

## Research Article

# Simulated Annealing with Previous Solutions Applied to DNA Sequence Alignment

**Ernesto Liñán-García and Lorena Marcela Gallegos-Araiza**

*Facultad de Sistemas, Universidad Autónoma de Coahuila, Saltillo, Coahuila, 25280 México, Mexico*

Correspondence should be addressed to Ernesto Liñán-García, ernesto\_linan\_garcia@uadec.edu.mx

Received 1 July 2012; Accepted 25 July 2012

Academic Editors: M. F. Abbod, M. Arif, and P. Trunfio

Copyright © 2012 E. Liñán-García and L. M. Gallegos-Araiza. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

A new algorithm for solving sequence alignment problem is proposed, which is named SAPS (Simulated Annealing with Previous Solutions). This algorithm is based on the classical Simulated Annealing (SA). SAPS is implemented in order to obtain results of pair and multiple sequence alignment. SA is a simulation of heating and cooling of a metal to solve an optimization problem. In order to select randomly a current solution, SAPS algorithm chooses a solution from solutions that have been previously generated within the Metropolis Cycle. This simple change has led to increase the quality of the solution to the problem of aligning genomic sequences with respect to the classical Simulated Annealing algorithm. The parameters of SAPS, for certain instances, are tuned by an analytical method, and some parameters have experimentally been tuned. SAPS has generated high-quality results in comparison with the classical SA. The instances used are specific genes of the AIDS virus.

## 1. Introduction

Sequence alignment is one of the most important and challenging problems in computational biology and bioinformatics [1, 2]. Finding the optimal alignment of a set of sequences is known as a NP-complete problem [3]. Alignment of sequences can be an important tool to measure the similarity of two or more sequences. Sequence Alignment is classified as a combinatorial optimization problem [4], which is solved by using computer algorithms. These algorithms lead to represent, to process, and to compare genetic information to determine evolutionary relationships among living beings [3]. The sequence alignment highlights areas of similarity among sequences. The similarities among sequences may indicate functional or evolutionary relationships among genes or proteins [5].

The problem of sequence alignment is to obtain the maximum alignment of a set of  $n$  genomic sequences, which is denoted as  $S = \{s_1, s_2, \dots, s_n\}$ ; each sequence of this set is formed by the alphabet  $\Sigma = \{A, C, G, T\}$ . The solution to this problem is represented by  $S' = \text{Max}(S)$ , which denotes a set  $S' = \{s'_1, s'_2, \dots, s'_n\}$  with the alphabet  $\Sigma' = \Sigma \cup \{-\}$ .  $S'$  represents the optimal alignment of  $S$ .

Exact algorithms have been applied to solve the sequence alignment problem. For example, dynamic programming has been one of the most used to solve the sequence alignment problem [6, 7]. The disadvantage of using exact algorithms is that these generate optimal solutions for small problems, but for large problems, exact algorithms become inefficient. For this reason, several metaheuristic methods have been designed to obtain suboptimal alignments. Metaheuristics have also been applied to solve this problem [8], for example, Ant Colony Algorithm [9], Simulated Annealing [10, 11], Genetic Algorithms [12], among others. The disadvantage is that metaheuristics do not guarantee optimal solutions, but solutions generated can be very close to optimal solution in a reasonable processing time.

The proposed algorithm is a modified version of classical Simulated Annealing. SAPS includes a new way to select a current solution after the Metropolis Cycle is finished. In general, SAPS generates better solutions to sequence alignment problem than the classical Simulated Annealing. SAPS was tested with different genes of AIDS virus.

This paper is organized as follows: in Section 2, classical simulated annealing algorithm is described. In Section 3, the SASP algorithm is explained in detail. In Section 4,

```

1: Setting initial and final temperatures
2: Create  $S_{\text{current}}$  from Initial solution  $S_{\text{initial}}$ 
3:  $T = T_{\text{initial}}$ 
4: While ( $T > T_{\text{final}}$ ) do // Temperature Cycle
5:   While (stop criteria) // Metropolis Cycle
6:     Create  $S_{\text{new}}$  using a perturbation to  $S_{\text{current}}$ 
7:     Obtain difference similarity between  $S_{\text{new}}$  and  $S_{\text{current}}$ 
8:     If (difference  $\leq 0$ ) then
9:       Accept  $S_{\text{new}}$ 
10:    else
11:      Boltzmann probability =  $\exp(-\text{difference}/T)$ 
12:      If (Boltzmann probability) > random (0,1) then
13:        Accept  $S_{\text{new}}$ 
14:      end if
15:    end if
16:  end while
17:  Decrease  $T$ 
18: end while

