightarrow \mathcal{C} \geq x(8\alpha_2 + 3\alpha_1 + 1) - 8\alpha_2 - 2\alpha_1 - 4\alpha_2 - 2\alpha_1 - 1 + 10\alpha_2 + 4\alpha_1 + 1$
$\Rightarrow \mathcal{C} \geq x(8\alpha_2 + 3\alpha_1 + 1) - 2\alpha_2$
$\Rightarrow \mathcal{C} \geq \frac{1}{2}(2x + 1)(8\alpha_2 + 3\alpha_1 + 1) - 2\alpha_2 - \frac{1}{2}(8\alpha_2 + 3\alpha_1 + 1)$
$\Rightarrow \mathcal{C} \geq \frac{1}{2}m_1(8\alpha_2 + 3\alpha_1 + 1) - 2\alpha_2 - \frac{1}{2}(8\alpha_2 + 3\alpha_1 + 1)$
$\Rightarrow \mathcal{C} \geq \frac{1}{4}n(8\alpha_2 + 3\alpha_1 + 1) - 2\alpha_2 - \frac{1}{2}(8\alpha_2 + 3\alpha_1 + 1)$

Since the right chain is symmetric to the left one, both chains will have the same number of vertices if $n = 2m_1$, i.e., all the vertices of the right chain will also have at least $\mathcal{C}$ weighted contact. So the total weighted contact will be at least $2\mathcal{C}$ or $\frac{1}{2}n(8\alpha_2+3\alpha_1+1)-12\alpha_2-3\alpha_1-1$.

If $n = 2m_1+1$ then let $n_1 = n-1$. The total weighted contact for these n_1 vertices will be at least $\frac{n_1}{2}(8\alpha_2+3\alpha_1+1)-12\alpha_2-3\alpha_1-1$. The remaining vertex will have at least $2\alpha_2$ weighted contact. So the total weighted contact will be at least $\frac{n-1}{2}(8\alpha_2+3\alpha_1+1)-12\alpha_2-3\alpha_1-1)+2\alpha_2$ or $\frac{n}{2}(8\alpha_2 + 3\alpha_1 + 1) - 14\alpha_2 - \frac{9}{2}\alpha_1 - \frac{3}{2}$.

5.3.3.2 case 2: $m_1 = 2x$

The analysis for this case will be easier to understand with the help of Fig. 5.5 and Table 5.2. Firstly, let, n is even.

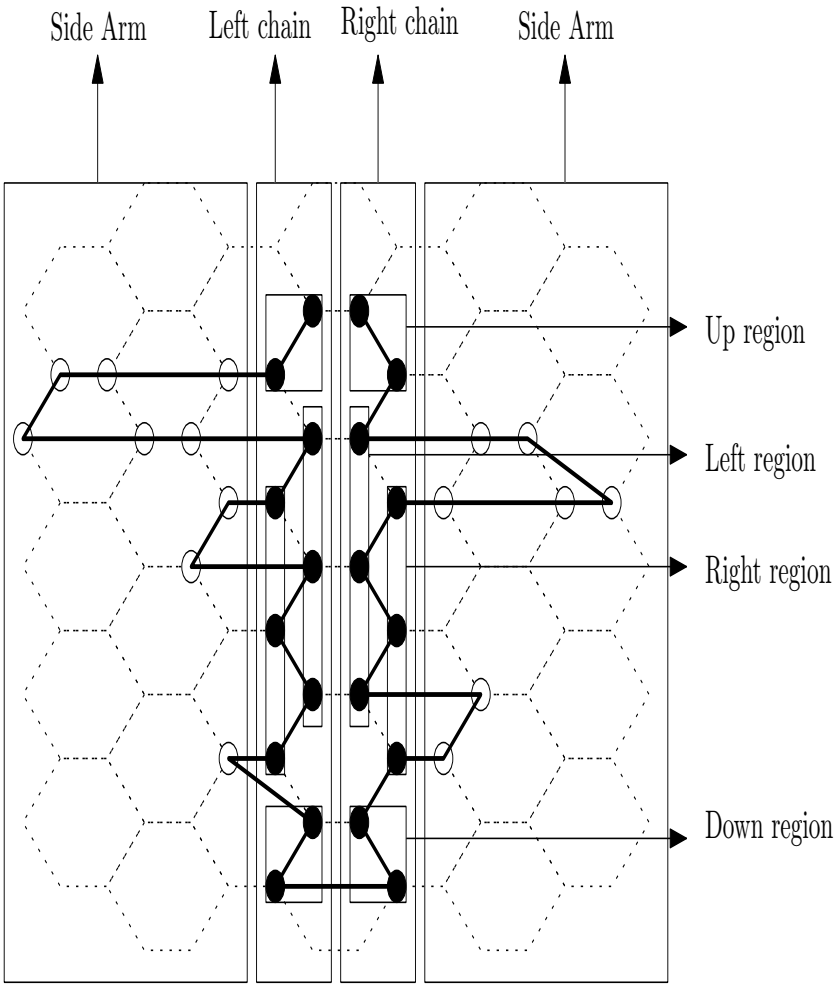

Figure 5.5: Showing different portion of left chain and right chain for $m_1 = 2x$. The concept of the figure borrowed from [46, 47].

So, the total weighted contact ($\mathcal{C}$) of all the vertices in the left chain, can be computed as follows:

$\mathcal{C} \geq (x-2)\times(4\alpha_2+\alpha_1)+(x-2)\times(4\alpha_2+2\alpha_1+1)+(2\alpha_2+\alpha_1+1)+(3\alpha_2+\alpha_1)+(3\alpha_2+\alpha_1+1)+2\alpha_2$

$\Rightarrow \mathcal{C} \geq x(4\alpha_2+\alpha_1)+x(4\alpha_2+2\alpha_1+1)-2(4\alpha_2+\alpha_1)-2(4\alpha_2+2\alpha_1+1)+(2\alpha_2+\alpha_1+1)+(3\alpha_2+\alpha_1)+(3\alpha_2+\alpha_1+1)+2\alpha_2$

$\Rightarrow \mathcal{C} \geq x(8\alpha_2+3\alpha_1+1)-6\alpha_2-3\alpha_1$

$\Rightarrow \mathcal{C} \geq \frac{1}{2}m_1(8\alpha_2+3\alpha_1+1)-6\alpha_2-3\alpha_1$

$\Rightarrow \mathcal{C} \geq \frac{1}{4}n(8\alpha_2+3\alpha_1+1)-6\alpha_2-3\alpha_1$

Table 5.2: Weighted contact for different regions of the left chain, when number of vertex $m_1 = 2x$

Region	no. of vertex $m_1 = 2x$	non-diagonal edge ($weight = 1$)	small diagonal edge($weight = \alpha_2$)	big diagonal edge($weight = \alpha_1$)
lR-left chain	$x-2$	0	4	1
rR-left chain	$x-2$	1	4	2
uR-left chain	1	1	2	1
	1	0	3	1
dR-left chain	1	1	2	1
	1	0	2	0

Since the right chain is symmetric to the left one, both chains will have the same number of vertices if $n = 2m_1$, i.e., the total weighted contact considering all the vertices of the right chain will also be at least $\mathcal{C}$.

So the total weighted contact will be at least $2\mathcal{C}$ or $\frac{1}{2}n(8\alpha_2 + 3\alpha_1 + 1) - 12\alpha_2 - 6\alpha_1$.

If $n = 2m_1 + 1$ then let $n_1 = n - 1$. This n_1 vertices will have at least $\frac{n_1}{2}(8\alpha_2 + 3\alpha_1 + 1) - 12\alpha_2 - 6\alpha_1$ weighted contact. The remaining vertex will have at least $2\alpha_2$ weighted contact.

So the total weighted contact will be at least $\frac{n-1}{2}(8\alpha_2 + 3\alpha_1 + 1) - 12\alpha_2 - 6\alpha_1 + 2\alpha_2$ or $\frac{n}{2}(8\alpha_2 + 3\alpha_1 + 1) - 14\alpha_2 - \frac{15}{2}\alpha_1 - \frac{1}{2}$.

So, combining the two cases, we get that the total weighted contact is at least $\frac{n}{2}(8\alpha_2 + 3\alpha_1 + 1) - 14\alpha_2 - \frac{15}{2}\alpha_1 - \frac{1}{2}$. Now we need to take the alternating edges into our consideration. For every alternating edge we get two extra weighted contact for the two vertices (each having one). So, for n H's and k alternating edges we get a total weighted contact of at least $\frac{n}{2}(8\alpha_2 + 3\alpha_1 + 1) - 14\alpha_2 - \frac{15}{2}\alpha_1 - \frac{1}{2} + 2k$. Hence we get the following approximation ratio A_c:

$$A_c = \frac{(6\alpha_2 + \alpha_1 + 3)n - \frac{1}{2}k}{\frac{n}{2}(8\alpha_2 + 3\alpha_1 + 1) - 14\alpha_2 - \frac{15}{2}\alpha_1 - \frac{1}{2} + 2k} \tag{5.1}$$

To analyse the ratio established above, we now discuss the corresponding ratios for specific value of α_1, α_2 and k.

i) **Case 1:** $\alpha_1 = \alpha_2 = \alpha$ (Let) :

$$A_c = \frac{(7\alpha + 3)n - \frac{1}{2}k}{\frac{n}{2}(11\alpha + 1) - \frac{43}{2}\alpha - \frac{1}{2} + 2k} \tag{5.2}$$

Now, if $\alpha = 0$, that means their is no effect of diagonal edge on weighted contact, then,

$$A_c = \frac{3n - \frac{1}{2}k}{\frac{n}{2} - \frac{1}{2} + 2k} \tag{5.3}$$

Now we can compute the value of k for which approximation ratio will be at least 6.

$\frac{3n-\frac{1}{2}k}{\frac{n}{2}-\frac{1}{2}+2k} \leq 6$
$\Rightarrow (3n - \frac{1}{2}k) \leq 6(\frac{n}{2} - \frac{1}{2} + 2k)$
$\Rightarrow 3n - \frac{1}{2}k \leq 3n - 3 + 12k$
$\Rightarrow \frac{25}{2}k \geq 3$
$\Rightarrow k \geq \frac{6}{25}$ or $k \geq 1$

That means if their is no effect of diagonal edge the lattice become similar to hexagonal lattice and the approximation ratio found shows same result of [29], where Jiang and Zhu works with hexagonal lattice.

