Protein Folding Optimization in a Hydrophobic-Polar Model for Predicting Tertiary Structure Using Fruit Fly Optimization Algorithm

by

Sajib Chatterjee

Roll No: 1707554

A thesis submitted in partial fulfillment of the requirements for the degree of Master of Science in Computer Science and Engineering

Department of Computer Science and Engineering
Khulna University of Engineering & Technology
Khulna 9203, Bangladesh
February 2020

Declaration

This is to certify that the thesis work entitled "Protein Folding Optimization in a Hydrophobic-Polar Model for Predicting Tertiary Structure Using Fruit Fly Optimization Algorithm" has been carried out by Sajib Chatterjee in the Department of Computer Science and Engineering (CSE), Khulna University of Engineering & Technology (KUET), Khulna, Bangladesh. The above thesis work or any part of this work has not been submitted anywhere for the award of any degree or diploma.

24/02/2020

Signature of Supervisor
(Dr. Pintu Chandra Shill)

Sajib Chatterjee

Signature of Candidate
(Sajib Chatterjee)
Roll No: 1707554

Approval

This is to certify that the thesis work submitted by Sajib Chatterjee entitled "Protein Folding Optimization in a Hydrophobic-Polar Model for Predicting Tertiary Structure Using Fruit Fly Optimization Algorithm" has been approved by the board of examiners for the partial fulfillment of the requirements for the degree of Master of Science in Computer Science & Engineering in the Department of Computer Science and Engineering, Khulna University of Engineering & Technology, Khulna, Bangladesh in February 2020.

BOARD OF EXAMINERS

24/02/2020

1. Prof. Dr. Pintu Chandra Shill
Professor
Department of Computer Science and Engineering(CSE)
Khulna University of Engineering & Technology(KUET)

Chairman
(Supervisor)

24/02/2020

2. Head
Department of Computer Science and Engineering(CSE)
Khulna University of Engineering & Technology(KUET

Member

24.02.2020

3. Prof. Dr. K. M. Azharul Hasan
Professor
Department of Computer Science and Engineering(CSE)
Khulna University of Engineering & Technology(KUET)

Member

24.2.2020

4. Prof. Dr. Sk. Md. Masudul Ahsan
Professor
Department of Computer Science and Engineering(CSE)
Khulna University of Engineering & Technology(KUET)

Member

24.02.2020

5. Prof. Dr. Md. Anisur Rahman
Professor
Department of Computer Science and Engineering(CSE)
Khulna University(KU)

Member
(External)

Acknowledgment

All the commendation on account of the almighty God, whose benison helped me to effectively fulfil this research work. I show suggestiveness and indefinable gratitude to my supervisor Dr. Pintu Chandra Shill, Professor, Department of Computer Science and Engineering, Khulna University of Engineering & Technology for his great helpful achievement in giving suggestion and inspiration. I acknowledge his certain collaboration and proper leading throughout the construction procedure. He has been a great source of efficient and fruitful concepts, deep knowledge and all time response for me.

I applaud all the faculty members of the Department of Computer Science and Engineering who helped me providing inspirations to perform the work. I would also like to applaud my family and friends for their sincere support.

February, 2020 Sajib Chatterjee

Abstract

The prediction of the three-dimensional structure of a protein from its amino acid sequence is an experiment that is very much well known optimization problem which is known as the Protein Folding Optimization (PFO) in many years. The PFO problem states to the computational problem of how to predict the local structure of a protein from its amino acid. PFO problem is the NP-hard and most challenging problem. Various kind of optimization algorithm already applied for solving the PFO problem, but none of the existing algorithm not provide the accurate result within optimal time. Fruit Fly Optimization Algorithm (FOA) is a recent metaheuristics algorithm that have the intensity and diversity characteristics of searching technique. Therefore, we applied FOA for solving PFO problem in the HP (Hydrophobic-Polar) cubic lattice model. In order to increase the convergence of the FOA, we have designed and developed three different operators of FOA: smell-based search, local vision-based search and global vision-based search technique for the perspective of PFO problem. The proposed algorithm is based on two extra mechanisms centroid hydrophobic and moderator mechanism, which are accountable for improving the accomplishment of the algorithm. The centroid hydrophobic mechanism tries to move the hydrophobic monomers to the center position of the structure. The moderator mechanisms try to move a part of monomers in the protein sequence each possible directions and place at the position where the maximum energy value found. This two extra mechanisms improved the performance of the propose algorithm magically. Moreover, we have developed a reconstruction operator for producing an accurate 3D structure of protein sequences by erasing overlapping in cubic lattice points. The experiment result shows of our proposed Fruit Fly Optimization Algorithm for Protein Folding Optimization (PFO_FOA) provide better accuracy than the existing algorithms.

Contents

PAGE

LIST OF TABLES

Table No. **Description** **Page**

LIST OF FIGURES

Figure No. **Description** **Page**

Nomenclature

PFO	Protein Folding Optimization
FOA	Fruit Fly Optimization Algorithm
HP	Hydrophobic Polar
l	Length of protein sequence
F_c	Objective function
E_c	Energy function of conformation
S	Protein sequence
L	Left
R	Right
U	Up
D	Down
F	Forward
B	Backward
LV	Local vision
GV	Global vision
H_s	Distance from the core position of any monomer

CHAPTER I

Introduction

1.1 Background

The prediction of the three-dimensional structure of a protein from its amino acid sequence is an experiment that is usually known as Protein Folding Optimization (PFO) problem. The potential impact of significant advances in structure prediction is enormous, since we already have ample evidence of the importance of 3D structure information in so many areas of biology. The PSP problem states to the computational problem of how to predict the local structure of a protein from its amino acid. Protein structure is controlled by its structure and Degrees Of Freedom (DOF) is reduced by cubic lattice model, take into account only specific interactions and therefore speed up the calculation of energy [1].

1.2 Motivation

Protein is an important substance found in every cell in the human body. Although all the necessary information needed for life is encoded in DNA, the process of life maintenance, replication, defense and reproduction are carried out by proteins. Protein regulates all the activities of human organisms [1].A protein structure created from sequence of amino acid which is depending on the tertiary structure that builds unusual forms of protein which causes many types of diseases. Some of them are cystic fibrosis, Alzheimer’s disease, parkinson’s disease, different types of cancer and mad cow. The primary structure is predicted by the tertiary structure, will be capable of improving these diseases [2]. If we can correctly design a method that gives sufficient information about the 3D native structure, then it is possible to explain how the protein works, how to cure diseases by appropriate drug design and so on [3]. For all these reasons, protein folding optimization (PFO) is a very important and tough problem in computational biology [4]. Many existing algorithms [2-14] has already solved the PFO problem. Our motivation is to develop an algorithm that can solve the protein folding optimization problem more efficiently than the remaining algorithms.

Fruit Fly Optimization Algorithm (FOA) is a recent procedure that has shown robust performance than all other approaches in solving different optimization problems [5].

Intensification and diversification are the main characteristics of FOA that help the algorithm to find a solution more quickly and efficiently. Though it is a metaheuristics algorithm, it can be designed for any optimization problem and the algorithm is suitable for the PFO problem as well [6].

1.3 Objectives

The objective of this thesis is to develop a method to predict the tertiary structure of protein known as protein folding optimization problem using the fruit fly optimization algorithm (FOA). Moreover, the three operators of FOA: Smell Based Search, Local Vision Based Search and Global Vision Based Search have been designed to solve PFO problem. For improving the performance of FOA design and developed two extra mechanisms: centroid hydrophobic and moderator mechanism, that make the PFO more acuratable.

1.4 Contribution of the thesis

We have proposed a method to predict the tertiary structure of protein, known as protein folding optimization problem using modified fruit fly Optimization algorithm (FOA). The three operators of FOA: Smell Based Search, Local Vision Based Search and Global Vision Based Search have been designed to solve PFO problem with two extra mechanisms: centroid hydrophobic and moderator mechanism. We have also designed a reconstruction mechanism to get the valid structure. The simulation results show that FOA performs accurately in the instance of PFO. The goals of this thesis can be summarized as follows:

a) We have implement the operators of the FOA algorithm with respect to the PFO problem.

b) We measures the performance of our mentioned algorithm with GAAM [7] for the PFO problem.

c) We use the intensification and diversification characteristics of FOA algorithm to solve the problem efficiently.

d) We have predict the three-dimensional native structure from the given amino acid sequence in less computational time than existing systems.

e) We finds optimal solutions with lower standard deviation than the present approximation algorithms.

1.5 Protein Folding Optimization Problem

Protein Folding Optimization is a procedure where an amino acid monomer sequence will be given consider as input conformation, from which we have to determine the accurate tertiary structure with actual functionality [7].

1.5.1 Protein Structure

When two or more amino acids connected by peptide bonds, then it is called protein. There are four categories of structures found in protein that are primary structure, secondary (2D) structure, tertiary (3D) structure and quaternary structure. The chain of amino acid sequence can be referred to as the primary structure. After the folding or coiling the primary structure the secondary structure (2D) can be formed. There are two levels of 2D structures observed in proteins that are alpha (α) helix and beta (β) pleated sheet. Within a single polypeptide chain the layout of the secondary structure of amino acids sequence or protein is the tertiary (3D) structure of protein or polypeptide. On the other hand the quaternary structure of protein is the alignment of the amino acid monomers in different polypeptide chains [8]. The natural view of the four different categories of protein structures is shown in Fig 1.1.

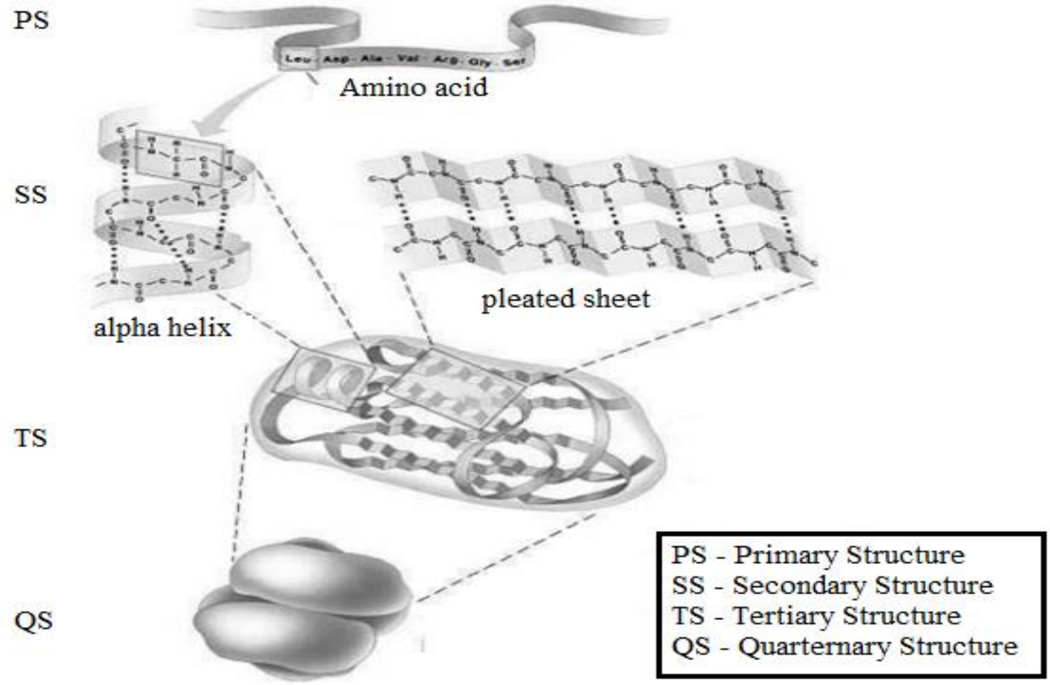

Fig 1.1. Types of protein structures [8]

1.5.2 HP (Hydrophobic-Polar) Model

The protein folding optimization procedure can be specified by a common model such as HP (hydrophobic-polar) model, which was introduced by Ken A. Dill in 1985 [9]. The PFO has been proved as NP-complete problem representing by the HP model [10]. Following of this model amino acids can be categories into two levels, hydrophobic (H) and polar (P) or hydrophilic. Within in a unit distance, the maximum amount of hydrophobic-hydrophobic (H-H) contact [7, 11] is the main theme of the HP model. Because the maximum amount of hydrophobic-hydrophobic (H-H) contacts illustrate the most static structure of protein. If C represents the most static conformation or structure, then the objective function (F_C) considering to HP model can be described as follows,

$$F_C = Max(H - H) \tag{1.1}$$

The law of thermodynamics describes that the most static conformation contains the minimum energy value. Now if E_C represents the energy value of the conformation then the energy value function can be describe as,

$$E_C = (-1) * F_C \tag{1.2}$$

When the amount of hydrophobic-hydrophobic (H-H) increases, the energy function value is decreasing. Energy minimization represents the forming the optimal structure or conformation with minimum energy value [11]. In addition, representing by the HP model, protein folding optimization problem can be regarded as NP-hard optimization problem [12].

1.5.3 Search Space Formation

Suppose, $S =$ HPHHPHPHH is an amino acid sequence. We can represent the sequence of amino acids as a set of binary strings, if the length of the sequence is l then it can be represented as a string $S = \{S_1, S_2, ..., S_l\}$ where $S_i \in \{H, P\}$ and $i = 1,2, \dots.., l$ [7]. Here the task is to determine the 3D native conformation of a protein or polypeptide from the given primary sequence. At first, candidate structures have to be chosen for the amino acid sequence uniform randomly. In the cubic lattice, an individual amino acid of a specific protein structure is represented by three dimensional coordinate (x, y, z). When no cubic lattice point is holding by more than one amino acid monomer then the conformation is called valid conformation [13]. If any overlapping occurs in the cubic lattice positions, then the conformation is called invalid conformation. For each possible candidate structure, there are some directions associated with it. If X_i represents a particular structure of the population, then the possible directions are left (L), right (R), forward (F), backward (B), up (U) and down (D) [7]. For an amino acid sequence of length l, the number of directions will be $(l-1)$. Because, the first position of the first amino acid is fixed for representing other amino acids with respect to its position.

$$X_{i,j} \in \{L, R, F, B, U, D\}\, where, j = 1,2, \ldots., l-1.$$

Protein structures are formed according to these directions in the cubic lattice where each amino acid has unique coordinate value. For example, a structure is represented as $BBBURDFULURRUR$ [7]. Furthermore, a set of candidate structures can be termed as a population.

1.5.4 Energy Value Calculation

After the formation of a population, the energy of specific solution is calculated using an energy function. In the cubic lattice, the maximum amount of hydrophobic-hydrophobic (H-H) touches for each individual amino acid is 4, but only the first amino acid and last amino can have maximum 5 touches. At the case of each amino acid is represent by i then we have to start the checking of contact its second adjacent $(i+2)$ position. Now if $c_m$and c_n are two adjacent amino acids and C is the set of all valid structures, then the energy function (F_E) is given in [7, 11] as follows:

$$F_E = \sum_{m=1}^{l} \sum_{n=m+2}^{l} a(c_m, c_n) * e_{m,n} \tag{1.3}$$

Where,

$$a(c_m, c_n) = \begin{cases} 1, if \text{ condition\_1 holds} \\ 0, \text{otherwise} \end{cases}$$

$$\text{condition\_1} = c_m, c_n \text{ are adjacent and not connected amino acids}$$

$$e_{m,n} = \begin{cases} -1, if\ c_m \text{ and } c_n \text{ are hydrophobic amino acids} \\ 0, \text{otherwise} \end{cases}$$

Where two hydrophobic-hydrophobic amino acid monomers that are contacted with each other by cubic lattice points, then H-H contact value is at a -1. And the summation of all values of H-H contacts represents the energy value of the conformation or structure [7].

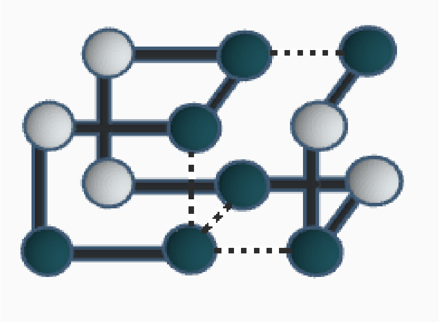

Fig 1.2. Calculation of energy value. Here, the white amino acids or monomer are P (polar) type and the blue amino acids are H (hydrophobic) type. The protein energy value $F_E = -4$ for sequence S = {H, H, P, H, H, P, H, P, H, P, H} in the representing in cubic lattice.

1.5.5 Selecting New Population

After the calculation of energy all conformations, a conformation with minimum energy value is chosen as the optimal native tertiary conformation or structure. In this thesis, a fruit fly optimization algorithm (PFO_FOA) has been used with HP cubic lattice model for determining the tertiary structure of protein. In a natural protein, hydrophobic amino acids attend to be near to the center position and the hydrophilic amino acids attend to move in the outer portion. The most static conformation of protein illustrates maximum hydrophobic amino acids in the center position of the conformation. We have reconstructed the primary operators of FOA for solving protein folding optimization procedure. Furthermore the proposed algorithm develops two efficient mechanisms, centroid hydrophobic and moderator mechanism. These two efficient mechanisms raise the contribution of the PFO_FOA algorithm sincerely. Working with these two efficient mechanisms, the three primary operators of FOA algorithm also raise the energy values of the individual solution. After applying primary and efficient operator many invalid conformation may be created. The invalid conformation represents a conformation where in a cubic lattice position two or more amino acids hold at the same position. If any invalid conformation is created after applying primary and efficient mechanisms, then the reconstruction mechanism reconstructs the invalid conformation and produces a valid conformation.

