© International Journal of Combinatorial Optimization Problems and Informatics, Vol. 9, No. 3, Sep-Dec 2018, pp. 41-54. ISSN: 2007-1558.

Parallel Evolutionary Multi-Quenching Annealing for Protein Folding Problem

Anylu Melo Vega[1], Juan Frausto-Solís[1], Ernesto Liñán García[2], Guadalupe Castilla Valdez[1], Juan Javier González Barbosa[1], David Terán Villanueva[1], and Juan Paulo Sánchez Hernández [3]

[1]Tecnologico Nacional de México- Instituto Tecnológico de Cd. Madero, México,
[2]Universidad Autónoma de Coahuila, México,
[3] Universidad Politécnica del Estado de Morelos, México.

anylumelovega@gmail.com, juan.frausto@itcm.edu.mx, ernesto_linan_garcia@uadec.edu.mx,gpe_cas@yahoo.com.mx, jjgonzalezbarbosa@hotmail.com, david_teran01@yahoo.com.mx, juan.paulosh@upemor.edu.mx

Abstract. The Protein Folding Problem (PFP) consists in determining the functional three-dimensional structure or Native Structure (NS) of a protein, which normally has the lowest Gibbs energy. In this paper, a new hybrid Parallel Evolutionary Multi-Quenching Annealing Algorithm (PEMQA) is proposed to obtain high-quality solutions for the target proteins. PEMQA generates an initial population of solutions using a Genetic Algorithm (GA). Furthermore, a Multi-Quenching Algorithm (MQA) is executed in an independent core using each of these Genetic Algorithm (GA) solutions. A master process determines which MQA delivers the best solution. PEMQA uses shared memory parallel programming and is implemented in SMMP (Simple Molecular Mechanics for Proteins). The incorporation of evolutionary processes in a PEMQA algorithm allows an improvement in MQA capacity of exploration. Results obtained with PEMQA outperform most of those achieved by the classic SA reported in current state of the art literature.

Key words: Multi-quenching Annealing, Parallel Genetic Algorithms, Protein Folding Problem, Simulated Annealing.

1. Introduction

Proteins are complex molecular machines involved in almost all cellular functions of the living organisms. Each protein is made of a specific sequence of amino acids, which folds into certain tertiary three dimensional structure named Native Structure (NS). Protein Folding Problem (PFP) has been a scientific challenge during the last 50 years and is one of the most difficult problems to solve [1]. It consists in determining the native structure of a protein given its amino acids sequence, which is the primary structure [2]. In computational molecular biology and biochemical physics, PFP is one of the most challenging problems when modeling amino acid sequences for synthetic protein design [3]. It has remarkable interactions in successful biotechnological applications such as the design of peptides, proteins, enzymes and biocatalysts.[4]. In computational molecular biology and particularly in computer simulations it is common the use of small peptides as scale models in discovering and understanding the properties of larger amino acids systems. Additionally, the peptides are used as benchmarks to evaluate algorithms[5].

The natural process of protein folding is done in an extremely fast way inside a cell and it is not yet exactly known how nature carries this process. A functional protein has a specific structure, which is obtained from a minimal energy configuration or one very close to the optimal one, in order to perform their biological function. This structure is known as the Native Structure (NS) [6]. PFP is an NP-complete optimization problem [7]; thus, designing an efficient algorithm to solve this problem is a significant computational challenge. This is because in PFP there is a very large number of possible conformations or structures, even for the small proteins known as peptides. For instance the Met-Enkephalin peptide, is expected to contain 10^{11} local minima [8]. Moreover, extremely large execution times for some PFP instances can be required, sometimes as long as 150 CPU days [9]. For this reason Parallel Computing (PC) is commonly used [10],[11],[12],[13],[14] with the intention of taking advantage of utilizing a greater number of computer cores in order to reduce the execution time.

Received Dec 26, 2016 / Accepted Dec 11, 2017

SA algorithms are among the most successful in solving PFP, particularly in some cases of small peptides they improved some energy values for the conformational parameters [2]. Classic SA algorithms have an excellent ability to explore the search space while avoiding being trapped in local optimums to achieve good performance for some NP-Complete problems, especially those involved in large-scale problems that contain many local optima [5],[7],[15]. Nevertheless, the main disadvantage of SA is that during the search process, only one solution around a neighborhood is evaluated at each iteration. This feature unfortunately impacts the quality and execution time of the algorithm [16]. On the other hand, evolutionary algorithms (EAs) [17] have the characteristic of exploring globally the solution space, considering a collection of solutions in each iteration; thus, they can simultaneously explore several regions of the solution space. In addition, evolutionary operations, such as crossover and mutation, share information between individuals of each population, generation by generation, preserving the best characteristics of the population. Nevertheless, EAs in general are slow versus local search algorithms [18]. Premature convergence is another disadvantage of EAs [19]. These problems can be avoided both by hybridization with algorithms that include strategies to escape from local optima [20] and parallel computing design techniques. For these reasons, some hybridization of EAs and SAs have been proposed for other problems in the literature [21], [22], [23].

Among the algorithms that have been most successful in the field of PFP are the Multi-Quenching Annealing (MQA) algorithm [24] and Chaotic Multi-Quenching Annealing algorithm[24], [25]. According to the revised literature, a hybridization of MQA with EAs have not been published for PFP. This paper addresses this gap in the literature and develops a new algorithm based on SAs and EAs, incorporating evolutionary features to MQA to produce the hybrid sequential algorithm. And finally the parallel algorithm for peptides is developed and experimentally assessed.

More specifically, a new Parallel Evolutionary Multi-Quenching Annealing (PEMQA) for PFP applied to peptides is proposed. PEMQA is not a simple parallelization of MQA algorithm but an extension of it by adding evolutionary features. Hence, it is a hybridization of MQA and evolutionary search strategies. The sequential version of this hybrid algorithm is SEMQA (Sequential Evolutionary Multi-Quenching Annealing), and the parallelized variant of the sequential is PEMQA. Given that the execution times of PFP instances are extremely large; thus, an efficient hybrid algorithm for PFP requires to be parallelized [14] and the parameters of both search processes must be conveniently tuned.

This paper is organized as follows: in section 2, the Protein Folding Problem is described; section 3 presents the new algorithm Parallel Evolutionary Multi-Quenching Annealing (PEMQA); in section 4 the results of the experimental work with PEMQA are showed; and finally, in section 5, the conclusions are presented and discussed.