If $\alpha = 1$, the approximation ratio is $\frac{5}{3}$ for $k > 10$ (as shown in [46]).

ii) **Case 2:** $\alpha_1 \neq \alpha_2$:

Now, assume that weighted contact is inversely proportional to the contact edge length. Then, $\alpha_1 = \frac{1}{2}$ and $\alpha_2 = \frac{1}{\sqrt{3}}$. Let, named it as *natural assignment*.

From equation 5.1,

$$A_c = \frac{(6\frac{1}{\sqrt{3}} + \frac{1}{2} + 3)n - \frac{1}{2}k}{\frac{n}{2}(8\frac{1}{\sqrt{3}} + 3\frac{1}{2} + 1) - 14\frac{1}{\sqrt{3}} - \frac{15}{2}\frac{1}{2} - \frac{1}{2} + 2k}$$

$\Rightarrow A_c = \frac{6.964n - \frac{1}{2}k}{3.559n - 16.08 + 2k}$

(5.4)

From Equation 5.4 it can be seen that approximation ratio of A_1 tends to reach $\frac{6.964}{3.559}$. Now we compute the value of k so that our approximation ratio is at most $\frac{6.964}{3.559}$ or 1.96.

$\frac{6.964n - \frac{1}{2}k}{3.559n - 16.08 + 2k} \leq \frac{6.964}{3.559}$

$\Rightarrow 3.559 \times 6.964n - 3.559 \times \frac{1}{2}k \leq 3.559 \times 6.964n - 16.08 \times 6.964 + 2 \times 6.964k$

$\Rightarrow 15.7075k \geq 16.08 \times 9.964$

$\Rightarrow k \geq \frac{16.08 \times 9.964}{15.7075}$

$\Rightarrow k \geq 7.12$ or $k \geq 8$

So, if the total number of H-runs is greater or equal than 8, then Algorithm ChainArrangement will achieve an approximation ratio of 1.96, assuming that the weighted contact is inversely proportional to the contact edge length.

Theorem 4. *For any given HP string, Algorithm ChainArrangement achieves an approximation ratio of* 1.96 *for* $k > 8$ *considering weighted contact under natural assignment, where* k *is the total number of H-runs,* n *is the total number of H and* $n = 2rs$ *with* $s = 1.5r$. □

5.4 Conclusions

In this chapter we have analysed Algorithm ChainArrangement. We use weighted contact, where different length of contact weight get different weight. The idea improves the approximation ratio to 1.96.

Chapter 6

Weighted Contact Analysis in the Hexagonal Prism Lattice with Diagonal

In this chapter, we deduce the approximation ratio considering the concept of weighted contact for Algorithm HelixArrangement and LayerArrangement.

6.1 Upper bound

In this section, we give an upper bound using the concept of weighted contact for hexagonal prism lattice with diagonals. In our assumption, the chemical binding force between the two amino acids which are connected via a big diagonal edge is smaller than that between the two amino acids which are connected via small diagonal edge. We introduce three additional parameters α_1, α_2 and α_3 $(0 \leq \alpha_1 \leq \alpha_2 \leq \alpha_3 \leq 1)$ to measure the loss of binding power relative to the binding power given by non-diagonal edges. Here we will count weighted contact of 1 for each non-diagonal contact edge or non-diagonal layer contact edge, weighted contact of α_1 for big diagonal contact edge, weighted contact of α_2 for small diagonal contact edge and weighted contact of α_3 for diagonal layer contact edge.

Lemma 6. *Let p be an HP string and $G = (V,\ E)$ be a hexagonal prism lattice with diagonals. The total weighted contact in a conformation ϕ is at most $(5+\alpha_1+6\alpha_2+6\alpha_3)\times n-\frac{1}{2}k$, where k is the total number of H-runs and n is the total number of H.*

Proof: Every vertex in the lattice G has exactly twenty neighbours comprising 3 non-diagonal neighbours, 3 big diagonal neighbours, 6 small diagonal neighbours, 2 non-diagonal layer neighbours and 6 diagonal layer neighbours (see Fig. 2.5). In this conformation, every H-vertex has exactly two binding edges. If the binding edges are big diagonal edges, then we get the minimum loss of the weighted contacts. Hence weighted contact will be at most $(5+\alpha_1+6\alpha_2+6\alpha_3)\times n$ for n Hs. Again, we can say that there will be at least $\frac{1}{2}k$ loss edges in ϕ for k H-runs. Therefore the weighted contact is at most $(5+\alpha_1+6\alpha_2+6\alpha_3)n-\frac{1}{2}k$ in ϕ. □

6.2 Weighted contact analysis of Algorithm HelixArrangement

Now we focus on deducing an approximation ratio for Algorithm HelixArrangement considering weighted contact. For a vertex, Table 6.1 provides an account for the different types of edges with different weights where the adjacent vertex is in different layers. Every vertex from its layer achieves 3 contacts, 2 of them are due to short diagonal edges (weight=α_2) and 1 is due to long diagonal edge (weight=α_1). From its immediate upper layer it achieves 3 contacts, 1 of them is due to non-diagonal layer edge (weight=1) and 2 of them are due to diagonal layer edges (weight=α_3). From its immediate lower layer it achieves 3 contacts, 1 of them is due to non-diagonal layer edge (weight=1) and 2 of them are diagonal layer edges (weight=α_3).

So, the total weighted contact ($\mathcal{C}$) of all the vertices can be computed as follows:

$$\mathcal{C} \geq (\alpha_1 + 2\alpha_2 + 1 + 2\alpha_3 + 1 + 2\alpha_3)n - 12(1 + 2\alpha_3) + 2k$$
$$\mathcal{C} \geq (\alpha_1 + 2\alpha_2 + 4\alpha_3 + 2)n - 12(1 + 2\alpha_3) + 2k$$

Hence considering weighted contact we get the following approximation ratio A_3:

$$A_3 = \frac{(5 + \alpha_1 + 6\alpha_2 + 6\alpha_3)n - \frac{1}{2}k}{(\alpha_1 + 2\alpha_2 + 4\alpha_3 + 2)n - 12(1 + 2\alpha_3) + 2k} \tag{6.1}$$

To further examine the ratio deduced above, we now consider some specific values of α_1, α_2, α_3 and k. It is natural to assume that the weighted contact is inversely proportional to the contact edge length. In that case we have $\alpha_1 = \frac{1}{2}$, $\alpha_2 = \frac{1}{\sqrt{3}}$ and $\alpha_3 = \frac{1}{\sqrt{2}}$. In what follows, we will refer to this assignment as the *natural assignment*. Then, under natural

Table 6.1: Weighted contacts for a vertex

Region	Contacts	non-diagonal edges ($weight = 1$)	short diagonal edges ($weight = \alpha_2$)	long diagonal edges ($weight = \alpha_1$)	non-diagonal layer edges ($weight = 1$)	diagonal layer edges ($weight = \alpha_3$)
From its layer	3	0	2	1	0	0
From immediate upper layer	3	0	0	0	1	2
From immediate lower layer	3	0	0	0	1	2

assignment, from Equation 6.1 we have the following:

$$A_2 = \frac{(5 + \frac{1}{2} + 6\frac{1}{\sqrt{3}} + 6\frac{1}{\sqrt{2}})n - \frac{1}{2}k}{(\frac{1}{2} + 2\frac{1}{\sqrt{3}} + 4\frac{1}{\sqrt{2}} + 2)n - 12(1 + 2\frac{1}{\sqrt{2}}) + 2k} \Rightarrow A_2 = \frac{13.2067n - \frac{1}{2}k}{6.483n - 29 + 2k} \tag{6.2}$$

From Equation 6.2 it can be seen that A_2 tends to reach $\frac{13.2067}{6.483} \approx 2$. Now we compute the value of k so that our approximation ratio is at most $\frac{13.2067}{6.483}$ or 2.

$\frac{13.2067n - \frac{1}{2}k}{6.483n - 29 + 2k} \leq \frac{13.2067}{6.483}$
$\Rightarrow \frac{9}{2}k \geq 58$
$\Rightarrow k \geq 12.88 \approx 13$

So, if the total number of H-runs is greater than 13, then Algorithm HelixArrangement will achieve an approximation ratio of 2.

Theorem 5. *For any given HP string, Algorithm HelixArrangement achieves an approximation ration of 2 for $k > 13$ considering weighted contact under natural assignment, where k is the total number of H-runs and n is the total number of H's.* □

6.3 Weighted contact analysis for Algorithm LayerArrangement

Now we focus on deducing an approximation ratio for Algorithm LayerArrangement considering weighted contacts. Table 6.2 provides an account for the different types of edges with different weights at different regions of the upper layer. Every vertex in the lR-up layer and rR-up layer has at least 8 contacts, due to 1 non-diagonal edge (weight=1), 3 short diagonal edges (weight=α_2), 1 long diagonal edge (weight=α_1), 1 non-diagonal layer edge (weight=1) and 2 diagonal layer edges (weight=α_3). Every vertex in the i_lR-upper layer and the i_rR-upper layer has at least 12 contacts, due to 1 non-diagonal edge, 5 short diagonal edges, 2 long diagonal edges, 1 non-diagonal layer edge and 3 diagonal layer edges. Every vertex in the uR-upper layer and the dR-upper layer has at least 6 contacts, due to of 1 non-diagonal edge, 2 short diagonal edges, 1 long diagonal edge, 1 non-diagonal layer edge and 1 diagonal layer edge. Every vertex in the i_uR-upper layer and the i_dR-upper layer has at least 11 contacts, due to 1 non-diagonal edge, 4 short diagonal edges, 2 long diagonal edges, 1 non-diagonal layer edge and 3 diagonal layer edges. Every vertex in the upper layer achieves at least 14 contacts, due to 1 non-diagonal edge, 6 short diagonal edges, 3 long diagonal edges, 1 non-diagonal layer edge and 3 diagonal layer edges.

