Analysis of an Immune Algorithm for Protein Structure Prediction

Andrew J. Bennett, Roy L. Johnston, and Eleanor Turpin
University of Birmingham, Birmingham, United Kingdom
E-mail: ajb@tc.bham.ac.uk

Jun Q. He
Aberystwyth University, Aberystwyth, United Kingdom

Keywords: HP lattice bead model, immune algorithm, population diversity tracking, protein modelling

Received: June 23, 2008

The aim of a protein folding simulation is to determine the native state of a protein from its amino acid sequence. In this paper we describe the development and application of an Immune Algorithm (IA) to find the lowest energy conformations for the 2D (square) HP lattice bead protein model. Here we introduce a modified chain growth constructor to produce the initial population, where intermediate infeasible structures are recorded, thereby reducing the risk of attempting to perform wasteful point mutations during the mutation phase. We also investigate various approaches for population diversity tracking, ultimately allowing a greater understanding of the progress of the optimization.

Povzetek: V članku je opisan razvoj in izvedba imunskega algoritma (IA) za iskanje najnižje energijske strukture za 2D (kvadratne) HP mrežno nanizanega modela proteina.

1 Introduction

Predicting the 3-dimensional secondary and tertiary structure of a protein molecule from its (primary structure) amino acid sequence alone is an important problem in chemical biology [1]. Under certain physiological conditions, the amino acid chain will reliably fold into a specific native state (biologically active conformation). The protein folding problem is the search for this native state for a given sequence of amino acid residues. The reliability of protein folding is said to be dominated by the presence of a "folding funnel" on the folding energy landscape since systematic or random searching is clearly infeasible for large numbers of amino acids [2]. Therefore, discovering the nature of the folding energy landscape is necessary to develop a better understanding of the folding dynamics [3].

Many protein models have been developed, ranging from simple, minimalist models such as the HP lattice bead model [4], to more complicated and computationally expensive models such as off-lattice interpretations. The most common lattice structures are 2D square and 3D cubic. More computationally intense models include the dynamical lattice and all-atom models, both introducing more complicated fitness functions.

In this work, the standard HP lattice bead model has been incorporated into an immune algorithm. Despite the minimalistic approach employed by this model, it has been shown to belong to the "NP-Hard" set of problems [5]. Monte Carlo [6], chain growth algorithms [4], simulated annealing [7], genetic algorithms [5, 8, 9], ant colony optimization [10] and more recently immune algorithms [11] have been developed by many researchers as heuristic and approximate solutions for this and other computationally hard problems.

2 Methodology

2.1 The HP lattice bead model

In this work, the standard HP lattice bead model is embedded in a 2-dimensional square lattice, restricting bond angles to only a few discrete values [4]. Interactions are only counted between topological neighbours, that is between beads (representing amino acids) that lie adjacent to each other on the lattice, but which are not sequence neighbours [3]. The energies corresponding to the possible topological interactions are as follows:

$$\epsilon_{HH} = -1.0 \quad \epsilon_{HP} = 0.0 \quad \epsilon_{PP} = 0.0 \tag{1}$$

By summing over these local interactions, the energy of the model protein can be obtained:

$$E = \sum_{i<j} \epsilon_{ij} \Delta_{ij} \tag{2}$$

where

$$\Delta_{ij} = \begin{cases} 1 & \text{if i and j are topological neighbours,} \\ & \text{but are not sequence neighbours;} \\ 0 & \text{otherwise.} \end{cases}$$

The HP lattice model recognises only the hydrophobic interaction as the driving force in protein folding, with

