Journal of Computer Science 10 (10): 1890-1899, 2014
ISSN: 1549-3636

doi:10.3844/jcssp.2014.1890.1899 Published Online 10 (10) 2014 (http://www.thescipub.com/jcs.toc)

A COMPARATIVE STUDY FOR RHODOPSIN PROTEIN FOLDING PROBLEM

[1]Iman Ahmed Mahmoud, [2]Amr Ahmed Badr and [3]Mostafa Abdel Azim

[1]Department of Computer Science, College of Computing and Information Technology,
Arab Academy for Science, Technology and Maritime Transport, Cairo, Egypt
[2]Department of Computer Science, Faculty of Computers and Information, Cairo University, Cairo, Egypt
[3]Department of Computer Science, College of Computing and Information Technology,
Arab Academy for Science, Technology and Maritime Transport, Cairo, Egypt

Received 2014-02-04; Revised 2014-03-13; Accepted 2014-05-06

ABSTRACT

Proteins are very important components in any living cells. A number of diseases such as Retinitis pigmentosa, Stargadt-like macular degeneration and Doyne Honeycomb Retinal Dystrophy (DHRD) diseases are shown to result from misfunctioning of proteins. Protein folding problem is a way to predict the best and optimal 3D molecular structure (tertiary structure) of a protein which is then considered to be a sign for the protein's proper functionality. This comparative study's purpose is to calculate the protein's energy using the Empirical Conformational Energy Program for Peptides (ECEPP) package and experiments were performed on the Rhodopsin proteinusing three different evolutionary algorithms in order to find the best energy in parallel with the best structure for the protein and a comparison for the results obtained from the three algorithms was performed. It was found that the best result was -11.8 obtained from the Extended Compact Genetic Algorithm (ECGA). ECGA has proved from the obtained results to be the best algorithm from the chosen algorithms in the comparative study in obtaining the Rhodpsin protein's energy and its equivalent structure.

Keywords: Extended Compact Genetic Algorithm, Rhodopsin Protein, Particle Swarm Optimization Algorithm, Memetic Algorithm and Protein Folding Problem

1. INTRODUCTION

Proteins are very important in all living algortihms' cells. They are involved in almost all cell functions. Each protein within the body has certain functionality. Some proteins such as Enzymes are responsible for facilitating biochemical reactions where they are referred to as catalysts because they speed up chemical reactions, while others such as Contractile proteins are involved in body movements, also there exist antibodies which are specialized proteins in defense against germs and foreign invaders to our body, all of these are examples for important proteins in the human body. In spite of huge and different functionalities proteins perform, all proteins are like long necklaces with differently shaped beads. Each "bead" is a small molecule called amino acids which are considered the 'building blocks' of proteins. Proteins typically contain its own sequence from 50 to 2,000 amino acids in a linear arrangement hooked end-to-end in many different combinations to end up in formation of a long chain. These acid chains do not remain straight and order (DHH, 2007). They twist and fold upon themselves and that's what we call protein folding.

Inappropriate protein folding is one way in which a protein balancemay get damaged. The mis-folded protein can be non functional totally or not optimal in its functionality that's required from this protein. There are a number of serious diseases such as Retinitis pigmentosa, Stargadt-like macular degeneration and

Corresponding Author: Iman Ahmed Mahmoud, Department of Computer Science, College of Computing and Information Technology, Arab Academy for Science, Technology and Maritime Transport, Cairo, Egypt

Doyne Honeycomb Retinal Dystrophy (DHRD) which have a common property which is they all appear to involve inappropriate folding of a particular protein (Lee and Yu, 2005).

1.1. Motivation of the Work

This study helps a lot in accelerating better drug discovery for various diseases for the retina such as night blindness and also discovering other new proteins which help the Rhodopsin protein in functioning and this may help in discovering new diseases for the retina and finding a proper cure for them.

1.2. Related Work

Many researchers before tried to find and reach the proper three dimensional structure of the protein only depending on the amino acid sequence (Calabretta *et al*., 1995; Dill *et al*., 2008). All proteins are constructed from same twenty amino acids but with different number of amino acids chained and linked together to form this unique protein and also this linked chain is unique in structure and order (Calabretta *et al*., 1995). The protein structure is formed in three levels; the first level is “the primary structure of the protein” which represents the linked linear chain of the amino acids sequence in the residue along the polypeptide chain and all other subsequent levels depend mainly on the primary structure of the protein. From the primary structure we obtain the secondary structure of the protein which is the local conformation of the almost alike amino acids residue that are close in the primary sequence and it’s built from the segments of the protein polypeptide chain. At last, the tertiary structure is formed from the secondary structure which represents the folding of the polypeptide chain which as a result represents the real final three dimensional structure of the protein. And because it is possible to change the order of the twenty amino acids which in return forms different amino acid sequence in the protein, therefore the tertiary structure (3-dimensional structure) changes as a consequence for it. And so, at the end of the folding process the protein reaches a stable 3-dimensional structure for it (Calabretta *et al*., 1995; Whitford, 2005).

