Analytically Tuned Simulated Annealing Applied to the Protein Folding Problem

Juan Frausto-Solis[1], E.F. Román[1], David Romero[2], Xavier Soberon[3], and Ernesto Liñán-García[4]

[1] ITESM Campus Cuernavaca, Paseo de la Reforma 182-A Col. Lomas de Cuernavaca, 62589, Temixco Morelos, México
juan.frausto@itesm.mx, A00376933@itesm.mx
[2] IMASS UNAM
davidr@matcuer.unam.mx
[3] IBT UNAM
soberon@ibt.unam.mx3
[4] Universidad Autónoma de Coahuila
elinan@mail.uadec.mx

Abstract. In this paper a Simulated Annealing algorithm (SA) for solving the Protein Folding Problem (PFP) is presented. This algorithm has two phases: quenching and annealing. The first phase is applied at very high temperatures and the annealing phase is applied at high and low temperatures. The temperature during the quenching phase is decreased by an exponential function. We run through an efficient analytical method to tune the algorithm parameters. This method allows the change of the temperature in accordance with solution quality, which can save large amounts of execution time for PFP.

Keywords: Peptide, Protein Folding, Simulated Annealing.

1 Introduction

The protein folding problem (PFP) is one of the most challenging problems in the bioinformatics area. The folding protein process starts with an initial protein state (i.e. special configuration of amino acids' atoms), followed by intermediate states and ends in a final state. The final state is known as native structure, which is characterized by the minimal energy in the last configuration of amino acids' atoms. The natural protein folding process is not yet completely understood; the protein follows an unknown path from any conformation to its native structure [1]. It seems that in natural folding, the protein does not explore all its possible states [2]. In order to save time, computational folding simulation helps to find the native structure of a given protein and avoids generating all the possible states. Ab Inition Methods are very popular to predict protein final conformation. The protein states are characterized by their energy which depends on the interaction among their atoms. Atomic energies are affected by position of atoms, torsion angles and distance among atoms. The force fields are used to measure the configuration energies of a protein;

Y. Shi et al. (Eds.): ICCS 2007, Part II, LNCS 4488, pp. 370–377, 2007.

these include many interactions among atoms, affecting different energies; the most important are: 1) Torsional energy; 2) Hydrogen bonds energy; 3) Non–bonded energy; 4) Electrostatic energy. The most popular and successful software systems for calculating force fields are AMBER [3], CHARMM [4], ECEPP/2 [5] and ECEPP/3 [6]. Heuristic methods are used for solving PFP, the most common are: Genetic Algorithms, Simulated Annealing (SA), Neural Network, and Tabu Search. SA provides excellent solutions [7]–[20] in a short execution time [21]–[22]; it is an analogy with thermodynamics and the way that liquids freeze and crystallize. The SA parameters must be tuned for finding good solutions; these parameters are obtained by an analytical method [27] or by experimentation [31]-[32]. Analytical methods are used for defining the parameters with formal models; on the other hand, in experimental methods, the parameters are defined by trial and error. Once SA is tuned, it is executed to obtain very good solutions; during the execution, the temperature changes in accordance with equilibrium stochastic, which is detected by three methods [24]: (1) trial and error, (2) mean and standard deviation and (3) accepted solutions vs proposed solutions criterion. Recently, a new method was developed [23] to set the cooling scheme parameters in SA Algorithms, this method establishes that both, the initial and final temperature are a function of the maximum and minimum cost increment obtained from the neighborhood structure. This method has been applied to solve NP-Hard Problems like Satisfiability problem [SAT] [23]–[24]. This papers deals with a new SA algorithm for PFP. The proposed algorithm has two phases named Quenching and Annealing Phases. The first phase is an analogy of the physical quenching process, which is similar to the annealing process but the temperature, is quickly decreased until a quasi-thermal equilibrium is reached. In the case of PFP, the energy is changed in a chaotic way because it has extreme variations. The quenching phase is applied at very high temperatures and decreased with an exponential function. Once the quasi-thermal is reached by this function, the algorithm starts the annealing phase, which gradually reduces the temperature values adapting the analytical tuning [23] methods to PFP.