Name	Length	E^*	Sequence
HP-18a	18	-9	$PHP_2HPH_3PH_2PH_5$
HP-18b	18	-8	$HPHPH_3P_3H_4P_2H_2$
HP-18c	18	-4	$H_2P_5H_2P_3HP_3HP$
HP-20a	20	-10	$H_3P_2(HP)_2HP_2(HP)_2HP_2H$
HP-20b	20	-9	$HPHP_2H_2PHP_2HPH_2P_2HPH$
HP-24	24	-9	$H_2P_2(HP_2)_6H_2$
HP-25	25	-8	$P_2HP_2(H_2P_4)_3H_2$
HP-36	36	-14	$P_3H_2P_2H_2P_5H_7P_2H_2P_4H_2P_2HP_2$
HP-48	48	-23	$P_2H(P_2H_2)_2P_5H_{10}P_6(H_2P_2)_2HP_2H_5$
HP-50	50	-21	$H_2(PH)_3PH_4P(HP_3)_3P(HP_3)_2HPH_4(PH)_4H$

Table 1: Benchmark HP sequences used in the present study [12]. The lowest energies that have been found for these sequences are indicated by E^*.

many native structures protecting the hydrophobic core with polar residues, resulting in a compact arrangement [11]. This idea reflects the repulsive nature of the interactions between the hydrophobic residues and the surrounding water molecules [3].

2.2 The coordinate system

A previous study has illustrated how a local coordinate system offers better performance than a global one for studying protein folding [2]. In this work, a *local coordinate system* is used to define the folding conformation of the model proteins, that is the position of bead j is defined relative to beads $(j-1)$ and $(j-2)$. As the energy is identical for rotationally related structures, the bond between the first two beads lies along the x-axis, with these beads having coordinates (0,0) and (1,0) respectively. As a result, the search spaced is halved. The bond joining the $(j-1)^{th}$ and j^{th} beads can be left, right or straight ahead relative to the bond joining the $(j-2)^{th}$ and $(j-1)^{th}$ bead, corresponding to an integer representation of 0, 1 and 2 respectively. The protein conformation is therefore expressed as a *conformation vector*, containing a list of 0's, 1's and 2's.

For this study, a set of well investigated protein benchmark sequences have been considered: the tortilla HP benchmark sequences [12]. They range in length from eighteen to fifty beads and are listed in Table 1. The table also includes the energy, E^*, of the putative global minimum (or conformations, since all of these structures have degenerate global minima) for each sequence.

3 The immune algorithm

An immune algorithm [13] is inspired by the clonal selection principle employed by the human immune system. In this process, when an antigen enters the body, B and T lymphocytes are able to clone upon recognition and bind to it [13]. Many clones are produced in response and undergo many rounds of somatic hypermutation. The higher the fitness of a B cell to the available antigens, the greater the chance of cloning. Cells have a certain life expectancy, allowing a higher specific responsiveness for future antigenic attack [11].

The IA presented here includes the aging, cloning and selection operators used in a previous study by Cutello *et al.* [11], with modified constructor and mutation operators. The constructor employs a backtracking algorithm that records some of the possible mutations by testing bead placement during chain growth. These possibilities are exploited and updated during the mutation process, preventing an infeasible conformation from occurring based on the preceding self-avoiding structure for a particular point in the model protein chain. In retaining this information, infeasible mutations are not explored, allowing a greater number of constructive mutations to be investigated. Figure 1 illustrates the stages involved in placing two consecutive beads during the chain growth phase. Before committing a bead to the lattice, all possible directions are explored, 1(a), and from the valid options available, a random choice is made, 1(b). Again all possible choices are investigated, marking any infeasible options (note that choosing left will not result in a self avoiding conformation), 1(c), and a valid choice is selected from the remaining options, 1(d). Any remaining valid choices are left unmarked for use in the first mutation phase after the initial chain growth. Once a valid mutation has been made, the entire structure is reconstructed as before marking any infeasible directions as a result of the new conformation vector.