```

ALGORITHM 1: Pseudocode of classical Simulated Annealing.

```

1: Tune initial temperature ( $T_{\text{initial}}$ )
2: Tune final temperature ( $T_{\text{final}}$ )
3: Setting cooling factor
4: Create  $S_{\text{current}}$  from Initial solution  $S_{\text{initial}}$ 
5: Calculate similarity of  $S_{\text{current}}$ 
6:  $T = T_{\text{initial}}$ 
7: Setting  $L_{\text{cm}}$ 
8: While ( $T > T_{\text{final}}$ ) do
9:    $n = 1$ 
10:  While ( $L_{\text{cm}} > n$ )
11:    Create  $S_{\text{new}}$  adding or removing gaps to  $S_{\text{current}}$ 
12:    Calculate similarity of  $S_{\text{new}}$ 
13:    Obtain difference similarity between  $S_{\text{new}}$  and  $S_{\text{current}}$ 
14:    If (difference  $\leq 0$ ) then
15:       $S_{\text{current}} = S_{\text{new}}$ 
16:      If similarity ( $S_{\text{new}}$ ) > similarity ( $S_{\text{better}}$ ) then
17:         $S_{\text{better}} = S_{\text{current}}$ 
18:      end if
19:    else
20:      Boltzmann probability =  $\exp(-\text{difference}/T)$ 
21:      If (Boltzmann probability) > random(0,1) then
22:         $S_{\text{current}} = S_{\text{new}}$ 
23:      end if
24:    end if
25:     $n = n + 1$ 
26:  end while
27:  Decrease  $T$ 
28:  Increase  $L_{\text{cm}}$ 
29: end while

```

ALGORITHM 2: Pseudo code SA applied to Sequence Alignment.

the analytical tuning method is described. In Section 5, the implementation of the SASP is detailed. In Section 6, the experimentation and results are described. Finally, Section 7 discusses the conclusions.

## 2. Classical Simulated Annealing

The classical Simulated Annealing is an algorithmic process that simulates the gradual metal cooling for crystallization. This algorithm usually starts at high value of temperature,

```

1: Tune initial temperature ( $T_{\text{initial}}$ )
2: Tune final temperature ( $T_{\text{final}}$ )
3: Setting the factor cooling
4: Setting  $S_{\text{current}}$  with  $S_{\text{original}}$ 
5: Add  $S_{\text{original}}$  to  $SS_{\text{betters}}$  and to  $S_{\text{better}}$ 
6: Calculate similarity of  $S_{\text{current}}$ 
7:  $T = T_{\text{initial}}$ 
8: Setting  $L_{\text{cm}}$ 
9: While ( $T > T_{\text{final}}$ ) do
10:    $n = 1$ 
11:   While ( $L_{\text{cm}} > n$ )
12:     Create  $S_{\text{new}}$  adding or removing gap to  $S_{\text{current}}$ 
13:     Calculate similarity of  $S_{\text{new}}$ 
14:     Obtain difference of similarity between  $S_{\text{new}}$  and  $S_{\text{current}}$ 
15:     If (difference  $\leq 0$ ) then
16:        $S_{\text{current}} = S_{\text{new}}$ 
17:       If similarity ( $S_{\text{new}}$ ) > similarity ( $S_{\text{better}}$ ) then
18:          $S_{\text{better}} = S_{\text{current}}$ 
19:         Add  $S_{\text{better}}$  to  $SS_{\text{betters}}$ 
20:       end if
21:     else
22:       Boltzmann probability =  $\exp(-\text{difference}/T)$ 
23:       If (Boltzmann probability) >  $\text{random}(0,1)$  then
24:          $S_{\text{current}} = S_{\text{new}}$ 
25:       end if
26:     end if
27:      $n = n + 1$ 
28:   end while
29:   Decrease  $T$ 
30:   Increase  $L_{\text{cm}}$ 
31:   Randomly Choose  $S_{\text{current}}$  of  $SS_{\text{betters}}$ ,  $S_{\text{better}}$  or  $S_{\text{original}}$ 
32: end while