2. Protein folding problem

This section provides the basic concepts of the problem being addressed.

2.1 Amino acid sequence

An amino acid is a type of organic acid which contains an acid functional group and an amine functional group on adjacent carbon atoms. Amino acids are the building blocks of proteins. As it is shown in Figure 1, every alpha amino acid has a carbon atom, Cα, bonded to: a carboxylic acid group, –COOH; an amino group, –NH_2; a hydrogen atom, H; and an R group which is unique for every amino acid [6].

2.2 Dihedral angles as problem variables

The protein atoms are in a 3D form in Cartesian Coordinates. There are four types of torsion angles: Phi (φ), Psi (ψ), Omega (ω) and Chi (χ) where:

- φ is the dihedral angle between the amino group and the alpha carbon.
- ψ is the dihedral angle between the alpha carbon and the carboxyl group.
- ω is the angle between two amino acids in a sequence.
- χ is the angle of the side chains

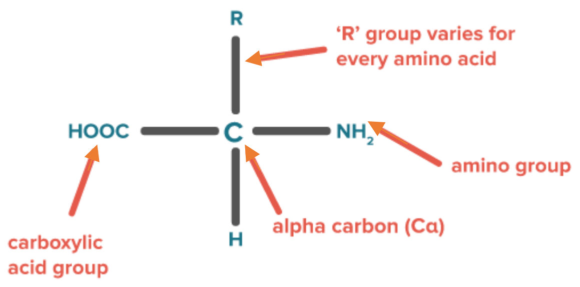

Figure 1 *Amino acid structure*

Phi, Psi, and Omega angles are represented in Figure 3. The values for these angles denote the problem's variables; thus, in all the instances of PFP the variables are discrete values in a range of [-180°, 180°] degrees [26].

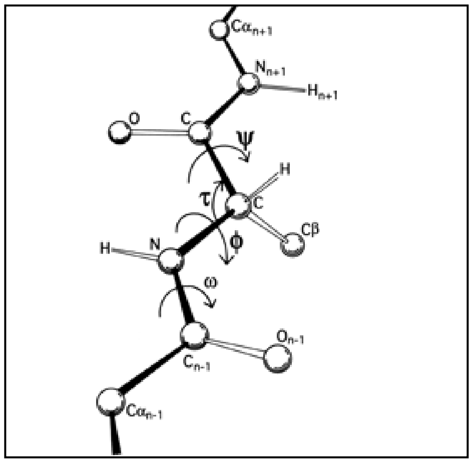

Figure 3 *Dihedral angles*

The values for φ and ψ angles can be validated using the Ramachandran Plot (RP) [26], that is shown in Figure 4. This graph represents the restrictions for the angles, and is used to verify their feasibility. RP provides the ranges for φ and ψ; the family of RP plots delimits regions in which the proposed angles correspond to well-folded proteins. Consequently, the proposed angles for any amino acids sequence are compared with these RPs to evaluate whether the structure of this amino acids sequence is valid or not [26].

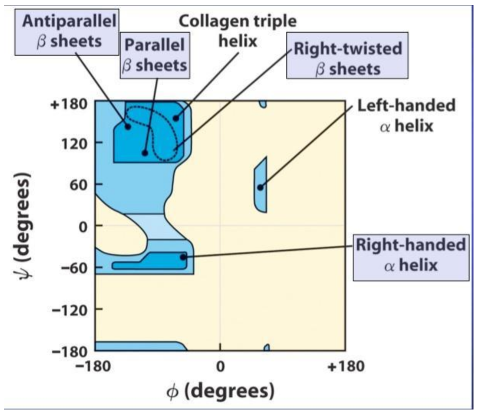

Figure 4 *Ramachandran Plot*

2.2 Protein Folding Problem definition

The PFP aim is in finding the tertiary 3D structure of a protein or minimal energy structure. The only input information is the primary amino acid sequence, this is known as the Ab initio approach [7]. The problem formulation is based on the energy function proposed by Eisenmenger [27], which uses the ECEPP force field [28] associated to the internal potential energy of the proteins. ECEPP/2 field is calculated in Equation 1 by summation of the Lenard Jones energy, the electrostatic energy, the hydrogen bonds energy and the torsion energy. The first three correspond to the spatial energy and each one is calculated as the sum of the pairwise interaction energies between atoms. Alternatively, the torsion energy is obtained as the sum of torsion angles for each bond.

$$E_{Tot(\Phi)} = E_{LJ} + E_{el} + E_{hb} + E_{tor} \tag{1}$$

The PFP formulation is given as follows:

1. A sequence of *n* amino acids $\alpha_1, \alpha_2, \alpha_3, \ldots, \alpha_n$, which represents the primary structure of the protein is given.
2. An energy function $f(\varphi_1, \varphi_2, \ldots, \varphi_m)$, which represents the free energy is given.

The PFP solution consists in finding the native structure in which the protein has the total minimum energy value. Therefore, the problem is to minimize the energy given by the following equation (2):

$$\min E(\Phi) = \sum_{j>i}\left(\frac{A_{ij}}{r^{12}{}_{ij}} - \frac{B_{ij}}{r^{10}{}_{ij}}\right) + 332\sum_{j>i}\frac{q_i q_j}{\varepsilon r_{ij}} + \sum_{j>i}\left(\frac{C_{ij}}{r^{12}{}_{ij}} - \frac{D_{ij}}{r^{6}{}_{ij}}\right) + \sum_{n} U_n(1 \pm \cos(k_n \Phi_n)) \tag{2}$$

Where:

- $\Phi = (\varphi_1, \varphi_2, \ldots, \varphi_m)$, ϵ Ra^3 (Ramamchandran space) [26] is a vector containing the protein torsion angles.
- r_{ij} is the distance between the atoms *i* and *j* given in Å.
- $\frac{A_{ij}}{r^{12}{}_{ij}}$ *and* $\frac{B_{ij}}{r^{10}{}_{ij}}$, are the attraction and repulsion LJ potentials produced by pairs of atoms *i* and *j*. The attraction potential is directly proportional to the LJ empirical potential (*A*), and inversely proportional to the distance between them, to power 12. The repulsion potential is directly proportional to the LJ empirical potential (*B*), and inversely proportional to the distance between them, to power 10.
- C_{ij}, and D_{ij} are the parameters of the empirical potentials for the hydrogen bond Energy.
- q_i and q_j are the partial charges acting on the atoms *i* and *j*, respectively.
- ε is the dielectric constant which is usually set to $\varepsilon = 2$, its value in the inner space of the protein [27].
- 332 is a correction factor for using the energy units expressed in kcal/mol [27].
- U_n is the energetic torsion around the bond *n*.
- k_n is the multiplicity of the torsion angle φ_n.