In the basic IA set up, there are a maximum of 10,000,000 fitness evaluations, with the maximum number of generations set to 500,000. In order to estimate the optimal combination of parameters, we adopted the procedure used by Cutello *et al.*, whereby the maximum B-Cell age and the number of clones were each varied from 1 to 10. Population sizes examined were 10, 25, 50, 100 and 200. This provided a combination of 500 different parameter sets for each sequence, which was applied to all the benchmark sequences up to 25 beads in length. All fitness evaluations for the best success rates were collated and graded for overall performance. As a result of this preliminary testing, the results presented below were obtained using a maximum B-Cell age of 4, 3 clones and a popu-

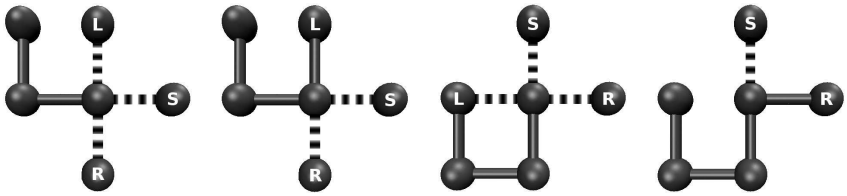


Figure 1: Stages of the chain growth algorithm investigating left (L), right (R) and straight (S) availability (a), random selection from all available directions (b), at the next locus investigation of L, R and S availability (c) and random selection from remaining available S and R directions (d).

lation size of 10. All results quoted are averaged over 30 independent runs.

4 Results

4.1 Algorithm comparison

With CPU time being hardware dependent, the number of fitness evaluations (together with the percentage success rate) have been used to assess the efficiency of the algorithm, as shown in Table 2 for the benchmark sequences.

It is apparent from Table 2 that, although the use of memory B-Cells [11] hinders the discovery of global minima for some of the smaller sequences, it enhances the search for the larger, more difficult to find sequences. The memory ability allows mid to high fitness conformations to remain in the population for a longer number of generations. For larger sequences, this allows a more detailed exploration in certain areas of the potential energy surface, permitting the memory B-Cells to converge towards the global solution much sooner. In contrast, for smaller sequences the mid to high fitness range is much smaller, thereby preventing a rapid exploration of the potential energy surface by retaining unfavourable segments of local structure for a larger number of generations. Generally, the use of memory B-Cells allows a more diverse inspection of the potential energy surface, due to a greater number of the degenerate conformations being found. This is achieved as favourable fragments of local structure are not rapidly disposed of during the retirement process, hindering efficiency as a consequence.

The algorithm presented here shows promising results, being comparable to the work of Cutello *et al.* [11]. While our success rates for the larger sequences (e.g. HP-48) are a little lower, in some cases our number of fitness evaluations show an improvement.

4.2 Analysis of global minima

The compact structural arrangement present in all global minima (GMs) is apparent from the example GMs shown in Fig. 2. With the driving force being the hydrophobic topological contact, it can be seen that compact hydrophobic cores give rise to high fitness conformations. Inspection of the HP-48 global minimum (i) allows us to understand the poor success rate for this sequence. The 5×5 hydrophobic core presents a problem to the IA (or other optimization algorithms [3]) in achieving convergence, as a single misplaced hydrophobic bead will result in only a metastable conformation. The problem does not exist for the HP-50 sequence (j), due to the presence of two small hydrophobic cores coupled by a chain of hydrophobic beads, which explains the higher success rate and fewer average structure evaluations necessary for HP-50, compared with HP-48 and (when using memory B-cells) even the much shorter HP-36 sequence [3]. The work of Cutello *et al.* supports this idea [11], as similar magnitudes of the number of fitness evaluations for these problematic sequences can be seen, with a much lower success rate for HP-48 than for any other instance.

4.3 Tracking population diversity

The much larger populations required to ensure population diversity can be problematic for both GAs and IAs. In this section, a single run, with population size 200 for sequence HP-20a has been analyzed. The global minimum was found in generation 28, at which point the algorithm was terminated due to meeting the search criteria. In order to help us understand the progress of the optimization and ultimately to improve the methodology, monitoring population diversity and the progress of the algorithm is beneficial.