1.6 Organization of the thesis

Chapter II related works studies on PFO is described. The protein folding optimization problem has been solved by different exact algorithm, heuristic algorithm and metaheuristic algorithm. This chapter discus about the proposed technique of the existing algorithms. It also briefly describes some of the existing algorithm limitations. It also includes alluring advantages of existing algorithm by the proposed method.

Chapter III presents the proposed methodology named PFO_FOA and describes its working procedure in detail. This chapter first describes the fruit fly optimization algorithm with its operators. Then discuss about the proposed fruit fly optimization algorithm for solving protein folding optimization problem also includes the overall functional diagram of the proposed mechanism. It then describes the three basic operators design and two extra mechanisms. It also describes the reconstruction operator for invalid structure.

Chapter IV shows the dataset and simulated results of the proposed methodology. Here it demonstrates the improved performance of PFO_FOA in comparison with Genetic algorithm with advanced mechanism (GAAM) [7] which is the state of the art of this problem.

Chapter V lists the concluding remarks gathered from this study. It also includes what future direction of researches are needed to explore for more desirable PFO problem and FOA algorithm.

CHAPTER II

Literature Review and Related Works

2.1 Introduction

To predict the protein tertiary structure different approaches were proposed. The approaches include exact, heuristics and meta-heuristics which are described below.

2.2 Exact Algorithms

Exact algorithm's term is used in that process where the algorithm provides the best or optimal result of the optimization problem. In the case of complex optimization problem (like NP-hard problem) exact algorithm provide approximate value with polynomial-time, but in case of well-known optimization problem exact algorithm needed exponential time [14]. Many exact algorithms used to solve PFO problem, some of these are described below.

2.2.1 Dynamic Programming

In 2005 Zhao et al. [15] proposed a dynamic programming (DP) algorithm for protein folding optimization problem. In this algorithm, using dynamic programming procedure a directed graph is built and then find the optimum path by searching, based on the 2D structure propensity of the polypeptide or protein sequence. Here the peptide bond of the desired secondary structure can be any one element of the set $\Omega=\{\alpha\alpha,\alpha\beta,\beta\beta,\beta\alpha,\alpha\_B,\beta\_B,\alpha\_E,\beta\_E,\alpha,\beta\}$. The two dimensional structure trend to the coefficient of the protein sequence create a matrix that constitutes all the possible propensity values for the desired structure. By improving the matrix a directed graph G is drawn with k vertices from which the secondary structure of the corresponding protein can be predicted. In graph G, all the paths are directed and a path from the initial node to the last node can be termed as a solution of the PSP problem. The scoring function of the path represents a path that associate with weight. By this process by using dynamic programming approach an optimal path is computed. The algorithm provides 76.70% accuracy for the average three-state.

Advantages

Because of utilizes SSPC of dynamic programming approach can conquer the faults of the methods based on a single amino acid's propensity and the approach is faster.

Disadvantages

Dynamic programming takes exponential space for large instances.

In 2017 Sabzekar et al [16] proposed an efficient dynamic programming algorithm for predicting protein β-sheet. In this paper brute-force calculation in this formation space leads to truck with a connective detonation problem with unmanageable computational complexity. To achieve stable long range interaction, a rising approach is to detail and rank all β-sheet outline for a given protein and find the one with the highest score. The problem with solution is that the search space and the problem grow exponential with respect to the number of β-strand. Mainly generate and search the space of the problem efficiently to curtail the time complexity of the problem. Two tree structure, called sheet-tree and grouping-tree, are proposed. Firstly, more correct β-sheet structure is beginning of experimental all possible formation. Secondly, by the process of searching the space, the time complexity of the process is reduced efficiently. Which make the proposed method on target to predict β-sheet structure with high number of β-strand. The prediction of tertiary structure two types: 1) low accuracy and 2) exponential increase of the conformational space of the problem with the length of the primary sequence. The proposed method, each βstrand can merge with at most two other β-strand.

Advantages:

If an optimal solution contains optimal sub solutions then a problem exhibits optimal substructure. When a recursive algorithm would visit the same sub problems repeatedly, then a problem has overlapping sub problems.

Disadvantages:

Increasing the number of β-strands, the total number of nodes of the tree and therefore the computation time of the problem increase.

2.2.2 Greedy Algorithm

In 2005 Tuffery et al. [17] proposed a greedy algorithm for predicting structure of protein. Start at the N- terminus of the structure of protein with the end of the C-terminus the greedy algorithm performs the incremental construction. Here, the length of a protein is L+3 and the number of protein reconstructions structure to be pursued hugely. The main idea ahead greedy algorithms is that, the every i^{th} positions of any conformation , and after constructing all the possible spread of the conformation, only a highest number of hydrophilic (H) conformations (heap size) into all the structures produced are remain fixed for the next iteration. The total procedure remaining same until reach to the C-terminal vastly is performed. For a protein of length L + 3, after the total procedure all possible of reconstructing the protein structure number is $\prod_{i=1}^{L} n_i$ and the complication of the search procedure (the average number of states per remainder) is $\sqrt[L+s]{\prod_{i=1}^{L} n_i}$.

Advantages

Greedy Algorithm takes less space than dynamic programming.

Disadvantages

One problem is that here the conformations of the entire sequence are not taking account into account in the selection, without when the extremity is achieved.

2.3 Heuristics Algorithms

The main motivation of using heuristic algorithm in bioinformatics is that maximum bioinformatics problems are difficult to solving at polynomial time of their size with optimally. In many practical situations, the acceptable solution to an optimization problem can be produced by the heuristic algorithm. But the produced solutions is to be correct, there is no formal proof. Under the given constraints, when there is no acquainted process to find an optimal solution the heuristic algorithm are used. For solving the problems which are always near thought to be NP-hard (exactly, not provide solution in polynomial time) [18]. Some heuristic algorithms are described below.

2.3.1 Alphabet Reduction Algorithm

One of the heuristic algorithms is an alphabet reduction algorithm, which was proposed by Bacardit et al. [19] in 2007. This algorithm minimizes the alphabet size of the including variables that are involved for predicting the tertiary structure. At the situation of protein folding optimization problem the alphabet reduction is to convert to a two letter hydrophobic/ hydrophilic (polar) HP from twenty letters alphabet this an example at this case. An extended compact genetic algorithm was used as this mechanism for optimizing to a fixed number of type or letters from the distribution of the twenty letters of alphabet number. A rigorous information theory measure was chosen for the fitness function of such minimization process as the mutual knowledge. The Minimum Description Length (MDL) principle was used as the fitness function according to this mechanism. For maintaining the accuracy and complexity of the system the MDL principle was applied.

$$Fitness = TL * W + EL \qquad (2.1)$$

Where TL stands for theory length (which represent the complexity of the output) and EL stands for exceptions length (which represent the accuracy of the output). This fitness function has to be minimized. W is a weight that maintains the connection between TL and EL.

Advantages:

Alphabet reduction algorithm works depending on the number of letters in the alphabet. So, it works very faster than all other exact algorithms. Since the number of strings is not a concern for this algorithm, so it can compute large instance in a considerable time.

Disadvantages:

One problem is that for the mutual knowledge to provide a reliable fitness function, the dataset does not have efficient number of solutions.

2.3.2 Hybrid Hill-climbing and Genetic Algorithm

In 2011 Shih-Chieh Su et al. [20] proposed a Hill climbing algorithm for protein structure prediction problem. Here, a genetic algorithm which was based on the elite-based reproduction strategy (ERS-GA) has been proposed by the authors. On basis of ERS-GA, for the protein folding optimization on two dimensional triangular lattice problem the system has been extended with a hybrid or mixed of hill climbing and genetic algorithm (HHGA). A n length amino acid sequence consider as an input of the problem, in the two dimensional triangular lattice this amino acid sequence was encoded as a chromosome with a string of lenth (n-1) which are represent by the symbol of {L, R, RU, RD, LU, LD}, that are marked on the folding directions left, right, right-up, right-down left-up and left-down, gradually. Within a fixed range, a starting population was produced randomly in the (n – 1) dimensional space. The population size was set at fixed 200 length for experimenting in this paper. Within the population every chromosome needs to be computed its fitness function. Here hydrophobic-polar model was used to calculate the energy value as the fitness function as follows.

$$F = \sum_{i,j} \Delta d_{i,j} \; p_{i,j} \qquad (2.2)$$

Where,

$$P_{i,j} = \begin{cases} -1.0, \text{ the pair of H}-\text{H residues} \\ 0.0, \text{ Others} \end{cases}$$

$$\Delta d_{i,j} = \begin{cases} 1, \text{if } s_m \text{ and } s_n \text{ are adjacent but not connected amino acids} \\ 0, \text{otherwise} \end{cases}$$

The goal of an optimization algorithm like hybrid hill climbing and genetic algorithm (HHGA) is to optimize the fitness value consider as the free energy. According to the free energy value the computed chromosomes are sorted then. Then at the next reproduction procedure this sorted populations is considered as the base populations. On basis of the fitness value of the populations, the knowledge of the selecting solutions are copied or modified within the reproduction process. The crossover, mutation and selection are the main operators of the reproduction procedure in GA. The first half of the population has gone to the next iteration or generation and the second half of the population is used for generating offspring of the crossover and mutation operations in this procedure. After that a local search has been applied to each individual of the population for better performance in their mechanism. This process continues until the best fitness value is found.

Advantages

Hill climbing algorithm was better for finding a local optimal solution (local optimal solution represents that solution cannot become better by taking into account a neighboring aspect).

Disadvantages

Since it attempts to find a better solution by gradually exchanging a single position of the candidate solution, there is no possibility always finds the best optimum (the global solution) among all of the population in the searching space.

2.3.3 A New Heuristic Algorithm (Integer Programming Model) for Protein Folding in the HP Model

Metodi traykov et al [21] exhibited a new heuristic process basis of main integer programming model extended from a background applied to Contact Map Overlap problem at 2016. The Contact Map Overlap problem is similar to the conversion to a problem of searching like that optimizes the number of overlapping edges that indicate the HP folding problem. From the computational experiment they show that for arbitrary long protein sequences the idea decomposing the HP problem to tractable subproblems has been worked better. The length of the subproblems then changing to the PATHFINDER function and then replacing the PATHFINDER to an advanced integer programming approach able to solve optimally of all instances.

Advantages

The authors indicate that the programming of the algorithm is not difficult for that reason this procedure are easily applicable.

Disadvantages

Appling advanced integer programming models, the authors capable of finding optimal result only for instances which are less than length size 100.

Protein folding optimization in a cubic lattice in hydrophobic-polar model has been developed by Nicola Yanev et al [22] at 2016. The authors presented a new mixed integer programming formulation for solving the protein folding problem with exact algorithm, and two heuristic algorithms which are stated as a combinatorial optimization problem in a simple cubic lattice. The proposed model allows for finding optimal folds for sequences of up to 100 elements on a computer with average capabilities using a mixed linear integer programming problem with x_{ik} binary and also using appropriate solvers like CPLEX) or GUROBI.

2.4 Meta-Heuristics

A meta-heuristic represent an algorithmic framework that is totally problem independent (not depend on the problem) that provides the near optimal solution in polynomial time whereas exact algorithm fails to solve those [23]. The main difference between heuristics and meta-heuristics algorithms is heuristic algorithms are problem dependent, whereas meta-heuristics are problem independent. Meta-heuristics are developed specially to find a solution that is "good enough" for computing time. Some meta-heuristics approaches for protein structure prediction problem are described here.

2.4.1 Improved Bees Algorithm

In 2015 Nanda Dulal Jana et al.[24] developed an Adaptive Polynomial Mutation based Bees Algorithm (APM-BA) which is a nature inspired or swarm intelligence based algorithm based on the foraging behavior of honey bees colony for solving the protein structure prediction problem in two dimensional AB off-lattice model. They proposed the main strategy by their scoring are not improved during the execution phase, then the adaptive polynomial mutation technique has been applied to each of the best searching bees mutation processes. They designed a neighborhood solution of each of the say, B picked site in the local search procedures. For representing the number of iterations they used parameter trial conduct to unskilled search before better position is produced. In this paper authors show that the proposed strategy is able to make exploration of the search space and preventing stuck in local optima.

Advantages

Here have a high probability to go out from marked site to unmarked site and made inspection on the searching population space..

Disadvantages

Bees Algorithm has the pitfall when solving multimodal optimization problems the premature convergence due to absence of diversity in the search space. BA has the limitation of premature convergence due to many local minimal solutions in the search space. In this paper concerns limited number of protein sequences with shorter lengths.

2.4.2 Memetic Algorithm

A Memetic Algorithms are population-based metaheuristics for 3-D protein structure prediction problem was developed by Leonardo Correa et al. at 2016[25]. In this paper, the memetic algorithm uses a structured population and a local search strategy which was incorporated with a Simulated Annealing algorithm, as well as ad-hoc crossover and mutation operators to deal with the problem. Here designed a new strategy for extracting, representing and manipulating structural data from experimentally determined 3-D protein structures. They used structural knowledge stored in the Protein Data Bank, by using an Angle Probability List (APL) to initialize the solutions of each agent in an attempt to reduce the size of the search space and inject high-quality solutions as starting point. APL used the previous occurrences in known protein structures experimentally determined which is presenting the preferences of an amino acid residue in a specified protein according to its secondary structure.

Advantages

The proposed MA could predict good approximations to the three-dimensional protein structures, regarding structural analysis. It is possible to apply the developed method to other classes of proteins.

Disadvantages

In this paper, there is no experimental result showed on the larger protein sequences.

2.4.3 Genetic Algorithm

A genetic algorithm for protein structure prediction problem was developed in 1997 Khimasia et al. [26]. For simple lattice based protein structure prediction methods the performance of the simple genetic algorithm is introduced. In this paper two important decisions are found that are required multipoint crossovers and require a local perturbation for effectively concluded the GA procedure.

2.4.3.1 Improving genetic algorithms by systematic crossover

Another genetic algorithm for protein structure prediction problem was developed by Unger and Ron in 2004 [27]. In the conventional Monte Carlo steps, structures change through mutation. At the crossovers procedure the polypeptide chain is exchanged between conformations. Different genetic operators are repeated up to valid structures are created.

A genetic algorithm with 2D protein folding simulations with the HP model (GAOSS) has been proposed by Chenhua Huang et al. in 2010 [28]. In this paper only nine benchmark dataset used for computation. The authors of GAOSS compared their simulation results with other four corresponding mechanisms. The comparison ensures the possibility of finding more protein structures and also ensure the computing speed of searching protein structures.

2.4.3.2 Genetic algorithm with particle swarm optimization

A particle swarm optimization based genetic algorithm with advanced mutation has been proposed by Cheng-Jian Lin and Shih-Chieh Su in 2011 [29]. GA is disabled for efficient mutation operation aimed at each residue. In which, the mutation was based on particle swarm optimization and the cognitive components encourage the particle to move toward their own best positions. By calculating the difference between the current particle and the local best particle and the difference between the current particle and the global best particle determine the variations of the current position. Through, the mutation process only one child replaces its parent and go to the next generation and PSO improves the mutation mechanism. Mutation plays an important role in PSO to ensure the searching capability of near global optimal solution. A phenotype based crowding mechanism in 2014 developed by Custódio et al. [30] for maintenance of useful diversity within the populations. The distanced between two individual 3D structure are calculated by crowding mechanism for the positioning of hydrophobic monomers. The algorithm granted multiple solutions capabilities for this reason. The computationally expensive term in this process is $\sqrt{\sum_{i=1.\ j<i}^{N} \frac{(p_{ij} - q_{ij})^2}{N(N\text{-}1)/2}}$. Where N is the number hydrophobic amino acids and p_{ij} (resp.q_{ij}) denoting the distance between H monomers i and j on the parental (resp. new) structure.

2.4.3.3 Genetic algorithm with advanced mechanisms

The complication of protein structure prediction (PSP) acts as the computational problem to predict the real structure of a protein from its amino acid sequence. In this paper [7], tested to the protein structure prediction in a hydrophobic polar model on a cubic lattice. This Genetic algorithm different system used and extended with crowding, clustering, repair, local search and opposition based. Here, the population P of their algorithm generated popSize (population size) solutions $xi = 1,2,3 \dots popSize$. Each population has a fixed amino acid length (L) size L-1 absolute direction $: x_i, j = \{L, R, U, D, F, B\}$; $i = 1,2 \dots posize \ and \ j = 1,2 \dots L - 1$. For every produced sub part of the population the algorithm used different point (one, two or multi-point) crossover randomly applied and the mutation segment randomly selects one, two or three directions. The mutation operation ensures that improve the efficiency of the native structure of protein.

The local search used to improve convergence speed and raise the quality of conformation by consecutive monomer where one of the consecutive monomers must be hydrophobic (H). If the local movement of 1 or 2 consecutive monomers improves the native structure, then the movement is accepted or not rejected. Infeasible solution to feasible solution that don't attend the lattice point for more than one monomer it's the repair mechanism used the backtracking algorithm. Here, infeasible solution means that two or more amino acid in the protein sequence occupied the same lattice point of the corresponding cubic lattice model. Here, the local search performed by repair and evolutionary process with one or two consecutive monomers movement.