```

ALGORITHM 3: Pseudo code of SAPS.

and then this parameter is decreased until a final temperature is reached. The final temperature typically is very close to zero ( $T_{\text{final}} \approx 0$ ) [13, 14]. Through a cooling function, the temperature value is decreased from the initial temperature to the final temperature. There are cooling functions that have been used in the simulated annealing algorithm [15–18]; the most common cooling function is defined by  $T_{k+1} = \alpha T_k$ . This function decreases the temperature value by a  $\alpha$  factor, which does a range of  $0.70 \leq \alpha < 1.0$ . A gradual cooling is applied when  $\alpha$  is very close to 1, and a fast cooling is applied when  $\alpha$  is very close to 0.70.

The classical Simulated Annealing has two cycles; the first cycle is named Cycle of Temperature. Into this cycle, value temperature is decreased by a cooling function. The second cycle is named Metropolis Cycle, and it is applied to generate, to accept, or to reject solutions for the problem to be optimized. Algorithm 1 shows the pseudo code of the classical Simulated Annealing. The initial and final temperature values are set (see line 1). These values are obtained by an analytical (see Section 4) or experimental way. It is recommended that the initial temperature is as high as possible, and the final temperature is as close to zero. The initial solution  $S_{\text{initial}}$  of the problem to be optimized is created (see line 2). The current solution  $S_{\text{current}}$  is set

to  $S_{\text{initial}}$ . Set  $T$  to initial temperature (see line 3). The temperature cycle is executed from the initial temperature to the final temperature (see lines 4–18). The Metropolis Cycle gets started (see lines 5–16). This cycle takes a number of times specified in the stop criterion. A new solution  $S_{\text{new}}$  is created within the Metropolis Cycle by creating a small perturbation to the current solution  $S_{\text{current}}$  (see line 6). The difference between these two solutions ( $S_{\text{new}}$  and  $S_{\text{current}}$ ) is obtained. If the difference is less or equal than zero (see line 8), the new solution is accepted (see line 9). If the difference is greater than zero, the Boltzmann probability is calculated (see line 11). If the Boltzmann probability is higher than a random value between 0 and 1 (see line 12) then the new solution is accepted (see line 13). After the Metropolis Cycle is completed, the temperature value is decreased (see line 17).

Algorithm 2 shows the pseudo code of the SA, which is applied to obtain solutions to the problem of aligning two or more genomic sequences. The Simulated Annealing algorithm is modified then it can be implemented to solve the problem of alignment sequence. The values of initial and final temperatures are tuned by using an analytical method (see lines 1-2). The cooling factor value  $\alpha$  is set to a value very close to 1 ( $\alpha \approx 1$ ) (see line 3). The current solution  $S_{\text{current}}$  is set to the original solution  $S_{\text{initial}}$  (see line 4). The similarity

TABLE 1: HIV genes of human and simian.

Gen	Number of bases (human)	Number of bases (simian)
pol	3011	3179
env	2570	2564
gag	1502	1532
vif	578	644
nef	371	791
rev	351	351
tat	259	304
vpu	248	245
vpr	237	306

TABLE 2: Values of parameters.

Algorithm	Gen	$T_{\text{initial}}$	$T_{\text{final}}$	$\Delta Z_{\text{max}}$	$\gamma$
SA	env	2,639.71	0.43	26.53	1.06
SAPS	env	1,269.61	0.43	12.76	1.07
SA	gag	1,211.90	0.43	12.18	1.07
SAPS	gag	365.16	0.43	3.67	1.08
SA	nef	883.55	0.43	8.88	1.07
SAPS	nef	787.04	0.43	7.61	1.07
SA	pol	369.14	0.43	3.71	1.08
SAPS	pol	436.80	0.43	4.39	1.08
SA	rev	5,726.18	0.43	57.55	1.06
SAPS	rev	6,651.52	0.43	66.85	1.06
SA	tat	6,151.04	0.43	61.82	1.06
SAPS	tat	6,594.80	0.43	66.28	1.06
SA	vif	666.64	0.43	6.70	1.07
SAPS	vif	1,349.21	0.43	13.56	1.07
SA	vpr	1,248.71	0.43	12.55	1.07
SAPS	vpr	1,784.02	0.43	17.93	1.07
SA	vpu	4,556.07	0.43	45.79	1.06
SAPS	vpu	1609.90	0.43	16.16	1.07