Plenty of approaches have been used recently in order to predict the optimum three dimensional structure of a certain protein. Efforts to predict the molecular structure of a protein from its amino acid sequence only isn’t an easy task but has many advantages and helpful in many applications such as drug and medication industry for various diseases (Merkle *et al*., 1993; Piccolboni and Mauri, 1998).

In order to solve this problem it has been always assumed that the native conformation corresponds to the global minimum free energy state of the system. As a result for this assumption, it was important to develop efficient global energy minimization techniques. In return many techniques have been used and developed to try to solve this problem which is the protein folding problem but it has been found that the protein folding problem is a difficult optimization problem due to the non-linearity and the multi model of the energy function (Merkle *et al*., 1993; 1996).

Most techniques were performed on the penta-peptide Met-Enkephalin protein using different computational methods.

One of these computational approaches was using a Parallel Fast Messy Genetic Algorithm (PFMGA) on the Met-Enkephalin protein. Experiments were done to estimate the scalability of the PFMGA design in particular for the application of the energy minimization for the Met-Enkephalin protein. Experiments were performed using 1, 2, 4, 8, 16, 32, 64 and 128 processors. Unfortunately the conformational energies are not as low as those obtained in studies using refined energy models but are near the lowest known for the PFMGA model (Merkle *et al*., 1993).

Also, another computational method have also been used before to try to solve the protein folding problem for the Met-Enkephalin protein which was using the elitism based compact genetic algorithm with the help also of the Empirical Conformational Program for Peptides (ECEPP) package to calculate the energy for the Met-Enkephalin protein. It has been showed that ECGA reached the minimum required energy better than other techniques (Badr *et al*., 2008).

Finally the last computational method we will show here concerning the energy minimization of proteins is a technique called” Hybrid Genetic Clonal Selection Algorithm” used also to find the minimized energy for Met-Enkephalin protein. An enhancement over clonal selection algorithm was made to minimize the energy of the protein by adding the crossover function from Genetic Algorithm. Experiments performed showed that the Met-Enkephalin protein reached its minimized energy which is -20.919 using the enhanced algorithm (Mohamed *et al*., 2010).

In this research we will use the Rhodopsin protein which wasn’t investigated a lot before in many fields and below is a brief on the Rhodopsin protein.

1.3. Outline of the Work

The aim of this study is applying three different evolutionary algorithms for both predicting the Rhodpsein protein's proper 3-dimensional structure (tertiary structure) starting from its primary structure which is its linear amino acid sequence only and predicting the energy of the Rhodopsin protein.

Also the second aim is performing a comparative study between the performance and the results of applying the three evolutionary algorithms to the Rhodopsin protein.

2. MATERIALS AND METHODS

2.1. Rhodopsin Protein

We choose the Rhodopsin protein to be our target protein for the work in this research. Rhodopsin protein, the visual pigment (sometimes called the visual purple) is a biological pigment of vertebrate rod cell in the retina that is responsible for the first steps in the perception of light. It has been studied intensively for at least two decades because it is both fascinating and accessible (Sakmar, 2002).

Rhodopsin properties are very unique and fascinating which allow it to function as a visual photoreceptor. It also serves as a prototype of the largest family of membrane receptors in the human genome which is the G-protein coupled receptor family which is known for being extremely sensitive to light, enabling vision in low-light conditions.

Rhodopsin is bound to the plasma membrane of the rod behind the retina and forms transmembrane protein complexes within it. Rhodopsin undergoes a cyclic decomposition and reconstitution in response to the presence of light. This rather complicated cycle is the basis for absorption of light and its transduction into a nervous signal.

Rhodopsin is composed of two components: Scotopsin and 11-cis-retinal. When combined, these two components create the Rhodopsin moleculeas shown in **Fig. 1**.

Energy from impinging light excites the electrons in the 11-cis-retinal and converts it to 11-trans-retinal. Because 11-trans-retinal is not compatible with the scotopsin, it begins to detach from it and the Rhodopsin conjugate begins to break up into its component parts. One of the breakdown components is metarhodopsin II, which is an enzyme that affects strongly the change in the rod membrane's charge.

The disintegration of Rhodopsin into 11-cis-retinal and scotopsin is progressive, with a series of short-lived intermediate components formed, as shown **Fig. 2**. The final result is release of the two components of Rhodopsin from each other completely.

Obviously, Rhodopsin has to be regenerated, or the ability to respond to light will be completely lost in few seconds at most. This takes place through two paths. First path is when the 11-trans-retinal is re-converted to the 11-cis-retinal form by an isomerase enzyme. Since the scotopsin is present (having been removed before from the Rhodopsin), it immediately will combine with 11-cis-retinal to regenerate new Rhodopsin. Second path is when the 11-cis-retinal is generated from 11-trans-retinol, or vitamin A. Vitamin A is a derivative form of 11-trans-retinal. The isomerase reaction can convert the trans form to the cis isomer, making new 11-cis-retinal available to recombine with scotopsin. By this second pathway additional Rhodopsin is manufactured.

