Novel Heuristic Search Methods for Protein Folding and Identification of Folding Pathways

by

Alena Shmygelska

B.Sc. Computer Science, Slippery Rock University of Pennsylvania, 2001
B.A. Biology, Slippery Rock University of Pennsylvania, 2001
Minors in Physics and Mathematics, Slippery Rock University of Pennsylvania, USA, 2001

A THESIS SUBMITTED IN PARTIAL FULFILMENT OF
THE REQUIREMENTS FOR THE DEGREE OF

Doctor of Philosophy

in

The Faculty of Graduate Studies

(Computer Science)

The University of British Columbia

September 2006

Abstract

Proteins form the very basis of life. If we were to open up any living cell, we would find, apart from DNA and RNA molecules whose primary role is to store genetic information, a large number of different proteins that comprise the cell itself (for example the cell membrane and organelles), as well as a diverse set of enzymes that catalyze various metabolic reactions. If enzymes were absent, the cell would not be able to function, since a number of metabolic reactions would not be possible. Functions of proteins are the consequences of their functional 3D shape. Therefore, to control these versatile properties, we need to be able to predict the 3D shape of proteins; in other words, solve the protein folding problem. The prediction of a protein's conformation from its amino-acid sequence is currently one of the most prominent problems in molecular biology, biochemistry and bioinformatics.

In this thesis, we address the protein folding problem and the closely-related problem of identifying folding pathways. The leading research objective for this work was to design efficient heuristic search algorithms for these problems, to empirically study these new methods and to compare them with existing algorithms.

This thesis makes the following contributions: (1) we show that biologically inspired approaches based on the notion of *stigmergy* – where a collection of agents modifies the environment, and those changes in turn affect the decision process of each agent (particularly artificial colonies of ants that give rise to such properties as self-organization and cooperation also observed in proteins) is a promising field of study for the protein folding problem; (2) we develop a novel adaptive search framework that is used to identify and to bin promising candidate solutions and to adaptively retrieve solutions when the search progress is unsatisfactory; (3) we develop a new method that efficiently explores large search neighbourhoods by performing biased iterated solution construction for identifying folding pathways; and (4) we show that our algorithms efficiently search the vast search landscapes encountered and are able to capture important aspects of the process of protein folding for some widely accepted computational models.

Contents

Abstract ii

Contents iii

List of Tables vi

List of Figures xii

Abbreviations xxiii

Acknowledgements xxv

Dedications xxvi

1 Introduction 1
- 1.1 Structure and Function of Proteins 1
  - 1.1.1 Three Levels of Protein Structure 3
  - 1.1.2 Protein Functions 4
- 1.2 The Protein Folding Problem and its Importance 5
  - 1.2.1 Protein Folding Paradoxes 5
  - 1.2.2 The Thermodynamic Hypothesis 6
  - 1.2.3 The Central Dogma of Protein Folding 6
  - 1.2.4 Forces in Protein Folding 7
  - 1.2.5 Motivations for Studying Protein Folding Problems 8
- 1.3 An Overview of Computational Approaches to Protein Folding . . 9
  - 1.3.1 Homology Modeling 10
  - 1.3.2 Threading 10
  - 1.3.3 Novel Fold Recognition Methods 10
  - 1.3.4 *Ab initio* Methods 11
- 1.4 Thesis Statement 11
- 1.5 Organization of the Thesis 12

2 Description of Problems, Models, and Notations 14
2.1 Protein Folding on the Lattice 16
2.1.1 Square and Cubic Lattices 16
2.1.2 Hydrophobic Polar Energy Potential 16
2.1.3 Face-Centered Cubic Lattice Model 17
2.1.4 Beta Sheet Energy Potential Used with the FCC Lattice . . 18
2.2 Off-Lattice Protein Folding 20
2.2.1 Models for Off-Lattice Protein Folding 20
2.2.2 Energy Potentials for Off-Lattice Protein Tertiary Structure Prediction 22
2.3 The Problem of Folding Pathway Identification 23
2.3.1 Models Used for the Identification of Folding Pathways . 24
2.3.2 Objective Functions Used for the Identification of Folding Pathways 24

3 Background and Related Work 26
3.1 The Protein Folding Problem 26
3.1.1 Search Algorithms Used for Protein Folding 28
3.1.2 Summary and Classification of Search Methods 36
3.1.3 An Overview of Existing Research in 2D and 3D Hydrophobic Polar Folding 39
3.1.4 An Overview of Existing Research in FCC β-Sheet Protein Folding 41
3.2 The Problem of Identifying Folding Pathways 43
3.2.1 Theoretical and Experimental Work on Protein Folding Pathways 44
3.2.2 An Overview of Computational Approaches for Identifying Folding Nuclei from the Native Conformation 45

4 An Ant Colony Optimization for 2D and 3D Hydrophobic Polar Protein Folding 47
4.1 Description of the Algorithm 48
4.1.1 Construction Phase, Pheromone, and Heuristic Values . . 51
4.1.2 Local Search 52
4.1.3 Update of the Pheromone Values 53
4.2 Empirical Results and Discussion 54
4.2.1 Results for Standard Benchmark Instances 55
4.2.2 Result for New Biological and Random Data Sets 57
4.2.3 Characteristic Performance Differences between ACO and PERM 61

4.2.4 Discussion of ACO Results 71
4.3 Summary 76

5 Adaptive Bin Framework Search, Introduced for the FCC β-Sheet Protein Folding Problem 77
5.1 The Bin Framework and the Bin Framework Monte Carlo Algorithm 78
5.1.1 Storing Conformations in the Bin Framework 82
5.1.2 Retrieving Conformations from the Bin Framework 87
5.2 Empirical Results and Discussion 88
5.2.1 Comparison of Results with the Literature 89
5.2.2 Further Comparison for Homopolymers of Length 12, 24, 32, and 64 90
5.2.3 Discussion of the Bin Framework 99
5.3 Summary 106

6 Construction Search for Identifying Folding Pathways 108
6.1 Description of the Problem 108
6.2 Description of the Algorithm 109
6.2.1 Generation of the Polymer Graph 110
6.2.2 Sampling of Folding Pathways 110
6.2.3 Collective Analysis of Low Effective Contact Order Pathways 113
6.3 Empirical Results and Discussion 114
6.4 Summary 121

7 Conclusions and Future Directions 127
7.1 ACO for 2D and 3D Hydrophobic Polar Folding 127
7.2 Adaptive Bin Framework for the FCC $\beta-$ Sheet Model 128
7.3 Construction Search for Identification of Folding Pathways 130
7.4 Summary 130

Bibliography 134

A Ant Colony Optimization 145

B Adaptive Bin Framework Monte Carlo Search 152

Index 168

List of Tables

4.1 Benchmark instances for the 2D and 3D HP Protein Folding Problem used in this study with optimal or best known energy values E^*. Most instances for 2D and 3D HP can also be found at http://www.cs.sandia.gov web site; Sequence S1-9 (2D) is taken from [67], and the last two instances (2D) are from [110]. H_i and P_i indicate a string of i consecutive H's and P's, respectively; likewise, $(s)_i$ indicates an i-fold repetition of string s. 56

4.2 Comparison of the solution quality obtained in 2D by the evolutionary algorithm of Unger and Moult (EA) [135], the evolutionary Monte Carlo algorithm of Liang and Wong (EMC) [77], the Multi-Self-Overlap Ensemble algorithm of Chickenji *et al.* (MSOE) [24], the pruned-enriched Rosenbluth method (PERM) and ACO. For EA and EMC, the reported energy values are the lowest among five independent runs, and the values in parentheses are the numbers of valid conformations scanned before the lowest energy values were found. Missing entries indicate cases where the respective method has not been tested on a given instance. The CPU times reported in parentheses for MSOE were determined on a 500 MHz CPU, and those for PERM and ACO are based on $100 - 200$ runs per instance on our reference 2.4 GHz Pentium IV machine. The energy values shown in bold face correspond to currently best-known solution qualities. 58

4.3 Comparison of the solution quality obtained in 3D by the hydrophobic zipper (HZ) algorithm [33], the constraint-based hydrophobic core construction method (CHCC) [150], the core-directed chain growth algorithm (CG) [13], the contact interactions (CI) algorithm [132], the pruned-enriched Rosenbluth method (PERM) and ACO. For CI, only the best energies obtained are shown. For HZ, CHCC and CG, the reported CPU times are taken from [13]; these are the expected times for finding optimal solutions on a Sparc 1 workstation. In the case of HZ, the reported CPU times are based on an extrapolation from the measured times required for finding suboptimal conformations with the energy values listed here. The CPU times for PERM and ACO were determined on our reference 2.4 GHz Pentium IV machine based on $50-100$ runs per instance. The energy values shown in bold face correspond to currently best-known solution qualities. 59

5.1 Comparison of the solution quality obtained for the homopolymer of length $N = 64$ by the Monte Carlo Simulated Annealing (MCSA) [52], the Replica Exchange Monte Carlo (REMC) with a linear set of temperatures [52] and the Parallel-hat Tempering algorithm (PHAT) [154] with our implementation of Monte Carlo (MC), REMC and PHAT and our new Bin Framework Monte Carlo (BINMC). The time reported for MCSA and REMC from [52] is the estimated time required to run on 2.4 GHz machines (in the original paper authors used 500 MHz processor, therefore, reported times in [52] are conservatively divided by a factor of 4.8). The time reported for Parallel-hat tempering [154] is the estimated time to run on 2.4 GHz as well (in the original paper [154] CPU of 750 MHz was used, therefore we conservatively applied a factor of 3.2). The authors in [52] also implemented REMC with an exponential set of temperatures. The exact temperatures were not specified in the paper [52], however, and the authors could not recall it in personal communication (the lowest energy observed in this case was -374). The number of runs used for comparison was 10 for all algorithms. In the last column of the table, we report p-values indicating the probability that the null hypothesis of no difference between the mean energies reached over 10 runs for BINMC and a particular algorithm listed (within the same CPU cut-off time) is true; we used the Mann-Whitney U test to calculate p-values [59]; * indicates that p-values are below the significance level of 0.05 of wrongly rejecting the null hypothesis. 91

5.2 Comparison of the average solution quality obtained and the average time required for the homopolymers of lengths $N = 12, 24, 32$ for the re-implemented MC, REMC, PHAT, and BINMC. The time cut-off used was 1 hr on 2.4 GHz reference machine, and the averages were calculated from 10 independent runs. Temperature sets used for the re-implemented algorithms are the same as in Table 5.1. In the last column of the table we report p-values indicating the probability that the null hypothesis of no difference between the mean CPU run-time (for the homopolymer of length 24) or the mean energies reached over 10 runs (for the homopolymer of length 32) for BINMC and a particular algorithm listed is true; we used the Mann-Whitney U test to calculate p-values [59]; * indicates that p-values are below the significance level of 0.05 of wrongly rejecting the null hypothesis. 96

5.3 Comparison of the solution quality obtained for the homopolymers of length $N = 64$ and $N = 32$ by re-implemented MC, REMC with the linear set of temperatures, PHAT, and our new BINMC on 2.4 GHz reference machine with a time cut-off of 10 hrs over 10 independent runs. In the last column of the table, we report p-values indicating the probability that the null hypothesis of no difference between the mean energies reached over 10 runs for BINMC and a particular algorithm listed (within the same CPU cut-off time) is true; we used the Mann-Whitney U test to calculate p-values [59]; * indicates that p-values are below the significance level of 0.05 of wrongly rejecting the null hypothesis. 97

6.1 Set of proteins used in this study. The kinetic data about folding pathways of these proteins is available in [76] and [107]. 126

A.1 Biological sequences of length ≈ 30 145
A.2 Performance comparison of PERM and ACO on biological sequences of length ≈ 30 in 2D 146
A.3 Performance comparison of PERM and ACO on biological sequences of length ≈ 30 in 3D 146
A.4 Biological sequences of length ≈ 50 147
A.5 Performance comparison of PERM and ACO on biological sequences of length ≈ 50 in 2D 147
A.6 Performance comparison of PERM and ACO on biological sequences of length ≈ 50 in 3D 148
A.7 Random sequences of length 30 148
A.8 Performance comparison of PERM and ACO on random sequences of length 30 in 2D 149
A.9 Performance comparison of PERM and ACO on random sequences of length 30 in 3D 149
A.10 Random sequences of length 50 150
A.11 Performance comparison of PERM and ACO on random sequences of length 50 in 2D 150
A.12 Performance comparison of PERM and ACO on random sequences of length 50 in 3D 151

B.1 Vectors for the best found conformation of 64 amino acids (total energy $= -391$, short-range energy $= -212$, long-range energy $= -179$). 153

B.2 Continued, vectors for the best found conformation of 64 amino acids (total energy = −391, short-range energy = −212, long-range energy = −179). 154
B.3 Triplets of vectors, angles: θ_1 (between vectors $\mathbf{v_{i-1}}$ and $\mathbf{v_i}$), θ_2 (between vectors $\mathbf{v_i}$ and $\mathbf{v_{i+1}}$), θ_3 (between vectors $\mathbf{v_{i-1}}$ and $\mathbf{v_{i+1}}$), short-range energy contributions for the best found conformation of 64 amino acids (total energy = −391, short-range energy = −212, long-range energy = −179). 155
B.4 Continued, triplets of vectors, angles: θ_1 (between vectors $\mathbf{v_{i-1}}$ and $\mathbf{v_i}$), θ_2 (between vectors $\mathbf{v_i}$ and $\mathbf{v_{i+1}}$), θ_3 (between vectors $\mathbf{v_{i-1}}$ and $\mathbf{v_{i+1}}$), short-range energy contributions for the best found conformation of 64 amino acids (total energy = −391, short-range energy = −212, long-range energy = −179). 156
B.5 Long-range interactions for the best found conformation of 64 amino acids (total energy = −391, short-range energy = −212, long-range energy = −179). 157
B.6 Continued, long-range interactions for the best found conformation of 64 amino acids (total energy = −391, short-range energy = −212, long-range energy = −179). 158
B.7 Continued, long-range interactions for the best found conformation of 64 amino acids (total energy = −391, short-range energy = −212, long-range energy = −179). 159
B.8 Continued, long-range interactions for the best found conformation of 64 amino acids (total energy = −391, short-range energy = −212, long-range energy = −179). 160
B.9 Vectors for the best found conformation of 32 amino acids (total energy = −161, short-range energy = −112, long-range energy = −49). 161
B.10 Triplets of vectors, angles: θ_1 (between vectors $\mathbf{v_{i-1}}$ and $\mathbf{v_i}$), θ_2 (between vectors $\mathbf{v_i}$ and $\mathbf{v_{i+1}}$), θ_3 (between vectors $\mathbf{v_{i-1}}$ and $\mathbf{v_{i+1}}$), short-range energy contributions for the best found conformation of 32 amino acids (total energy = −161, short-range energy = −112, long-range energy = −49). 162
B.11 Long-range interactions for the best found conformation of 32 amino acids (total energy = −161, short-range energy = −112, long-range energy = −49). 163
B.12 Vectors for the best found conformation of 24 amino acids (total energy = −109, short-range energy = −68, long-range energy = −41). 164

B.13 Triplets of vectors, angles: θ_1 (between vectors $\mathbf{v_{i-1}}$ and $\mathbf{v_i}$), θ_2 (between vectors $\mathbf{v_i}$ and $\mathbf{v_{i+1}}$), θ_3 (between vectors $\mathbf{v_{i-1}}$ and $\mathbf{v_{i+1}}$), short-range energy contributions for the best found conformation of 24 amino acids (total energy = -109, short-range energy = -68, long-range energy = -41). 165
B.14 Long-range interactions for the best found conformation of 24 amino acids (total energy = -109, short-range energy = -68, long-range energy = -41). 166
B.15 Vectors for the best found conformation of 12 amino acids (total energy = -39, short-range energy = -28, long-range energy = -11). 167
B.16 Triplets of vectors, angles: θ_1 (between vectors $\mathbf{v_{i-1}}$ and $\mathbf{v_i}$), θ_2 (between vectors $\mathbf{v_i}$ and $\mathbf{v_{i+1}}$), θ_3 (between vectors $\mathbf{v_{i-1}}$ and $\mathbf{v_{i+1}}$), short-range energy contributions for the best found conformation of 12 amino acids (total energy = -39, short-range energy = -28, long-range energy = -11). 167
B.17 Long-range interactions for the best found conformation of 12 amino acids (total energy = -39, short-range energy = -28, long-range energy = -11). 167

List of Figures

1.1 (a) Structure of an amino acid (electrical charges are not indicated); (b) peptide bond between neighbouring amino acids. 2

1.2 (a) Dihedral angles (ϕ,ψ); (b) Ramachandran plot of allowed values for the dihedral angles. All regions shown in white are disallowed. 3

1.3 (a) Major elements of secondary structure: α-helix (left), β-sheet (right); (b) tertiary structure of Myoglobin. 4

2.1 (a) An example of a protein conformation on the 2D square lattice; (b) an example of a protein conformation on the 3D cubic lattice. Two types of monomers are considered: hydrophobic amino acids are colored in red, polar are green, yellow dotted lines represent energy interactions. 17

2.2 (a) Depiction of the 12 base vectors for the FCC lattice model; (b) an example of several FCC lattice cells. 19

2.3 The effective contact order (ECO) between i and j is equal to 5, compared to the contact order (CO) of 7 between the same residues. 25

3.1 An example of a protein's funneled energy landscape. 28

4.1 Generic outline of Ant Colony Optimization (for static combinatorial problems). 49

4.2 The local structural motifs that form the solution components underlying the construction and local search phases of our ACO algorithm in 3D. 49

4.3 The underlying protein sequence (Sequence S1-1 from Table 4.1) is HPHPPHHPHPPHPHHPPHPH; black circles represent hydrophobic amino acids, while white circles symbolize polar amino acids. The dotted lines represent the H-H contacts underlying the energy calculation. The energy of this conformation is -9, which is optimal for the given sequence. 50

4.4 The iterative first improvement local search procedure that is performed by selected ants after the construction phase. 53
4.5 The native conformation of Sequence S1-8 from Table 4.1 (64 amino acids; energy -42), found by ACO in an average CPU time of 1.5 hours and by PERM in $t_1 = t_2 = t_{exp} = 78$ hours. 60
4.7 Mean CPU time (natural log transformed) required by ACO *vs* PERM for reaching the best solution quality, as observed over 10 runs with a cut-off time of 1 CPU hour for sequences of length 30 and 50 in 3D. The left and right plots show the results for the biological and random test-sets, respectively. Performance results for instances of size 30 are indicated by circles, while stars mark results for instances of size 50. The dashed lines indicate the band within which performance differences are not statistically significant (significance was determined using the Mann-Whitney U test and setting the significance level at 0.05 [59]). Mean run-times were obtained from 10 runs per instance and algorithm. We only show data points for the runs where the best known solution quality was reached at least in some runs out of 10 by both algorithms. When unsuccessful runs were present, the expected time was calculated as in [101]. For biological sequences of length 50, there are four instances and for random instances of length 50, there are two instances for which this was the case. Detailed results for both successful and unsuccessful runs are given in Appendix A. 63
4.8 Left side: Lowest energy conformation of a biological sequence (B50-7, 45 amino acids, energy -17) that is harder for PERM ($t_1 = 271$, $t_2 = 299$, $t_{exp} = 284$ CPU seconds) than for ACO ($t_{exp} = 130$ CPU seconds; cut-off time 1 CPU hour). Right side: Lowest energy conformation of a biological sequence (B50-5, 53 amino acids, energy -22) that is much harder for ACO than for PERM; within a cut-off time of 1 CPU hour, both ACO and PERM reached this energy in 10 out of 10 runs in $t_{avg} = 820$ and $t_1 = 5$, $t_2 = 118$, $t_{exp} = 9$ CPU seconds on average, respectively. 64

4.9 Left side: Unique minimal energy conformation of a designed sequence, D-1 (length 50, energy -19); ACO reaches this conformation much faster than PERM when folding from the left end (mean run-time over 100 successful runs for ACO: 236 CPU seconds, compared to $t_1 = 3\,795$, $t_2 = 1$, $t_{exp} = 2$ CPU seconds for PERM). Right side: Unique native conformation of another designed sequence, D-2 (length 60, energy -17). ACO finds this conformation much faster than PERM folding from either end (mean run-time over 100 successful runs for ACO: 951 CPU seconds, compared to $t_1 = 9\,257$, $t_2 = 19\,356$, $t_{exp} = 12\,524$ CPU seconds for PERM). 65

4.10 Left side: Lowest energy conformation of random sequence R50-9 (50 amino acids, energy -30), which is harder for PERM when folding from the left end than for ACO. With a cut-off time of 1 CPU hour, ACO reached this energy in 10 out of 10 runs with $t_{exp} = 1000$ CPU seconds, while PERM failed to find a conformation with this energy in 7 out of 10 runs when folding from the left end ($t_1 = 9\,892$, $t_2 = 2$, $t_{exp} = 3$ CPU seconds). The conformation displayed here has relative directions: DSRUURLRRLDDLURRULRSDURURLRRDDUSRRSULRRDUULSLLUR. Right side: Lowest energy conformation of random sequence R50-7 (50 amino acids, energy -38), which is much harder for ACO than for PERM. With a cut-off time of 1 CPU hour, PERM reached this energy in two out of 10 runs when folding from the left and in 10 of 10 runs when folding from the right end in $t_1 = 15\,322$, $t_2 = 46$, $t_{exp} = 92$ CPU seconds, while the lowest energy reached by ACO over ten runs was -37. The conformation displayed here has relative directions: SUURUDULUUDSSUUSRUDDLDSSUURRUULDLDRDDURLLDLRUSUR. 66

4.11 Distributions of H-H contact order for 500 conformations of Sequence S1-7 from Table 4.1 (60 amino acids) in 2D (left side) and Sequence S1-5 from Table 4.1 (48 amino acids) in 3D (right side) found by ACO and PERM. 67

4.12 Mean hydrophobic solvent accessible area as a function of prefix length for a biological sequence (B50-4, 50 amino acids) in 2D (left side) and Sequence S2-6 from Table 4.1 (48 amino acids) in 3D. Crosses and circles represent mean values for an ensemble of 100 native structures found by ACO and PERM, respectively. . . . 67

4.13 Mean number of H-H contacts as a function of prefix length for a biological sequence (B50-4, 50 amino acids) in 2D (left side) and Sequence S2-6 from Table 4.1 (48 amino acids) in 3D. Crosses and circles represent mean values for an ensemble of 100 native structures found by ACO and PERM, respectively. 68
4.14 Mean H-H contact order as a function as a function of prefix length for a biological sequence (B50-4, 50 amino acids) in 2D (left side) and Sequence S2-6 from Table 4.1 (48 amino acids) in 3D. Crosses and circles represent mean values for an ensemble of 100 native structures found by ACO and PERM, respectively. 68
4.15 Effect of the relative weights of pheromone information, α, and heuristic information, β, on the average CPU time required for obtaining minimal energy conformations of Sequence S1-8 in 2D (length 64, left side) and Sequence S2-5 in 3D (length 48, right side). 73
4.16 Effect of the pheromone persistence parameter, ρ, on the average CPU time required for obtaining minimal energy conformations of Sequence S1-8 in 2D (length 64, left side) and Sequence S2-5 in 3D (length 48, right side). 73
4.17 Mean CPU time required for finding minimum energy conformations of Sequence S1-7 in 2D (length 60, left side) and Sequence S2-5 in 3D (length 48, right side), as a function of ant colony size and the maximum number of non-improving local search steps. . . 74
4.18 Mean CPU time required for finding minimum energy conformations of Sequence S1-8 in 2D (length 64, left side) and Sequence S2-5 in 3D (length 48, right side), as a function of the probability of retaining previous directions ($\hat{p}$) during long-range mutation moves. 75

5.1 An illustration of how candidate solutions at a given state of the bin framework relate to the search space of a given problem instance. E_0 is the best solution quality found so far and serves as an estimate of the ground state energy, ΔE is the energy range of interest, and conformations within this range are binned. Each bin i has energy threshold E_i^+, diversity threshold HD_i, and energy window ΔE_i. 81

5.2 High-level outline of the main body of the Bin Framework Monte Carlo algorithm. E_0 is the current best solution quality; *noImprRetrieve* is a parameter of the algorithm that specifies number of non-improving steps over the best energy that are tolerated before a new conformation is retrieved from the bin system; HD_{MAX} and HD_{MIN} are parameters defining the Hamming distance diversity criteria for high-energy and low-energy conformations correspondingly; β_{MC} and β_{bin} are parameters that refer to the inverse temperatures used for the Monte Carlo run and for the bin framework correspondingly; ΔE_i is a parameter that represents the window of energies by each bin i; ΔE is a parameter defining the range of energies of interest. Conformations whose energy falls within this range are attempted to be stored in the bin framework. The functions *PlaceIntoBin* and *CheckPlaceIntoBin* are defined in Figures 5.3 and 5.4 correspondingly. 83

5.3 Outline for the function used to place the conformation with the lowest energy encountered so far into the bin system, where c is the conformation with energy $E(c)$ to be placed into a bin, E_0 is the estimated ground state energy (the lowest energy seen in the simulation so far), HD_{MAX} and HD_{MIN} are parameters defining the Hamming distance diversity criteria for high-energy and low-energy conformations correspondingly, ΔE_i is a parameter that represents the window of energies considered by each bin i, ΔE is a parameter defining the range of energies of interest – conformations whose energy falls within this range are attempted to be stored in the bin framework. 84

5.4 Outline for the function used to place the conformation c with a low energy $E(c)$ encountered into the bin system, where E_0 is the estimated ground state energy (the lowest energy seen in the simulation so far), HD_{MAX} and HD_{MIN} are parameters defining the Hamming distance diversity criteria for high-energy and low-energy conformations correspondingly, ΔE_i is a parameter that represents the window of energies considered by each bin i, ΔE is a parameter defining the range of energies of interest – conformations whose energy falls within this range are attempted to be stored in the bin framework. 85

5.5 Placement of the low-energy conformation c with energy $E(c)$ into an appropriate bin i with the energy threshold E_i^+ and the Hamming distance criterion HD_i. In order for the conformation c to be stored in the bin framework the following conditions need to be satisfied: (1) $E(c) \leq E_i^+$, and (2) the Hamming distance (HD) between conformation c and conformations c' with the same energy already stored in the bin (if there are any) is larger or equal to HD_i. 86

5.6 Outline for the function used to retrieve a conformation from the bin system. E_0 is the current best solution quality; β_{bin} is the inverse temperature used for the bin framework during the retrieval of conformations. 88

5.7 (a) The lowest energy conformation of the FCC 64 amino acids homopolymer found by our Bin Framework Monte Carlo method (energy -391, short-range energy is -212, long-range energy is -179). The detailed description of this conformation is also found in Appendix B; (b) same conformation, view from above; (c) a low-energy conformation of the FCC 64 amino acids homopolymer found by BINMC (energy -387, short-range energy is -212, long-range energy is -175); this was found in the same run that lead to the conformation with the best energy of -391; (d) another low-energy conformation of the FCC 64 amino acids homopolymer found in the same run of BINMC (energy -388, short-range energy is -212, long-range energy is -176). 92

5.8 (a) A low-energy conformation of the FCC 64 amino acids homopolymer found by BINMC (energy -387, short-range energy is -208, long-range energy is -179); (b) another low-energy conformation of the FCC 64 amino acids homopolymer found in the same run of BINMC (energy -389, short-range energy is -212, long-range energy is -177); (c) a low-energy conformation of the FCC 64 amino acids homopolymer found by BINMC (energy -387, short-range energy is -208, long-range energy is -179); (d) another low-energy conformation of the FCC 64 amino acids homopolymer found in the same run of BINMC (energy -389, short-range energy is -220, long-range energy is -169). 93

5.9 (a) The low-energy conformation of the FCC 64 amino acids homopolymer found by our bin framework (energy -387, short-range energy is -220, long-range energy is -167); (b) another low-energy conformation of the FCC 64 amino acids homopolymer found by our bin framework (energy -387, short-range energy is -220, long-range energy is -167); (c) the same conformation as in part (a), view from above; (d) the same conformation as in part (b), view from above. 94