2 Analytical Tuning

2.1 Setting Initial and Final Temperatures

Analytical tuning can be helpful for setting up the initial temperature. The probability of accepting any new solution is near to 1 at high temperatures, so, the deterioration of cost function is maximal. The initial temperature C(1) is associated with the maximum deterioration admitted and the defined acceptance probability. Let Si be the current solution and Sj a new proposed one, and Z(Si) and Z(Sj) are the costs associated to Si and Sj; the maximum and minimum deteriorations are expressed as ΔZmax and ΔZmin. Then, the probability P($\Delta Zmax$) of accepting a new solution with the maximum deterioration is (1) and then C(1) can be calculated as in (2). In a similar way, the final temperature is established according to the probability P(ΔZmin) of accepting a new solution with the minimum deterioration (see (3)).

$$\exp\left(\frac{-\Delta Z \max}{C(1)}\right) = P(\Delta Z \max) \ . \tag{1}$$

$$C(1) = \frac{-\Delta Z \max}{\ln(P(\Delta Z \max))} \ . \tag{2}$$

$$C(f) = \frac{-\Delta Z \min}{\ln(P(\Delta Z \min))} \ . \tag{3}$$

With these parameters, SA is able to find solutions near the optimal or in some cases, the optimal one. The initial temperature can be extremely high because according to (2), C(1) is extremely affected by $\Delta Z max$.

2.2 Setting the Markov Chain Length

SA can be devised with constant or variable Markov Chains (MC). Let L(k) be the number of iterations at k temperature in Metropolis Loop (ML); it can be set as a multiple of variables of the problem. In SA with constant MC, L(k) is set as a constant for all the temperatures; in other implementations, ML is stopped by a certain number of accepted solutions. On other hand, analytical methods determine L(k) with a simple Markov model [23]; at high temperatures, only a few iterations are required because the stochastic equilibrium is quickly reached; nevertheless, at low temperatures a more exhaustive exploration is needed, so, a larger L(k) is used. Let L(1) be L(k) at C(1) and Lmax be the maximum MC length; C(k) is decreased by the cooling function (4), where α parameter is between 0.7 and 0.99 [21]–[22] and L(k) is calculated with (5):

$$C(k+1) = \alpha C(k) \ . \tag{4}$$

$$L(k+1) = \beta L(k) \ . \tag{5}$$

In (5), β is the increment coefficient of MC (>1); so, L(k+1) > L(k) and L(1) =1 and the last MC L(f) is equal to Lmax. The functions (4) and (5) are applied successively in SA from C(1) to C(f); consequently C(f) and Lmax can be obtained in (6) and (7).

$$C(f) = \alpha^n C(1) \ . \tag{6}$$

$$L\max = \beta^n L(1) \ . \tag{7}$$

In (6) and (7), n is the step number from C(1) to C(f); so we get (8) and (9).

$$n = \frac{\ln C(f) - \ln C(1)}{\ln \alpha} \ . \tag{8}$$

$$\beta = \exp\left(\frac{\ln L\max - \ln L(1)}{n}\right) . \tag{9}$$

This tuning approach prevents: a) SA spends a large amount of time making computations even though the stochastic equilibrium is indeed reached or b) SA stops far away the equilibrium state. So, SA becomes faster than other implementations. As we have shown, Metropolis parameters depend only on the definition of the C(1) and C(f) shown in section 2.1. Lmax must be set to a value that allows a good exploration (between 1 to 4 times the neighborhood size or 63% to 99%) [23].

3 Implementation

The general cooling scheme was tested with two small proteins (Met^5–enkephaline, C-peptide). SMMP was used [28]–[29], and the objective to evaluate the conformation energy function with ECEPP/2 [5]. Neighbor solutions were selected randomly (angles in [-180º, 180º]), and C(1) and C(f) were calculated using $P(\Delta Zmax)=0.7$ and $P(\Delta Zmin)=0.3$. If the former probability were superior to 0.70 and closer to 1, it would allow excellent exploration, but SA would be inefficient; on the contrary, with lower values, SA would have a short exploration level but it would not be able to find a good solution. The initial temperature ($C(1)=1.76x10^{25}$) is extremely high because the high values of the energies; therefore, $\Delta Zmax$ has extremely high or low values; C(f) is set as 0.001. At the end of the process, a small probability to accept deteriorations is enough and after trial and error, 0.3 was chosen. In other words, the general cooling scheme establishes the adequate value of C(1) to perform a better stochastic walk. Nevertheless, the cooling function allows that the temperature decreases very fast at the beginning of the process (chaos phase) and the chaos phase. The cooling function at this phase is given by (10), and it uses (11) and (12):