Animals that live in dark environments (such as deep-water fishes and cave creatures) always have far more rods, because it is important to them to be able to see in the minimum amount of light. Adaptation to dark in most animals is a matter of generating more 11-cis-retinal from vitamin A and combining it to scotopsin to make more Rhodopsin. Similarly, reduction of sensitivity to light means a reduction of the availability of Rhodopsin and hence a conversion of the 11-cis-retinal to the inactive trans-retinal form and conversion of trans-retinal back to vitamin A, making it unavailable for conjugation to scotopsin.

Anything which interferes with the Rhodopsin cycle will obviously affect vision, especially in the dark for any creature animal or human being. An individual on a diet deficient in vitamin A can have supplements so as to make him capable of producing enough 11-cis-retinal to see effectively in dim light. Drugs or chemicals that affect vitamin A metabolism may lead to problems in vision.

Since Rhodopsin protein is very important for vision, if anything goes wrong with the protein folding of Rhodopsin it will cause a lot of damage and serious diseases such as retinitis pigmentosa (night blindness).

As a result, our main concern in this research was to try to find the suitable energy to perform the binding of the angles for the protein to be folded well so as to prevent any damage.

As a result for what is needed in this research, Evolutionary Algorithms (EAs) were the first choice for the work in this research. Evolutionary Algorithms (EAs) have a very common way in general to solve application problems which are: The problem is first represented then the EA is applied on this representation to reach almost an optimum solution for this problem.

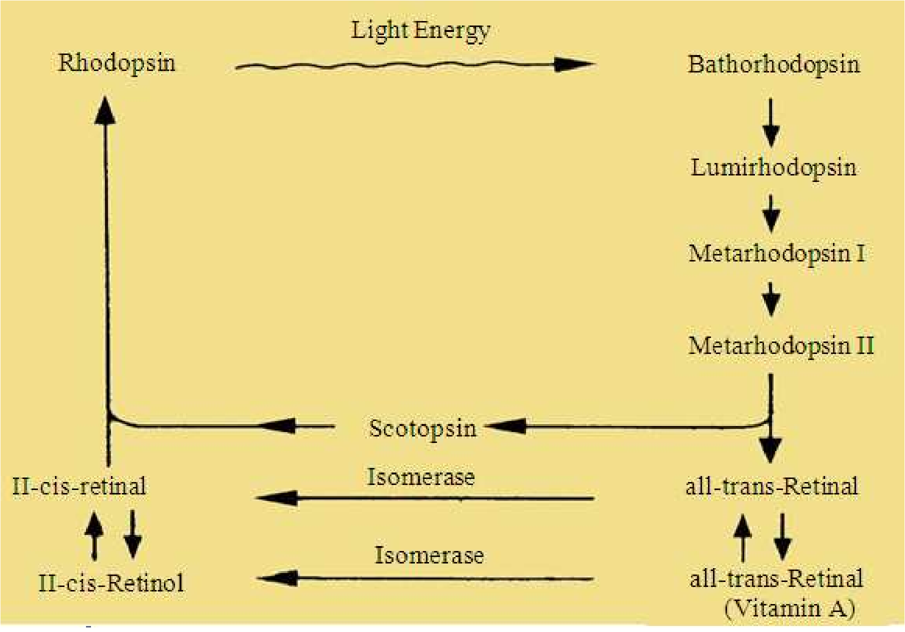

Fig. 1. Formation and Decomposition of Rhodopsin (Caceci, 1998; Radetic and Pelikan, 2010; Paul and Iba, 2002)

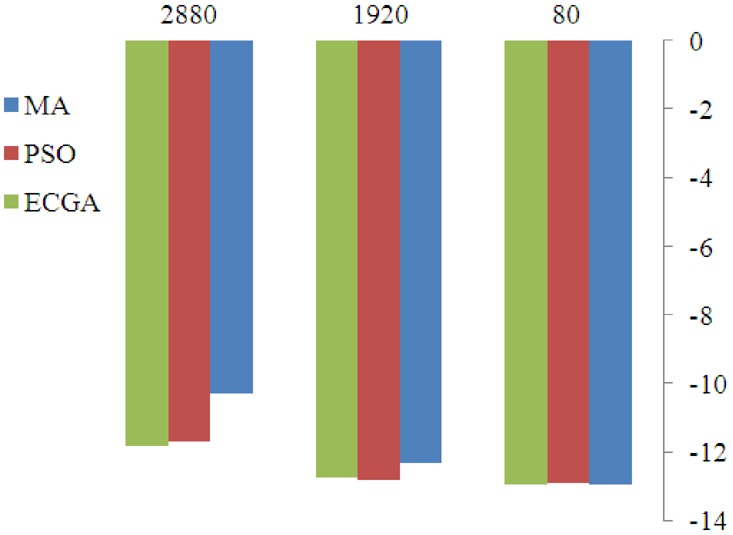

Fig. 2. Results of ECGA, PSO and MA

Below is a brief description for each of the algorithms needed to understand this comparative study (Piccolboni and Mauri, 1998).