of this solution is calculated by comparing base by base (see line 5). The variable  $T$  is set to the initial temperature  $T_{\text{initial}}$  (see line 6). The Metropolis Cycle length is set to an initial value  $L_{\text{cm}}$  (see line 7). This cycle has an increasing length, at high temperature, it has a low value, and it is increased as the temperature is decreased. The length of Metropolis Cycle is increased by a factor  $\gamma$ , where  $\gamma$  must be greater than 1. Temperature cycle is executed (see lines 8–29) with a logic condition that  $T$  is greater than  $T_{\text{final}}$ . Within this cycle, the variable  $n$  is updated with value 1 (see line 9), and within the metropolis cycle, this variable is incremented ( $n = n + 1$ ) (see line 25).

The Metropolis Cycle is executed (see lines 10–26). At the end of the Metropolis Cycle, the temperature is decreased (see line 27), and the Metropolis Cycle length is increased (see line 28). Within the Metropolis Cycle, new solutions  $S_{\text{new}}$  are generated by modifying the current solution  $S_{\text{current}}$ . This is done by adding or removing gaps into DNA sequences (see line 11). The similarity of new solutions is calculated (see line 12), and the difference of similarities between  $S_{\text{new}}$  and  $S_{\text{current}}$  is calculated (see line 13). This difference

is denoted by  $\Delta S = S_{\text{current}} - S_{\text{new}}$ . The new solutions are accepted when these are better than current solutions, so current solutions are replaced by new solutions (see line 15). When new solutions are of low quality (worse solutions) than current solutions, then new solutions are accepted using the Boltzmann probability (see line 22). This probability is directly related to the current value of the temperature and the quality difference between  $S_{\text{new}}$  and  $S_{\text{current}}$ . The Boltzmann probability is calculated by the following equation  $P(S_{\text{new}}) = e^{(-\Delta S/T)}$ . As the temperature value is decreased, the probability of  $P(S_{\text{new}})$  is decreased, which is of range  $0 < P(S_{\text{new}}) < 1$ .

### 3. Simulated Annealing with Previous Solutions

In order to generate high-quality solutions to sequence alignment, the classical SA was modified, so the SAPS algorithm is a modified version of the classical SA. After the Metropolis Cycle execution is done, the selection of a current solution  $S_{\text{current}}$  is done. During the execution of Metropolis Cycle, the best solutions are stored in a set named  $SS_{\text{betters}}$ .

The best of all solutions created in this cycle is stored in  $S_{\text{better}}$ . The original sequence is stored in  $S_{\text{original}}$ . After the Metropolis cycle is finished, a current solution  $S_{\text{current}}$  is randomly selected from  $S_{\text{original}}$ ,  $S_{\text{better}}$ , or  $SS_{\text{betters}}$ . So  $S_{\text{current}} \in \{S_{\text{original}}, S_{\text{better}}, SS_{\text{betters}}\}$ .

The Metropolis Cycle length of SAPS is growing, which ranges from an initial  $L_{\text{initial}}$  value to a final  $L_{\text{final}}$  value. At high temperature,  $L_{\text{initial}}$  is set to a small value and as the temperature value is increased, the value of the Metropolis Cycle length is increased until  $L_{\text{final}}$ . when  $T_{\text{final}}$  is reached,  $L_{\text{final}}$  is reached too. Thus, an increasing number of solutions are created as the temperature is decreased. At high temperatures, a small number of solutions are created and as the temperature is decreased, the number of solutions is increased with a factor  $\gamma$ , where  $\gamma > 1$ .

Algorithm 3 shows the pseudo code of SAPS, some lines of code were added to SA, for example, at line 5,  $S_{\text{better}}$  and  $SS_{\text{betters}}$  are set with  $S_{\text{original}}$ . At line 19,  $S_{\text{better}}$  is added to  $SS_{\text{betters}}$ . At line 31,  $S_{\text{current}}$  is chosen from  $S_{\text{original}}$ ,  $S_{\text{better}}$ , or  $SS_{\text{betters}}$ .