In summary; PFP consists in finding the native structure by finding the values:

- $f^*(\varphi 1, \varphi_2, \ldots, \varphi_n)$ that is the minimum energy value, and
- $\Phi^* = (\varphi_1, \varphi_2, \ldots, \varphi_n)$ that is the optimal configuration.

3. Hybrid SA Evolutionary Algorithm Proposed

In this section, we present the heuristic methods and techniques required to develop PEMQA.

3.1 Simulated Annealing

The Classic Simulated Annealing (SA) algorithm emulates the process of tempering alloys metals by heating until its melting point, holding the temperature and afterwards performing a very slow cooling, in order to produce perfect crystalline forms in its molecular structure for improving the metal physical properties [16]. The algorithm is tuned to a high temperature value T_0 and as the process advances, the temperature gradually is decreased until reaching a final temperature T_f very close to zero. The algorithm uses cooling functions to decrease the temperature, the cooling is commonly applied using a lineal geometrical function

$T_{k+1} = \alpha T_k$. Alpha must be tuned as it highly depends on the nature of the problem being addressed. In the SA literature for the PFP problem alpha values are recommended in the range of $0.70 \leq \alpha < 1.0$ [16]. The initial and final temperature values, as well as the parameter α, were experimentally set. The classic simulated annealing algorithm has a Metropolis Cycle, which is a method that produces neighbor solutions based on the current solution and accepts or reject these solutions according to the Boltzmann probability distribution [29].

3.2 Multi-Quenching Annealing

MQA [24] is an SA algorithm with two well defined phases: the Quenching phase (QP) and the Annealing phase (AP). MQA is shown in Algorithm 1. QP implements the search process with extremely high temperatures. Then, an abrupt drop in temperature is applied to continue the search process. AP is a classic annealing phase where the search is performed from high to low temperatures. The annealing phase is applied (line 11) as usual beginning from a high temperature $T = Tthreshold$ and decreasing until the freezing temperature T_f. However, MQA has an additional quenching phase which applies an annealing algorithm with a fast cooling scheme. In MQA the algorithm searches for a solution where the energy variations stabilize until reaching a dynamic equilibrium. MQA finishes when the final temperature is reached [24]. In this annealing process, the initial temperature value T_0 and final temperature T_f are analytically tuned (line 1) and the number of iterations of each phase is determined by an analytical model [24]. MQA applies a QP phase (line 12) when $T >$ Tthreshold and applies the equations $T_{k+1} = \alpha \tau_k T_k$, with $\tau_k = \tau^2_{k-1}$. In QP, α values range from $0.70 \leq \alpha < 0.8$ and τ_k is lower than 1; because $\tau_k = \tau^2_{k-1}$ a new quenching phase is started when τ reaches an ε value close to zero. When Tthreshold is reached, the annealing phase is triggered and the temperature is gradually decreased by $T_{k+1} = \alpha T_k$; and α takes values in the range $0.9 \leq \alpha < 1$ [24].

Algorithm 1. MultiQuenching Process (MQA)

```
1.  $T_k = T_i$ = Analytical tunning ( )  // initial temperature ; $T_f$ = Analytical tunning ( )  //final temperature,
2.  $T_{treshold}$ = Set_value ( ); α _value = initial_alpha_value, $\tau$ = initial_tau_value
3.  $S_i$= Initial_solution( primary_sequence), $E_i$ = Energy ($S_i$),$S_{min}$= $Si$ , $Emin$ = Energy ($S_{min}$)
4.  While ($T_k > T_f$) do
5.  convergence = false
6.      While (convergence = false) do     //Metropolis_cycle; $0< \tau_k <1$, $0.7<\alpha <1$
7.      $Sk$ = Perturbation ($Si$); $E_k$ = Energy ($S_k$); $Delta\_E = Ek - Ei$
8.      If ($Delta\_E \leq 0$) then  $Si = Sj$
9.      else if ($e^{Delta\_E/Tk}$) > random [0,1] then $Si=Sj$
10.     If ($Si < Smin$); save (Smin); save ($Emin$ = Energy ($Smin$))
11.     If ($T \leq T_{treshold}$) then $T_{k+1} = \alpha T_k$    //Apply_annealing_phase( );
12.     else If ($\tau \leq \varepsilon$ ) then $\tau_k = \tau^2_{k-1}$; $T_{k+1} = \alpha (1- \tau_k) T_k$ endif //Apply_quenching_phase( )
13.     ConvergenceTest(convergence)
14.     end_while
15. end_do
```

3.3 Genetic Algorithm

Genetic Algorithms (GA) [17] are a class of evolutionary algorithms that emulate the evolution process of living organisms. According to evolution theory, individuals not well adapted to the environment are in general extinguished. Moreover, the most adapted individuals commonly survive and can be selected for reproduction. In this theory, a new improved population replaces the previous one. As this process is ruled by natural selection and transference of genes, the new population is usually improved and with superior genes to its preceding generation [30].

GAs are applied to different optimization problems where the objective function is unknown or where this function is known but it is difficult or impossible to derive. In GAs the problem variables are coded as an individual's genes, and an aptitude or fitness function is established. Thus, each individual is a candidate solution with a degree of utility established by its fitness.

Algorithm 2 shows GA process, which evolves through *n* iterations a set of chromosomes that coded the amino acids chains (line 4). The genetic processes are: tournament selection (line 6), SBX crossover and uniform mutation (line 7). Selection imitates the survival of members from one generation to the next. The crossover is an operator that simulates the reproduction of living beings and exchanges the information of the chromosomes between individuals. In another hand, the mutation simulates the alterations that occur in nature when new individuals are reproduced, mutating some genes with a probability that allows to diversify the population. [30]. When the evolutionary process ends, final population is obtained and the best individual is selected (line 11).