Improved solutions are then compared with the localest solution of the population P and the closeness between two individuals solutions are determine by computing hamming distance of the individuals. Hamming distance determines the difference between corresponding point direction of two individual structures. The crowding mechanism can compared between two individuals. Crowding mechanism ensures that a good solution formed at the last of the process of evolutionary.

Advantages

Genetic algorithm provides optimized local search by multiple pathways. In general, genetic algorithm gives outstanding performance due to its inherited advantages [7].

Disadvantages

Crossover and mutation are two key operations in the genetic algorithm. Crossover have the capability of linking two different chromosomes very robustly for constructing a new conformation, while mutation may be restricted by local optimum [7]. GA always found the optimum solution, a little more slowly than the dynamic programming for the smaller instances.

2.4.4 Particle Swarm Optimization

In 2012 Mansour et al. [11] developed an efficient PSO algorithm for 3D structure of protein. By exploring search space of probable solutions the 3D structure with minimum free energy has been returned. Here, a selective solution of protein sequence called as a particle which designed by a n length's array (with indexing 0, 1, ..., n-1), where n represents the of the selected amino acid sequence of protein. The d^{th} position of the amino acid sequence is represented by X_d and the actual value of this position or element may be one of the six possible directions {b, f, u, d, l, r}. Staring with a randomly created set of solutions N or particles collected in a swarm. That is, every position X_d of amino acid d (d = 1, 2, ...,n-1) is addressed a random value for the selective solution/particle i (i = 0, 1, ..., N-1). A particle is invalid if it occupied the same value on the 3D cubic structure. In this algorithm, invalid particles are reconstructed using a backtracking algorithm with repair function. Then each hydrophobic amino acid of the sequence is searched for any non-consecutive (monomer not connected by direct bonding) and around of six positions of the lattice which are hydrophobic amino acids and these particles position are scored as energy values function calculation. The main operation of the algorithm, update the swarm location using the velocity of the particle at every stage. Finding the optimal solutions these operators explored new searching areas in the search space. When two successive iterations may not improve the observed result, then the algorithm terminated.

2.4.4.1 Discrete particle swarm optimization

Particle swarm optimization (PSO) is a swarm based computer intelligence algorithm. A discrete particle swarm optimization algorithm ($DPSO_{HP}$) for solving the secondary and tertiary structure of protein with HP lattice models-based protein structure prediction problem has been developed in 2014 by Xiao et al. [31]. Based on the set concept and the possibility theory of DPSO$_{\mathrm{HP}}$ on a set-based PSO (S-PSO) used the discrete particle swarm optimization method.

A particle in the algorithm is defined as a set of elements. The velocity of a particle is defined as the elements associated with possibility. For both tertiary and secondary structure of protein, starting from the center of the cubic lattice or square lattice of the particles construct a sequence of proteins at the middle position of the respective lattice. For solving overlap in the amino acid sequence within the lattice position at first the protein sequence is organized by HP model. PSO is mimics the movements of organisms in a school of flying birds. When searching solution (food) in a gradual space, adjusting each particle (bird) flying velocities and positions at step by step according to the particle own experience and other particle experiences. Here, the special velocity and position updating is the bodily representation of the framework of DPSO$_{\mathrm{HP}}$.

Velocity and position are two main attributes of every particle $i\ where, i = 1, 2, \dots, m$, which are denoted by $V_i\ = < v_i^1, v_i^2, \dots\dots v_i^n >$ and $x_i\ = < x_i^1, x_i^2, \dots\dots x_i^n >$. Where $j\ (j = 1,2, \dots, n)$ denotes the j$^{\mathrm{th}}$ dimension of the particle i. The vector $pbest_i\ = < pbest_i^1, pbest_i^2, \dots\dots pbest_i^n >$ is the best solution found by particle I so forward and the vector $gbest_i\ = < gbest_i^1, gbest_i^2, \dots\dots gbest_i^n >$ is the optimal solutions found by all the particles so advanced.

The velocity updating process starts by calculating Coefficient×Velocity

$$\omega v_i^j = \left\{ \frac{e}{p^{'}(e)} \middle| e \in E \right\} \tag{2.3}$$

Where ω an inertia weight. Then respectively determine the values of $pbest_i^j - x_i^j$, generate random number r_{ij} from [0, 1]. Then determine $cr_{ij}(pbest_i^j - x_i^j)$ and $\omega v_i^j + cr_{ij}(pbest_i^j - x_i^j)$ and update the value of v_i^jat equation (7).

Using the newly updated value to improve its solution efficiency for every particle the position updating process is done. Two middle amino acid of a protein sequence Mid-1 and Mid, respectively, where, Mid = $\lceil n/2 \rceil$ is chosen for path construction. After choosing middle amino acids, these monomers are placed in the center position in the lattice or cubic board. Then the particle randomly selects left or right part for folding.

When all amino acid occupied every possible position in the lattice, then no new amino acid can be placed on the path construction procedure. During the construction phase for that reason protein cannot fold anymore, solving this problem a path retrieval procedure applied for producing feasible structure from infeasible structure. In this process the total sequence divided into two parts. Then checked which portion (Left or Right) of the sequence the motionlessness occurred and performed path retrieval mechanism.

Another particle swarm optimization mechanism for solving protein structure prediction problem has been introduced by by Hamed Khakzad at al. in 2015 [32]. The authors of this paper introduced a graphical processing unit (GPU) based parallel architecture for accelerating the PSP problem. The pitfall of the proposed algorithm provides slow performance when implemented on the CPU.

Advantages

> With respect to solution accuracy and speed, the PSO algorithm provide efficient result for searching for the ground states of the protein structures. By applying the PSO on the mutation process in GA optimized the solutions [28]. This allows greater diversity and exploration over a single population.

Disadvantages

> The algorithm provides better results for short length protein, but the result of the long length sequence the search space is not efficient [11].

2.4.5 Ant Colony Optimization

In 2002 Shmygelska et al. [33] exhibited the ACO approach for protein folding optimization problem. In which, they have highlighted the global pheromone update and randomly chosen starting position in the folding procedure.

2.4.5.1 An improved ant colony optimization algorithm

In 2003 the same authors improved the ACO algorithm [34] including long range moves that allow performing updates of the protein with high densities, the used of selected local search and improving ants. The time complexity of the local search phase has been reduced by selective local search reduces with time critical operation only on promising is performed, low energy conformations.

In 2006, D. Chu al. developed a parallel ant colony optimization for predicting the 3D protein structure prediction problem [35]. The simulation result of the parallel ant colony indicate that this procedure is outperformed and climbed than the single colony optimization algorithm.

2.4.5.2 ACO-metaheuristic for 3D-HP protein folding

The concept of Ant Colony Optimization (ACO) was used for solving tertiary structure of protein with HP model for protein folding optimization in 2015 by N. Thilagavathi and T. Amudha. [36]. ACO is a meta-heuristic which mimics the foraging nature of real ants for solving optimization problems. The concept of foraging is that the members of the population will communicate indirectly with each other. Ants are agents that build structures and give solutions. Each ant gives a unique structure. The best solution (structure) is then chosen among all possible solution. Here HP cubic lattice model was chosen for computing free energy. The cubic model has 6 possible moves as F (Forward), B (Backward), L (Left), R (Right), U (Up) and D (Down) to pas a remain in the lattice. Each residue has six possible movements in six directions; among the possible directions, one direction was selected which is minimum. After determining every possible direction, here checked to mark whether the movement leads towards the correct direction. In lattice model, the hydrophobic (H) interactions are driving force for protein folding. And each conformation must be taking a self-avoiding path. No two amino acid monomers not occupied same cubic lattice position. For each amino acid sequence the energy value has been computed depend on the number of H-H contacts. Getting the best tertiary structure of protein the concern is to minimize the energy value. Based on the pheromone trail of the move and the heuristic information the possibility of transition was chosen. At the starting , the level of pheromone is fixed at a constant value and after the construction phase the pheromone value has been updated. By two stages: Local update and Global update, the pheromone value are modified. All iterations, ants modify the pheromone value internally while marked every path.

$$P_{ij} \leftarrow (1-\rho) P_{ij} + \rho\, P_0 \qquad (2.4)$$

Where the pheromone amount of two position (i, j) within the tertiary cubic lattice model is represents by P_{ij} , the term (1- ρ) can be interpreted as trail evaporation and ρ is the persistence of the trail . P_0 is set a small positive constant value which is the pheromone initialization value. At the global update, the level of pheromone was reduced and this will be reduced the possibility. As a result, the searching process will be more different.

$$P_{ij} \leftarrow (1-\rho) P_{ij} + \Delta P_{ij} \tag{2.5}$$

Where,

$$\Delta P_{ij} = \begin{cases} -E_{gb}, if\,(i, j) \in \text{best solution} \\ 0, \text{otherwise} \end{cases}$$

E_{gb} represents the global energy of the best folding structure. On the paths of the optimal solution, the global update rule is used to provide a greater amount of pheromone. The next step of pass has been chosen on heuristic information and pheromone matrix value at the construction phase. In this paper three different types of energy functions have been applied. The energy functions are represented by D85, K99 and I09. The letter of the energy function describes the first character of the author last part of the published year.

Energy function D85

From the local structure of protein sequence, the substitute energy value was calculated, the name of this energy value is 'Free Energy (FE)' function. The energy value is score as -1 when both amino acid Si and S_j are not adjacent and both are hydrophilic H amino acid and also have topological contact between them. Otherwise, it is scored as 0. If, 'c' represents the conformation state of a protein sequence,

$$E_{D85}(c) = \sum_{s_i s_j} E(s_i, s_j) \tag{2.6}$$

$$\text{Where, } E(s_i, s_j) = \begin{cases} -1, if\ s_i \text{ and } s_j \text{ are both H and form a topological contact} \\ 0, \text{otherwise} \end{cases}$$

Energy function K99

Depending upon the closeness between two hydrophobic (H) amino acid the energy value has been calculated. All nonadjacent amino acids, which are topological connected are used in this energy calculation.

$$E_{K99}(c) = \sum_{s_i, s_j} E(s_i, s_j) \tag{2.7}$$

Where,

$$E(s_i, s_j) = \begin{cases} -1, if\ s_i \text{ and } s_j \text{ are both H and form a topological contact} \\ -\dfrac{1}{d_{s_i, s_j} kL_H}, if\ s_i \text{ and } s_j \text{ are both H but the difference between them is} > 1 \\ 0, \text{otherwise} \end{cases}$$

Energy function I09

The reconstruct fitness function (E_{I09}) is defined as

$$E_{I09}(c) = \alpha E_{D85} + H_C + P_C \tag{2.8}$$

Here, α is a constant value that ensures this will be the prevalent term within high integer constant value. H_c denotes H-compliance, which measures the distance of H amino acids to the core or center. P_c denotes P-compliance that computes how close polar (P) amino acids are to the outer border.

In this procedure at starting population start with five solutions or ants and each solution, try to fold every possible direction by folding process and produced different folding structures or solutions using the energy functions. The computational complexity of the methodology represents number of possible structures,

Possibility of structure = n^6

Where,

n = Structure length

6 = All possible directions in cubic lattice

Advantages

The algorithm uses three different energy functions that provide good solutions to this problem.

Disadvantages

The initial population starts with only five solutions or ants this is a major obstacle for finding the best solution.

2.5 Multi-objective optimization algorithm

Multi-Objective (MO) optimization has been proposed in 2015 by Garza-Fabre et al [37] for protein structure prediction problem. Here, HP lattice model has been used which is an abstract formulation for PSP problem. Under the HP model, the PSP can be defined as finding a best structure such that the total number of interactions among hydrophobic amino acids is large as possible. MO approach is used as a substitute constraint handling approach in the sense that infeasible solutions can also provide important information for solving PSP problem.

Single-objective optimization refers to solving a problem based on only one objective function. For Single-objective optimization problem, the task is to search a structure x such that

$$x \in X_F \qquad (2.9)$$

$$\text{And minimize } OF(x)$$

Here X$_F$ denotes the possible, feasible set and OF(x) is the objective function of x. The destination of this procedure is to find the optimal solution(s) based on the objective function, so that

$$x_{op} \in X_F \qquad (13), \qquad \text{such that } E(x_{op}) = min\{E(x) | X_F\}$$

Similarly, a multi-objective optimization problem can be accurately defined as solving a problem based on multiple objective functions. For multi -objective optimization problem, the task is to search a structure x such that

$$x \in X_F$$

$$\text{And minimize } OF(x) = [OF_1(x), OF_2(x), \ldots\ldots\ldots OF_k(x)]^T$$

Where T defines a set of trade-offs among the conflicting objectives. Here, a fitness landscape has been introduced which consists of a triplet term (X, N, ξ). The first element represents the search space, the second element represents a function that maps each possible solution to a set of solutions or neighbor solutions and the third element defines an ordering relation between the solutions and is directly related to the objective function.

Protein structure prediction (PSP) problem can be stated as an energy minimization problem. In HP model, amino acids can be categories as into two types (hydrophobic (H) and hydrophilic (P) types) based on their hydrophobicity nature. And protein can be termed as a chain of H and P type monomers. The goal of the problem is to optimize the H-H contacts using a cubic lattice. Such contacts are referred to as topological contacts. There is an energy function that design every conformation, c with an energy value (E_v) as stated in [20].

$$E_v(c) = \sum_{s_i, s_j} e(s_i, s_j) \qquad (2.10)$$

Where, $e(s_i, s_j) = \begin{cases} -1, if\ s_i \text{ and } s_j \text{ are both H and they form a topological contact} \\ 0, \text{otherwise} \end{cases}$

The PSP problem with hydrophobic-polar model can be represented in terms of multi-objective optimization by defining an extra objective function. With MO optimization, a two-objective function equation of the PSP problem can define as follows:

$$OF(x) = [OF_1(x), OF_2(x)]^T \qquad (2.11)$$

Where,

$$OF_1(x) = E(x)$$

$$OF_2(x) = Collisions(x)$$

Where x is a particular conformation, $OF(x)$ is the objective function of x. Here, $OF_1(x)$ and $OF_2(x)$ are to be minimized.

On three-dimensional cubic lattice, five directions have been introduced here {F (front) , U (up), D (down), L (left), R (right)}. And on the two dimensional lattice, there are 3 directions {F, L, R}. No backtracking algorithm has been used because the algorithm ensures each conformation to be one step self-avoiding. The first amino acid is assumed to be fixed and has forward direction. For a sequence of length l, there are $l-2$ encoding decisions that has to be taken. Also, a feasible conformation must have two properties: connectivity and self-avoidance.

Advantages

Two objective functions have been used here that provides better performance than one objective function.

Disadvantages

The MO approach can be proved not efficient when a search trend towards the possible are is not predefined.

2.6 Discussion

Form this literature review and related works studies, it is clear that different categories of algorithms are already applied for solving the well-known protein folding optimization problem. But none of this algorithms are not accurately predict the tertiary structure of protein. All of these algorithms have some drawbacks. For all of these reason, we applied a new metaheurestic algorithm the Fruit Fly Optimization algorithm (FOA) for solving the protein folding optimization problem.

CHAPTER III

Fruit Fly Optimization Algorithm for Protein Folding Optimization Problem

3.1 Fruit Fly Optimization

Fruit fly optimization algorithm is the most recent transformative computation technique which was called attention to by Wen Tsao Pan in 2011 [38].The Fruit Fly Optimization Algorithm (FOA) is another canny strategy on the food finding behavior of the fruit fly, It impersonates the foraging behaviors of drosophila. The fruit fly itself is better than different species in sensing and perception, particularly in osphresis and vision. The osphresis organs of fruit flies can discover a wide range of aromas drifting noticeable all around; it can even smell food source from 40 km away [5,37]. At that point, after it draws near to the food area, it can likewise utilize its sensitive vision to discover food and the organization's rushing area, and fly towards that heading as well. Because of its benefits of being straightforward, actualize, and use the problem-oriented specific search operators, the FOA applied in some various fields, for example, Travelling Salesperson Problem [39],multidimensional knapsack problem [40],structural damage identification [41],annual power load forecasting model [42],Tuning of PID Controller [43], Twin support vector machines [44], financial distress [5], semiconductor final testing [45], and steelmaking casting [46], project scheduling problem [47]. On one hand, with respect to the exhibition of the FOA, it has demonstrated promising potential in tackling the complex problems. The behaviors of the fruit flies could be demonstrated in Fig 3.1.

