3

Enhanced Genetic Algorithm for Protein Structure Prediction based on the HP Model

Nashat Mansour[1], Fatima Kanj[1] and Hassan Khachfe[2]
[1]*Department of Computer Science and Mathematics, Lebanese American University,*
[2]*Department of Biology and Biomedical Sciences, Lebanese International University,*
Lebanon

1. Introduction

Proteins are organic compounds that are made up of combinations of amino acids and are of different types and roles in living organisms. Initially a protein is a linear chain of amino acids, ranging from a few tens up to thousands of amino acids. Proteins fold, under the influence of several chemical and physical factors, into their 3-dimensional structures which determine their biological functions and properties. Misfolding occurs when the protein folds into a 3D structure that does not represent its correct native structure, which can lead to many diseases such as Alzheimer, several types of cancer, etc… (Prusiner, 1998). Hence, predicting the native structure of a protein from its primary sequence is an important and challenging task especially that this protein structure prediction (PSP) problem is computationally intractable.

The primary structure of a protein is a linear sequence of amino acids connected together via peptide bonds. Proteins fold due to hydrophobic effect, Vander Waals interactions, electrostatic forces, and Hydrogen bonding (Setubal & Meidanis, 1997). The secondary structures are three-dimensional structures characterized by repeating bonding patterns of α-helices and β-strands. Proteins further fold into a tertiary structure forming a bundle of secondary structures and loops. Furthermore, the aggregation of tertiary structure regions of separate protein sequences leads to quaternary structures. These structures are depicted in Fig. 1 (Rylance, 2004).

Computational approaches for PSP can be classified as: homology modeling, threading, and *ab initio* methods (Floudas, 2007). Approaches in the first two groups use known protein structures from protein data banks (PDB). Approaches in the third group solely rely on the given amino acid sequence. A survey of PSP approaches appeared in Sikder and Zomaya (2005). Homology modeling uses sequences of known structures in the PDB to align with the target protein's sequence for which the 3D structure is to be predicted (Kopp & Schwede, 2004; Notredame, 2002; Pandit et al., 2006).

Threading is similar to homology modeling. But, instead of finding similar sequences to deduce the native conformation of the target protein, threading assumes that the target structure is similar to another existing structure, which should be searched for (Lathrop et al., 1998; Jones 1998; Skolnick et al., 2004). The threading of a sequence to a fold is evaluated by either environment-based or knowledge-based mean-force-potentials derived from the PDB.

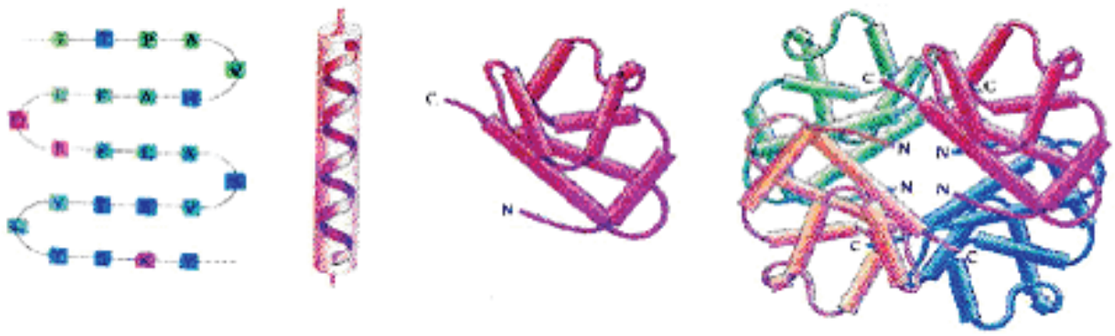


Fig. 1. Primary, secondary, tertiary and quaternary structures

Ab initio approaches do not rely on known structures in the PDB. Instead, they predict the 3D structure of proteins given their primary sequences. The underlying strategy is to find the best possible structure based on a chosen energy function. Based on the laws of physics, the most stable structure is the one with the lowest possible energy (Anfinsen, 1973). The main challenge of these approaches is to search for the most stable structure in a huge search space. Models such as the Hydrophobic-Polar (HP) models have been developed in order to reduce the size of the search space. Other models use the detailed representation of proteins with all the corresponding atoms, based on force fields.