Algorithm 2. Genetic Process (GA)

```
1.  //Chromosome code is created from peptide structure
2.  Generate ($P$(0))
3.  t:=0
4.  while not Termination_criterion($P$(t)) do
5.      Evaluate ($P$(t)) // Energy value
6.      $P$'(t)    = Selection($P(t)$)
7.      $P$'(t)    = Apply_Reproduction_Ops($P$'(t))
8.      $P$(t + 1)    = Replace ($P$(t), $P$'(t))
9.      t = t +1
10. end while
11. BestSolution= FindBest($P$(t))//Finding the best individual
12. return BestSolution
```

3.4 PEMQA parallelization model

PFP requires an intensive computational level given the huge number of interactions between different atoms in the amino acids chain, leading to exponential growth when the length of the protein chain increases; thus, parallel computing is a programming technique that has driven the advancement in the field of computational biology [1], [10],[11],[12],[13],[14].

The proposed parallelization model does not seek to accelerate the search to find conformations of lower energy, because PFP is characterized by having a large number of local minima in low energy conformations [8], so this approach would inevitably increase the probability of being trapped in local optima. In contrast the designed parallelization model rather performs a broad exploration, using the AG for this exploratory process. Once the convergence criterion of the GA is fulfilled, a set of MQA processes are applied in parallel processing threads to clones of the best solution, functioning in this way as perturbation processes in order to intensify the search on conformations of Low power. The framework of the parallelization model applied to the proposed hybrid parallel algorithm is shown on Figure 5.

This model was implemented using OpenMP; an API for shared parallel memory that uses a master thread, with a team of slave processors running parallel code. When executing multiple processes the communication between the processors is only used at the end of the execution to generate statistics [14].

3.5 Sequential Evolutionary MultiQuenching Annealing (SEMQA)

Traditionally MQA and GA begin their process from a set of random solutions [17], [24]. Both are designed to obtain better solutions along the algorithm process. At the end of the process they achieve the optimal or close to the optimal solution. However, in practice, the achieved solution could be far from the global optimum, this is because there is no guarantee that the algorithm does not get trapped in a local optimum. The hybrid algorithm SEMQA (Sequential Evolutionary MultiQuenching Algorithm) incorporates the diversifying Boltzmann machine to avoid a premature stagnation.

SEMQA is a new algorithm that combines the MQA and GA algorithms, and its behavior is describe in the following lines:

1. SEMQA reads the amino acid sequence a_1, a_2,…, a_n. and transforms this information in a genetic codification of a protein.
2. A population of solutions is created, each gene of this solutions represents an angle that a protein can take.

3. The population evolves applying the genetic operators: selection, crossover and mutation, generating one hundred new generations. With this evolution SEMQA searches in a global search space.
4. At end of the last generation, SEMQA transforms the structure of the solution in order to start a new process. A MQA is then started with the best solutions found so far. For the moment, the main purpose in this process is to search in a local space area with good results.
5. Finally when SEMQA reaches its final temperature, the best solution is obtained.

SEMQA keeps the best solution obtained during the executions; however, the general process follows the traditional GA and SA approaches as can be seen in Algorithm 3. However, SEMQA is not executed because it was designed as a first development phase. In the following section we present the parallelization of SEMQA.

Algorithm 3. SEMQA Sequential Evolutionary Multi-Quenching Annealing

```
1.  BestEnergy = infinite
2.  GlobalBest = null
3.  Setting = ( currtem, alfa, tao, beta)
4.  Configs [][] = init_population ( amino_seq )
5.  For ( i=1 to 100 )
6.      Selection ( Configs )
7.      Crossover ( Config1, Config2 )
8.      Mutate ( Configs )
9.      iterBest = GetBest ( Configs )
10.     If  ( energy ( iterbest ) < BestEnergy ) then
11.            GlobalBest = iterBest
12.            BestEnergy = energy(iterBest)
13.     End if
14. End for
15. angles = getAngles( GlobalBest )
16.      While ( $Temperature_k$ > criteria_stop ) do
17.          Metropolis_cycle(angles, currtem, alfa, tao, beta)
18.          $Temperature_{k+1} = \alpha (1- \tau) temperature_k$
19.          $\tau = \tau^2$
20.          If $\tau > 0$ then
21.              Apply_quenching phase(angles, currtem, alfa ,tao, beta)
22.              Else
23.                    Apply_annealing_phase(angles, currtem, alfa, tao, beta)
24.          End if
25.     End do
26. Return OptimalConfigurationAngles( )
```

3.6 Parallel Evolutionary Multi-Quenching Annealing (PEMQA)

In this work a new Parallel Evolutionary Multi-Quenching Algorithm (PEMQA) for PFP is proposed. This algorithm is hybridized through strategies from MQA and GA which operate in an integrated and synergistic approach in the parallel process. In PEMQA these approaches do not work independently, rather they cooperate along the PEMQA iterations until the convergence condition is fulfilled. The components of PEMQA are shown in Figure 5 and are described as follows:

1. The algorithm takes the amino acid sequence of the protein <a_1, a_2,.., a_n> which represents the primary structure; this sequence is the only information required in the Ab-initio method.

2. A set of *m* random solutions are generated with the purpose of constructing the first population; each solution is a chain of angles that meets the Ramachandran restrictions.
3. For each solution, the fitness value is obtained using the ECEPP/2 energy function.
4. The population of solutions evolves through selection, crossover and mutation operations until the convergence criterion is reached by the final population. The *GA-Best Solution* in the final population is selected to evolve in the MQA processes.
5. Clones of the *GA-Best solution* evolves in parallel in *n* MQA processes using different random seeds until the stochastic equilibrium is reached. In Section 3, the Algorithm 1 describes the MQA process
6. Finally, the *Best-Solution* of the *n* MQA processes is chosen. This is the global solution achieved by the PEMQA algorithm.

Algorithm 4 shows PEMQA procedure and Figure 5 the framework.

