Journal of Universal Computer Science, vol. 19, no. 15 (2013), 2207-2223
submitted: 21/6/12, accepted: 30/8/13, appeared: 1/9/13 © J.UCS

Cluster Perturbation Simulated Annealing for Protein Folding Problem

Juan Frausto-Solís
(UPEMOR, Jiutepec, México
juan.frausto@upemor.edu.mx)

Mishael Sánchez-Pérez
(Center for Genomic Sciences, UNAM, Cuernavaca, México
mishael@ccg.unam.mx)

Ernesto Liñan-García
(UADEC, Saltillo, México
ernesto_linan_garcia@uadec.edu.mx)

Juan Paulo Sánchez-Hernández
(ITESM Campus Cuernavaca, UPEMOR, Jiutepec, México
juan.paulosh@upemor.edu.mx)

Manoj Ramachandran
(Toc H Institute of Science and Technology, Ernakulam, India
manojkochi@gmail.com)

Abstract: In this paper, an improved Simulated Annealing algorithm for Protein Folding Problem (PFP) is presented. This algorithm called Cluster Perturbation Simulated Annealing *(CPSA)* is based on a brand new scheme to generate new solutions using a cluster perturbation. The algorithm is divided into two phases: Cluster Perturbation Phase and the Reheat Phase. The first phase obtains a good solution in a small amount of time, and it is applied at very high temperatures. The second phase starts with a threshold temperature and reheats the system for a better exploration. CPSA reduces the execution time of the Simulated Annealing Algorithm without sacrificing quality to find a native structure in PFP in Ab-Initio approaches.
Key Words: Simulated Annealing, Protein Folding, Tuned SA, Cluster Perturbation Simulated Annealing
Category: J.0, J.3, J.7

1 Introduction

The design of modern techniques to ameliorate the computational cost of NP Hard problems (NPH) are a huge challenge nowadays [Fraenkel 1993]. Implementation of new optimization methods is important in any instance of NPH such as Protein Folding Problem (PFP) [Unger and Moult 1993]. Thus, heuristic methods avoiding the generation of all possible solution states of the protein has been proposed [Chivian et al., 2005]. New Metaheuristics based on the

Monte Carlo method have been proposed to solve the PFP [Zhan et al., 2006, Frausto-Solis et al., 2009]. In the present work, the proposed algorithm uses only the primary structure of the protein without the use of chaperones, in order to reach the final three-dimensional structure. In the other hand, Ab Initio methods obtain predictions of the protein structures without using information of previously solved structures [M.Zaki 2008]. Computational techniques should be designed to find the optimal solution of the protein folding model, which has the minimum energy and thus determine the three-dimensional native structure of the protein. The paper is organized as follows: In section two, protein folding problem is presented; in section three, Simulated Annealing is briefly reviewed, and Cluster Perturbation method is introduced; section four shows implementation details of the proposed method; and the results of experimentation with this method are shown in section five; finally the conclusions are presented at the end of the paper.

2 Protein Folding Problem

Proteins are linear polypeptide compounds of 20 naturally-occurring amino acids joined by peptide bonds. The proteins fold into a unique biological configuration. The atoms of a protein are intertwined in a three-dimensional geometrical structure where the free Gibbs energy is the lowest [Anfinsen 1961]. The geometry of the protein can be described by the Cartesian coordinates of each atom. The dihedral angles along the backbone of the protein determine the geometrical shape of the protein in the folding process. Commonly these dihedral angles are ϕ, ψ, ω, and χ. In recent years, physics based methods combined with computational procedures that explore the conformational space have been used as a general solution to the problem. These models are based on the thermodynamic hypothesis that the biological native structure is folded as a near minimum energy conformation [Anfinsen 1973, Dill et al., 2008].

The assumption that the protein biologically folds in a minimum energy state, is the most common paradigm used for small protein folding, although there are some well known exceptions [Sohl Julie L. et al., 1998, Baker and Agard 1994]. Molecular dynamics in conjunction with physical potentials are used to analyze the folding behavior. According to physical models, the simulation of large scale proteins can be really difficult for an accurate prediction of protein folding [Dill et al., 2007]. These physical models of energy depend on the distance and interaction among the atoms that conform the protein. For this purpose, the energy can be calculated using the torsion angles of the structure and the distance between any pair of atoms. In the folding process, the possible conformation structures that the protein can take are extremely high, and the required computational time can be unsuitable for practical applications [Levinthal 1968]. In general, PFP is defined as follows:

Given:

- A sequence of n amino acids $(a_1, a_2, a_3, ...a_n)$, representing the primary structure of a protein, and
- An initial set of m dihedral angles $(\sigma_1, \sigma_2, \sigma_3, ..., \sigma_m)$ representing a three-dimensional structure of the protein or peptide,

The problem is:

- To determine the minimal free energy $f_{min}(\sigma_1, \sigma_2, \sigma_3, ...\sigma_m)$ of the given sequence, and
- To find the optimal set of angles, which represents the protein native structure.

The last definition, involves PFP for proteins and peptides; however, as was previously mentioned there are some exceptions [Sohl Julie L. et al., 1998].

Energy Force Fields have been used to represent the molecular interaction of the atoms. The basic form of a force field is conformed by bonded terms (Hydrogen and Torsion Bonds) and nonbonded terms (Electrostatic, and Van der Waals bonds) [Momany et al., 1975].

