

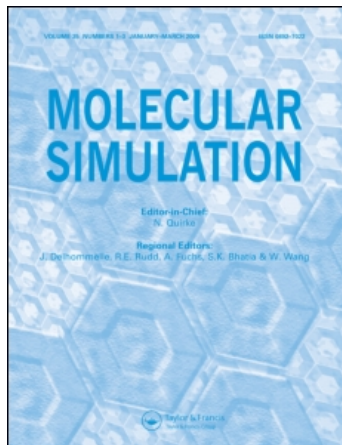
This article was downloaded by:

On: 14 January 2011

Access details: *Access Details: Free Access*

Publisher *Taylor & Francis*

Informa Ltd Registered in England and Wales Registered Number: 1072954 Registered office: Mortimer House, 37-41 Mortimer Street, London W1T 3JH, UK



Molecular Simulation

Publication details, including instructions for authors and subscription information:

<http://www.informaworld.com/smpp/title~content=t713644482>

Protein Structure Prediction as a Hard Optimization Problem: The Genetic Algorithm Approach

Mehul M. Khimasia^a; Peter V. Coveney^b

^a Theory of Condensed Matter, Cavendish Laboratory, University of Cambridge, Cambridge, U.K. ^b Schlumberger Cambridge Research, High Cross, Cambridge, U.K.

To cite this Article Khimasia, Mehul M. and Coveney, Peter V.(1997) 'Protein Structure Prediction as a Hard Optimization Problem: The Genetic Algorithm Approach', *Molecular Simulation*, 19: 4, 205 — 226

To link to this Article: DOI: 10.1080/08927029708024151

URL: <http://dx.doi.org/10.1080/08927029708024151>

PLEASE SCROLL DOWN FOR ARTICLE

Full terms and conditions of use: <http://www.informaworld.com/terms-and-conditions-of-access.pdf>

This article may be used for research, teaching and private study purposes. Any substantial or systematic reproduction, re-distribution, re-selling, loan or sub-licensing, systematic supply or distribution in any form to anyone is expressly forbidden.

The publisher does not give any warranty express or implied or make any representation that the contents will be complete or accurate or up to date. The accuracy of any instructions, formulae and drug doses should be independently verified with primary sources. The publisher shall not be liable for any loss, actions, claims, proceedings, demand or costs or damages whatsoever or howsoever caused arising directly or indirectly in connection with or arising out of the use of this material.

PROTEIN STRUCTURE PREDICTION AS A HARD OPTIMIZATION PROBLEM: THE GENETIC ALGORITHM APPROACH

MEHUL M. KHIMASIA^a and PETER V. COVENEY^b

^a*Theory of Condensed Matter, Cavendish Laboratory, University of Cambridge,
Madingley Road, Cambridge CB3 0HE, U.K.;*

^b*Schlumberger Cambridge Research, High Cross, Madingley Road,
Cambridge CB3 0EL, U.K.*

(Received February 1997; Accepted March 1997)

Protein structure prediction can be shown to be an NP-hard problem; the number of conformations grows exponentially with the number of residues. The native conformations of proteins occupy a very small subset of these, hence an exploratory, robust search algorithm, such as a genetic algorithm (GA), is required. The dynamics of genetic algorithms tend to be complicated and problem-specific. However, their empirical success warrants their further study. In this paper, guidelines for the design of genetic algorithms for protein structure prediction are determined. To accomplish this, the performance of the *simplest* genetic algorithm is investigated for simple lattice-based protein structure prediction models (which is extendible to real-space), using energy minimization. The study has led us to two important conclusions for 'protein-structure-prediction-genetic-algorithms'. Firstly, they require high resolution building blocks attainable by multi-point crossovers and secondly they require a local dynamics operator to 'fine tune' good conformations. Furthermore, we introduce a statistical mechanical approach to analyse the genetic algorithm dynamics and suggest a convergence criterion using a quantity analogous to the free energy of a population.

Keywords: Energy minimization; lattice models; simple exact models

1. INTRODUCTION

Conceptually, proteins fold from their 1D polymer chain of amino acids (primary structure) to 3D stable, 'unique' conformations (tertiary structure). Anfinsen [1] showed that folding requires knowledge of the amino acid sequence alone; the determination of the native (biologically functioning) structure from its sequence is known as the *protein folding* problem. Much

research has gone into elucidating the folding dynamics [2] but a practical theory is still beyond our understanding.

The folding problem attempts to understand the dynamics of the folding – *how* the sequence of amino acids arrive at the native state. It is clear that a full solution to this problem would be able to predict the tertiary structure of an amino acid sequence. Unfortunately, such a solution is beyond our understanding at present.

However, amino acid sequences are effectively being determined at a higher rate than that of their corresponding native structures. Since knowledge of the native structure is important to understanding the function of a protein, a potentially more practical problem is protein structure prediction. There is a possible confusion in terminology between the two terms, *protein folding* and *protein structure prediction*, as researchers use the terms loosely and interchangeably. The difference must be stressed as there are ‘folding simulations’, in the literature, which carry out a conformational search in a manner that is not attributable to the true folding dynamics of proteins (e.g. [3]).

Protein folding mainly concerns the dynamics of the problem which experimentalists study using hydrogen-exchange techniques, for example; theorists often use computer simulations of simple models, often on lattices, to elucidate the folding dynamics. Protein structure prediction, however, is only interested in the end result; experimentalists often use crystallisation techniques coupled with x-ray diffraction to determine the tertiary structure, whereas theorists tend to use computer-based optimization methods. This is the approach discussed in this paper. The problem involves two aspects: (1) the specification of the function to optimize and (2) the choice of a search algorithm.

The prediction of protein structures using optimization methods have been more successful using comparative modelling [4] techniques which include sequence alignment, threading [5] and the use of secondary structural propensities [6]. However, a physical approach based on energy minimization is used here. It is clear that the interactions between the residues and the solvent molecules drive the protein towards its native state. Determining the fundamental interactions is important, not only for protein structure prediction, but also for the protein folding problem. A further reason for the physical approach is that comparative modelling requires the known structure of one or more homologous proteins – these cannot always be guaranteed to exist.

The second aspect of the protein structure prediction problem is concerned with designing a search strategy for the energy minimization. It is clear that a blind search through the conformational space is impossible,

as it would take a time greater than the age of the universe. A similar 'paradox' for real proteins, based on observed protein folding times ($\sim 10^{-3}$ s), was first noted by Levinthal [7]. Furthermore, several authors have proved various models of protein structure prediction to be computationally NP-hard* [9–11]. This implies that no efficient algorithm can be designed to guarantee finding the native state amongst the exponentially many. A related idea to the complexity of the conformational search is the concept of 'rugged' energy landscapes [12]. Although optimal solutions are not guaranteed, robust, exploratory, non-deterministic search algorithms (e.g. simulated annealing [13] and genetic algorithms [14]) can locate good, near-optimal solutions, within a reasonable time. In the present paper, we focus on the design of a good conformational search strategy for the problem, leaving the discussion of the best choice of energy functions for protein structure prediction to a later paper.