2.2. The Algorithms

2.2.1. Genetic Algorithm

GA was developed in the early 70's in USA by J. Holland, K. DeJong, D. Goldberg. GA's were inspired and developed based on the Darwanian principle "Survival of the fittest" through biological systems. The representation of a solution in genetic algorithm is in the shape of a string called "chromosomes" which also consists of elements called "genes".

GA's pattern is to work using a random population of solutions which are called the chromosomes. Those chromosomes' fitness is evaluated with a fitness function.

As a result for this fitness, offspring chromosomes are then produced by exchanging through crossover and mutation best chromosomes information which are later evaluated to decide which good solutions will be used in the next offspring and which weak ones will be eliminated. And iteratively, generations are produced n order to obtain the best fit near to optimality result (Elbeltagi *et al*., 2005).

Pseudo code for GA

```
1. Initialize a random population of P solutions (chromosomes).
2. For each (i) belongs to P; calculate the fitness of (i).
3. For (i) = 1 to n (number of generations given):
     • Either perform Mutation or crossover
     • If crossover chosen:
         ✓ Select two parents i1 and i2 and generate their offspring from their crossover together.
     • If mutation chosen:
         ✓ Select 1 chromosome (i) randomly and generate an offspring from its mutation.
4. Calculate the fitness of the output offspring from step (3) and if the new offspring is better than the worst chromosome, then replace the worst chromosome by it.
5. Repeat step (2),(3) and (4).
6. If best fit has been obtained, then stop the algorithm.
7. If best fit hasn't been obtained, repeat steps from 2 to 5.
```

2.2.2. Memetic Algorithm

MA was developed in the early 80's by (Dawkins, 2006) when he was coining to the theory of Universal Darwinism and stating that evolution is not exclusive to biological systems only but applicable also to any system that adopts the principles of inheritance and selection. Inspired by both The Darwinian principle of natural evolution and Dawkins' notion of a meme, the term "Memetic Algorithm" (MA) was first introduced by Moscato in his technical reportin 1989 (Krasnogor *et al*., 2006). The representation of a solution in Memetic algorithm is in the shape of a string called "chromosomes" consisting of set of elements called "memes" (Elbeltagi *et al*., 2005; Krasnogor *et al*., 2006).

MA's main aspect is that all chromosomes and offsprings are allowed to gain experience through a local search before being in the evolutionary process. MA.s work is similar to that of GA.s where an initial population is generated at random. A local search is then performed on each member in the population to improve its experience and results in a local optimum solution. As in GA, crossover and mutation are performed on off springs to produce new off springs. Local Search is performed on these off springs so that local optimality is always maintained through these off springs (Elbeltagi *et al*., 2005; Krasnogor *et al*., 2006).

Pseudo code for MA

```
1. Initialize a random population of P solutions (chromosomes).
2. For each (i) belongs to P;
     • Calculate the fitness of (i)
     • Perform local search at (i)
3. For (i)=1 to n (number of generations given):
     • Either perform Mutation or crossover
     • If crossover chosen:
         ✓ Select two parents i1 and i2 and generate their offspring from their crossover together.
         ✓ Perform local search at new offspring.
     • If mutation chosen:
         ✓ Select one chromosome (i) randomly and generate an offspring from its mutation.
         ✓ Perform local search at new offspring.
4. Calculate the fitness of the output offspring from step (3) and if the new offspring is better than the worst chromosome, then replace the worst chromosome by it.
5. Repeat step (2),(3) and (4).
6. If best fit has been obtained, then stop the algorithm.
7. If best fit hasn't been obtained, repeat steps from 2 to 5.
```

2.2.3. Particle Swarm Optimization Algorithm

PSO was developed in (Eberhart and Kennedy, 1995) [95 PSO]. PSO was inspired by the social behavior of flocks of birds migrating from one place to another and how this flock with their cooperation find their own path till they reach their target (new destination) each solution is called a "particle" and refers to a bird in the flock [last]. The evolutionary process here in PSO is different than that of GA; new birds are not created from old ones through the process. Rather, the birds develop their social behavior in order to move towards their destination. As a result, the process always involves both social interaction and intelligence to learn from both local and global search (Elbeltagi *et al*., 2005).