The force field represents the physical properties, which are tested by structural data-sets obtained from classical analysis like x-ray crystallography. However, the development of new parameters is a laborious task that includes many interactions of atoms, affecting their internal energy.

Equation 1 is the simplest potential energy function to represent the basic features of the protein folding in an atomic approach.

$$\begin{aligned} E_{total} = \sum_{bonds} k_b(b - b_0)^2 + \sum_{angles} k_\theta(\theta - \theta_0)^2 + \sum_{torsion} k_\theta[\cos(n\phi + \delta) + 1] \\ + \sum_{nonbond} [\frac{q_i qj}{r_{ij}} + \frac{A_{ij}}{r_{ij}^{12}} - \frac{C_{ij}}{r_{ij}^6}] \end{aligned} \tag{1}$$

The most common force fields are AMBER [Cornell et al., 1995], CHARMM [Brooks et al., 1983], and ECEPP [Nemethy et al., 1983]. New heuristic methods have been used to solve protein folding problem with these kind of force fields [Frausto-Solis et al., 2007]. One of the most effective is Simulated Annealing (SA) [Chivian et al., 2005, Kirkpatrick et al., 1983, Xu et al., 2003]. The algorithm has been successfully applied in NPH problems [Fonseca et al., 2012, Černý 1985]. SA algorithm is modified in the present work to improve the SA applicability for PFP. As a result, Cluster Perturbation Simulated Annealing (CPSA) is proposed. In the proposed algorithm, the Simulated Annealing parameters should be tuned for finding better solutions than the classical SA algorithm [Frausto-Solis et al., 2006, Pérez et al., 2002]. ECEPP energy function is calculated using SMMP software [Eisenmenger et al., 2006]. This package has successfully used for PFP on Monte Carlo algorithms [Zhan et al., 2006], Genetic Algorithms [Sudha et al., 2013],[Sindhu et al., 2012] and other techniques [Bahamish et al., 2008].

3 Cluster Perturbation Simulated Annealing

CPSA is based on Monte Carlo like methods such as SA. Simulated Annealing could obtain better performance by applying different improvement techniques such as Ad hoc cooling function, improving the generation of new solutions, and the analytical tuning techniques.

3.1 Analytical Tuning

Simulated Annealing was proposed by Kirkpatrick [Kirkpatrick et al., 1983] and by Cerny [Černý 1985] as an extension of the well known Metropolis algorithm [Metropolis et al., 1953].

SA is a simulation of the thermodynamic process where a metal is heated to extremely high temperatures until it melts and then is allowed to cool slowly. The atoms are rearranged with a molecular lowest energy conformation until they obtain a quasi-dynamic equilibrium at the end of the cooling process.

Although, this thermodynamic process is heuristically referred to the cooling of metal, it can be applied in optimization problems. The objective function is related to the energy function of the original problem. SA is really successful in combinatorial optimization problems, where it is difficult to reach the global optimum [Martinez-Rios et al., 2008],[Nolte et al., 2001],
[Peter J. M. van Laarhoven et al., 1992].

Algorithm 1 SA Algorithm

```
Require: $T_i, T_f, S_i, \alpha$
  while $T_i \leqslant T_f$ do
    while METROPOLIS CYCLE do
      $S_j = Perturbation(S_i)$
      $\delta E = E(S_j) - E(S_i)$
      if $\delta E \leqslant 0$ then
        $S_i = S_j$
      else
        if $\exp \frac{-\delta E}{T_i} > random[0,1]$ then
          $S_i = S_j$
        end if
      end if
    end while
    $T_i = \alpha * T_i$
  end while
```

The Algorithm 1 is the pseudocode of classical SA. As it can be observed, the algorithm employs an initial temperature (T_i) and a final temperature (T_f). An initial solution (S_i) is randomly generated, and a geometrical cooling function $T_i = \alpha T_i$ that decrements the temperature by an α factor is applied.

The solution cycles through different states until the best solution is reached; the current solution is an estimation of the global optimum. In the algorithm, a new solution (S_j) is obtained by a generation function referred to a perturbation function, which is used on the previous solution (S_i) for generating a fresh one. A sequence of solutions $S_1, S_2, S_3 ... S_n$ corresponds to a sequence of energy states $e(S_1), e(S_2), e(S_3) ... e(S_n)$ where the energies tend to decrease with the decrements of the cooling function to reach the lowest energy conformation of the system. SA algorithm computes the energy decrement $\delta E = E(S_j) - E(S_i)$; if $(\delta E) < 0$, then the new solution is accepted; otherwise, the new configuration is accepted using the Boltzmann criterion that accepts low quality solutions with a certain probability. SA uses an initial temperature T_i that is related to the acceptance probability of the new solution. At highest temperatures, the acceptance probability of a new solution $P(\Delta Z_{MAX})$ is close to one. However, the acceptance probability is decreased when the temperature is decremented until the final temperature T_f is reached and the minimum acceptance probability $P(\Delta Z_{MIN})$ is obtained.