Force field models use an energy objective function that evaluates the structure of a protein. This function attempts to represent the actual physical forces and chemical reactions occurring in a protein. Atoms are modeled as points in 3D with zero volume but with finite mass and charge, and bonds among atoms are modeled as Newtonian springs. The energy function is usually based on molecular mechanics and force fields components such as bond lengths, bond angles, dihedral angles, van der Waals interactions, electrostatic forces, etc.... Examples of force-field based methods are found in: Schulze-Kremer (2000), Klepeis and Floudas (2003), Datta et al. (2008), Li et al. (2006), Srinivasan and Rose (2000) and Mansour et al. (2009).

The HP model simplifies the protein by assigning each amino acid to be a point in a 2D or 3D lattice (Unger and Moult, 1993b) which is either hydrophobic (H) or polar (P) (Dill, 1985). According to this model, the most stable structure is the one with the hydrophobic amino acids lying in its core. The underlying concept is that hydrophobic amino acids tend to avoid contact with the solvent and hence tend to move inside the structure whereas the polar ones remain on the outside. The main energy function used in this model is the total number of the hydrophobic interactions between the amino acids and the goal is to have a lattice with minimum energy, i.e., with maximum number of H-H contacts. The objective is to fold a string of Hs and Ps on a three dimensional coordinates system in a self-avoiding walk. Candidate solutions are represented as a string of characters (b, f, u, d, l, r) representing the six directions: backward, forward, up, down, left and right.

Despite the reduction in search space, the problem of predicting protein structures in the HP model is still intractable (Unger and Moult, 1993a). Hence, heuristic and meta-heuristics algorithms have been reported for finding good sub-optimal solutions. In the early nineties, Unger and Moult (1993b) developed a genetic algorithm (GA) combined with the Monte Carlo method to fold proteins on a two dimensional lattice and they extended their work

later to a 3D lattice. Later, a standard GA was developed (Patton et al., 1995) and it outperformed that of Unger and Moult (1993b) by reaching higher number of hydrophobic contacts with less number of energy evaluations. More recently, Johnson et al. (2006) proposed a genetic algorithm with a backtracking method to resolve the collision problem.

Heuristic methods based on assumptions about the folding mechanism were proposed, such as the hydrophobic zipper (Dill et al., 1993), the constrained hydrophobic core construction algorithm (Yue & Dill, 1995), and the contact interactions method (Toma & Toma, 1996). A branch and bound algorithm was developed by Chen and Huang (2005). The algorithm evaluates the importance of every possible position of the hydrophobic amino acids and only those promising locations are preserved for more branching at every level.

Methods based on the Monte Carlo (MC) algorithm have also been proposed: the MC based growth algorithm (Hsu et al., 2003), and the evolutionary MC algorithm (Liang & Wong, 2001). Further, a modified particle swarm optimization algorithm for the protein structure prediction problem in the 2D toy model was proposed by Zhang and Li (2007). An Ant Colony Optimization algorithm was proposed by Shmygelska and Hoos (2005) for both 2D and 3D lattice models.

In this paper, we present a genetic algorithm for the protein structure prediction problem based on the cubic 3D hydrophobic polar (HP) model. This algorithm is enhanced with heuristics that repair infeasible outcomes of the crossover operation and ensure that the mutation operation leads to fitter and feasible candidate solutions. The PSP solutions produced by this GA are experimentally evaluated by comparing them with previously published results.

The rest of the paper is organized as follows. Section 2 describes the GA algorithm for the PSP problem. Section 3 presents our experimental results. Section 4 concludes the paper.

2. Enhanced genetic algorithm

Genetic Algorithms simulate the concept of natural evolution (Holland, 1975). They are based on the operations of population reproduction and selection for the purpose of achieving optimal results. Through artificial evolution, successive generations search for fitter adaptations in order to solve a problem. Each generation consists of a population of chromosomes, and each chromosome represents a possible solution. The Darwinian principle of reproduction and survival of the fittest and the genetic operations of recombination and mutation are used to create a new offspring population from the current population. The process is repeated for many generations with the aim of maximizing the fitness of the chromosomes. In the following subsections, we describe an enhanced genetic algorithm (EGA) that is adapted for solving the protein structure prediction problem. A flowchart of this EGA is given in Fig. 2.