So, the total weighted contact ($\mathcal{C}$) of all the vertices of the upper layer can be computed as follows:

$\mathcal{C} \geq 2\times(1+3\alpha_2+1\alpha_1+1+2\alpha_3)\times(r-2)+2\times(1+5\alpha_2+2\alpha_1+1+3\alpha_3)\times(r-2)+2\times(1+2\alpha_2+\alpha_1+1+\alpha_3)\times\frac{s+3}{2}+2\times(1+4\alpha_2+2\alpha_1+1+3\alpha_3)\times(\frac{s-3}{2})+(1+6\alpha_2+3\alpha_1+1+3\alpha_3)\times(2x-2r-2s-4)$
$\Rightarrow \mathcal{C} \geq rs(3\alpha_3+6\alpha_2+3\alpha_1+2)+r(4\alpha_3+4\alpha_2+4)-s(2\alpha_3+6\alpha_2+3\alpha_1)-(38\alpha_3+62\alpha_2+27\alpha_1+24)$
$\Rightarrow \mathcal{C} \geq m_1(3\alpha_3+6\alpha_2+3\alpha_1+2)+r(4\alpha_3+4\alpha_2+4)-s(2\alpha_3+6\alpha_2+3\alpha_1)-(38\alpha_3+62\alpha_2+27\alpha_1+24)$
$\Rightarrow \mathcal{C} \geq \frac{n}{2}(3\alpha_3+6\alpha_2+3\alpha_1+2)+r(4\alpha_3+4\alpha_2+4)-s(2\alpha_3+6\alpha_2+3\alpha_1)-(38\alpha_3+62\alpha_2+27\alpha_1+24)$

Since the upper layer is symmetric to the lower layer, the calculation would be the same for both layers, if $n = 2m_1$. So the total weighted contact will be at least $2\mathcal{C}$ or $n(3\alpha_3+6\alpha_2+3\alpha_1+2)+r(8\alpha_3+8\alpha_2+8)-s(4\alpha_3+12\alpha_2+6\alpha_1)-2(38\alpha_3+62\alpha_2+27\alpha_1+24)$.

If $n = 2m_1 + 1$, then let $n_1 = n - 1$. The total weighted contacts for these n_1 vertices will be at least,

Table 6.2: Weighted contacts for different regions of the upper layer

Region	no. of vertex	non-diagonal edge ($weight = 1$)	short diagonal edge ($weight = \alpha_2$)	long diagonal edge ($weight = \alpha_1$)	non-diagonal layer edge ($weight = 1$)	diagonal layer edge ($weight = \alpha_3$)
lR-upper layer	$r-2$	1	3	1	1	2
rR-upper layer	$r-2$	1	3	1	1	2
i_lR-upper layer	$r-2$	1	5	2	1	3
i_rR-upper layer	$r-2$	1	5	2	1	3
uR-upper layer	$\frac{s+3}{2}$	1	2	1	1	1
dR-upper layer	$\frac{s+3}{2}$	1	2	1	1	1
i_uR-upper layer	$\frac{s-3}{2}$	1	4	2	1	3
i_dR-upper layer	$\frac{s-3}{2}$	1	4	2	1	3
mR-upper layer	$(rs - 2r - 2s - 4)$	1	6	3	1	3

$n_1(3\alpha_3+6\alpha_2+3\alpha_1+2)+r(8\alpha_3+8\alpha_2+8)-s(4\alpha_3+12\alpha_2+6\alpha_1)-2(38\alpha_3+62\alpha_2+27\alpha_1+24).$

The remaining vertex will have at least 2 weighted contact. So the total weighted contact will be at least,

$(n-1)(3\alpha_3+6\alpha_2+3\alpha_1+2)+r(8\alpha_3+8\alpha_2+8)-s(4\alpha_3+12\alpha_2+6\alpha_1)-2(38\alpha_3+62\alpha_2+27\alpha_1+24)+2$

$n(3\alpha_3+6\alpha_2+3\alpha_1+2)+r(8\alpha_3+8\alpha_2+8)-s(4\alpha_3+12\alpha_2+6\alpha_1)-(79\alpha_3+130\alpha_2+57\alpha_1+48)$

So, combining the two cases, we get that the total weighted contact is at least,

$n(3\alpha_3+6\alpha_2+3\alpha_1+2)+r(8\alpha_3+8\alpha_2+8)-s(4\alpha_3+12\alpha_2+6\alpha_1)-(79\alpha_3+130\alpha_2+57\alpha_1+48).$

Now we need to take the alternating edges into our consideration. For every alternating edge we get 2 weighted contacts for the two vertices (each having 1). So, for n H's and k alternating edges we get a total of at least,

$n(3\alpha_3+6\alpha_2+3\alpha_1+2)+r(8\alpha_3+8\alpha_2+8)-s(4\alpha_3+12\alpha_2+6\alpha_1)-(79\alpha_3+130\alpha_2+57\alpha_1+48)+2k$ weighted contacts.

Hence we get the following approximation ratio A_4:

$$A_4 = \frac{(5+\alpha_1+6\alpha_2+6\alpha_3)n-\frac{1}{2}k}{n(3\alpha_3+6\alpha_2+3\alpha_1+2)+r(8\alpha_3+8\alpha_2+8)-s(4\alpha_3+12\alpha_2+6\alpha_1)-(79\alpha_3+130\alpha_2+57\alpha_1+48)+2k} \tag{6.3}$$

To further examine the ratio established above, we again consider the natural assignment for α_1, α_2, α_3 and k. Then, from Equation 6.3 we have the following:

$$A_4 = \frac{(5+\frac{1}{2}+6\frac{1}{\sqrt{3}}+6\frac{1}{\sqrt{2}})n-\frac{1}{2}k}{n(3\frac{1}{\sqrt{2}}+6\frac{1}{\sqrt{3}}+3\frac{1}{2}+2)+r(8\frac{1}{\sqrt{2}}+8\frac{1}{\sqrt{3}}+8)-s(4\frac{1}{\sqrt{2}}+12\frac{1}{\sqrt{3}}+6\frac{1}{2})-(79\frac{1}{\sqrt{2}}+130\frac{1}{\sqrt{3}}+57\frac{1}{2}+48)+2k}$$

$$\Rightarrow A_4 = \frac{13.2067n-\frac{1}{2}k}{9.085n+18.2756r-12.7566s-207.417+2k} \tag{6.4}$$

From Equation 6.4 it can be seen that A_4 tends to reach $\frac{13.2067}{9.085} = 1.45$.

Now we compute the value of k so that our approximation ratio is at most $\frac{13.2067}{9.085}$ or 1.45. Assuming, $18.2756r = 12.7566s$, i.e., $s \approx 1.5r$ We have the following:

$\frac{13.2067n - \frac{1}{2}k}{9.085n - 207.417 + 2k} \leq \frac{13.2067}{9.085}$

$\Rightarrow 30.9559k \geq 207.417 \times 13.2067$

$\Rightarrow k \geq 88.49$

$\Rightarrow k \geq 89$

So, if the total number of H-runs is greater than 89 and $s = 1.5r$, then Algorithm LayerArrangement will achieve an approximation ratio of 1.45.

Theorem 6. *For any given HP string, Algorithm LayerArrangement achieves an approximation ratio of* 1.45 *for* $k > 89$ *considering weighted contact under natural assignment, where* k *is the total number of H-runs,* n *is the total number of H and* $n = 2rs$ *with* $s = 1.5r$. □

Chapter 7

Visualize Software

In this chapter we elaborately describe our visualize software, which shows the structure of protein under Algorithm ChainArrangement, Algorithm HelixArrangement and Algorithm LayerArrangement. For making this software we use Visual Studio 2010 and OpenGL platform.

7.1 Software Description

In this software three algorithms are implemented - Algorithm ChainArrangement [46, 47], Algorithm HelixArrangement and Algorithm LayerArrangement. As said in previous chapter, Algorithm ChainArrangement is implemented in hexagonal lattice with diagonals and Algorithm HelixArrangement, Algorithm LayerArrangement are implemented in hexagonal prism lattice with diagonals.

For the software,

Input : An sequence of 1,0 equivalent to HP String weather H=1 and P=0 considered.

Output(to see folded protein structure press 'q'):

1. Length of HP String
2. Total number of H in that HP String
3. Folded protein structure

- Green lines show the folded structure of protein
- Blue lines show the contacts generated by the structure
- Filled green circle represents H bead
- Unfilled green circle represents P bead

4. Total contact generated by folded protein structure

7.2 Data source and Results

Energy matrix used in this experiment is given in Table 7.1.

Table 7.1: Energy matrix for HP model introduced in [13]

	H	P
H	-1	0
P	0	0

The HP benchmark sequence listed in Table 7.2, Table 7.3, Table 7.4 are from [59, 53] and represent standard sequence in this area of research. The benchmark problems have been studied before by authors [52, 3, 31, 17].

7.2.1 Result for Algorithm ChainArrangement [46, 47]

In this section we show result of folded structure of protein by Algorithm ChainArrangement and show the result of contact under different benchmark sequence input.

7.2.1.1 Simulation Result of Algorithm ChainArrangement

Some simulation result of Algorithm ChainArrangement are given below,

Input Sequence: HPC1 : H1P1H2P2H4P1H3P2H2P2H5P1H4P2H2P3H2P7H2

Output:

Length of HP String: 48.
Total number of H in that HP String: 27
Total contact: -160
Folded protein structure: See Fig. 7.1.

Input Sequence: HPC2 : H4P1H2P1H5P2H2P1H3P1H2P5H2P1H2P2H2P1H2P2H3P1H1

Output:

Length of HP String: 48.
Total number of H in that HP String: 30

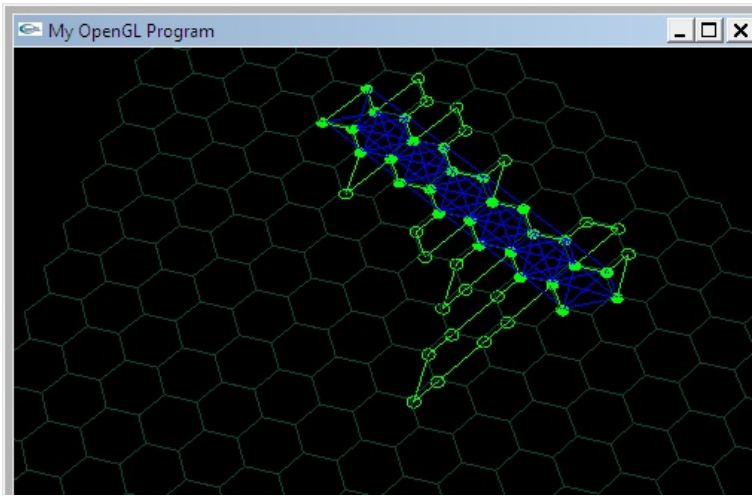

Figure 7.1: Folding for sequence HPC1 by Algorithm ChainArrangement. Green lines indicate binding edges, blue lines indicate contact edges.