Analytical tuning method for SA algorithm obtains the initial and final temperatures (T_i) and (T_f) from equations 2 and 3 [Sanvicente-Sánchez et al., 2004]. These temperatures are associated with the maximum and minimum deterioration of the objective function ($\varDelta Z_{MAX}, \varDelta Z_{MIN}$) and the associated acceptance probability from equation 2 and 3.

$$\exp(\frac{-\varDelta Z_{MAX}}{T_i}) = P(\varDelta Z_{MAX}) \tag{2}$$

$$\exp(\frac{-\varDelta Z_{MIN}}{T_f}) = P(\varDelta Z_{MIN}) \tag{3}$$

$$T_i = \frac{-\varDelta Z_{MAX}}{\ln(P(\varDelta Z_{MAX}))} \tag{4}$$

$$T_f = \frac{-\varDelta Z_{MIN}}{\ln(P(\varDelta Z_{MIN}))} \tag{5}$$

Other parameters are also used in this analytical tuning method; one of them is the length of the Markov chain (L_k) of the Metropolis cycle (or number of iterations) [Frausto-Solis et al., 2006]. SA can be adapted using constant or variable Markov Chains. The first approach is in general more easily implemented; however, the last one has shown to achieve a better performance than constant Markov chains [Frausto-Solis et al., 2007]. In order to adjust the variable Markov Chains, the analytical tuning method determines the length of the metropolis cycle.

– The initial temperature in PFP according to equation (4) is usually extremely high. At the highest temperatures, SA has a uniform distribution for new solutions because all of them have an acceptance probability almost equal to one.

  Therefore, at these temperatures, the length (L_k) of the first Metropolis cycle can be settled as small as possible. As a consequence, $L_k = L_1 \approx 1$ is used. In this case, the stochastic equilibrium is reached by the first configuration randomly obtained by the algorithm. When the temperature is decreased to the next cycle, the corresponding Markov chain length L_k must be incremented.

– In SA, the decrements of the temperature parameter can be done by a geometrical cooling function $T_{k+1} = \alpha T(k)$. By contrast, the length of the Markov chain L_k must be incremented by $L(k+1) = \beta L(k)$, where the β factor increases the size of the Markov chain.

– When SA is in the last metropolis cycles, the energy of the protein is too low, and a dynamic equilibrium is reached. This time a bad solution, has a truly small acceptance probability, and it is extremely close to zero. Thus, the length of the Markov chain is really long, and the maximum number of Metropolis cycle is obtained by $L_{MAX} = \beta^n L(1)$.

– To calculate the number of steps (n) to arrive from T_i to T_f we use $n = \frac{\ln T_f - \ln T_i}{\ln \alpha}$. This is the number of times that the Metropolis cycle is executed. Once n is determined, the β coefficient can be calculated using equation 6.

$$\beta = exp\left(\frac{\ln L_{MAX} - \ln L_1}{n}\right) \tag{6}$$

Applying the previous assumptions Tuned Simulated Annealing (TSA) algorithm is obtained. TSA avoids the excessive computational tasks for reaching stochastic equilibrium and make the algorithm become faster than the classical SA [Frausto-Solis et al., 2006],[Frausto-Solis et al., 2007].

3.2 Cluster Perturbation and Reheat

To reduce the computational cost of the Tuned Simulated Annealing, a cluster perturbation method is proposed (CPSA). The perturbation technique in SA consists in modifying the initial solution vector (S_i) with a neighborhood upgrade or modifying the neighborhood order, to generate a new solution (S_n). Perturbation in SA for PFP uses a regular perturbation technique, where one dihedral angle of the chain $\sigma_1, \sigma_2, \sigma_3, ...\sigma_m$ is incremented or decremented by a factor μ.

The Cluster perturbation approach reduces the number of variables to a cluster of certain dihedral angles. Thus, the possible number of dihedral angles to be changed are lower than the regular model.

The protein chain is divided into fragments of four amino acids $a_1, a_2, a_3, ...a_4$ [Levitt 1976]. Each fragment i is composed of all the dihedral angles of the selected amino acids represented by $F_i\{\sigma_{1,i}, \sigma_{2,i}, \sigma_{3,i}, \sigma_{4,i}\}$. The protein is clustered in a new set of variables compound by the first amino acid set of variables a_1 and the last amino acid set of variables a_4 of each fragment. Thus, the amino acids a_1,a_4 are characterized by a set of dihedral angles $(\sigma_{1,i}, \sigma_{4,i})$, in order to obtain a cluster of dihedral angles defined by $Cluster = \sigma_{1,1}, \sigma_{4,1}, \sigma_{1,2}, \sigma_{4,2} ... \sigma_{1,n}, \sigma_{4,n}$, where n is the number of fragments that the protein or peptide is divided. Using this approach the number of variables to perturbate is reduced.

To determine which set is going to be perturbated by the algorithm, (the normal variables or the cluster variables), a uniform probability distribution is used, and a λ factor in the range $0.7 < \lambda < 0.9$ is chosen.

Algorithm 2 Perturbation

```
Require: $T_i, T_t, P(\sigma^{'}), S_i$
  $\lambda = random[0, 1]$
  if $T_i \geqslant T_t$ then
    if $m \leqslant P(\sigma^{'})$ then
      $S_j = ClusterPerturbation(S_i)$
    else
      $S_j = Perturbation(S_i)$
    end if
  else
    $S_j = Perturbation(S_i)$
  end if
  return $S_j$
```

CPSA reduces the computational cost used at very high temperatures to obtain a feasible solution without changing the alpha value of the cooling function. With cluster perturbation alone, the minimal energy obtained for $Met^5 - enkephaline$ is -3.9 $kcal/mol$. To improve the quality of the algorithm, the system is reheated to reach $Met^5 - enkephaline$ minimal energy near -10 $Kcal/mol$.