Figure 3(a) assigns a colour to each of the three possible direction decisions (corresponding to alleles in a genetic sense) made when placing each successive bead. It can be seen that initial structure generation, using the IA's constructor, is indeed statistically uniform, showing the frequency of left (grey), right (light blue) and straight ahead (dark blue) choices at each locus of the model protein chain to be very similar. In contrast, Fig. 3(b) illustrates how this statistical distribution is skewed in the final population (generation 28), in that the IA has concentrated its search to a much narrower region of the potential energy surface. It should also be noted that position 6 in the chain has a very low frequency of the straight ahead choice (dark blue), because (for most population members) previous direction decisions preclude (for structural and/or energetic reasons) this choice from being made at this chain position.

Sequence	No Memory B-Cells		Memory B-Cells	
	%Success	No. Evaluations	%Success	No. Evaluations
HP-18a	100	89,578	100	117,251
HP-18b	100	40,167	100	200,740
HP-18c	100	87,761	100	72,270
HP-20a	100	26,207	100	312,405
HP-20b	100	15,221	100	30,414
HP-24	100	26,580	100	49,616
HP-25	100	79,042	100	95,123
HP-36	63	4,867,993	90	3,082,014
HP-48	3	6,318,721	3	4,195,086
HP-50	50	4,904,031	96	853,706

Table 2: Comparison of the percentage success and average number of structure evaluations with and without using memory B-Cells

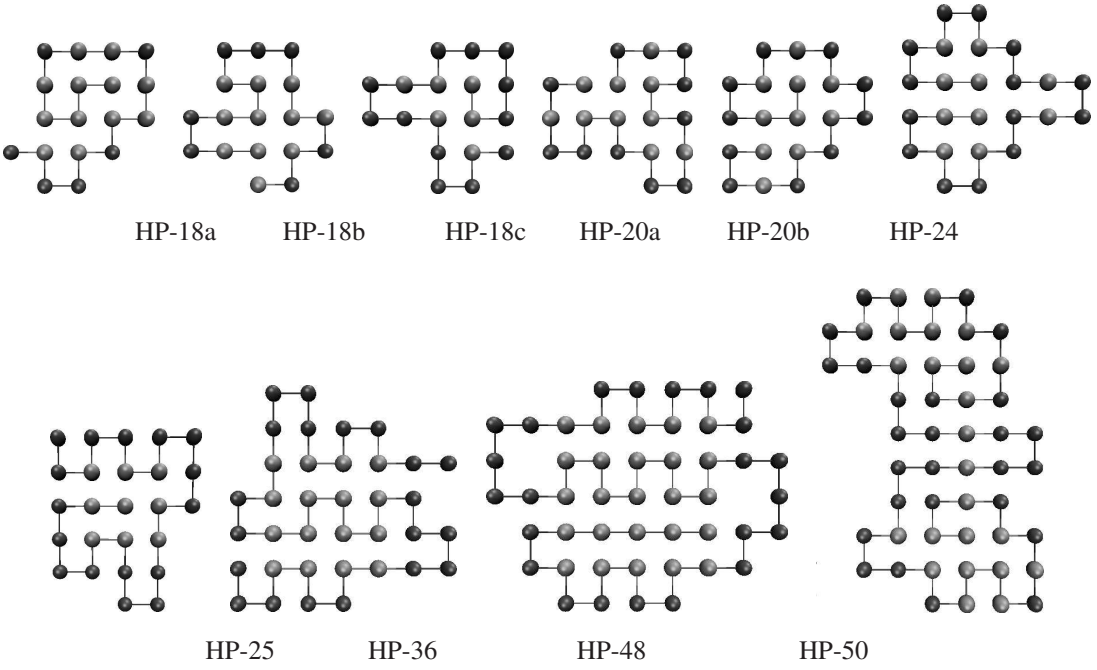
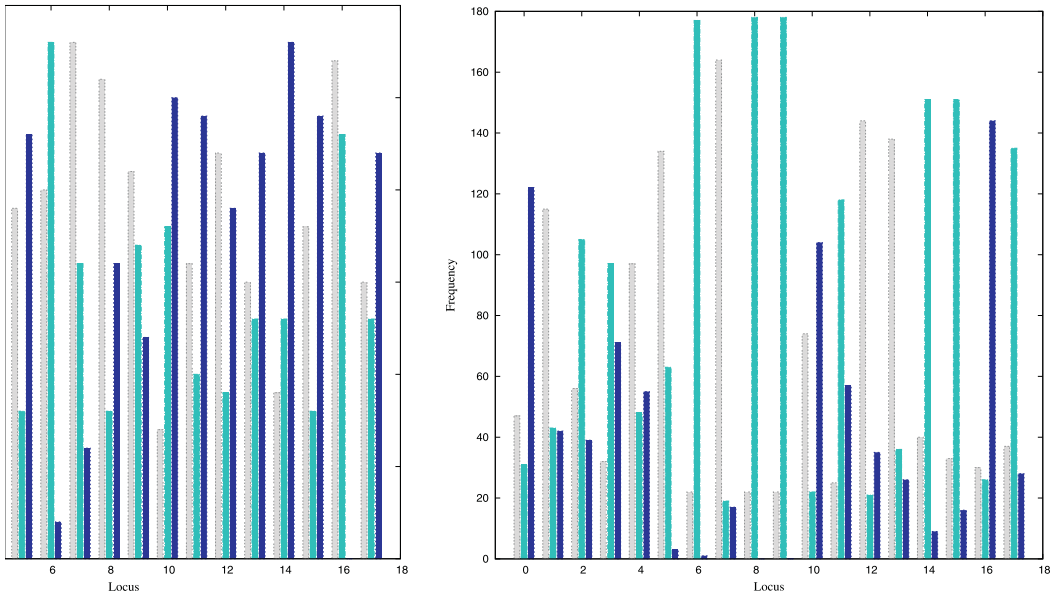