5.10 The lowest energy conformation of the FCC homopolymers of 12, 24, and 32 (same conformation from different point of view) amino acids (from left to right) found by all of the algorithms (corresponding energies are: -39, short-range energy is -28, long-range energy is -11 for the 12 amino acid polymer; -109, short-range energy is -68, long-range energy is -41 for the 24 amino acid polymer; and -161, short-range energy is -112, long-range energy is -49 for the 32 amino acid polymer). The detailed description of these conformations is also found in Appendix B. 95

5.11 Run-time distributions of CPU times on our 2.4 GHz reference machine to obtain a sub-optimal solution quality of -158 for the homopolymer of length 32 (part (a)) and to obtain sub-optimal solution quality of -370 for the homopolymer of length 64 (part (b)) using Monte Carlo (MC), Replica Exchange Monte Carlo (REMC), Parallel-hat Tempering (PHAT), and the Bin Framework Monte Carlo (BINMC). We fit the RTD of BINMC for the homopolymer of length 64 with exponential distribution, to show by example that RTDs for all algorithms are exponential. For all algorithms, 100 independent runs were performed. In all of them, the target energy was reached. 99

5.12 Distributions of energies visited by different replicas in a representative run of the Replica Exchange Monte Carlo (REMC) (part (a)) and in a representative run of the Parallel-hat Tempering Monte Carlo (PHAT) (part (b)) for the homopolymer of length 64. The time cut-off used was 28 min on our reference machine. 100

5.13 Distributions of energies visited by the Monte Carlo (MC) and our Bin Framework Monte Carlo (BINMC) for the homopolymer of length 64. The time cut-off used was 28 min on our reference machine. 100

5.14 Mean CPU time required for finding minimum energy conformations (-161) of the 32 amino acid homopolymer over 20 independent runs on our reference machine as a function of the number of non-improving steps (*noImprRetrieve*) before retrieving a conformation from bins. Error bars indicate standard deviation observed. Dashed line indicates that energy of -161 was not found after 2 weeks' CPU time on our reference machine. 102
5.15 Mean CPU time required for finding minimum energy conformations (-161) of the 32 amino acid homopolymer over 20 independent runs on our reference machine as a function of combination of parameters: the energy range of interest (ΔE) and the bin's temperature (T_{bin}), $p = e^{-\Delta E/T_{bin}}$. Error bars indicate standard deviation observed. 102
5.16 Mean CPU time required for finding minimum energy conformations (-161) of the 32 amino acid homopolymer over 20 independent runs on our reference machine as a function of the Hamming distance criterion (HD_{MAX} is varied, $HD_{MIN} = 0.1$ is kept constant, part (a) and HD_{MIN} is varied, $HD_{MAX} = 0.6$ is kept constant, part (b)). Error bars indicate standard deviation observed. . . 104
5.17 Mean CPU time required for finding minimum energy conformations (-161) of the 32 amino acid homopolymer over 20 independent runs on our reference machine as a function of the energy window width ΔE_i. Error bars indicate standard deviation observed. 105
5.18 Mean CPU time required for finding minimum energy conformations (-161) of the 32 amino acid homopolymer over 20 independent runs on our reference machine as a function of the bin's capacity. Error bars indicate standard deviation observed. 106

6.1 Polymer graph generated in phase one of our algorithm for the chymotrypsin inhibitor 2 (CI2) protein containing 65 nodes (n) and 82 edges (m). 110

6.2 Illustration of the sampling phase of the algorithm. Edges that are added to the polymer graph are highlighted in bold, while edges that still have to be added are represented in non-bold. Initially, we start with a fully extended polymer, and edges that are to be added are weighted by their chain separation. At each step one edge is probabilistically added based on its ECO weight, and the ECO weights of the edges that are still to be added have to be revised at each step. The process of addition of edges stops when all of the edges are added to the polymer graph and their respective ECOs shrink to 1. 112

6.3 Outline for the probabilistic constructive local search used in the sampling of the folding pathways stage of our algorithm, where k is the number of non-covalent edges already added to the polymer graph and m is the total number of non-covalent edges in the polymer graph. 113

6.4 Outline of the analysis phase of our algorithm that identifies folding nuclei contacts in the low effective contact order pathway ensemble. Parameter Δl is the length threshold used to identify contacts whose formation proceeds rapidly. 115

6.5 Band representation of the experimental and predicted folding nuclei for our test set of 27 proteins, Part I. The upper band represents H/X experimental data (darker-shaded residues represent those that gain protection first during folding, lighter-shaded residues represent residues that have slow exchange in the native state H/X [76]). The middle band (if available) represents experimental Φ-values (the darker the shade, the higher the Φ-value ($0 \leq \phi \leq 1$) [93]. The lower band represents our folding nuclei predictions, shaded according to percentage of involvement in low-ECO events (the exact scale in percent is given on the right side; proteins are grouped according to the maximum percentage of edge usage). 117

6.6 Band representation of the experimental and predicted folding nuclei for our test set of 27 proteins, Part II. The upper band represents H/X experimental data (darker-shaded residues represent those that gain protection first during folding, lighter-shaded residues represent residues that have slow exchange in the native state H/X [76]). The middle band (if available) represents experimental Φ-values (the darker the shade, the higher the Φ-value ($0 \leq \phi \leq 1$) [93]. The lower band represents our folding nuclei predictions, shaded according to percentage of involvement in low-ECO events (the exact scale in percent is given on the right side; proteins are grouped according to the maximum percentage of edge usage). 118

6.7 Band representation of the experimental and predicted folding nuclei for our test set of 27 proteins, Part III. The upper band represents H/X experimental data (darker-shaded residues represent those that gain protection first during folding, lighter-shaded residues represent residues that have slow exchange in the native state H/X [76]). The middle band (if available) represents experimental Φ-values (the darker the shade, the higher the Φ-value ($0 \leq \phi \leq 1$) [93]. The lower band represents our folding nuclei predictions, shaded according to percentage of involvement in low-ECO events (the exact scale in percent is given on the right side; proteins are grouped according to the maximum percentage of edge usage). 119

6.8 Predicted order of contact formation for the chymotrypsin inhibitor 2 protein (pdb id 2ci2). Contact matrix and three-dimensional structure representations are gray-scale-coded according to percent contact usage (the scale is given on the right). Indexes n_1 and n_2 represent indexes of the residues along the backbone. Darker-shaded edges (contacts) are predicted to form first. 120

6.9 Predicted order of contact formation for the B1 immunoglobulin-binding domain of protein G (left) and protein L (right), (pdb ids 1pga and 2ptl). Darker shaded edges (contacts) are predicted to form first. 121

6.10 Predicted order of contact formation for the the barnase protein (pdb id 1a2p). Darker shaded edges (contacts) are predicted to form first. 122

6.11 Predicted order of contact formation for the chicken src SH3 domain (pdb id 1srm). Darker shaded edges (contacts) are predicted to form first. 123

6.12 Predicted order of contact formation for the CheY protein (pdb id 3chy). Darker shaded edges (contacts) are predicted to form first. . 123

6.13 Predicted order of contact formation for the ubiquitin protein (pdb id 1ubi). Darker shaded edges (contacts) are predicted to form first. 124
6.14 Predicted order of contact formation for the T4 lysozyme protein (pdb id 3lzm). Darker shaded edges (contacts) are predicted to form first. 124
6.15 Predicted order of contact formation for the bovine pancreatic trypsin inhibitor protein (pdb id 1bpi). Darker shaded edges (contacts) are predicted to form first. 125
6.16 Predicted order of contact formation for the B domain of protein A (pdb id 1bdd). Darker shaded edges (contacts) are predicted to form first. 125

Abbreviations

2D	Two-Dimensional
3D	Three-Dimensional
ACO	Ant Colony Optimization
ATP	Adenosine Triphosphate
BINMC	Bin Framework Monte Carlo Algorithm
CASP	Critical Assessment of Techniques for Protein Structure Prediction
CG	Core-directed Chain Growth Method
CHCC	Constraint-based Hydrophobic Core Construction Method
CI	Contact Interactions Method
CO	Contact Order
CPU	Central Processing Unit
DNA	Deoxyribonucleic Acid
DSSP	Database of Secondary Structure Assignments for Proteins
EA	Evolutionary Algorithm
ECO	Effective Contact Order
ELP	Energy Landscape Paving
EMC	Evolutionary Monte Carlo Algorithm
FCC	Face Centered Cubic
FIRST	Freely Rotating Rodes Method
GA	Genetic Algorithm
GCP	Graph Coloring Problem
GNM	Gaussian Network Model
H-bonding	Hydrogen Bonding
H-contact	Hydrophobic Contact
H-H	Hydrophobic-Hydrophobic
H/X	Hydrogen-Deuterium Exchange
HD	Hamming Distance
HP	Hydrophobic Polar
HZ	Hydrophobic Zipper
JSP	Job-Shop Scheduling Problem
MBS	Model-Based Search
MC	Monte Carlo Method

MCMC	Markov Chain Monte Carlo
MCSA	Monte Carlo Simulated Annealing
MD	Molecular Dynamics
MSOE	Multi-Self-Overlap Ensemble
MUCA	Multicanonical Algorithm
NMR	Nuclear Magnetic Resonance
NP-hard	Non-deterministic Polynomial-Time Hard
REMC	Replica Exchange Monte Carlo
RMSD	Root Mean Square Deviation
RNA	Ribonucleic Acid
PDB	Protein Data Bank
PERM	Pruned-Enriched Rosenbluth Method
PHAT	Parallel-Hat Tempering
QAP	Quadratic Assignment Problem
RAM	Random-Access Memory
RTD	Run-Time Distribution
SA	Simulated Annealing
SAT	Propositional Satisfiability Problem
SLS	Stochastic Local Search
TS	Tabu Search
TSP	Traveling Salesman Problem
VRP	Vehicle Routing Problem

Acknowledgements

I would like to thank my supervisor Holger Hoos for his guidance, thoroughness, and keen insight. I would also like to thank Steven Plotkin for providing pointers throughout my research and for bringing up and discussing relevant questions. My respectful and profound gratitude to my supervisory committee: Holger Hoos, Steven Plotkin, Alan Mackworth, Nando de Freitas, Anne Condon, for discussing research ideas, for your keen observations, for helping with different aspects of the research, giving valuable suggestions, providing constructive criticism, and proofreading the thesis. Thanks to members of the BETA lab who were always there to discuss any research-related and unrelated issues. My gratitude and profound appreciation to the National Science and Engineering Council of Canada (NSERC) for funding my doctoral studies. Special thanks to Sam Thangiah, my undergraduate research supervisor who taught me how gratifying research curiosity can be when one discovers answers to intriguing scientific questions.

My utmost gratitude and admiration to my family (my mom Klara and my brother Valik) for all the remarkable and little things they do in life, who inspire me, give me greatest warmth and happiness, who prove each day that *"to live without roots takes a stout heart" [E. M. Remarque]*. My sincere gratitude to my loyal and incomparable friends for guarding my solitude during the trying time of doctoral research, sharing your time, your thoughts, your aspirations, creating the treasure of common memories and emotions. Thank you for sharing the apple-green sky, snow covered mountains, deserts and crystal lakes springing to life and taking on a meaning beyond their natural beauty in your company. Our time together and our travel adventures helped me to find new meaning in old appearances during my time here. *"There are no reasons for our closeness... one is in need of a friend in order for one to be oneself" [R. Rolland]*.

I am incredibly grateful that I completed this scientific journey accompanied and supported by all of you, and now, *"...embarked upon my illicit exploration of the world" [A. de S.-Exupery]*, here is my thesis.

With my love to my Mom: I know myself through you.

In memory of my Grandfather, Volf Buhman.

IF

If you can keep your head when all about you
Are losing theirs and blaming it on you,
If you can trust yourself when all men doubt you,
But make allowance for their doubting too;
If you can wait and not be tired by waiting,
Or being lied about, don't deal in lies,
Or being hated, don't give way to hating,
And yet don't look too good, nor talk too wise:

If you can dream – and not make dreams your master,
If you can think – and not make thoughts your aim;
If you can meet with Triumph and Disaster
And treat those two impostors just the same;
If you can bear to hear the truth you've spoken
Twisted by knaves to make a trap for fools,
Or watch the things you gave your life to, broken,
And stoop and build 'em up with worn-out tools:

If you can make one heap of all your winnings
And risk it all on one turn of pitch-and-toss,
And lose, and start again at your beginnings
And never breath a word about your loss;
If you can force your heart and nerve and sinew
To serve your turn long after they are gone,
And so hold on when there is nothing in you
Except the Will which says to them: "Hold on!"

If you can talk with crowds and keep your virtue,
Or walk with kings – nor lose the common touch,
If neither foes nor loving friends can hurt you,
If all men count with you, but none too much;
If you can fill the unforgiving minute
With sixty seconds' worth of distance run,
Yours is the Earth and everything that's in it,
And – which is more – you'll be a Man, my son!

Rudyard Kipling

Если

Если сумеешь сохранить свой разум,
Когда все будут ныть, во всём винить тебя,
И веру сохранишь, хоть потускнеешь разом
В глазах людей (пусть, ты им не судья);

Если сумеешь ждать и цели добиваться
И, повстречав обман, ты не погрязнешь в лжи,
А ненависть застав, не станешь огрызаться,
И всё без лишних слов, без слов пустых;

Если уметь мечтать, мечте не покаряясь,
И мыслить живо там, где мысль мертвят,
Встречать успех, несчастье, улыбаясь,
И видеть, в чём обман они сулят;

Если не будешь сломлен ты, когда услышишь
В устах лжеца растлённой правду, что берёг,
Когда всё то, чем ты живёшь и дышишь,
Растопчат, загрязнят подошвы дураков;

Если сумеешь ставить всё на карту,
Всё, что достиг, всё то, чем жизнь полна,
И потерять, и, хоть пошло всё к чёрту,
Начать сначала и ни слова о потере не сказать;

Если напрячь сумеешь каждый мускул,
Свой каждый нерв и сердца каждый вздрог,
Чтоб глыбы своротить, пройти труднейший путь свой:
Усталый, средь врагов, ты всё равно пройдёшь;

Если сумеешь быть и жить с толпою,
Не покорившись ей, не став жесток и глуп;
Если, увидев власть перед собою
Не соблазнишься, не протянешь рук;

Если сумеешь оценить в минуте
Её простор шестидесяти секунд,–
Ты овладеешь жизни главной сутью
И, более того, – ты человек, мой друг.

перевод с английского Вольфа Бухмана
translated from English by Volf Buhman

Chapter 1

Introduction

DNA makes RNA, RNA makes proteins, and proteins make us.

Francis Crick

In the entire realm of science, there is probably no class of molecule currently known that can compete with proteins when it comes to diversity of functions performed. Proteins serve as structural units, participate in the catalysis or inhibition of various chemical reactions, perform a transport role, carry out energy transduction, control metabolic pathways, stabilize the architecture within the cell, perform a protective function and many others. If there is a job to be done in the biological organism, there almost always exists a protein to perform the required task.

The protein folding problem has played a prominent role in the fields of biomolecular physics and algorithm design for over 50 years. The importance of this problem increases continually with the exponential growth in the number of known protein sequences and only linear increase in the number of known protein structures [12].

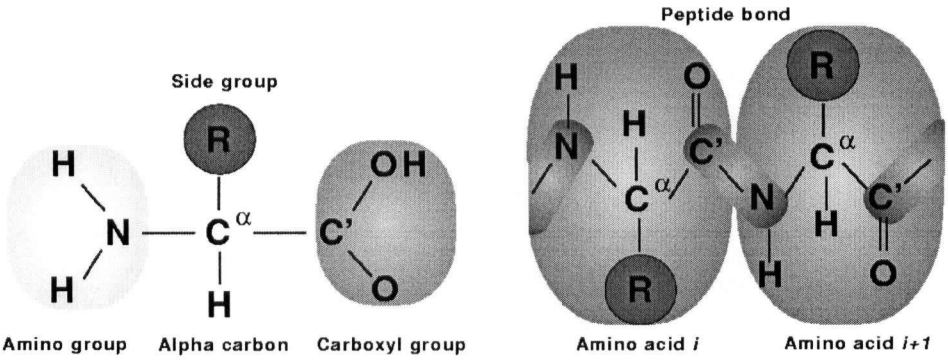
1.1 Structure and Function of Proteins

Proteins are the final products of most genes. They are chain-like polymers of small subunits (*amino acids*). Twenty different amino acids are distinguished. Modified forms of these 20 amino acids do exist, but they are less common. The chemical structure common to all amino acids, with the exception of Proline, is given in Figure 1.1. Each amino acid has an amino group (NH_3^+), a carboxyl group (COO^-), a hydrogen atom (H), and a side chain (*R-group*, attached to a central alpha-carbon C^α) that is different from one amino acid to another. In Proline, the side-chain bridges to the nitrogen atom of the amino group of the backbone (backbone atoms are N, C^α, C, O, and hydrogen atoms attached to nitrogen and alpha-carbon). The standard amino acids can be loosely grouped into classes based on their chemical properties. Commonly accepted classes are hydrophobic, polar or hydrophilic, and

charged [17]. Within these broad classes, further classifications are possible, for example: aromatic or aliphatic, large or small [17].

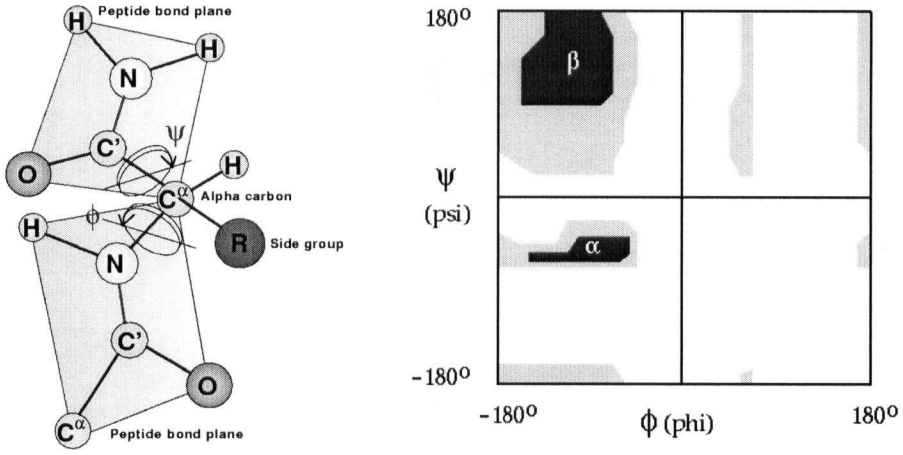
The arrangement of amino acids with their distinct side chains gives each protein its unique character. The amino acids join via a *peptide bond* between the amino group and carboxyl group, as shown in Figure 1.1, part (b). Therefore, the protein (*polypeptide*) has polarity: a free amino group (*amino terminus or N - terminus*) on its left and a free carboxyl group (*carboxyl terminus* or *C - terminus*) on its right end.

The peptide bond is planar and quite rigid, see Figure 1.2. Therefore, the polypeptide chain has rotational freedom only about the bonds formed by the alpha-carbon. The angles that denote this rotational freedom are noted as ϕ - rotation around the C^{α}-N axis, and ψ - rotation in reference to the C^{α}-C axis; thus, a protein of length n has $2n$ torsion angle degrees of freedom for positioning the backbone (main chain). The dihedral angles ψ and ϕ are graphically depicted in Figure 1.2, part (a). However, the rotational freedom about these angles is limited by steric hindrance between the side chains of the residues and the peptide backbone. The possible conformations of a given polypeptide chain are quite limited and are represented using a Ramachandran plot (a plot of permissible ϕ *vs.* ψ combinations; see Figure 1.2, part (b)).



(a) (b)

Figure 1.1: (a) Structure of an amino acid (electrical charges are not indicated); (b) peptide bond between neighbouring amino acids.



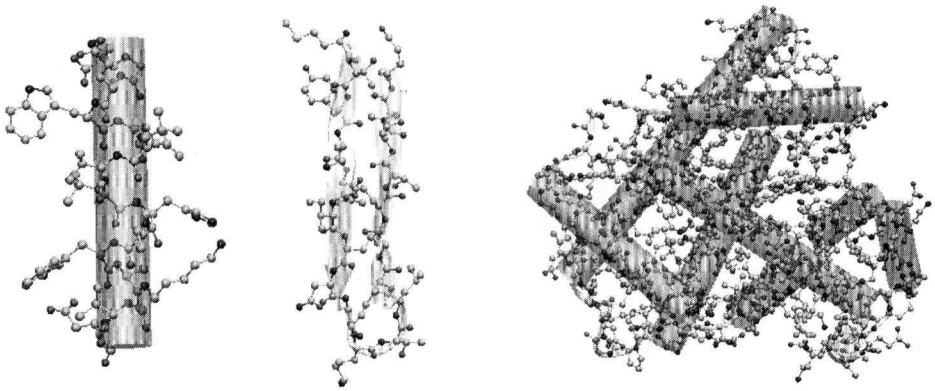
(a) (b)

Figure 1.2: (a) Dihedral angles (ϕ,ψ); (b) Ramachandran plot of allowed values for the dihedral angles. All regions shown in white are dis-allowed.

1.1.1 Three Levels of Protein Structure

The linear sequence of amino acids constitutes the *primary structure* of the protein. Amino acids in the protein usually influence parts of the backbone (amide group and carboxyl group) to interact with each other by means of hydrogen bonds to form a *secondary structure*. The three most common secondary structure elements are: α - *helix*, which results from hydrogen bonding among near-neighbour amino acids (stabilized by hydrogen bonds between the carbonyl oxygen of the amino acid residue at position n with the amide group, NH, of residue $n + 4$), as shown in Figure 1.3, part (a); β - *sheet*, which involves extended protein chains packed side by side that interact by hydrogen bonding (hydrogen bonds occur between carbonyl oxygen and the amide group of adjacent strands; see Figure 1.3, part (a)); and *turns*, which usually connect α-helices and β-sheets (turns involve $180°$ change in the direction of the chain, and are stabilized by a hydrogen bond between the carbonyl oxygen of the residue at position n and the amide group, NH, of residue $n + 3$). Helices are the most abundant secondary structure elements in *globular proteins* (for example, enzymes), followed by sheets, and then turns [92]. Those regions that cannot be classified as one of the standard three classes of secondary structure are classified as *random coil*. The complete three-dimensional

shape of a polypeptide is its *tertiary structure*. Figure 1.3, part (b) shows the tertiary structure of the protein myoglobin. Most *globular* proteins (soluble proteins) take this roughly spherical shape. Many proteins are composed of multiple subunits [143]. They additionally possess a fourth level of structure – *quaternary structure*, which involves association of two or more polypeptide chains to form larger protein molecules that function as dimers, trimers, tetramers, *etc.* (for example, hemoglobin functions as a tetramer). Complexes higher than octamers are rarely observed [143].



(a) (b)

Figure 1.3: (a) Major elements of secondary structure: α-helix (left), β-sheet (right); (b) tertiary structure of Myoglobin.

1.1.2 Protein Functions

Proteins are very versatile molecules that assume a number of functions [74], including the following:

1. a structural role – give cells their integrity and shape (for example, viral coat proteins, molecules of the cytoskeleton such as actin filaments, epidermal keratin)

2. an enzymatic role in a number of metabolic pathways – binding and carrying substances, regulating different catalytic reactions (for example, ribonuclease – an enzyme that degrades RNA)

3. a hormonal role – carrying signals from one cell to another (for example, insulin)
4. gene-regulating activity (for example, DNA polymerase)
5. a transport and storage role (for example, hemoglobin, myoglobin, ferritin)
6. energy transduction – performing electron transfer (for example, ATP synthase)
7. and an immune system role (including proteins involved in cell-cell recognition and signaling, for example, immunoglobulins that function as antibodies)

1.2 The Protein Folding Problem and its Importance

The protein folding problem consists of predicting the functional (*tertiary* or *native*) structure of the protein given its linear sequence of amino acids (*primary* structure).

Even though extensive progress has been made in understanding the process of protein folding, as described in Chapter 3, there still remain a number of unanswered questions about this process, which is guided just by the propensities coded in the sequence itself. For instance, we would like to find out what exactly is coded within the sequence, and how it determines the structure; we would also like to know if proteins evolve to have the best sequence, and consequently the best structure for their function.

1.2.1 Protein Folding Paradoxes

The mystery of protein folding provides three challenging paradoxes [147]:

1. The *Levinthal paradox*: How can a protein find its low-energy state in time less than geological? On average, the number of independent conformations per amino acid residue is about 10. This means that for a protein of length n, there are 10^n possible conformations. Therefore, folding cannot proceed via random sampling of the space of conformations. Consequently there must exist folding pathways that allow folding to proceed efficiently. Little is still known about the folding pathways of proteins, and properties of the intermediate states that a protein undergoes during folding.
2. The *Hoyle paradox*, which asks how foldable proteins evolved within the lifetime of the universe. The number of possible proteins of length 100,

for example, is 20^{100}, which exceeds by far the number of protons in the universe times the turnover time of a typical protein. Thus, it is difficult to argue that any particular protein is perfect for its task.

3. The *marginal stability* or *Honig paradox*, which deals with the issue of why proteins seem to have such a delicate compensation of entropy and energy during intermediate stages of the folding process. The native state of globular proteins is typically only $5 - 15$ kcal/mol more stable than the denatured state [74]. This is the equivalent of about one or two water-water hydrogen bonds. Precisely why proteins have marginal stability is unclear – it could be to facilitate turnover of proteins, or maybe proteins are as stable as they need to be and there is no selective advantage to optimizing the stabilizing interactions further.

1.2.2 The Thermodynamic Hypothesis

In order to predict the tertiary structure of a protein, we need to know something about the nature of its *native state* (functional three-dimensional state). An important characteristic of the native state is described by the *Thermodynamic Hypothesis*. It states that the native state represents the ground state of lowest Gibbs free energy, and has been proposed by Anfinsen on the basis of denaturalization - renaturalization experiments of the ribonuclease enzyme [4]: "The native conformation is determined by the totality of inter-atomic interactions and hence by the amino acid sequence, in a given environment." Usually, the native state of a protein is at the minimum free energy. Exceptions to this rule are very rare (for example, in the case of prions, whose folding happens under kinetic control) [27].

1.2.3 The Central Dogma of Protein Folding

Resulting from Anfinsen's experiment, the Central Dogma of protein folding (or the Central Dogma of genomics) states that "sequence determines structure determines function". This dogma assumes that the protein sequence and sequence alone in the given environment determines a protein's tertiary structure, and that the function of the protein is primarily determined by its structure [27]. Molecules that can assist in the folding of some proteins (*chaperons*) do not violate this dogma, because chaperons do not induce a fold in proteins that is different from one adopted when the proteins are allowed to fold on their own in diluted solution; chaperons just expedite the folding or prevent unwanted aggregation. The dogma asserts that it should be possible, ultimately, to deduce the function of a protein from its structure.

1.2.4 Forces in Protein Folding

To approach the *ab initio* protein folding problem, one needs to quantify physical forces that determine the structure of the protein. These include: hydrophobic, hydrogen bonding, electrostatic, and van der Waals forces. These forces, together with covalent interactions, determine conformation along the folding pathway, the shape of folded and misfolded proteins, and also interactions between proteins, proteins and other molecules, as well as proteins and the solvent.

Hydrophobic Force

It is widely believed that the folding of *globular* proteins (most enzymes) is primarily driven by the hydrophobic force [27]. Non-polar atoms, such as those in hydrocarbons, are trying to reduce their contact with aqueous environments. The folded state of globular proteins is reflected by a balance between the opposing energetics of H-bonding between hydrophilic R-groups and the aqueous environment and repulsion from the aqueous environment by hydrophobic R-groups. The hydrophobicity of certain amino acid R-groups (for example, Leucine, Isoleucine, and Valine) tends to drive them away from the exterior of proteins into the interior. In contrast, hydrophilic side-chains (for example, Arginine, Aspartic acid, and Asparagine) make hydrogen bonds with the aqueous environment, and are preferentially found on the surface of folded proteins. This driving force, characterized by the absence of hydrogen bonding between water and non-polar groups (rather than the presence of favourable interactions between non-polar groups themselves), restricts the available conformations into which a protein may fold. Most proteins that occur in the aqueous, intracellular environment or in plasma are of a globular nature because their structure is determined primarily by the hydrophobic force: they are approximately spherical in shape, or consist of several different domains (average domain length is $200 - 300$ amino acids) [92]. A domain is a "folding unit" of a protein, the part of the protein sequence that folds largely independently of the rest of the sequence.