The algorithm is divided into two phases: the cluster annealing phase and the reheat annealing phase. In the first phase, the new solutions are generated by cluster perturbation , and in the reheat phase, the new solutions are generated by regular annealing perturbation.

The reheat mechanism is used to increase the actual value of a particular temperature T_i in a specific range as shown in Figure 1. The additional computational cost of the reheat phase is not significant because only part of the algorithm is repeated since a certain temperature.

The reheat technique tries to avoid local optima in a certain phase of the CPSA. The temperature is divided into four new temperatures ranges obtained with the experimental tuning process. The correspondent new temperatures establish a threshold where the system is going to be reheated.

4 Implementation

In the present work, the temperature is decreased with a geometrical cooling function $T_i = \alpha * T_i, 0 < \alpha < 1$. Figure 2 shows the behavior of this function for different α values. When SA uses an alpha value near-zero, the cooling function decreases extremely fast, and the optimal solution can be lost. On the other hand, the cooling function decreases extremely slow by choosing an

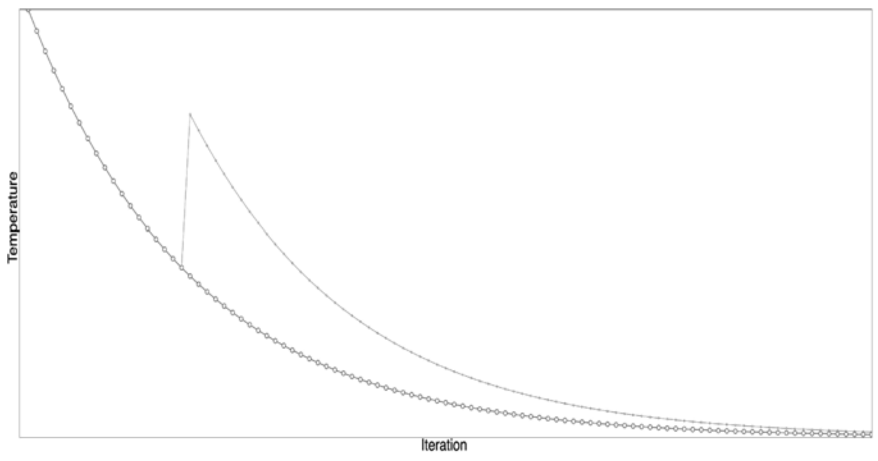

Figure 1: Cooling Behavior of SA with Reheat

alpha value very close to one. In this case the possibilities to find the optimum solution is increased. However, SA implementation can become impractically long. Usually the alpha parameter is chosen in the range $0.7 < \alpha < 0.99$ [Aarts and Korst 1988, Frausto-Solis et al., 2006].

In this work, a high α value of 0.95 was experimentally determined, considering a trade-off between execution time and quality solution.

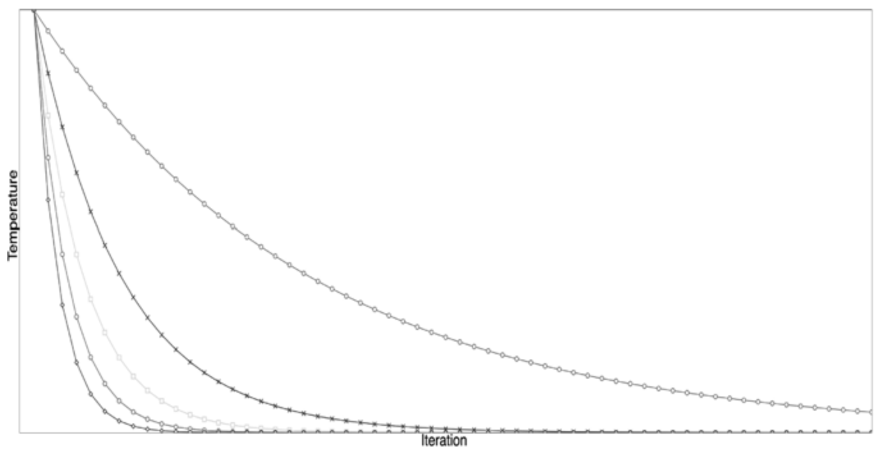

Figure 2: Behavior of Different values of α in the range of $0.7 < \alpha < 0.99$

$Met^5 - enkephaline$, $C - peptide$ and $T0335$ were used as target proteins (see Table 1). $T0335$ is a CASP Benchmark Protein. CASP (Critical Assessment of Techniques for Protein Structure Prediction) is a world wide community for protein structure prediction [Moult et al., 1995].

These proteins are chosen to be compared with the published results of Analytical Tuned Simulated Annealing, that has prediction accuracy of $1 - 3$ angstroms with respect to the protein native structure [Frausto-Solis et al., 2007, Frausto-Solis et al., 2009]. One of these methods uses a change in the alpha value and starts a new phase of the algorithm [Frausto-Solis et al., 2009]. In other hand, the new CPSA algorithm updates the temperature in a certain level. Thus, both strategies have the same prediction accuracy.

The energy function used in this work is ECEPP/2 [Momany et al., 1975]. SMMP software [Eisenmenger et al., 2001], is used to implement the protein folding simulation [Eisenmenger et al., 2006, Meinke et al., 2008].