Genetic algorithms were invented by John Holland [14] in his quest for a theory of adaptive processes. The concept was inspired by Darwin's evolutionary theory (loosely 'survival of the fittest') and in particular 'neo-Darwinism' according to which genetic recombinations and mutations play a dominant role in the evolution of a species. It is generally believed that Nature evolves so that individuals that are the best adapted to their co-evolving environment survive, while the poor ones die off; this is an example of optimization, more commonly referred to as 'adaptation' in biology [15]. Due to their highly nonlinear nature, genetic algorithms are difficult to analyse. There is no asymptotic global optimum coverage proof, as there is in simulated annealing [16], nor are there any general rules to design a genetic algorithm for a specific problem. However, their empirical success for the solution of numerous NP problems ([17] and references therein) warrants their further study.

Previous applications of genetic algorithms to the protein structure prediction problem [18–20, 3, 21–23] have not considered the genetic algorithm design issue. As in any problem, the simpler the algorithm is, the fewer the parameters it requires, the easier it is to understand and improve the performance. For this purpose we use a modified version of Goldberg's [24] Simple Genetic Algorithm (SGA), written in C. Dandekar and Argos [18–20] also use a version of the SGA but their work differs to ours in that our objectives are different. Firstly, they optimize the structural features of proteins (such as α -helices and β -strands) rather than adopting an energy minimization approach. Secondly, little

* A good introduction to the theory of NP-completeness and NP-hardness can be found in [8].

comparison was made with the performance of other algorithms. Schulze-Kremer [22] wrote an elaborate genetic algorithm to optimize real-space dihedral angles for a fully atomistic representation of proteins, based on CHARMM [25] energy minimization. Unfortunately, results were poor. Clearly, our principal research aim is to predict realistic structures as well, but only when a good method has been established. Unger and Moult [3,21] compared a Monte Carlo search with a 'genetic algorithm', using simple exact lattice models. However, their genetic algorithm is, strictly speaking, a hybrid genetic algorithm. It incorporates several Monte Carlo conformers with the occasional crossover between structures. For this reason, we call their approach a 'Genetic Algorithm-Monte Carlo' (GAMC). They compared this method with a Monte Carlo search and concluded that the genetic algorithm-Monte Carlo found lower energy solutions.

In this paper, we compare the Simple Genetic Algorithm with work by Unger and Moult [21]; Yue *et al.* [26] and Sali *et al.* [27] and determine guidelines for designing protein structure-prediction-genetic-algorithms.

The paper is structured in the following way. Section 2 highlights the conformational search issue and the need for a genetic algorithm approach. Following that, section 3 provides a description of the method for determining guidelines for genetic algorithm design. Section 4 discusses the Simple Genetic Algorithm used for the lattice conformational search. The HP-model [28] and REM-model [29] are described in section 5, as are the various methods used to minimize these potentials [26,3,21,27]. Lattice conformational search results from the SGA are compared with the other methods in section 6, while section 7 provides a discussion of the 'protein structure prediction-genetic algorithm' (PSP-GA) design principles that have emerged from this work.

2. WHY GENETIC ALGORITHMS FOR PROTEIN STRUCTURE PREDICTION?

Protein structure prediction is analytically difficult to solve. The problem is thought to stem from the exponential nature of the conformational search space. The number of conformations of a protein with N amino acid residues grows exponentially as γ^N where γ is the average number of conformations per residue (typically ~ 10). This suggests that an algorithm would require an exponential time to search the whole conformational space for the native state.

However, problems with exponentially growing search spaces are not new in physics (e.g. ideal gases) but many are solvable due to the symmetries and conservation laws that can be exploited. With proteins there is an added difficulty that the interactions are complex – it is not clear whether there are enough symmetries to reduce the problem to a tractable solution. In addition to this, proteins, although macromolecular, do not contain $\sim 10^{23}$ atoms to guarantee a valid statistical mechanical and thermodynamic treatment.

Furthermore, work by Unger and Moult [9]; Fraenkel [10]; Ngo and Marks [11] suggests that the protein structure prediction problem is NP-hard, that is computationally impossible to guarantee an exact solution. Bryngelson, Wolynes *et al.* [12], for example, advocate that the energy landscape of proteins must be ‘rugged’. This reflects the various energy barriers that have to be crossed and thus the hurdles that a conformational search algorithm must be able to deal with.

For these reasons ‘intelligent’ conformational search algorithms have become popular in structure prediction. Unlike gradient-based methods [30], which tend to terminate at local minima, genetic algorithms ‘hop’ around the conformational space independent of local derivatives. A selection process focuses the search in low energy areas, whereas a recombination stage maintains exploration of the search space.

3. DESIGNING GENETIC ALGORITHMS FOR PROTEIN STRUCTURE PREDICTION

A genetic algorithm is made up of 4 basic components: representation; selection; recombination and evaluation. Representation deals with formulating the specific problem as a digital string of parameters. This, combined with the evaluation function (in our case conformational energy) describes the optimization problem. The remaining two components, selection and recombination, provide the dynamics of the genetic algorithm search, which drive the population of solutions towards the global optimum. Within each unit, there are several options leading to numerous variations of genetic algorithms; some examples and corresponding parameters are listed in Table I. To determine guidelines for designing protein-structure-prediction-genetic-algorithms, we used the following principles:

1. A first approach should always be the simplest approach. In order to analyse the genetic algorithm dynamics, we used a genetic algorithm

TABLE I Examples of various options in designing a genetic algorithm

<i>GA Options and Parameters</i>
Population: static or variable size
Representation of solutions: bit string, reals, symbolic
Maximum number of generations or convergence criteria
Recombination operators: 1-pt crossover, uniform crossover, mutation, perturbation
Recombination probabilities: static, variable or dynamic
Selection methods: roulette, tournament, rank, elitism
Fitness scaling: linear with cut-off, quadratic, exponential

- with the simplest options and with the fewest parameters – a Simple Genetic Algorithm – to search lattice conformational space.
2. A systematic sample search of the parameter space was carried out to determine optimal parameter values for the SGA.
 3. Having calibrated the SGA in step 2, we compared the SGA conformational search ability using several test energy functions.
 4. Conformations, and their corresponding energies, generated by the SGA were compared with conformations predicted by other search methods.
 5. The time evolution of conformations generated by the SGA were observed. Minimal requirements and improvements for protein-structure-prediction-genetic-algorithms are proposed.