2.1 Chromosomal representation

A Chromosome in the population is encoded as an array of length N-1, where N is the number of amino acids in the respective protein. Each element in the array represents the position X_d of the corresponding amino acid d with respect to the preceding one and its value can be one of six characters {b, f, u, d, l, r}. These characters represent the following six directions, respectively {backward, forward, up, down, left, right}. In Fig. 3, a sample 3D

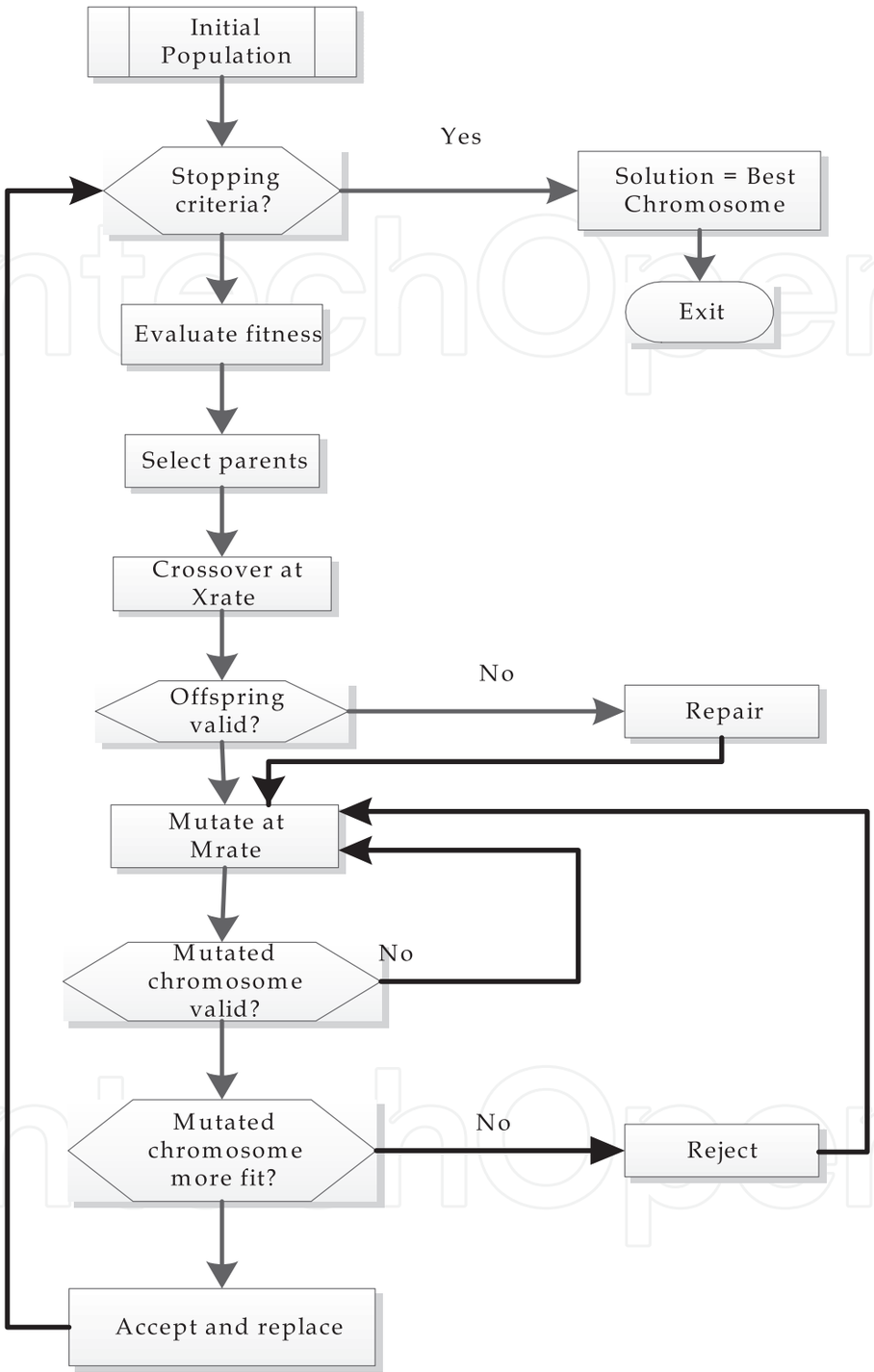


Fig. 2. Flowchart for the EGA.

structure is illustrated. This structure is represented as bbburdfulurrur, which is an array of directions (of length 14) is representing a protein sequence containing 15 amino acids where the first amino acid is omitted since it is the reference point. The gray balls represent the polar amino acids whereas the black balls represent the hydrophobic ones.