Table 1: Protein Folding Instances

Instance PFP	Amino acids	Number of Variables
$Met^5 - enkephaline$	5	19
C-peptide	13	68
T0335(Basilius subtilis)(2HE)	85	450

The algorithms TSA, CPSA, and their variants were executed thirty times. The runtime values and minimum and maximum values of energy were obtained for the purpose of comparing the target proteins, and the results are discussed in the next section.

5 Results

In this section, the results of several test cases are presented. The following implementations were compared and tested in this work: SMMP original code, Analytical Tuned Simulated Annealing (TSA) [Frausto-Solis et al., 2007], CPSA and their variants CPSA7, CPSA8, CPSA9 with $\lambda = (0.7, 0.8, 0.9)$ probability of cluster. The algorithms were implemented with a geometrical cooling function with $\alpha = 0.95$; all the results were obtained executing the algorithms thirty times and reporting the average. The results obtained for $Met^5 - enkephaline$ are shown on Table 2; CPSA9 overcomes the quality of the solution for the best solution case than other approaches. The best results in quality solution correspond to CPSA9 (columns 2 of Table 2). CPSA7 obtains the best results for the average solution case (columns 5 of Table 2).

Table 2: $Met^5 - enkephaline$ Results

	Best Solution		Worst Solution		Average Solution	
Approach	Energy Kcal/mol	Time sec	Energy Kcal/mol	Time sec	Energy Kcal/mol	Time sec
SMMP	-10.38	1093	-6.65	1093	-9.40	1091
TSA	-9.41	333.7	-4.3	338.6	-6.97	335.44
CPSAP7	-8.94	376.6	-6.35	376.79	-6.44	381.68
CPSAP8	-9.5	376.6	-4.45	376.8	-6.60	380.89
CPSAP9	-10.48	376.7	-5.03	376.73	-7.52	380.10

Table 2 shows the best, worst and average solution for that CPSA, SMMP and TSA. For the best solution case, CPSA found the lowest energy (-10.48). However, SMMP found the lower energy in the worst and average case for this small peptide.

Table 3: C-Peptide Results

	Best Solution		Worst Solution		Average Solution	
Approach	Energy Kcal/mol	Time sec	Energy Kcal/mol	Time sec	Energy Kcal/mol	Time sec
SMMP	-89.110	86215	-89.11	86215	-89.11	86215
TSA	-109.060	26875	-107.09	26863	-107.64	26898
CPSAP7	-109.060	27666	-103.09	27645	-107.92	27652
CPSAP8	-109.066	27644	-107.08	27653	-108.30	27653
CPSAP9	-109.067	26775	-107.50	26754	-108.42	26761

In Table 3, CPSA and TSA results for a little larger protein (C-peptide) of thirteen amino acids are shown. In this case CPSA9 obtains the best quality results (columns 2-6), that is a considerable reduction in computational time than the other two approaches. The other CPSA variants have better quality results than TSA; besides, the execution time of CPSA9 is significantly reduced, and all the results overcome the quality of the solution of TSA and SMMP.

Table 4: T0335Results

	Best Solution		Worst Solution		Average Solution	
Approach	Energy Kcal/mol	Time sec	Energy Kcal/mol	Time sec	Energy Kcal/mol	Time sec
SMMP	-404.93	800404	-403.47	800405	-404.57	800402
TSA	-403.65	40828	-366.00	33745	-373.49	37287
CPSAP7	-404.43	34408	-355.33	33457	-385.22	34965
CPSAP8	-434.28	33549	-366.55	33456	-386.55	36573
CPSAP9	-443.52	33514	-328.56	33510	-387.49	35726

The results obtained with CPSA for $T0335$ are shown in Table 4. First of all, it can be observed that the execution time is significantly reduced in contrast with TSA. Notice that the quality solution is also increased with CPSA9 for all the target proteins tested. Additionally, in the previous experimentation a three reheats technique of CPSA is applied. However, to reduce the computational cost of CPSA algorithms, the number of reheats can be reduced to only one. Finally, in Table 5 the best implementation of the Cluster perturbation method is compared versus TSA and SMMP. A significant reduction in execution time by using CPSA with one reheat can be observed for all the instances tested in this work. We observed that whether the number of the reheat is risen the computational cost of CPSA is increased. CPSA overcomes TSA with an important reduction in execution time, and the best quality solution is obtained. SMMP with constant default values obtains good results for the worst and average cases; however, the computational time taken for hard proteins like T0335 is excessively large. This can be avoided by reducing the number of iterations that the SMMP algorithm performs, but the quality of the solution would be affected.

Table 5: Final Results (All Proteins with the algorithms SMMP, TSA, and CP90)

		Best Solution		Worst Solution		Average Solution	
Approach	Protein	Energy Kcal/mol	Time sec	Energy Kcal/mol	Time sec	Energy Kcal/mol	Time sec
SMMP	MET	-10.38	1093	-6.65	1093	-9.40	1091
TSA	MET	-9.41	333	-4.30	338	-6.97	335
CPSA	MET	-10.71	368	-5.87	368	-7.75	368
SMMP	C-PEPTIDE	-89.11	86215	-89.11	86215	-89.11	86215
TSA	C-PEPTIDE	-109.06	26875	-107.09	26863	-107.64	26898
CPSA	C-PEPTIDE	-109.07	25802	-106.03	25816	-108.45	25810
SMMP	T0335	-404.93	800404	-403.47	800405	-404.57	800402
TSA	T0335	-403.65	40828	-366.00	33745	-373.49	37287
CPSA	T0335	-443.52	33444	-366.55	33456	-386.55	36573

According to the results presented in Table 5. CPSA obtained the best results for $Met^5 - enkephaline$, these results are to close with that obtained by those recently publised in the literature [Sudha et al., 2013].