### 4. Analytical Tuning Method

Some parameters of SAPS are tuned by the analytical method [19–22]. For example, in order to calculate the initial temperature, the maximum deterioration (defined by  $\Delta Z_{\text{max}}$ ) of the instance is applied. The probability of accepting a solution  $S$  is applied at high temperature. On other hand, the final temperature is calculated by applying the minimum deterioration (defined by  $\Delta Z_{\text{min}}$ ) of the instance and the probability of accepting a Solution  $S$  at low temperature.

The analytical tuning based on Boltzmann distribution can be helpful for setting up the initial temperature [21]. The probability of accepting any new solution  $S_{\text{new}}$  is very close to 1 ( $P(S_{\text{new}}) \approx 1$ ) at high temperatures, so the deterioration of cost function is maximal. The initial temperature ( $T_{\text{initial}}$ )

TABLE 3: Results of quality solutions.

Gen	Classical SA		SASP	
	Average (%)	Standard deviation	Average (%)	Standard deviation
pol	29.70	0.31	30.35	0.24
env	47.41	5.59	61.60	4.31
gag	40.33	3.30	47.20	1.01
vif	33.71	2.72	39.40	0.66
nef	34.84	1.07	36.56	0.44
rev	98.00	0.00	98.00	0.00
tat	98.00	0.00	98.00	0.00
vpu	52.40	10.60	77.65	8.63
vpr	37.37	3.38	50.24	4.26

TABLE 4: Results of processing time.

Gen	Processing time SA (s)	Processing time SAPS (s)
pol	171	224
env	149	142
gag	90	84
vif	19	20
nef	20	18
rev	34	19
tat	22	15
vpu	14	11
vpr	13	11

is associated with the maximum deterioration admitted and the defined acceptance probability  $P(S_{\text{new}})$ .

Let  $S_{\text{current}}$  be the current solution and  $S_{\text{new}}$  a new proposed one, and  $Z(S_{\text{current}})$  and  $Z(S_{\text{new}})$  are the costs associated to  $S_{\text{current}}$  and  $S_{\text{new}}$ , respectively; the maximum and minimum deteriorations are expressed as  $\Delta Z_{\text{max}}$  and  $\Delta Z_{\text{min}}$ , respectively. Then, the  $P(\Delta Z_{\text{max}})$  probability of accepting a new solution  $S_{\text{new}}$  with the maximum deterioration is defined by  $(P(\Delta Z) = \exp(-\Delta Z/T))$ . This equation basically is the Boltzmann Distribution, which is applied for calculating the  $T_{\text{initial}}$ . This temperature value is defined by  $T_{\text{initial}} = -\Delta Z_{\text{max}}/\ln(P(\Delta Z_{\text{max}}))$ . Similarly, the final temperature ( $T_{\text{final}}$ ) is established according to the probability of accepting a new solution  $S_{\text{new}}$  with the minimum deterioration. The equation to calculate the final temperature is defined by  $T_{\text{final}} = -\Delta Z_{\text{min}}/\ln(P(\Delta Z_{\text{min}}))$ .

There are other parameters of SAPS that are calculated by applying a particular cooling function; for example, the Metropolis Cycle length is calculated by applying  $T_{k+1} = \alpha T_k$ . The incremental factor of this cycle is also calculated and defined by  $\gamma$ .

The analytical method determines the Metropolis Cycle length  $L_k$  with a simple Markov model [22]; 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  $T_{\text{initial}}$  and let  $L_{\text{max}}$  be the maximum Metropolis Cycle length.

Let  $T_k$  be decreased by the cooling function ( $T_{k+1} = \alpha T_k$ ), and the  $L_{k+1}$  be calculated by the follow equation  $L_{k+1} = \gamma L_k$ , where  $\gamma$  is the rate of increment of Metropolis Cycle ( $>1$ ); so  $L_{k+1} > L_k$  and  $L_1$  have an initial value, and the last  $L_f$  Metropolis Cycle is equal to  $L_{\text{max}}$ . The functions  $T_{k+1} = \alpha T_k$  and  $L_{k+1} = \gamma L_k$  are applied successively in Simulated Annealing from  $T_{\text{initial}}$  to  $T_{\text{final}}$ ; consequently,  $T_n$  and  $L_{\text{max}}$  are obtained by  $T_n = \alpha^n T_{\text{initial}}$  and  $L_{\text{max}} = \gamma^n L_1$ , respectively.  $n$  is the step number from  $T_{\text{initial}}$  to  $T_{\text{final}}$ ; so we can get the  $n$  and  $\gamma$  as follows:  $n = (\ln(T_{\text{final}}) - \ln(T_{\text{initial}}))/\ln(\alpha)$  and  $\gamma = \exp((\ln(L_{\text{max}}) - \ln(L_1))/n)$ .