Hydrogen Bonding Force

Polypeptides contain numerous proton donors and acceptors both in their backbone and in the R-groups of the amino acids. The aqueous environment in which proteins are found also contains ample H-bond donors and acceptors. H-bonding, therefore, occurs not only within and between polypeptide chains but also with the surrounding aqueous medium [27].

Electrostatic Force

Electrostatic forces are mainly of three types: charge-charge, charge-dipole, and dipole-dipole [27]. Typical charge-charge interactions that favour protein folding are those between oppositely charged R-groups such as Lysine or Arginine and Aspartic acid or Glutamic acid. A substantial component of the energy involved in protein folding is a result of charge-dipole interactions, the interaction of ionized R-groups of amino acids with the dipoles of the surrounding water molecules. The slight dipole moment that exists in the polar R-groups of amino acids also influences their interaction with water. It is, therefore, understandable that the majority of the amino acids found on the exterior surfaces of globular proteins contain charged or polar R-groups.

Van der Waals Forces

There are both attractive and repulsive van der Waals forces that control protein folding. Attractive van der Waals forces involve the interactions among induced dipoles that arise from fluctuations in the charge densities occurring between adjacent uncharged non-bonded atoms. Repulsive van der Waals forces involve the interactions that occur when uncharged non-bonded atoms come very close together but do not induce dipoles. The repulsion is the result of the electron-electron repulsion that occurs as two clouds of electrons begin to overlap. Although van der Waals forces are extremely weak relative to other forces governing conformation, the huge number of such interactions that occur in large protein molecules make them significant in the folding of proteins [27].

1.2.5 Motivations for Studying Protein Folding Problems

The protein folding problem represents an optimization problem (continuous or discrete, depending on the model assumed). Even for simple models that discretize the space on a lattice (grid), it is an *NP*-hard combinatorial problem [91, 136]. A detailed description of different models used for protein folding is given in Chapter 2.

Motivations for studying protein folding are directly connected with the ability to deduce protein functions. Some of these motivations are:

1. being able to predict protein function given the primary structure of the protein (*via* tertiary structure prediction)
2. understanding a number of diseases that are directly caused by protein misfolding, aggregation and fibrillogenesis (some of which include Alzheimer's, Huntington's, cystic fibrosis, prion disease) [92]

3. designing drugs that specifically target certain aspects of a given protein's structure

4. designing proteins with desired structure and function

In addition to posing an important biological problem with multiple practical applications, the folding problem constitutes a significant mathematical optimization problem, that is, finding global minima in highly complex objective function (energy-potential) surfaces.

Experimental methods available for determining protein structure include X-ray crystallography and nuclear magnetic resonance (NMR). In the case of X-ray crystallography, the structure of proteins is determined by measuring directions and intensities of X-ray beams diffracted from a high-quality crystal of purified protein molecules. In comparison, NMR determines the structure of proteins using high magnetic fields and radio-frequency pulses to manipulate the spin states of nuclei. The position and intensities of the peaks of the resulting spectrum reflect the chemical environment. NMR requires purification of a protein but does not require crystallization. However, it is limited to smaller proteins (less than about 20 – 50 kilo Daltons). Thus, computational methods for predicting the structure of proteins are very attractive, since experimental methods (X-ray crystallography and NMR) are highly labour intensive and require purification and / or crystallization of proteins.

1.3 An Overview of Computational Approaches to Protein Folding

There are four major approaches to predicting protein structure:

1. homology (comparative) modeling

2. fold recognition (or sequence - structure threading)

3. novel fold recognition

4. *ab initio* prediction

Progress in the field of protein structure prediction is evaluated by the Critical Assessment of Structure Prediction (CASP) experiments (that take place every two years). CASP measures in a quantitative way (using structural similarity measures such as root mean square deviation, RMSD) the success in predicting the tertiary structure of a given set of proteins whose structure has been determined experimentally but not released to the public.

1.3.1 Homology Modeling

Homology modeling is based on finding a sequence with a similarity greater than $25 - 30\%$ with a known structure (stored in the Protein Data Bank, PDB [12]). When this level of similarity is found, it is commonly believed that the two sequences had a common ancestor. Homology modeling approaches use a known structure as a starting point for the 3D structure of the sequence. They take advantage of the fact that closely-related proteins have significant sequence similarity and therefore close structural similarity. The drawback of this method is that for every unknown sequence of interest, there has to be a relatively close homologue present in the database (PDB), which is not always the case. Usually when sequence identity is greater than 70%, a high prediction accuracy is possible.

1.3.2 Threading

Fold recognition (also known as threading or inverse folding) is usually used on sequences with a sequence identity less than or equal to 30% to sequences of known structure [15]. Proteins displaying this level of homology are believed to be either homologous (evolved from the same ancestor) or analogues (similarity is a result of convergent evolution). This approach stems from the observation that proteins with a level of sequence similarity that could not be detected by conventional homology searches sometimes adopt a particular three-dimensional fold. By aligning the residues in the unknown structure with the residues in a known fold, the tertiary location of the target residues can specify secondary structure, exposure, and spatial interaction with other residues. There are two known ways of threading: dynamic programming and the use of knowledge-based potentials. In the case of knowledge-based potentials, spatial interactions are often modeled by empirically-derived values quantifying the desirability of a residue of type i being a given distance from a residue of type j. The key problem for this approach is poor alignment between residues of interest and target residues [17].

1.3.3 Novel Fold Recognition Methods

This approach was known in earlier CASPs as *ab initio* fold prediction, but was renamed in CASP4 to better define current methodologies that are being used – particularly, to separate methodologies that use sequence homology from those that do not. Unlike pure *ab initio* prediction, novel fold recognition uses threading, and fragments from existing protein structures, as well as secondary structure prediction and multiple alignment techniques, to predict some features of the three-dimensional structure. If prediction is done *via* secondary structure prediction, the

method can be sensitive to errors in secondary structure prediction. After obtaining a seed structure from sequence homology, local optimization techniques (such as Monte Carlo methods [56, 88, 122] and Genetic Algorithms [18]) are used to further optimize the conformation.

1.3.4 *Ab initio* Methods

In cases when there is no significant homology with any sequence of known structure, *ab initio* methods based on energetic principles perform a search in the set of all possible conformations. But since this search space is usually exponentially large, it is very challenging to search efficiently. To accelerate conformational searching, simplified models employ techniques that permit coarse (large-scale) searching of the energy landscape. A variety of methods are used in conjunction with reduced complexity models and simplified potentials to perform broad searches through low-resolution structures, including Metropolis Monte Carlo methods [125, 127] and Genetic Algorithms [29, 103]. Methods developed in this thesis belong to the class of pure *ab initio* methods but can be extended to use sequence homology. A more detailed description of different methods used for coarse-grained searching is given in Chapter 3.

1.4 Thesis Statement

The primary topic of this thesis is the development of efficient local search methods for protein folding and related problems. Within this vast problem area (which has been studied in the literature for over 50 years with limited success!) we focus on three major directions for the development of more efficient search methods:

1. The development of biologically inspired approaches. The protein folding process has been described in the literature as a self-organizing process. Therefore, we ask whether biologically inspired search algorithms based on self-organization phenomena such as *stigmergy* (where an agent modifies the environment, and the decision process of other agents is influenced by this change in the environment) can be used as an efficient search metaphor. One of the most prominent biologically inspired approaches is Ant Colony Optimization, which is inspired by self-reshaping phenomena observed in real ant colonies [40].

2. The development of construction search methods. These methods work on partial candidate solutions, as opposed to complete conformations. We also focus on a combination of a construction-based search and search methods

based on complete conformations. This direction is particularly interesting, since at the present time the field of protein structure prediction is largely dominated by non-construction-based approaches working with complete conformations. Construction-based approaches are promising, since every round of construction that retains some properties of the previously seen candidate solutions is equivalent to switching to a larger search neighbourhood. This efficient move set can surmount larger energetic barriers than the standard bond moves used in the literature. Additionally, construction-based methods have been shown to perform well for other *NP-hard* problems.

3. The development of adaptive search strategies that can react to the amount of progress made during the search and adjust the search strategy accordingly. Most search methods proposed in the literature for the protein folding problem are non-adaptive. Standard methods have difficulty exploring low-energy regions efficiently due to the ruggedness of the search landscapes. Instead we propose to extract important features of the search landscape (by storing a promising and diverse set of local optima) and adapt the search strategy (adjust the amount of diversification *vs.* intensification) based on the search progress made.

This thesis makes contributions in the following areas:

1. in the development and evaluation of the first Ant Colony Optimization (ACO), a construction-based stigmergic stochastic local search algorithm based on the self-organizing phenomenon for widely studied lattice models of protein structure prediction

2. in the development and evaluation of an efficient construction-based search method for the identification of folding nuclei (structural regions that are important for driving the folding process)

3. in the development and evaluation of adaptive search strategies (based on a novel bin framework for storing promising local optima) that can be used in conjunction with current state-of-the-art methods for *ab initio* and *de novo* folding

1.5 Organization of the Thesis

Chapter 2 presents the protein folding problem and the closely-related problem of identifying folding pathways. Furthermore, in this chapter we introduce a set of basic notions and some widely used simplified models of protein structure. Chapter 3 describes general classes of search methods used in the literature for solving

protein folding problems, summarizes related work, and outlines the shortcomings of existing methods that motivated the algorithms introduced in this thesis. In Chapter 4, we introduce an Ant Colony Optimization algorithm for the 2D and 3D HP protein folding problem, describe major components of the outlined approach, and provide an empirical evaluation. In Chapter 5, we introduce an adaptive bin framework for the FCC β-sheet protein folding problem and compare its performance with results from the literature. In Chapter 6, we introduce a construction-based search algorithm for identifying folding pathways, provide theoretical and empirical justification for the proposed approach, and compare results with the experimental data available. Finally, in Chapter 7, we discuss how the approaches introduced in this thesis relate to each other in particular, and to the field of development of efficient search methods for protein structure prediction in general. We also discuss strengths and weaknesses of the proposed methods and outline some suggested directions for future research.

Chapter 2

Description of Problems, Models, and Notations

We can't solve problems by using the same kind of thinking we used when we created them.

Albert Einstein

In this chapter, we define the two problems addressed in this thesis: the protein folding problem and the closely-related problem of identifying folding pathways. Additionally, we introduce models and objective functions (energy potentials) that are used for each of the problems.

The *Ab Initio* Protein Folding Problem is the problem of prediction of protein tertiary structure from its amino acid sequence for a given energy function. It was proven that this problem and several variations of it are $\mathcal{NP}$-hard even when conformations are restricted to a grid, as in the case of lattice models (the proof was conducted for diamond and cubic lattices) [91, 136]. The technical definition of this problem is given in Chapter 4.

The Problem of Identifying Protein Folding Nuclei and Folding Pathways is the problem of identifying a set of native contacts that play an important role in folding along with identifying a time-ordered sequence of their formation. The technical definition of this problem is provided in Chapter 6.

The problem of identifying folding pathways without relying on the availability of the native state is believed to be generally more difficult than the protein structure prediction problem [66]. The only variant of this problem that has been extensively addressed in the literature is the one outlined above, when folding pathways are deduced from either the native state [2, 32, 36, 107, 144, 151] or additional experimental results [97]. There are no complexity results available for the search for folding nuclei, but it is widely believed that this problem is difficult [66, 144].

To solve these computationally difficult search problems, a number of reduced models have been introduced by biochemists and physicists. These models allow more efficient searching of the possible conformations for the protein structure prediction problem and more efficient searching of the possible folding pathways

for the folding pathway problem. Methods for reducing protein structure can be divided into two major classes: lattice and off-lattice models.

The main advantages of lattice models include the ability to develop efficient computational techniques for storing conformations, testing their self-avoidance, and calculating contact energy terms (long-range potential); the possibility of pre-calculating and storing in tables entire sets of some conformational transitions (lattice moves), and some elements of the energy function (for example, short-range potential); and reduction of the conformational search space. The main disadvantages of lattice models are related to the distortion of local protein geometry and a limited resolution of the model (obtained by calculating the root mean square deviation between the model and the native state). For low-complexity models, for example, the Face Centered cubic lattice (12 vectors), the best resolution achieved is approximately 2 Å; for high-complexity models, for example, Side Chain Only (646 vectors) and C_α-C_β-Side Groups (800 vectors), a resolution less than $1-2$ Å is possible. In general, resolution also depends on the protein's size and secondary structure content [66].

The main advantage of discrete off-lattice models is the ability to achieve high geometrical and resolution accuracy (less than $1-2$ Å) while still keeping low complexity [100]. The primary disadvantage of off-lattice models is loss of computational efficiency due to inability to test self-avoidance easily, slow calculation of long-range contact energy, and the impossibility of efficient pre-computation of a conformational move set.

To address the protein folding problem, we need to consider the following three issues:

1. the design of the model, *i.e.* the choice of the protein structure's representation (which has a desired level of accuracy)

2. the definition of the energy potential (energy potential should be able to discriminate between native and non-native states of proteins for the protein folding problem, and between entropically-favourable and non-favourable folding pathways for the folding pathways problem)

3. searching through the space of possible solutions or sample from the model in an efficient way to find the solution as fast as possible

In this chapter, we discuss the design of models and the definition of energy potentials chosen, and provide justifications for specific choices made. In the next chapter, we address the issue of searching through the space of possible conformations.

2.1 Protein Folding on the Lattice

To address the protein folding problem, we chose to concentrate on lattice models of protein representation, primarily due to the unavailability of a single universal energy function for off-lattice folding and a lack of comprehensive comparison results for state-of-the art methods using off-lattice models. Additionally, lattice models have a number of advantages that can be attributed to regular discretization of the space, conceptual simplicity, and wide usage in the protein folding literature. For small proteins, an extensive study of compact conformations on the lattice is possible and the evaluation of energies can be performed quite efficiently. However, lattice methods have a somewhat restricted ability to accurately represent secondary structure and backbone conformation. Recently, it has been shown that the computational advantages of lattice models outweigh the problems associated with their biases [66]. It has also been shown that lattice models can provide some valuable insights into the protein folding problem, including information about folding stages, comparing energy potentials, and testing optimality of the native state [15].

2.1.1 Square and Cubic Lattices

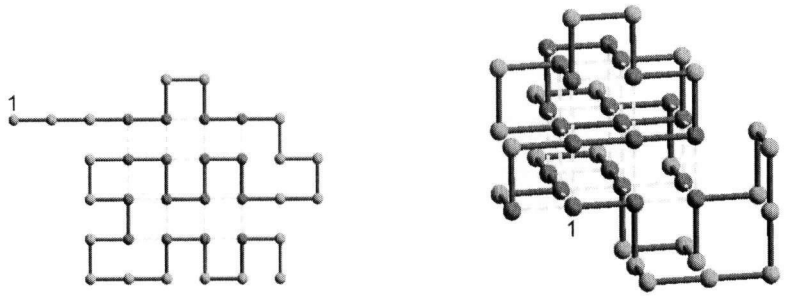
Historically, the most common and most widely studied lattices in the field of protein folding are square and cubic lattices. The protein conformations of the sequence are restricted to self-avoiding paths on a lattice. For the 2D model, a 2-dimensional square lattice is used; in the case of the 3D model, a 3-dimensional cubic lattice is commonly used. Examples of protein conformations on the 2D and 3D lattices with two types of amino acids (monomers) are shown in Figure 2.1.

2.1.2 Hydrophobic Polar Energy Potential

The Hydrophobic Polar (HP) model is based on the observation that the hydrophobic force is the primary driving force of folding in globular proteins. Most enzymes fall into this class [72].

In the HP model, every amino acid of a given protein is reduced to a single point on a lattice, and classified as either *hydrophobic* or *polar*. *Hydrophobic* residues are those that avoid contact with the aqueous environment, and therefore are more likely to be found in the interior of the protein, forming a *hydrophobic core*. *Polar* amino acids, in contrast, usually bear a charge; therefore, they interact with aqueous environments and can be found on the surface of the protein. The hydrophobic force is the only force considered by the HP model.

In the HP model, the energy is usually given by specifying a symmetric matrix of interaction energies between *hydrophobic* (H) and *polar* (P) residues that are



(a) (b)

Figure 2.1: (a) An example of a protein conformation on the 2D square lattice; (b) an example of a protein conformation on the 3D cubic lattice. Two types of monomers are considered: hydrophobic amino acids are colored in red, polar are green, yellow dotted lines represent energy interactions.

topological neighbours on the grid but not immediate neighbours in the sequence:

$$\begin{pmatrix} \varepsilon_{HH} & \varepsilon_{HP} \\ \varepsilon_{PH} & \varepsilon_{PP} \end{pmatrix}$$

In the standard HP model, the energy of a conformation is defined as the number of topological contacts between hydrophobic amino acids that are not neighbours in the given sequence, thus, $\varepsilon_{HH} = -1$ and $\varepsilon_{HP} = \varepsilon_{PH} = \varepsilon_{PP} = 0$. More specifically, a conformation c with exactly n such H-H contacts has free energy $E(c) = n \cdot (-1)$; for example, the conformation shown in Figure 2.1, part (a) has energy -14. Other variations of the energy potential for the HP model exist (see, *e.g.*, [21]).

While the HP model is most intuitively defined in 3D to match the geometry, self-avoidance, and entropy of real proteins [105], in fact the 2D model captures some properties of real proteins as well. Particularly, the perimeter-to-area ratio of a short 2D chain is a close approximation of the surface-to-volume ratio of the native states of proteins [22].

2.1.3 Face-Centered Cubic Lattice Model

Another important representative of lattice models is the Face-Centered Cubic (FCC) lattice. The FCC lattice is the usual lattice structure for the majority of crystalline metals. Even though the square and cubic lattices were explored the

most in the literature and both are important for empirical comparison with other existing approaches, the FCC lattice is preferred for predicting protein structure due to its ability to have low complexity (12 vectors) and still model real protein conformation with the best quality (coordinate root mean square deviation below 2 Å) when compared to other simple lattices [100]. It has also been shown that local packing of amino acids in proteins closely resembles a distorted FCC lattice [7], and that the FCC model has a reasonable description of secondary structure elements; furthermore, hydrogen bonding is geometrically accurate as compared to real proteins [66]. As a result, the FCC lattice is considered the best overall choice among the simpler regular lattices [66].

In the FCC model, the polypeptide is restricted to a face-centered cubic lattice [106] that has 12 base set vectors (shown in Figure 2.2):

$$v_{base} = \{\mathbf{e}_1, \mathbf{e}_2, \ldots, \mathbf{e}_{12}\}, \tag{2.1}$$

where $\mathbf{e}_1 = (1,1,0), \mathbf{e}_2 = (1,-1,0), \mathbf{e}_3 = (1,0,1), \mathbf{e}_4 = (1,0,-1), \mathbf{e}_5 = (0,1,1), \mathbf{e}_6 = (0,1,-1), \mathbf{e}_7 = (0,-1,1), \mathbf{e}_8 = (0,-1,-1), \mathbf{e}_9 = (-1,0,1), \mathbf{e}_{10} = (-1,0,-1), \mathbf{e}_{11} = (-1,1,0), \mathbf{e}_{12} = (-1,-1,0)$.

A protein chain of N residues is described by $N-1$ vectors, where vector $\mathbf{v}_i$ connects residues i and $(i+1)$. The 12 base vectors allow for the following valence angles between each pair of vectors: 60, 90, 120, and 180 degrees [52].

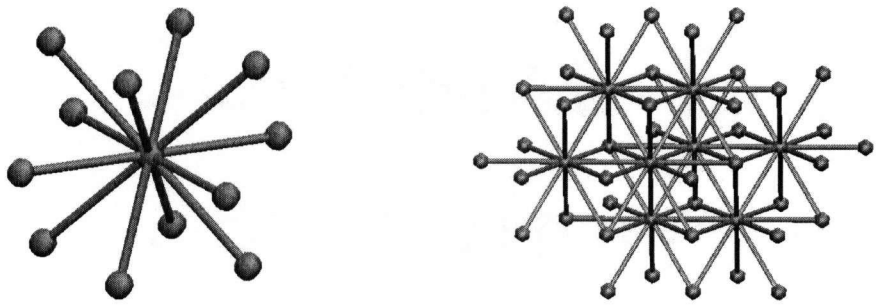
2.1.4 Beta Sheet Energy Potential Used with the FCC Lattice

To model β-sheet proteins and the stiffness of the polymer chain, the following definition of an extended, β-type chain conformation was defined in [52]: three vectors are in an extended state if

1. the angles between vectors $\mathbf{v}_{i-1}$ and $\mathbf{v}_i$ and between $\mathbf{v}_i$ and $\mathbf{v}_{i+1}$ are greater than 90 degrees
2. the dot product $\mathbf{v}_{i-1} \cdot \mathbf{v}_{i+1}$ is larger than 0, which means that the angle between vectors $\mathbf{v}_{i-1}$ and $\mathbf{v}_{i+1}$ is less than 90 degrees

The energy potential for this model is composed of two terms: the short-range potential $U_{i-1,i,i+1}$ that depends on three consecutive vectors in the chain $(\mathbf{v}_{i-1}, \mathbf{v}_i, \mathbf{v}_{i+1})$ and mimics conformational propensity to form an extended set of β-strands:

$$U_{i-1,i,i+1} = \begin{cases} -\varepsilon_B, & \text{if the triple } (\mathbf{v}_{i-1}, \mathbf{v}_i, \mathbf{v}_{i+1}) \text{ is in extended state,} \\ 0, & \text{otherwise} \end{cases} \tag{2.2}$$



(a) (b)

Figure 2.2: (a) Depiction of the 12 base vectors for the FCC lattice model; (b) an example of several FCC lattice cells.

and the long-range potential $V_{i,j}$ for two non-bonded chain residues, defined as:

$$V_{i,j} = \begin{cases} +\infty, & \text{for } r_{i,j} = 0 \text{ and } i \neq j, \\ -\varepsilon_A, & \text{for } r_{i,j} = 1 \text{ (in lattice units) and } |i-j| > 1, \\ 0, & \text{for } r_{i,j} > 1 \text{ (in lattice units)} \end{cases} \tag{2.3}$$

where $r_{i,j}$ is the lattice distance between residues i and j.

For a chain of length N, the total energy is defined as:

$$E = \sum_{i=2}^{N-1} U_{i-1,i,i+1} + \sum_{i=1}^{N} \sum_{j=1}^{N} V_{i,j}(1 - \delta_{ij}), \tag{2.4}$$

where δ_{ij} is the Kronecker delta ($\delta_{ij} = 1$ when $i = j$, and 0 otherwise) and the values of the force field parameters are defined as follows $\varepsilon_A = 1.0$ [ε_0] and $\varepsilon_B = 4.0$ [ε_0] (ε_0 is the unit of interaction energy and $\varepsilon_0 = 1.0$) as in Gront *et al.* to model a semi-flexible polymer [52].

This energy potential was chosen (in combination with the FCC lattice), since the accuracy of protein structure prediction methods for β-sheet proteins is the lowest among other structural classes of proteins [1]. Additionally, it exhibits

characteristics of more complicated energy potentials used for off-lattice models [52, 106], particularly, cooperative all-or-none folding transition, characteristic interplay between short- and long-interactions, secondary structure propensity.

2.2 Off-Lattice Protein Folding

In this section, we briefly summarize available and commonly used off-lattice models and provide an overview of potential energy functions. Even though our approaches are introduced and evaluated for lattices only, they can easily be extended to a discrete off-lattice case. We did not concentrate on discrete off-lattice representation due to the unavailability of universally used energy functions and the absence of an appropriate data set for which empirical results of the best-performing algorithms in the literature exists.

The two major difficulties that *ab initio* protein folding faces and that are further amplified in the off-lattice case are the exponentially large number of possible conformations, and the lack of the potential (the energy function) to discriminate between native folds and misfolded proteins. The usual approach for dealing with an exponentially large continuous search space is to simplify the protein structure representation, and to allow these simplified models to adopt only a small discrete number of conformational states [15, 100].

2.2.1 Models for Off-Lattice Protein Folding

The primary reason for choosing off-lattice models over lattice models is to obtain a better geometrical accuracy. Reduced or discrete state off-lattice models usually fix degrees of freedom of all side chains and assume that all bond lengths are constant. Internal (angular) coordinates for the main chain atoms are usually kept according to the "ideal geometry" of Alanine [113].

There are three generally acceptable ways in which side chains are modeled in reduced off-lattice models:

1. to ignore side chains and model only C^{α}, while side chain properties (such as secondary structure propensities, hydrophobicity values, participation in disulfide bridges *etc.*) are taken into account in the energy potential [29, 75, 113]

2. to use a single virtual atom to represent each side chain at the center of mass of the heavy atoms in the side chain [131, 146]

3. to find a conformation of the backbone and then add side chains and model them using rotamer libraries, optimizing their positions [43]

The backbone conformation is modeled by the (ϕ, ψ) dihedral angle pair for each amino acid. This backbone representation effectively reduces the allowed conformational space of the protein backbone, while uniquely defining atomic coordinates of the protein. Most discretized off-lattice models consider only m alternative (ϕ, ψ) states representing various alpha-helices, beta-strands, and loop conformations, where m = 4 to 32 [29, 100, 113]. The set of m allowed angle pairs is chosen by fitting the backbone coordinates to the coordinates of representative natural proteins.

The choice of model is particularly important in the off-lattice case, since each model is limited in its possible representational accuracy (which determines the predictive accuracy of protein conformations), and since off-lattice models are primarily chosen on the basis of their accuracy. The two discrete off-lattice models that show good prediction accuracies are the seven-state model used by Rooman *et al.* [113] and the four-state model by Park and Levitt [100]. Searching a conformational space using reduced off-lattice models still remains computationally very challenging, even for small proteins. After obtaining a reduced representation of the native fold of a protein, reduced models can be accurately reconstructed to all-atom protein representations [100].

In practice, most methods currently used in novel fold predictions (apart from using a reduced off-lattice model and searching in the discrete space) make extensive use of available structural information. This is done by developing scoring energy potentials that use structural information accumulated in protein databases, by using secondary structure predictions that employ homology information, and by choosing structural fragments from a database of known proteins to reduce the size of the conformational space during the search phase [18, 63, 122, 126, 131]. For this reason, the *ab initio* category was renamed "novel fold prediction" in CASP4 [89].

A number of novel fold methods including ROSETTA, which is currently the most successful method in the novel fold prediction section of CASP competitions [1, 16], enhance the search process by using a large library of short fragments. These fragments consist of a small number of residues with a specified three-dimensional structure [122]. Short segments of the sequence of interest are restricted to the local structures adopted by closely related sequences in the protein structure database [18, 63, 122].

Many novel fold prediction methods use well-established secondary structure prediction tools. These methods assemble tertiary structure from candidate fragments of secondary structure, either by fixing secondary structure as hard constraints or using it as soft constraints [56, 63, 88].

Most successful novel fold prediction methods represented in CASP use a combination of approaches based on threading, lattice folding, clustering, and structural

refinement using more detailed off-lattice models [125].

2.2.2 Energy Potentials for Off-Lattice Protein Tertiary Structure Prediction

Success in solving the protein structure prediction problem relies heavily on the choice of a potential energy function that can accurately discriminate between natively folded and misfolded protein conformations. Three types of energy potentials have been developed in the literature [25, 98, 99, 123]:

1. Physical potentials: based on empirical measurements of interactions between atoms, these functions are typically parametrized on data from small proteins [64, 141].

2. Knowledge-based or statistical potentials ("potentials of mean force", database-derived): based on the statistical properties and features of a database of proteins of known native structures, and transformed into various free energy parameters using statistical measures that calculate the ratio between observed and expected values (*e.g.*, the number of contacts, the distance between amino acids, the number of atoms in contact between different amino acids) [87, 99, 124, 137]. The interactions can be either distance-dependent or only contact-dependent.

3. Learning-based potentials: an energy function is derived that maximizes the difference between correct and incorrect structures (stability gap) by training using machine learning methods or linear optimization of parameters [141].

There are advantages and disadvantages to using each of these potentials. Physical potentials are mostly useful for simplified models of protein structure where not all of the forces are considered, since the exact potential energy governing protein folding is not known [141]. Knowledge-based potentials have achieved some degree of success but they suffer from the following limitations [25]:

1. The theoretical basis of these potentials is weak. There is no reason to believe that the interactions in proteins in the respective ground states of an ensemble of biological proteins would obey Boltzmann statistics.