3.1.1 Characteristic

Fruit fly's food discovering characteristics is isolated into a few essential strides as appeared in Fig 3.1, and the steps could be given as pursues ;

1) Random primary fruit fly swarm location is,
   Init A_axis
   Init B_axis

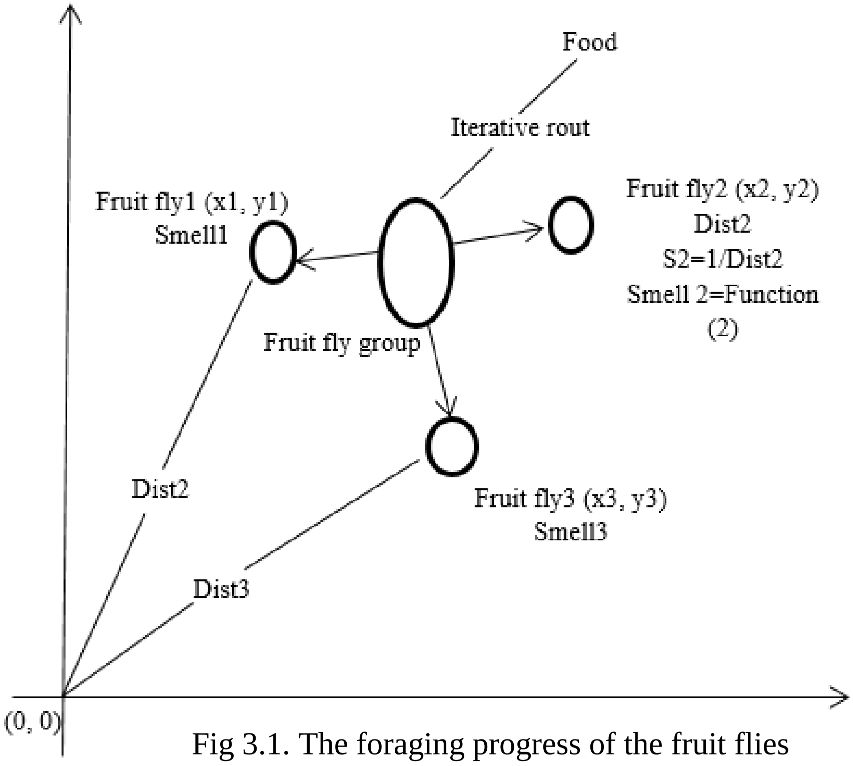

Fig 3.1. The foraging progress of the fruit flies

2) Find with random direction and distance to the olfactory organ.

   Ai = A_axis + RandomValue

   Bi = B_axis + RandomValue

3) Since food's location is unknown, the distance (Dist) to the archetype is calculate first, and the define value of smell intentness (S), which is the opposite of distance, is then calculated.

   $Dist_i= \sqrt{}(A_i^2 + B_i^2)$

   $S_i = 1/Dist_i$

4) Substitute smell intentness decision value (S) into smell intentness define function (fitness function) so as to search the smell intentness ($Smell_i$) of the separate location of the fruit fly.

   $Smell_i = Function(S_i)$

5) Search the fruit fly with maximum smell intentness (searching the maximal value) in the fruit fly swarm.

[bestSmellbestIndex] = max(Smell)

6) Keep best smell intentness value and a, b coordinate, the fruit fly swarm will usage vision to fly unto that location.

   Smellbest = bestSmell

   A_axis = A(bestIndex)

   B_axis = B(bestIndex)

7) Enter recurrent optimization to repeat the impersonation of steps 2-5, then define if the smell intentness is upper to the preceding iterative smell intentness, if so, implement step 6.

3.1.2 Operators

3.1.2.1 Smell-based search

In the standard FOA, the smell-based inquiry is the center pursuit strategy. During the technique of smell-based inquiry, S natural product flies is created around the area of each swarm and such created organic product flies build the sub-swarm. To comprehend the MSRCPSP, neighborhood based inquiry administrators focusing on the errand arrangement and the asset task are intended to actualize the smell-based hunt. Tooth and Wang [48] proposed a viable strategy to understand the RCPSP, which utilized an administrator by swapping two nearby exercises without priority relationship.

3.1.2.2 Knowledge-guided search

The practices of natural product flies are improved in the FOA. Basically, the hunt practices of natural product flies are constrained to an irregular pursuit utilizing olfactory in the FOA. Facilitate nourishment looking [49] notwithstanding the smell-based inquiry and vision-based inquiry. The natural product fly may effectively neglect to discover the nourishment source through just the irregular olfactory prompts, particularly in an unpredictable situation. Consequently, the sanctioned FOA can't accomplish great arrangements when being applied straightforwardly for complex streamlining problems. The information base comprises of two sections, for example the experiential undertaking list gave by the best natural product fly and the experiential probability of asset task for each assignment.

3.1.2.3 Vision-based search

After the smell-based inquiry, S organic product flies are created in each swarm. In the neighborhood vision-based pursuit, the arrangements in each sub swarms are assessed, and afterward the best produced arrangement is chosen to supplant the focal area of the sub swarm if better outcomes can be gotten. That is, it needs to select one best arrangement from S+1 arrangements. The TOPSIS [50] is a broadly embraced choice strategy for the enhancement issues with numerous criteria. It positions competitor arrangements dependent on the criteria data. What's more, the TOPSIS gives a quantitative measure to how great an answer is among a lot of up-and-comer arrangements, which is proper to finish the determination in the vision-based pursuit. Along these lines, the TOPSIS is utilized to choose the best arrangement in each sub swarm. The non-overwhelmed arranging strategy (NST) [50] is additionally generally utilized in the choice procedure in multi-target advancement issues. The NST underscores the uniform dispersion of arrangements just as the nature of arrangements, which is valuable for a way to deal with spread the whole Pareto front. Therefore, the 10 NST is employed to sort the candidate solutions in the global vision process.

3.1.3 Algorithm

Step 1: *Initialization, initialized the headquarters position of the swarms and the parameters of the algorithm.*

Step 2: *Smell-based search.*

Step 2.1*:* ***For*** *every fruit fly, randomly ordain a postulant location for the food source near the headquarters location of the swarms.*

Step 2.2: *Count the smell intentness of every individual fruit fly location.*

Step 3: *Vision-based search.*

Step 3.1: *Ordain the most possible location with the largest smell concentration.*

Step 3.2: *The fruit fly swarms into the location and the location of the swarm headquarters is updated.*

Step 4: *Stopping the condition. If the stopping condition is met, the algorithm ends; otherwise, repeat step 2 and step 3.*

3.2 Protein Folding Optimization using FOA

The fruit fly optimization algorithm (FOA) is a newly proposed metahuristic algorithm that's mimic the behavior of fruit flies. The fruit fly algorithms is a bio-inspired algorithm. It is inspired by the nature of fruit flies. The fruit fly is the best optimization of other fly species, especially in sensing and perception. The fruit fly searches the food by source using their smell and vision organs. Firstly they use osphresis organs to find all kinds of scents in the air. Then they fly toward to food. When they get close to the food, they use their vision organs to get closer. Using the intensive and diversity characteristics of fruit fly optimization algorithm searching process are divided into two stages that are smell phase and vision phase [5]. Vision phase included process of searching, local vision phase and global vision phase. There FOA divided into three parts: first the initialization of the optimization problem with different parameter values, post process, visualize result and this process continue until optimal value found [47].

The protein folding optimization problem are represented by the way, from the given sequence of amino acid that represent the primary structure using HP model create remove the self-avoiding path of protein sequence and produced a sequence that are represented by only H and P value. For example, HPHHPPHHHHPHHHPPHHPPHPHHHPHPHHPP HHPPPHPPPPPPPPHH is a sequence of 48 length amino acids that represented by HP model. For creating the search space or the initial population, produced different initial structure according to randomly selecting direction among all possible directions, for example, FRBULLLURRFLLDRRRRFLLLURFLLDLULUUUURBBBBBBBBBRR represent one initial structure that is randomly created by choosing random directions. The representation of directions for solving the protein folding optimization problem in the cubic lattice model are shown in Fig 3.2.

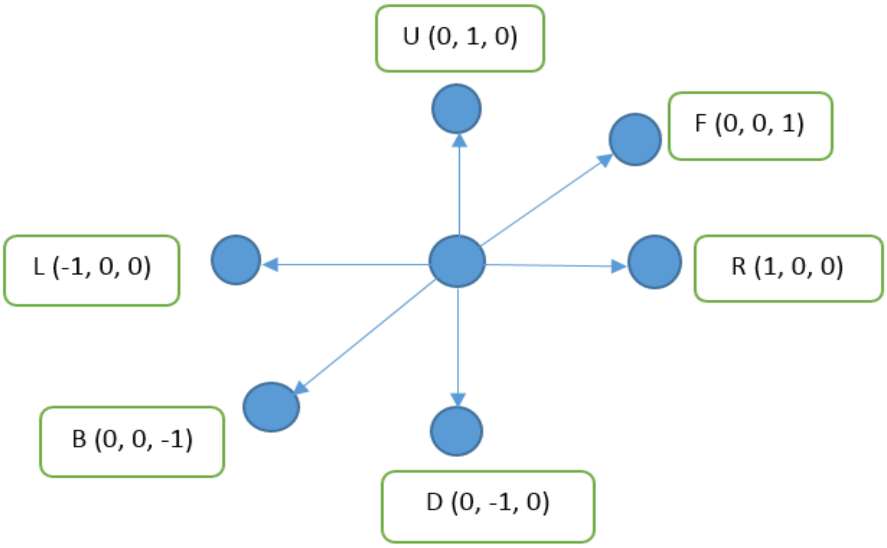

Fig 3.2. Direction representation in cubic lattice

After creating initial population apply different basic operators of selecting algorithm and try to search the optimal structure using intensive characteristics of the choosing method. After completing all the iteration process of the proposed method the optimal structure is chosen among all the structures in the search space. The final structure of protein sequence folding is RUUULDDFDRUULUFDRRDLDLULDBLBRBUFFLFURUFFLBBBRDB. The protein folding problem at a glance are shown in Fig. 3.3. In this figure the green monomers of the protein structure indicate hydrophobic amino acid and the blue monomer indicate the hydrophobic or polar amino acids.

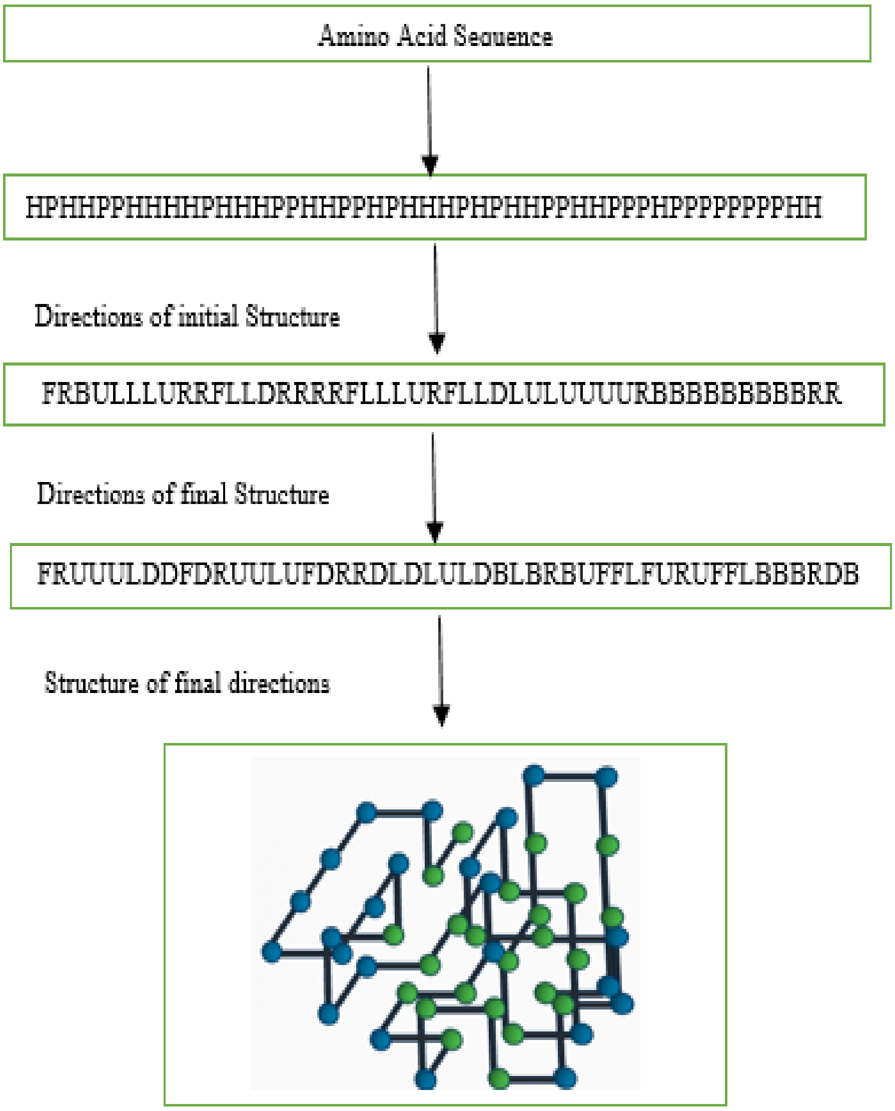
Amino Acid Sequence
HPHHPPHHHHPHHHPPHHPPHPHHHPHPHHPPHHPPPHPPPPPPPPHH
Directions of initial Structure
FRBULLLURRFLLDRRRRFLLLURFLLDLULUUUURBBBBBBBBBRR
Directions of final Structure
FRUUULDDFDRUULUFDRRDLDLULDBLBRBUFFLFURUFFLBBBRDB
Structure of final directions

Fig 3.3. Protein folding problem at a glance

The total process of PFO process using fruit fly optimization process is shown in a block diagram in Fig 3.4. The pseudo code of this proposed mechanism describes in algorithm 3.1.

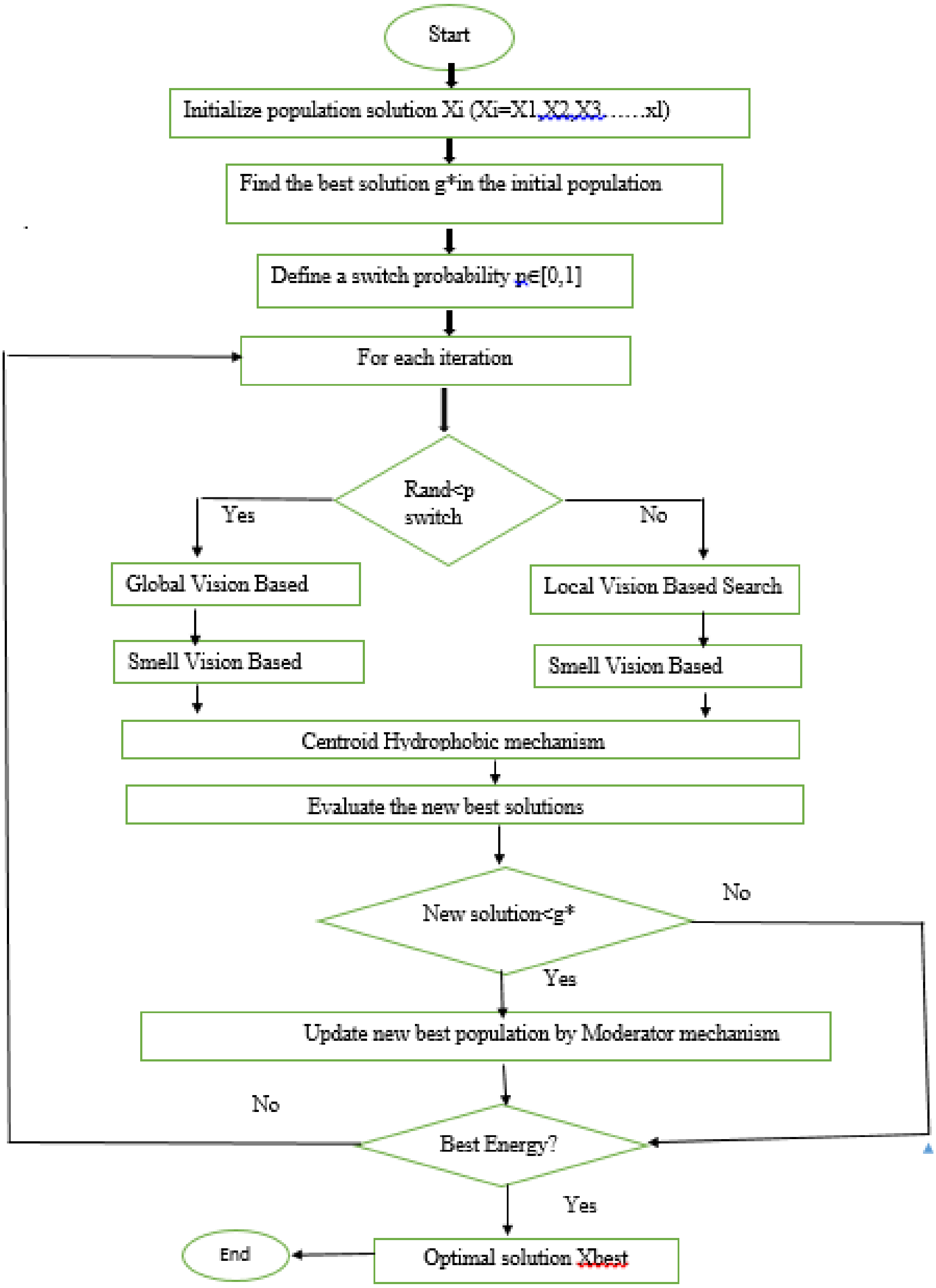
Start
Initialize population solution Xi (Xi=X1,X2,X3……xl)
Find the best solution g*in the initial population
Define a switch probability p∈[0,1]
For each iteration
Rand<p switch
Yes
No
Global Vision Based
Smell Vision Based
Local Vision Based Search
Smell Vision Based
Centroid Hydrophobic mechanism
Evaluate the new best solutions
New solution<g*
No
Yes
Update new best population by Moderator mechanism
No
Best Energy?
Yes
End
Optimal solution Xbest

Fig 3.4 Block diagram of PFO_FOA

3.2.1 Initialization of search space

The initial population of the PFO problem are created by randomly selected directions from all position of directions. For this problem the initial population set are represented by a one dimensional array. The length of the sequence is represented by *l* and the array length will be *l*-1 because we omit the first positions as it is fixed. Using six direction-L,R,U,D,F,B directions of three dimensional cubic lattice create initial search space where, L,R,U,D,F,B means left, right, up, down, forward, backward respectively. For creating the initial search space, instance of the population is generated randomly among these six possible directions. For example, in Fig 3.4. a structure of sequence is {H,H,P,P,H,H,H,P}. The first positions is fixed and no directions is set into this positions. In second step left directions left (L) has been set randomly. Similarly in third step upward (U) has been set and in step 4 ,5, 6, 7 and 8, right(R) , forward(F), right(R) down (D) and back (B) has been set respectively. After completion of the eight steps a new structure is created which represent an initial structure of the initial population. Similarly population size structures are create for executing next procedure. The process of producing initial search space is depicted in Fig 3.4.