Total contact: -186

Folded protein structure: See Fig. 7.2.

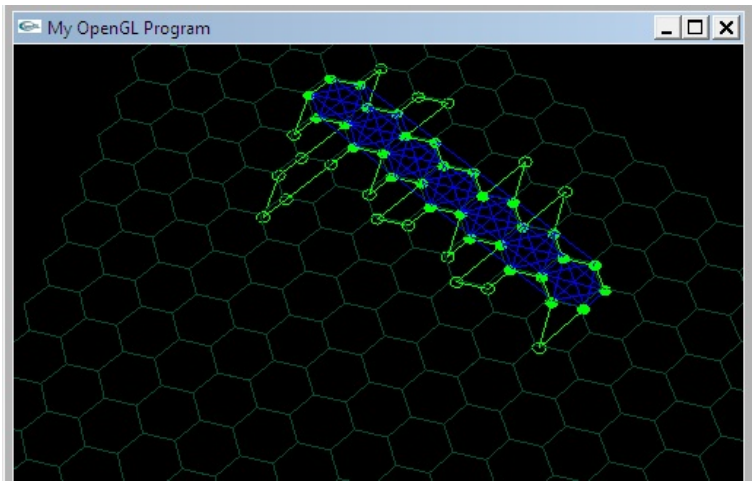

Figure 7.2: Folding for sequence HPC2 by Algorithm ChainArrangement. Green lines indicate binding edges, blue lines indicate contact edges.

7.2.1.2 Algorithm ChainArrangement under different benchmark sequence

Table 7.2 shows contacts tested under different benchmark sequence.

Table 7.2: HP model benchmark problems (length of 48) from[59, 53] for Algorithm ChainArrangement. Here 1 denoted for H and 0 denoted for P.

ID	Sequence	Total H	Total contact
HP1	101100111101110011001111101111001100011000000011	27	-160
HP2	111101101111100110111011000001101100110110011101	30	-186
HP3	011110111111001110011110111000110110011001110011	31	-184
HP4	011110011111001101100011111001111011100001101111	31	-182
HP5	001100111111001111011111100111011011000001101101	30	-176
HP6	111000110111101101101110000011100110011011111101	30	-184
HP7	011000111110111111011011000111000111001100110001	28	-168
HP8	011011011110011100000011110011011001101110011001	28	-170
HP9	011100001110110111111110011101101111001111100001	31	-182
HP10	011000000110001110110111100110110110011111110011	28	-168

7.2.2 Result for Algorithm HelixArrangement

In this section we show result of folded structure of protein by Algorithm HelixArrangement and show the result of contact under different benchmark sequence input.

7.2.2.1 Simulation Result of Algorithm HelixArrangement

Some simulation result of Algorithm HelixArrangement are given below,

Input Sequence: HP1 : H1P1H2P2H4P1H3P2H2P2H1P1H3P1H1P1H2P2H2P3H1P8H2

Output:

Length of HP String: 48.
Total number of H in that HP String: 23
Total contact: -252
Folded protein structure: See Fig. 7.3 and 7.4

Input Sequence: HP2 : H4P1H2P1H5P2H1P2H2P2H1P6H1P2H1P3H1P2H2P2H3P1H1

Output:

Length of HP String: 48.
Total number of H in that HP String: 23
Total contact: -256

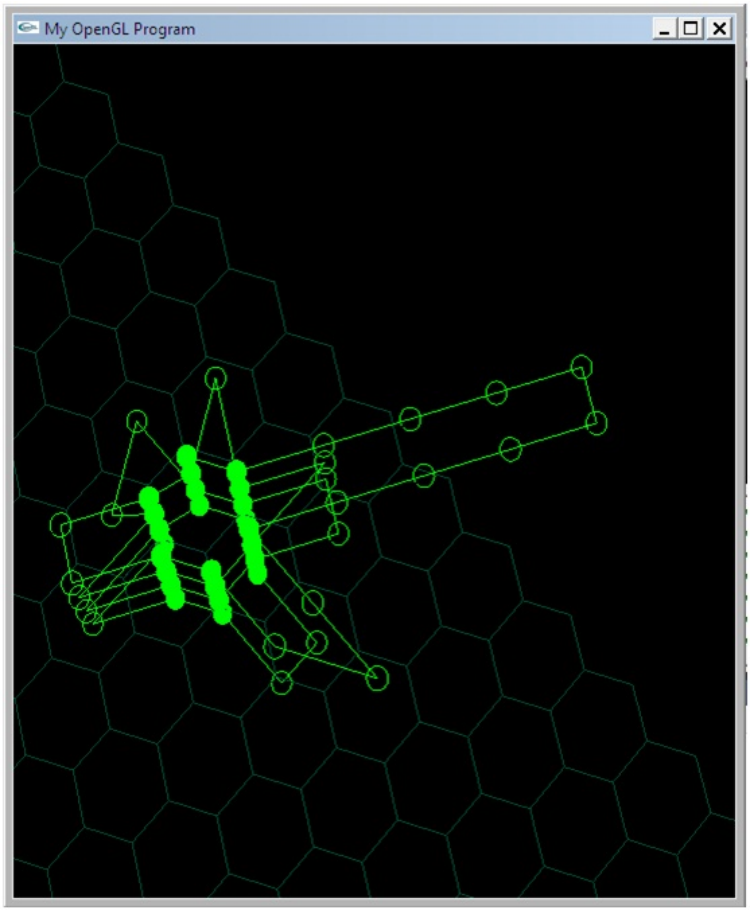

Figure 7.3: Top view of folding for sequence HP1 by Algorithm HelixArrangement

Folded protein structure: See Fig. 7.5 and 7.6

Input Sequence: HP3 : P1H1P1H2P1H6P2H1P1H1P2H1P1H2P1H1P1H1P3H1P2H2 P2H2P2H1P1H1P2H2

Output:

Length of HP String: 48.

Total number of H in that HP String: 25

Total contact: -268

Folded protein structure: See Fig. 7.7 and 7.8

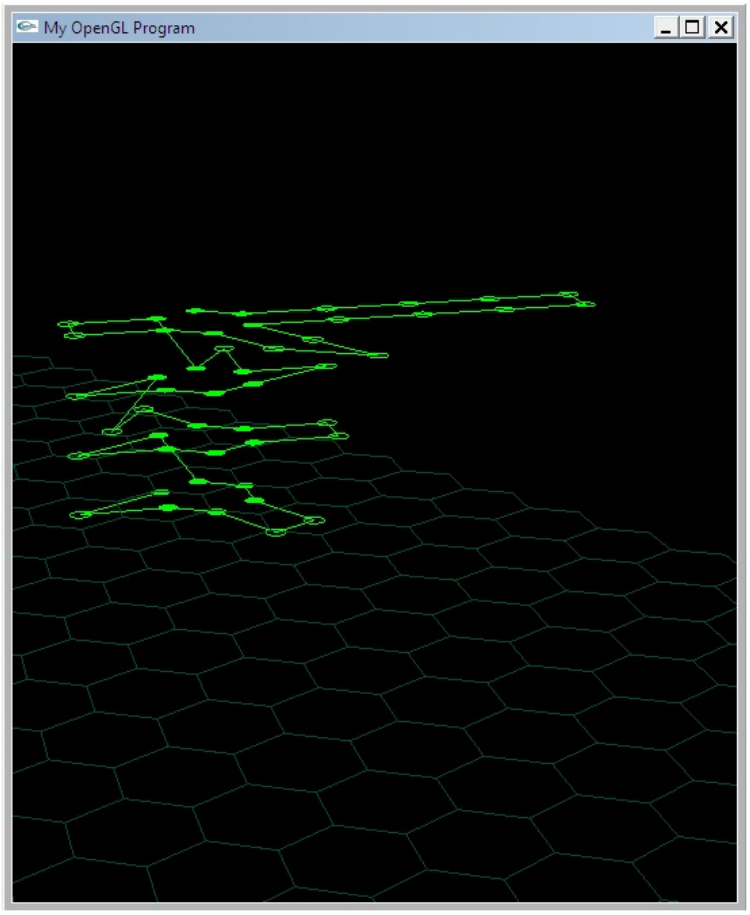

Figure 7.4: Side view of folding for sequence HP1 by Algorithm HelixArrangement

7.2.2.2 Algorithm HelixArrangement under different benchmark sequence

Table 7.3 shows contacts tested under different benchmark sequence.

7.2.3 Result for Algorithm LayerArrangement

In this section we show result of folded structure of protein by Algorithm LayerArrangement and show the result of contact under different benchmark sequence input.