The merit of simple exact lattice models [28] is the ability to test ideas easily and suggest extrapolations to real systems. Simple exact models are ‘simple’ since only a few parameters are required, and ‘exact’ since physical properties can be calculated exactly. Lattice models, although unrealistic in appearance, provide several advantages over real-space models. From a folding dynamics point of view, they can explore long time behaviour; while from a structure prediction/optimization point of view, as in our case, they provide a valuable test-bed for protein structure optimizers.

Runs with various initial conditions were carried out to ensure that the genetic algorithm produced similar results each time.

4. THE SIMPLE GENETIC ALGORITHM FOR LATTICE CONFORMATIONAL SEARCH

The Simple Genetic Algorithm (SGA), as defined by Goldberg [24], is the simplest of all genetic algorithms. The original SGA manipulated binary

strings which encode a trial solution of the problem at hand. However, a major modification for searching protein conformations on a cubic lattice is to use a more natural representation for this problem. Since a simple cubic lattice is spatially restricting, a string of bond directions represents a folded chain of beads; each symbol corresponds to an increment or decrement in the appropriate Cartesian coordinate of the successive monomer beads (see Tab. II). A conformation of the polymer is then translated to a set of monomer positions $\mathbf{r}_i (i = 1, \dots, N)$ (where N is the number of monomer beads (residues)).

This representation has access to *all* 6^N lattice conformations, including all the non-physical, non-self-avoiding conformations. Thus the search task is a formidable one. A random population of conformations is generated and manipulated according to the genetic algorithm dynamics (selection and recombination). The population size, \mathcal{P} , is kept fixed. All individuals are replaced at each iteration, except for two copies of the current best conformation – this is known as ‘elitism’.

Selection is linearly proportional to fitness so that the probability, P_i , of selecting the i^{th} conformation, with a fitness value F_i , to propagate to the next time step is given by:

$$P_i = \frac{F_i}{\sum_{j=1}^{\mathcal{P}} F_j} \quad (1)$$

Probabilities must be positive so a linear mapping with a cut-off value is used to convert the energy (E) minimization to a fitness (F) maximization:

$$F_i = \begin{cases} -E_i & \text{if } E_i < 0; \\ 0 & \text{if } E_i \geq 0. \end{cases} \quad (2)$$

TABLE II Bond directions describing lattice conformations. A bond direction corresponds to a change, Δr , in one of the Cartesian coordinates of the successive monomer, keeping all other coordinates the same as the previous monomer

Direction	Δr
(U)P	$r_z \rightarrow r_z + 1$
(L)EFT	$r_x \rightarrow r_x - 1$
(F)RONT	$r_y \rightarrow r_y + 1$
(B)ACK	$r_y \rightarrow r_y - 1$
(R)IGHT	$r_x \rightarrow r_x + 1$
(D)OWN	$r_z \rightarrow r_z - 1$

Selected individuals, strings, are modified in a recombination process, to generate new solutions. Figure 1 shows a schematic representation of the one-point crossover and genewise mutation operators used by the SGA. These operators act stochastically on the selected individuals with fixed probabilities. In one-point crossover, a random crossover point is chosen for a pair of selected individuals, and the bond directions ('genes') are swapped up to the crossover point. Mutations act on a single individual and randomly change the value of bond directions along the string. The workings of the SGA are summarised in Figure 2.

5. TEST ENERGY FUNCTIONS

A. Random Energy Model

Originally used in spin glass theory, the Random Energy Model (REM) [31] was applied to protein folding by Bryngelson and Wolynes [32] and later formulated for lattice protein models by Shakhnovich and Gutin [29]. In the Random Energy Model, a protein is described by a fixed sequence of random interaction energies. Although not discussed here, the random interaction model lends itself for analytical studies of the coil-globule transition in proteins [29]. Specifically, the conformational energy is calculated as:

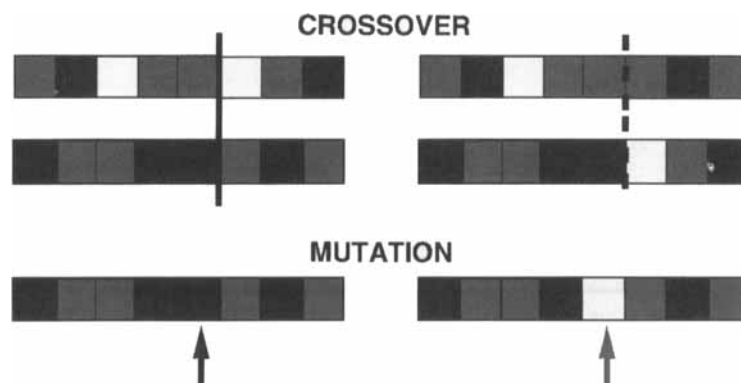


FIGURE 1 The top half represents the action of a one-point crossover operator. Below is an illustration of a single genewise mutation. In both cases the left side represents the 'before' situation, and the right side, the 'after' state.

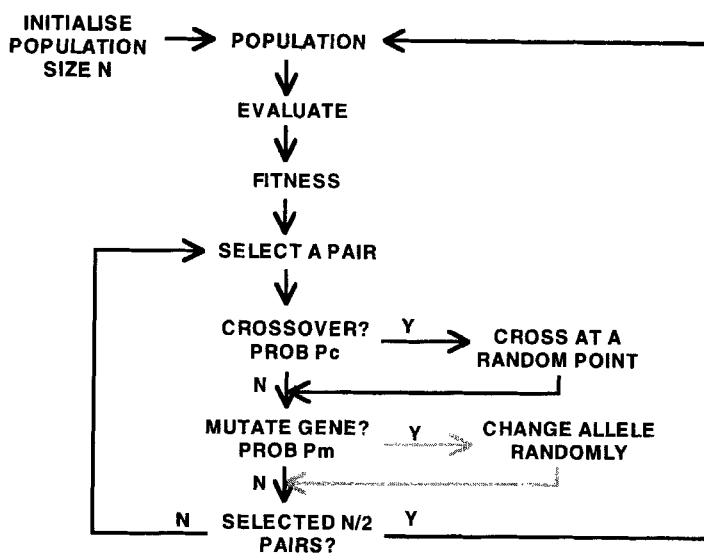


FIGURE 2 The simple genetic algorithm. A population of N conformers is initialised. These trial conformations are evaluated and selected, based on their fitness. Pairs of selected conformers are crossed over with a probability, P_c . Each bond direction ('gene') is then randomly mutated with a probability, P_m . A new population is selected in this manner and repeated for many generations (typically 2000).