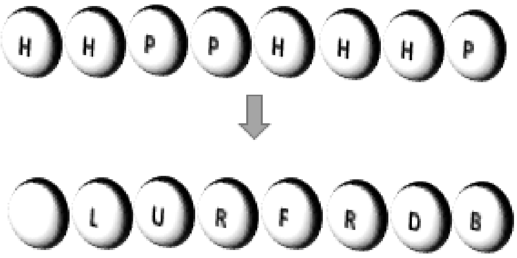

Fig 3.5. Initialization of search space

Generating an individual conformation or solution check the solution already exist or not. When generating the solution does not exist in the population then it accepted otherwise discard the conformation and generate new one. In the initialization process, the parameters of FOA algorithm are also initialized. A stopping criteria based conditional loop start executing until met condition. Every solution of the population is iterated and modified by the three operators of FOA algorithm. At first any conformation or structure modifies by the Smell Based Search then it modified by the Local Vision Based Search after that, this conformation mixed to another conformation in the population modified by the Global Vision Based Search operator. At each step of the operation of any operator the reconstruction operator perform for producing valid conformation from the invalid conformation that may be produced all the operators procedure.

Algorithm 3.1. PFO_FOA

```
Input:
    Number of iteration G
    Population size of the fruit fly swarm M
Initialization:
    Randomly generate a fruit fly swarm's initial location X(1), Y(1), Z(1) For i=1:1:M
    Randomly assign each fruit fly a direction and distance $X_i$=X(1)+ Random value ,
    $Y_i$= Y(1)+ Random value, $Z_i$= Z(1)+ Random value
    Calculate $Smell_i$ = Function($S_i$) and find out the best smell concentration
        bestSmell =max( Smel).
    Set the best smell concentration  Smellbest = bestSmell
Searching:
While K=1:1:G or the stopping criteria not met do
    For i=1:1:M
        SBS = Smell Based Search ( conformation, population[i])
        SBS = reconstruction (conformation, SBS)
        {LV1, LV2}= Local Vision Based Search ( conformation, population[i], population[j]
        LV1 = reconstruction (conformation, LV1)
        LV2 = reconstruction (conformation, LV2)
        {GV1, GV2}= Globall Vision Based Search ( conformation, population[i],
        population[j])
        GV1 = reconstruction (conformation, GV1)
        GV2 = reconstruction (conformation, GV2)
        CH = Centroid Hydrophobic (conformation, population[i]))
        Mod = Moderator (conformation, population[i]))
        CH = reconstruction (conformation, CH)
        Mod = reconstruction (conformation, Mod)
    End for
End for
population[newP] = bestEnergy(SBS,LV1,LV2, GV1, GV2)
output: bestEnergy, meanEnergy, stdEnergy
```

3.2.2 Smell Based Search

This is an operator that included in the initial iteration stage of our proposed methodology for updating the conformations. A smell based search process take a backup of the selecting solution and select some random portions of the selecting conformation. After that the selecting part try to fold in different directions, that directions are randomly selected. If the choosing new directions for that part make better energy value than the previous structure then it added to the population and go to the next iteration process. Here, one part is chosen from the given example sequence depending on the smell value. In this part have two monomers and the directions in that part are replaced by two directions that are randomly selected. In this type of operator operation, a little change occurs within the protein structure that shown in Fig 3.5. Here, the selecting part UR are folded to FB that are randomly choosen. The updated solution is the output of this operator. The pseudo code of the process is given in Algorithm 3.2.

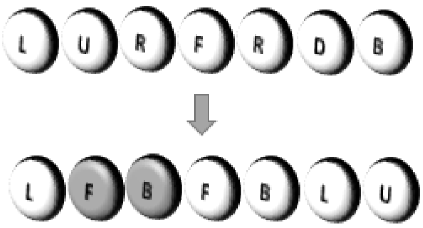

Fig 3.6. Smell Based Search

Algorithm 3.2. Smell Based Search

```
procedure smellBasedSearch (hp_sequence, folding_solution)
l = lengthOfhp_sequence
Duplicate folding_solution to form new_folding_solution
Set lPosition randomly from 1, 2, 3,. . ., l
s = Rand(0,(l- lPosition))
for i = s to (s+lPoint) do
        x_axis = rand(2,5);
            If(b=0) Then x_axis= x_axis+1, and b=1.
           Else x_axis= x_axis-1, and b=0.
           End if
           new_folding_solution [i] = direc_name[x_axis]
End for
output: new_folding_solution
End procedure
```

3.2.3 Local Vision Based Search

At local vision based search operator, firstly two random conformations are selected for modification. From the selecting conformation, the first conformation is divided into two parts depending on the value of local vision point (LV1), which chosen randomly. Then one of the two parts are selected and the selecting part is folded to opposite direction of its parent folding direction. The directions of the selected part are changes by opposite or alter direction of its own folding, such as F to B, U to D, L to R and vice versa. For the second selecting conformation, similarly divided and chose one part. The selecting part folding directions are changes according to the direction of the first selecting conformation direction of this part. In Fig. 3.6 first conformation divided into two parts or block, the first part directions are changes based on opposite direction and the last block is remain same as its parent. Second conformation dived into three parts, the starting and ending parts are the same folding direction of its parent and the middle part are folded according to the directions of the same part of the first conformation. Here, the first solution first portion LUR are folded to RDL and the second solution middle portion are folded in the direction of first solution middle portion RFB. The pseudo code of the process has been given in Algorithm 3.3.

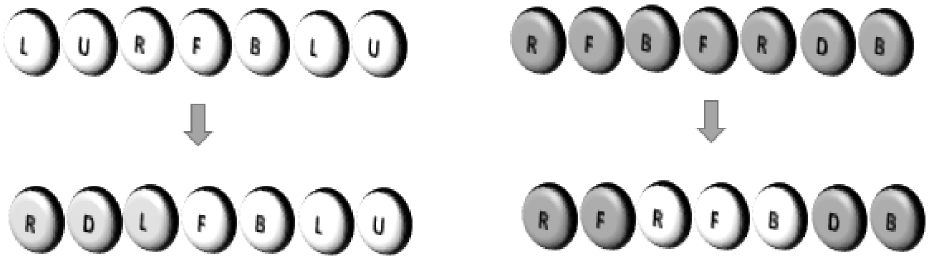

Fig 3.7. Local Vision Based Search

Algorithm 3.3. Local Vision Based Search

```
procedure localVisionBasedSearch (hp_sequence, folding_solution1, folding_solution2)
l = lengthOfhp_sequence
Duplicate folding_solution1 to form new_folding_solution1
Duplicate folding_solution2 to form new_folding_solution2
Set lPosition randomly from 1, 2, 3,. . ., l
s = Rand(0,(l- lPosition))
for i = s to (s+lPoint) do
```

```
        If(lv1=0) Then x_axis=
Opposite_direction_value[folding_solution1[i]], and lv1=1.
        Else x_axis= Direction_value[folding_solution2[i]], and lv1=0.
        End if
      new_folding_solution1[i] = direc_name[x_axis]
        If(lv2=0) Then x_axis=
Opposite_direction_value[folding_solution2[i]]+1, and lv2=1.
        Else x_axis= Direction_value[folding_solution1[i]]+1, and
lv2=0.
        End if
      new_folding_solution2[i] = direc_name[x_axis]
end for
output: new_folding_solution1 and new_folding_solution2.
end procedure
```

3.2.4 Global Vision Based Search

At first global vision based search operator, select two random conform from the population search space. From this two random conformation one individual position selected based on parameter value of global vision operator. This parameter value divided the each conformation into two blocks or part. Then the last parts of the selecting conformation are exchange directions to each other and construct two novel conformations. The interchanging of the last parts of the selecting conformation has been shown in Fig 3.7. Here, position 4 selected randomly that divided each selecting conformation into two parts. The last portion of the first conformation BLU and the last portion of the second conformation RDB are interchanged to each other and produced two new conformations. The pseudo code of the process has been given in Algorithm 3.4.

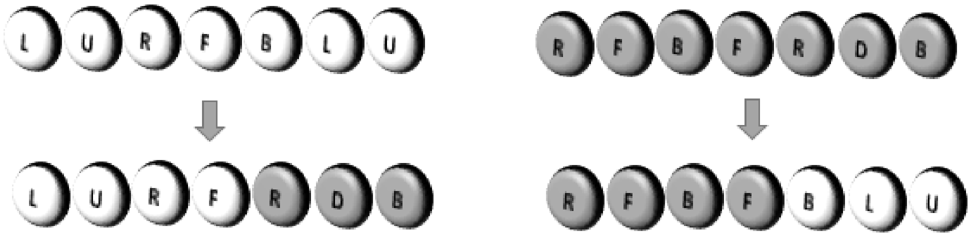

Fig 3.8. Global Vision Based Search

Algorithm 3.4. Global Vision Based Search

```
procedure globalVisionBasedSearch (hp_sequence, folding_solution1, folding_solution2)
    l = lengthOfhp_sequence
    p = Rand(0,l)
    Duplicate [l...n] values from folding_solution1 to new_folding_solution1.
    Duplicate [0...l] values from folding_solution2 to new_folding_solution1.
    Duplicate [l...n] values from folding_solution2 to new_folding_solution2.
    Duplicate [0...l] values from folding_solution1 to new_folding_solution2.
    Output: new_folding_solution1 and new_folding_solution2.
end procedure
```

3.2.5 Centroid Hydrophobic

The centroid hydrophobic operator measures the center or core of an individual conformation. Then compute the distance value of each hydrophobic (H) monomer from its core position. The center possition is calculated by the first hydrophobic and last hudrophobic amino acid position average value. It measures that whether each hydrophobic (H) amino acids of the conformation are in the center positions or not. If the hydrophobic amino acids are not in the center positions, it reform the structure in this way that the hydrophobic amino acids are remains at the center position. Similarly, it reforms the structure in that way such that the hydrophilic amino acid remains in the remote portion of the center position of the conformation. If (X_h, Y_h, Z_h) represent the core position and (X_i, Y_i, Z_i) represents i^{th}amino acid, then centroid hydrophobic is calculated as follows:

$$H_s = \frac{\sum (X_h - X_i)^2 + (Y_h - Y_i)^2 + (Z_h - Z_i)^2}{L_s} \qquad (3.1)$$

Where, L = number of amino acids in a structure

$$X_h = (x_{\max} - x_{\min})/2$$
$$Y_h = (y_{\max} - y_{\min})/2$$
$$Z_h = (z_{\max} - z_{\min})/2$$

H_s represents the distance value of hydrophobic amino acid from the core position. Determining all hydrophobic monomers distance values from the center, this mechanism try to move the hydrophobic amino acid in such a way that occupied at near of the center position of the structure. By changing the direction of folding of H monomer position , calculate the distance value, if the H_s is minimum than the previous direction then change the folding direction otherwise not changes.

Change_direction = Min($H_s[i]$)   Where, $H_s[i]$ represents all possible direction distance

Actually, this operator tries to follow the nature of a real protein folding procedure. The process of centroid hydrophobic mechanism are shown in Fig 3.8. From this example figure last monomer is hydrophobic monomer and it folded to the right direction of the previous amino acid. According to centroid hydrophobic, H monomers tend to be inner part. By changing the folding direction of the last amino acid from right direction to left direction, then the hydorphofobic amino acids are close to each other of the conformation. The energy value has also grown 3 to 4 due to this mechanism.

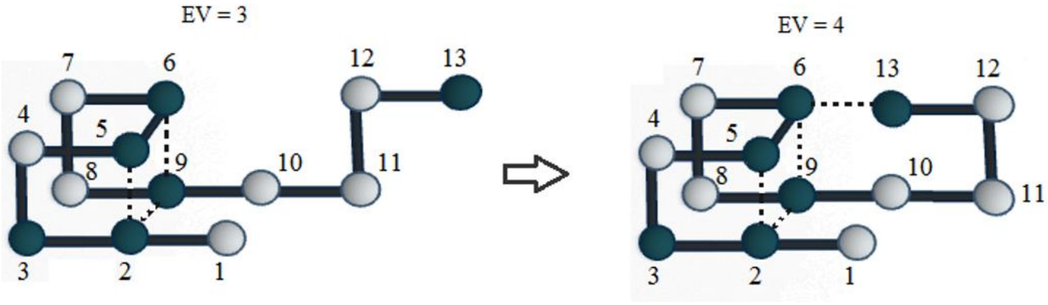

Fig 3.9. Centroid Hydrophobic mechanism

3.2.6 Moderator Mechanism

The moderator mechanism attempts to fold each individual monomer position of the last half of a protein conformation in such way that are folded in all possible directions. At first divided of the selecting conformation into two part. For each monomer in the last part of the selecting conformation, it checks the vacant and consucutive positions in the cubic lattice points. After that, try to move vacant position in the cubic lattic free position. When a monomer change its folding direction then other monomers which are associated to this monmer are remains same folding formation. After the modification process measures whether the free energy value of the newly construct conformation increases from the previous or not. If the energy value is not increased, then the cubic lattice point is not changed for the amino acid, otherwise, the position of the monomer is changed to the grown up position. By this extra mechanism, every monomer of the last part of the selecting conformations are folded with the direction which can provide improved energy value than the remaining direction.

In Fig 3.9(a) a sample conformation with thirteen monomers is given. This is a valid structure (no overlapping) with the energy value is 3. The main theme of moderator procedure is to improve the energy value of each conformation by changing the monomer directions in all directions that may be possible.

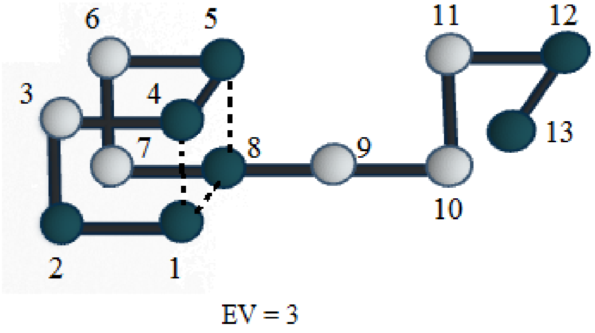

Fig 3.10(a). Before Moderator Mechanism

In this mechanism, we tries to move the monomers from the last position and gradually moves towards the middle monomer. Here the last amino acid position means 13th position monomer tries to move in every possible direction, then the energy value is not increasing. When changing all possible directions of the 13th monomer then this mechanism goes to the previous (12th) amino acid position. By this mechanism, when try to changes the folding direction of the second last monomer (12th), at the case of right direction where chosen then the energy value increase from 3 to 5 as shown in Fig 3.9(b). This direction modification process continue until reach to the middle monomer position, the mechanism tries to update the direction where maximum energy value gained for each monomer and improve the performance of the proposed algorithm.

3.2.7 Reconstruction

After performing all operation of FOA with all the extra mechanism, various invalid conformation (overlapping structure) may be constructed. Invalid conformation indicates that in one cubic lattice position more than one amino acids occupied the same position.

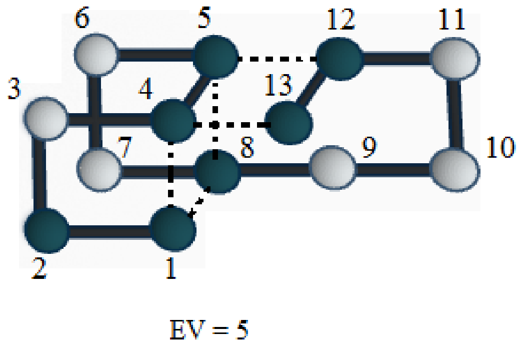

Fig 3.9(b). After Moderator Mechanism

When an inaccurate conformation is produced by different operator operation, then creating valid structure we have applied the backtracking algorithm to modify the collision point positions. In this process, check the previous monomer position where the collision occurred and tries to move that monomer position in such a way that molecule folded to another direction and produced different cubic lattice positions. If no free consecutive lattice point found where it can be placed, then change the previous amino acid folding direction and modify its position. When a valid conformation reach then stop this process otherwise continues the same process. The procedure of reconstruction mechanism are shown in Fig 3.10. The pseudo code of the process has been given in Algorithm 3.5.