$$c(k+1) = \alpha \times \gamma_k \times c(k) \tag{10}$$

$$\gamma_k = (1 - \tau_k) . \tag{11}$$

$$\tau_k = \tau_{k-1}^2 . \tag{12}$$

In (11) and (12), $0 < \tau_1 < 1$ but closer to one (e.g. 0.999); therefore (11) gradually converges to one and, the cooling function becomes equivalent to (4). The α value is changed according to the percentage of accepted solutions into the Markov chain, and its value is different for each implementation. When τ_1 reaches one, the quenching phase ends and the annealing phase starts. At the beginning of the quenching process, α is set as 0.7 There are five different tested analytical approaches: one with constant MC length equal to 3,600 for all the temperatures range, another one with adaptable MC length, and the other three with growing MC length (from L(1)=360 to L(f)=3,600); these values are obtained by different values of the α parameter and then equations (8) and (9) are applied for calculating the Metropolis parameters.

4 Results

The following implementations were tested and compared: 1) Original SMMP code [26]; 2) Experimental tuning with MC of constant length (ESAC); 3) Experimental tuning with MC of adaptable length as [6] (ESAP); 4) Experimental with MC of adaptable length as [6] and low dispersion as [6] (ESAD); 5) Analytical tuning with MC of constant length (ASAC); 6) Analytical tuning with MC of adaptable length as [6] (ASAA); 7) Analytical tuning with MC of growing length and regular cooling (α = 0.7, 0.85 and 0.95) (ASAR); 8) Analytical tuning with MC of growing length and slow cooling (α = 0.7, 0.85 and 0.98) (ASAS); 9) Analytical tuning with MC of growing length and regular/slow cooling (α = 0.7, 0.85, 95 and 0.98) (ASARS). Table 1 shows the results for *Met*^{5}*–enkephaline*, and table 2 shows the results for *C-peptide*. These tables show the average and standard deviation of the results obtained after thirty tests in each case. Results are displayed in terms of the cost for the final solution and the time required for finding it. All the results were validated in Ramachandran Plots [30] and we can notice that all the final configurations have angles into the feasible region. The final configuration of ASAR was also very similar to the one reported in PDB (Protein Data Bank, www.pdb.org). When the searching

Table 1. Met^5 Enkephaline Average of the results, this includes best and worst solutions

Average of the results. Time (minutes) and Energies (Kcal/mol)				
Approach	Average Time	Std. Dev	Average Energies	Std. Dev.
SMMP	11.9	0.05	-9.1674	2.3145
ESAC	3.1	0.05	-8.9461	2.2017
ESAP	13.4	7.86	-7.8249	1.5978
ESAD	2.1	0.24	-6.9538	1.0404
ASAC	5.5	0.02	-9.8721	0.5233
ASAA	3.6	0.50	-6.2292	2.0609
ASAR	2.2	0.12	-8.0136	1.4801
ASAS	4.5	0.12	-8.7191	1.5968
ASARS	4.0	0.07	-8.1564	1.3745
	Best Solutions		Worst Solutions	
Approach	Time	Energies	Time	Energies
SMMP	11.9	-10.3897	11.9	-6.6521
ESAC	3.1	-10.7110	3.1	-4.2083
ESAP	20.8	-10.7032	5.7	-5.2110
ESAD	2.3	-9.3143	1.9	-5.8780
ASAC	5.5	-10.7101	5.5	-6.2117
ASAA	4.6	-10.0857	3.1	-1.9257
ASAR	2.3	-10.6886	2.0	-3.2091
ASAS	4.6	-10.6768	4.4	-7.0253
ASARS	3.9	-10.6462	3.8	-5.4932

Table 2. C-Peptide Average of the results

Average of the results. Time (minutes) and Energies (Kcal/mol)				
Approach	Average Time	Std. Dev	Average Energies	Std. Dev.
SMMP	237.3	12.91	-97.8376	4.5717
ESAC	13.9	3.22	-76.8528	5.2947
ESAP	319.6	51.15	-88.8809	6.9504
ESAD	177.0	36.00	-73.6233	4.9742
ASAC	37.8	7.06	-82.6164	6.3692
ASAA	70.1	15.62	-78.9747	5.9442
ASAR	18.0	4.70	-77.3030	5.7723
ASAS	37.9	5.58	-80.6623	7.2507
ASARS	37.3	2.54	-80.1416	5.3733
	Best Solutions		Worst Solutions	
Approach	Time	Energies	Time	Energies
SMMP	244.2	-101.9443	243.3	-86.7523
ESAC	12.7	-90.4995	12.7	-67.1276
ESAP	278.5	-103.7011	286.8	-76.2635
ESAD	155.8	-80.3128	158.5	-65.1571
ASAC	27.5	-102.7710	-27.9	-74.0381
ASAA	70.3	-95.5888	59.7	-70.1984
ASAR	18.2	-91.9294	21.6	-67.3917
ASAS	48.5	-96.7650	49.4	-70.3337
ASARS	37.5	-95.4636	38.6	-70.3716