2. These potentials assume statistical independence of various interactions but from theoretical studies of protein folding it has been shown that correlations between interactions may arise to facilitate the folding process.

3. Some contributions can be underestimated due to the selection of the reference state or neglection of chain connectivity.

Learning-based potentials define for each protein a Z-score that represents the difference in energy between the correct and random states, divided by the standard deviation of the energy levels of random conformations. The best energy potential would be the one that maximizes Z-scores for all the proteins in the database [25]. To apply this method to a given protein model, one needs knowledge of both the native and non-native states of proteins under this model, which is computationally expensive [55].

One commonly accepted way to test a given energy potential function is to develop a set of *decoys* (a mixture of native and non-native protein folds) and then study the ability of different potentials to detect native structures [98, 99, 123]. The decoys are generated in different ways: putting the sequence from a particular protein onto the backbone of another protein, lattice models, fragment building, threading, or other alignment methods [99, 141].

Currently, there is no universal energy function that performs consistently well for multiple sequences. Therefore, most approaches used in CASP use different linear combinations of energy components based on all three types of energy potentials described previously; this makes comparison and development of novel search methods increasingly difficult. Therefore, in this work we restrict ourselves to lattice folding, and will consider off-lattice folding in future work.

2.3 The Problem of Folding Pathway Identification

The second problem we address, that is, the problem of identifying folding pathways (or folding nuclei) in native protein structures, is directly related to the problem of protein folding. In the general sense, when no information about the native state is used, it is a more difficult problem. It is now widely accepted that proteins fold by a "nucleation and growth" kinetic mechanism [36, 47]. A folding nucleus is defined as the critical set of interactions (the minimal stable element of structure) that must be present in order for the folding to result in a rapid assembly of the native state [35, 36].

The problem of identifying folding nuclei from the large number of native structures currently available in the Protein Data Bank remains unsolved and of great importance. Accurate predictions of folding nuclei can lead to a detailed description of the hierarchy of the folding process, make it possible to identify contacts that have high structural importance, and thus yield possible improvements in solving the ultimate protein problem – the folding problem – by restricting the conformational space that needs to be searched. Insights obtained during the study of folding nuclei can also be useful for rational protein design.

2.3.1 Models Used for the Identification of Folding Pathways

A variety of models have been introduced for the identification of folding nuclei. They range from lattice representations of proteins [65] (including 3D cubic lattices, as described previously), and reduced continuous C_α - C_β models similar to the off-lattice models described above (which are used in Molecular Dynamics simulations [36]), to all-atom models of protein structure [107]. Lattice models are very popular for protein folding. However, studying the structure and formation process of folding nuclei requires models with higher accuracy, since they impose unphysical constraints (distorted geometry) when information about the geometry of the native state is available. In order to search efficiently, a model has to be sufficiently reduced. We used the discrete C_α model in our approach and a graph representation of native contacts.

A protein is represented by a polymer graph (a connected undirected weighted graph) with one node per residue. Two residues are connected by an edge, if they are in contact in the native state. We used the following contact definition: if any of the heavy atoms of the side-chain or the C_α of one residue occur within the cut-off radius of 4.55 Å from heavy atoms of the side chain or the C_α of the second residue [26], residues are in contact. As in [26], native contacts between pairs of residues (i, j) with $j \leq i + 3$ are discarded, as such residues interact due to chain connectivity. Every edge of the polymer graph is augmented by the value of the objective function, discussed in the next section.

2.3.2 Objective Functions Used for the Identification of Folding Pathways

The majority of methods described in the literature for finding folding nuclei [2, 32, 36, 107, 144, 151] only consider topological (entropic) aspects of the protein energy surface, since the precise energetics of the protein folding process is still unknown. Nucleation during the folding process is a kinetic process and therefore, cannot be precisely determined from just the topological properties of the native state alone (a single nucleation phase). However, as shown by Φ-value analysis, transition states tend to be native-like, that is, have many native contacts [35]. Furthermore, it is known from experimental analysis that the locations of folding nuclei depends on the topology of the native state [65] and that the mechanism and speed of folding directly depends on the topology of the fold [104].

Different approaches capture the geometry (topology) of the native state in various ways. In this work, we adapted an objective function based on the notion of the effective contact order (ECO) first introduced by Dill [144]. While the contact order (CO) is the sequence separation $CO = |i - j|$ between two contacting

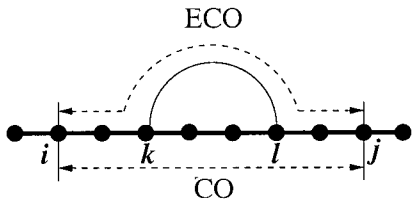


Figure 2.3: The effective contact order (ECO) between i and j is equal to 5, compared to the contact order (CO) of 7 between the same residues.

residues numbered i and j [104] and does not depend on the order of contact formation, the ECO of a contact is dependent upon other contacts already formed in a particular configuration. Thus ECO values are folding pathway dependent and directly relate to the entropy of folding.

Effective contact order of a newly-added contact is defined as the effective loop closure size (the number of steps, covalent and non-covalent links taken along the shortest path on the polymer graph), given that other contacts have been formed, as shown in Figure 2.3. It has also been shown that low-ECO pathways often coincide with the lowest free-energy pathways for simple models, such as the hydrophobic polar model. This observation also holds for real protein structures [144].

Thus, the objective function to be minimized in this case is the sum of individual effective contact orders for each contact (i, j):

$$ECO_{total} = \sum_{i,j \in \{e_1,\ldots,e_m\}, i \leq j+3} ECO(i, j, A_t) \tag{2.5}$$

where m is the total number of contacts $\{e_1, ..., e_m\}$ or edges in the polymer graph, and i, j iterate only through the set of the non-covalent native contacts (contacts between residues fewer than 3 residues apart are discarded due to chain connectivity). A_t is the set of edges present at the time t when contact (i, j) is added, $t \in [0, m-1]$, $A_0 = \emptyset$, and $A_{t+1} = A_t \cup \{(i, j)\}$.

Since the $ECO(i, j, A_t)$ of each contact depends on what contacts have already been formed, this problem is an optimization problem similar to the protein folding problem and maximizes the absolute value of the total energy of contacts formed.

Chapter 3

Background and Related Work

What makes the desert beautiful is
that somewhere it hides a well.

Antoine de Saint-Exupery

Search methods for hard optimization problems depend on the topology of the search space, which in turn is defined by the two issues discussed in Chapter 2: the choice of the model (more specifically, the resolution of the model – the representation of a protein – determines the size of the search space) and the choice of the potential energy function (an objective function defines the landscape of the search space). The speed of exploration of the search space for these exponentially large spaces is important. It ranges from Molecular Dynamics Simulations that directly integrate Newton's equation of motion (taking the very small step sizes required for numerical stability) to accelerated conformation or pathway searching that permits coarse sampling of energy landscapes. In this chapter, we provide an overview of the topology of the space, search methods, and particular algorithms that rate among the best for the problems of protein folding and identification of folding pathways from the native state topology.

3.1 The Protein Folding Problem

This section focuses on the problem of searching conformational space efficiently, with the goal of finding the global optimum as it applies to systems with complex search landscapes, and particularly to the protein folding problem. First, we will briefly summarize existing optimization methods for protein folding. Then, we will categorize the methods and specify the relationships between the stochastic local search methods employed for this problem. Finally, we will discuss promising directions for the development of efficient search methods for problems with complex energy landscapes, some of which have been explored in this thesis.

One important distinction that has to be made in the context of the protein folding problem is the difference between *sampling* and *searching*. The goal of

sampling is to generate a correct ensemble (defined by the Boltzmann probability) for the calculation of physical quantities; the goal of searching is to find a configuration or a set of configurations with a required property (usually a global energy minimum). In general, since searching involves finding one conformation with specific properties instead of making sure that the ensemble of conformations found conforms to a certain sampling requirement, searching can be more efficient than sampling. Unmodified Molecular Dynamics and Monte Carlo are efficient sampling methods, but inefficient search methods. Here and throughout the thesis we use the term 'efficient algorithm' to describe several desirable properties of a stochastic local search algorithm; besides clean design and reasonable space requirements, the main desired property is speed, the time it takes for an algorithm to obtain the global optimum or high-quality solutions. The term "efficient" as applied to stochastic local search algorithms (that may require exponential time to solve certain classes of problem instances) does not imply that algorithms have polynomial time complexity.

There are a number of search methods applicable to the protein folding problem that can be used in conjunction with reduced complexity models and simplified potentials to perform a broad search through low-resolution structures. The most widely used methods include Metropolis Monte Carlo and Simulated Annealing [63, 95, 122, 125, 153], as well as Genetic Algorithms [18, 29, 103]. The size of the neighbourhood considered in each local move of the search can be quite significant, and can allow a much more rapid sampling of the conformational space. For example, a simple change in torsion angle space produces large changes in Cartesian coordinates, fragment-based procedures speed up sampling by allowing jumps between different local structures in a single step, secondary structure restricts the size of the search space and can also contribute to more coarse sampling, and the usage of templates for search initialization aids the search.

Computer simulations suggest that the energy landscape along the folding pathway of the protein is often quite rugged, and usually passes through stable intermediate states with low energy [92]. In particular, so-called *molten globule* intermediates have been extensively studied [92]. In protein folding, a rugged landscape poses a problem since the system has to cross barriers that can be larger than the deterioration of the potential energy function an algorithm searching for low-energy states is willing to accept. These energy barriers arise due to at least three classes of interactions: local barriers separating stable states of the torsion angles (as observed from Ramachandran plots); and close encounters between side-chain atoms and backbone atoms that repel one another once they get too close. Additionally, energetic barriers occur due to the role that non-native interactions play during folding [11]. The most challenging feature of the protein folding problem is the fact that the objective function (energy) has a large number of local minima.

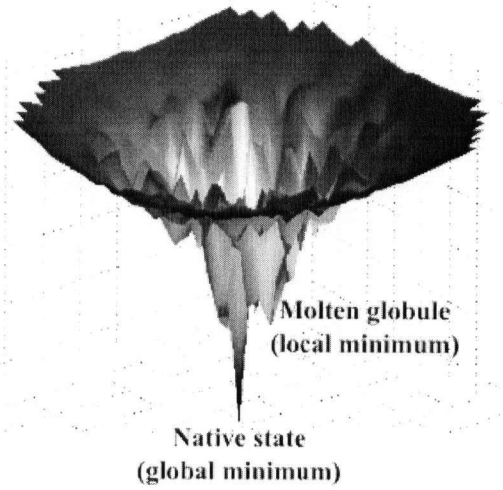


Figure 3.1: An example of a protein's funneled energy landscape.

Therefore, local optimization is likely to get arbitrarily stuck in one of them, possibly far away from the desired global minimum. Estimates of the number of local minima for the protein folding problem range from 1.4^n to 10^n for a protein with n residues. However, in most cases a search will not encounter all of them [27].

It is believed that evolution has generally led to funneled energy landscapes, with the native state at the base of the funnel, by requiring robust folding; this property is also called the principle of minimal frustration [105]. At the same time, the landscape is still rugged, with many local optima resulting from non-native contacts that can lead to specific folding intermediates. A typical example of the funneled energy landscape of a protein is given in Figure 3.1. It is generally hypothesized that improvements in energy function can lead to smoother search landscapes, and can result in obtaining more guidance from the potential energy function that will speed up the search [15].

3.1.1 Search Algorithms Used for Protein Folding

Two of the most commonly used algorithms for protein folding are Monte Carlo methods (and their variant, Simulated Annealing) and Genetic Algorithms.

Monte Carlo and Simulated Annealing

Monte Carlo (MC) methods are the most widely used methods for searching and sampling complex energy landscapes in statistical physics. Specifically, generalized ensemble methods based on the Monte Carlo method represent the state-of-the-art for *ab initio* protein structure prediction using discrete-state models [86, 94, 130].

A widely used set of Monte Carlo methods, known as Markov chain Monte Carlo (MCMC), use *Markov chains* to explore the state space. In this section, I will restrict the discussion of *Markov chains* to discrete state spaces, but note that the presented definitions and convergence requirements extend naturally to general state spaces.

A process is called a *Markov process* if the conditional probability $P(X_{t_n} = S_{i_n}|X_{t_{n-1}} = S_{i_{n-1}}, ..., X_{t_1} = S_{i_1})$ is independent of all states $S_{i_1}, ..., S_{i_{n-2}}$ but the immediate predecessor [71]. This condition results in a search procedure without memory that is inefficient in searching for a global optimum but is efficient for sampling, since the probability of visiting a particular conformation depends only on the energy of the proposed conformation and the energy of the current state.

The transition probability from state S_i to state S_j is defined as:

$$W_{ij} = W(S_i \rightarrow S_j) = P(X_{t_n} = S_j|X_{t_{n-1}} = S_i), \tag{3.1}$$

where it is required that $W_{ij} \geq 0$ and $\sum_j W_{ij} = 1$. The total probability $P(X_{t_n} = S_j)$ that at time t_n the system is in state S_j is then equal to:

$$\begin{aligned} P(X_{t_n} = S_j) &= \sum_i P(X_{t_n} = S_j|X_{t_{n-1}} = S_i)P(X_{t_{n-1}} = S_i) \\ &= \sum_i W_{ij}P(X_{t_{n-1}} = S_i). \end{aligned}$$

A Markov chain is said to be *ergodic* if it converges to an equilibrium distribution in the limit when time goes to infinity [3, 59]. This distribution is also known as the stationary or invariant distribution of the process and for physical systems in thermal equilibrium it corresponds to the Boltzmann distribution. To ensure ergodicity, the chain must satisfy the following properties [3]:

1. Irreducibility – for all states S_i and S_j, there exists a time t, such that $W_{ij}^t > 0$, *i.e.* the transition graph represented by the transition matrix W is connected; thus, the matrix W cannot be reduced to separate smaller matrices

2. Aperiodicity – chain does not get trapped in cycles; thus, ensuring that the stationary distribution is unique

The above conditions are hard to verify in practice. To surmount this difficulty, one often adopts an easier to verify alternative condition for convergence known as *detailed balance*. This is a sufficient, but not necessary condition [3]. The *detailed balance condition* is defined as [71]:

$$W_{ji}P(X_{t_n} = S_j) = W_{ij}P(X_{t_n} = S_i). \tag{3.2}$$

In practice, when the conformational space to be searched is exponentially large, a simulation becomes *quasi ergodic*. This means that trajectories between structurally diverse but statistically important energy minima have low (close to zero) transition probabilities that can result in the search or sampling process getting trapped in isolated minima; thus, the simulation may not necessarily reach the equilibrium and may not converge.

Monte Carlo (MC) and Simulated Annealing (SA) algorithms [82] generate a trajectory of states for a system. Each state is evaluated using an objective function – potential energy. If the value of the objective function (E) decreases, a new state is accepted. If it increases, the new state is accepted with a probability dependent on the difference in the Boltzmann weights between the states, proportional to $e^{-\Delta E/(k_B T)}$, where k_B is the Boltzmann constant. While the temperature T is kept constant in MC, it is varied according to some schedule (usually slowly decreasing) in SA. Extremely slow schedules are inefficient, and extremely fast schedules will get the system trapped in the local optima. Thermal annealing can be repeated multiple times, thus producing thermal cycles [37].

Generalized Ensemble Monte Carlo Methods

As mentioned above in the *canonical simulation*, the probability of crossing an energy barrier is proportional to the energy difference and the temperature (employing the Boltzmann probability):

$$P_B(E,T) \propto e^{-\beta \Delta E/k_B T}, \tag{3.3}$$

where $e^{-\beta \Delta E/k_B T}$ is the Boltzmann weight, the energy difference $\Delta E = E_j - E_i$, and the inverse temperature $\beta = 1/k_B T$. This results in a probability density function over energy levels that is bell-shaped, with a maximum around average energy at temperature T. It decreases exponentially with T. The barrier height is measured in units of $k_B T$. Thus, in canonical Monte Carlo (MC), very high and, more important, very low energy configurations are rarely sampled. To overcome

these problems, a number of generalized ensemble Monte Carlo methods have been developed [86]. Generalized ensemble methods compute the density of states (or related physical quantity) to perform a uniform sampling rather than sampling the states of the system using the Boltzmann weight (as is the case with Monte Carlo methods).

Here we provide a brief overview of the most widely used generalized ensemble methods.

1. **Multicanonical Algorithm** (MUCA) (other names for this method are Entropic Sampling, Flat Histogram method, Adaptive Umbrella Sampling, Uniform Sampling in energy) [86].

   The objective of this method is to overcome energy barriers in the rugged landscape by performing a random walk in the energy landscape (where ideally all the states are sampled with the same probability).

   To achieve this, configurations with energy E are updated with non-Boltzmann (multicanonical) weights [86]:

   $$w_{mu}(E) \propto e^{-\beta_0 E_{mu}(E;T_0)} = n^{-1}(E), \tag{3.4}$$

   where β_0 is an arbitrary reference inverse temperature $\beta_0 = 1/k_B T_0$, the multicanonical potential energy is defined as:

   $$E_{mu}(E, T_0) = k_B T_0 ln(n(E)) = T_0 S(E), \tag{3.5}$$

   and $n(E)$ is the density of states that quantifies how closely packed energy levels are, $S(E) = ln(n(E))$ is the multicanonical entropy. Thus, a uniform (flat) distribution of energy is obtained:

   $$P_{mu}(E) \propto n(E) w_{mu}(E) = const \tag{3.6}$$

   (all states are sampled with comparable frequency).

   Since the density of states is unknown, multicanonical weights have to be determined iteratively. Simulation is performed using the usual Metropolis criterion. The transition probability from state i with potential energy E_i to state j with energy E_j is given by:

   $$W_{ij} = \begin{cases} 1 & \Delta E_{mu} \leq 0 \\ e^{-\beta_0 \Delta E_{mu}} & \Delta E_{mu} > 0 \end{cases} \tag{3.7}$$

   where

$$\Delta E_{mu} = E_{mu}(E_j, T_0) - E_{mu}(E_i, T_0). \tag{3.8}$$

In the first run, a canonical simulation at a sufficiently high temperature T_0 is performed:

$$E^{(1)}_{mu}(E, T_0) = E, \tag{3.9}$$

and initially w_{mu} is equal to the Boltzmann weight:

$$w^{(1)}_{mu}(E, T_0) = e^{-E/k_B T_0}. \tag{3.10}$$

This run defines the maximum energy value E_{max} under which one obtains flat energy distribution, and is equal to the average potential energy at temperature T_0, ($E_{max} =< E >_{T_0}$) [86]. Above E_{max} we have the canonical distribution at $T = T_0$.

During the simulation, energy (occupancy) histograms $H(E)$ are generated that provide estimates of the probability of finding a configuration having energy E in a multicanonical ensemble. Subsequently, multicanonical weights $w_{mu}(E)$ are "re-weighted" by the estimate of the density of states [86, 94, 130].

The estimated density of states is modified iteratively by recording an occupancy histogram, $H(E)$, which counts the number of visits at each energy level [71]. When $H(E)$ is flat in the energy range of the random walk, the density of states has converged with the accuracy proportional to the factor f (the convergence factor) used for updating occupancy histograms The factor f is iteratively reduced in the process and histograms $H(E)$ are reset to get the desired accuracy. This iterative process can take significantly long time [86]. Finally, a single long productive run is conducted.

MUCA represents an efficient sampling method, but is an inefficient search method. The product of the estimated density of states and the Boltzmann weight produces an almost flat energy distribution. Sampling this distribution results in a one-dimensional (no temperature T dependence) walk in the energy landscape. It can therefore escape any energy barrier but is unable to search low energy regions efficiently [148, 149], since low-energy regions are not guaranteed to be sampled in great detail because a wide range of energies has to be sampled. Additionally, the weight factors used are derived from previous iterations. Thus, the process has no information concerning unexplored low-energy regions of the landscape, since it does not specifically concentrate on exploring them.

2. **Replica Exchange Monte Carlo** (REMC) (also known as the multiple Markov Chain method and Parallel Tempering) [86].

   A number of non-interacting copies (replicas) at different temperatures are simulated independently by MC. Every few steps, pairs of replicas are exchanged with a specified transition probability. The weight factor is a product of Boltzmann weights (is essentially known).

   In general, each replica can experience its own temperature, pressure or chemical potential. The acceptance criteria for a trial swap are derived from the product of the elementary moves used to construct it.

   A swap of conformations i and j is controlled by the following two parts:

   $i \rightarrow j$ is performed with the following acceptance probability:

$$p_i = min[1, e^{-\beta_i(E_j - E_i)}], \tag{3.11}$$

   where E_i and E_j are the energies of conformations i and j respectively,

   and $j \rightarrow i$ with probability:

$$p_j = min[1, e^{-\beta_j(E_i - E_j)}]. \tag{3.12}$$

   A swap move is a double move, whose acceptance probability is the product of p_i and p_j:

$$p_{ij} = min[1, e^{-\beta_i(E_j - E_i) - \beta_j(E_i - E_j)}] \tag{3.13}$$

   Thus, the transition probability is:

$$p_{ij} = min[1, e^{-(\beta_i - \beta_j)(E_j - E_i)}]. \tag{3.14}$$

   If the difference $(\beta_i - \beta_j)$ in temperature is large, the move has a very low acceptance probability. Thus, trial moves have high acceptance probabilities only if sôme degree of overlap exists between probability distribution functions (histograms) corresponding to neighbouring replicas.

   The probability of exchange $p_{ij} = 20\%$ in the literature is believed to provide a good balance between short-range and long-range moves [94].

   REMC is an example of Umbrella Sampling, where to overcome very high barriers, simulations with different umbrellas (biases) are coupled and their umbrellas periodically exchanged. In general we can't construct a perfect umbrella (bias) for efficient sampling or searching without knowledge of the "true" potential energy surface, but we can do this with umbrellas involving temperatures.

An advantage of this method is that unlike MUCA, it does not require determination of weight factors, since the weights are essentially *a priori* known (defined by Boltzmann probabilities).

The drawback of this method is that as the number of degrees of freedom (N) of the system increases, the required number of replicas also increases ($\sqrt{N}$), where as only a single replica is simulated in MUCA. Improvements of REMC introduced in the literature include hybrid approaches between REMC (for the weight factor determination) and MUCA, or Simulated Tempering production runs [86, 94, 130].

REMC methods are currently the best performing algorithms for *ab initio* protein structure prediction in the context of the search problem [130].

3. **Energy Landscape Paving** (ELP) [53]. This method is an example of a generalized Monte Carlo method that adapts acceptance criteria based on the search history. Hansmann *et al.* [53] introduced temporary energy surface deformation. In this method, the barrier height is decreased proportionally to the time the system stays in the minima (configurations are searched with time-dependent weights, similarly to Tabu Search (TS) that uses a time-dependent adaptive memory in the search [50]).

   Tabu Search does not differentiate between important and non-important regions of the landscape. Energy deformation techniques escape from local optima by deforming or smoothing the energy landscape (*e.g.*, lowering barriers between relevant parts of the search space similarly to other generalized ensemble approaches). Ideally, we are interested in transforming the original energy landscape into a funnel landscape (but it is likely impossible to do this in practice).

   ELP is designed to perform low-temperature MC with a modified energy expression, as follows, to steer the search away from the regions that have already been explored:

$$w(\hat{E}) = e^{-\hat{E}/k_B T}, \tag{3.15}$$

   where $\hat{E}$ is an augmented energy function E:

$$\hat{E} = E + f(H(q,t)), \tag{3.16}$$

   where $f(H(q,t))$ is a function of the time-dependent histogram of energies of states q sampled ($H(q,t)$) updated at each MC step. In ELP, the weight of a local minimum state decreases with the time the system stays in that minimum, thus the probability of escape increases by deforming the energy landscape until the local optimum is no longer favored.

With a short-term memory in the histogram and an infinite cost for 'forbidden' moves ELP is equivalent to Tabu Search. For $f(H(q,t)) = f(H(q))$ the method reduces to various generalized ensemble methods. This method has only been applied on short peptides using detailed atomic representation [53]. Our later work presented in Chapter 5 relates to the current approach and builds on improving sampling of low-energy conformations using adaptive strategies based on the search progress.

Other Algorithms Employed for Protein Folding

To make this high-level overview complete, we mention other methods employed for protein structure prediction, but it should be noted that they are less successful or not as widely used for more realistic models of protein structure representation in the context of *ab initio* folding.

Genetic Algorithms (GAs) [57] optimize the population of solutions by using biologically inspired operations of mutation of the solution and recombination of pairs of solutions (crossover). Possible solutions to a search problem are represented by genes, and a fitness function is used to evaluate the fitness of each gene (candidate solution). GAs have been applied extensively to the problem of protein folding [29, 44, 103, 131]. The advantage of GAs is that a population of solutions can overcome energy barriers by a recombination that is larger than the typical moves employed in MC. The ability of the GA to leave a region of attraction is enhanced by crossing over. However, the GA crossover tends to not work well on systems that are highly coupled and where crossover can be disruptive, which is the case with dense protein configurations.

The size of the population that goes onto the next generation can play the role of temperature in GA [138].

Model-based Search (MBS) is a stochastic local search that focuses the search on promising regions of the search space by storing a certain number of conformations L (local minima) in memory and expanding N conformations in every step of the algorithm [20]. During each step of the search, new local optima are stored and all conformations stored are ranked and pruned based on the scoring function that considers their energies and the radius of a local minimum they represent (the radius is estimated by the distance to the nearest neighbours using root mean square deviation). MBS has been shown to outperform in some cases simple MC used in the ROSETTA algorithm [20].

All of the described earlier methods work on complete conformations after the initial creation of the random initial state. The next method is an example of a construction-based search method that works with partial conformations. In the protein folding literature, construction-based methods for protein folding are

rare due to the problem of *attrition* when the chain runs into itself. The only construction-based method so far restricted to simplified lattice models of folding is the pruned-enriched Rosenbluth method (PERM) of Grassberger *et al.* [8, 60].

The pruned-enriched Rosenbluth method (PERM) is based on a biased sampling (Sequential Importance Sampling) and uses *pruning* (termination of construction) of partial configurations with low statistical weights and *enrichment* (creation of multiple copies) of partial configurations with high weights.

3.1.2 Summary and Classification of Search Methods

Search methods can generally be classified as being either model-based or non-model-based methods (also called instance-based methods [156]). In this context, model-based methods build a parametric or a non-parametric model of the search space that is updated during the search, and the new candidate solutions are generated using the model. Non-model-based search methods generate new candidate solutions using solely the current solution or the current population of solutions; they do not create a model and do not update it based on the search space encountered. The term 'model' is used here as in [156] to refer to an adaptive stochastic mechanism for generating candidate solutions.

Some standard non-model-based methods in their most standard implementation include Monte Carlo, Simulated Annealing, Replica Exchange Monte Carlo, Iterated Local Search, Iterative Improvement, Randomized Iterative Improvement, Variable Neighbourhood Descent, Variable Depth Search, and Evolutionary Algorithms (Genetic Algorithms are in this category). For a description of the stochastic local search methods mentioned in this classification and their application to other combinatorial problems, we refer to [59].

Other stochastic local search methods that build a parametric model (either probabilistic or deterministic) that is updated during the search include Importance Sampling (builds a model of the distribution of interest), Ant Colony Optimization (stores pheromone matrix), Multicanonical Monte Carlo (records histograms of energies), Energy Landscape Paving (accumulates temporary histograms of energies), Tabu Search (records tabu attributes, and adapts the tabu tenure in reactive Tabu Search [9]), and Dynamic Local Search (adapts penalty weights).

Examples of model-based methods that build a non-parametric model (record actual candidate solutions) include Model-based Search (MBS) (conformations are stored for future retrieval), the Pruned-Enriched Rosenbluth Method (parametric: weight thresholds for pruning and enrichment, and non-parametric: retrieval of partial conformations from the memory stack), and some population-based elitist search strategies that make use of candidate solutions accumulated over the past iterations of the search (*e.g.*, population-based Iterated Local Search [128]).

Any of the non-model-based approaches can be turned into a model-based approach. This can be accomplished in two ways: either by extracting some properties of the search landscape and reacting to the search progress made (turning the method into an adaptive search with parametric model) or by storing certain conformations and restarting the search with a new conformation found in previous iterations (turning the method into an adaptive search with a non-parametric model).