Figure 2: Examples of GM structures for the benchmark sequences.

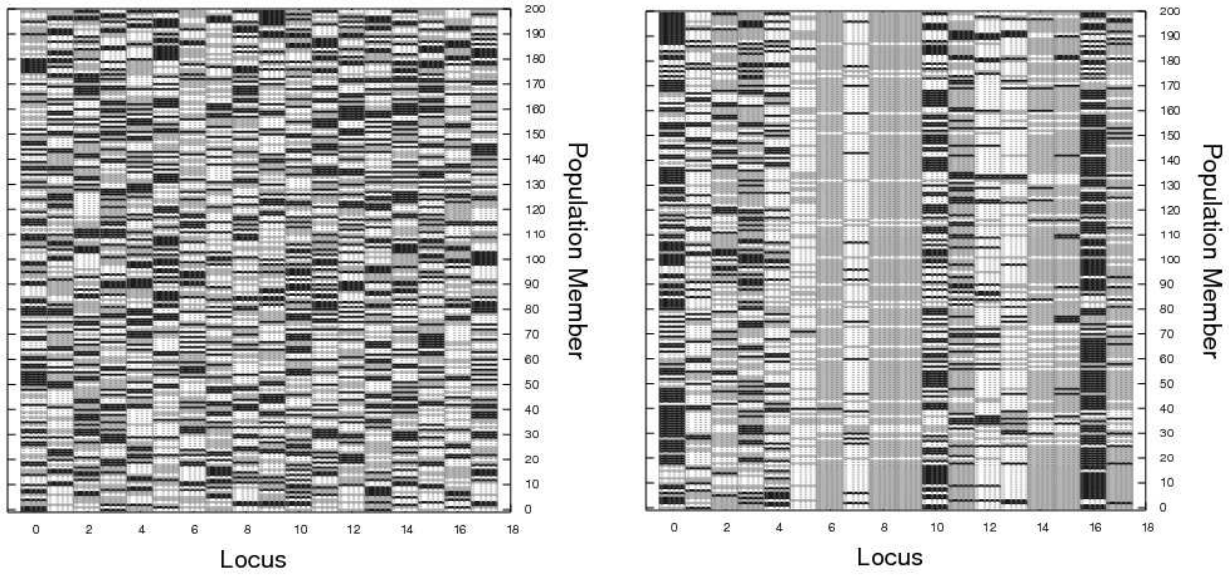
Figure 4 shows a graphical representation of the initial and final populations of the calculation. By plotting the conformation vector for each population member, the population can be quickly compared for diversity. Population members are ordered by descending fitness and the colour scheme is similar to the allele frequency distribution shown in Fig. 3, but with white replacing grey for the left choice. It is clear that initially the population has high diversity (in agreement with the allele frequency plot shown above), with the algorithm preserving favourable regions of local structure (corresponding to schemata in a GA sense) as the calculation converges. More detailed analysis of the final population shows that there are often correlations (or anti-correlations) between directions at specific loci, with certain combinations giving rise to favourable energies or infeasible structures, respectively.