$$\begin{aligned}
 E(\{\mathbf{r}_1 \dots \mathbf{r}_N\}) = & \sum_{i,j=1; i < j}^N B_{ij} \Delta(\mathbf{r}_i - \mathbf{r}_j) \\
 & + D_2 \sum_{i,j}^N \delta(\mathbf{r}_i - \mathbf{r}_j) \\
 & + D_3 \sum_{i,j,k}^N \delta(\mathbf{r}_i - \mathbf{r}_j) \delta(\mathbf{r}_j - \mathbf{r}_k)
 \end{aligned} \quad (3)$$

$$\Delta(\mathbf{r}_i - \mathbf{r}_j) = \begin{cases} 1 & \text{for } i \text{ and } j \text{ nearest neighbours} \\ 0 & \text{otherwise} \end{cases}$$

$$\delta(\mathbf{r}_i - \mathbf{r}_j) = \begin{cases} 1 & \text{if monomers } i, j \text{ occupy same site} \\ 0 & \text{otherwise} \end{cases}$$

where

$D_2 \equiv$ energetic penalty parameter for sites containing 2 or more monomers

$D_3 \equiv$ energetic penalty parameter for sites containing 3 or more monomers

$B_{ij} \equiv$ disorderly interaction energies

(energies are in units of $k_B T$ where k_B is Boltzmann's constant and T is the temperature in Kelvin)

The interaction matrix, B_{ij} , is symmetric but randomly generated with a Gaussian distribution:

$$P(B_{ij}) = (2\pi B^2)^{-1/2} \exp(-(B_{ij} - B_0)^2 / 2B^2) \quad (4)$$

The compactization observed in globule proteins is modelled in Eq. (3) using B_{ij} as a negative interaction potential with mean B_0 . B defines the spread, i.e. the standard deviation, of the compactization interactions. The greater the spread, the more heterogeneous the protein. A zero-spread corresponds to uniform interactions; this special case constitutes the Fixed Energy Model (FEM) which we have used for the larger polymers studied here (64 mer, 125 mer).

The last two terms in Eq. (3) represent excluded volume effects, that is they are energetic *penalties* for conformations with lattice sites occupied by more than two (D_2 term) or more than three (D_3 term) monomers. The objective of the genetic algorithm is to find conformations without multiple occupancies at a single site, by minimising this energy function.

B. HP-model

It is well known that correlations in the sequence of amino acid residues lead to substructures (secondary structures) common to all protein structures. These correlations can reduce the size of the 'alphabet' (code) from 20 symbols to a lesser number. The simplest and most interesting is the classification of residues into two types: H and P [33]. The energy function for this system favours interactions between HH monomer types and is indifferent to all other (PP and HP) interactions. This is known as the HP-model. It aims to highlight the importance of the hydrophobic effect in protein folding. In an aqueous medium, globular proteins tend to have a core of hydrophobic residues, surrounded by polar residues on the surface; in the HP-model, the H monomers correspond to hydrophobic residues that collapse to form a core surrounded by polar, P, monomers. Much work has been carried out by Dill and co-workers on this model [34, 35, 23, 26, 28].

The energy function for this model is:

$$\begin{aligned}
 E(\{\mathbf{r}_1 \dots \mathbf{r}_N\}) = & -|\varepsilon| \sum_{i,j=1; i < j}^N \Delta(\mathbf{r}_i - \mathbf{r}_j) \\
 & + \varepsilon_2 \sum_{i,j}^N \delta(\mathbf{r}_i - \mathbf{r}_j) \\
 & + \varepsilon_3 \sum_{i,j,k}^N \delta(\mathbf{r}_i - \mathbf{r}_j) \delta(\mathbf{r}_j - \mathbf{r}_k) \quad (5)
 \end{aligned}$$

$$\Delta(\mathbf{r}_i - \mathbf{r}_j) = \begin{cases} 1 & \text{if } i, j \text{ both H-type and nearest neighbours} \\ 0 & \text{otherwise} \end{cases}$$

$$\delta(\mathbf{r}_i - \mathbf{r}_j) = \begin{cases} 1 & \text{if monomers } i, j \text{ occupy same site} \\ 0 & \text{otherwise} \end{cases}$$

where

$|\varepsilon| \equiv$ strength of HH attraction (usually taken as 1)

$\varepsilon_2 \equiv$ energetic penalty parameter for sites containing two or more monomers

$\varepsilon_3 \equiv$ energetic penalty parameter for sites containing three or more monomers

Strictly speaking, the final two terms ($\varepsilon_2, \varepsilon_3$) in Eq. (5) are not included in the HP-model. We borrow these terms from the Random Energy Model to drive the conformational search towards self-avoiding conformations. This is necessary since *all* possible conformations are allowed; an energetic penalty is required to penalise conformations with multiple occupancies at a single site. For both models (Random Energy Model and HP), the penalty terms reduce to zero if the search is successful and a low energy self-avoiding conformation is found.

6. RESULTS

The first test function used was that from the Random Energy Model, Eq. (3). Shakhnovich *et al.* [27] used a Metropolis Monte Carlo algorithm with this energy function to find the native states of 27 mer heteropolymers. They interpreted the Monte Carlo 'folding' algorithm as modelling the folding dynamics of polymers. However, although Metropolis Monte Carlo methods asymptotically guarantee finding the thermal equilibrium state, it is unclear whether an interpretation beyond this has any validity. We view

their procedure as a protein structure optimization approach. These authors enumerated all compact (cubic) self-avoiding conformations, which allow them to determine the global minimum of each random 27 mer sequence generated. They reported [27] that three out of thirty sequences found the known global minimum; energies varied from -83.7 to -74.6 (in units of $k_B T$). Using the SGA, we found that three out of four sequences obtained compact, 100% cubic conformations (see Fig. 3), with energies ranging from -78.1 to -69.9 (units in $k_B T$). We cannot determine whether they are the global energy-minimum structures without carrying out a full enumeration; this is computationally time consuming due to the NP-complete nature of the problem, and, more importantly, unnecessary for our design purposes. Furthermore, we were unsuccessful in our correspondence with the authors and were unable to obtain the random interaction matrices specifically used in their work ([27]).