Model-based methods that build a model which does not concentrate on low energy regions result in good sampling but often inefficient search methods (*e.g.*, MUCA). In our work, we introduce an Ant Colony Optimization implementation which is a probabilistic parametric model-based search method (see Chapter 4) and a novel bin framework, which represents a non-parametric model-based search method (see Chapter 5). Our proposed approaches to protein folding are based on the observation that model-based searches seem to perform better for *ab initio* protein folding problems, for example, PERM outperforms most methods in Hydrophobic-Polar 2D square and 3D cubic lattice models, and MBS outperforms ROSETTA in some cases for off-lattice protein folding. This is most likely due to the complexity of the energy landscapes encountered, since for other combinatorial problems such as the Traveling Salesman Problem (TSP), the Graph Coloring Problem (GCP), and the Propositional Satisfiability Problem (SAT), many non-model-based searches perform well [59].

Now let us direct our attention to the height of the barriers that can be surmounted by the search. This is an important property of search methods dealing with rugged landscapes. In general, there are a limited number of ways of increasing the height of the barrier that can be surmounted effectively by a given search method. The increase in barrier height is required in order to prevent *quasi-ergodicity*, and to make a search procedure more efficient in terms of being able to effectively escape from local optima. This achieved in one of the following ways:

1. Change $E(x)$ (for non-construction-based search). Specific examples of this strategy used for protein folding include allowing but penalizing infeasible conformations (soft-core potentials) [24]; scaling energy function $E(x)$ [145], or approximating it using other simplified functions [140]; adding additional biasing potential (the generally accepted name for this approach is Umbrella Sampling [94]). The addition of the bias can be time or search progress dependent and deformation can be short-lived, as in Energy Landscape Paving [53].

2. Change acceptance criteria (transition between samples for non-construction-based search), *e.g.*, Multicanonical Monte Carlo [94], Broad Histogram

Method [30], Tabu Search (based on number of times visited) [73]. Reactive search strategies such as Energy Landscape Paving [53], which performs local elevation (modification of the acceptance criterion) based on the amount of time spent in the well also can be classified under this category. Also, modification of the Boltzmann acceptance criterion directly relates to raising the temperature (this strategy is the most commonly explored in the literature, particularly in Simulated Annealing [54], Replica Exchange Monte Carlo [94], and Simulated Tempering [94]).

3. Increase (adapt) the neighbourhood size considered, via local moves (1−, 2−, 3−, 4− residue moves, addition of one residue during construction); non-local moves (macro-mutation – a mutation of multiple solution components at once, reptation – a snake-like motion of the polymer happening by diffusion of stored length along its own contour, GA cross-over, construction starting from partial conformations, adding multiple residues). Multi-Scale Modeling approaches also belong into this category [45]. An example of indirect neighbourhood change is an exchange of conformations at different temperatures in the Replica Exchange Monte Carlo method [94].

4. Specify the region of interest from which samples are generated, *e.g.*, Importance Sampling and Sequential Importance Sampling (construction-based search methods) [60]. An example of a non-construction search method in this category is Model-based Search (MBS) [20].

The search methods proposed in this work are closely related to the last category, which has not been widely studied in the literature. We also use ideas from the other categories described above.

An important quality of the search method is how it balances intensification of the search against diversification.

Intensification can be performed in the following ways:

1. enrichment – direct (copying) or indirect (reinforcement) of parts for both construction and non-construction searches

2. replacement of unfit parts (component solutions that have low objective function values)

3. decreasing neighbourhood size

4. making acception criteria more stringent – lowering temperature

Diversification can be achieved by:

1. random construction (random restart of the search)
2. random mutation of a complete candidate solution that is accepted
3. random walk – accept a number of random mutations
4. increasing neighbourhood size
5. relaxing acceptance criteria – raising temperature, or by adding a search-progress dependent factor

To guarantee efficient searching during the search process, one may want to vary the amount of intensification and diversification based on the search progress or landscape features encountered during the search. Currently, very few adaptive strategies have been tested for the protein folding problem. Our work described in Chapter 5 makes contributions to this area.

Apart from choosing an efficient algorithm, it is also important to choose an efficient set of moves used by the algorithm (as defined by the neighbourhood relations used). Moves employed by search algorithms include mutation moves in Monte Carlo, mutations and crossovers in Genetic Algorithms.

Neighbourhood relations describe a set of conformations that can be reached from a given solution. The size of the neighbourhood is important, as is the diversity between the original candidate solution and the neighbours that can be reached in a single step. For example, it has been shown that certain local moves work better than others. Elofsson *et al.*introduced local moves for protein folding that disrupt only the local region of a protein chain while keeping the rest of the protein fixed [44]. After each change, the dihedral angles at the boundary of the region were changed to compensate for the local change that took place. They showed that MC and GA methods using these local moves perform better than mutations that change the dihedral angles of a protein, and therefore propagate the change along the chain. Similarly, in the case of lattice models, so-called pull moves have been proposed with the same objective [73].

In our work, described in the next chapter, we have designed long-range moves for 2D and 3D HP models to guarantee a more efficient search of the energy landscape.

3.1.3 An Overview of Existing Research in 2D and 3D Hydrophobic Polar Folding

We now direct our attention towards models that were chosen to test newly proposed methods. A number of well-known heuristic optimization methods have

been applied to the 2D and 3D HP Protein Folding Problem, including Evolutionary Algorithms (EAs) [68, 69, 102, 134, 135] and Monte Carlo (MC) algorithms [8, 24, 60, 77, 96, 110, 115]. The latter have been found to be particularly robust and effective for finding high-quality solutions to the HP Protein Folding Problem [60]. Unger and Moult [134, 135] presented an early application of Evolutionary Algorithms to protein structure prediction. Their non-standard EA incorporates characteristics of the Monte Carlo methods such as reliance on the Boltzmann probability for determining the acceptance ratio. Currently among the best known algorithms for the HP Protein Folding problem are various Monte Carlo algorithms, including the 'pruned-enriched Rosenbluth method' (PERM) of Grassberger *et al.* [8, 60]. As mentioned previously, PERM is a biased chain growth algorithm that evaluates partial conformations and employs pruning and enrichment strategies to explore promising partial solutions.

Liu *et al.* [152] introduced a biased sampling algorithm that is very similar to PERM except that statistical weights for pruning and enrichment are calculated based on the simulation of a number of chains in parallel. Their algorithm seems to be inferior to PERM in terms of performance.

Other Monte Carlo methods (less successful than PERM) developed to address this problem include the dynamic Monte Carlo algorithm by Ramakrishnan *et al.* [110] based on a *four-cycle* change move that disconnects the chain. Liang *et al.* [77] introduced an evolutionary Monte Carlo (EMC) algorithm that works with a population of individuals, where each individual performs Monte Carlo (MC) optimization. They also implemented a variant of EMC that reinforces certain secondary structures (alpha helices and beta sheets). Chikenji *et al.* introduced the Multi-Self-Overlap Ensemble (MSOE) Monte Carlo method [24], which considers overlapping chain configurations.

Besides general optimization methods, there are other heuristic methods that rely on specific heuristics based on intuitions or assumptions about the folding process. These include methods that consider the co-operativity of folding or the existence of a hydrophobic core. Co-operativity is believed to arise from local conformational choices that result in a globally optimal state without an exhaustive search [33]. Among these methods are the hydrophobic zipper method (HZ) [33], the contact interactions method (CI) [132], the core-directed chain growth method (CG) [13], and the constraint-based hydrophobic core construction method (CHCC) [150].

The hydrophobic zipper (HZ) strategy developed by Dill *et al.* is based on the hypothesis that once a hydrophobic contact is formed it cannot be broken, and that other contacts are formed in accordance with previously folded parts of the chain (co-operativity of folding) [33]. The contact interactions (CI) algorithm developed by Toma and Toma [132] combines the idea of HZ with a Monte Carlo search pro-

cedure that assigns different conformational freedom to the different residues in the chain. Thus, this allows previously formed contacts to be modified according to their computed mobilities. The core-directed chain growth method (CG) devised by Beutler and Dill [13] biases construction towards finding a good hydrophobic core. This is achieved by using a specifically designed heuristic function and by approximating the hydrophobic core with a square (in 2D) or a cube (in 3D); the result is a restrictive heuristic that finds only certain native states [13]. The constraint-based hydrophobic core construction method (CHCC) by Yue and Dill [150] is complete, *i.e.*, always guaranteed to find a global optimum. It attempts to find the hydrophobic core with the minimal possible surface area by systematically introducing geometric constraints and by pruning branches of a conformational search tree. A similar but more efficient complete constraint satisfaction search method has been proposed by Backofen *et al.* [5] for the more complex face-centered cubic lattice.

Other Monte Carlo methods that have been particularly useful in off-lattice protein folding, as mentioned in Section 3.1.1, but have not been tested and compared on the set of standard benchmarks include generalized ensemble methods, such as Umbrella Sampling [133] (with Replica Exchange Sampling [52, 86] being the most common variant) and Multicanonical Monte Carlo sampling [10, 86]. Replica Exchange Monte Carlo (Parallel Tempering) has also been applied to the off-lattice HP model [62].

Currently, when applied to the square and cubic lattice HP model, none of these algorithms appears to completely dominate the others in terms of solution quality and run-time. Results of comparing the algorithms mentioned above with our proposed algorithm on a set of well-studied instances are provided in Chapter 4.

3.1.4 An Overview of Existing Research in FCC β-Sheet Protein Folding

The following algorithms have been implemented and tested for the FCC lattice of β-sheet proteins: classical Metropolis Monte Carlo [52], Multicanonical (MUCA) Monte Carlo (or Entropy Sampling Monte Carlo) [52], Replica Exchange Monte Carlo (REMC) [52], and Parallel-hat Tempering Monte Carlo (a variant of REMC that utilizes an additional weight factor based on the histogram of energies sampled by each temperature) [154].

MC, MUCA and REMC have been described previously in Section 3.1.1. It is worth noting the differences between REMC and Parallel-hat Tempering [154], given that Parallel-hat Tempering is the best-performing search method for this model among all the other methods tested in the literature. The latter is an extension of the Replica Exchange method that results in more efficient search by the

introduction of a new hat-like weight factor to each replica. This weight factor results in both low- and high-energy acceptance probabilities being exponentially reinforced, which allows the algorithm to overcome higher barriers and explore a wider range of energies for each replica. The weight factor chosen was:

$$w(E) = exp(-E/k_BT + \sqrt{2}|E - \langle E\rangle|/\sigma), \tag{3.17}$$

where $\langle E\rangle$ is the average energy of the system at temperature T and σ is the root mean square deviation of the energy. Both are updated iteratively, in the m^{th} step of the MC:

$$\langle E\rangle = \frac{\sum_{i=1}^{m-1} E_i e^{-\sqrt{2}|E_i - \langle E\rangle_i|/\sigma_i}}{\sum_{i=1}^{m-1} e^{-\sqrt{2}|E_i - \langle E\rangle_i|/\sigma_i}} \tag{3.18}$$

$$\sigma = \left[\frac{\sum_{i=1}^{m-1} E_i^2 e^{-\sqrt{2}|E_i - \langle E\rangle_i|/\sigma_i}}{\sum_{i=1}^{m-1} e^{-\sqrt{2}|E_i - \langle E\rangle_i|/\sigma_i}} - \left(\frac{\sum_{i=1}^{m-1} E_i e^{-\sqrt{2}|E_i - \langle E\rangle_i|/\sigma_i}}{\sum_{i=1}^{m-1} e^{-\sqrt{2}|E_i - \langle E\rangle_i|/\sigma_i}}\right)^2\right]^{1/2}. \tag{3.19}$$

In the initial few steps, the authors set the average energy to $\langle E\rangle = E$. Local movements used in each replica are accepted according to the following probability:

$$p_{1\to 2} = exp[-\beta\Delta E + \sqrt{2}/\sigma(|E_2 - \langle E\rangle| - |E_1 - \langle E\rangle|)]. \tag{3.20}$$

The following observations were made when the move from the state with energy E_1 to the state with energy E_2 is uphill:

1. If the energies (E_1 and E_2) are low compared with the average $\langle E\rangle$, the acceptance rate is decreased (the probability of these low and rare energies is enhanced by not transitioning to a higher energy state). From the search perspective, this helps to hold on to the lower energy values found during the search.

2. If the energies (E_1 and E_2) are high compared with the average, the acceptance rate is increased and the unusually high energies are sampled with increased probability. This mechanism has the ability to overcome larger energy barriers.

The following observations were made when the move from E_1 to E_2 is downhill:

1. If the energies (E_1 and E_2) are low compared with the average $\langle E\rangle$, the move is accepted, as in the case of the canonical simulation.

2. If the energies (E_1 and E_2) are high compared with the average, the move may have not been accepted, even though it is downhill, since the acceptance rate is decreased.

Exchange moves between replicas at different temperatures (say, i and j) similarly follow a modified acceptance probability:

$$p_{i\rightarrow j} = exp[(\beta_i - \beta_j)(E_i - E_j) + \sqrt{2}(|E_j - \langle E_i\rangle|/\sigma_i - |E_i - \langle E_i\rangle|/\sigma_i + |E_i - \langle E_j\rangle|/\sigma_j - |E_j - \langle E_j\rangle|/\sigma_j)].$$

This results in a similar hat-like modification of the exchange probabilities between replicas as described above for probabilities of moves within a single replica.

As of to-date, no comprehensive study of different search methods has been conducted in the protein folding literature for more complex off-lattice discrete models. This could be because researchers are inclined to use different energy functions in addition to using a limited set of algorithms for sampling and a different set of proteins for testing. The most comprehensive studies of search and sampling methods are limited to lattice models, particularly the HP square, and cubic lattices and the FCC lattice. The HP model is not very realistic for the study of real proteins due to a parity problem, the inability to represent protein geometry closely, and a significant distortion of the secondary structure elements. The FCC lattice, as mentioned previously, has been shown to be the most accurate lattice among elementary lattices. Additionally, the state-of-the-art searching and sampling methods (MUCA, REMC, modified REMC that uses histogram data) have been tested extensively for the FCC lattice of β-sheet proteins [52, 106, 154]. Therefore, in our own work for the development, testing, and comparison of new adaptive search methods, we have adopted the FCC model of β-sheet proteins. Our goal is to be able to compare computational results to the results described in the literature.

3.2 The Problem of Identifying Folding Pathways

In this section, we describe theoretical, experimental, and computational work available for the problem of identifying protein folding pathways. Our own work builds on theoretical and experimental data from the literature. Our goal is to avoid placing restrictive assumptions on the underlying process of folding and folding nuclei formation while still developing an efficient and reliable approach for the identification of folding pathways.

3.2.1 Theoretical and Experimental Work on Protein Folding Pathways

Experimental results from studying protein folding kinetics and thermodynamics, along with a number of computational studies for a variety of lattice models, suggested three viable mechanisms for the process of protein folding [84]:

1. **The framework (hierarchical) or diffusion-collision model** proposes that local elements of native stable secondary structure form independently of tertiary structure; they collide and adhere to form tertiary interactions. The rate-limiting step is the docking of the secondary structure elements [27].

2. **The nucleation model** suggests that proteins have a small set of interactions common to most of the conformations in the transition state ensemble [27]. This small set of interactions is called a *folding nucleus*, and can bring together distant parts of the chains (largely formed by long-range interactions with stabilizing short-range interactions). The nucleation mechanism does not require secondary structure elements to be formed before the transition state is reached. In contrast to the diffusion-collision model, secondary structure may be formed simultaneously with the folding of the tertiary structure.

3. **The hydrophobic collapse model** suggests that a protein rapidly collapses around its hydrophobic side-chains and then rearranges itself from the restricted conformational space. In this model, secondary structure is directed by tertiary interactions [27]. This model is plausible for small proteins.

One way of studying and testing the nucleation mechanism and diffusion-collision model, stems from the fact that they predict different effects of mutations on the folding rate. The diffusion-collision model predicts that the stabilization of any local secondary structure element (an α-helix or a β-strand) will always lead to an acceleration of folding. In contrast, the nucleation model predicts that the strength of tertiary contacts formed in the transition state is the primary determinant of folding rate. According to the nucleation model, stabilization of local structure accelerates folding rates only if a particular element is present in the transition state ensemble. Through directed site mutagenesis on real proteins it has been observed that the nucleation model seems to be correct for small and medium size proteins [84]. However, for certain proteins in simulations the diffusion-collision model seemed to play the role [155].

Another important partly computational study that provided insight into the protein folding process was conducted by Plaxco *et al.* [104]. They suggested that the average separation between residues interacting in the native state (*contact*

order) can be used as a general descriptor of protein topology. The contact order of a given conformation is defined as:

$$CO = \frac{1}{N \cdot C} \sum_{i,j} \delta_{ij} |i - j| \tag{3.21}$$

where $\delta_{ij} = 1$ if residues i and j are in contact and 0 otherwise; C is the total number of contacts, and N is the protein length. For a number of two-state folding proteins, contact order was reported to exhibit a statistically significant correlation with the logarithm of the folding rate in an aqueous environment. Proteins that had a higher contact order (had more local contacts) folded faster. Dinner and Karplus later showed that apart from contact order, stability of the native state (to a much smaller degree) determines the folding rate as well [34]. The precise nature of the folding nucleus (the ratio and distribution of long- and short-range interactions, its size, and other topological characteristics) or of multiple folding nuclei for longer proteins still has to be determined.

Experimental methods for identifying folding nuclei include site-directed mutagenesis (Φ-value analysis), which affects the overall folding rate and protein stability [46], and identifying folding cores using hydrogen-deuterium exchange (H/X) experiments [76]. Both experimental techniques are time- and labour-intensive, and can therefore benefit from additional computational analysis.

3.2.2 An Overview of Computational Approaches for Identifying Folding Nuclei from the Native Conformation

A number of computational methods for the prediction of folding nuclei already exist in the literature, but most of them rely on restrictive assumptions about the nature of nuclei or the process of folding. Current approaches in the literature to identifying folding nuclei include time-intensive molecular dynamic unfolding of the native structures [36], a Gaussian network model (GNM) based on an analysis of the highest frequency fluctuations near the native state [32], an elastic network and constraint network models of freely rotating rods (FIRST) [107], clustering of native contacts and the use of low effective contact order search [144], a motion planning approach (the probabilistic roadmap method) for prediction of folding pathways [2], minimum cuts unfolding of native structures [151], amino acid conservation within a family or super-family [112], search for free energy saddle points on networks of protein unfolding pathways [49], and determination of the transition state by restraining Monte Carlo simulations with a pseudo-energy function based on the set of experimental Φ-values [97].

Most of these methods are based on a number of restrictive assumptions about the folding nuclei or the search process that may not necessarily hold. For exam-

ple, these assumptions include an assumption that folding nuclei are evolutionarily conserved [112], that residues participating in folding nuclei are those that are most physically constrained [107], or that low effective contact order clusters of contacts can not be partially formed [144]. The goal of our work, presented in Chapter 6 of this thesis, is to develop a simple and efficient method of identifying folding nuclei that does not rely on restrictive assumptions about the nature of folding nuclei.

Chapter 4

An Ant Colony Optimization for 2D and 3D Hydrophobic Polar Protein Folding

Nature uses only the longest threads to weave her patterns, so each small piece of her fabric reveals the organization of the entire tapestry.

Richard Feynman

Ant Colony Optimization (ACO) is a population-based stochastic search method for solving a wide range of combinatorial optimization problems. ACO is based on the concept of *stigmergy* – indirect communication between members of a population through interaction with the environment. An example of stigmergy is the communication of ants during the foraging process: ants indirectly communicate with each other by depositing pheromone trails on the ground and thereby influencing the decision processes of other ants [14]. This simple form of communication between individual ants gives rise to complex behaviors and capabilities of the colony as a whole. In real life, ants follow higher concentrations of *pheromone* and find the shortest path to a food source. Our goal is to develop an algorithm that captures the emergence of higher-order intelligence from low-level communication of a population of agents, similar to the functioning of an ant colony.

From a computational point of view, ACO is an iterative construction search method in which a population of simple agents ('ants') repeatedly constructs candidate solutions to a given problem. This construction process is probabilistically guided by heuristic information on the given problem instance as well as by a shared memory of experiences gathered by the ants in previous iterations ('pheromone trails').

Following the seminal work by Dorigo *et al.* [40, 41], ACO algorithms have been successfully applied to a broad range of hard combinatorial problems: the Traveling Salesman Problem (TSP), the Quadratic Assignment Problem (QAP),

the Job-shop Scheduling Problem (JSP), the Graph Coloring Problem (GCP), the Vehicle Routing Problem (VRP), and others (*e.g.*, [37, 38, 42]).

In this work, we address the *ab initio* protein folding problem. It can be formally defined as follows: Given an amino acid sequence $s = s_1 s_2 \dots s_n$ and an energy function $E(c)$, find an energy-minimizing conformation of s, *i. e.*, find $c^* \in C(s)$ such that $E(c^*) = \min\{E(c) \mid c \in C(s)\}$, where $C(s)$ is the set of all valid conformations for s.

ACO, which has been very successfully applied to other combinatorial problems [14], appears to be a very attractive computational method for solving the protein folding problem, since it combines aspects of chain growth and permutation-based search with ideas closely related to reinforcement learning. These concepts and ideas apply rather naturally to protein folding: by folding from multiple initial folding points, guided by the energy function and experience from previous iterations of the algorithm, an ensemble of promising, low-energy complete conformations is obtained. These conformations are further improved by a subsidiary local search procedure and then evaluated to update the accumulated pheromone values that are used to bias the generation of conformations in future iterations of the algorithm.

In this chapter, we introduce an ACO algorithm for the 2D and the 3D HP protein folding problem, and discuss the promising results obtained [120, 121]. This is the first application of the ACO algorithm to this highly relevant problem[1].

4.1 Description of the Algorithm

The 'ants' in our ACO algorithm iteratively undergo three phases: (1) the *construction phase*, during which each ant constructs a candidate solution by sequentially growing a conformation of the given HP sequence, starting from a folding point that is chosen uniformly at random among all sequence positions; (2) the *local search phase*, when ants further optimize protein conformations folded during the construction phase; and (3) the *pheromone update phase*, when ants update the pheromone matrix (representing the collective global memory of the colony) based on the energies of the conformations obtained after the construction and local search phases. A general outline of ACO is shown in Figure 4.1.

The solution components used during the construction process, the local search phase, and the pheromone update consists of the following local structural motifs (or relative folding directions): *straight* (S), *left* (L), *right* (R) in 2D, and *straight*

[1] A version of this chapter has been published. A. Shmygelska and H.H. Hoos, (2005) An Ant Colony Optimisation Algorithm for the 2D and 3D Hydrophobic Polar Protein Folding Problem, BMC Bioinformatics, 6:30.

```
procedure ACO
    initialize pheromone trails;
    while (termination condition not satisfied) do
        construct candidate conformations;
        perform local search;
        update pheromone values;
    end
end
```

Figure 4.1: Generic outline of Ant Colony Optimization (for static combinatorial problems).

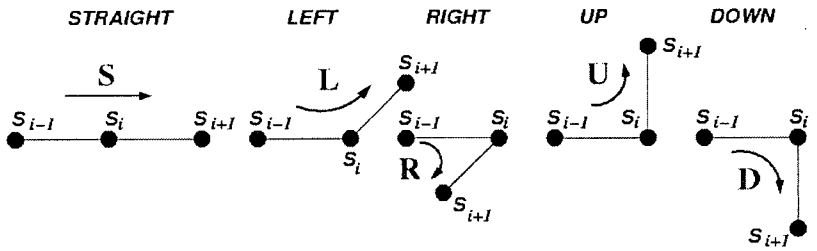


Figure 4.2: The local structural motifs that form the solution components underlying the construction and local search phases of our ACO algorithm in 3D.

(S), *left* (L), *right* (R), *up* (U), *down* (D) in 3D. These relative directions indicate the position of each amino acid on the 2D or 3D lattice relative to its direct predecessors in the given sequence (see Figure 4.2). In 3D, the relative folding directions are defined as in [6]: A local coordinate system is associated with every sequence position, such that S corresponds to the direction of the x axis, L to the direction of the y axis, and U to the direction of the z axis. Each local motif corresponds to a relative rotation of this coordinate system (for the forward construction, S = no rotation, L = 90° counter-clockwise around the z axis, R = 90° clockwise around the z axis, U = 90° clockwise around the y axis, D = 90° counter-clockwise around the y axis).

Since conformations are rotationally invariant, the position of the first two amino acids can be fixed without loss of generality. Hence, we represent candidate conformations for a protein sequence of length n by a sequence of local structural motifs of length $n - 2$. For example, the conformation of Sequence S1-1 from Table 4.1 shown in Figure 4.3 corresponds to the motif sequence LSLLRRLRLL-SLRRLLSL.

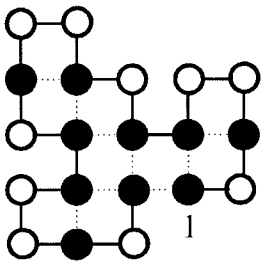


Figure 4.3: The underlying protein sequence (Sequence S1-1 from Table 4.1) is HPHPPHHPHPPHPHHPPHPH; black circles represent hydrophobic amino acids, while white circles symbolize polar amino acids. The dotted lines represent the H-H contacts underlying the energy calculation. The energy of this conformation is -9, which is optimal for the given sequence.

During the construction phase, ants fold a protein from an initial folding point by probabilistically adding one amino acid at a time based on the two following sources of information: pheromone matrix values τ (which represent previous search experience and reinforce certain structural motifs) and heuristic function values η (which reflect current energy of the considered structural motif); details of this process are provided in the next section. The relative importance of τ and η is determined by the parameters α and β, respectively.

Similar to other ACO algorithms known from the literature, our algorithm for the HP Protein Folding Problem incorporates a local search phase that takes the initially built protein conformation and attempts to optimize its energy further, using probabilistic long-range moves that are described in detail in the next chapter.

Finally, the pheromone update procedure is based on two mechanisms: uniform pheromone evaporation is modeled by decreasing all pheromone levels by a constant factor ρ (where $0 < \rho \leq 1$), and pheromone reinforcement is achieved by increasing the pheromone levels associated with the local folding motifs used in a fraction of the best conformations (in terms of energy values) obtained during the preceding construction and local search phase. Furthermore, to prevent search stagnation when all of the pheromone is accumulated on very few structural motifs, we introduce an additional renormalization mechanism for the pheromone levels (controlled by a parameter θ, where $0 \leq \theta < 1$; details are given below).

Our ACO algorithm iterates construction, local search, and pheromone update phases until a termination condition is satisfied; in the context of this work, we mostly terminated the algorithm upon reaching a given energy threshold. In the following sections, we describe the three search phases in detail.

4.1.1 Construction Phase, Pheromone, and Heuristic Values

During the construction phase of ACO, each ant first determines a starting point within the given protein sequence; this is done by uniform random choice. From this starting point, the sequence is folded in both directions, adding one residue at a time. Each ant performs probabilistic chain-growth construction of the protein conformation, by which the structure is extended either to the left or to the right in every step, such that the ratio of unfolded residues at each end of the protein remains (roughly) unchanged.

Here, we assume that folding is performed in 3D; the 2D case is handled analogously by considering three relative directions $\{S, L, R\}$ instead of five $\{S, L, R, U, D\}$ (see also [119]). The relative directions in which the conformation is extended in each construction step are determined probabilistically based on a heuristic function $\eta_{i,d}$ and pheromone values $\tau_{i,d}$, according to the following formula:

$$p_{i,d} := \frac{[\tau_{i,d}]^{\alpha}[\eta_{i,d}]^{\beta}}{\sum_{e\in\{S,L,R,U,D\}}[\tau_{i,e}]^{\alpha}[\eta_{i,e}]^{\beta}} \tag{4.1}$$

The pheromone values $\tau_{i,d}$ indicate the desirability of using the local structural motif with relative direction $d \in \{S, L, R, U, D\}$ at sequence position i. Initially, all $\tau_{i,d}$ are equal, such that local structural motifs are chosen in an unbiased way. Throughout the search process, however, the pheromone values are updated to bias folding towards the use of local motifs that occur in low-energy structures. The updating mechanism will be described in more detail later.