7.2.3.1 Simulation Result of Algorithm LayerArrangement

Some simulation result of Algorithm LayerArrangement are given below,

Input Sequence: HPL1 : H2P2H2P4H7P1H3P2H2P1H2P1H2P1H3P1H3P1H2P1H2P1 H3P1H2P1H6P2H2P8H3P3H2P1H2P6H2P1H4

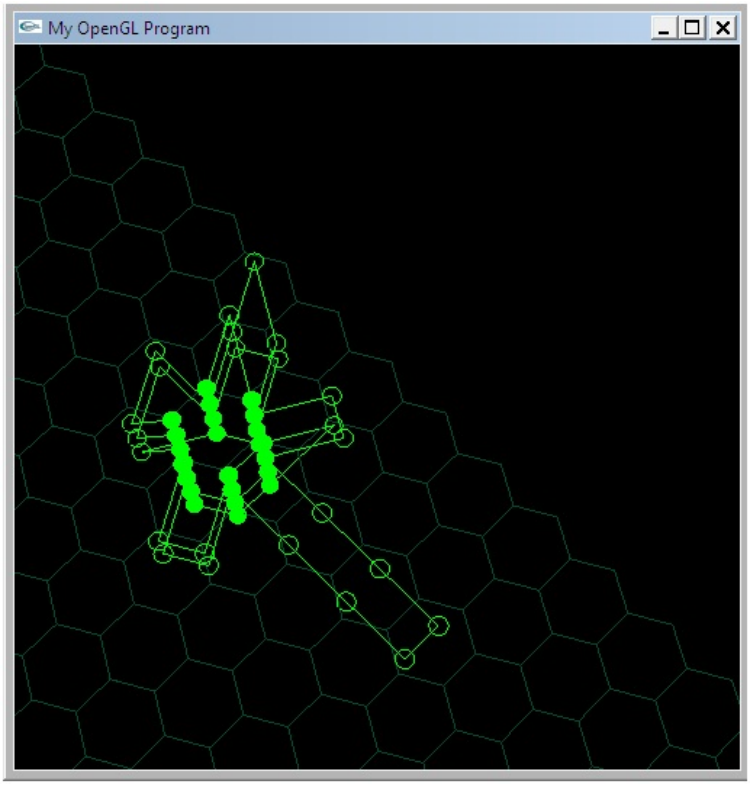

Figure 7.5: Top view of folding for sequence HP2 by Algorithm HelixArrangement

Output:

Length of HP String: 95.

Total number of H in that HP String: 56

Total contact: -636

Folded protein structure: See Fig. 7.9, 7.10 and 7.11

Input Sequence: HPL2 : H2P2H2P4H7P1H3P2H2P1H2P1H2P1H3P1H3P1H2P1H2P1H3P1H2P1H6P2H2P8H3P3H2P1H2P6H2P1H4P3H5P3H3P1H2P3H4P3H5P1H2P2H2P2H4

Output:

Length of HP String: 140.

Total number of H in that HP String: 83

Total contact: -1010

Folded protein structure: See Fig. 7.12, 7.13 and 7.14

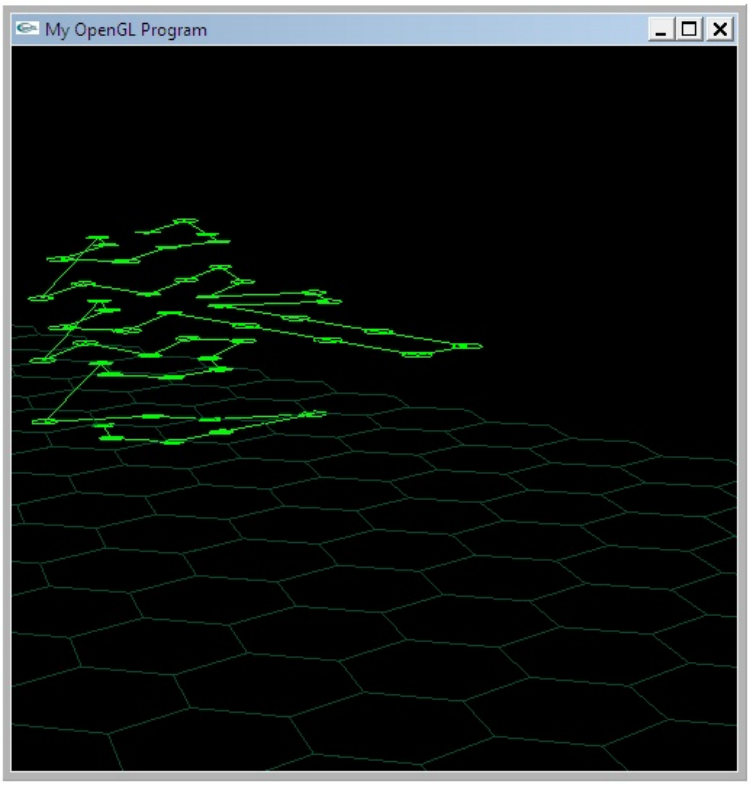

Figure 7.6: Side view of folding for sequence HP2 by Algorithm HelixArrangement

7.2.3.2 Algorithm LayerArrangement under different benchmark sequence

Table 7.4 shows contacts tested under different benchmark sequence.

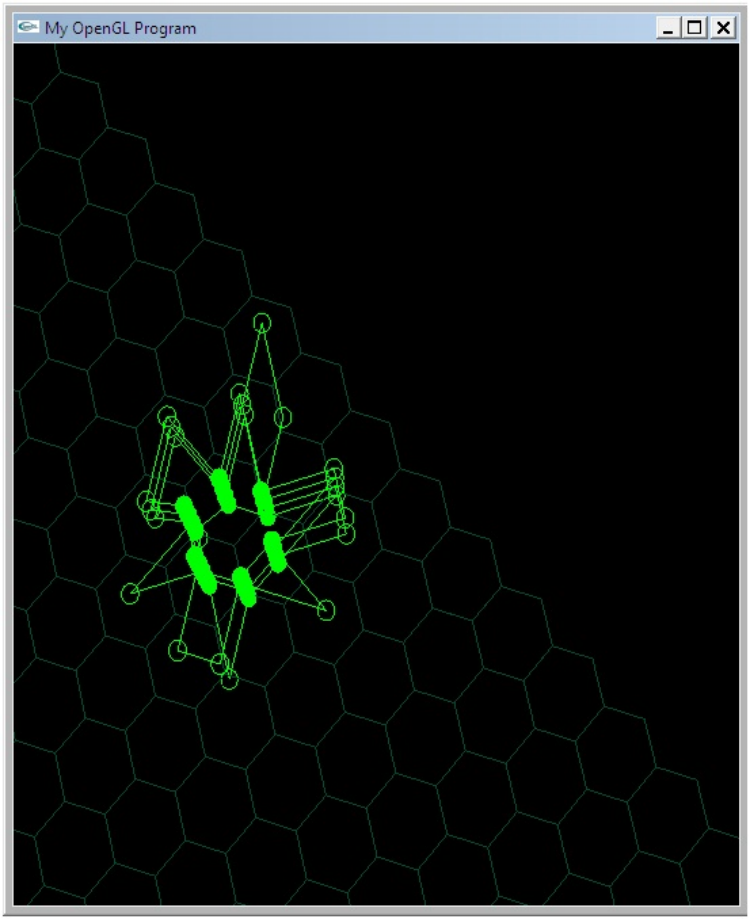

Figure 7.7: Top view of folding for sequence HP3 by Algorithm HelixArrangement

Table 7.3: HP model benchmark problems (length of 48) from[59, 53] for Algorithm HelixArrangement. Here 1 denoted for H and 0 denoted for P.

ID	Sequence	Total H	Total contact
HP1	101100111101110011001011101011001100010000000011	24	-252
HP2	111101101111100100110010000001001000100110011101	24	-256
HP3	010110111111001010010110101000100110011001010010	24	-260
HP4	010110010111001101100011111001011010100001001010	24	-258
HP5	001000101111001111011011100101010010000001101101	24	-242
HP6	111000110101101101101000000010100100010011111101	24	-268
HP7	010000101110101111011011000101000111001100110001	24	-258
HP8	011011101111001110000001011001101000110101011000	24	-250
HP9	010100001010100101111110011101001011001011100001	24	-240
HP10	011000000110001110100101100100100110011111110011	24	-250

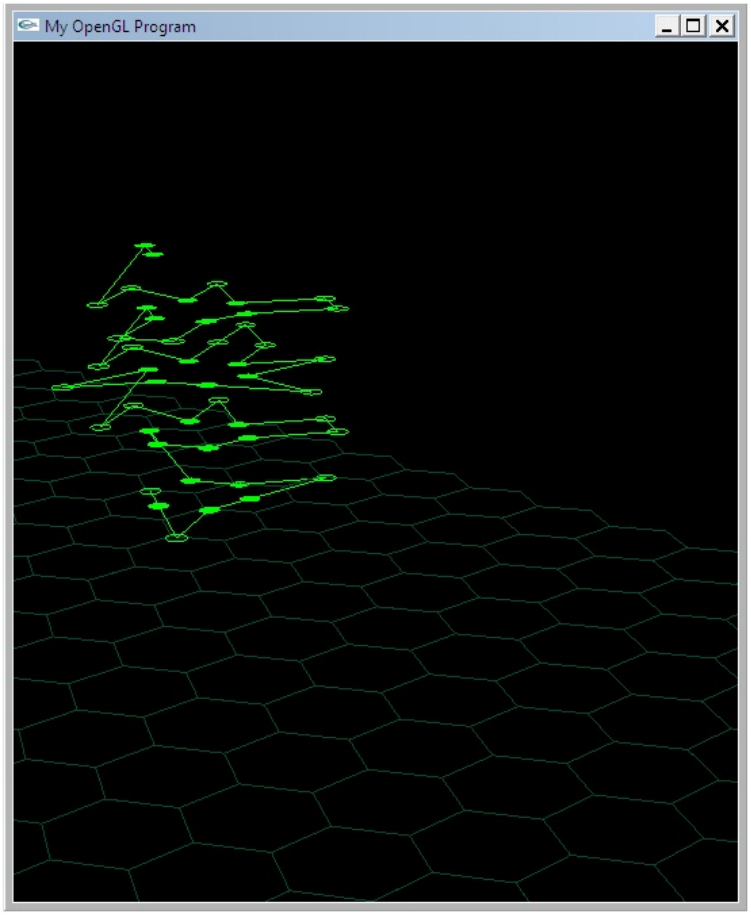

Figure 7.8: Side view of folding for sequence HP3 by Algorithm HelixArrangement

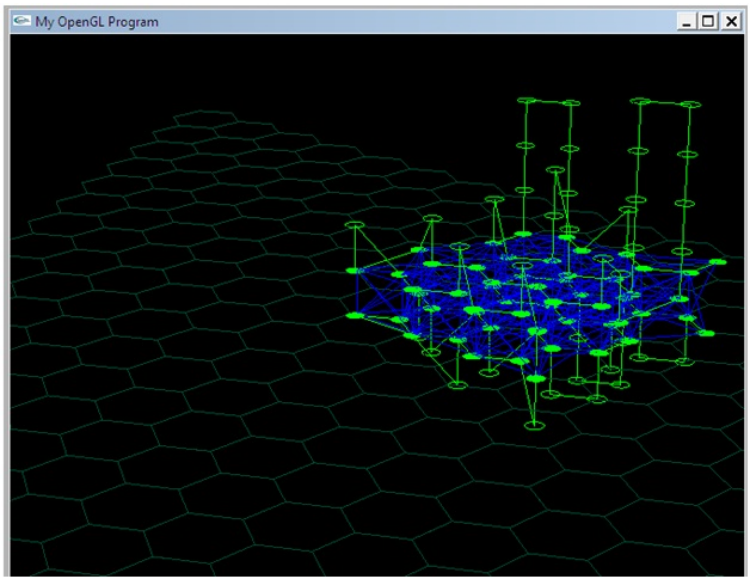

Figure 7.9: Side view of folding for sequence HPL1 by Algorithm LayerArrangement. Green edges are binding edges. Blue edges are contact edges.