Longer polymers (64 mer, 125 mer) are more challenging since the conformational space grows exponentially with the polymer length. We continued to use the Random Energy Model with $B=0$. This corresponds to uniform interactions and thus guarantees that cubic conformations ($4 \times 4 \times 4$, $5 \times 5 \times 5$ respectively) occupy the multiply-degenerate ground states. The 64 mer reached 91% cubicity and the 125 mer runs found a conformation with 85% cubicity (Fig. 4). Since we are restricting our discussion to cubic lattices, it is more accurate to describe structures according to how 'cubic' they are, rather than using the more general term 'compactness'.

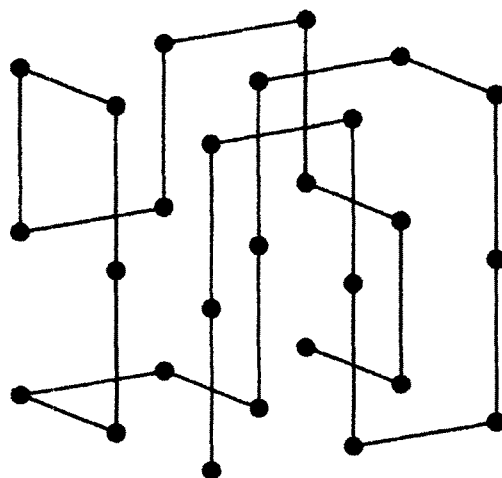
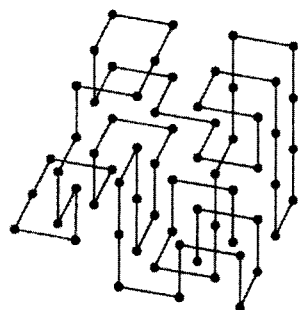


FIGURE 3 Example of a 100% cubic 27 mer conformation found by the SGA.

number of nearest neighbours=74



number of nearest neighbours=149

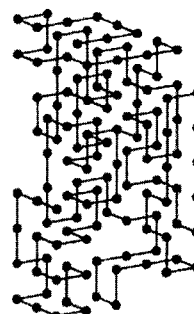


FIGURE 4 Left: 91% cubic conformation for a 64 mer homopolymer. Right: 85% cubic conformation for a 125 mer homopolymer.

It is unfortunate that 100% cubic conformations were not found; however, we are using the simplest genetic algorithm. Nevertheless, this initial exercise was useful to establish optimal values for the genetic algorithm parameters. The optimal genetic algorithm parameters for short polymers were: a minimum population size of 400; a 20% probability of crossover and a 4% probability of mutating a bond direction. Longer polymers required a larger minimum population size of 1000; a 90% probability of crossover and a 2% probability of mutation. Having established a good set of genetic algorithm parameters, the SGA was analyzed using the second test function, the HP-model.

Unger and Moult studied random HP-sequences of length 27 and 64 monomers [21]. They used two optimization methods: a Metropolis Monte Carlo (MC) method and a variation of the Monte Carlo method that incorporates a genetic algorithm (GAMC). The genetic algorithm-Monte Carlo method corresponds to a population of Metropolis Monte Carlo conformers which 'mix' between themselves through a crossover operation. The comparisons between these methods and our SGA are shown in Tables III and IV.

Our SGA beat Monte Carlo in 17 of the 20 test sequences and equalled it in finding low energy conformations for two sequences. On average, conformations generated by our SGA were 1.1 energy units lower than those found by the Monte Carlo method for the 27 mers and 8.3 units lower for the 64 mers.

When compared to the genetic algorithm-Monte Carlo method, the SGA performed on average as well as the genetic algorithm-Monte Carlo for the 27 mers; SGA conformational energies were on average 0.5 unit higher. In

TABLE III SGA comparisons with Unger and Moults 27 mer results. The sequence number corresponds to the number used by Unger and Moults to label their HP sequences. E_{SGA} is the lowest energy found by the SGA. The 'num steps' column reports the number of energy evaluations carried out by the SGA to reach the lowest energy state. ΔE_{MC} is the energy difference between the lowest energies found by the SGA and Unger and Moults Monte Carlo procedure: $\Delta E_{MC} = E_{SGA} - E_{MC}$. Similarly, ΔE_{GAMC} is the energy difference between the SGA and Unger and Moults GAMC method

sequence	E_{SGA}	Num Steps	ΔE_{MC}	ΔE_{GAMC}
273d.1	-8	2.9E+04	-1	1
273d.2	-8	2.1E+04	1	1
273d.3	-8	1.1E+05	-2	0
273d.4	-15	2.7E+05	-4	0
273d.5	-7	9.6E+03	0	1
273d.6	-11	9.6E+04	-2	0
273d.7	-11	2.2E+04	-1	1
273d.8	-4	8.6E+04	0	0
273d.9	-7	6.0E+04	-1	0
273d.10	-10	5.6E+04	-1	1
AVERAGE	-8.9	7.6E+04	-1.1	0.5

TABLE IV SGA comparisons with Unger and Moults 64 mer results. The sequence number corresponds to the number used by Unger and Moults to label their HP sequences. E_{SGA} is the lowest energy found by the SGA. The 'num steps' column reports the number of energy evaluations carried out by the SGA to reach the lowest energy state. ΔE_{MC} is the energy difference between the lowest energies found by the SGA and Unger and Moults Monte Carlo procedure: $\Delta E_{MC} = E_{SGA} - E_{MC}$. Similarly, ΔE_{GAMC} is the energy difference between the SGA and Unger and Moults GAMC method

sequence	E_{SGA}	Num Steps	ΔE_{MC}	ΔE_{GAMC}
643d.1	-21	6.4E+05	-9	6
643d.2	-26	1.5E+06	-9	3
643d.3	-36	1.7E+06	-12	-1
643d.4	-30	1.9E+06	-12	4
643d.5	-28	6.3E+05	-8	4
643d.6	-22	4.6E+05	-6	7
643d.7	-17	2.1E+05	-2	3
643d.8	-28	1.4E+06	-9	1
643d.9	-29	9.8E+05	-10	3
643d.10	-20	3.1E+05	-6	4
AVERAGE	-25.7	9.8E+05	-8.3	3.4

the 64 mer case, SGA conformations were on average 3.4 energy units higher than the conformational energies found by the genetic algorithm-Monte Carlo. There was one 64 mer sequence for which the SGA found a lower energy conformation than the genetic algorithm-Monte Carlo. An

important note is the speed at which the SGA found low energy conformations. Only a fraction of the total number of steps were required for the SGA when compared to the Monte Carlo and genetic algorithm-Monte Carlo. For example, for the 27 mer, to reach a solution of comparable quality, the number of steps required by the SGA was 3% of the number of steps required by the Monte Carlo method, and 4% for the genetic algorithm-Monte Carlo.