Algorithm 4. PEMQA Parallel Evolutionary Multi-Quenching Annealing

```
27. BestEnergy = infinite
28. GlobalBest = null
29. Setting = ( currtem, alfa, tao, beta)
30. Call OMP_set_num_threads(n);
31. Configs [][] = init_population ( amino_seq )
32. For ( i=1 to 100 )
33.      Selection ( Configs )
34.     Crossover ( Config1, Config2 )
35.     Mutate ( Configs )
36.     iterBest = GetBest ( Configs )
37.      If  ( energy ( iterbest ) < BestEnergy ) then
38.          GlobalBest = iterBest
39.     BestEnergy = energy(iterBest)
40.     End if
41. End for
42. TID = OMP_GET_THREAD_NUM()
43. If (TID = 0) then
44.       NTHREADS = OMP_GET_NUM_THREADS()
45. End if
46.     angles = getAngles( GlobalBest )
47.     While ( $Temperature_k$ > criteria_stop ) do
48.         Metropolis_cycle(angles, currtem, alfa, tao, beta)
49.         $Temperature_{k+1} = \alpha (1- \tau) temperature_k$
50.         $\tau = \tau^2$
51.         If $\tau > 0$ then
52.             Apply_quenching phase(angles, currtem, alfa ,tao, beta)
53.             Else
54.                 Apply_annealing_phase(angles, currtem, alfa, tao, beta)
55.         End if
56.      End do
57. END PARALLEL
58. Return OptimalConfigurationAngles( )
```

4. Experimentation

To assess PEMQA performance, three peptides; the Met^5-Enkephalin, Proinsulin^{31} and C-Peptide^{13} which are PFP benchmark instances were used. The instances were obtained from the protein data bank (PDB) [31]. Met Enkephalin, also called opioid growth factor (OGF) has a sequence of 5 amino acids and 19 variables. This peptide has been widely used for computational algorithms assessment [32]. The best-known energy for this instance is -10.72 kcal/mol [33]. In another hand, the connecting peptide, or C-peptide has 13 amino acids and 68 variables and best known energy of -101.94 kcal/mol [25]. Proinsulin is a prohormone precursor to insulin made in the beta cells of the islets of Langerhans. The Proinsulin instance has 31 amino acids and 132 variables and its best-known energy is -162.56 kcal/mol [25]. PEMQA is a parallel implementation in SMMP [27], which calculates the energy of a protein using a simplified model of the protein that maintain fixed the bond angles and only allows changes in the dihedral angles. The ECEPP/2 energy field was used to evaluate the energy of the resulting structure.

Experimentation was carried out in the high-performance cluster Ehécatl of the Instituto Tecnológico de Ciudad Madero, with the following characteristics: Intel® Xeon® processor at 2.30 GHz, Memory 64GB (4x16GB) ddr4-2133 and Linux CentOS operating system. Each one of the test instances was executed 30 times, and the best, average and worst solutions were obtained and are shown in Table 1.

A comparison with CMQA [25] was performed and the results are shown in Table 2. These results correspond to the common instances used by both algorithms. It should be mentioned that CMQA is used as a reference to assess the performance of the algorithm because it is among the best SA algorithms for the PFP used instances. The best, the average and the execution time for PEMQA and CMQA are shown. CMQA is based on a hybrid Multi-Quenching Annealing algorithm that uses chaotic functions to increase the exploration potential of the PFP solutions space. PEMQA achieved the best solution for the studied instances, and it is worth mentioning that this algorithm gets the best solution in much less time. Nevertheless, CMQA exhibits better performance if we consider the solutions average energy. This fact could be explained by the increase of exploration produced by the CMQA chaotic functions [25]. However, the main objective of solving PFP is to obtain the lowest energy solution.

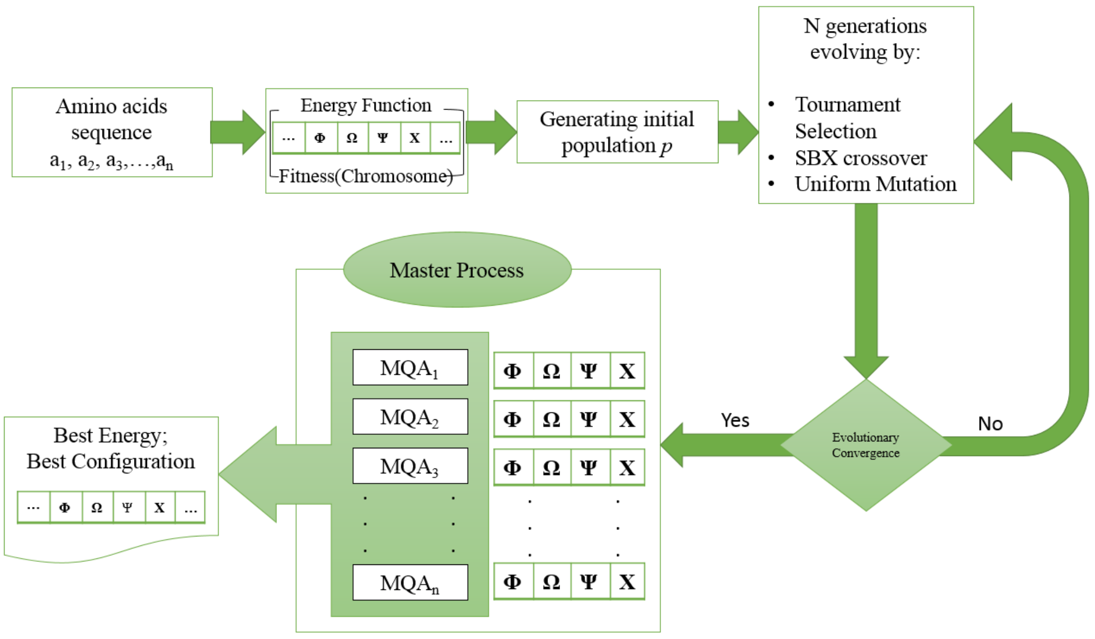

Figure 5 *PEMQA framework*

Table 1 *Results achieved by the PEMQA algorithm*

Test Instances			Best Solution	Average Solution	Worst Solution
Protein	#Amino acids/ #variables	α			
Met-Enkephalin	5/19	.85	-8.5513	-4.5890	-2.9115
C-Peptide	13/68	.70	-74.1826	-63.9787	-60.3583
Proinsulin	31/132	.95	-165.5225	-102.6336	-70.9187

When a solution of PFP is found for a given protein, not only the minimum energy value but also the optimum configuration of the protein should be measured. The optimal configuration of a peptide has the minimum deviation compared to the NS. To know how well the quality of the proposed solution is, the average distance RMSD between atoms of the given peptide and NS is used. When RMSD is close to zero the proposed solution has a structure aligned with the known native structure. However, because all the atoms of the protein are weighted by the calculation in the same way. RMSD becomes a very sensitive calculation.