The heuristic values $\eta_{i,d}$ are based on the energy function E. They are defined according to the Boltzmann distribution as $\eta_{i,d} := e^{-\gamma \cdot h_{i,d}}$, where γ is a parameter called the inverse temperature, as in [60], and $h_{i,d}$ is the number of new H-H contacts achieved by placing amino acid i at the position specified by direction d.

During construction, it may happen that the chain cannot be extended without running into itself. This situation is called *attrition*, which our algorithm overcomes as follows. First, starting at the end at which attrition occurred, half of the sequence that has been folded up to this point is unfolded. Then, this segment of the chain is refolded; the first residue (*i.e.*, the last one that was unfolded) is placed such that its relative direction differs from what it had been when attrition occurred, while all of the subsequent residues are folded in a feasible direction that is chosen uniformly at random. This backtracking mechanism is particularly important for longer protein sequences in 2D, where infeasible conformations are frequently encountered during the construction phase.

4.1.2 Local Search

Similar to other ACO algorithms known from the literature, our algorithm for the HP Protein Folding Problem incorporates a local search phase, based on a long-range mutation move that has been designed to avoid infeasible conformations. It also has a number of important advantages over the more commonly used point mutation moves or Monte Carlo moves (*i.e.*, the end, crankshaft, and corner moves [114]). It is easy to implement; it decreases the number of infeasible conformations encountered, even when the protein is very compact (at high densities); it considers a larger neighbourhood that subsumes the single-point mutation neighbourhood; and it has some validity in terms of the physical processes taking place during the protein folding process. Similar attempts have been previously undertaken, but these involved disconnection of the chain [110].

From studies of protein folding dynamics, it is known that proteins display a broad range of motions that range from localized motions to slow large-scale movements [27]. Inspired by this complex process, we designed a long-range mutation move that starts by selecting a residue whose relative direction is randomly mutated and then adapts the rest of the chain by probabilistically changing relative directions starting from this initial position [121]. During this adaptation, the previous relative direction for each residue with a probability $\hat{p}$ ($0 \leq \hat{p} \leq 1$) is left unchanged, if it is still feasible. Otherwise (*i.e.*, with probability $1 - \hat{p}$, or if the previous direction has become infeasible), a different relative direction is chosen, where the probability for each direction d is proportional to the corresponding heuristic value $\eta_{i,d}$. Formally, this can be written as follows:

$$P[d_i := \hat{d}] := \begin{cases} \hat{p} & \text{if } \hat{d} = d_{prev} \text{ and} \\ & \text{feasible}(s, d_{prev}) \\ \dfrac{\eta_{i,\hat{d}}}{\sum_{e \in \{S,L,R,U,D\}} \eta_{i,e}} & \text{if } \hat{d} \neq d_{prev} \text{ or} \\ & \text{infeasible}(s, d_{prev}), \end{cases} \tag{4.2}$$

where $P[d_i := \hat{d}]$ is the probability of choosing direction $\hat{d}$ as the relative direction d_i at sequence position i.

In our initial implementation, we investigated using the following two types of local search: greedy (Iterative Improvement) and non-greedy (Probabilistic Iterative Improvement). Consequently, we had two types of ants in the colony: *forager* ants that use greedy local search while constructing solutions using the pheromone matrix, and *improving* ants that exploit the best solution found so far by using non-greedy local search. The number of *improving* ants compared with *forager* ants was much smaller. Both local searches employed a long range mutational move designed by us to decrease the number of infeasible configurations considered, especially when a protein is quite compact.

Unlike in our initial implementation [121], the local search phase of the latest version of our ACO algorithm (which has been shown to work better for both the 2D and the 3D HP) is a simple iterative first improvement procedure that is based on the long-range mutation move. The outline of this local search procedure is shown in Figure 4.4. Iterative first improvement accepts a new conformation generated via long-range mutation only if the solution quality of a new conformation c' improves over the current solution quality (energy) of c. This search process is greedy in the sense that it does not allow worsening steps, and it is terminated when no improving steps have been found after a specific number of scans through the chain (this number is a parameter of the algorithm).

```
procedure IterativeImprovementLS(c)
   input: candidate conformation c
   output: candidate conformation c'
   while (termination condition not satisfied) do
      i := random({1, ..., n});
      c' := longRangeMove(c, i);
      if E(c') ≤ E(c) then
         c := c';
      end
   end
   return(c)
end
```

Figure 4.4: The iterative first improvement local search procedure that is performed by selected ants after the construction phase.

Since this local search procedure has a relatively high time-complexity, in each iteration of ACO it is only applied to a certain fraction of the highest-quality conformations constructed by the ants in the preceding construction phase.

4.1.3 Update of the Pheromone Values

After each construction and local search phase, pheromones are updated according to

$$\tau_{i,d} := \rho \cdot \tau_{i,d}, \tag{4.3}$$

where $0 < \rho \leq 1$ is the pheromone persistence, a parameter that determines how much of the information gathered in previous iterations is retained. Subsequently, selected ants with low-energy conformations update the pheromone values according to

$$\tau_{i,d} := \tau_{i,d} + \Delta_{i,d,c}, \tag{4.4}$$

where $\Delta_{i,d,c}$ is the relative solution quality of the given ant's candidate conformation c, if that conformation contains local structural motif d at sequence position i, and zero otherwise. We use the relative solution quality, $E(c)/E^*$, where E^* is the known minimal energy for the given protein sequence or an approximation based on the number of H residues in the sequence, in order to prevent premature search stagnation for sequences with large energy values.

As a further mechanism for preventing search stagnation, we use an additional renormalization of the pheromone values that is conceptually similar to the method used in the $\mathcal{MAX}$-$\mathcal{MIN}$ Ant System [129]. After the standard pheromone updates according to Equations 3 and 4, all τ values are normalized such that $\sum_{d\in\{S,L,R,U,D\}} \tau_{i,d} = 1$ for every residue i; additionally, whenever for a given sequence position i the minimal normalized pheromone value

$$\min_{d\in\{S,L,R,U,D\}} \tau_{i,d} / \sum_{d\in\{S,L,R,U,D\}} \tau_{i,d}$$

falls below a threshold θ (which is a parameter of the algorithm), the minimal $\tau_{i,d}$ value is set to θ, while the maximal $\tau_{i,d}$ value is decreased by $\theta - \min_{d\in\{S,L,R,U,D\}} \tau_{i,d}$. (If there is more than one minimal $\tau_{i,d}$ value, all of these are increased to θ, and if there is more than one maximal $\tau_{i,d}$ value, one of them is chosen uniformly at random.) This guarantees that the probability of selecting an arbitrary local structural motif for the corresponding sequence position does not become arbitrarily small, and hence ensures the probabilistic approximate completeness of our algorithm (see [59]).

4.2 Empirical Results and Discussion

In the following work, we address the following questions: is ACO a competitive method for solving the *ab initio* protein folding problem under the 2D and 3D HP models? How does its performance scale with sequence length? What is the role of the parameters of the ACO algorithm for the efficiency of the optimization process? Which classes of structures (if any) are solved more efficiently by ACO than by any other known algorithms? Finally, it should be noted that our ACO algorithm for this problem is based on very simple design choices, in particular with respect to both the solution components reinforced in the pheromone matrix and the subsidiary local search procedure. We discuss which of the many design choices underlying our algorithm should be reconsidered in order to achieve further performance improvements.

To compare ACO with algorithms for the 2D and 3D HP Protein Folding Problem described in the literature, we tested it on a number of standard benchmark

instances as well as on two newly created data sets. We obtained one of these sets by randomly generating amino acid sequences with hydrophobicity value characteristic of globular proteins, while the other consists of biological sequences that we translated into HP strings using a standard hydrophobicity scale. These new data sets will be described in more detail later in this section.

4.2.1 Results for Standard Benchmark Instances

The 21 standard benchmark instances for 2D- and 3D-HP protein folding shown in Table 4.1 have been widely used in the literature [8, 13, 24, 68, 77, 119, 134, 135]. Experiments on these standard benchmark instances were conducted by performing a number of independent runs for each problem instance (in 2D: 500 runs for sequence length $n \leq 50$, 100 runs for $50 < n \leq 64$, and 20 runs for $n > 64$; in 3D: 100 runs for each sequence). Unless explicitly indicated otherwise, we used the following parameter settings for all experiments: $\alpha := 1$, $\beta := 2$, $\rho := 0.8$ and $\theta := 0.05$. Furthermore, all pheromone values were initialized to $1/3$ in 2D and to $1/5$ in 3D, and a population of 100 ants was used, 50% of which were allowed to perform local search. The local search procedure was terminated when no improvement in energy had been obtained after between 1 000 (for $n \leq 50$) and 10 000 (for $n > 50$) scans through the protein sequence. We used an elitist pheromone updating scheme in which only the best 1% of all ants was allowed to perform pheromone updates. The probability $\hat{p}$ of keeping the previous direction when feasible during the long-range mutation move was set to 0.5. These settings were determined in a series of experiments in which we studied the influence of different parameter settings, and will be discussed further later. All experiments were performed on PCs with 2.4 GHz Pentium IV CPUs, 256Kb cache and 1Mb RAM, running Redhat Linux (our reference machine). Run-time was measured in terms of CPU time.

Most studies of EA and MC methods in the literature, including [68, 77, 134, 135], report the number of valid conformations scanned during the search. This makes a performance comparison difficult, since run-time spent for backtracking and the checking of partial or infeasible conformations, which may vary substantially between different algorithms, is not accounted for. We therefore compared ACO to the best-performing algorithm from the literature for which performance data in terms of CPU time is available – PERM [60] (we used the most recent implementation, which was kindly provided by P. Grassberger). We note that the most efficient PERM variant for the HP Protein Folding Problem uses an additional penalty of 0.2 for H-P contacts [61]. Since this corresponds to an energy function different from that of the standard HP model underlying our ACO algorithm as well as other algorithms developed in the literature, we used the best performing variant

Seq. No.	Length	E^*	Protein Sequence
			2D HP
1	20	-9	$(HP)_2PH_2PHP_2HPH_2P_2HPH$
2	24	-9	$H_2(P_2H)_7H$
3	25	-8	$P_2HP_2(H_2P_4)_3H_2$
4	36	-14	$P_3H_2P_2H_2P_5H_7P_2H_2P_4H_2P_2HP_2$
5	48	-23	$P_2H(P_2H_2)_2P_5H_{10}P_6(H_2P_2)_2HP_2H_5$
6	50	-21	$H_2(PH)_3PH_4PH(P_3H)_2P_4H(P_3H)_2PH_4(PH)_3PH_2$
7	60	-36	$P_2H_3PH_8P_3H_{10}PHP_3H_{12}P_4H_6PH_2PHP$
8	64	-42	$H_{12}(PH)_2(P_2H_2)_2P_2H(P_2H_2)_2P_2H(P_2H_2)_2P_2HPHPH_{12}$
9	85	-53	$H_4P_4H_{12}P_6(H_{12}P_3)_3HP_2(H_2P_2)_2HPH$
10	100	-50	$P_3H_2P_2H_4P_2H_3(PH_2)_2PH_4P_8H_6P_2H_6P_9HPH_2PH_{11}P_2H_3PH_2PHP_2HPH_3P_6H_3$
11	100	-48	$P_6HPH_2P_5H_3PH_5PH_2P_4H_2P_2H_2PH_5PH_{10}PH_2PH_7P_{11}H_7P_2HPH_3P_6HPH_2$
			3D HP
1	48	-32	$HPH_2P_2H_4PH_3P_2H_2P_2HPH_3PHPH_2P_2H_2P_3HP_8H_2$
2	48	-34	$H_4PH_2PH_5P_2HP_2H_2P_2HP_6HP_2HP_3HP_2H_2P_2H_3PH$
3	48	-34	$PHPH_2PH_6P_2HPHP_2HPH_2(PH)_2P_3H(P_2H_2)_2P_2H_PHP_2HP$
4	48	-33	$PHPH_2P_2HPH_3P_2H_2PH_2P_3H_5P_2HPH_2(PH)_2P_4HP_2(HP)_2$
5	48	-32	$P_2HP_3HPH_4P_2H_4PH_2PH_3P_2(HP)_2HP_2HP_6H_2PH_2PH$
6	48	-32	$H_3P_3H_2PH(PH_2)_3PHP_7HPHP_2HP_3HP_2H_6PH$
7	48	-32	$PHP_4HPH_3PHPH_4PH_2PH_2P_3HPHP_3H_3(P_2H_2)_2P_3H$
8	48	-31	$PH_2PH_3PH_4P_2H_3P_6HPH_2P_2H_2PHP_3H_2(PH)_2PH_2P_3$
9	48	-34	$(PH)_2P_4(HP)_2HP_2HPH_6P_2H_3PHP_2HPH_2P_2HPH_3P_4H$
10	48	-33	$PH_2P_6H_2P_3H_3PHP_2HPH_2(P_2H)_2P_2H_2P_2H_7P_2H_2$

Table 4.1: Benchmark instances for the 2D and 3D HP Protein Folding Problem used in this study with optimal or best known energy values E^*. Most instances for 2D and 3D HP can also be found at http://www.cs.sandia.gov web site; Sequence S1-9 (2D) is taken from [67], and the last two instances (2D) are from [110]. H_i and P_i indicate a string of i consecutive H's and P's, respectively; likewise, $(s)_i$ indicates an i-fold repetition of string s.

of PERM [60] based on the standard energy function in our experiments. It may be noted that the chain growth process in PERM can start from the N- or C-terminus of the given HP sequence; in many cases, this results in substantial differences in the performance of the algorithm. To capture this effect, we always ran PERM in both directions. In addition to the respective average run-times, t_1 and t_2, we report the expected time for solving a given problem instance when performing both runs concurrently, $t_{exp} = 2 \cdot (1/t_1 + 1/t_2)^{-1}$. For all runs of PERM, the following parameter settings were used: inverse temperature $\gamma := 26$ and $q := 0.2$.

The results obtained on standard 2D benchmark instances (see Table 4.2) indicate that ACO is competitive with the EA and MC methods described in the literature; it works very well on sequences of sizes up to 64 amino acids and produces high quality suboptimal configurations for the longest sequences considered here (85 and 100 amino acids). On average, ACO requires less CPU time than PERM for finding best-known conformations for Sequence S1-8. However, PERM performs better for Sequences S1-6 and S1-7 as well as for the longer sequences of 85 to 100 residues (Sequences S1-9 to S1-11).

Sequence S1-8 has a very symmetrical optimal state (see Figure 4.5), which, as argued in [60], would be difficult to find for any chain-growing algorithm. All algorithms from the literature that we are aware of have problems folding this sequence. ACO, on the other hand, is able to handle this instance quite well, since a number of ants folding from different starting points in conjunction with a local search procedure that involves large-scale mutations originating from different sequence positions can produce good partial folds for various parts of the chain. In comparison with other algorithms for the 2D HP Protein Folding Problem considered here (EA, EMC, MSOE), ACO generally shows very good performance on standard benchmark instances.

In the case of the 3D HP Protein Folding Problem (see Table 4.3), the majority of algorithms for which we were able to find performance results in the literature use heuristics that are highly specialized for this problem. Unlike HZ, CG, and CI, ACO finds optimal (or best-known) solution qualities for all sequences. However, PERM (when folding from the N-terminus) and CHCC consistently outperform ACO on these standard 3D HP benchmark instances, and CG reaches best-known solution qualities substantially faster in many cases. We note that for Sequences S2-3 and S2-7, PERM's performance is greatly dependent on the folding direction.

4.2.2 Result for New Biological and Random Data Sets

To thoroughly test the performance of ACO, we created two new data sets of random and biological sequences of length ≈ 30 and ≈ 50 amino acids (ten sequences for each length; for details, see Appendix A). Random sequences were generated

Seq. No.	Length	Energy	GA	EMC	MSOE	PERM	ACO
1	20	**-9**	**-9** (30 492)	**-9** (9 374)		**-9** (< 1 sec)	**-9** (< 1 sec)
2	24	**-9**	**-9** (30 491)	**-9** (6 929)		**-9** (< 1 sec)	**-9** (< 1 sec)
3	25	**-8**	**-8** (20 400)	**-8** (7 202)		**-8** (12 sec)	**-8** (< 1 sec)
4	36	**-14**	**-14** (301 339)	**-14** (12 447)		**-14** (< 1 sec)	**-14** (4 sec)
5	48	**-23**	**-23** (126 547)	**-23** (165 791)		**-23** (15 min)	**-23** (1 min)
6	50	**-21**	**-21** (592 887)	**-21** (74 613)		**-21** (2 sec)	**-21** (15 sec)
7	60	**-36**	-34 (208 781)	-35 (203 729)		**-36** (5 sec)	**-36** (20 min)
8	64	**-42**	-37 (187 393)	-39 (564 809)	-39	-40 (2 days)	**-42** (1.5 hrs)
9	85	**-53**		-52 (44 029)		**-53** (11 sec)	**-53** (20 % of runs 1 day)
10	100	**-50**			**-50** (50 hrs)	**-50** (50% of runs 2 hrs)	-49 (12 hrs)
11	100	**-48**			-47	**-48** (2 min)	-47 (10 hrs)

Table 4.2: Comparison of the solution quality obtained in 2D by the evolutionary algorithm of Unger and Moult (EA) [135], the evolutionary Monte Carlo algorithm of Liang and Wong (EMC) [77], the Multi-Self-Overlap Ensemble algorithm of Chickenji *et al.* (MSOE) [24], the pruned-enriched Rosenbluth method (PERM) and ACO. For EA and EMC, the reported energy values are the lowest among five independent runs, and the values in parentheses are the numbers of valid conformations scanned before the lowest energy values were found. Missing entries indicate cases where the respective method has not been tested on a given instance. The CPU times reported in parentheses for MSOE were determined on a 500 MHz CPU, and those for PERM and ACO are based on $100 - 200$ runs per instance on our reference 2.4 GHz Pentium IV machine. The energy values shown in bold face correspond to currently best-known solution qualities.

Seq. No.	Length	Energy	HZ	CHCC	CG	CI	PERM	ACO
1	48	**-32**	-31(15 000 min)	**-32** (30 min)	**-32** (9.4 min)	**-32**	**-32** (0.1 min)	**-32** (30 min)
2	48	**-34**	-32 (71 000 min)	**-34** (2.3 min)	**-34** (35 min)	-33	**-34** (1 min)	**-34** (420 min)
3	48	**-34**	-31 (82 000 min)	**-34** (30 min)	**-34** (62 min)	-32	**-34** (30.5 min)	**-34** (120 min)
4	48	**-33**	-30 (1 600 000 min)	**-33** (71 min)	**-33** (29 min)	-32	**-33** (2 min)	**-33** (300 min)
5	48	**-32**	-30 (110 000 min)	**-32** (32 min)	**-32** (12 min)	**-32**	**-32** (1 min)	**-32** (15 min)
6	48	**-32**	-29 (180 000 min)	**-32** (80 min)	**-32** (460 min)	-30	**-32** (1 min)	**-32** (720 min)
7	48	**-32**	-29 (220 000 min)	**-32** (110 min)	**-32** (64 min)	-30	**-32** (1 min)	**-32** (720 min)
8	48	**-31**	-29 (14 000 min)	**-31** (530 min)	**-31** (38 min)	-30	**-31** (0.3 min)	**-31** (120 min)
9	48	**-34**	-31 (16 000 min)	**-34** (8.3 min)	-33	-32	**-34** (8 min)	**-34** (450 min)
10	48	**-33**	**-33** (4 100 min)	**-33** (4.8 min)	**-33**(1.1 min)	-32	**-33** (0.15 min)	**-33** (60 min)

Table 4.3: Comparison of the solution quality obtained in 3D by the hydrophobic zipper (HZ) algorithm [33], the constraint-based hydrophobic core construction method (CHCC) [150], the core-directed chain growth algorithm (CG) [13], the contact interactions (CI) algorithm [132], the pruned-enriched Rosenbluth method (PERM) and ACO. For CI, only the best energies obtained are shown. For HZ, CHCC and CG, the reported CPU times are taken from [13]; these are the expected times for finding optimal solutions on a Sparc 1 workstation. In the case of HZ, the reported CPU times are based on an extrapolation from the measured times required for finding suboptimal conformations with the energy values listed here. The CPU times for PERM and ACO were determined on our reference 2.4 GHz Pentium IV machine based on $50 - 100$ runs per instance. The energy values shown in bold face correspond to currently best-known solution qualities.

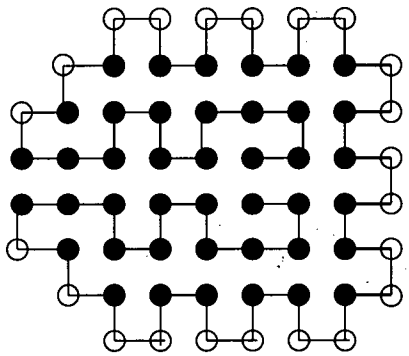


Figure 4.5: The native conformation of Sequence S1-8 from Table 4.1 (64 amino acids; energy -42), found by ACO in an average CPU time of 1.5 hours and by PERM in $t_1 = t_2 = t_{exp} = 78$ hours.

based on the observation that most globular proteins have a fairly uniform amino acid profile, and that the percentage of hydrophobic residues of the majority of globular proteins falls in the range of 40–50% [90]. Thus, we chose the probability of generating character H at each position of a sequence to be 0.45, and in the remaining cases (*i.e.*, with probability 0.55), we generated a P.

For the biological test sets, ten sequences were taken from the PDBSELECT data set with homology $< 25\%$ from the Protein Data Bank (PDB) in order to obtain a non-redundant representative set of proteins. These protein sequences were translated into HP strings using the hydrophobicity scale classification of RASMOL [116], according to which the following amino acids were considered hydrophobic: *Ala*, *Leu*, *Val*, *Ile*, *Pro*, *Phe*, *Met*, *Trp*, *Gly*, and *Tyr*. Non-standard amino acid symbols, such as X and Z, were skipped in this translation.

Figures 4.6 and 4.7 illustrate the performance of ACO *vs* PERM in terms of mean CPU time over 10 runs per instance and algorithm. For practical reasons, each run was restricted to 1 CPU hour on our reference machine, and the lowest energies obtained in these runs (listed in Appendix A) are not necessarily optimal.

As can be seen from these results, in 2D, ACO performs roughly comparably to PERM (PERM's t_{exp} was calculated as described in the previous subsection): ACO reaches the same energies as PERM, but on some instances, particularly of length 50, requires more run-time. In 3D, ACO generally requires a comparable amount of run-time on sequences of length 30 and outperforms PERM on one random sequence of length 30. It performs noticeably worse, however, on sequences of length 50 and in some cases does not reach the same energy. We also gener-

ated longer sequences of length 75; for these, ACO failed to reach the minimal energy values obtained by PERM in a number of cases. The run-times for both algorithms are reported in detail in Appendix A; we note that on some sequences, the performance of PERM depends significantly on the direction of folding. Interestingly, there is no significant difference in performance between the biological and random test-sets for either PERM or ACO.

In summary, the performance of ACO is comparable with that of PERM (the best known algorithm for the 2D and 3D HP Protein Folding Problem) on biological and random sequences of length 30–50, but worse on longer sequences. This scaling effect is significantly more pronounced in 3D than in 2D. We note that neither ACO nor PERM were optimized for short sequences ($n \leq 30$); however, by using parameter settings different from the ones specified earlier, the performance of both algorithms can be significantly improved in this case.

4.2.3 Characteristic Performance Differences between ACO and PERM

To further investigate the conditions under which ACO performs well compared to PERM, we visually examined native conformations found by both algorithms, paying special attention to conformations for which one of the two algorithms does not perform well (see Figures 4.8 and 4.10). Based on our observations, we hypothesized that PERM usually performs well on sequences that have a structural nucleus in the native conformation at one of the ends of the sequence, particularly the end from which PERM starts folding the sequence. On the other hand, PERM has trouble folding sequences whose native conformations have structural nuclei in the middle of the sequence. In comparison, ACO is not significantly affected by the location of the structural nucleus (or multiple nuclei) in the sequence, since it uses construction from different folding points as well as the long-range mutation moves in local search, which can initiate re-folding from arbitrary sequence positions. Here, we use the term 'structural nucleus' to refer to a predominantly locally folded part of the chain that can be folded sequentially relatively easily based on local sequence information [27]. For most sequences considered in this study, we observed a single structural nucleus, which is not surprising, given their relatively short length; however, it is generally believed that longer sequences have multiple folding nuclei [27].

The left side of Figure 4.8 shows an example of a relatively short biological sequence (B50-7, 45 amino acids) with a unique native hydrophobic core in the 2D HP model. (This is rare for HP sequences, which usually have a high ground state and hydrophobic core degeneracy. According to our observations, of the 11 standard benchmark instances in 2D, only Sequences S1-1, S1-3, and S1-4 have a

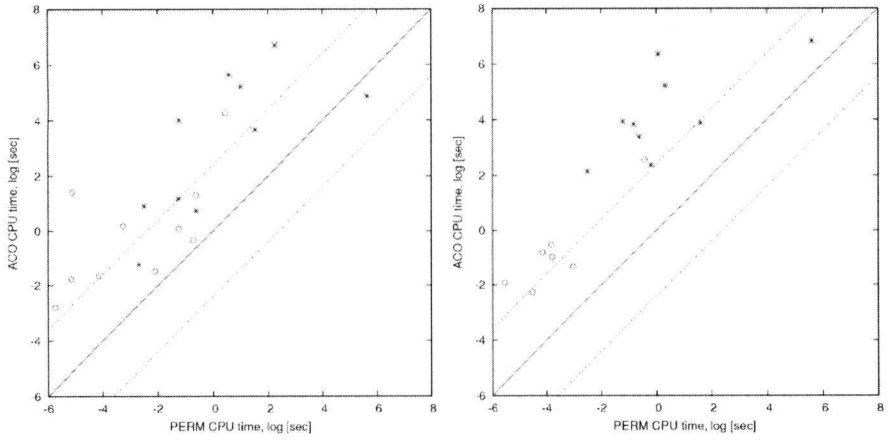


Figure 4.6: Mean CPU time (natural log transformed) required by ACO *vs* PERM for reaching the best solution quality, as observed over 10 runs with a cut-off time of 1 CPU hour for sequences of length 30 and 50 in 2D. The left and right plots show the results for the biological and random test-sets, respectively. Performance results for instances of size 30 are indicated by circles, while stars mark results for instances of size 50. The dashed lines indicate the band within which performance differences are not statistically significant (significance was determined using the Mann-Whitney U test and setting the significance level at 0.05 [59]). Mean run-times were obtained from 10 runs per instance and algorithm, and we only show data points for the runs where the best known solution quality was reached at least in some runs out of 10 by both algorithms. When unsuccessful runs were present, the expected time was calculated as in [101]. For random sequences of length 50, there are three instances for which this was the case. Detailed results for both successful and unsuccessful runs are given in Appendix A.

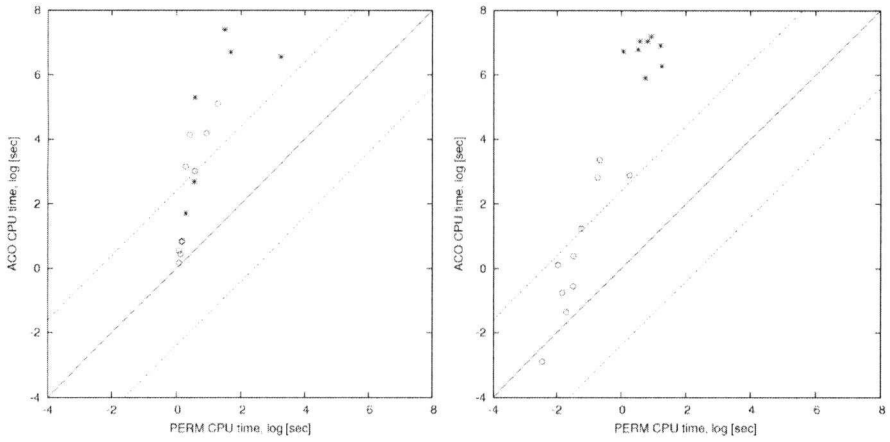


Figure 4.7: Mean CPU time (natural log transformed) required by ACO *vs* PERM for reaching the best solution quality, as observed over 10 runs with a cut-off time of 1 CPU hour for sequences of length 30 and 50 in 3D. The left and right plots show the results for the biological and random test-sets, respectively. Performance results for instances of size 30 are indicated by circles, while stars mark results for instances of size 50. The dashed lines indicate the band within which performance differences are not statistically significant (significance was determined using the Mann-Whitney U test and setting the significance level at 0.05 [59]). Mean run-times were obtained from 10 runs per instance and algorithm. We only show data points for the runs where the best known solution quality was reached at least in some runs out of 10 by both algorithms. When unsuccessful runs were present, the expected time was calculated as in [101]. For biological sequences of length 50, there are four instances and for random instances of length 50, there are two instances for which this was the case. Detailed results for both successful and unsuccessful runs are given in Appendix A.