Each bird looks in a specific direction then they communicate together and identify the bird in the best location. As a result, each bird speeds towards the best bird using a velocity that depends on its current position. Later, each bird starts a new local search from its new current position and this repeats until all the birds in the flock reach their required destination (Elbeltagi *et al*., 2005).

Pseudo code for PSO

```
1. Initialize a random population of P solutions (particles).
2. For each (i) belongs to P;
   • Calculate the fitness of (i)
3. Initialize the value of the weight factor (w).
4. For each particle (i):
   • Set the best position of the particle (i) as pBest
   • If the fitness of (i) is better than pBest, then set pBest to be equal to the fitness of the particle (i)
   • Calculate for each particle (i) velocity
   • Update for each particle (i) its own position
5. Set gBest as the best fitness between all particls' fitness.
6. Update the value of the weight factor (w).
7. If optimum solution is obtained terminate.
8. If optimum solution is not obtained, repeat steps 4,5 and 6.
```

2.2.4 Compact Genetic Algorithm

The poor behavior of simple Genetic Algorithms (sGA) in some problems has led to the development of other types of algorithms such as compact Genetic Algorithm (cGA) (Rastegar and Hariri, 2009). Compact Genetic Algorithm (cGA) is an Estimation of Distribution Algorithm (EDA). The Estimation of Distribution Algorithms (EDAs) is a class of algorithms which has been developed recently.

Pseudo code for cGA

```
1. Initialize a probability vector p[i] and all its values is equal to (0.5).
2. Generate two new chromosomes x and y.
3. Make the two chromosomes (x and y) compete together.
4. A winner and a loser chromosome will be the output from step (2).
5. Update the probability vector according to the winner chromosome from step (3).
6. If the vector has converged, then stop the algorithm.
7. If the vector didn't converge, repeat steps from 2 to 5.
```

The cGA is one of the simplest algorithms that generate offspring population according to the estimated probabilistic model of the parent population instead of using traditional recombination and mutation operators. T he main concept in this technique is to prevent the disruption of partial solutions contained in a provided solution by building a probabilistic model (Harik *et al*., 1999b). This algorithm initializes a Probability Vector (PV) and then two solutions are randomly generated by using this PV. The generated solutions are ranked based on their fitness values. Then, the PV is updated based on these solutions. This process of adaptation continues until the PV converges. The cGA represents the population as a PV over a set of solutions and imitates the order-one behavior of the simple GA (sGA) with the uniform crossover (Harik *et al*., 2006; 1999b; Rastegar and Hariri, 2009). The cGA only needs a small amount of memory; therefore, it may be quite useful inapplications which have much memory constraints (Harik *et al*., 2006).

2.2.5. Extended Compact Genetic Algorithm

The Extended Compact Genetic Algorithm (ECGA) was proposed by (Harik *et al*., 1999a; Thyago *et al*., 2008). The idea of ECGA is to solve hard problems by learning genetic linkage. ECGA is aparticular GA that uses the Marginal Product Model (MPM) to summarize important information on the population and to sample a new and may be also a better population (Thyago *et al*., 2008). Moreover, ECGA represents the joint probability distribution of genes or variables. Unlike the model used in cGA, MPMs. in ECGA can represent the probability distribution for more than one gene at a time (Sestry and Goldberg, 2000). Furthermore, ECGA adopts the Minimum Description Length (MDL) as the criterion to determine how good the learned joint probability distribution is. ECGA is considered to be reliable and accurate because of the ability of detecting building block (Hung and Chen, 2006). Harik's numerical experiments indicated also that ECGA has better performance than a simple GA does when solving hard problems (Thyago *et al*., 2008).

Pseudo code for ECGA

```
1. Initialize a population of size (N) randomly
2. For first generation, each (i) in (N);
   • Find the fitness value of each individual (chromosomes)
   • Perform a tournament selection of size (s).
   • Build a probabilistic model for the population using a greedy MPM search
   • Sample the probabilistic model generated for appearance of new individuals
```

3. If MPM model has converged, then terminate.
4. If MPM model hasn't converged, repeat all steps on step 2 until convergence of MPM model has been obtained

ECGA is an algorithm with clear features and steps as stated above in the pseudo code of the ECGA. ECGA starts by initializing a population with size (N) randomly. Afterwards, ECGA finds the fitness value of each individual in the initialized population. ECGA then starts building a probabilistic model for the population using a greedy MPM search. After the probabilistic model being built for the population, the model is being sampled for new individuals to be created. The ECGA stop criterion occurs only when the MPM model has converged. In case the MPM model hasn't converged, a new population is generated using the available MPM model and all steps being done before by ECGA are repeated until the MPM model converges.

Two features of ECGA are considered to be very clear which are the algorithm's population and selection. In ECGA, a remodeling for the population (chromosomes) occurs after each generation and that's why the structure of the models might not be very stable and so in return a concrete population is required, moreover selection of elite chromosomes can't be replaced by a simple update as what occurs in cGA (Harik *et al*., 1999b).