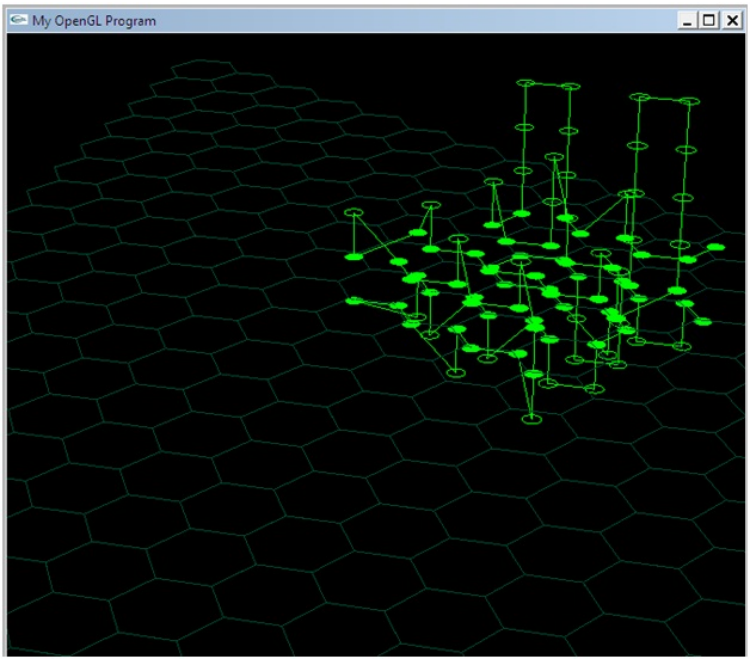

Figure 7.10: Side view of folding for sequence (contact edges are not shown) HPL1 by Algorithm LayerArrangement

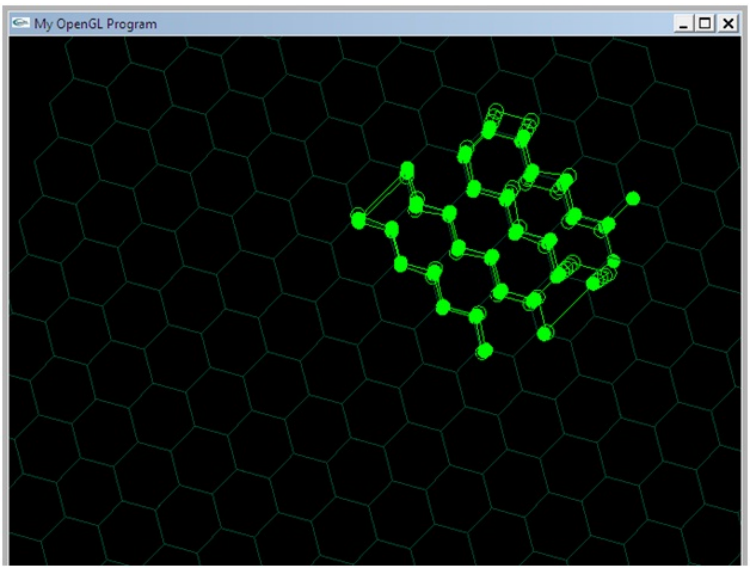

Figure 7.11: Top view of folding for sequence HPL1 by Algorithm LayerArrangement

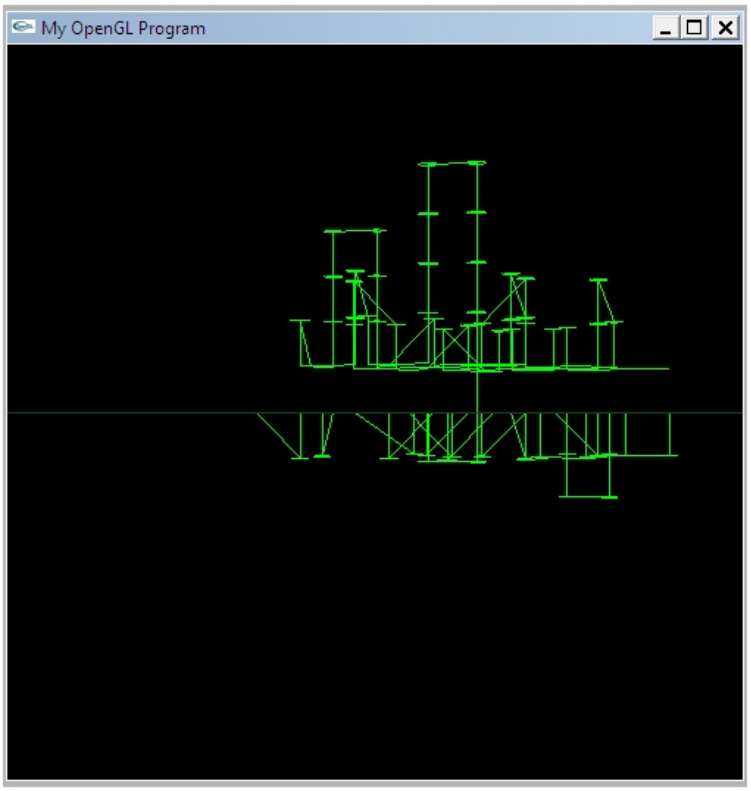

Figure 7.12: Side view(Parallel to x-axis) of folding for sequence HPL2 by Algorithm LayerArrangement

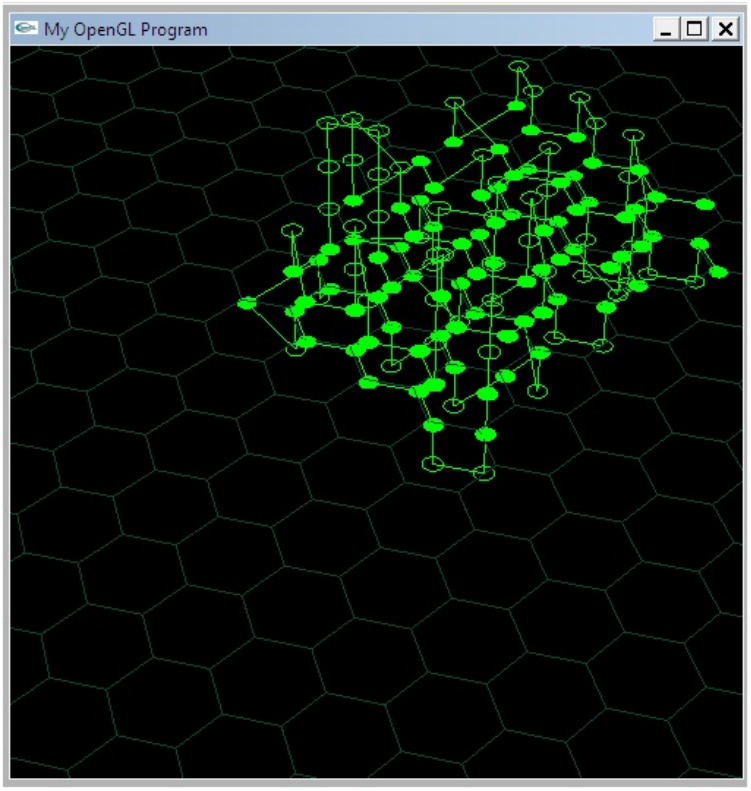

Figure 7.13: Side view of folding for sequence HPL2 by Algorithm LayerArrangement

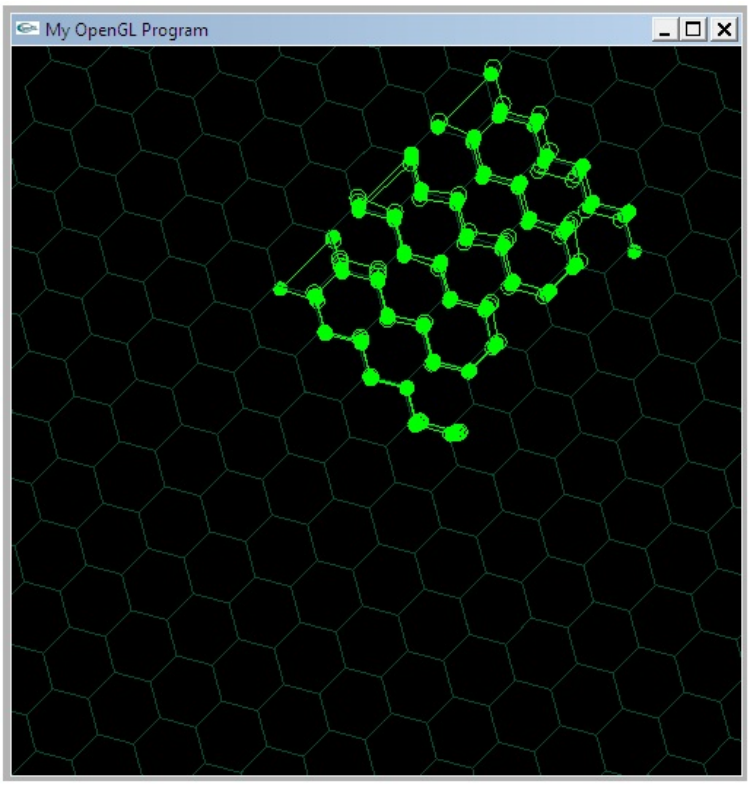

Figure 7.14: Top view of folding for sequence HPL2 by Algorithm LayerArrangement

Table 7.4: HP model benchmark problems for Algorithm LayerArrangement. Here r is number of chains in a layer and s is number of H-beads in a chain