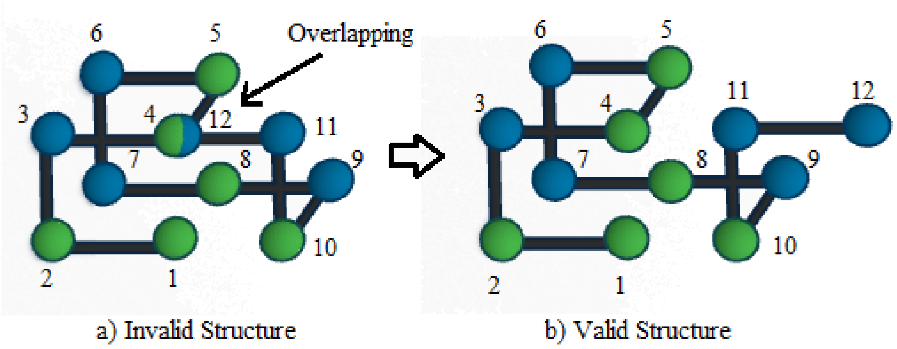

Fig 3.11. Reconstruction

Algorithm 3.5. Reconstruction

```
1: procedure Reconstruction (sequence, solution )
2: n = lengthOfSequence
3:  for i = 1 to n do
4: if overlap exists then
5:         for j=i-1 to 1 do
```

```
    if solution[j] have free lattice point then
    move solution[i] to the free lattice point
    end if
    end for
end if
end for
output: solution.
end procedure
```

CHAPTER IV

Simulation Results

4.1 Introduction

In this chapter, the simulation result of proposed FOA for solving the protein folding optimization problem (PFO_FOA) are shown. Then we compared our simulation result with existing methods and formulate the comparison results. The proposed PFO_FOA algorithm has been implemented in Java programming language. We produced our simulation result using an Intel Core i5 computer with 2.60 GHz CPU and 4 GB RAM on windows operating system (64 bit). For the implementation, programming language was Java SE Development Kit 7 platform and Netbeans IDE 7.2.1 has been used.

4.2 Data Sets

Our developed PFO_FOA algorithm has been tested on 4 identical datasets [7]. Each set contains some sequences of amino acids. There are variations in the length of the sequences. The first dataset contains 10 sequences of 48 lengths each. The second dataset also contains 10 sequences of 64 lengths. The third and fourth dataset contains sequences of different lengths. Each amino acid in the sequences is encoded with H or P according to their nature. Here, H represents hydrophobic amino acid and P represents hydrophilic amino acid. The dataset is shown in Table 4.1.

4.3 Effect of Extra Mechanisms

The effect of the extra mechanisms has been test over the first dataset of sequences. Because the set contains a large number of sequences and the length is medium. The test result of sequences S1.1 to S1.5 are shown in Fig 4.1. In this chart show comparative analysis of the best energy value of the sequences. Here show the result of FOA without applying the centroid hydrophobic and moderator mechanism, FOA with applying centroid hydrophobic, FOA with applying moderator mechanism, FOA with applying centroid hydrophobic and moderator mechanism.

Table 4.1: Dataset used in the developed algorithm

Set	Protein Sequence	Length
1	HPH2P2H4PH3P2H2P2HPH3PHPH2P2H2P3HP8H2	48
	H4PH2PH5P2HP2H2P2HP6HP2HP3HP2H2P2H3PH	48
	PHPH2PH6P2HPHP2HPH2PHPHP3HP2H2P2H2P2HPHP2HP	48
	PHPH2P2HPH3P2H2PH2P3H5P2HPH2PHPHP4HP2HPHP	48
	P2HP3HPH4P2H4PH2PH3P2HPHPHP2HP6H2PH2PH	48
	H3P3H2PHPH2PH2PH2PHP7HPHP2HP3HP2H6PH	48
	PHP4HPH3PHPH4PH2PH2P3HPHP3H3P2H2P2H2P3H	48
	PH2PH3PH4P2H3P6HPH2P2H2PHP3H2PHPHPH2P3	48
	PHPHP4HPHPHP2HPH6P2H3PHP2HPH2P2HPH3P4H	48
	PH2P6H2P3H3PHP2HPH2P2HP2HP2H2P2H7P2H2	48
2	P2H5P3H2P5H2P3HP6HPHP3HP2HP2HP5HP4H2PH2P2HP2HP	64
	P2HPHP2HP2H3PH4P2H3P4HPHP3HPHP3HPHP5HPHP2HPHP3HP2HP2	64
	HPH2P2H2PHP5H3PH4P2HP2HPH2P3HPHP2H3PH2PHP5H8P3	64
	HP2H2P2HP2HPHP2HP4HP6HPHPH3P2HPHP3HPHP2H2P2HP2HP2HPH3PH	64
	HP3H2P2HPHP3HP3HPH2P3H2PHPH2PHP2HP3HP2HPH3P2HP2HP2H3PH4	64
	HP2H2PH4P6H2P2HP4H2P3HP2HPH2PHP4H2P4HP5HP4HPH2	64
	P4HP3HP3H4PH2P5HP2HPH2PHPHP5HP10H4P4H2P2H	64
	P3H3P2HPHP2HP2H2P3HP2HP2H2PHP3HP7HPH3PH5P2H2P3HP2H	64
	HP2HP2H3P4HPHP3HPH2PH5P4HPHPHP4HPHP3H2PHP4HP2H2PHP	64
	P2HP2HP2H3P3HPHP2HP2HP6HP2H3P2HP2HP2HPHP6H3P5HPHP	64
3	HPHP2H2PHP2HPH2P2HPH	20
	H2P2HP2HP2HP2HP2HP2HP2H2	24
	P2HP2H2P4H2P4H2P4H2	25
	P3H2P2H2P5H7P2H2P4H2P2HP2	36
	P2HP2H2P2H2P5H10P6H2P2H2P2HP2H5	48
	H2PHPHPHPH4PHP3HP3HP4HP3HP3HPH4PHPHPHPH2	50
	P2H3PH8P3H10PHP3H12P4H6PH2PHP	60
	H12PHPHP2H2P2H2P2HP2H2P2H2P2HP2H2P2H2P2HPHPH12	64
4	P2H3PH3P3HPH2PH2P2HPH4PHP2H5PHPH2P2H2P	46
	PHPH3PH3P2H2PHPH2PH3PHPHPH2P2H3P2HPHP4HP2H P2H2P2HP2H	58
	P2H2P5H2P2H2PHP2HP7HP3H2PH2P6HP2HPHP2HP5H3P4H2PH2P5H2P4H4 PHP8H5P2HP2	103
	P3H3PHP4HP5H2P4H2P2H2P4HP4HP2HP2H2P3H2PHPH3P4H3P6H2P2HP2H PHP2HP7HP2H3P4HP3H5P4H2PHPHPHPH	124
	HP5HP4HPH2PH2P4HPH3P4HPHPH4P11HP2HP3HPH2P3H2P2HP2HPHPHP8 HP3H6P3H2P2H3P3H2PH5P9HP4HPHP4	136

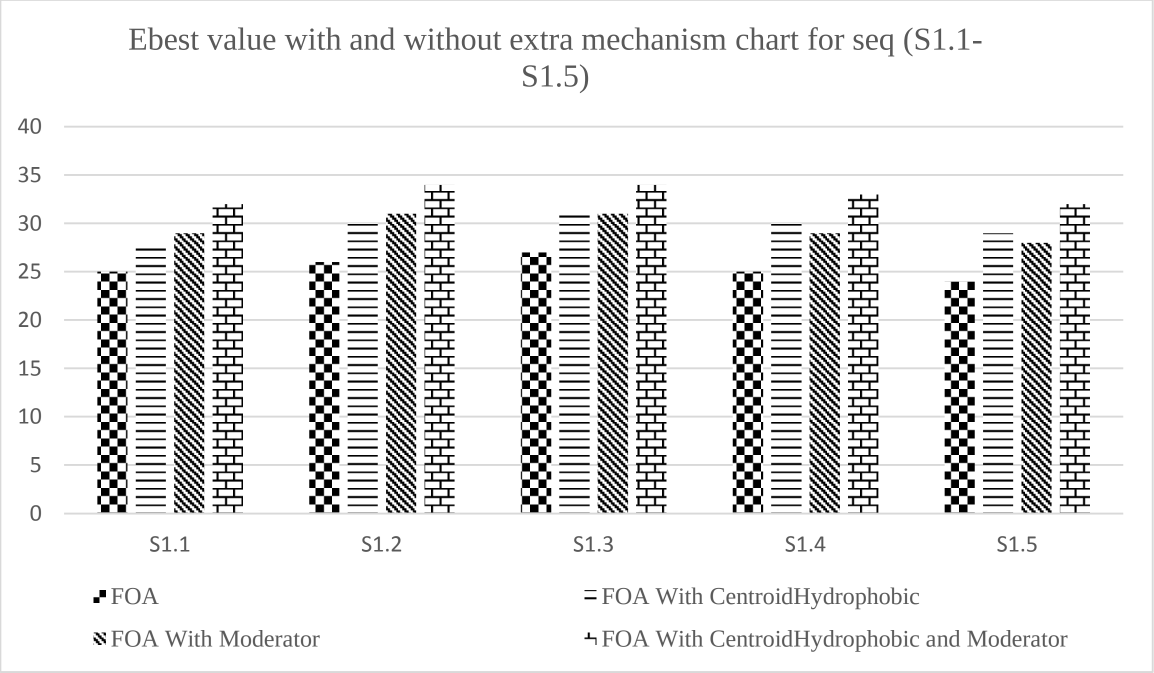

Fig 4.1. Ebest value with and without extra mechanism chart for seq. (S1.1-S1.5)

The test result of sequences S1.6 to S1.10 are shown in Fig 4.2. From the comparative result of Ebest value, it can be show that the centroid hydrophobic and moderator mechanisms increase the performance of the PFO_FOA algorithm. Because two extra mechanisms mimic the behaviors of real protein folding problem.

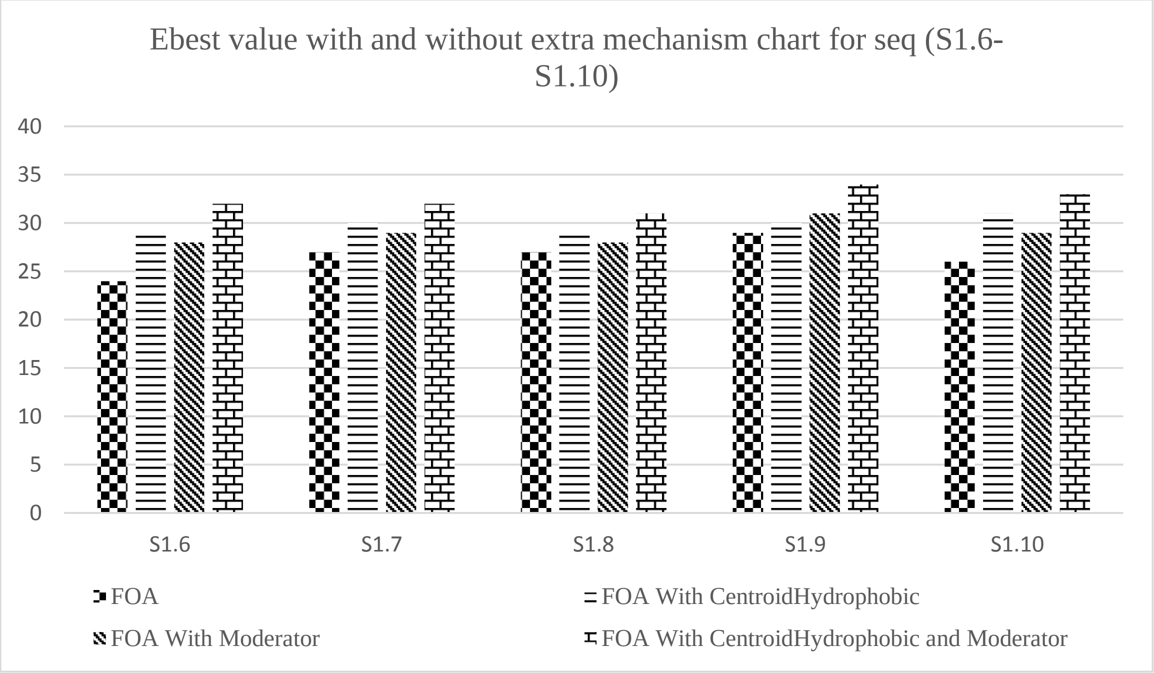

Fig 4.2. Ebest value with and without extra mechanism chart for seq. (S1.6-S1.10)

4.3 Comparison with Existing Algorithm

Genetic algorithm with advanced mechanism (GAAM) [7] is the state of art paper for the case of protein folding optimization problem. This mechanism was developed by Borko Bošković and Janez Brest in 2016. In GAAM, authors produced their result with comparing with the multiple minima genetic algorithm (GAHP) [52], memetic algorithm (MA) [53], ant colony optimization algorithm (ACO) [54], Constraint handling through multi-objective optimization (MO-FR) [55], improving genetic algorithms (HGA) [56], estimation of distribution algorithm (EDA) [57], clustered memetic algorithm with local heuristics [58]. In this paper authors showed their experimental results are better than all of the comparing methods. For that reasons, we have compared our proposed algorithm with the GAAM[7], GAHP[52], MA[53], MO-FR[55], HGA[56], EDA[57] . The results have been shown in Table 4.2, Table 4.3, Table 4.4, Table 4.5, Table 4.6, Table 4.7, Table 4.8 and Table 4.9. In the tables, the sequence number is generated by a dataset number followed by a serial number of the dataset. Such as, S1.1 represents dataset 1 and 10 number of sequences with length 48 of the dataset.

For dataset 1, we compared the best energy value of our proposed algorithm with GAAM[7], GAHP[52] and MA[53]. The comparison chart are shown in Fig 4.3.

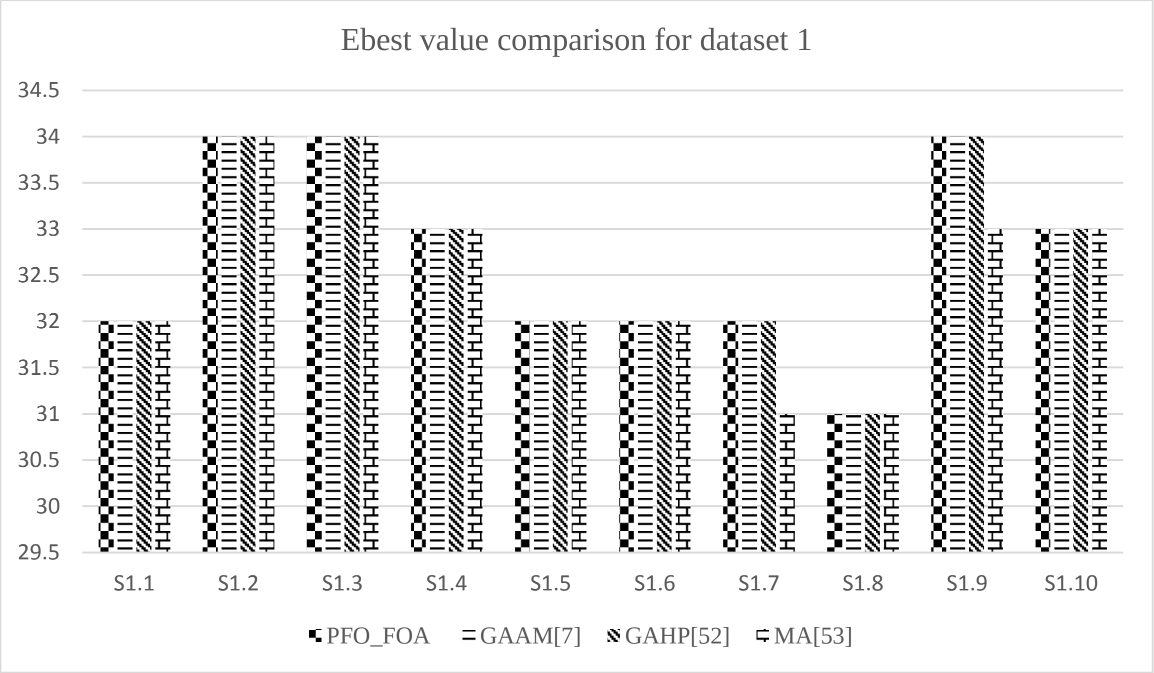

Fig 4.3. Comparison of E_{best} value of dataset 1.

We have computed the E_{best}, E_{mean} and E_{std} for all the sequences into dataset 1. For dataset 1, we compared our simulation result with GAAM[7], GAHP[52] and MA[53]. The comparison results with GAAM[7] are shown in table 4.2 and comparison with GAHP[52] and MA[53] are shown in table 4.3. The length of all the sequences in dataset 1 is 48. We make 50 independent runs on each individual sequence on dataset 1. The E_{mean} value indicates the average energy value of all the performed runs and E_{std} value represent the standard deviation of all the performed runs. Standard deviation value actually indicate the distance of each run result with the average result. From table 4.2, we can notice that PFO_FOA shows better E_{mean} values for all the sequences in dataset 1. From the comparison table it is clear that our proposed algorithm not only always found the best energy value but also produced better E_{mean} values and E_{std} values for all the sequences on the perspective of the GAAM [7].

Table 4.2. Results of dataset 1 with the number of runs was 50, compared with GAAM[7] .

Seq.	PFO_FOA			GAAM[7]		
	E_{best}	E_{mean}	E_{std}	E_{best}	E_{mean}	E_{std}
S1.1	32	32.00	0.00	32	31.82	0.38
S1.2	34	34.00	0.00	34	33.08	0.77
S1.3	34	34.00	0.00	34	33.26	0.44
S1.4	33	33.00	0.00	33	32.22	0.54
S1.5	32	32.00	0.00	32	31.58	0.49
S1.6	32	32.00	0.00	32	31.38	0.18
S1.7	32	32.00	0.00	32	30.62	0.56
S1.8	31	31.00	0.00	31	30.38	0.48
S1.9	34	34.00	0.00	34	33.02	0.37
S1.10	33	33.00	0.00	33	32.28	0.45

Table 4.3 shows the comparative result of E_{best} , E_{mean} and E_{std} value with the GAHP[52] and MA[53]. From this comparison table we can show the PFO_FOA perform better energy value for all 10 sequences of 48 length monomers.