From all the above stated properties of ECGA, it is clear that ECGA can speed up the solutions of the problems that are partially deceptive. That's why ECGA is much more preferable than normal cGA (Harik *et al*., 1999b).

Some proteins' functionalities are important and act as enzymes which in return act as a catalyst for its chemical process which is essential for its functionality. What makes the functionality of a protein to be very accurate and highly specific is the surface pattern of each protein especially regarding its shape which is known to be very complicated, very unique and individual. This accurate surface pattern generates from the unique three dimensional structure of each protein's polypeptide chain. Therefore it's totally believed until now that any protein can be determined and understood from its amino acid chain but the problem exactly we're facing is how the information in the amino acid sequence can be easily encoded and translated to the three dimensional structure which then is mainly responsible for this protein's functionality and that's what we so call "protein folding problem" (Szilagyi *et al*., 2007).

Many computational methods have been used in order to find the minimum free energy which leads then to the stable conformation three-dimensional structure of the protein. But it was found to be very difficult to find the minimum free energy of the folded protein (Unger and Moult, 1993).

In this research the chosen algorithms for this comparison were the MA since it's better than simple GAs in their learning properties in each generation. PSO was the second chosen algorithm since it's a fast learning and intelligent algorithm. The last chosen algorithm was the ECGA which is very capable in learning through generations and obtaining their best properties and handing it out through generations to help in keeping always the best properties of the population.

2.6. The Proposed Structure Model for Protein Folding Problem

In this section we will illustrate the protein folding proposed structure model which has been used to reach the most suitable structure with equivalent minimized energy for Rhodopsin protein.

Pseudo code for the Proposed Structure Model

```
1. Obtain the energy of the given Rhodopsin protein
   using ECEPPAK.
2. Choose the required algorithm from the three given
   for minimization (optimization) of output energy
   from step 1:
   • For (i = 1 to 3)
       Choose ALG_i
         ✓ If (i = 1)
             Choose (Memetic Algorithm (MA)):
               →Parameters entered:
                 1. Population size
                 2. Number of generations
                 3. Crossover rate
                 4. Mutation rate
             •For (i = 1 to n)
               → Calculate energy from Iteration
               1(I1)
               → If (energy in I2<I1)
                    Complete calculating energy
                 Else if (I2 > I1)
                    Print out "Error"
                 Else
                 Check if energy is stable and
                 doesn't change anymore, then stop
                 and the equivalent structure
                 obtained is the optimum structure.
         ✓ Else if (i = 2)
```

```
            Choose (Particle Swarm Optimization
            Technique (PSO)):
                →Parameters entered:
                  1.   Population size
                  2.   Number of generations
                  3.   V max
                  4.   W
          •For (i=1 to n)
              → Calculate energy from Iteration
              1(I1)
              → If (energy in I2<I1)
                      Complete calculating energy
                 Else if (I2 > I1)
                        Print out "Error"
                 Else
                 Check if energy is stable and
                 doesn't change anymore, then stop
                 and the equivalent structure
                 obtained is the optimum structure.
    Else if (i = 3)
      ✓ choose (Extended Compact Genetic
        Algorithm (ECGA)):
        →Parameters entered:
          1.  Population size
          2.  Chromosome length
          3.  Seed
          4.  Cross over probability
          5.  Tournament size
      •    For (i = 1 to n)
             → Calculate energy from Iteration 1(I1)
             → If (energy in I2<I1)
Complete calculating energy
               Else if (I2 > I1)
                 Print out "Error"
               Else
                 Check if energy is stable and doesn't
                 change anymore, then stop and the
                 equivalent structure obtained is the
                 optimum structure.
                 Else Print error.
  3.   Repeat steps 2 and 3 for other non-chosen
       algorithms to realize the best algorithm for
       minimization of the Rhodopsin protein energy.
```

First of all we had to start to evaluate the energy on the Rhodopsin protein and that was done by the energy evaluator package ECEPPAK. The ECEPPAK allows us to study the energy of the required structure of the protein. So, the ECEPPAK helped us eventually to find out the individual energy.

Table 1. Results of ECGA, PSO and MA

Alg./population size	80	192	288
MA	-12.96	-12.32	-10.29
PSO	-12.93	-12.85	-11.72
ECGA	-12.97	-12.76	-11.82

The next step shows how to find suitable techniques to minimize the energy evaluated as much as possible until the optimum structure of the Rhodopsin protein is reached and that was performed using three different evolutionary algorithms which are: Memetic Algorithm (MA), Particle Swarm Optimization Algorithm (PSO) and Extended Compact Genetic Algorithm (ECGA) which were implemented using Microsoft Visual C++. A special dll file called "alleg40" was used in order to perform the drawing of the protein in the coding partition which allows the connection between the energy evaluation file obtained from the ECEPPAK and the ECGA coding to perform the final drawing and structure for the optimum structure of the Rhodopsin protein. This special alleg40.dll file is used mostly for graphics in games. The result of trying to reach the optimum structure has been drawn step by step until the optimum structure with minimum energy reached using the Ramachandran Plot Explorer which is designed to make it easy to examine the conformation of a polypeptide (**Table 1**).