process of these implementations is close to the end, the variables of the problem converge to a specific value; the total variables of *Met*^{5}*–enkephaline* are nineteen, of which only seventeen are clearly convergent. We made additional experimentation using a Genetic Algorithm obtaining the worst results. With *Met*^{5}*–enkephaline* the algorithm reached only -3.5 Kal/mol in average, and with C-Peptide an average of -57 Kcal/mol was obtained.

5 Conclusions

A SA algorithm for Folding Problems is presented in this paper. This algorithm uses extremely high temperatures in a chaos (quenching) phase allowing the exploration of a bigger percentage of the solution space than previous SA approaches; the algorithm uses two phases, one for the chaos phase where temperatures are too high and the other for lower temperatures. The results presented in the paper with two peptides show that the new approach is able to find solutions with better quality than the classical SA algorithms. According to this experimentation, the general cooling scheme presented here obtains very good results. This method is useful for setting the initial temperature of SA applied to the protein folding problem and it guarantees to reach more suitable solutions. This method also provides a good technique to save time execution at high temperatures using dynamic Markov Chains. The values of the cost function for the best configurations obtained with the analytical implementations

are fairly close to each other and they are very close to those obtained by experimental tuning procedures. For the quenching phase, a cooling function decreasing gradually the temperature of the system is presented. The most remarkable advantage of this tuning method is the saving of time in setting the initial and final temperatures. Now we are validating these approaches with larger common proteins; it represents a very interesting greater challenge.

References

1. Anfinsen, C.: Principles that govern the folding of protein chains. Science 181, (1973) 223 – 230.
2. Levinthal, C.: Are there pathways for protein folding?. J. Chem. Phys. 65, (1968) 44 – 45.
3. Ponder, J.: Case, Force fields for protein simulations. Adv. Prot. Chem. 66, (2003) 27 – 85.
4. Brooks, R., Bruccoleri, R., Olafson, B., States, D., Swaminathan, S., Karplus, M.: A program for macromolecular energy, minimization, and dynamics calculations. J. Comp. Chem. 4, (1983) 187 – 217.
5. Momany, F., McGuire, R., Burgess, A., Scheraga, H.: Energy Parameters in Polypeptide. 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. Vol 79, No. 22, (1975).
6. Nemethy, G., Gibson, K., Palmer, K., Yoon, C., Paterlini, G., Zagari, A., Rumsey, S., Scheraga, H.: Energy parameters in polypeptides. 10. Improved geometrical parameters and nonbonded interactions for use in the ECEPP/3 algorithm with application to proline-*containing peptides*. J. Phys. Chem. 18, 323. (1992).
7. Morales, L., Garduño, R., Romero, D.: Application for simulated annealing to the multiple – minima problem in small peptides. J. Biomol. Str. And Dyn. 8, (1991) 1721 – 735.
8. Morales, L., Garduño, R., Romero, D.: The multiple – minima problem in small peptide revisited. The threshold accepting approach. J. Biomol. Str. And Dyn. 9, (1992).
9. Hansmann, U., Okamoto, Y.: Prediction of Peptide Conformation by the Multicanonical Algorithm. arXiv: cond-mat/9303024 v1, (1993).
10. Okamoto, Y.: Protein Folding Problem as Studied by New Simulation Algorithms. Recent Research Developments in Pure & Applied Chemistry. Proc. Acad. Sci. USA 1987, 84, (1998) 6611-6615.
11. Garduño, R., Romero, D.: Heuristic Methods in conformational space search of peptides. J. Mol. Str. 308, (1994) 115 – 123.
12. Simons, K., Kooperberg, C., Huang, E., Baker, D.: Assembly of Protein Tertiary Structures from Fragments with Similar Local Sequences using Simulated Annealing and Bayesian Scoring Functions. J. Mol. Biol. 268, (1997) 209 – 225.
13. Pillardy, J., Czaplewski, C., Liwo, A., Lee, J., Ripoll, D., Kazmierkiewicz, R., Odziej, S., Wedemeyer, W., Gibson, K., Arnautova, Y., Saunders, J., Ye, Y., Scheraga, H.: Recent improvements in prediction of protein structure by global optimization of a potential energy function. PNAS vol 98. No. 5. (2000) 2329 – 2333.
14. Hiroyasu, T., Miki, M., Ogura, S., Aoi, K., Yoshida, T., Okamoto, Y., Dongarra, J.: Energy Minimization of Protein Tertiary Structure by Parallel Simulated Annealing using Genetic Crossover. Proceedings of 2002 Genetic and Evolutionary Computation Conference (GECCO 2002) Workshop Program. (2002) 49-51.