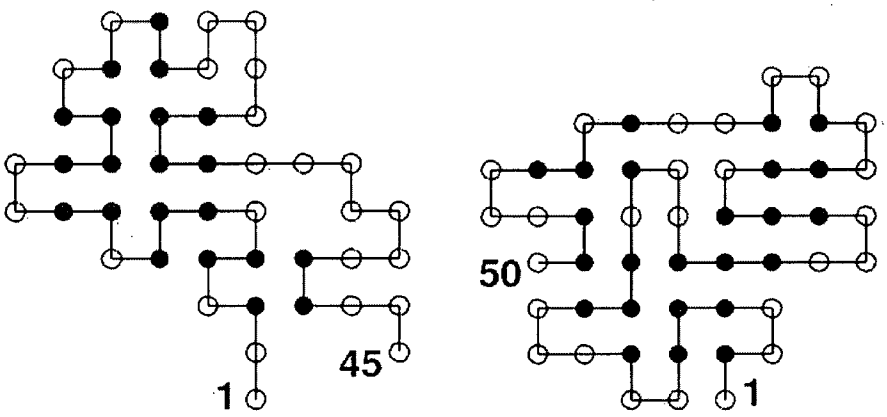


Figure 4.8: Left side: Lowest energy conformation of a biological sequence (B50-7, 45 amino acids, energy -17) that is harder for PERM ($t_1 = 271$, $t_2 = 299$, $t_{exp} = 284$ CPU seconds) than for ACO ($t_{exp} = 130$ CPU seconds; cut-off time 1 CPU hour). Right side: Lowest energy conformation of a biological sequence (B50-5, 53 amino acids, energy -22) that is much harder for ACO than for PERM; within a cut-off time of 1 CPU hour, both ACO and PERM reached this energy in 10 out of 10 runs in $t_{avg} = 820$ and $t_1 = 5$, $t_2 = 118$, $t_{exp} = 9$ CPU seconds on average, respectively.

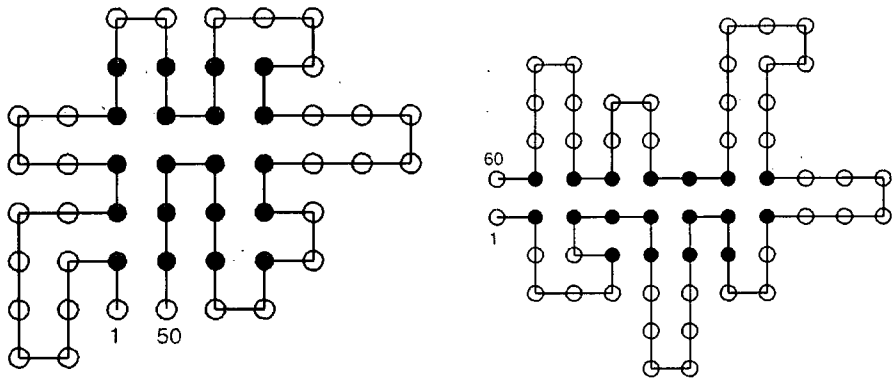


Figure 4.9: Left side: Unique minimal energy conformation of a designed sequence, D-1 (length 50, energy -19); ACO reaches this conformation much faster than PERM when folding from the left end (mean run-time over 100 successful runs for ACO: 236 CPU seconds, compared to $t_1 = 3\,795$, $t_2 = 1$, $t_{exp} = 2$ CPU seconds for PERM). Right side: Unique native conformation of another designed sequence, D-2 (length 60, energy -17). ACO finds this conformation much faster than PERM folding from either end (mean run-time over 100 successful runs for ACO: 951 CPU seconds, compared to $t_1 = 9\,257$, $t_2 = 19\,356$, $t_{exp} = 12\,524$ CPU seconds for PERM).

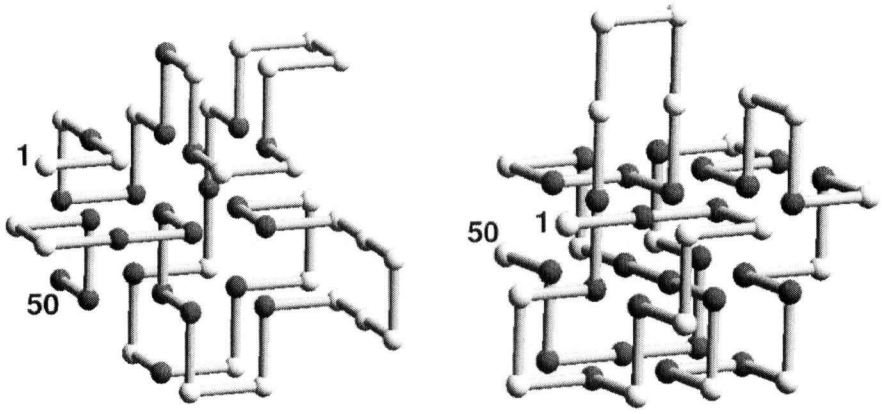


Figure 4.10: Left side: Lowest energy conformation of random sequence R50-9 (50 amino acids, energy -30), which is harder for PERM when folding from the left end than for ACO. With a cut-off time of 1 CPU hour, ACO reached this energy in 10 out of 10 runs with $t_{exp} = 1000$ CPU seconds, while PERM failed to find a conformation with this energy in 7 out of 10 runs when folding from the left end ($t_1 = 9\,892$, $t_2 = 2$, $t_{exp} = 3$ CPU seconds). The conformation displayed here has relative directions: DSRUURLRRLDDLURRULRSDURURL-RRDDUSRRSULRRDUULSLLUR. Right side: Lowest energy conformation of random sequence R50-7 (50 amino acids, energy -38), which is much harder for ACO than for PERM. With a cut-off time of 1 CPU hour, PERM reached this energy in two out of 10 runs when folding from the left and in 10 of 10 runs when folding from the right end in $t_1 = 15\,322$, $t_2 = 46$, $t_{exp} = 92$ CPU seconds, while the lowest energy reached by ACO over ten runs was -37. The conformation displayed here has relative directions: SUURUDULUUDSSUUSRUDDLDSSU-URRUULDLDRDDURLLDLRUSUR.

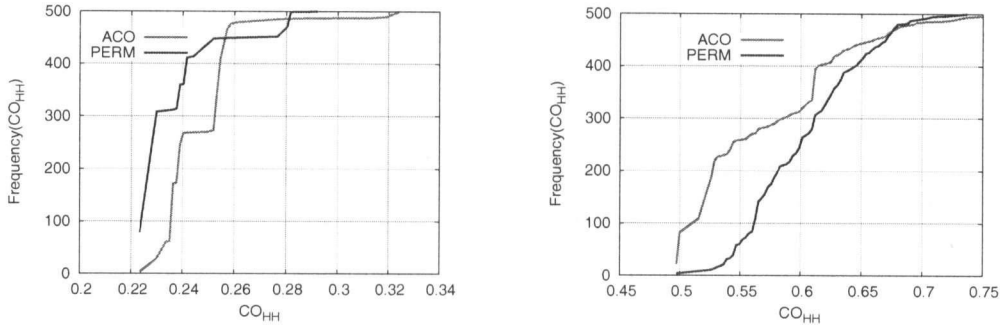


Figure 4.11: Distributions of H-H contact order for 500 conformations of Sequence S1-7 from Table 4.1 (60 amino acids) in 2D (left side) and Sequence S1-5 from Table 4.1 (48 amino acids) in 3D (right side) found by ACO and PERM.

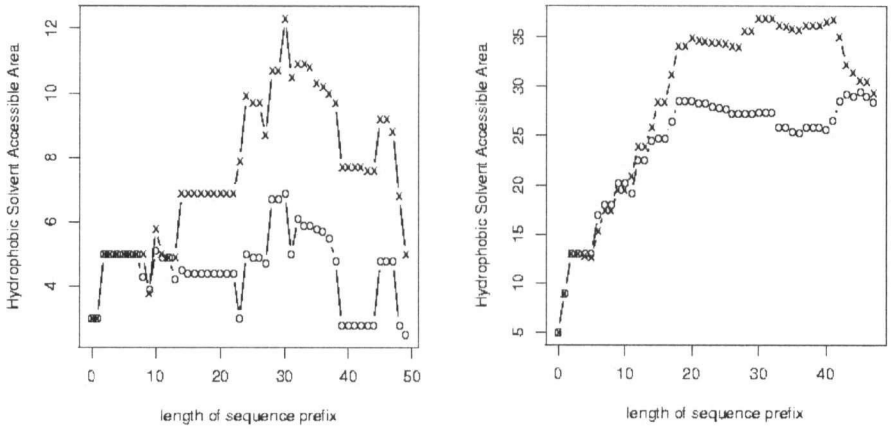


Figure 4.12: Mean hydrophobic solvent accessible area as a function of prefix length for a biological sequence (B50-4, 50 amino acids) in 2D (left side) and Sequence S2-6 from Table 4.1 (48 amino acids) in 3D. Crosses and circles represent mean values for an ensemble of 100 native structures found by ACO and PERM, respectively.

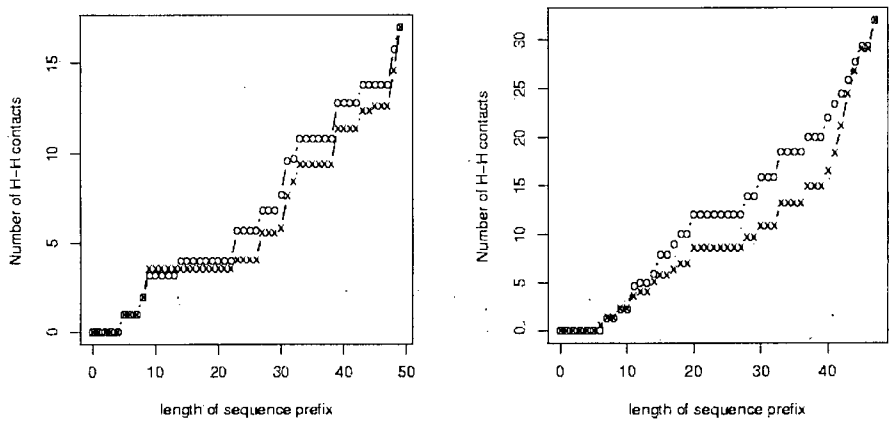


Figure 4.13: Mean number of H-H contacts as a function of prefix length for a biological sequence (B50-4, 50 amino acids) in 2D (left side) and Sequence S2-6 from Table 4.1 (48 amino acids) in 3D. Crosses and circles represent mean values for an ensemble of 100 native structures found by ACO and PERM, respectively.

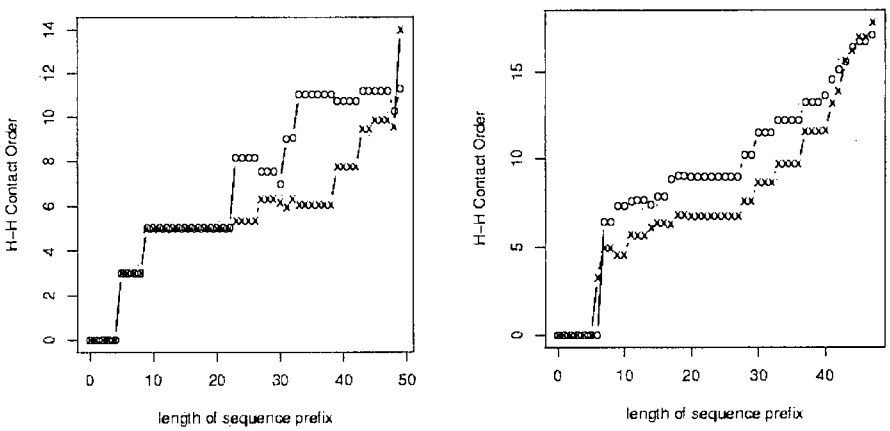


Figure 4.14: Mean H-H contact order as a function as a function of prefix length for a biological sequence (B50-4, 50 amino acids) in 2D (left side) and Sequence S2-6 from Table 4.1 (48 amino acids) in 3D. Crosses and circles represent mean values for an ensemble of 100 native structures found by ACO and PERM, respectively.

unique hydrophobic core; in 3D, none of the sequences studied here have a unique hydrophobic core.) This sequence has no structural nuclei at its ends; instead, the two ends interact with each other. ACO outperforms PERM by a factor of 2 on this sequence in terms of CPU time. Using a cut-off time of 1 CPU hour per run, PERM found the optimum with energy -17 in an average run-time of 284 CPU seconds ($t_1 = 271$ sec, $t_2 = 299$ sec), while using the same cut-off time and machine, ACO found the optimum in an average run-time of 130 CPU seconds.

We also designed two additional sequences, D-1 and D-2, of length 50 and 60, respectively, that have a unique native state in which both ends of the sequence interact with each other (see Figure 4.9). Sequence D-1 also has a structural nucleus near its C-terminus. When testing the performance of PERM and ACO on these sequences, we found that on D-1, ACO requires a mean run-time of 236 CPU seconds, compared to $t_1 = 3\,795$, $t_2 = 1$, $t_{exp} = 2$ CPU seconds for PERM (values are based on 100 successful runs). When this sequence was reversed, PERM started folding the sequence from the structural nucleus, and its mean run-time dropped to 1 CPU second. A result similar to that for sequence B50-7 was obtained for Sequence D-2, which has no structural nuclei at the ends, but a native state in which the ends interact with each other. Here, ACO was found to require a mean run-time of 951 CPU seconds (again, mean run-times were obtained from 100 successful runs), compared to $t_1 = 9\,257$, $t_2 = 19\,356$, $t_{exp} = 12\,525$ CPU seconds for PERM. As expected, reversing the folding order of the sequence in this case, did not cause a decrease in PERM's run-time.

We also analyzed native conformations of sequences on which PERM outperforms ACO and observed that the end from which PERM starts folding is relatively compact and forms a structural nucleus in the resulting conformation. An example of a conformation with the structural nucleus at the beginning of the sequence (near the N-terminus, *i.e.*, residue 1) is shown in the right panel of Figure 4.8. For this biological sequence (B50-5, 53 amino acids), PERM finds an optimal conformation with an energy of -22 in $t_1 = 5$, $t_2 = 118$, $t_{exp} = 9$ CPU seconds, while the average run-time for ACO is 820 CPU seconds. Our ACO algorithm generally performs worse than PERM on sequences that have structural nuclei at the ends, because it tends to spend substantial amounts of time compacting local regions in the interior of the sequence, while PERM folds more systematically from one end. These observations also hold in 3D, as seen from two random sequences folded in 3D (see Figure 4.10).

To further investigate our hypothesis, we studied differences between the distributions of native conformations found by ACO and PERM, respectively. For this purpose, we introduced the notion of *relative H-H contact order*, which captures arrangement of H residues in the core of the folded protein, and thus determines the topology of the conformation (the closely related concept of contact order was

first defined in [104]). Relative H-H contact order is defined as follows:

$$CO_{H\text{-}H} := \frac{1}{l \cdot n} \sum_{(i,j) \in HH, i<j-1} |i - j|, \tag{4.5}$$

where l is the number of H-H contacts, n is the number of H residues in the sequence, and i and j are interacting H residues that are not neighbours in the chain (contact (i, j) belongs to the set of H-H contacts HH). Intuitively, $CO_{H\text{-}H}$ specifies the average sequence separation between H-H residues in contact per H in the sequence.

Figure 4.11 shows the cumulative frequency distributions of relative H-H contact order values for sets of native conformations of a 2D (left panel) and 3D (right panel) standard benchmark instance, respectively, found by ACO and PERM over 500 independent runs. Each run was terminated as soon as a native conformation had been found. These results show that the ACO algorithm finds a set of native conformations with a wider range of H-H contact order values than does PERM. In particular, ACO finds conformations with high relative H-H contact order compared with PERM (more distant parts of the chain interact, for example, relative $CO_{H\text{-}H} = 0.324$ for Sequence S1-7 in 2D and relative $CO_{H\text{-}H} = 0.75$ for Sequence S2-5 in 3D are not found by PERM; similar results were obtained for other sequences). These results further support our hypothesis that in both, 2D and 3D, PERM is biased toward a more restricted set of native conformations. We performed analogous experiments for the case where PERM is allowed to keep certain statistics from one run to another, that is, runs are no longer independent, as in [60]. We found no significant differences in the set of conformations obtained.

To further examine the topological differences between ensembles of native conformations found by the two algorithms, we also looked at the hydrophobic solvent accessible area (defined as $SA_{H\text{-}H} := \sum_h E_h$, where E_h is the number of unoccupied lattice sites around H residue h), the number of H-H contacts, and the H-H contact order as a function of the length of the sequence prefix (starting from the N-terminus of the sequence, where PERM starts folding). In this analysis, we calculated the properties of interest mentioned above for the native conformations found in 100 independent runs by ACO and PERM, and plotted the mean values of the respective quantities as functions of the sequence prefix length (see Figures 4.12, 4.13, and 4.14).

As seen in Figure 4.12, ACO is less greedy than PERM, both in 2D (left side) and in 3D (right side). It also tends to leave more lattice sites around H residues accessible for future contacts with other H residues that appear later in the chain. This is also reflected in the mean number of H-H contacts formed when folding prefixes of increasing length; ACO tends to form fewer H-H contacts than PERM for

short- and medium-size prefixes (see Figure 4.13). By examining the dependence of absolute H-H contact order (defined as $\frac{1}{l}\sum_{(i,j)\in HH, i<j-1}|i-j|$, the average sequence separation per H-contact) on prefix length, we furthermore observed that, different from PERM, ACO realizes the bulk of its local H-H interactions in the middle of the given sequence (see Figure 4.14). This further confirms that ACO is capable of finding native conformations with structural folding nuclei that are not located at or near the end of a given protein sequence. The results illustrated in Figures 4.12, 4.13, and 4.14 are typical for all 2D and 3D HP instances we studied.

4.2.4 Discussion of ACO Results

Although conceptually rather simple, our ACO algorithm is based on a number of distinct components and mechanisms. A natural question to ask is whether and to which extent each of these contributes to the performance reported in the previous section. A closely related question concerns the impact of parameter settings on the performance of ACO.

To address these questions, we conducted several series of experiments. In this context, we primarily used three standard test sequences: Sequence S1-7 of length 60 and Sequence S1-8 of length 64 (long sequences) in 2D, as well as Sequence S2-5 of length 48 in 3D (all standard benchmark sequences for 3D are 48 amino acids in length). These sequences were chosen because the CPU time required to find the best-known solutions was sufficiently small to perform a large number of runs (100–200 per instance) needed to obtain reliable distributions.

Following the methodology of Hoos and Stützle [58], we measured run-time distributions (RTDs) of our ACO algorithm, which represent the (empirical) probability distribution over the run-time required to reach (or exceed) a given solution quality. The solution qualities used here are the known optimal or best-known energies for the respective sequences.

Pheromone Values and Heuristic Information

Two important components of any ACO algorithm are the heuristic function, which indicates the desirability of using particular solution components during the construction phase, and the pheromone values, which represent information learned over multiple iterations of the algorithm. Three parameters control the influence of the pheromone information versus heuristic information on the construction of candidate solutions: the relative weight of the pheromone information, α; the relative weight of the heuristic information, β; and the pheromone persistence, ρ.

In the first experiment, we investigated the impact of pheromone (α) and heuristic information (β), and their relative importance for the performance of our ACO

algorithm. As can be seen from the results shown in Figure 4.15, both the pheromone values and the heuristic information are important in 2D and 3D. When either is ignored either ($\alpha := 0$ or $\beta := 0$, respectively), the algorithm performs worse, particularly for longer 2D sequences ($n > 50$). For short 2D sequences with $n \leq 50$, the pheromone matrix does not appear to play a significant role, since sequences are generally easily solved by the subsidiary local search procedure alone. The optimal settings for α and β for most problem instances seem to be around $\alpha = 1$ and $\beta = 2$, as shown in Figure 4.15. It should be noted that in 3D, pheromone information appears to be less important than in 2D, which suggests that the specific solution components used in our algorithms are somewhat less meaningful in 3D. The goal of the next experiment was to further explore the role of experience accumulated over previous iterations in the form of pheromone values. To this end, we varied the pheromone persistence, ρ, while keeping other parameters constant. The results shown in Figure 4.16 show that in 2D, it is important to utilize past experience (*i.e.*, to choose $\rho > 0$), but also to weaken its impact over time (*i.e.*, to use $\rho < 1$). At the same time, closer examination revealed that for $\rho > 0$, attrition, or the construction of inextensible partial conformations, is a major problem, resulting from the accumulation of pheromone from multiple conformations. This is why the backtracking mechanism described earlier is extremely important for the performance of our algorithm in 2D. In 3D, for the previously stated reasons and because of the fact that the attrition problem is much less severe, the impact of the persistence parameter is generally smaller than in 2D.

Ant Colony Size and Length of Local Search Phase

During the initial empirical evaluation of our algorithm, we observed that ant colony size, *i.e.*, the number of ants used in each iteration, and the duration of local search, expressed as the number of non-improving search steps we are willing to consider before terminating the local search procedure, are correlated and significantly affect its performance. To further investigate this phenomenon, we conducted additional experiments in which we fixed the ant colony size and varied the maximal number of non-improving steps during local search, and vice versa. In this series of experiments, different colony sizes were considered, from a single ant up to a population of 5 000 ants. The number of non-improving steps in local search was varied, from 100 to 10 000. The results, shown in Figure 4.17, indicate that there is an optimal colony size of about 100 ants for both 2D and 3D. ACO is quite robust with respect to colony size, but performance decreases for very small or very large colony sizes. Intuitively, this is the case because the use of a population of ants provides diversification to the search process, which enables it to explore different regions of the underlying search space. Very small popula-

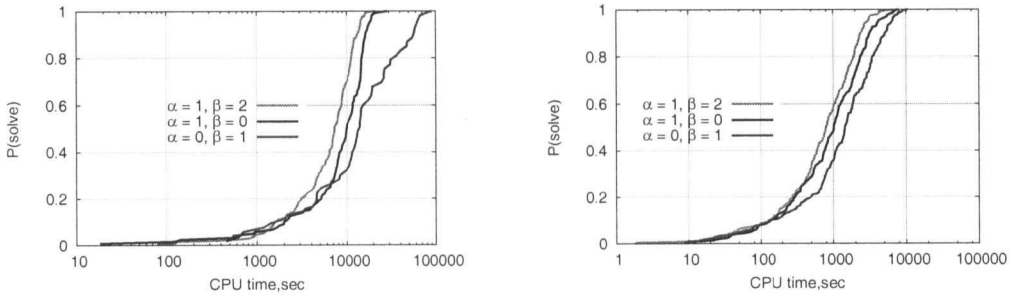


Figure 4.15: Effect of the relative weights of pheromone information, α, and heuristic information, β, on the average CPU time required for obtaining minimal energy conformations of Sequence S1-8 in 2D (length 64, left side) and Sequence S2-5 in 3D (length 48, right side).

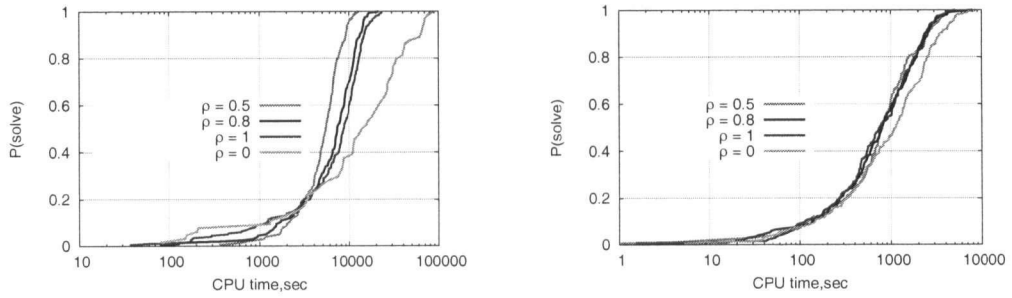


Figure 4.16: Effect of the pheromone persistence parameter, ρ, on the average CPU time required for obtaining minimal energy conformations of Sequence S1-8 in 2D (length 64, left side) and Sequence S2-5 in 3D (length 48, right side).

tions provide insufficient diversification, and the search stagnates easily, while for very large populations, the additional time required for running the search phases for each ant on the same sequential machine is no longer amortized by increased efficiency of the overall search process.

Our results also indicate that the performance of ACO is more sensitive to the number of non-improving steps than to ant colony size. The optimal value for the maximum number of non-improving steps tolerated (per ant) before the local search phase terminates was found to be around 1 000 for short 2D sequences ($n \leq 50$) and around 10 000 for long 2D sequences ($n > 50$). The latter value also appeared to be optimal for all 3D sequences considered here. This observation follows the intuition that more degrees of freedom, as present for longer sequences and in higher dimensions, require more time for local optimization, since for any conformation, improving neighbours tend to be rarer and hence harder to find.

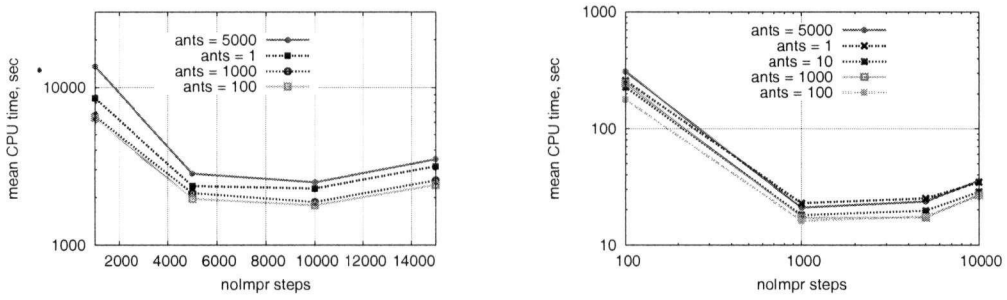


Figure 4.17: Mean CPU time required for finding minimum energy conformations of Sequence S1-7 in 2D (length 60, left side) and Sequence S2-5 in 3D (length 48, right side), as a function of ant colony size and the maximum number of non-improving local search steps.

Selectivity and Persistence of Local Search

As described previously, our ACO algorithm uses selective local search, *i.e.*, local search is performed only on a certain fraction of the lowest energy conformations. We observed that ACO is fairly robust with respect to the fraction of conformations to which local search is applied. Good performance was obtained for local search selectivity values between 5% and 50%, but performance was found to deteriorate

when local search is performed by all ants. Intuitively, similar to colony size, local search selectivity has an impact on search diversification. If too few ants perform local search, insufficient diversification is achieved, which typically leads to premature stagnation of the search process. On the other hand, if local search is performed by too many ants, the resulting substantial overhead in run-time can no longer be amortized by increased search efficiency.

Similarly to selective local search, pheromone update was performed only by the certain fraction of so-called 'elitist ants' whose solution quality after the local search phase is highest within the population. As in the case of local search selectivity, ACO shows robustly high performance for elitist fractions between 1% and 50% (results are not shown here), but performance deteriorates markedly when all ants in the colony are allowed to update the pheromone matrix.

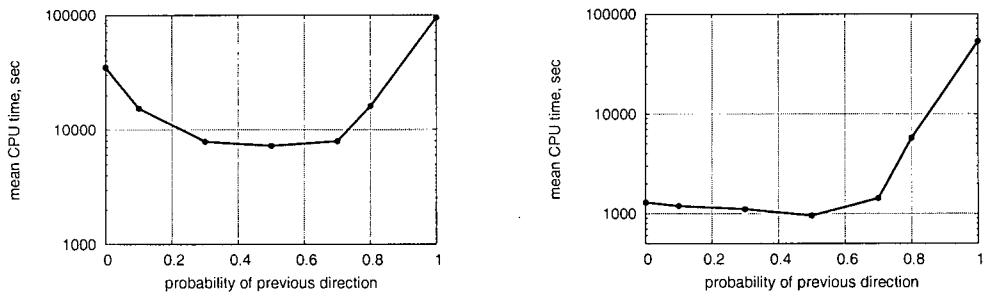


Figure 4.18: Mean CPU time required for finding minimum energy conformations of Sequence S1-8 in 2D (length 64, left side) and Sequence S2-5 in 3D (length 48, right side), as a function of the probability of retaining previous directions ($\hat{p}$) during long-range mutation moves.