The three algorithms proceed in clear progress towards the optimal interacting angles 3D structure, by generating individuals conforming to higher fitness probability distribution in a clear GUI which allows the user to choose which algorithm to choose to perform on the chosen protein.

3. RESULTS

The best energy result which was equal to -11.8 was obtained from the Extended Compact Genetic Algorithm (ECGA).

The second best energy result which was equal to -11.72 was obtained from the Particle Swarm Optimization algorithm (PSO).

The least energy result which was equal to -10.29 was obtained from the Memetic Algorithm (MA).

4. DISCUSSION

The main purpose in this research was to find the structure in conjunction with the minimum energy that is compatible with the reached optimum structure of the protein, the fitness of the individuals is calculated in terms of energy. And of course in our research the best algorithm used for the energy and three-dimensional structure is the algorithm which was able to obtain the minimum energy which is almost near to that of the lab result which is -12.

Afterwards a comparative study was performed between the results and outputs of the three algorithms on the same protein.

The best result obtained as mentioned in results section was from ECGA because ECGA has two very important properties which made this result obtained from it compared to the other two algorithms which is the "linkage learning" and "MPM models".

ECGA's linkage learning property helps in learning properties of generations through transferring those properties of gene building blocks crossover which are linked and related together by a certain property required and preferred to be available in the coming generations. As a result, the whole solution will be swapped to divided sub problems instead of single genes which will help a lot to work on individually to obtain better solution output.

ECGA's using MPM model as a structure helps in translating this structure easily to a linkage map with the partition used. As a result, defining exactly which genes in the generation should be tightly linked together through crossover in order to preserve their wanted properties through other coming generations.

The near result to that of ECGA occurred from the PSO since this algorithm involves both social interaction and intelligence so that individuals learn from their own experience (local search) and also from the experience of others around them (global search).

Also, this near result is because PSO is known for its simplicity, convenience, fast convergence and fewer parameters.

In spite the good properties of PSO, but also it has a very bad disadvantage which getting easily trapped in the local optimum which makes this algorithm a second choice after ECGA until a solution could be found and applied to solve this point.

From the reasons that was the cause of the least energy obtained by MA was since MA algorithm has the same aspect as Simple Genetic Algorithms (SGA) in addition to a local experience on each population member to improve its experience so that local optimality only is always maintained through off springs. Moreover, MA crossover operation is for each gene by its own which in return results that the fitness value was lessened for each chromosome unlike building blocks in linkage learning property of ECGA.

5. CONCLUSION

In this study, the molecular required structure of Rhodopsin protein which states the global minimum free energy state of the system has been obtained by three algorithms which was the best of them is the ECGA.

The measurement of the probability distribution quality is always done according to the Minimum Description Length principle (MDL). Since the base of ECGA is the probability distribution as mentioned before, so in return the MDL concept prevents in accuracy and complexity and as a result probability distributions of high quality are provided.

Moreover, the probabilistic model obtained from each generation makes ECGA reflects the problem structure more and so in return better performance is being achieved through exploiting and exploring the relationship between the genes due to balancing both exploitation and exploration for a high quality result.

In further research, other algorithms may be used or enhancements of ECGA also can be used (such as IECGA) or other evolutionary algorithms (such as Ant colony and Shuffled frog leaping algorithms) over Rhodopsin protein's energy so as to compare the results with the results from this study and find out better optimization for the protein folding problem of the Rhodopsin protein if exists.

Also in proposed future work, a hyberdized model for PSO can be used to overcome the problem of local optimum fast convergence in it and compare the results.

6. REFERENCES

Badr, A., I.M. Aref, B.M. Hussein and Y. Eman, 2008. Solving protein folding problem using elitism-based compact genetic algorithm. J. Comput. Sci., 4: 525-529.

Caceci, T., 1998. Anatomy and physiology of the eye. The Vertebrate Eye.

Calabretta, R., S. Nolfi and D. Parisi, 1995. An artificial life model for predicting the tertiary structure of unknown proteins that emulates the folding process. Adv. Artificial Life, 929: 862-875. DOI: 10.1007/3-540-59496-5_349c

Dawkins, R., 2006. The Selfish Gene: 30th Anniversary. 1st Edn., Oxford University Press, Oxford, ISBN-10: 0191574066, pp: 384.

DHH, 2007. The Structures of Life. Department of Health and Human, United States of America.