Table 2 *Comparison of PEMQA and CMQA results*

Protein	PEMQA			CMQA		
	Best Solution (Kcal/mol)	Average Solution (Kcal/mol)	Time (minutes)	Best Solution (Kcal/mol)	Average Solution (Kcal/mol	Time (minutes)
Met-Enkephalin	**-8.5513**	-4.5890	**0.5311**	-7.1804	**-5.4390**	1.2335
Proinsulin	**-127.1729**	-102.6336	**17.6052**	-126.9481	**-122.8490**	42.6216

For this reason, another metric named TM-Score (TM) is used. TM also measures the protein structure similarity between two peptides. In practice, one of them is the NS or the best-known structure and the other is the peptide under study. Proteins with TM> 0.5 are considered similar to the NS or best topology. Hence, these values are an indicator of a good solution [34]. As an example, the quality of the solution of PEMQA executed for Met-Enkephalin is shown in Table 3. Notice that for larger α parameters values, not only the energy values obtained by the algorithm are better but also the mentioned quality metrics indicate that the structure is better aligned.

Table 3 *Energy, TM- Score and RMSD values for Met Enkephalin*

Α	Energy value of PEMQA	TM-Score	RMSD
0.70	-5.32	0.50109	0.65
0.75	**-8.37**	**0.59189**	**0.06**
0.80	-6.92	0.42097	1.03
0.85	**-8.55**	**0.51690**	**0.20**
0.90	-7.35	0.56351	0.33
0.95	**-7.62**	**0.59416**	**0.05**

Figure 6 shows a comparison between the native structure of Met-Enkephalin and the structure obtained by PEMQA. The quality of the solution is measured by using not only the obtained energy, but also using the RMSD measure and TM-Score [34].

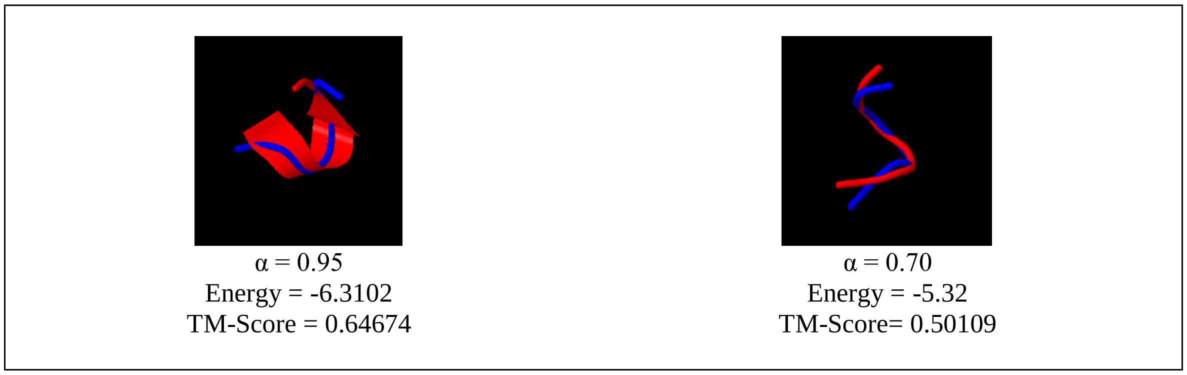

Figure 6 *TM-align simulation for Met-Enkephalin*

The results obtained for the three performance indicators with three different α values for PEMQA algorithm are shown in Table 4. These results are compared with those reported for the state-of-the-art algorithms which have obtained the lowest energy values for the studied instances: *Multiphase Simulated Annealing Based on Boltzmann and Bose-Einstein Distribution* (MPSABBE) [28]; This work uses four different annealing phases depending on the process temperature. Which incorporates the Bose-Einstein distribution for modeling the transition between the gaseous and liquid state. The second work is a hybrid algorithm called *Chaotic*

Multiquenching Annealing (CMQA) [24]. CMQA applies the MuliQuenching strategy and chaotic functions. It is worth mentioning that these algorithms do not incorporate parallelization strategies.

Table 4 contains the results for the TMScore and RMSD metrics, the average energy and the value of the alpha parameter that configures the annealing processes. In the first column are the names of the peptides studied, the second contains the different configurations of the alpha parameter evaluated, the following three columns correspond to the TMScore, RMSD and Average Energy values for the PEMQA algorithm. The same data for MPSABBE and CMQA are presented in the last six columns.

Table 4 *Comparison of PEMQA, MPSABBE and CMQA results*.

Protein	Configuration	PEMQA			MPSABBE [29]			CMQA [25]		
	$\alpha_{Annealing}$	TMScore	RMSD	Energy (Avg)	TMScore	RMSD	Energy (Avg.)	TMScore	RMSD	Energy (Avg) cal/mol
Met-Enkephalin	0.7	0.50	0.65	**-5.32**	---	---	---	---	---	---
	0.75	**0.59**	**0.06**	**-8.37**	---	0.45	-3.08	---	---	-3.28
	0.95	**0.59**	**0.06**	**-7.62**	---	0.36	-5.06	---	---	-5.07
Proinsulin	0.7	0.183	3.38	**-127.17**	---	---	---	---	---	---
	0.75	0.2117	3.38	**-122.89**	---	3.13	-94.25	---	---	-93.99
	0.95	0.2284	**2.56**	**-165.52**	---	3.12	-122.43	---	---	-120.76

For RMSD, a value close to 0 indicates the best alignment to the NS and for TMScore a value greater than 5 indicates major alignment. For the Met-Enkephalin PEMQA obtained the lowest energy value with α = 0.75 and α = 0.95. In another hand for these α's the RMSD and TMScore values are 0.06 and 0.59 respectively, which indicates that the configuration of lower energy also presents the best alignment to the native structure. MPSABBE and CMQA do not report TMScore information for the peptides under study (a dashed line is used for this cases), but the RMSD indicates that it is far from the native structure.

Met Enkephalin is a peptide used as a test for classical simulated annealing implementations because is difficult to improve the best solution found in the literature when a classic SA variant is applied. The best average solution obtained with the classical SA for Met Enkephalin was -5.0634 kcal/mol [8]. This solution was overtaken recently by MPSABBE[29], where the best average solution obtained was -7.2787 kcal/mol. MPSABBE algorithm modified the classical SA using not only the Boltzmann distribution for accepting solutions, but also uses the Bosse-Einstein distribution for very low temperatures. It is important to note that our best average solution for this peptide was -8.55 kcal/mol.