For simple protein models such as the HP lattice bead model, the Hamming distance (d_H, which is the number of bit differences between two conformation vectors) can be used as a simple measure of similarity between structures in the population. Figure 5(a) plots the frequency of the Hamming distances between all pairs of structures in the population as a function of generation. (As the population size is 200, there are a total of 19,900 pair Hamming distances). It can be seen how the diversity of the population changes as the calculation approaches the global minimum (which is found in generation 28). Combining this with a plot of the best, worst and average fitnesses in the population, as a function of generation (Fig. 5(b)), it should be noted that structural diversity shows a more uniform spread (beginning around generation 20) as favourable segments of local structure begin to dominate the population, with the search focussing on a much more concentrated area of the potential energy surface. It is also evident that the



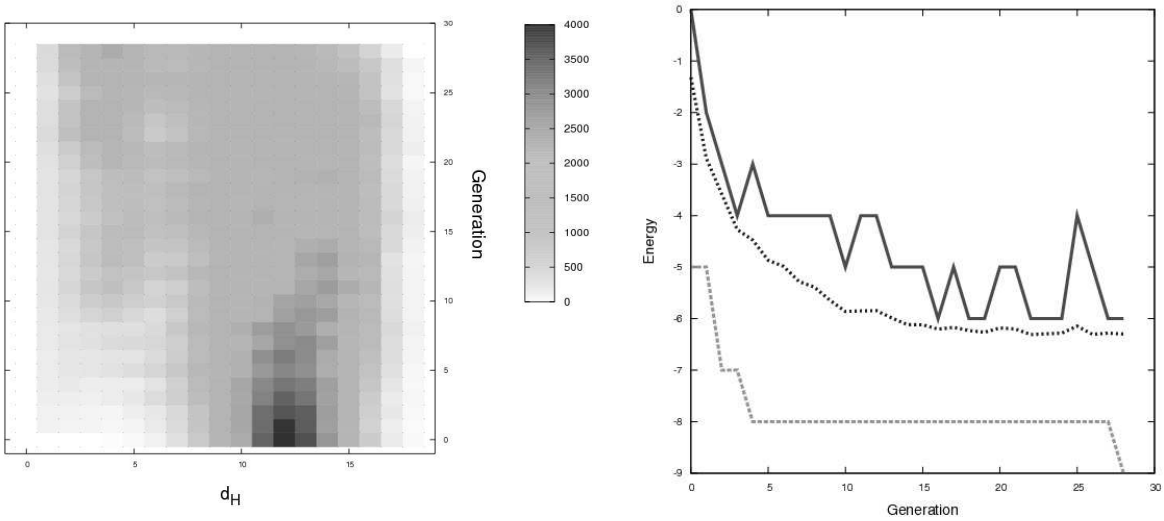
(a) (b)

Figure 3: The frequency of alleles at each locus along the model protein chain for the initial population (a) and the final population (b), left (light grey), right (dark grey) and straight ahead (black).