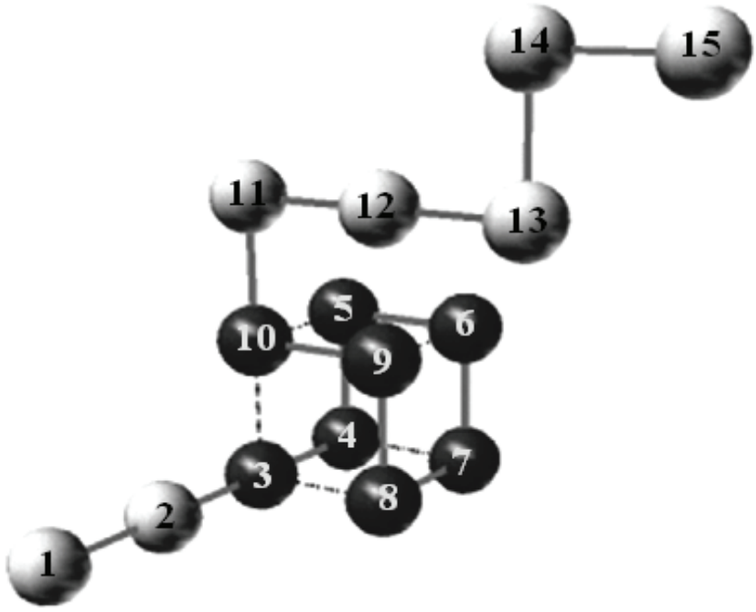


Fig. 3. A candidate PSP solution

GA's population is an array of POP Chromosomes. The initial population of PSP solutions is randomly generated. That is, each position X_d of amino acid d (d = 1, 2,…, N-1) is assigned a random value for the chromosome i (i = 0, 1, …., POP-1). In our implementation, we use POP = 200.

2.2 Fitness function

The fitness function is given by the sum of the hydrophobic contacts (H-H) between non adjacent amino acids. Since we are using the cubic lattice, the maximum number of possible contacts per amino acid is four, except for the first and last amino acids, which might have up to five contacts. The goal is to maximize the fitness value of the chromosomes to obtain protein structures with the most compact hydrophobic core and, thus, with the lowest energy. For example, in Fig. 3, the fitness value of the displayed structure is 5. The hydrophobic contacts are displayed in dotted lines and there are five of them between the following pairs of hydrophobic amino acids: (3, 8), (3, 10), (4, 7), (5, 10) and (6, 9).
Evaluating the fitness value of a chromosome is simple. Every hydrophobic amino acid in the sequence is checked for any non-adjacent (not connected by a bond) hydrophobic amino acids in the six positions around it on the lattice, at a distance 1, and the number of these amino acids is counted.

2.3 Reproduction scheme and convergence

The whole population is considered a single reproduction unit within which tournament selection is performed. In this selection method, chromosomes are compared in a "tournament," with the higher-fitness chromosome being more likely to win. The tournament process is continued by sampling, with replacement, from the original population until a full complement of parents has been chosen. We use the binary tournament method, where we randomly select two pairs of chromosomes (i.e. 2 tournaments with 2 members each) and choose as the two parents the winner chromosomes

that have the higher fitness value from each pair. The tournament selection method is chosen since it is not very sensitive to the scaling of the fitness function.
To ensure that good candidate solutions are preserved, the best-so-far protein structure is saved. Convergence is detected when the best-so-far structure does not change its fitness value for 10 generations. After convergence, the best-so-far protein structure becomes the final PSP solution found.

2.4 Genetic operators and acceptance heuristics

The genetic operators employed in GA are 1-point crossover and mutation at the rates 0.5 and 0.1, respectively. Crossover is applied to pairs of chromosomes provided by tournament selection, where position k along the chromosome is selected at random between 1 and N, and all genes between k and N are swapped to create two new chromosomes. That is, the amino acids that lie between k and N will have their location in space (represented by the direction with respect to the preceding amino acid) exchanged. This may lead to collisions which occur if two or more amino acids lie at the same point on the cubic 3D lattice. If collisions occur, the protein structure becomes invalid. Invalid structures are repaired using a heuristic repair function, if possible; otherwise, the initial structure is restored. The repair function detects a collision and tries to repair it locally by finding an alternative empty location for the amino acid which caused the collision. If no such location is available, then it searches for previous amino acids whose locations can be modified. If modifications are performed for more than three amino acids or if none can be modified, then it is assumed that the structure cannot be repaired and the pre-crossover protein structure is returned.
Mutation is applied to randomly selected genes; that is the position of amino acid, d, in the 3D cubic lattice is changed to another position randomly selected from {b, f, u, d, l, r}. Eventually, the new offspring population replaces the parent population. But, if this mutation leads to an invalid protein structure, it is rejected and mutation will be repeated until a valid structure is found. Furthermore, the fitness value of this valid structure is computed. If the fitness of the mutated structure is higher than that of the initial pre-mutation structure, the mutated structure is accepted; otherwise, the initial structure is restored.