6 Conclusions

In this paper Cluster Perturbation Simulated Annealing algorithm (CPSA) for small proteins in PFP is presented. The new algorithm remains simple as the classical simulated annealing (SA) with a geometrical cooling scheme, and a well-known tuning parameters is applied. At the beginning of the algorithm, the solution is generated with a new technique called cluster perturbation. This approach applies a probability distribution in a perturbation scheme to generate new solutions by modifying fragments of the old ones. A particular fragment is conformed by the initial and final set of variables which are adjusted by the perturbation scheme. This first phase reduces the execution time of the algorithm compared with the original SMMP and TSA (Simulated Annealing algorithm using tuning parameters). Basically, in TSA new solutions are generated by typical generation functions as in the classical SA. The performance of simulated annealing algorithm is really ameliorated by using CPSA. The important reduction of computational cost of the cluster perturbation approach let us add new techniques for improving the quality of the solution of the algorithm. In CPSA, a reheat technique is applied as a second phase. The time taken for this phase is worthwhile because it improves the quality of solution while the total execution time is still lower than TSA. In this paper, a tuning method is adapted for CPSA.

It includes initial, final and reheat temperatures, number of metropolis cycles, and the metropolis length chains. In conclusion, as the experimentation shows the execution time is really better than TSA while the quality obtained with CPSA is very close or better than the ones obtained by TSA implementation and SMMP.

Acknowledgments and Authors' Contribution

Authors F.S.-J and S.P.-M contributed equally to development of this article. We thank the acknowledges DGAPA-UNAM and CCG-UNAM for a postdoctoral scolarship also thanks CONACyT-México for PhD scholarship.

References

[Aarts and Korst 1988] E. Aarts and J. Korst. *Simulated Annealing and Boltzmann Machines: A Stochastic Approach to Combinatorial Optimization and Neural Computing*. Wiley, 1 edition, 1988.

[Anfinsen 1973] C. Anfinsen. Principles that govern the folding of protein chains. *Science*, 181(96):223–230, 1973. ISSN 0036-8075. doi: 10.1126/science.181.4096.223.

[Anfinsen 1961] C. B. Anfinsen. The influences of three-dimensional configuration on the chemical reactivity and stability of proteins. *Journal of Polymer Science*, 49(151):31–49, 1961. ISSN 1542-6238. doi: 10.1002/pol.1961.1204915103. URL `http://dx.doi.org/10.1002/pol.1961.1204915103`.

[Bahamish et al., 2008] H. A. A. Bahamish, R. Abdullah, and R. A. Salam. Protein Conformational Search Using Bees Algorithm. In *Proceedings of the 2008 Second Asia International Conference on Modelling & Simulation (AMS)*, AMS '08, pages 911–916, Washington, DC, USA, 2008. IEEE Computer Society. ISBN 978-0-7695-3136-6. doi: 10.1109/AMS.2008.65. URL `http://dx.doi.org/10.1109/AMS.2008.65`.

[Baker and Agard 1994] D. Baker and D. A. Agard. Kinetics versus thermodynamics in protein folding. *Biochemistry*, 33(24):7505–7509, 1994. doi: 10.1021/bi00190a002. URL `http://pubs.acs.org/doi/abs/10.1021/bi00190a002`.

[Brooks et al., 1983] B. R. Brooks, R. E. Bruccoleri, B. D. Olafson, D. J. States, S. Swaminathan, and M. Karplus. CHARMM: A program for macromolecular energy, minimization, and dynamics calculations. *Journal of Computational Chemistry*, 4(2):187–217, 1983. ISSN 1096-987X. URL `http://dx.doi.org/10.1002/jcc.540040211`.

[Chivian et al., 2005] D. Chivian, D. E. Kim, L. Malmström, J. Schonbrun, C. A. Rohl, and D. Baker. Prediction of CASP6 structures using automated robetta protocols. *Proteins: Structure, Function, and Bioinformatics*, 61(S7):157–166, 2005. ISSN 1097-0134. doi: 10.1002/prot.20733. URL `http://dx.doi.org/10.1002/prot.20733`.

[Cornell et al., 1995] W. D. Cornell, P. Cieplak, C. I. Bayly, I. R. Gould, K. M. Merz, D. M. Ferguson, D. C. Spellmeyer, T. Fox, J. W. Caldwell, and P. A. Kollman. A Second Generation Force Field for the Simulation of Proteins, Nucleic Acids, and Organic Molecules. *Journal of the American Chemical Society*, 117(19):5179–5197, 1995. doi: 10.1021/ja00124a002. URL `http://pubs.acs.org/doi/abs/10.1021/ja00124a002`.

[Dill et al., 2007] K. A. Dill, S. B. Ozkan, T. R. Weikl, J. D. Chodera, and V. A. Voelz. The protein folding problem: when will it be solved? *Current Opinion in Structural Biology*, 17(3):342–346, 2007. ISSN 0959-440X.