For dataset 2, we compared the best energy value of our proposed algorithm with GAHP[52] and MA[53]. The comparison chart are shown in Fig 4.4.

Table 4.3. Results of dataset 1 with the number of runs was 50, compared with GAHP[52] and MA[53].

Seq.	PFO_FOA			GAHP[52]			MA[53]
	E_{best}	E_{mean}	E_{std}	E_{best}	E_{mean}	E_{std}	E_{best}
S1.1	32	32.00	0.00	32	30.72	0.67	32
S1.2	34	34.00	0.00	34	31.26	0.59	34
S1.3	34	34.00	0.00	34	32.08	0.80	34
S1.4	33	33.00	0.00	33	31.16	0.81	33
S1.5	32	32.00	0.00	32	30.52	0.73	32
S1.6	32	32.00	0.00	32	29.86	0.78	32
S1.7	32	32.00	0.00	32	29.82	0.56	31
S1.8	31	31.00	0.00	31	29.32	0.58	31
S1.9	34	34.00	0.00	34	31.92	0.66	33
S1.10	33	33.00	0.00	33	31.08	0.56	33

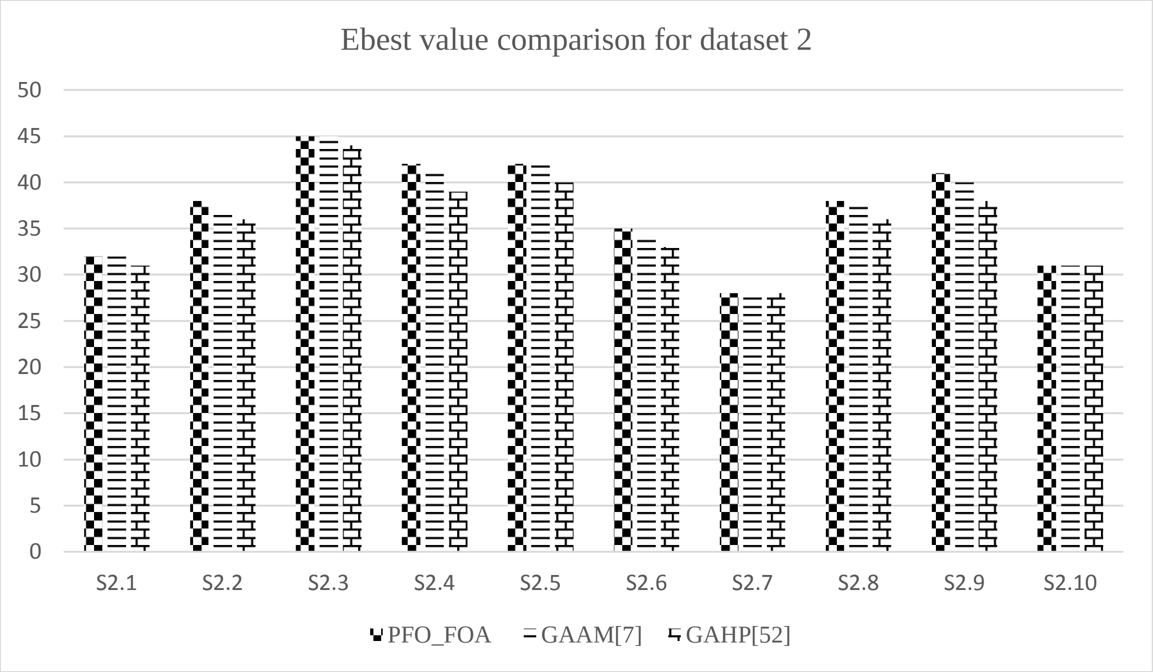

Fig 4.4. Comparison of E$_{best}$ value of dataset 2.

We have computed the E_{best}, E_{mean} and E_{std} for all the sequences into dataset 2. For dataset 2, we compared our simulation result with GAHP[52] and MA[53]. The comparison results are shown in table 4.4. The length of all the sequences in dataset 2 is 64. We make 50 independent runs on each individual sequence on dataset 2. From comparison table 4.4, it is clear that proposed algorithm make best energy value for all sequences of length 64. For some sequences (S2.2, S2.4, S2.6, S2.9), we found better E_{best} values with comparing GAAM [7] and GAHP[52]. For all sequences of dataset 2, our proposed PFO_FOA always obtained better E_{mean} and E_{std} values than both of GAAM [7] and GAHP[52].

Table 4.4. Results of dataset 2 with the number of runs was 50, compared with GAAM[7] and GAHP[52].

Seq.	**PFO_FOA**			**GAAM[7]**			**GAHP[52]**		
	E_{best}	E_{mean}	E_{std}	E_{best}	E_{mean}	E_{std}	E_{best}	E_{mean}	E_{std}
S2.1	32	32.00	0.00	32	30.86	0.60	31	28.50	1.10
S2.2	38	38.00	0.00	37	35.12	0.71	36	33.18	1.22
S2.3	45	45.00	0.00	45	43.54	0.37	44	41.88	0.87
S2.4	42	42.00	0.00	41	39.74	0.59	39	36.02	1.39
S2.5	42	42.00	0.00	42	40.62	0.72	40	37.96	1.12
S2.6	35	35.00	0.00	34	33.52	0.50	33	31.52	0.86
S2.7	28	28.00	0.00	28	28.00	0.00	28	26.70	0.70
S2.8	38	38.00	0.00	38	36.54	0.54	36	33.72	0.85
S2.9	41	41.00	0.00	40	38.00	0.60	38	36.32	0.93
S2.10	31	31.00	0.00	31	31.00	0.00	31	28.90	0.88

For dataset 3, we compared the best energy value of our proposed algorithm with GAAM [7], MO+FR[55], HGA[56], EDA[57], HGA[56] and CMA[58]. The comparison chart are shown in Fig 4.5.

We have computed the E_{best}, E_{mean} and E_{std} for all the sequences into dataset 3. For dataset 2, we compared our simulation result with GAAM [7], MO+FR[55], HGA[56], EDA[57], HGA[56] and CMA[58]. The comparison results with GAAM [7] and MO+FR[55] are shown in table 4.5, comparison with EDA[57] and HGA[56]are shown in table 4.6 and comparison with CMA[53] are shown in table 4.7. The length of all the sequences in dataset3 are 20 to 64. We make 50 independent runs on each individual sequence on dataset 3.

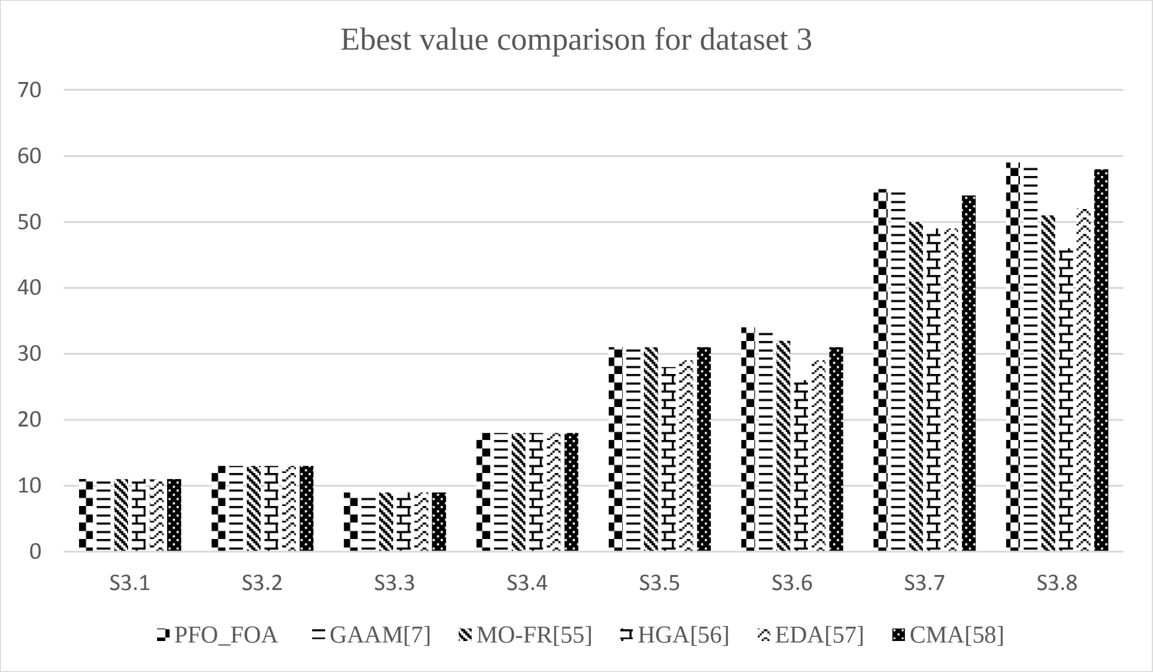

Fig 4.5. Comparison of E_{best} value of dataset 3.

From comparison table 4.5, it is clear that proposed algorithm make best energy value for all sequences of different lengths. For some sequences (S3.6, S3.7), we found better E_{best} values with comparing GAAM [7] and MO+FR[55]. For all sequences of dataset 3, our proposed PFO_FOA always obtained better E_{mean} and E_{std} values than both of GAAM [7] and MO+FR[55].

Table 4.5. Results of dataset 3 with the number of runs was 50, compared with GAAM[7] and MO+FR[55].

Seq.	PFO_FOA			GAAM[7]			MO+FR[55]	
	E_{best}	E_{mean}	E_{std}	E_{best}	E_{mean}	E_{std}	E_{best}	E_{mean}
S3.1	11	11.00	0.00	11	11.00	0.00	11	11
S3.2	13	13.00	0.00	13	13.00	0.00	13	12.96
S3.3	9	9.00	0.00	9	9.00	0.00	9	9
S3.4	18	18.00	0.00	18	18.00	0.00	18	16.84
S3.5	31	31.00	0.00	31	31.00	0.00	31	27.39
S3.6	34	34.00	0.00	34	33.96	0.14	32	27.40
S3.7	55	55.00	0.00	55	54.46	0.50	50	44.45
S3.8	59	59.00	0.00	59	59.00	0.00	51	45.63

Table 4.6 show the comparative result of E_{best} , E_{mean} and E_{std} value with the HGA[56] and EDA[57]. From this comparison table we can show the PFO_FOA perform better energy value for all 8 sequences of different length monomers.

Table 4.6. Results of dataset 3 with the number of runs was 50, compared with HGA[56] and EDA[57].

Seq.	**PFO_FOA**			**HGA[56]**			**EDA[57]**		
	E_{best}	E_{mean}	E_{std}	E_{best}	E_{mean}	E_{std}	E_{best}	E_{mean}	E_{std}
S3.1	11	11.00	0.00	11	10.52	0.54	11	10.82	0.38
S3.2	13	13.00	0.00	13	11.28	0.90	13	12.02	0.94
S3.3	9	9.00	0.00	9	8.54	0.64	9	8.96	0.19
S3.4	18	18.00	0.00	18	15.76	1.05	18	16.40	0.80
S3.5	31	31.00	0.00	28	24.60	1.57	29	27.24	0.92
S3.6	34	34.00	0.00	26	23.02	1.48	29	25.70	1.26
S3.7	55	55.00	0.00	49	41.18	2.75	49	46.30	2.04
S3.8	59	59.00	0.00	46	40.40	2.50	52	46.78	2.28

Table 4.7 show the comparative result of E_{best} , E_{mean} and E_{std} value with the CMA[58]. From this comparison table we can show the PFO_FOA perform better energy value for all 8 sequences of different length monomers. From this table, we can notify that PFO_FOA provide not only provide best energy value but also provide better E_{mean} and E_{std} value.

Table 4.7. Results of dataset 3 with the number of runs was 50, compared with HGA[56] and CMA[58].

Seq.	**PFO_FOA**			**CMA[58]**		
	E_{best}	E_{mean}	E_{std}	E_{best}	E_{mean}	E_{std}
S3.1	11	11.00	0.00	11	11.00	0.00
S3.2	13	13.00	0.00	13	13.00	0.00
S3.3	9	9.00	0.00	9	9.00	0.00
S3.4	18	18.00	0.00	18	18.00	0.00
S3.5	31	31.00	0.00	31	31.00	0.00
S3.6	34	34.00	0.00	31	31.00	0.00
S3.7	55	55.00	0.00	54	52.52	0.20

S3.8	59	59.00	0.00	58	56.30	0.35

For dataset 4, we compared the best energy value of our proposed algorithm with GAAM[7], GAHP[52] and MO-FR[55]. The comparison chart are shown in Fig 4.6.

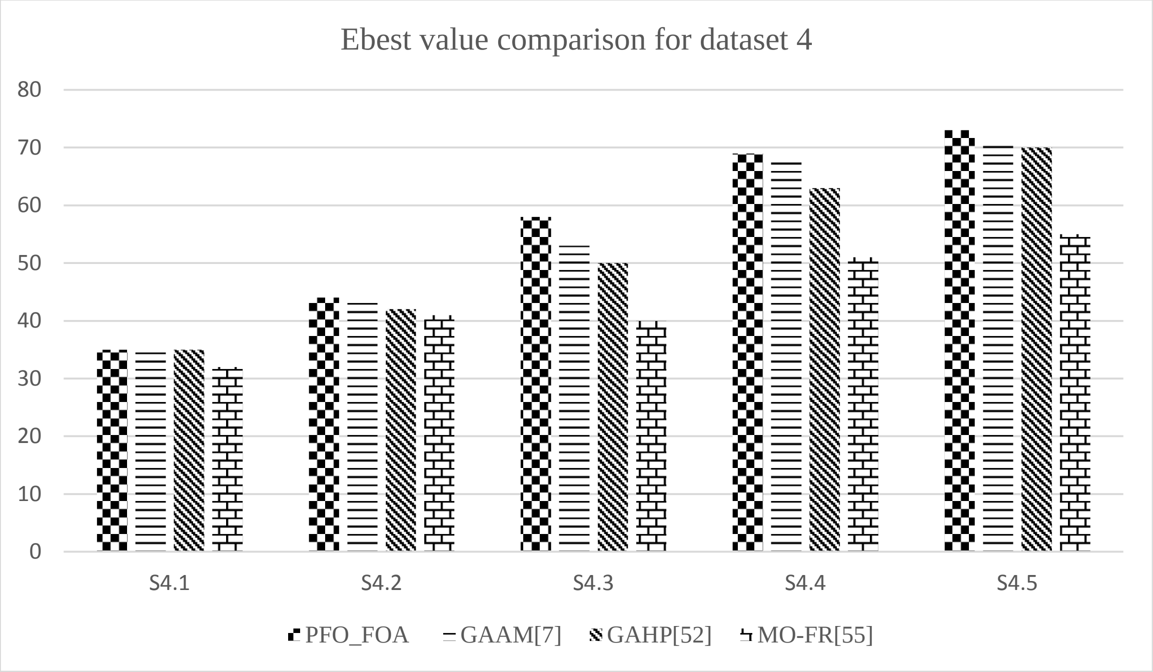

Fig 4.6. Comparison of E_{best} value of dataset 4.

Table 4.8 and 4.9 shown the E_{best}, E_{mean} and E_{std} for all the sequences into dataset 4. For dataset 4, we compared our simulation result with GAAM[7], GAHP[52] and MO-FR[55]. Dataset 4 contains different five sequences of length 46-136. We make 50 independent runs on each individual sequence on dataset 4. From comparison table 4.8, it is clear that proposed algorithm make best energy value for all sequences of long lengths. Especially when the length of the protein sequences are long PFO_FOA obtained better E_{best}, E_{mean} and E_{std} values than GAAM [7] and GAHP[52].

Table 4.8. Results of dataset 4 with the number of runs was 50, compared with GAAM[7] and GAHP[52].

Seq.	**PFO_FOA**			**GAAM[7]**			**GAHP[52]**	
	E_{best}	E_{mean}	E_{std}	E_{best}	E_{mean}	E_{std}	E_{best}	E_{mean}
S4.1	35	35.00	0.00	35	34,42	0.49	35	33.04
S4.2	44	44.00	0.00	44	41.92	0.49	42	40.04
S4.3	58	58.00	0.00	53	50.80	0.99	50	46.58

S4.4	69	66.83	0.88	68	64.78	1.25	63	58.12
S4.5	73	70.62	0.94	71	68.54	1.20	70	62.22

Table 4.9 show the comparative result of E_{best} , E_{mean} and E_{std} value with the HGA[56] and EDA[57]. From this comparison table we can show the PFO_FOA perform better energy value for all 5 long sequences of different length monomers. From this table, we can notify that PFO_FOA provide not only provide best energy value but also provide better E_{mean} and E_{std} value.

Table 4.9. Results of dataset 4 with the number of runs was 50, compared with MO-FR[55].