3. Experimental results

In this section, we report the empirical results of the proposed GA and compare them to those of published techniques: one by Patton et al. (1995), which proposed a standard genetic algorithm for this problem and reported better results than those achieved by Unger and Moult (1993b); the second is by Johnson et al. (2006), which reports better results than those achieved by Patton et al. (1995) for the smaller sequences.
We use two sets of benchmark sequences employed first by Unger and Moult (1993b). These are amino acid sequences of Hs and Ps generated randomly: 10 sequences are of length 27 and 10 sequences of length 64 (Tables 1 and 2). We evaluate the results using the following metrics:

- Fitness value: It is the total number of non consecutive H-H contacts.
- Number of fitness evaluations: This is the number of times the fitness function is computed to reach the final fitness score for a specific sequence. This metric is used as an indicator of the efficiency of our algorithm.

We executed our EGA program on a PC running MS-Windows XP operating system with a 2.33 GHz CPU and 2 GByte RAM memory.

Seq #	Sequence
273d.1	Phphphhhpphphpppppppppphhp
273d.2	Phhpppppppppphhpphhpphpphph
273d.3	Hhhhpppppphpppppphhhpppppppph
273d.4	Hhhppphhhhpppphphpphhpphpphh
273d.5	Hhhhpppphphhpppphhpppppppppp
273d.6	Hpppppphphhhpphhpppphpppphph
273d.7	Hpphphhpppphpppppphphhphphhh
273d.8	Hpppppppppppphphpppppppphphh
273d.9	Pppppppphhhpppphphpppphpphppp
273d.10	Pppphhphphphphpphhphhphppp

Table 1. Benchmark sequences of length 27

Seq #	Sequence
643d.1	pphhhhhpppphhpppppphhpppphpppppphphpppphpphpphpppppphpppphhphhpphpphp
643d.2	pphphpphpphhhphhhhpphhhpppphphpppphphpppphphpppppphphpphphpppphpphpp
643d.3	hphhpphhphpppppphhhphhhhpphpphphhpppphphpphhhphhphpppppphhhhhhhhppp
643d.4	hpphhpphpphphpphpppphpppppphphphhhpphphpppphphpphhpphpphpphphhhph
643d.5	hpppphhpphphpppphpppphphhpppphhphphhphpphpppphpphphhhpphpphpphhhphhhh
643d.6	hpphhphhhhpppppphhpphpppphhpppphpphphhphpppphhpppphpppppphpppphphh
643d.7	pppphpppphpppphhhhphhpppppphpphphhphphpppppphpppppppppppphhhhpppphhpph
643d.8	ppphhhpphphpphpphhpppphpphpphhphpppphpppppppphphhhphhhhhpphhpppphpph
643d.9	hpphpphhhpppphphpppphphhphhhhhpppphphphpppphphpphhphpppphpphhphp
643d.10	pphpphpphhhpppphphpphpphpppppphpphhhpphpphpphphpppppphhhpppppphphp

Table 2. Benchmark sequences of length 64

Tables 3 and 4 show the results of our EGA in comparison with the previously published results of Johnson et al. (2006) for proteins with lengths 27 and the results of Patton et al. (1995) for 64 amino acids, respectively .

Table 3 results show that the proposed EGA produces fitness values that as the same as those of the technique of Johnson et al. (2006) in 9 out of 10 cases with a better value in the remaining case. However, the numbers of evaluations of fitness values is significantly less in the afore-mentioned 9 cases; it is greater in the 10th case where EGA clearly managed to search larger areas of the search space that enabled it to find a fitter protein structure.

Table 4 results show that the proposed EGA produces fitter protein structures than Patton et al. (1995) in 70% of the 10 cases, whereas the remaining 30% of the cases are identical.