(a) (b)

Figure 4: Graphical representation of an initial population (a) and final population (b) of B-Cells, left (light grey), right (dark grey) and straight ahead (black). Population members are sorted by descending fitness, with structures of the highest energy at the bottom of the plot.



(a) (b)

Figure 5: (a) The density of pairwise Hamming distances, d_H, between population members throughout the calculation. (b) The change in energy throughout the calculation, showing the best (dashed), worst (solid) and average (doted) energies in each generation.

population diversity drastically decreases during the final stages of the calculation, not just in the final generation. This confirms that the calculation has not discovered the global minimum by chance, but a directed search strategy has been employed.

5 Conclusions

Although implementation of a modified constructor for use in the mutation phase of the IA has not always given greater success rates (especially for more challenging sequences), it has allowed for a more efficient search to be performed in some cases, showing a descrease in the number of fitness evaluation performed. The use of population diversity tracking allows a greater understanding of the algorithm's ability to explore areas of the potential energy surface of these simple model proteins. Areas of favourable local structure along the chain can be assessed, illustrating the important allele combinations that give rise to the determination of global minima. We are currently applying these approaches to more realistic protein models.

Acknowledgements

AJB thanks Dr Benjamin Curley for programming assistance and the University of Birmingham for PhD funding. Calculations were performed on the University of Birmingham's BlueBEAR 1500+ processor cluster, funded by the EPSRC (under SRIF3) and the University of Birmingham.

References

[1] K.M. Merz (ed.). *The Protein Folding problem and Structure Prediction*. Birkhauser, 1994.

[2] N. Krasnogor, W.E. Hart, J. Smith, and D. Pelta. Protein Structure Prediction with Evolutionary Algorithms. In *Proc. 1999 International Genetic and Evolutionary Computation Conference (GECCO99)*, Orlando, Florida, USA, 1999, pages 1569–1601.

[3] G.A. Cox and R.L. Johnston. Analyzing Energy Landscapes for Folding Model Proteins. *J. Chem. Phys.*, 124(20):204714, 2006.

[4] T. Beutler and K. Dill. A Fast Conformational Search Strategy for Finding Low Energy Structures of Model Proteins. *Protein Sci.* 5(10):2037–2043, 1996.

[5] R. Unger and J. Moult. Genetic Algorithms for Protein Folding Simulations. *J. Mol. Biol.*, 231:75–81, 1993.

[6] R. Ramakrishnan, B. Ramachandran, and J.F. Pekny. A Dynamic Monte Carlo Algorithm for Exploration of Dense Conformational Spaces in Heteropolymers. *J. Chem. Phys.*, 106(6):2418–2425, 1997.

[7] S. Kirkpatrick and C.D. Gellat. Optimization by Simulated Annealing. *Science*, 220(4598):671–680, 1983.

[8] W. Liu and B. Schmidt. Mapping of Genetic Algorithms for Protein Folding onto Computational Grids. *IEICE T. Inf. Syst.*, 89(2):589–596, 2006.

[9] J. Song, J. Cheng, T. Zeng, and J. Mau. A Novel Genetic Algorithm for HP Model Protein Folding. In *Proc. Sixth International Conference on Parallel and Distributed Computing Applications and Technologies (PDCAT'05)*, Dalian, China, 2005, pages 935–937.

[10] A. Shmygelska and H.H. Hoos. An Improved Ant Colony Optimization Algorithm for the 2D HP Protein Folding Problem. *Lect. Notes Comput. Sci.*, 2671:400–412, 2003.

[11] V. Cutello, G. Niscosia, M. Pavone, and J. Timmis. An Immune Algorithm for Protein Structure Prediction on Lattice Models. *IEEE T. Evol. Comput.*, 11(1):101–117, 2007.

[12] W.E. Hart, S. Istrail. HP Benchmarks. `www.cs.sandia.gov/tech_reports/compbio/tortilla-hp-benchmarks.html`.

[13] L.N. de Castro and J. Timmis. *Artifical Immune Systems and Their Applications*. Springer-Verlag, 1999.