Conclusions

In this work, a new algorithm PEMQA for Protein Folding Problem for peptides is proposed. This algorithm obtains competitive results in terms of performance and computational time when searching for the minimum energy value as the comparison with CMQA points out. It also obtains good results when measuring RMSD and TM-Score metrics, exhibiting important proximity to the Native Structure. And demonstrates its capacity to refine homolog models; which is an open problem in the proteins science [35]. However, the energy average values obtained indicate the need to improve the algorithm, which is why the need for future work aimed at tuning and improving the current model, as well as the search for new hybrid parallel models. In general, for small proteins, PEMQA has better execution time than CMQA with a good performance and it also shows a good performance.

Acknowledgements

The authors would like to acknowledge to CONACYT, TECNM and PRODEP. A special acknowledgement to the Laboratorio Nacional de Tecnologías de la Información del Instituto Tecnológico de Ciudad Madero for the access to the cluster. This work has been partially supported by CONACYT Project 254498

References

[1] K. A. Dill and J. L. MacCallum, "The Protein-Folding Problem, 50 Years On," *Science (80-.).*, vol. 338, no. 6110, pp. 1042–1046, 2012.
[2] L. B. Morales, R. Garduño-Juárez, and D. Romero, "Applications of Simulated

Annealing to the Multiple-Minima Problem in Small Peptides," *J. Biomol. Struct. Dyn.*, vol. 8, no. 4, pp. 721–735, Feb. 1991.

[3] G. A. Khoury, J. Smadbeck, C. A. Kieslich, and C. A. Floudas, "Protein folding and de novo protein design for biotechnological applications," *Trends Biotechnol.*, vol. 32, no. 2, pp. 99–109, Feb. 2014.

[4] C. T. Zhang and K. C. Chou, "Monte Carlo simulation studies on the prediction of protein folding types from amino acid composition.," *Biophys. J.*, vol. 63, no. 6, pp. 1523–9, Dec. 1992.

[5] L. Zhan, J. Z. Y. Chen, and W.-K. Liu, "Conformational study of Met-enkephalin based on the ECEPP force fields.," *Biophys. J.*, vol. 91, no. 7, pp. 2399–404, Oct. 2006.

[6] L. Olivares-Quiroz, L. García, and -Colín Scherer, "Plegamiento de las proteínas: Un problema interdisciplinario," *Rev. Soc. Quím. Méx*, vol. 48, pp. 95–105, 2004.

[7] P. Crescenzi, D. Goldman, C. Papadimitriou, A. Piccolboni, and M. Yannakakis, "On the Complexity of Protein Folding," *J. Comput. Biol.*, vol. 5, no. 3, pp. 423–465, Jan. 1998.

[8] A. Nayeem, J. Vila, and H. A. Scheraga, "A Comparative Study of the Simulated-Annealing and Monte Carlo-with-Minimization Approaches to the Minimum-Energy Structures of Polypeptides: [Met]-Enkephalin," *Journal of Computational Chemistry*, vol. 12, no. 5. John Wiley & Sons, Inc., pp. 594–605, Jun-1991.

[9] P. Bradley, "Toward High-Resolution de Novo Structure Prediction for Small Proteins," *Science (80-.).*, vol. 309, no. 5742, pp. 1868–1871, 2005.

[10] Y. Zhang and L. Wu, "Artificial Bee Colony for Two Dimensional Protein Folding," *Adv. Electr. Eng. Ssytems*, vol. 1, no. 1, pp. 19–23, 2012.

[11] J. H. Meinke, U. H. E. Hansmann, J. H. Meinke, and U. H. E. Hansmann, "Parallelization of ECEPP/3 in SMMP," *From Comput. Biophys. to Syst. Biol.*, vol. 36, pp. 219–222, 2007.

[12] H. G. Naik, "Parallel Algorithms for Computational Biology," *Perform. Comput.*, pp. 1–20, 2007.

[13] Y. Sakae, T. Hiroyasu, M. Miki, and Y. Okamoto, "Protein structure predictions by parallel simulated annealing molecular dynamics using genetic crossover," *J. Comput. Chem.*, vol. 32, no. 7, pp. 1353–1360, 2011.

[14] A. Y. Zomaya, *Parallel Computing for Bioinformatics and Computational Biology*. 2004.

[15] F. P. Agostini, D. D. O. Soares-Pinto, M. A. Moret, C. Osthoff, and P. G. Pascutti, "Generalized simulated annealing applied to protein folding studies," *J. Comput. Chem.*, vol. 27, no. 11, pp. 1142–1155, 2006.

[16] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi, "Optimization by Simulated Annealing," *Sci. New Ser.*, vol. 220, no. 4598, pp. 671–680, 1983.

[17] D. E. Goldberg, "Genetic algorithms in search, optimization, and machine learning.," *Reading, MA Addison-Wesley.*, 1989.

[18] T. Weise, Y. Wu, R. Chiong, K. Tang, and J. Lässig, "Global versus local search : the impact of population sizes on evolutionary algorithm performance," *J. Glob. Optim.*, 2016.

[19] V. T. do Ó and R. Tinós, "A Self-Organizing Genetic Algorithm for Protein Structure Prediction," *Learn. Nonlinear Model.*, vol. 8, no. 3, pp. 135–147, 2010.

[20] Y. Sakae, T. Hiroyasu, M. Miki, K. Ishii, and Y. Okamoto, "Combination of genetic

crossover and replica-exchange method for conformational search of protein systems," 2015.

[21] P.-H. Chen and S. M. Shahandashti, "Hybrid of genetic algorithm and simulated annealing for multiple project scheduling with multiple resource constraints," *Autom. Constr.*, vol. 18, no. 4, pp. 434–443, Jul. 2009.

[22] S. W. Mahfoud and D. E. Goldberg, "Parallel recombinative simulated annealing: A genetic algorithm," *ELSEVIER Parallel Comput.*, vol. 21, pp. 1–28, 1995.

[23] Y. Sakae, T. Hiroyasu, and M. Miki, "Parallel Simulated Annealing using Genetic Crossover," pp. 1–15, 2000.