[Dill et al., 2008] K. A. Dill, S. B. Ozkan, M. S. Shell, and T. R. Weikl. The Protein Folding Problem. *Annual Review of Biophysics*, 37(1):289–316, 2008. doi: 10.1146/annurev.biophys.37.092707.153558. URL `http://www.annualreviews.org/doi/abs/10.1146/annurev.biophys.37.092707.153558`. PMID: 18573083.

[Eisenmenger et al., 2001] F. Eisenmenger, U. Hansmann, S. Hayryan, and C. Hu. [SMMP] a modern package for simulation of proteins. *Computer Physics Communications*, 138(2), 2001.

[Eisenmenger et al., 2006] F. Eisenmenger, U. H. Hansmann, S. Hayryan, and C.-K. Hu. An enhanced version of SMMP–open-source software package for simulation of proteins. *Computer Physics Communications*, 174(5):422–429, 2006. ISSN 0010-4655. doi: 10.1016/j.cpc.2005.10.013. URL `http://www.sciencedirect.com/science/article/pii/S0010465505005850`.

[Fonseca et al., 2012] G. H. G. Fonseca, S. S. Brito, and H. G. Santos. A Simulated Annealing Based Approach to the High School Timetabling Problem. In H. Yin, J. A. F. Costa, and G. D. A. Barreto, editors, *Intelligent Data Engineering and Automated Learning - IDEAL 2012*, volume 7435 of *Lecture Notes in Computer Science*, pages 540–549. Springer, 2012. ISBN 978-3-642-32638-7.

[Fraenkel 1993] A. Fraenkel. Complexity of protein folding. *Bulletin of Mathematical Biology*, 55:1199–1210, 1993. ISSN 0092-8240. URL `http://dx.doi.org/10.1007/BF02460704`. 10.1007/BF02460704.

[Frausto-Solis et al., 2006] J. Frausto-Solís, H. Sanvicente-Sánchez, and F. Imperial-Valenzuela. ANDYMARK: An Analytical Method to Establish Dynamically the Length of the Markov Chain in Simulated Annealing for the Satisfiability Problem. In T.-D. Wang, X. Li, S.-H. Chen, X. Wang, H. Abbass, H. Iba, G.-L. Chen, and X. Yao, editors, *Simulated Evolution and Learning*, volume 4247 of *Lecture Notes in Computer Science*, pages 269–276. Springer Berlin / Heidelberg, 2006.

[Frausto-Solis et al., 2007] J. Frausto-Solis, E. Román, D. Romero, X. Soberon, and E. Liñán-García. Analytically Tuned Simulated Annealing Applied to the Protein Folding Problem. In Y. Shi, G. van Albada, J. Dongarra, and P. Sloot, editors, *Computational Science – ICCS 2007*, volume 4488 of *Lecture Notes in Computer Science*, pages 370–377. Springer Berlin / Heidelberg, 2007. ISBN 978-3-540-72585-5.

[Frausto-Solis et al., 2009] J. Frausto-Solis, X. Soberon-Mainero, and E. Liñán-García. MultiQuenching Annealing Algorithm for Protein Folding Problem. In A. Aguirre, R. Borja, and C. Garciá, editors, *MICAI 2009: Advances in Artificial Intelligence*, volume 5845 of *Lecture Notes in Computer Science*, pages 578–589. Springer Berlin / Heidelberg, 2009.

[Garey and Johnson 1979] M. R. Garey and D. S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman and Co., 1979.

[Pérez et al., 2002] Pérez, Joaquín and Pazos, Rodolfo and Velez, Laura and Rodriguez, Guillermo. Automatic Generation of Control Parameters for the Threshold Accepting Algorithm. In C. Coello Coello, A. de Albornoz, L. Sucar, and O. Battistutti, editors, *MICAI 2002: Advances in Artificial Intelligence*, volume 2313 of *Lecture Notes in Computer Science*, pages 125–144. Springer-Verlag, 2002.

[Kirkpatrick et al., 1983] S. Kirkpatrick, C. Gelatt, and M. Vecchi. Optimization by Simulated Annealing. *Science;*, 220(4598):671–680, 1983. ISSN 0036-8075. doi: 10.1126/science.220.4598.671.

[Levinthal 1968] C. Levinthal. Are There Pathways For Protein Folding. *Journal of Medical Physics*, 65(1):44–45, 1968.

[Levitt 1976] M. Levitt. A simplified representation of protein conformations for rapid simulation of protein folding . *Journal of Molecular Biology;*, 104(1):59–107, 1976. ISSN 00222836. doi: 10.1016/0022-2836(76)90004-8.

[Martinez-Rios et al., 2008] F. Martinez-Rios and J. Frausto-Solis. An hybrid simulated annealing threshold accepting algorithm for satisfiability problems using dynamically cooling schemes. *W. Trans. on Comp.*, 7:374–386, May 2008. ISSN 1109-2750. URL `http://dl.acm.org/citation.cfm?id=1457927.1457931`.

[Meinke et al., 2008] J. H. Meinke, S. Mohanty, F. Eisenmenger, and U. H. Hansmann. SMMP v. 3.0 Simulating proteins and protein interactions in Python and Fortran. *Computer Physics Communications*, 178(6):459–470, 2008. ISSN 0010-4655. doi: 10.1016/j.cpc.2007.11.004. URL `http://www.sciencedirect.com/science/article/pii/S0010465507004614`.