Length	Sequence	Total H	Total contact	r	s
7	P1H6	6	10	r=2	s=1
23	P2H8P7H3P1H2	13	78	r=2	s=3
29	P2H2P1H5P7H3P1H5P1H2	17	130	r=2	s=4
37	P2H2P1H5P7H3P1H2P1H5P2H3P1H2	22	186	r=2	s=5
44	P2H2P1H5P7H3P1H2P1H2P1H2P1H2P2H3P1H2P2H2P1H2	25	232	r=2	s=6
51	P2H2P1H5P7H3P1H2P1H2P1H2P1H2P2H3P1H2P2H2P1H2 P1H2P2H2	29	282	r=2	s=7
61	P2H2P1H6P6H3P1H2P1H2P1H2P1H2P2H3P1H2P2H5P1H2 P2H4P6H2	35	332	r=2	s=8
65	P2H2P1H5P7H3P1H2P1H2P1H2P1H2P2H3P1H2P2H2P1H2 P1H2P2H4P6H2P1H3	36	366	r=2	s=9
69	P2H2P1H5P7H3P1H2P1H2P1H2P1H2P2H3P1H2P2H2P1H2 P1H2P2H4P6H2P1H7	40	404	r=2	s=10
29	P2H2P1H5P7H6P1H2P1H2	17	106	r=4	s=2
44	P2H8P7H3P1H2P1H5P2H3P1H2P2H2P1H2	27	220	r=4	s=3
61	P2H2P1H5P7H3P1H5P1H2P2H6P2H2P1H2P1H2P2H4 P6H2	35	308	r=4	s=4
69	P2H2P1H5P7H3P1H2P1H5P2H3P1H2P2H2P1H5P2H4 P6H2P1H7	42	402	r=4	s=5
81	P2H2P1H5P7H3P1H2P1H2P1H2P1H2P2H3P1H2P2H2P1H2 P1H2P2H4P6H2P1H9P1H5P2H2	49	532	r=4	s=6
91	P2H2P1H5P7H3P1H2P1H2P1H2P2H3P1H2P2H2P1H2 P1H2P2H4P6H2P1 H9P1H5P2H2P1H6P1H2	57	634	r=4	s=7
105	P2H2P1H6P6H3P1H2P1H2P1H2P2H3P1H2P2H5P1H2 P2H4P6H2P1H9P1H5P2H2P1H6P1H2P1H2P1H3P5H2	66	728	r=4	s=8
121	P2H2P1H5P7H3P1H2P1H2P1H2P2H3P1H2P2H2P1H2 P1H2P2H4P6H2P1H9P1H5P2H2P1H6P1H2P1H2P1H4 P4H3P1H3P2H3P4H2	74	868	r=4	s=9
131	P2H2P1H5P7H3P1H2P1H2P1H2P2H3 P1H2P2H2P1H2P1H2P2H4P6H2P1 H9P1H5P2H9P1H2P1H2P1H3P5H3 P1H3P2H3P4H3P1H2P2H4	81	954	r=4	s=10
43	P2H2P1H5P7H6P1H2P1H2P2H3P1H2P2H2P1H1	25	166	r=6	s=2
64	P2H8P7H3P1H2P1H5P2H3P1H2P2H2P1H2P1H3P1H4 P6H2P1H2	38	300	r=6	s=3
80	P2H2P1H5P7H3P1H5P1H2P2H6P2H2P1H2P1H2P2H4 P6H2P1H9P1H5P2H1	50	458	r=6	s=4
96	P2H2P1H5P7H3P1H2P1H5P2H3P1H2P2H2P1H5P2H4 P6H2P1H9P1H5P2H2P1H9P1H2P1H1	63	648	r=6	s=5
120	P2H2P1H5P7H3P1H2P1H2P1H2P2H3P1H2P2H2P1H2 P1H2P2H4P6H2P1H9P1H5P2H2P1H6P1H2P1H2P1 H3P5H3P1H3P2H3P4H1	72	874	r=6	s=6
135	P2H2P1H5P7H3P1H2P1H2P1H2P2H3P1H2P2H2P1H2 P1H2P2H4P6H2P1H9P1H5P2H2P1H6P1H2P1H2P1 H3P5H3P1H3P2H3P4H3P1H3P1H5P1H2	84	1022	r=6	s=7

Chapter 8

Conclusions

In this thesis we have introduced the hexagonal prism lattice with diagonals for HP model protein folding and presented several algorithms with analysis from several point of view. In this chapter, we draw the conclusion by highlighting the major contributions made in this thesis. We also provide some directions for future research. We have presented some basic approaches to solve protein folding problem studied in structural bioinformatics. We have discussed the problem in HP Model under different lattice.

In chapter 4, we introduce hexagonal prism lattice with diagonal for HP model. It alleviates the basic parity problem in cubic lattice for protein folding and also increase the cardinality of neighbour than other recognized lattice like FCC, BCC, SC etc. We provide two approximation algorithms on hexagonal prism lattice with diagonal and in chapter 6 analysis it using weighted contact concept.

8.1 Major Contribution

In this thesis, we introduce hexagonal prism lattice with diagonals. This lattice model removes some of the well known problems of protein folding in other lattices, e.g., parity problem. The major contributions that have done in this thesis are as follows.

- We present two novel approximation algorithms to solve the protein folding problem in the hexagonal prism lattice with diagonals in HP model. For any given HP string, our first algorithm, Algorithm HelixArrangement achieves an approximation ratio of 2 for $k > 16$, where k is the total number of H-runs and n is the total number of H. Our second algorithm, Algorithm LayerArrangement achieves an approximation ratio

of $\frac{9}{7}$ under some parametric constraints. Both algorithms are polynomial in terms of the length of the given HP string.

- We incorporate the concept of weighted contact which has biological motivation. Considering weighted contact we analyse our two algorithms as well as previous algorithm on a different lattice. In particular we first apply the concept of weighted contact on a previous algorithm (Algorithm ChainArrangement) of Shaw et al.[46]. Considering weighted contact, the Algorithm ChainArrangement provides 1.96-approximation ratio for $k > 8$, where k is number of sequence of Hs in the HP string. This new analysis on hexagonal lattice with diagonal improve the performance of the algorithm.

- Considering weighted contact, Algorithm HelixArrangement achieves an approximation ratio of 2 for $k > 13$ and Algorithm LayerArrangement achieves 1.45-approximation ratio for $k > 89$, where k is number of sequence of Hs in the HP string.

- We develop a simple visualization software for the approximation algorithms and tested under standard dataset. This software simulate the Algorithm ChainArrangement, Algorithm HelixArrangemnt and Algorithm LayerArrangement. Protein structure generate from this algorithm along with their contacts are shown in simulation output. The test under standard dataset results similar approximation ratio that theoretically found.

8.2 Future Plans

A number of future research directions is presented below,

- In this thesis we worked only with HP model. We will also investigate other variants of the HP model like the HP model with side chains. We will also work with other model different from HP model, e.g., MJ (Miyazawa-Jernigan) model [41, 42].

- Several approximation algorithm can be developed for improving the approximation ratio. In this thesis we have used weighted contact which is based on length of contact edges for improving the approximation ratio. Next target could be analyze the ratio on basis of angular distance between the beads.

- We have idea to apply more heuristics and meta-heuristics algorithms on this lattice and compare with other lattices.

- It is a open problem to find the hardness of algorithm for folding a protein in HP model for hexagonal prism lattice with diagonals. So it could be a possible research direction.

Bibliography

[1] R. Abagyan, M. Totrov, and D. Kuznetsov. Icm: a new method for structure modeling and design: Applications to docking and structure prediction from the distorted native conformation. *J. Comp. Chem.*, 15:488–506, 1994.

[2] R. Agarwala, S. Batzogloa, V. Dancik, S. Decatur, S. Hannenhalli, M. Farach, S.Muthukrishnan, and S. Skiena. Local rules for protein folding on a triangular lattice and generalized hydrophobicity in the hp model. *Journal of Computational Biology*, 4(3):276–296, 1997.

[3] A. A. Albrecht, A. Skaliotis, and K. Steinhofel. Stochastic protein folding simulation in the three-dimensional hp-model. *Comput. Biol. Chem.*, 32:248–255, 2008.

[4] K. Arnold, L. Bordoli, J. Kopp, and T. Schwede. The swiss-model workspace: a web-based environment for protein structure homology modelling. *Bioinformatics*, 22(2):195–201, January 2006.

[5] B. Berger and T. Leighton. Protein folding in the hydrophobic-hydrophilic (hp) model is np-complete. *Journal of Computational Biology*, 5(1):27–40, 1998.

[6] H. M. Berman, T. Battistuz, T. N. Bhat, W. F. Bluhm, P. E. Bourne, K. Burkhardt, Z. Feng, G. L. Gilliland, L. Iype, S. Jain, P. Fagan, J. Marvin, D. Padilla, V. Ravichandran, B. Schneider, N. Thanki, H. Weissig, J. D. Westbrook, and C. Zardecki. The protein data bank. *Acta Crystallogr. D. Biol. Crystallogr.*, 58:899–907, June 2002.

[7] L. Bianchi, L. M. Gambardella, and M. Dorigo. An ant colony optimization approach to the probabilistic traveling salesman problem. In *PPSN*, pages 883–892, 2002.

[8] H.-J. Böckenhauer and D. Bongartz. Protein folding in the hp model on grid lattices with diagonals. *Discrete Applied Mathematics*, 155(2):230–256, 2007.

[9] H.-J. Böckenhauer, A. Z. M. D. Ullah, L. Kapsokalivas, and K. Steinhöfel. A local move set for protein folding in triangular lattice models. In *WABI*, pages 369–381, 2008.

[10] P. Bradley, K. M. Misura, and D. Baker. Toward high-resolution de novo structure prediction for small proteins. *Science*, 309(5742):1868–1871, September 2005.

[11] B. Brooks, R. Bruccoleri, B. Olafson, D. States, S. Swami-nathan, and M. Karplus. Charmm: A program for macromolecular energy, minimization, and dynamics calculations. *J. Comput. Chem.*, 4:187–217, 1983.