4. Conclusion and future work

We have proposed a genetic algorithm that is enhanced with heuristic methods. These heuristics are incorporated into the crossover and mutation operations for the purposes of dealing with infeasible intermediate candidate solutions and of guiding the search into fitter regions of the search space. The empirical work shows that this enhanced genetic algorithm gives better results in terms of the protein structures, the algorithm efficiency, or both. Future work would consider larger proteins and visualization of the results. Furthermore, predicting the structures of large proteins is likely to require parallel processing in order to reduce execution time.

Seq #	Proposed EGA		Johnson et al. (2006)	
	Fitness	#Fitness Eval.	Fitness	#Fitness Eval.
273d.1	9	1,450	9	15,854
273d.2	10	5,473	10	19,965
273d.3	8	1,328	8	7,991
273d.4	15	5,196	15	23,525
273d.5	8	1,184	8	3,561
273d.6	12	18,012	11	14,733
273d.7	13	4,920	13	23,112
273d.8	4	654	4	889
273d.9	7	1,769	7	5,418
273d.10	11	3,882	11	5,592

Table 3. Results for sequences of length 27

Seq #	Proposed EGA	Patton et al. (1995)
	Fitness	Fitness
643d.1	28	27
643d.2	32	30
643d.3	40	38
643d.4	35	34
643d.5	36	36
643d.6	31	31
643d.7	25	25
643d.8	35	34
643d.9	34	33
643d.10	27	26

Table 4. Results for sequences of length 64

5. Acknowledgement

This work was partially supported by the Lebanese American University and the National Council for Scientific Research.

6. References

Anfinsen, C.B. (1973). Principles that govern the folding of proteins, *Science*, 181-187.

Chen, M. and Huang, W. (2005). A Branch and Bound Algorithm for the Protein Folding Problem in the HP Lattice Model. *Genomics, Proteomics & Bioinformatics*, Vol. 3, No. 4.

Datta, A., Talukdar, V. and Konar, A. (2008). Neuro-Swarm Hybridization for Protein Tertiary Structure Prediction. In proceedings of the 2nd National Conference on Recent Trends in Information Systems (ReTIS-08), Jadavpur University, Kolkata.

Dill, K. A. (1985). Theory for the folding and stability of globular proteins. *Biochemistry*, 24, 6, 1501-1509.

Dill, K.A., Fiebig, K.M. and Chan, H.S. (1993). Cooperativity in Protein-Folding Kinetics. *Proceedings of the National Academy of Sciences of the United States of America*, 90, 1942-1946.

Floudas, C.A. (2007). Computational methods in protein structure prediction. *Biotechnology and Bioengineering*, 97(2), 207-213.

Holland, J. (1975). *Adaptation in Natural and Artificial Systems*. University of Michigan Press. Michigan.

Hsu, H.P., Mehra, V., Nadler, W. and Grassberger, P. (2003). Growth Algorithm for Lattice Heteropolymers at Low Temperatures. *Journal of Chemical Physics*, 118, 444-51.

Johnson, C. and Katikireddy, A. (2006). A Genetic Algorithm with Backtracking for Protein Structure Prediction. *Proceedings of the 8th Annual Conference on Genetic and Evolutionary Computation, Washington, USA*.

Jones, D.T. (1998). THREADER: protein sequence threading by double dynamic programming. In *Computational Methods in Biology* (ed. Salzberg, S., Searl, D. and Kasif, S.). Amsterdam: Elsevier Science.

Klepeis, J.L. and Floudas, C.A. (2003). *Ab initio* tertiary structure prediction of proteins. *Journal of Global Optimization*, 25, 113–140.

Kopp, J. and Schwede, T. (2004). Automated protein structure homology modeling: a progress report. *Pharmacogenomics Journal*, 5, 4, 405-416.

Lathrop, R.H. et al. (1998). Computational Methods in Molecular Biology. *Elsevier Press*, 12, 227-283.

Li, W., Wang, T., Li, E., Baker, D., Jin, L., Ge, S., Chen, Y. and Zhang, Y. (2006). Parallelization and performance characterization of protein 3D structure prediction of Rosetta. *IEEE 20th Int. Parallel and Distributed Processing Symposium*, Rhodes Island, Greece.

Liang, F. and Wong, W.H. (2001). Evolutionary Monte Carlo for protein folding simulations. *Journal of Chemical Physics*, 115, 7, 3374-3380.

Notredame, C. (2002). Recent progress in multiple sequence alignment: a survey. *Pharmacogenomics Journal*, 3(1), 131-144.