Further studies were carried out using HP-sequences taken from Yue *et al.* [26]. They designed ten 48 mer sequences and determined the native conformational states using a constraint-based hydrophobic core construction method. This method determines the global minima of HP-sequences by constructing conformations with a core of H (hydrophobic) residues that also minimize the surface area of the conformation. The difference in the conformational energies found by the SGA and the native state energy is labelled as ΔE_N . Yue *et al.* also used a conformational search algorithm for HP-sequences called a 'hydrophobic zipper' (HP). In this case, the H monomers attract nearby H monomers and bring them together in a process akin to nucleation. The energy difference between conformations found by this method and the SGA is denoted as ΔE_{HZ} . The comparisons are summarised in Table V. Conformations found by the

TABLE V SGA comparisons with Yue and Dill's 48 mer studies. Sequence number corresponds to the number used by Yue *et al.* to label their HP sequences. E_{SGA} is the lowest energy found by the SGA. The 'num steps' column reports the number of energy evaluations carried out by the SGA to reach the lowest energy state. ΔE_{HZ} is the energy difference between the lowest energies found by the SGA and Yue *et al.*'s hydrophobic zipper method: $\Delta E_{HZ} = E_{SGA} - E_{HZ}$. Similarly, ΔE_N is the energy difference between the SGA and the native state energy found by Yue *et al.* using the constraint-based hydrophobic core construction method

sequence	E_{SGA}	# Steps	ΔE_{HZ}	ΔE_N
483d.81	-24	1.6E+06	7	8
483d.82	-24	4.7E+05	8	10
483d.83	-23	1.9E+05	7	10
483d.84	-24	8.6E+05	7	10
483d.85	-28	2.4E+05	2	4
483d.86	-25	5.5E+05	4	7
483d.87	-27	3.8E+05	2	5
483d.88	-26	4.2E+05	3	5
483d.89	-27	1.3E+05	4	7
483d.90	-26	7.0E+04	3	7
AVERAGE	-25.4	4.9E+05	9.4	7.3

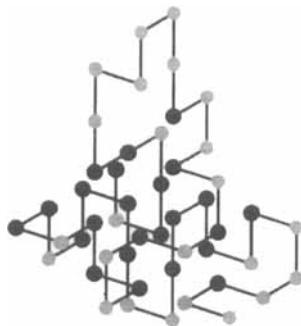


FIGURE 5 Example of a compact HP conformation, with a hydrophobic core, found by the SGA.

SGA were on average 6.0 energy units higher. An example of a compact conformation by the SGA is shown in Figure 5. In no cases did the SGA equal or beat conformational energies obtained by the hydrophobic zipper method.

Runs took from several minutes for short polymers (27 mers, 48 mers) to hours for longer cases (64 mers, 125 mers) on a DEC 3000 Alpha workstation.

7. DISCUSSION

The conformations found by the SGA are not promising in themselves. However, the aim of the study is to analyse the simple genetic algorithm and determine what factors are important in designing protein-structure-prediction-genetic-algorithms. What can we learn from these results about protein-structure-prediction-genetic-algorithms?

It is clear from the 27 mer studies that the SGA performs well for short polymers and tends to find compact, low energy structures. However, as was evident in our initial investigation of 64 mers and 125 mers using the Random Energy Model, the SGA performs below average. Although the SGA method was better than the Monte Carlo method, it did not perform as well as the genetic algorithm-Monte Carlo and was even worse than the hydrophobic zipper algorithm. To some extent, it is not surprising that the hydrophobic zipper method performed better than the SGA. Hydrophobic zipper is a specialised search algorithm for the HP-model and generates conformations by explicitly forming H-H contacts. It is not uncommon for

specialised algorithms to solve certain cases of an NP-complete problem – however, a *general* solution remains difficult [36].

In general the SGA finds it hard to generate compact conformations for the longer polymers – why? We investigated this further using the HP-model. The best way to analyse the dynamics of the SGA is to observe its time evolution rather than merely the end result. Thus, the lowest energy conformations were observed at various time points. In several cases, the SGA produced two clusters of residues with hydrophobic (H) cores connected by a ‘thread’ – similar to loop regions in real proteins. It is promising that such conformations (see Fig. 6) are possible since proteins with sub-structures connected by loop regions are common in Nature. However, this is not the lowest energy conformation for the HP-model, so we conclude that the clustering is due to restrictions imposed by the genetic algorithm dynamics, that is to selection and recombination. Selection is an important aspect of

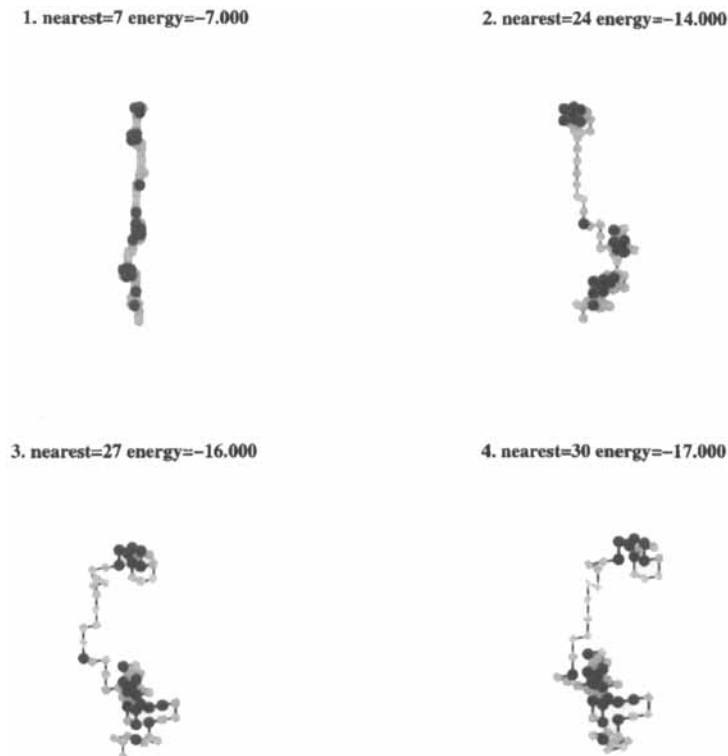


FIGURE 6 Evolution of the best conformation per generation: formation of clusters in the Simple Genetic Algorithm. Dark = H = hydrophobic, Light = P = polar.

the genetic algorithm dynamics but we do not think it is the reason for the occurrence of clustered solutions; if a lower energy globule existed in the population, the selection function would have assigned it a high probability and elitism would guarantee its selection into the next generation.