[12] P. Crescenzi, D. Goldman, C. H. Papadimitriou, A. Piccolboni, and M. Yannakakis. On the complexity of protein folding. *Journal of Computational Biology*, 5(3):423–465, 1998.

[13] K. A. Dill. Theory for the folding and stability of globular proteins. *Biochemistry*, 24:1501–1509, 1985.

[14] K. A. Dill and J. L. MacCallum. The protein-folding problem, 50 years on. *Science*, 338(6110):399–307, 23 November 2012.

[15] M. Dorigo and M. Birattari. Ant colony optimization. In *Encyclopedia of Machine Learning*, pages 36–39. 2010.

[16] M. Dorigo, M. Birattari, and T. Stützle. Metaheuristic. In *Encyclopedia of Machine Learning*, page 662. 2010.

[17] I. Dotú, M. Cebriàn, P. V. Hentenryck, and P. Clote. On lattice protein structure prediction revisited. *IEEE/ACM Trans. Comput. Biology Bioinform.*, 8(6):1620–1632, 2011.

[18] Y. Duan, C. Wu, S. Chowdhury, M. C. Lee, G. Xiong, W. Zhang, R. Yang, P. Cieplak, R. Luo, T. Lee, J. Caldwell, J. Wang, and P. Koll-man. A point-charge force field for molecular mechanics simulations of proteins based on condensed-phase quantum mechanical calculations. *J. Comput. Chem.*, 24(16):1999–2012, December 2003.

[19] U. Hansmann and Y. Okamoto. Monte carlo simulations in generalized ensemble: Multicanonical algorithm versus simulated tempering. *Phys. Rev. E*, 54:5863–5865, 1996.

[20] W. Hart and S. Istrail. Fast protein folding in the hydrophobic-hydrophilic model within three-eighths of optimal. *Journal of Computational Biology*, 3(1):53–96, 1996.

[21] W. Hart and S. Istrail. Lattice and o-lattice side chain models of protein folding: Linear time structure prediction better than 86% of optimal. *Computational Biology*, 4(3):241–259, 1997.

[22] G. Helles. A comparative study of the reported performance of ab initio protein structure prediction algorithms. *J. R. Soc. Interface*, 5(21):387–396, April 2008.

[23] V. Heun. Approximate protein folding in the hp side chain model on extended cubic lattices. In *ESA*, pages 212–223, 1999.

[24] T. Hoque, M. Chetty, and L. S. Dooley. A hybrid genetic algorithm for 2d fcc hydrophobic-hydrophilic lattice model to predict protein folding. In *Australian Conference on Artificial Intelligence*, pages 867–876, 2006.

[25] T. Hoque, M. Chetty, and A. Sattar. Protein folding prediction in 3d fcc hp lattice model using genetic algorithm. In *IEEE Congress on Evolutionary Computation*, pages 4138–4145, 2007.

[26] A. S. M. S. Islam and M. S. Rahman. On the protein folding problem in 2d-triangular lattices. *Algorithms for Molecular Biology*, 8:30, 2013.

[27] A. S. M. S. Islam and M. S. Rahman. Protein folding in 2d-triangular lattice revisited - (extended abstract). In *IWOCA*, pages 244–257, 2013.

[28] M. K. Islam, M. Chetty, A. Z. M. D. Ullah, and K. Steinhöfel. A memetic approach to protein structure prediction in triangular lattices. In *ICONIP (1)*, pages 625–635, 2011.

[29] M. Jiang and B. Zhu. Protein folding on the hexagonal lattice in the hp model. *J. Bioinformatics and Computational Biology*, 3(1):19–34, 2005.

[30] B. John and A. Sali. Comparative protein structure modeling by iterative alignment, model building and model assessment. *Nucleic. Acids. Res.*, 31(14):3982–3992, July 2003.

[31] L. Kapsokalivas, X. Gan, A. A. Albrecht, and K. Steinhöfel. Population-based local search for protein folding simulation in the MJ energy model and cubic lattices. *Computational Biology and Chemistry*, 33(4):283–294, 2009.

[32] D. Karaboga. Artificial bee colony algorithm. *Scholarpedia*, 5(3):6915, 2010.

[33] I. Kessler and M. Livingston. The expected number of parts in a partition of n. *Monatshefte für Mathematik*, 81(3):203–212, 1976.

[34] J. Klepeis and C. Floudas. Prediction of -sheet topology and disulfide bridges in polypeptides. *Journal of Computational Chemistry*, 24(2):191–208, 2002.

[35] P. Y. Lam, P. K. Jadhav, C. J. Eyermann, C. N. Hodge, Y. Ru, L. T. Bacheler, J. L. Meek, M. J. Otto, M. M. Rayner, and Y. N. Wong. Rational design of potent, bioavailable, nonpeptide cyclic ureas as hiv protease inhibitors. *Science*, 263(5145):380–384, January 1994.

[36] R. Lathrop and T. Smith. Global optimum protein threading with gapped alignment and empirical pair score functions. *Mol. Biol.*, 255(4):641–665, 1996.

[37] R. H. Lathrop. The protein threading problem with sequence amino acid interaction preferences is np-complete. *Protein. Eng.*, 7(9):1059–1068, September 1994.

[38] N. Lesh, M. Mitzenmacher, and S. Whitesides. A complete and effective move set for simplified protein folding. In *7th Annual International Conference on Research in Computational Molecular Biology (RECOMB) 2003*, pages 188–195. ACM Press, 2003.

[39] C.-J. Lin and M.-H. Hsieh. An efficient hybrid taguchi-genetic algorithm for protein folding simulation. *Expert Systems with Applications*, 36:12446–12453, 2009.

[40] G. Mauri, A. Piccolboni, and G. Pavesi. Approximation algorithms for protein folding prediction. In *Symposium on Discrete Algorithms, (SODA)*, pages 945–946, 1999.

[41] S. Miyazawa and R. L. Jernigan. Estimation of effective interresidue contact energies from protein crystal structures: quasi-chemical approximation. *Macromolecules*, 18(3):534–552, 1985.

[42] S. Miyazawa and R. L. Jernigan. Residue-residue potentials with a favorable contact pair term and an unfavorable high packing density term, for simulation and threading. *Journal of Molecular Biology.*, 256(3):623–644, 1996.

[43] A. Newman. A new algorithm for protein folding in the hp model. In *Symposium on Discrete Algorithms, (SODA)*, pages 876–884, 2002.

[44] A. Newman and M. Ruhl. Combinatorial problems on strings with applications to protein folding. In *LATIN*, volume 2976 of *Lecture Notes in Computer Science*, pages 369–378. Springer, 2004.

[45] D. T. Pham, M. Castellani, M. Sholedolu, and A. Ghanbarzadeh. The bees algorithm and mechanical design optimisation. In *ICINCO 2008, Proceedings of the Fifth International Conference on Informatics in Control, Automation and Robotics, Intelligent Control Systems and Optimization, Funchal, Madeira, Portugal, May 11-15, 2008*, pages 250–255, 2008.

[46] D. L. Shaw, A. S. M. S. Islam, M. S. Rahman, and M. Hasan. Protein folding in hp model on hexagonal lattices with diagonals. *BMC Bioinformatics*, 15(S-2):S7, 2014.

[47] D. L. Shaw and S. Karmaker. Algorithms of bioinformatics. *B.Sc. Engg. Thesis, Department of Computer Science and Engineering, BUET*, 2013.

[48] A. Shmygelska, R. Aguirre-Hernández, and H. H. Hoos. An ant colony optimization algorithm for the 2d hp protein folding problem. In *Ant Algorithms*, pages 40–53, 2002.

[49] A. Shmygelska and H. H. Hoos. An improved ant colony optimisation algorithm for the 2d hp protein folding problem. In *Canadian Conference on AI*, pages 400–417, 2003.

[50] A. Shmygelska and H. H. Hoos. An ant colony optimisation algorithm for the 2d and 3d hydrophobic polar protein folding problem. *BMC Bioinformatics*, 6:30, 2005.

[51] M. Sippl. Calculation of conformation ensembles from potentials of mean force. *J. Mol. Biol.*, 213:859–883, 1990.

[52] K. Steinhofel, A. Skaliotis, and A. A. Albrecht. Relating time complexity of protein folding simulation to approximations of folding time. *Comp. Phys. Comm.*, 176:165–170, 2007.

[53] K. A. D. T. C. Beutler. A fast conformational search strategy for finding low energy structures of model proteins. *Protein Science*, 5(10):2037–2043, 1996 Oct.

[54] J. Teo and H. A. Abbass. A true annealing approach to the marriage in honey-bees optimization algorithm. *International Journal of Computational Intelligence and Applications*, 3(2):199–211, 2003.

[55] C. Thachuk, A. Shmygelska, and H. H. Hoos. A replica exchange monte carlo algorithm for protein folding in the hp model. *BMC Bioinformatics*, 8, 2007.

[56] A. Z. M. D. Ullah and K. Steinhöfel. A hybrid approach to protein folding problem integrating constraint programming with local search. *BMC Bioinformatics*, 11(S-1):39, 2010.

[57] R. Unger and J. Moult. Genetic algorithms for protein folding simulations. *Journal of Molecular Biology*, 231:75–81, 1993.

[58] S. Wu, J. Skolnick, and Y. Zhang. Ab initio modeling of small proteins by iterative TASSER simulations. *BMC. Biol.*, 5(17):399–407, 2007.

[59] K. Yue, K. M. Fiebig, P. D. Thomas, H. S. Chan, E. I. Shakhnovich, and K. A. Dill. A test of lattice protein folding algorithms. *Proceedings of the National Academy of Sciences of the United States of America*, 92(1):325–329, 1995.

[60] Y. Zhang. I-TASSER server for protein 3D structure prediction. *BMC Bioinformatics*, 9, 2008.

[61] Y. Zhang and J. Skolnick. The protein structure prediction problem could be solved using the current PDB library. *Proc. Natl. Acad. Sci.*, 102(4):1029–1034, January 2005.

[62] Y. Zhang and L. Wu. Bacterial chemotaxis optimization for protein folding model. In *ICNC (4)*, pages 159–162, 2009.