## 5. Implementation

SASP was tested with all of the most HIV virus genes of human and simian. The nine genes of the human virus were compared with the nine genes of simian virus; for example, the gen named “env” of HIV human was aligned with the gen “env” of HIV simian, the gen named “gag” of HIV human was aligned with the gen “gag” of HIV simian, and so successively. The information of the virus genes is shown in Table 1. The parameters  $T_{\text{initial}}$ ,  $T_{\text{final}}$ ,  $\Delta Z_{\text{max}}$ , and  $\gamma$  are tuned by analytical method. The factor  $\gamma$  is higher than 1, and it is very close to 1. The values of these parameters are shown in Table 2. In this table, the values of initial temperatures are high; these values are related to the maximum deterioration and the probability of accepting solutions at high temperatures. It is observed that the final temperature has a value very close to zero (0.43); this is because the minimum deterioration is equal to 1.0. The parameters  $L_1$  and  $L_{\text{max}}$  have the values 2, and 300, respectively.

## 6. Experimentation and Results

In Table 3, the results of the experiments are shown. The information shown is the average similarity and the standard deviation of the genes of both viruses (HIV Human and HIV Simian). The results show that the average obtained by SASP is of better quality than the average obtained by the classical SA. Table 4 shows that the SAPS processing time generally is better than the processing time of SA.

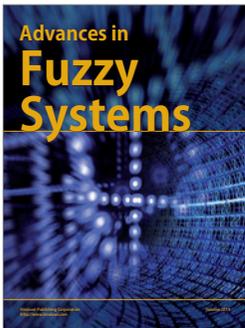
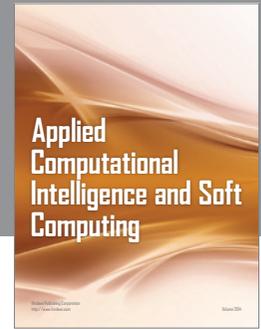
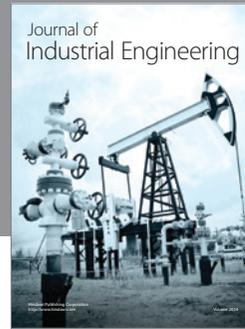
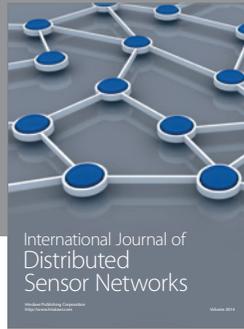
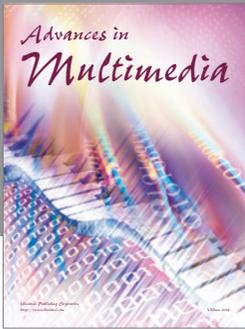
## 7. Conclusions

In this paper, a new approach is to make efficient the classical Simulated Annealing algorithm proposed to solve the problem of aligning genomic sequences. This approach is called SAPS. After completing the Metropolis Cycle, a current solution is selected randomly from the best solutions' set, the best solution and the initial solution. This change in the classical simulated annealing resulted in an improved efficiency to solve the problem of aligning sequences. The parameters of the algorithms SA and SAPS were tuned using a tuning method, specifically the initial temperature, final temperature, and Metropolis Cycle length.

This approach to tune the parameters depends directly on the instance to test. With a preprocessing of the instance, the minimum and maximum deteriorations are calculated. With these values and the probability of acceptance, the initial and final temperatures are calculated.