15. Vila, J., Ripoll, D., Scheraga, H.: Atomically detailed folding simulation of the B domain of staphylococcal protein A from random structures. PNAS vol 100. No. 25. 14812 – 14816.
16. Hung, L., Samudrala, R.: PROTINFO: Secondary and tertiary protein structure prediction. Nucleic Acids Research, Vol. 31, No. 13: (2003) 3296 – 3299.
17. Chen, W., Li, K., Liu, J.: The simulated annealing method applied to protein structure prediction. Third international conference on machine learning and cybernetics, Shanghai. (2004).
18. Liwo, A., Khalili, M., Scheraga, H.: Ab initio simulations of protein-folding pathways by molecular dynamics with the united-residue model of polypeptide chains. PNAS 2005, vol. 102. No. 7. (2004) 2362 – 2367.
19. Alves, R., Degréve, L., Caliri, A.: LMProt: An Efficient Algorithm for Monte Carlo Sampling of Protein Conformational Space. Biophysical Journal; ProQuest Medical Library. 87, 3. (2004).
20. Lee, J., Kim, S., Lee, J.: Protein structure prediction based on fragment assembly and parameter optimization. Biophisycal Chemestry 115 (2005) 209 – 214.
21. Kirkpatrick, S., Gelatt, C., Vecchi, M.: Optimization by simulated annealing. Science, Number 4598, 220, 4598. (1983) 671 – 680.
22. Cerny, V.: Thermo dynamical approach to the traveling salesman problem: An efficient simulation algorithm. Journal of Optimization Theory and Applications, 45(1). (1985) 41 – 51.
23. Sanvicente, H., Frausto, J.: A Method to Establish the Cooling Scheme in Simulated Annealing Like Algorithms. ICCSA 2004. Springer Verlag. LNCS, ISSN: 0302-9743. (2004).
24. Sanvicente, H.: Metodología de paralelización del ciclo de temperatura en algoritmos tipo recocido simulado. Tesis doctoral, ITESM Campus Cuernavaca, México. (2003).
25. Sanvicente, H., Frausto, J.: Optimización de los diámetros de las tuberías de una red de distribución de agua mediante algoritmos de recocido simulado. Ingeniería hidráulica en México. Vol XVIII, num. 1, (2003) 105 – 118.
26. Sanvicente, H., Frausto, J., Imperial, F.: Solving SAT Problems with TA Algorithms Using Constant and Dynamic Markov Chains Length. AAIM'05. Springer Verlag. LNCS, ISSN: 0302-9743, (2005).
27. Frausto, J., Sanvicente, H., Imperial, F.: ANDYMARK: An analytical method to establish dynamically the length of the Markov chain in simulated annealing for the satisfability problem. Lecture Notes in Computer Science, Springer Verlag. LNCS, ISSN:0302-9743, (2006).
28. Eisenmenger, F., Hansmann, U., Hayryan, S., Hu, C.: SMMP: A modern Package for Protein Simulation. Comp. Phys. Comm. 138, 192, (2001).
29. Eisenmenger, F., Hansmann, U., Hayryan, S., Hu, C.: An Enhanced Version of SMMP – Open source software package for simulation of proteins. Comp. Phys. Comm, (2006) 174-422,
30. Ramachandran, G. N., Ramakrishnan, C., Sasisekharan, V.: Stereochemistry of polypeptide chain configuration. J. Mol. Biol.7, (1963) 95 – 99.
31. Perez Joaquin O., Pazos Rodolfo, Velez Laura, Rodríguez Guillermo: Automatic Generation of Control Parameters for the Threshold Accepting Algorithm, LNCS 2313, Springer Verlag, MICAI (2002) 118-127.
32. Perez Joaquin O., Pazos R.A., Romero David, Santaolaya Rene., Rodríguez Guillermo, Sosa V:,: Adaptive and Scalable Allocation of Data-Objects in the Web, LCNS 2667 Springer Verlag, ICCSA (2003) 134-143.