The problem must therefore lie in the recombination stage: it appears that the one-point crossover and mutation operations cannot inter-digitate the clusters successfully. We believe that these recombination operators would fold the clusters *into* each other creating many sites with multiple occupancies; these are penalised by the excluded volume terms in Eq. (5) heavily reducing the ‘survival probability’ of such structures. The difficulty appears to be in the formation of only two large clusters, or ‘building blocks’; we think that more clusters but smaller in size would be easier to manipulate. How do we promote this in the genetic algorithm dynamics?

Genetic algorithms manipulate partial solutions in their search for the overall optimal solution [14, 24]. These partial solutions or ‘building blocks’ correspond to sub-strings of a trial solution – in our case local sub-structures within the overall conformation. Clearly, the level of crossover influences the size of the building blocks. A multi-point crossover would generate smaller building blocks (a higher ‘resolution’) and consequently smaller clusters that should facilitate their inter-digitation. In the case of the 27 mer, the one-point crossover generates building blocks of 13.5 monomers, on average. Based on this argument, 48 mers and 64 mers require at least two to three point crossovers to reduce the size of the building blocks to ~ 10 . Furthermore, the local perturbation dynamics of the Monte Carlo methods aids the local ‘fine tuning’ of conformations as manifested by the genetic algorithm-Monte Carlo method. It is believed that the two recombination operators of multi-point crossover and a local perturbation are required for any protein-structure-prediction-genetic-algorithm to be fully effective. Work is under way to design a protein-structure-prediction-genetic-algorithm to optimize real protein structures (rather than lattice models) which includes these requirements.

One further problem with genetic algorithms is ‘knowing when to stop’. Most optimization algorithms deal with a single solution at a time and decide to stop when there has been no change in the cost or energy function for a successive number of steps. Since genetic algorithms deal with an ensemble of solutions, a quantity analogous to the statistical mechanical free energy is used. The ‘population free energy’, F , is calculated from its ‘partition function’, Z :

$$Z = \sum_{i=1}^N e^{-E_i} \quad (6)$$

where the sum is over the total number of conformers in a population and E_i is the energy of the i^{th} conformer. Hence,

$$F = -\ln(Z) \quad (7)$$

This approach is advantageous over using the mean energy of the population in two ways. Firstly, the mean energy fluctuates around the equilibrium making it difficult to use as a stop criterion, and secondly, F contains more information on the shape of the energy distribution, including, in particular, its 'entropy'. F appears to play the role of a Lyapounov function in the genetic algorithm dynamics – convergence is synonymous with its minimal value. Figure 7 plots the energy distributions of the population at various time steps. The plot shows very clearly that the genetic algorithm dynamics converge the population of conformations to an equilibrium distribution. This is characterised by F as shown in Figure 8.

In conclusion, the genetic algorithm approach to the protein structure prediction problem offers a promising potential method of solution. Genetic algorithms are fast and efficient at searching the rugged conformational landscapes presented by protein molecules. We have established some

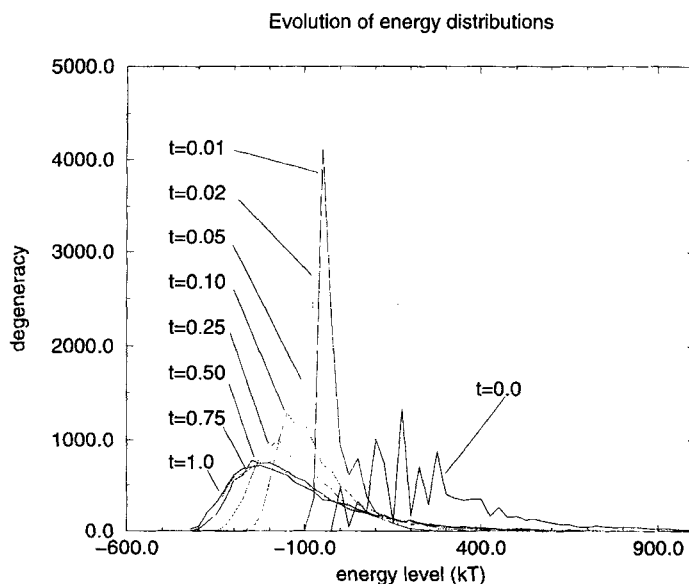


FIGURE 7 Evolution of a population: plotting the energy distributions at various time steps. t is the percentage of the total run-time elapsed. The population converges within 50% of the run time.

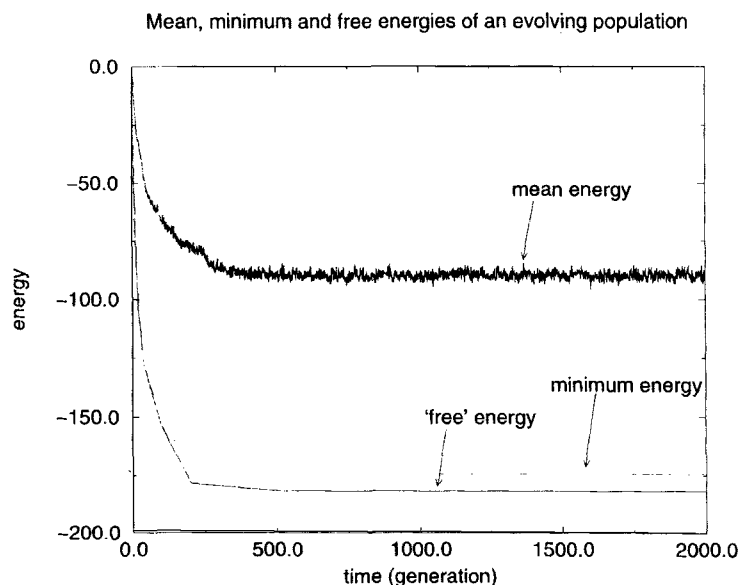


FIGURE 8 Evolution of the energies of a population. The free energy, which characterises the energy distribution of a population, reaches a convergence, or equilibrium point.

guidelines for designing protein-structure-prediction-genetic-algorithms in this paper and are currently implementing them in an improved genetic algorithm to search for realistic protein structures.

Acknowledgements

MK would like to thank Paul van der Schoot and Steven Brenner for interesting discussions. The authors would also like to thank Ken Dill and Kaizhi Yue for various information. MK is financially supported by an EPSRC award.