## References

- [1] R. M. Karp, "Mapping the genome: some combinatorial problems arising in molecular biology," in *Proceedings of the 25th Annual ACM Symposium on the Theory of Computing*, pp. 278–285, May 1993.
- [2] E. S. Lander, R. Langridge, and D. M. Saccocio, "Mapping and interpreting biological information," *Communications of the ACM*, vol. 34, no. 11, pp. 33–39, 1991.
- [3] L. Wang and T. Jiang, "On the complexity of multiple sequence alignment," *Journal of Computational Biology*, vol. 1, no. 4, pp. 337–348, 1994.
- [4] C. H. Papadimitriou and K. Steiglitz, *Combinatorial Optimization: Algorithms and Complexity*, Dover Publications, Mineola, NY, USA, 1998.
- [5] J. Setubal and J. Meidanis, *Introduction to Computational Molecular Biology*, PWS Publishing, 1997.
- [6] O. Gotoh, "An improved algorithm for matching biological sequences," *Journal of Molecular Biology*, vol. 162, no. 3, pp. 705–708, 1982.
- [7] S. B. Needleman and C. D. Wunsch, "A general method applicable to the search for similarities in the amino acid sequence of two proteins," *Journal of Molecular Biology*, vol. 48, no. 3, pp. 443–453, 1970.
- [8] I. Ö. Bucak and V. Uslan, "An analysis of sequence alignment: Heuristic algorithms," in *Proceedings of the 32nd Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC'10)*, pp. 1824–1827, September 2010.
- [9] L. Chen, L. Zou, and J. Chen, "An efficient ant colony algorithm for multiple sequences alignment," in *Proceedings of the 3rd International Conference on Natural Computation (ICNC '07)*, pp. 208–212, August 2007.
- [10] J. Kim, S. Pramanik, and M. J. Chung, "Multiple sequence alignment using simulated annealing," *Computer Applications in the Biosciences*, vol. 10, no. 4, pp. 419–426, 1994.
- [11] S.-M. Chen and C.-H. Lin, "Multiple DNA sequence alignment based on genetic simulated annealing techniques," *International Journal of Information and Management Sciences*, vol. 18, no. 2, pp. 97–111, 2007.
- [12] C. Notredame, D. G. Higgins, and J. Heringa, "T-coffee: a novel method for fast and accurate multiple sequence alignment," *Journal of Molecular Biology*, vol. 302, no. 1, pp. 205–217, 2000.
- [13] V. Cerny, "Thermodynamical approach to the traveling salesman problem: an efficient simulation algorithm," *Journal of Optimization Theory and Applications*, vol. 45, no. 1, pp. 41–51, 1985.
- [14] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi, "Optimization by simulated annealing," *Science*, vol. 220, no. 4598, pp. 671–680, 1983.
- [15] E. Aarts and J. Korst, *Simulated Annealing and Boltzmann Machines: a Stochastic Approach to Combinatorial Optimization And Neural Computing*, Wiley-Interscience Series in Discrete Mathematics and Optimization, John Wiley & Sons, Chichester, UK, 1989.
- [16] L. Ingber, "Simulated annealing: practice versus theory," *Mathematical and Computer Modelling*, vol. 18, no. 11, pp. 29–57, 1993.
- [17] U. Kjærulff, "Optimal decomposition of probabilistic networks by simulated annealing," *Statistics and Computing*, vol. 2, no. 1, pp. 7–17, 1992.
- [18] P. J. Van Laarhoven and E. H. L. Aarts, *Simulated Annealing: Theory and Applications*, Kluwer Academic Publishers, 1987.
- [19] J. Frausto-Solis, E. F. Román, D. Romero, X. Soberon, and E. Liñán-García, "Analytically tuned simulated annealing applied to the protein folding problem," *Lecture Notes in Computer Science*, vol. 4488, no. 2, pp. 370–377, 2007.
- [20] J. Frausto-Solis, X. Soberon-Mainero, and E. Liñán-García, "MultiQuenching annealing algorithm for protein folding problem," *Lecture Notes in Computer Science*, vol. 5845, pp. 578–589, 2009.
- [21] J. Frausto-Solis, 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," *Lecture Notes in Computer Science*, vol. 4247, pp. 269–276, 2006.
- [22] H. Sanvicente-Sánchez and J. Frausto-Solis, "A method to establish the cooling scheme in simulated annealing like algorithms," *Lecture Notes in Computer Science*, vol. 3045, pp. 755–763, 2004.




**Hindawi**

Submit your manuscripts at  
<http://www.hindawi.com>