[Metropolis et al., 1953] N. Metropolis, A. Rosenbluth, M. Rosenbluth, A. Teller, and E. Teller. Equation of State Calculations by Fast Computing Machines. *The Journal of Chemical Physics*, 21(6):1087–1092, 1953. doi: 10.1063/1.1699114.

[Momany et al., 1975] F. A. Momany, R. F. McGuire, A. W. Burgess, and H. A. Scheraga. Energy parameters in polypeptides. VII. Geometric parameters, partial atomic charges, nonbonded interactions, hydrogen bond interactions, and intrinsic torsional potentials for the naturally occurring amino acids. *The Journal of Physical Chemistry*, 79(22):2361–2381, 1975. doi: 10.1021/j100589a006. URL `http://pubs.acs.org/doi/abs/10.1021/j100589a006`.

[Moult et al., 1995] J. Moult, J. T. Pedersen, R. Judson, and K. Fidelis. A large-scale experiment to assess protein structure prediction methods. *Proteins: Structure, Function, and Bioinformatics*, 23(3):ii–iv, 1995. ISSN 1097-0134.

[M.Zaki 2008] C. B. M.Zaki. *Protein Struture Prediction.* Humana Press, Totowa,New Jersey, USA, 2008. ISBN 978-1-59745-574-9.

[Nemethy et al., 1983] 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 occurring amino acids. *The Journal of Physical Chemistry*, 87(11):1883–1887, 1983. doi: 10.1021/j100234a011. URL `http://pubs.acs.org/doi/abs/10.1021/j100234a011`.

[Nemethy et al., 1992] G. Nemethy, K. D. Gibson, K. A. Palmer, C. N. Yoon, G. Paterlini, A. Zagari, S. Rumsey, and H. A. Scheraga. Energy parameters in polypeptides. 10. Improved geometrical parameters and nonbonded interactions for use in the ECEPP/3 algorithm, with application to proline-containing peptides. *The Journal of Physical Chemistry*, 96(15):6472–6484, 1992. doi: 10.1021/j100194a068. URL `http://pubs.acs.org/doi/abs/10.1021/j100194a068`.

[Nolte et al., 2001] A. Nolte and R. Schrader. Simulated Annealing and Graph Colouring. *Comb. Probab. Comput.*, 10:29–40, January 2001. ISSN 0963-5483. URL `http://dl.acm.org/citation.cfm?id=971184.971186`.

[Peter J. M. van Laarhoven et al., 1992] E. H. L. A. Peter J. M. van Laarhoven and J. K. Lenstra. Job Shop Scheduling by Simulated Annealing. *Operations Research*, Vol. 40:113–125, 1992.

[Sanvicente-Sánchez et al., 2004] H. Sanvicente-Sánchez and J. Frausto-Solís. A Method to Establish the Cooling Scheme in Simulated Annealing Like Algorithms. In A. Laganá, M. Gavrilova, V. Kumar, Y. Mun, C. Tan, and O. Gervasi, editors, *Computational Science and Its Applications – ICCSA 2004*, volume 3045 of *Lecture Notes in Computer Science*, pages 755–763. Springer Berlin / Heidelberg, 2004. ISBN 978-3-540-22057-2.

[Sindhu et al., 2012] G. Sindhu and S. Sudha. Prediction of Protein Tertiary Structure Using Genetic Algorithm. *Soft Computing Techniques in Vision Science*, 395:15–22, 2012. doi: 10.1007/978-3-642-25507-6_2. URL `http://dx.doi.org/10.1007/978-3-642-25507-6_2`.

[Sohl Julie L. et al., 1998] Sohl Julie L., Jaswal Sheila S., and Agard David A. Unfolded conformations of [alpha]-lytic protease are more stable than its native state. *Nature*, 395(6704):817–819, oct 1998. ISSN 0028-0836. doi: 10.1038/27470. 10.1038/27470.

[Sudha et al., 2013] S. Sudha, S. Barkar, and S. Krishnaswamy. Protein Tertiary Structure prediction using evolutionary algorithms. *International Journal of Emerging Technologies in Computational and Applied Sciences (IJETCAS)*, pages 338–348, 2013.

[Unger and Moult 1993] R. Unger and J. Moult. Finding the lowest free energy conformation of a protein is an NP-hard problem: Proof and implications. *Bulletin of Mathematical Biology*, 55:1183–1198, 1993. ISSN 0092-8240. URL `http://dx.doi.org/10.1007/BF02460703`. 10.1007/BF02460703.

[Černý 1985] V. Černý. Thermodynamical approach to the traveling salesman problem: An efficient simulation algorithm. *Journal of Optimization Theory and Applications;*, 45(1):41–51, 1985. ISSN 0022-3239. doi: 10.1007/BF00940812.

[Xu et al., 2003] J. Xu, M. Li, D. Kim, and Y. Xu. RAPTOR: optimal protein threading by linear programming. *J Bioinform Comput Biol*, 1(1):95–117, Apr. 2003. ISSN 0219-7200. URL `http://view.ncbi.nlm.nih.gov/pubmed/15290783`.

[Zhan et al., 2006] L. Zhan, J. Z. Chen, and W.-K. Liu. Conformational Study of Met-Enkephalin Based on the ECEPP Force Fields . *Biophysical Journal*, 91(7): 2399–2404, 2006.