Seq.	**PFO_FOA**			**MO-FR[55]**	
	E_{best}	E_{mean}	E_{std}	E_{best}	E_{mean}
S4.1	35	35.00	0.00	32	28.92
S4.2	44	44.00	0.00	41	34.52
S4.3	58	58.00	0.00	40	35.35
S4.4	69	66.83	0.88	51	43.56
S4.5	73	70.62	0.94	55	46.94

From all of the comparison result tables, we can see that the simulation result of our proposed PFO_FOA is better than the GAAM [7], GAHP[52], MA[53], MO-FR[55], HGA[56], EDA[57] for especially the E_{mean} and the standard deviation value for all the datasets. Using the intensive and diversity characteristics of FOA with extra mechanisms make the proposed algorithm accurate for the protein folding optimization problem.

CHAPTER V

Conclusions and Discussions

5.1 Conclusions

The Protein folding optimization problem is the very well-known optimization problem in computation biology that accurately predict the native three dimensional structure of protein from the given primary sequence of amino acids. The most challenging task in this problem is that when the number of amino acids increases in the protein sequences then a huge range of search space are created. This exponential increase of search space make NP-hard problem of the protein folding optimization problem. Under the specified model that is called the HP model the representation of the protein folding problem is NP-hard problem. The fruit fly optimization is a recent bio-inspired algorithm that's mimic the behaviors of searching food of fly. Using the intensive and diversity characteristics of the fruit fly optimization algorithm we solved the protein folding optimization problem. The redesigned the basic operators of the fruit fly optimization algorithm with two extra mechanisms. The centroid hydrophobic and moderator mechanisms are the two extra mechanisms. The centroid hydrophobic mechanism mimic the behaviors of real protein folding and make accurate folding directions. The moderator mechanism check a fixed part of the protein sequences to every possible directions and make better energy value. This two mechanism improves the performance of the fruit fly optimization algorithm magically. Our proposed algorithm make better accuracy that all of the sequences into all the datasets. Our algorithm also includes a reconstruction mechanism that create accurate structure of protein sequences from invalid structures. Lastly, We compared our simulation results with the genetic algorithm with an advanced mechanism (GAAM) [7] which is the state of the art and from the results and also compared with GAHP[51], MA[52], MO-FR[54], HGA[55], EDA[56] , it is clear that the performance of our algorithm is better than GAAM [7], GAHP[51], MA[52], MO-FR[54], HGA[55], EDA[56].

5.2 Future Work

In future, research for the PFO problem can be done with the for larger sequences than the existing sequences. The performance of the FOA algorithm for the problem largely depends on population initialization. An efficient population initialization technique can be improved for the PFO problem. The execution time is a great factor in performance analysis. So our future work should be done on designing more efficient population initialization technique. Besides, we will try to decrease the execution time of our algorithm to increase the performance. A detailed study on parameters of FOA may provide better results and less execution time for this problem. Since there is no fixed rule for the parameters of FOA, finding the right value for the parameters is a tough task. So more experiment and study of parameters may give a better result in the case of PFO problem.

REFERENCES

1. B. Bošković and J. Brest, "Protein folding optimization using differential evolution extended with local search and component reinitialization", Information Sciences, vol. 454-455, pp. 178-199, 2018. Available: 10.1016/j.ins.2018.04.072.

2. M. Yousef, T. Abdelkader and K. El-Bahnasy, "Performance comparison of ab initio protein structure prediction methods", *Ain Shams Engineering Journal*, 2019. Available: 10.1016/j.asej.2019.03.004.

3. D. Palu, Alessandro, A. Dovier, and E. Pontelli. "Heuristics, optimizations, and parallelism for protein structure prediction in CLP (FD)", In proceedings of the 7th ACM SIGPLAN international conference on Principles and practice of declarative programming, Lisboa, Portugal, July 11-13, 2005.

4. H. Rakhshani, L. Idoumghar, J. Lepagnot and M. Brévilliers, "Speed up differential evolution for computationally expensive protein structure prediction problems", *Swarm and Evolutionary Computation*, 2019. Available: 10.1016/j.swevo.2019.01.009.

5. W. Pan, "A new Fruit Fly Optimization Algorithm: Taking the financial distress model as an example", *Knowledge-Based Systems*, vol. 26, pp. 69-74, 2012. Available: 10.1016/j.knosys.2011.07.001.

6. Iscan, Hazim, and M. Gunduz. "A survey on fruit fly optimization algorithm." 2015 11th International Conference on Signal-Image Technology & Internet-Based Systems (SITIS). IEEE,2015.Available:10.1109/SITIS.2015.55.

7. B. Bošković and J. Brest, "Genetic algorithm with advanced mechanisms applied to the protein structure prediction in a hydrophobic-polar model and cubic

lattice", *Applied Soft Computing*, vol. 45, pp. 61-70, 2016. Available: 10.1016/j.asoc.2016.04.001

8. “Protein structure – about education” Web. 23 December 2019. <http:// biology.about.com/od/molecularbiology/ss/protein-structure.htm>.

9. K. Dill, "Theory for the folding and stability of globular proteins", *Biochemistry*, vol. 24, no. 6, pp. 1501-1509, 1985. Available: 10.1021/bi00327a032.

10. B. BERGER and T. LEIGHTON, "Protein Folding in the Hydrophobic-Hydrophilic (HP) Model is NP-Complete", *Journal of Computational Biology*, vol. 5, no. 1, pp. 27-40, 1998. Available: 10.1089/cmb.1998.5.27.

11. N. Mansour, F. Kanj and H. Khachfe, "Particle swarm optimization approach for protein structure prediction in the 3D HP model", *Interdisciplinary Sciences: Computational Life Sciences*, vol. 4, no. 3, pp. 190-200, 2012. Available: 10.1007/s12539-012-0131-z.

12. Dill, Ken A. "Theory for the folding and stability of globular proteins." Biochemistry, vol. 24, no. 6, pp.1501-1509, 1985.

13. Lau, K. Fun, and Ken A. Dill. "A lattice statistical mechanics model of the conformational and sequence spaces of proteins." Macromolecules, vol. 22, no. 10, pp. 3986-3997, 1989.

14. Nemhauser, George L., and Laurence A. Wolsey. "Integer programming and combinatorial optimization." Wiley, Chichester. GL Nemhauser, MWP Savelsbergh, GS Sigismondi (1992). Constraint Classification for Mixed Integer Programming Formulations. COAL Bulletin 20, pp. 8-12, 1988.

15. Zhao, Jing, P. Song, Q. Fang, and J. Luo "Protein secondary structure prediction using dynamic programming." Acta biochimica et biophysica Sinica, vol. 37, no .3, pp. 167-172, 2005 .

16. Sabzekar, Mostafa "Protein β-sheet prediction using an efficient dynamic programming algorithm." *Computational biology and chemistry* , vol. 7, no. 0 ,pp: 142-155,2017. Available: 10.1016/j.compbiolchem.2017.08.011

17. Tuffery, Pierre, F. Guyon, and P. Derreumaux. "Improved greedy algorithm for protein structure reconstruction." Journal of computational chemistry, vol. 26, no. 5, pp. 506-513, 2005.

18. Adewumi, Aderemi O., Babatunde A. Sawyerr, and M. Montaz Ali. "A heuristic solution to the university timetabling problem." Engineering Computations, vol. 26, no.8, pp. 972-984, 2009.

19. Bacardit, J., Stout, M., Hirst, J. D., Sastry, K., Llorà, X., and Krasnogor, N. "Automated alphabet reduction method with evolutionary algorithms for protein structure prediction". In Proceedings of the 9th annual conference on Genetic and evolutionary computation, London,UK, pp. 346-353, July 07 - 11, 2007.

20. Su, Shih-Chieh, C. Lin, and C. Ting. "An effective hybrid of hill climbing and genetic algorithm for 2D triangular protein structure prediction." Proteome science, vol. 9, no. 1, 2011.

21. M. Traykov, S. Angelov, and N. Yanev, "A New Heuristic Algorithm for Protein Folding in the HP Model" *JOURNAL OF COMPUTATIONAL BIOLOGY*, vol. 23, no. 0, pp. 1-7, 2016. Available: 10.1089/cmb.2016.0015.

22. N. Yanev, M. Traykov, P. Milanov, and B. Yurukov, "Protein Folding Prediction in a Cubic Lattice in Hydrophobic-Polar Model" *JOURNAL OF COMPUTATIONAL BIOLOGY*, vol. 24, no. 0, pp. 1-10, 2016. Available: 10.1089/cmb.2016.0181.

23. Prakasam, Anandkumar, and N. Savarimuthu, "Metaheuristic algorithms and probabilistic behaviour: a comprehensive analysis of Ant Colony Optimization and its variants." Artificial Intelligence Review, vol. 45, no. 1, pp. 97-130, 2016.

24. N. Dulal Jana, J. Sil and S. Das, "Improved Bees Algorithm for Protein Structure Prediction Using AB Off-Lattice Model," Advances in Intelligent Systems and Computing 378, pp. 39-52, 2015. Available: 10.1007/978-3-319-19824-8_4.

25. L. Corr ˆ ea, B. Borguesan, C. Farf ´ an, M. Inostroza-Ponta, and M arcio Dorn, “A Memetic Algorithm for 3-D Protein Structure Prediction Problem,” *IEEE/ACM Transactions on Computational Biology and Bioinformatics,* vol. 15, no. 1, pp: 1-14, 2016. Available: 10.1109/TCBB.2016.2635143.

26. Khimasia, Mehul M., and Peter V. Coveney. "Protein structure prediction as a hard optimization problem: the genetic algorithm approach." *Molecular Simulation*, vol. 19, no. 4, pp. 205-226, 1997.

27. Unger, Ron. “The genetic algorithm approach to protein structure prediction”*Applications of Evolutionary Computation in Chemistry*, Springer Berlin Heidelberg, pp. 153-175, 2004.

28. C. Huang, X. Yang and Z. He, "Protein folding simulations of 2D HP model by the genetic algorithm based on optimal secondary structures", *Computational Biology and Chemistry*, vol. 34, no. 3, pp. 137-142, 2010. Available: 10.1016/j.compbiolchem.2010.04.002.

29. C. Lin and S. Su. "Protein 3D HP model folding simulation using a hybrid of genetic algorithm and particle swarm optimization." International Journal of Fuzzy Systems, vol. 13, no. 2, pp. 140-147, June 2011.

30. Custódio, F. Lima, H. JC Barbosa, and L. Emmanuel Dardenne. "A multiple minima genetic algorithm for protein structure prediction." Applied Soft Computing, vol. 15, pp. 88-99, 2014.

31. Xiao, Jing, L. Li, and X. Hu. "Solving lattice protein folding problems by discrete particle swarm optimization." Journal of Computers, vol 9, no. 8, pp. 1904-1913, 2014.

32. Khakzad, Hamed, Y. Karami, and S. Shahriar ARAB. "Accelerating protein structure prediction using particle swarm optimization on GPU." *BioRxiv*, 2015. Available: 10.1101/022434.

33. Shmygelska, Alena, R. Aguirre-Hernandez, and H. H. Hoos. "An ant colony optimization algorithm for the 2D HP protein folding problem". International Workshop on Ant Algorithms. Springer Berlin Heidelberg, 2002.

34. Shmygelska, Alena, and H. H. Hoos. "An improved ant colony optimisation algorithm for the 2D HP protein folding problem." Conference of the Canadian Society for Computational Studies of Intelligence. Springer Berlin Heidelberg, 2003.

35. Chu, Daniel, and A. Zomaya. "Parallel ant colony optimization for 3D protein structure prediction using the HP lattice model." In *Parallel Evolutionary Computations*, Springer, Berlin, Heidelberg, pp. 177-198, 2006. Available: 10.1007/3-540-32839-4_9.

36. N. Thilagavathi and T. Amudha. "ACO-metaheuristic for 3D-HP protein folding optimization", ARPN Journal of Engineering and Applied Sciences, 2015.

37. G. Fabre, Mario, E. Rodriguez-Tello, and G. Toscano-Pulido. "Constraint-handling through multi-objective optimization: The hydrophobic-polar model for protein structure prediction." *Computers & Operations Research* 53, pp. 128-153, 2015.

38. Pan, W.T., "A New Evolutionary Computation Approach: Fruit Fly Optimization Algorithm" *Conference of Digital Technology and Innovation Management*, Taipei, 2011.

39. N. S.Choubey, "Fruit Fly Optimization Algorithm for Travelling Salesperson Problem", *International Journal of Computer Applications*, vol. 107, no. 18, pp. 22-27, 2014. Available: 10.5120/18851-0385.

40. L. Wang, X. Zheng and S. Wang, "A novel binary fruit fly optimization algorithm for solving the multidimensional knapsack problem", *Knowledge-Based Systems*, vol. 48, pp. 17-23, 2013. Available: 10.1016/j.knosys.2013.04.003.

41. S. Li and Z. Lu, "Multi-swarm fruit fly optimization algorithm for structural damage identification", *Structural Engineering and Mechanics*, vol. 56, no. 3, pp. 409-422, 2015. Available: 10.12989/sem.2015.56.3.409.

42. H. Li, S. Guo, C. Li and J. Sun, "A hybrid annual power load forecasting model based on generalized regression neural network with fruit fly optimization algorithm", *Knowledge-Based Systems*, vol. 37, pp. 378-387, 2013. Available: 10.1016/j.knosys.2012.08.015.

43. Han, Jiuqi, P. Wang, and X. Yang. "Tuning of PID controller based on fruit fly optimization algorithm." *2012 IEEE International Conference on Mechatronics and Automation*. IEEE, pp. 409-413, 2012. Available: 10.1109/ICMA.2012.6282878.

44. S. Ding, X. Zhang and J. Yu, "Twin support vector machines based on fruit fly optimization algorithm", *International Journal of Machine Learning and Cybernetics*, vol. 7, no. 2, pp. 193-203, 2015. Available: 10.1007/s13042-015-0424-8.

45. X. L. Zheng, L. Wang, S. Y. Wang, "A novel fruit fly optimization algorithm for the semiconductor final testing scheduling problem", *Knowledge-Based Systems* 57, pp: 95-103, 2014.

46. J. Q. Li, Q. K. Pan, K. Mao, P. N. Suganthan, "Solving the steelmaking casting problem using an effective fruit fly optimization algorithm", *Knowledge-Based Systems* 72, pp: 28-36, 2014.

47. Wang, Ling, and X. Zheng. "A knowledge-guided multi-objective fruit fly optimization algorithm for the multi-skill resource constrained project scheduling problem." *Swarm and Evolutionary Computation* 38, pp: 54-63, 2018.

48. C. Fang, L. Wang, "An effective shuffled frog-leaping algorithm for resource-constrained project scheduling problem", *Computers & Operations Research*, vol.39, no. 5, pp: 890-901,2012.

49. J. Guo, A. Guo, "Crossmodal interactions between olfactory and visual learning in Drosophila", *Science* vol. 309, no.0, pp: 307-310, 2015.

50. C. L. Hwang, K. Yoon, "Multiple attribute decision making: methods and applications a state-of-the-art survey", Springer, 2012.

51. K. Deb, A. Pratap, S. Agarwal, T. Meyarivan, “A fast and elitist multi-objective genetic algorithm: NSGA-II*”, IEEE Transactions on Evolutionary Computation* , vol. 6, no.0, pp: 182-197, 2002.

52. F. L. Custodio,´ H. J. Barbosa, L. E. Dardenne, “A multiple minima genetic algorithm for protein structure prediction”, Applied Soft Computing Journal 15 (2014) 88–99. doi:10.1016/j.asoc.2013.10.029.

53. A. Bazzoli, A. G. B. Tettamanzi, “A Memetic Algorithm for Protein Structure Prediction in a 3D-Lattice HP Model,Applications of Evolutionary Computing”, Vol. 3005 of Lecture Notes in Computer Science, Springer Berlin Heidelberg, 2004, pp. 1–10. doi:10.1007/ 978-3-540-24653-4_1.

54. A. Shmygelska, H. Hoos, “An ant colony optimisation algorithm for the 2d and 3d hydrophobic polar protein folding problem”*, BMC Bioinformatics* ,vol. 6 no. 1, 2005, 30. doi:10.1186/1471-2105-6-30

55. M. Garza-Fabre, E. Rodriguez-Tello, G. Toscano-Pulido, “Constrainthandling through multi-objective optimization: The hydrophobic-polar model for protein structure prediction”, Computers & Operations Research 53 (2015) 128–153. doi:10.1016/j.cor.2014.07.010.

56. R. Konig,¨ T. Dandekar, Improving genetic algorithms for protein folding simulations by systematic crossover, Biosystems 50 (1) (1999) 17–25. doi:10.1016/S0303-2647(98)00090-2.

57. R. Santana, P. Larranaga, J. Lozano, Protein Folding in Simplified Models With Estimation of Distribution Algorithms, Evolutionary Computation, IEEE Transactions on 12 (4) (2008) 418–438. doi:10.1109/TEVC.2007.906095.

58. M. Islam, M. Chetty, Clustered Memetic Algorithm With Local Heuristics for Ab Initio Protein Structure Prediction, Evolutionary Computation, IEEE Transactions on 17 (4) (2013) 558–576. doi:10.1109/TEVC. 2012.2213258.

Published Articles

The following research article are published during this research work

1. Sajib Chatterjee and Dr. Pintu Chandra Shill, " Protein Folding Optimization in a Hydrophobic-Polar Model for Predicting Tertiary Structure Using Fruit Fly Optimization Algorithm", *2019 10th International Conference on Computing, Communication and Networking Technologies (ICCCNT-2019)*, IEEE, 2019.

.