Mansour, N., Kehyayan, C., and Khachfe, H. (2009). Scatter search algorithm for protein structure prediction. *International Journal of Bioinformatics Research and Applications*, Vol. 5, No. 5, 501-515.

Pandit, S.B., Zhang, Y., and Skolnick, J. (2006). Tasser-lite: an automated tool for protein comparative modeling. *Biophysics Journal*, 91, 11, 4180-4190.

Patton, A. L., Punch, W. F. and Goodman, E. D. (1995). A standard GA approach to native protein conformation prediction. *In Proceedings of the 6th International Conference on Genetic Algorithms.*

Prusiner, S.B. (1998). Prions. Proceedings of the National Academy of Sciences of the United States of America, 95, 13363-13383.

Rylance, G. (2004). Applications of Genetic Algorithms in Protein Folding Studies. First year report, School of Chemistry, University of Birmingham, England.

Schulze-Kremer, S. (2000). Genetic algorithms and protein folding. *Methods in Molecular Biology*, 143, 175-222.

Setubal, J. and Meidanis, J. (1997). *Introduction to computational molecular biology*. Boston: PWS Publishing Company.

Shmygelska, A. and Hoos, H. H. (2005). An Ant Colony Optimization Algorithm for the 2D and 3D Hydrophobic Polar Protein Folding Problem, *BMC Bioinformatics*, 6(30).

Sikder, A.R. and Zomaya, A.Y. (2005). An Overview of Protein-Folding Techniques: Issues and Perspectives. *International Journal of Bioinformatics Research and Applications*, 1, 1, 121-143.

Skolnick, J., Kihara, D. and Zhang, Y. (2004). Development and large scale benchmark testing of the PROSPECTOR 3 threading algorithm. *Proteins*, 56, 502-518.

Srinivasan, R. and Rose, G.D. (2002). Ab initio prediction of protein structure using LINUS. *PROTEINS: Structure, Function, and Genetics*, 47, 489-495.

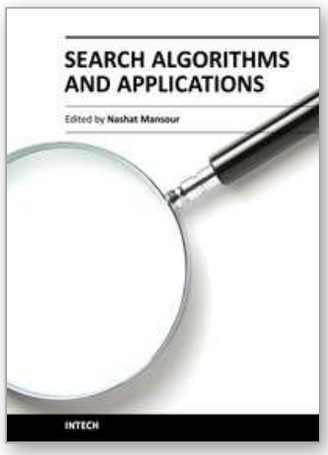
Toma, L. and Toma, S. (1996). Contact interactions method: A new algorithm for protein folding simulations. *Protein Science*, 5, 147-153.

Unger, R. and Moult, J. (1993a). Finding the Lowest Free Energy Conformation of a Protein is an NP-Hard Problem: Proof and Implications. *Bulletin of Mathematical Biology*, 1183-1198.

Unger, R. and Moult, J. (1993b). Genetic algorithms for protein folding simulations. *Journal of Molecular Biology*, 231, 75-81.

Yue, K. and Dill, K.A. (1995). Forces of Tertiary Structural Organization in Globular Proteins. *Proceedings of the National Academy of Sciences of the United States of America*, 92, 146-150.

Zhang, X. and Li, T. (2007). Improved Particle Swarm Optimization Algorithm for 2D Protein Folding Prediction. *The 1st International Conference on Bioinformatics and Biomedical Engineering*, 53-56.



Search Algorithms and Applications

Edited by Prof. Nashat Mansour

ISBN 978-953-307-156-5
Hard cover, 494 pages
Publisher InTech
Published online 26, April, 2011
Published in print edition April, 2011

Search algorithms aim to find solutions or objects with specified properties and constraints in a large solution search space or among a collection of objects. A solution can be a set of value assignments to variables that will satisfy the constraints or a sub-structure of a given discrete structure. In addition, there are search algorithms, mostly probabilistic, that are designed for the prospective quantum computer. This book demonstrates the wide applicability of search algorithms for the purpose of developing useful and practical solutions to problems that arise in a variety of problem domains. Although it is targeted to a wide group of readers: researchers, graduate students, and practitioners, it does not offer an exhaustive coverage of search algorithms and applications. The chapters are organized into three parts: Population-based and quantum search algorithms, Search algorithms for image and video processing, and Search algorithms for engineering applications.

InTech Europe
University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

InTech China
Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821

IntechOpen