[24] J. Frausto-Solis, X. Soberon-Mainero, and E. Liñan-Garcia, "MultiQuenching annealing algorithm for protein folding problem," *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 5845 LNAI, pp. 578–589, 2009.

[25] J. Frausto-Solis, E. Liñáan-García, M. Sanchez-Perez, and J. P. Sanchez-Hernandez, "Chaotic multiquenching annealing applied to the protein folding problem," *Sci. World J.*, vol. 2014, 2014.

[26] G. N. Ramachandran, C. Ramakrishnan, and V. Sasisekharan, "Stereochemistry of polypeptide chain configurations.," *J. Mol. Biol.*, vol. 7, no. 1, pp. 95–99, 1963.

[27] F. Eisenmenger, U. H. E. Hansmann, S. Hayryan, and C.-K. Hu, "[SMMP] A modern package for simulation of proteins LONG WRITE - UP," *Comput. Phys. Commun.*, vol. 138, pp. 192–212, 2001.

[28] G. Nemethy, M. S. Pottle, and H. a Scheraga, "Energy parameters in polypeptides. 9. Updating of geometrical parameters, nonbonded interactions and hydrogen bond interactions for the naturally occuring amino acids," *J. Phys. Chem.*, vol. 87, no. 11, pp. 1883–1887, 1983.

[29] J. Frausto-Solis, E. Liñán-García, J. P. Sánchez-Hernández, J. J. González-Barbosa, C. González-Flores, and G. Castilla-Valdez, "Multiphase Simulated Annealing Based on Boltzmann and Bose-Einstein Distribution Applied to Protein Folding Problem.," *Adv. Bioinformatics*, vol. 2016, p. 7357123, 2016.

[30] J. Singh Bhalla and A. Aggarwal, "Prediction of Protein Structure using Parallel Genetic Algorithm," *Int. J. Comput. Appl.*, vol. 81, pp. 7–11, 2013.

[31] "RCSB Protein Data Bank - RCSB PDB." [Online]. Available: http://www.rcsb.org/pdb/home/home.do.

[32] F. Eisenmenger and U. H. E. Hansmann, "Variation of the Energy Landscape of a Small Peptide under a Change from the ECEPP/2 Force Field to ECEPP/3." 1997.

[33] L. Zhan, J. Z. Y. Chen, W.-K. Liu, J. Hughes, T. W. Smith, H. W. Kosterlitz, L. A. Fothergill, B. A. Morgan, H. R. Morris, W. H. Graham, E. S. C. II, R. P. Hicks, G. Nemethy, M. S. Pottle, H. A. Scheraga, M. J. Sippl, G. Nemethy, H. A. Scheraga, G. Nemethy, K. D. Gibson, K. A. Palmer, C. N. Yoon, G. Paterlini, A. Zagari, S. Rumsey, H. A. Scheraga, Z. Li, H. A. Scheraga, U. H. E. Hansmann, J. N. Onuchic, U. H. E. Hansmann, Y. Okamoto, J. N. Onuchic, D. A. Evans, D. J. Wales, B. von Freyberg, W. Braun, F. Eisenmenger, U. H. E. Hansmann, H. Meirovitch, E. Meirovitch, A. G. Michel, M. Vásquez, I. P. Androulakis, C. D. Maranas, C. A. Floudas, Z. Li, H. A. Scheraga, A. G. Michel, C. Ameziane-Hassani, N. Bredin, L. Zhan, J. Z. Y. Chen, W.-K. Liu, U. H. E. Hansmann, Y. Sugita, Y. Okamoto, N. Rathore, T. A. K. IV, J. J. de Pablo, L. Carlacci, K. Vengadesan, N. Gautham, A. Mitsutake, M. Irisa, Y. Okamoto, F. Hirata, J. Koča, P. H. J. Carlsen, Z. Kříža, P. H. J. Carlsen, J. Koča, M. Kinoshita,

Y. Okamoto, F. Hirata, M. Kinoshita, Y. Okamoto, F. Hirata, A. Mitsutake, M. Kinoshita, Y. Okamoto, F. Hirata, A. Mitsutake, M. Kinoshita, Y. Okamoto, F. Hirata, B. A. Berg, H.-P. Hsu, M.-Y. Shen, K. F. Freed, M. H. Zaman, M.-Y. Shen, R. S. Berry, K. F. Freed, T. Montcalm, W. Cui, H. Zhao, F. Guarnieri, S. R. Wilson, B. A. Berg, K. Y. Sanbonmatsu, A. E. García, U. H. E. Hansmann, M. Masuya, Y. Okamoto, E. Marinari, G. Parisi, J. J. Ruiz-Lorenzo, K. Hukushima, K. Nemoto, F. Wang, D. P. Landau, F. Wang, D. P. Landau, U. H. E. Hansmann, L. T. Wille, B. A. Berg, T. Neuhaus, B. A. Berg, T. Neuhaus, J. Lee, M.-H. Hao, H. A. Scheraga, D. J. Wales, H. A. Scheraga, D. J. Wales, J. P. K. Doye, L. Zhan, B. Piwowar, W.-K. Liu, P. J. Hsu, S. K. Lai, J. Z. Y. Chen, L. Zhan, J. Z. Y. Chen, W.-K. Liu, S. K. Lai, P. E. Gill, W. Murray, M. H. Wright, F. Eisenmenger, U. H. E. Hansmann, S. Hayryan, C.-K. Hu, J. S. Richardson, G. D. Rose, L. M. Gierasch, J. A. Smith, E. G. Hutchinson, J. M. Thornton, C. M. Venkatachalam, G. Némethy, M. P. Printz, G. Némethy, H. A. Scheraga, E. J. Milner-White, B. M. Ross, R. Ismail, K. Belhadj-Mostefa, and R. Poet, “Conformational study of Met-enkephalin based on the ECEPP force fields.,” *Biophys. J.*, vol. 91, no. 7, pp. 2399–404, Oct. 2006.

[34] J. Xu and Y. Zhang, “How significant is a protein structure similarity with TM-score = 0.5?,” *Bioinformatics*, vol. 26, no. 7, pp. 889–95, Apr. 2010.

[35] K. A. Dill, S. B. Ozkan, M. S. Shell, and T. R. Weikl, “The protein folding problem.,” *Annu. Rev. Biophys.*, vol. 37, pp. 289–316, 2008.