Dill, K.A., S.B. Ozkan, M.S. Shell and T.R. Weikl, 2008. The protein folding problem. Ann. Rev. Biophys., 37: 289-316. PMID: 18573083

Eberhart, R. and J. Kennedy, 1995. A new optimizer using particle swarm theory. Proceedings of the 6th International Symposium on Micro Machine and Human Science, Oct. 4-6, IEEE Xplore Press, Nagoya, pp: 39-43. DOI: 10.1109/MHS.1995.494215

Elbeltagi, E., T. Hegazy and D. Grierson, 2005. Comparison among five evolutionary-based optimization algorithms. Adv. Eng. Inform., 19: 43-53. DOI: 10.1016/j.aei.2005.01.004

Harik, G., D.E. Goldberg, E.C. Paz and B.L. Miller, 1999a. The gambler's ruin problem, genetic algorithms and the sizing of populations. Evolout. Comput., 7: 231-253. DOI: 10.1162/evco.1999.7.3.231

Harik, G.R., F.G. Lobo and D.E. Goldberg, 1999b. The Compact genetic algorithm. IEEE Trans. Evolut. Comput., 3: 287-297. DOI: 10.1109/4235.797971

Harik, G.R., F.G. Lobo and K. Sastry, 2006. Linkage learning via probabilistic modeling in the Extended Compact Genetic Algorithm (ECGA). Stud. Comput. Intell., 33: 39-61. DOI: 10.1007/978-3-540-34954-9_3

Hung, P.C. and Y.P. Chen, 2006. IECGA: Integer extended compact genetic algorithm. Proceedings of the 8th Annual Conference on Genetic and Evolutionary Computation, Jul. 08-12, Seattle, WA, USA, pp: 1415-1416. DOI: 10.1145/1143997.1144222

Krasnogor, N., A. Aragon and J. Pacheco, 2006. Memtic Algorithms. In: Metaheuristic Procedures for Training Neural Networks, Alba, E. and R. Martí (Eds.), Springer, New York, ISBN-10: 0387334157, pp: 225-248.

Lee, C. and M.H. Yu, 2005. Protein folding and diseases. J. Biochem. Molecular Biol., 38: 275-280. PMID: 15943901

Merkle, L.D., G.H. Gates and G.B. Lamont, 1993. Application of the parallel fast messy genetic algorithm to the protein folding problem. Proceedings of the International Supercomputer Users Group Conference, (UGC' 93), pp: 189-195.

Merkle, L.D., R.L. Gaulke and G.B. Lamont, 1996. Hybrid genetic algorithms for polypeptide energy minimization. Proceedings of the ACM Symposium on Applied Computing, Feb. 17-19, ACM., Philadelphia, PA, USA, pp: 305-311. DOI: 10.1145/331119.331198

Mohamed, A.O., A.A. Hegazy and A. Badr, 2010. Solving protein folding problem using hybrid genetic clonal selection algorithm. Int. J. Comput. Sci. Netw. Security, 10: 94-98.

Paul, T.K. and H. Iba, 2002. Linear and combinatorial optimizations by estimation of distributed algorithms. Evolut. Computat.

Piccolboni, A. and G. Mauri, 1998. Application of evolutionary algorithms to protein folding prediction. Artificial Evolut., 1363: 123-135. DOI: 10.1007/BFb0026595

Radetic, E. and M. Pelikan, 2010. Spurious dependencies and EDA scalability. Proceedings of the 12th Annual Conference on Genetic and Evolutionary Computation, Jul. 07-11, Portland, OR, USA, pp: 303-310. DOI: 10.1145/1830483.1830543

Rastegar, R. and A. Hariri, 2009. A step forward in studying the compact genetic algorithm. Evolutionary Comput., 14: 287-299. DOI: 10.1162/evco.2006.14.3.277

Sakmar, T.P., 2002. Structure of rhodopsin and the superfamily of seven-helical receptors: The same and not the same. Curr. Opin. Cell Biol., 14: 189-195. DOI: 10.1016/S0955-0674(02)00306-X

Sestry, K. and D.F. Goldberg, 2000. On extended compact genetic algorithm. Evolutionary Computat.,

Szilagyi, A., J. Cardos, S. Osvath and L. Barna, 2007. Protein Folding. In: Neural Protein Metabolism and Function, Lajtha, A. and N. Banik (Eds.), Berlin Heidel Berg, pp: 303-343.

Thyago, S.P.C.D., D.E. Goldberg and K. Sastry, 2008. Improving the efficiency of the extended compact genetic algorithm. Proceedings of the 10th Annual Conference on Genetic and Evolutionary Computation, Jul. 12-16, Atlanta, GA, USA, pp: 467-468. DOI: 10.1145/1389095.1389181

Unger, R. and J. Moult, 1993. Genetic algorithms for protein folding simulations. J. Molecular Biol., 231: 75-81. PMID: 8496967

Whitford, D., 2005. The Three-Dimensional Structure of Proteins. In: Protein Structure and Function, Whitford, D. (Ed.), Wiley and Sons, England, pp: 39-83.