References

- [1] Anfinsen, C. B. (1973). "Principles that govern the folding of proteins", *Science*, **181**, 187.
- [2] Karplus, M. and Shakhnovich, E. (1992). "Protein Folding: Theoretical Studies of Thermodynamics and Dynamics", in *Protein Folding*, Creighton, T. E., ed., W. H. Freeman and Company, Chapter 4.
- [3] Unger, R. and Moult, J. (1993). "Genetic Algorithms for Protein Folding Simulations", *J. Mol. Biol.*, **231**, 5967.
- [4] Moult, J., Pedersen, J., Judson, R. and Fidelis, K. (1995). "A Large Scale Experiment to Assess Protein Structure Prediction Methods", *Proteins: Struct. Func. Gen.*, **23**, ii.
- [5] Taylor, W. R. (1988). "Pattern matching methods in protein sequence comparison and structure prediction", *Prot. Eng.*, **2**, 77.

- [6] Garnier, J., Osguthorpe, J. D. and Robson, B. (1978). "Analysis of the accuracy and implications of simple methods for predicting the secondary structure of globular proteins", *J. Mol. Biol.*, **120**, 97.
- [7] Levinthal, C. (1968). "Are there pathways for protein folding?", *J. Chem. Phys.*, **65**, 44.
- [8] Garey, M. R. and Johnson, D. S. (1979). *Computers and Intractability: A Guide to the Theory of NP-completeness*, W. H. Freeman and Company, San Francisco.
- [9] Unger, R. and Moulton, J. (1993). "Finding the Lowest Free Energy Conformation of a Protein is an NP-hard Problem: Proof and Implications", *Bull. Math. Biol.*, **55**, 1183.
- [10] Fraenkel, A. (1993). "Complexity of Protein Folding", *Bull. Math. Biol.*, **55**, 1199.
- [11] Ngo, J. T. and Marks, J. (1992). "Computational complexity of a problem in molecular structure prediction", *Prot. Eng.*, **5**, 313.
- [12] Bryngelson, J., Onuchic, J., Socci, N. and Wolynes, P. (1995). "Funnels, Pathways and the Energy Landscape of Protein Folding: a Synthesis", *Proteins: Struct. Func. Gen.*, **21**, 167.
- [13] Kirkpatrick, S., Gelatt, C. D. and Vecchi, M. P. (1983). "Optimization by Simulated Annealing", *Science*, **220**, 671.
- [14] Holland, J. (1975). *Adaptation in Natural and Artificial Systems*, University of Michigan Press.
- [15] Coveney, P. and Highfield, R. (1995). *Frontiers of Complexity*, Faber and Faber.
- [16] Aarts, E. and Korst, J. (1989). *Simulated Annealing and Boltzmann Machines*, John Wiley and Sons.
- [17] Davis, L. (ed.) (1991). *Handbook of Genetic Algorithms*, Van Nostrand Reinhold.
- [18] Dandekar, T. and Argos, P. (1992). "Potential of genetic algorithms in protein folding and protein engineering simulations", *Prot. Eng.*, **5**, 637.
- [19] Dandekar, T. and Argos, P. (1994). "Folding the Main Chain of Small Proteins with the Genetic Algorithm", *J. Mol. Biol.*, **236**, 844.
- [20] Dandekar, T. and Argos, P. (1996). "Identifying the Tertiary Fold of Small Proteins with Different Topologies from Sequence and Secondary Structure using the Genetic Algorithm and Extended Criteria Specific for Strand Regions", *J. Mol. Biol.*, **256**, 645.
- [21] Unger, R. and Moulton, J. (1993). "Genetic Algorithms for 3D Protein Folding Simulations", in *Proc. of the Fifth Intl. Conf. on Gen. Algo.*, Forrest, S., ed., Morgan Kaufmann, p. 581.
- [22] Schulze-Kremer, S. (1992). "Genetic Algorithms for Protein Tertiary Structure Prediction", in *Parallel Problem Solving from Nature 2*, Manner, R. and Manderick, B., eds., Elsevier Science Publishers, p. 391.
- [23] Sun, S., Thomas, P. D. and Dill, K. A. (1995). "A simple protein folding algorithm using a binary code and secondary structure constraints", *Prot. Eng.*, **8**, 769.
- [24] Goldberg, D. (1989). *Genetic Algorithms in Search, Optimization and Machine Learning*, Addison-Wesley.
- [25] Brooks, B. R., Brucoleri, R. E., Olafson, B. D., Sates, D., Swaminathan, S. and Karplus, M. (1983). "CHARMM: A Program for Macromolecular Energy, Minimization and Dynamics Calculations", *J. Comp. Chem.*, **4**, 187.
- [26] Yue, K., Fiebig, K. M., Thomas P. D., Chan, H. S., Shakhnovich, E. and Dill, K. A. (1995). "A test of lattice protein folding algorithms", *Proc. Natl. Aca. Sci. USA*, **92**, 325.
- [27] Shakhnovich, E., Farztdinov, G., Gutin, A. and Karplus, M. (1991). "Protein Folding Bottlenecks: A Lattice Monte Carlo Simulation", *Phys. Rev. Lett.*, **67**, 1665.
- [28] Dill, K. A., Bromberg, S., Yue, K. et al. (1995). "Principles of Protein Folding – A perspective from simple exact models", *Prot. Sci.*, **4**, 561.
- [29] Shakhnovich, E. and Gutin, A. (1989). "Formation of unique structure in polypeptide chains: Theoretical investigation with the aid of a replica approach", *Biophys. Chem.*, **34**, 187.
- [30] Mackay, D. H. J., Cross, A. J. and Hagler, A. T. (1990). "The Role of Energy Minimization in Simulation Strategies of Biomolecular Systems", in *Prediction of protein structure and the principles of protein conformation*, Fasman, G. D., ed., Plenum Press.
- [31] Derrida, B. (1980). "Random Energy Model: Limit of a Family of Disordered Models", *Phys. Rev. Lett.*, **45**.
- [32] Bryngelson, J. and Wolynes, P. (1987). "Spin Glasses and the Statistical Mechanics of Protein Folding", *Proc. Natl. Aca. Sci. USA*, **84**, 7524.
- [33] Lau, K. F. and Dill, K. A. (1989). "A Lattice Statistical Mechanics Model of the Conformational and Sequence Spaces of Proteins", *Macromol.*, **22**, 3986.

- [34] Chan, H. S. and Dill, K. A. (1991). "Polymer Principles in Protein Structure and Stability", *Ann. Rev. Biophys. Biophysic. Chem.*, **20**, 447.
- [35] Yue, K. and Dill, K. A. (1993). "Sequence-structure relationships in proteins and copolymers", *Phys. Rev. E*, **48**, 2267.
- [36] Lawler, E. L., Lenstra, J. K., Rinnooy Kan, A. H. G. and Shmoys, D. B., eds. (1985). *The Travelling Salesman Problem*, John Wiley and Sons.