In a final experiment, we studied the impact of the persistence of local search, *i.e.*, of the probability $\hat{p}$ of retaining (feasible) previous relative directions during long-range mutation moves. As can be seen in Figure 4.18, good performance is generally obtained for $\hat{p}$ values between 0.3 and 0.7. Both extreme cases, *i.e.*, $\hat{p} = 0$, which corresponds to an extremely H-contact-greedy mutation operator, and $\hat{p} = 1$, in which re-folding always follows previous directions when feasible, result in a substantial decrease in performance. When $\hat{p} = 0$, the decrease in performance in 3D is smaller than in 2D. This result is due to the fact that there is no severe attrition in 3D as in 2D, where greedy placement of H residues leads to early

formation of very compact partial conformations that often cannot be extended into valid complete conformations. The performance decrease for high $\hat{p}$ values is due to insufficient ability of the chain to fold into a new conformation accommodating well the local change in structure that triggered the re-folding.

4.3 Summary

We have shown that all components of our ACO algorithms contribute to its performance. In particular, performance is affected by the following components and parameters, listed in order of decreasing impact: pheromone values, termination criterion for local search, persistence of long-range moves, ant colony size, pheromone persistence, heuristic function, selectivity of local search, and selectivity of pheromone update (*i.e.*, fraction of elitist ants).

One of the important results of our study is the observation that the subsidiary local search procedure is crucial for the performance of the algorithm. In particular, to ensure that high-quality conformations are obtained, it is very important to allow the local search procedure to run sufficiently long. In an earlier version of our algorithm [121], we used substantially more stringent termination criteria. These criteria forced us to additionally use non-greedy local search (probabilistic iterative improvement, which accepts worsening steps) in addition to the greedy local search procedure used here. The results presented in this study indicate that by using a new and simpler local search procedure, ACO achieves better performance. This finding is probably due to the fact that the new local search procedure is based on a type of long-range move that leads to a larger effective search neighbourhood.

Generally, our empirical results indicate that our new ACO algorithm is able to fold sequences that have structural folding nuclei somewhere within the sequence and not at the ends more efficiently than the best-performing methods for this problem. Additionally, our algorithm finds a more diverse set of optimal states for HP sequences. Considering the simplicity of the underlying local search procedure and the evidence of the importance of all of the parameters of the algorithm, this property is a result of the search diversification provided by the population-based, probabilistically biased, adaptive construction mechanism that is characteristic of ACO algorithms.

Chapter 5

Adaptive Bin Framework Search, Introduced for the FCC β-Sheet Protein Folding Problem

Twilight was beginning. Blue veils hung in the trees. It was not any definite object that he recognized – no houses or villages or hills – it was the landscape itself that suddenly spoke.

Erich Maria Remarque

The performance of stochastic local search algorithms is critically dependent on the properties of the search landscape encountered, such as the number and distribution of local minima, the degree of landscape ruggedness (measured using fitness distance correlation, for example), and detailed information on the plateau and basin structure of a given landscape. Therefore, reactive search strategies that can extract important features of the landscape and adapt the search strategy accordingly are invaluable for solving problems with complex energy landscapes.

It is evident from an analysis of the literature describing search methods for protein folding, that adaptive search strategies have not been widely studied for this problem. In this chapter, we describe a general bin framework that can be used adaptively as an efficient search method for this complex problem. We also provide an extensive empirical comparison with existing state-of-the-art methods, which we carried out by re-implementing these methods and carrying out an empirical study.

In this part of our work, we chose the FCC lattice over the cubic lattice for the following reasons: (1) the FCC lattice, as mentioned previously, is free from some of the artifacts encountered on the cubic lattice (such as the parity problem, inability to model secondary structure, and others); (2) due to the lack of results in the literature for the state-of-the-art search methods for protein folding (such as

REMC) for the cubic lattice, we are unable to conduct an empirical comparison with the best search methods for protein folding on a simpler cubic lattice. Additionally, the FCC lattice was chosen over more realistic discrete off-lattice models due to the unavailability of universally used energy functions and the absence of an appropriate data set for which empirical results of the best-performing algorithms (such as REMC) in the literature exists.

We developed a new stochastic local search that uses a novel bin framework for storing a diverse set of conformations (candidate solutions) in memory. Promising conformations, that satisfy the energy and diversity criteria, encountered during the search are stored for future retrieval when a search stagnation is detected (both storage and retrieval mechanisms presented here represent an adaptive component of the search). Solutions are retrieved from a pool of stored conformations according to their energy.

5.1 The Bin Framework and the Bin Framework Monte Carlo Algorithm

To be able to search complex energy landscapes efficiently, we devised a flexible framework that stores a population of candidate solutions that are of good quality (have low energy) and represent a diverse set of conformations encountered during the search. This is performed by adapting energy and diversity thresholds during the search for bins that store conformations at different energy ranges. Energy thresholds for each bin are calculated based on the conformations already stored in the bin, and retrieving conformations adaptively once search stagnation is detected.

Promising conformations whose energy is lower than a bin's threshold and that satisfy the diversity criterion are placed into the appropriate bin according to their energies (each bin considers a certain window of energies, see Figure 5.1). It should be noted that sorting conformations into bins by a window of different energies is suggested in order to pick a diverse set of promising conformations that can help to overcome large energy barriers during the search. Thus, a pool of promising conformations encountered during the search and stored for future re-use is subdivided into bins of a certain energy window size ΔE_i. The energy window size represented by bins is a parameter that for simplicity is kept constant for all of the bins ($\Delta E_1 = \Delta E_2 = \ldots = E_i = \ldots = \Delta E_n$). For example, bin 1 can have conformations of energy values ranging from 0 to -10 [ε_0], bin 2 will hold conformations of energies -10 to -20 [ε_0], and so on.

Under the FCC model associated with each conformation c encountered, there are the following properties:

1. its total energy, $E(c)$, subdivided into short-range $E_{SR}(c)$ and long-range energy $E_{LR}(c)$ according to the FCC β-sheet potential used [52]
2. an array of residues in extended β-state, $c.BetaEnergy[i]$, for $i = 2 \ldots N-2$ (where N is the protein length)
3. its vector representation on the FCC lattice, $V = v_1 v_2 \ldots v_{N-1}$

The following properties and parameters are associated with each bin i:

1. The capacity of the bin, $bins[i].capacity$, specifying the number of conformations the bin can hold. This is a parameter of the algorithm that for simplicity is set to the same fixed value for all bins.
2. The current number of conformations stored, $bins[i].currentNumber$. The content of a bin, *i.e.*, conformations, is stored as a sorted list (from the lowest to the highest) according to the objective function (energy) for easy future retrieval, since lower-energy conformations are more desirable.
3. The bin's energy threshold, E_i^+. This is the highest energy that a conformation can reach and still be placed into the bin. It is adapted during the search based on the energies of conformations placed in the bin. To be selective when the bin is full, and to be less selective when it still has room, the highest energy among conformations already in the bin is picked as a threshold if the capacity has been reached; the threshold is equal to 0 until then.
4. The Hamming distance diversity threshold, HD_i, which specifies how different a conformation has to be from other conformations of the same energy already stored in order for the new conformation to get placed into a bin. The Hamming distance between a newly considered conformation c and all conformations c's with the same energy already in the bin is calculated as follows:

$$HD = \sum_{i=2}^{N-2} |c'.BetaEnergy[i]/numberOf(c') - c.BetaEnergy[i]|/\varepsilon_B N \tag{5.1}$$

where $\varepsilon_B = 4.0$ [ε_0] is the energy contribution of each β-residue, as in [52]. The sum goes from $i = 2$ to $i = N-2$, since the first residue and the two last residues can never be in the extended β-state. We consider different Hamming distance criteria HD_i for different bins i. As has been observed experimentally and is consistent with the funneled picture of the energy landscape, there are more conformations at higher energies, therefore, the criterion should be more stringent, and fewer conformations at lower energies,

where the Hamming distance criterion can be relaxed. This is performed by specifying two Hamming distance parameters to the algorithm HD_{MAX} (the highest Hamming distance used for bins with high energy) and HD_{MIN} (the lowest Hamming distance used for low-energy bins). A particular Hamming distance criterion used for bin i is determined by assigning linearly spaced Hamming distance criteria for bins between values HD_{MAX} and HD_{MIN} in the following way:

$$HD_i = HD_{MIN} + (i-1)(HD_{MAX} - HD_{MIN})/numBinsTotal, \quad (5.2)$$

where $numBinsTotal$ is the total number of bins in the bin framework, which is not a parameter of the algorithm but instead is defined by the ratio of two other parameters, as described below; i is the number of a bin ($i \in [1 \ldots numBinsTotal]$).

5. The width of the energy window represented by each bin, ΔE_i. For simplicity, it is kept constant for all bins.

The bin framework also has the following properties and parameters associated with it:

1. The total number of bins, $numBinsTotal$.

2. The energy range of interest, ΔE, which is a parameter of the algorithm. If conformations encountered during the search have energies within this range, we store them in the bin framework, provided they satisfy the diversity criterion. Even though the width of the energy range of interest is kept constant, ΔE shifts down the funnel as the current estimate of the ground state E_0 is updated.

3. The current estimate of the ground state energy E_0, which is the lowest energy found so far.

4. The inverse temperature used for retrieval of conformations from bins $\beta_{bin} = 1/k_B T_{bin}$. This is a parameter of the algorithm.

Figure 5.1 depicts some of the properties of bins and conformations in a bin and the overall relationship of conformations stored in the framework to the energy landscape. We keep the number of bins constant during the run of the algorithm. This number is determined by the interval of energies of interest and the energy window width used:

$$numBinsTotal = \Delta E / \Delta E_i. \quad (5.3)$$

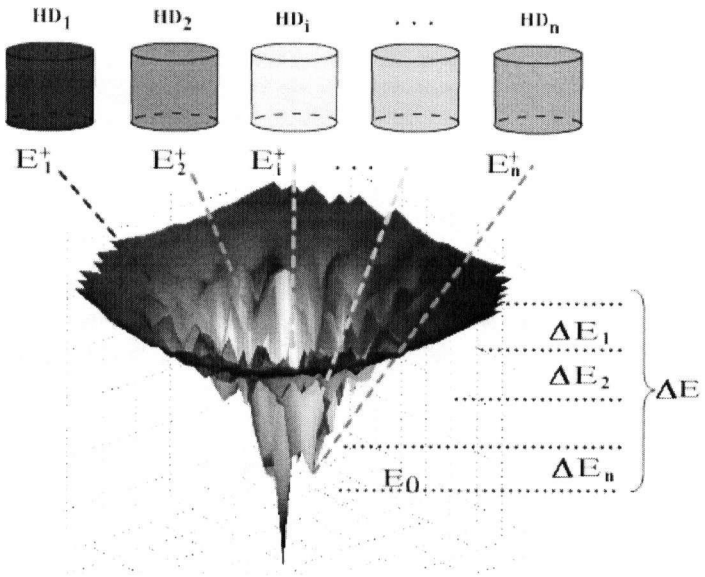


Figure 5.1: An illustration of how candidate solutions at a given state of the bin framework relate to the search space of a given problem instance. E_0 is the best solution quality found so far and serves as an estimate of the ground state energy, ΔE is the energy range of interest, and conformations within this range are binned. Each bin i has energy threshold E_i^+, diversity threshold HD_i, and energy window ΔE_i.

The base energy of interest, defined as $E_0 + \Delta E$, is the highest energy a conformation c can have and still be considered for placement into the bin system. The range ΔE is the estimate of the maximal barrier height that needs to be surmounted to get to lower energy states, and is a parameter of the algorithm.

The bin framework is conceptually a very rich framework that allows for a great amount of flexibility. Design decisions adopted in our implementation of the algorithm proposed here were guided by the principles of simplicity and efficiency. Many other variations are possible and will be studied as part of future work.

In this work, the novel bin framework introduced is used to adaptively restart a canonical Monte Carlo search when it stagnates. In general, our bin framework can be used in combination with any stochastic local search algorithm. We chose a combination with the Monte Carlo method since this is the most widely studied search method applied to the protein folding problem. We therefore call this approach the Bin Framework Monte Carlo (BINMC) method. The outline of the algorithm is given in Figure 5.2.

We used the same move set as described in [52, 154] which involves $n/2$ attempts at double-bond moves (the location of the move is chosen uniformly at random and two bond vectors are modified) and two attempts at chain-end moves (where we are attempting to modify the location of the first or the last residue).

5.1.1 Storing Conformations in the Bin Framework

We run a single Monte Carlo chain search at a low enough constant temperature T_{MC}, which is a parameter of the algorithm; note that $T_{MC} = 1/k_B\beta_{MC}$. During a Monte Carlo run, a conformation c is placed into a bin, or attempted to be placed into a bin, if $E(c) \leq E_0 + \Delta E$ and either of the following conditions is satisfied:

1. The energy of a conformation c is the lowest energy observed so far ($E(c) < E_0$). The conformation is stored in the bin system and the estimate of the ground state E_0 is updated (function *PlaceIntoBin* is called – see Figure 5.3).

2. If a conformation c is accepted according to the regular Metropolis criterion, we check to see if the energy of c, $E(c) = E$, is lower than the appropriate bin's threshold energy E_i^+ and if the Hamming distance diversity criterion is satisfied – the Hamming distance between the conformation c and other conformations c' with the same energy E should be larger or equal to HD_i. We then add it to the bin (function *CheckPlaceIntoBin* is called – see Figures 5.4 and 5.5).

After a new conformation is added to the bin, we drop the conformation with the highest energy from the bin if the capacity of the bin has been reached.

```
procedure BINMC(Conformation c, noImprRetrieve, β_MC, β_bin, HD_MAX, HD_MIN, ΔE_i, ΔE)
  input: candidate conformation c,
         the number of non-improving steps noImprRetrieve,
         the inverse temperature for the Monte Carlo run β_MC,
         the inverse temperature for the bin framework β_bin,
         Hamming distance diversity criteria HD_MAX and HD_MIN,
         the window of energies considered by each bin i: ΔE_i,
         the range of energies of interest ΔE
  output: the lowest energy conformation c
  noImpr := 0;
  E_0 := 0;
  while (termination condition not satisfied) do
    run single chain Monte Carlo at inverse temperature β_MC for conformation c
    //during the run store accepted conformations that satisfy
    //energy criterion (defined by ΔE and bins' energy thresholds)
    //and the Hamming distance criteria (defined by HD_MAX and HD_MIN);
    if (accepted new conformation c)
      if (E(c) < E_0)
        PlaceIntoBin(c, E_0, HD_MAX, HD_MIN, ΔE_i, ΔE);
      else
        CheckPlaceIntoBin(c, ΔE_i, ΔE);
      end
    end
    if no improvement has bin made over the best energy E_0 in the current run
      noImpr := noImpr+1;
    else
      noImpr := 0;
    end
    if (noImpr > noImprRetrieve)
      //retrieve conformation from bins at inverse temperature β_bin
      c := RetrieveFromBin(β_bin,E_0);
      noImpr := 0;
    end
  end
  return c;
end
```

Figure 5.2: High-level outline of the main body of the Bin Framework Monte Carlo algorithm. E_0 is the current best solution quality; *noImprRetrieve* is a parameter of the algorithm that specifies number of non-improving steps over the best energy that are tolerated before a new conformation is retrieved from the bin system; HD_{MAX} and HD_{MIN} are parameters defining the Hamming distance diversity criteria for high-energy and low-energy conformations correspondingly; β_{MC} and β_{bin} are parameters that refer to the inverse temperatures used for the Monte Carlo run and for the bin framework correspondingly; ΔE_i is a parameter that represents the window of energies by each bin i; ΔE is a parameter defining the range of energies of interest. Conformations whose energy falls within this range are attempted to be stored in the bin framework. The functions *PlaceIntoBin* and *CheckPlaceIntoBin* are defined in Figures 5.3 and 5.4 correspondingly.

procedure *PlaceIntoBin(Conformation c, E_0, HD_{MAX}, HD_{MIN}, ΔE_i, ΔE)*
input: candidate conformation c,
the lowest energy found so far E_0,
Hamming distance diversity criteria HD_{MAX} and HD_{MIN},
the window of energies considered by each bin i: ΔE_i,
the range of energies of interest ΔE
output: none
$E_0 := E(c)$;
//since all ΔE_is are equal and are supplied to the algorithm
//as parameters, a particular ΔE_i is known and index i of the bin that stores
//energy range to which E belongs to is calculated as:
$i := E/\Delta E_i$;
if ($bin[i]$ is empty)
//linearly redistribute HD criteria between
//HD_{MAX} and HD_{MIN} to the bins (HD_j)
$HD_j := HD_{MIN}+(j-1)(HD_{MAX}\text{-}HD_{MIN})/numBinsTotal$;
end
insert conformation c in $bins[i]$ as the top conformation;
shift other conformations already stored (if any) by one entry;
if ($bins[i].currentNumber > bins[i].capacity$)
drop highest energy conformation from $bin[i]$ and recalculate E_i^+;
else
$bins[i].currentNumber$:=$bins[i].currentNumber + 1$;
end
end

Figure 5.3: Outline for the function used to place the conformation with the lowest energy encountered so far into the bin system, where c is the conformation with energy $E(c)$ to be placed into a bin, E_0 is the estimated ground state energy (the lowest energy seen in the simulation so far), HD_{MAX} and HD_{MIN} are parameters defining the Hamming distance diversity criteria for high-energy and low-energy conformations correspondingly, ΔE_i is a parameter that represents the window of energies considered by each bin i, ΔE is a parameter defining the range of energies of interest – conformations whose energy falls within this range are attempted to be stored in the bin framework.

```
procedure CheckPlaceIntoBin(Conformation c, ΔE_i, ΔE)
  input: candidate conformation c,
         the window of energies considered by each bin i: ΔE_i,
         the range of energies of interest ΔE
  output: boolean variable indicating if c was stored
  if (E(c) > E_0 + ΔE)
    //energy of c is outside the energy range of interest ΔE
    return false;
  end
  i := E(c)/ΔE_i;
  //if bin still has place or if E(c) is lower than bin's threshold;
  if (bins[i].currentNumber < bins[i].capacity or E(c) ≤ E_i^+)
    find insertion place for conformation c in bins[i];
    //make sure we satisfy diversity criterion;
    if (there are other conformations c' with the same energy in the bin)
      for (all residues) do
        calculate β − HammingDistance between c and
        average β configuration of all c's;
      end
      //normalize HammingDistance to get value between 0 and 1;
      normHD:=HammingDistance/N;
    else
      normHD:=1;
    end
    if (normHD > HD_i)
      insert conformation c;
      shift other conformations already stored by one entry if there are any with higher energies;
      if (bins[i].currentNumber > bins[i].capacity)
        drop the highest energy conformation;
        recalculate E_i^+;
      else
        bins[i].currentNumber:= bins[i].currentNumber+1;
      end
      return true;
    else
      return false;
    end
  else
    return false;
  end
end
```

Figure 5.4: Outline for the function used to place the conformation c with a low energy $E(c)$ encountered into the bin system, where E_0 is the estimated ground state energy (the lowest energy seen in the simulation so far), HD_{MAX} and HD_{MIN} are parameters defining the Hamming distance diversity criteria for high-energy and low-energy conformations correspondingly, ΔE_i is a parameter that represents the window of energies considered by each bin i, ΔE is a parameter defining the range of energies of interest – conformations whose energy falls within this range are attempted to be stored in the bin framework.

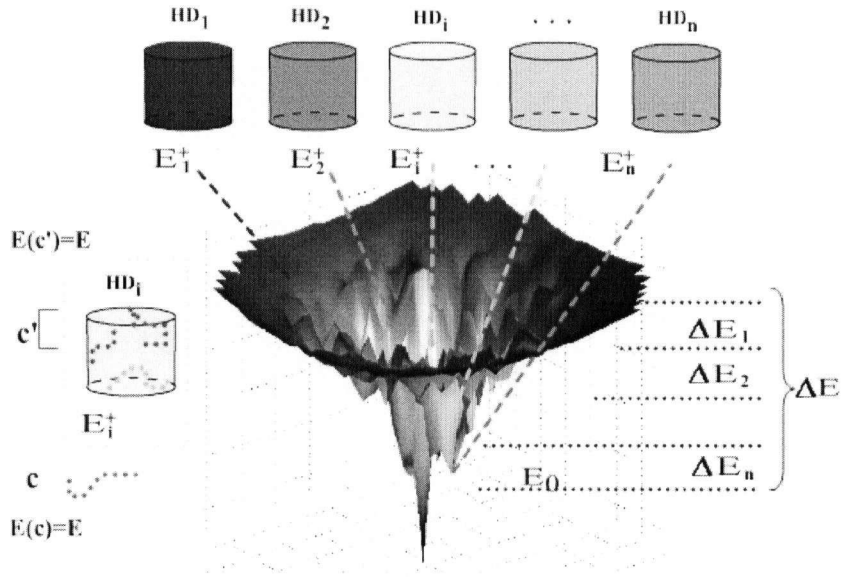


Figure 5.5: Placement of the low-energy conformation c with energy $E(c)$ into an appropriate bin i with the energy threshold E_i^+ and the Hamming distance criterion HD_i. In order for the conformation c to be stored in the bin framework the following conditions need to be satisfied: (1) $E(c) \leq E_i^+$, and (2) the Hamming distance (HD) between conformation c and conformations c' with the same energy already stored in the bin (if there are any) is larger or equal to HD_i.

In general, we need to make sure that conformations in the pool, or the union of all conformations stored in bins (1) represent all energy levels of interest, which is ensured by binning conformations at different levels; (2) represent as many possible β and non-β configurations as possible, which is taken care of by the Hamming distance criterion.

5.1.2 Retrieving Conformations from the Bin Framework

Let us now think about what we search for and what we most fear during the search. We want to reach the best local optimum, or the global optimum, as quickly as possible, and the worst hindrance is to get stuck in a local optimum that is still far from the global optimum. Therefore, our choice of search strategy (for example, the proposal mechanism for new candidate solutions) can be guided by the search progress made. During the search process, one can adaptively adjust important parameters of the algorithm to further the search (it may be noted that this is closely related to the concept of reactive search [9]). By varying the search strategy according to *a priori* defined transition probabilities, the approach results in an algorithm that sacrifices an exact relationship with the canonical ensemble. However, if we are interested in finding global minima, and not in obtaining the canonical ensemble to calculate physical properties of interest, this method can result in reducing the non-convergence, or quasi-ergodicity, in rugged energy landscapes. Therefore, we chose an adaptive strategy for retrieval: retrieval of a conformation from the bin system is performed when the search stagnates.

The simplest way to recognize when the search stagnates is to record the time (the number of steps) during which we have not observed any improvement on the lowest energy, (see, *e.g.*, [59]). This criterion relates directly to the objective of the search. Thus, if no improvement on the lowest energy has been seen for a certain number of steps ($noImprRetrieve$, which is a parameter of the algorithm) we retrieve a conformation from the bin system.

To retrieve a conformation from the bin framework we first need to choose a bin, that is, the range of energies from which we will pick a conformation, and then choose a particular conformation from the selected bin. Since lower energy conformations are preferred, given that the energy landscape of real proteins is believed to be funneled [105], we choose a bin and conformation according to the corresponding energy threshold and the energy of the conformation, respectively. Conformations can be chosen with or without replacement; here we limited ourselves to choosing conformation with replacement, since the same conformation can yield a different fold each time it is picked. The outline for the described simple retrieval strategy is given in Figure 5.6.

procedure *RetrieveFromBin*(β_{bin},E_0)
  input: the inverse temperature for the bin framework β_{bin},
    the lowest energy found so far E_0
  output: candidate conformation c
  if ($bins.totalStored = 0$)
    //no conformations are stored
    return null;
  end
  choose $binNum$ with probability $e^{-\beta_{bin}(E_i^+ - E_0)}$ among all bins i;
  choose $confNum$ inside the bin with probability $e^{-\beta_{bin}(bins[binNum].conf[j].Energy - E_0)}$
    among all conformations j stored in the bin $binNum$;
  $c := bins[binNum].conf[confNum]$;
  return c;
end

Figure 5.6: Outline for the function used to retrieve a conformation from the bin system. E_0 is the current best solution quality; β_{bin} is the inverse temperature used for the bin framework during the retrieval of conformations.

As in the stochastic tunneling approach [145], to lessen exponential decay of the probability function we used Boltzmann-based modified weights proportional to $e^{-\beta(E-E_0)}$. This weighting preserves the location of all minima, but maps the entire energy space from E_0 to the maximum energy 0 onto the interval $[0, 1]$. The dynamic process following the Boltzmann distribution can therefore pass through energy barriers of an arbitrary height.

5.2 Empirical Results and Discussion

To evaluate the performance of generalized ensemble methods re-implemented from the literature and the bin framework compared with the available data in the literature, we measured the average solution quality, median solution quality, and 25- and 75-percentiles. We also recorded the best solution quality obtained over 10 runs with the specified cut-off time. To further evaluate the performance of our BINMC algorithm and a number of generalized ensemble methods re-implemented from the literature, we analyzed the run-time distributions (RTDs) to obtain the best-known solution quality (or in some cases specified sub-optimal solution qualities) for instances of different sizes in 100 independent runs. The following lengths for the FCC β-homopolymers were used: 12, 24, 36, and 64. All experiments were performed on PCs with 2.4 GHz Pentium IV CPUs, 256Kb cache, and 1Mb RAM, running Redhat Linux (our reference machine). Their run-time was measured in

terms of the absolute CPU time required to obtain the specified solution quality or better.

5.2.1 Comparison of Results with the Literature

First, we compared our implementation of the simple Monte Carlo (MC) and the Replica Exchange Monte Carlo (REMC) methods with the earlier implementation of Gront *et al.* [52] described in the literature. In [52], the authors tested algorithms implemented on the homopolymer of length $N = 32$ and $N = 64$, but only provided results for the homopolymer of length 64.[1] For the protein of length 64 the authors believed that the lowest energy they reached (-374) is the ground state, but as was shown later [154], lower energies exist for this system (an energy of -387 has been reported in [154]).

In Table 5.1 we provide results averaged over 10 independent runs as done in [52] (for MC and REMC), and in [154] (for Parallel-hat Tempering (PHAT)). It should be noted that even though it is not evident from the paper [52], from personal communications with D. Gront, "MC with linear set of temperatures" means that they ran MC and annealed the temperature from 2.75 to 1.25 $[\varepsilon_0/k_B]$ (where $\varepsilon_0 = 1$ is the unit of interaction energy [154]). The exact annealing schedule was not provided, therefore, for our MC we chose a constant temperature of 1.25. Thus, results for the MCSA of Gront *et al.* may not be exactly comparable with our implementation of pure MC. We used average CPU times reported in [52] and [154] as the cut-off time for our algorithms, since the number of iterations varies significantly based on the implementation used. As seen from Table 5.1, our implementations on MCSA and REMC are comparable to the implementation reported in [52]; as discussed in the table caption, differences in execution environments were accounted for.

Our implementation of PHAT performed worse than was described in [154]. We, therefore, contacted Y. Zhang and tried to verify every aspect of the algorithm. Unfortunately, however, he could not reproduce the precise details and results, including the coordinates of the best state found, -387, due to data loss. Our novel BINMC algorithm performs better than MC and REMC, and than our implementation of PHAT. We used the following set of parameters for the bin framework for the homopolymer of length 64: $\Delta E = 30$, $\Delta E_i = 5$, $T_{MC} = 1.25$, $T_{bin} = 6.521$ (since units of T are $[\varepsilon_0/k_B]$, the inverse temperature is calcu-

[1] We contacted D. Gront [52] and Y. Zhang [154] but they no longer had the information or the ability to reproduce the data. Both authors commented that all methods were able to reach what they believe is the global minimum quite easily for the polymer of length 32. However, they could not tell what the exact value of the energy was, nor were they able to provide any information on the conformation with the lowest energy found.

lated as: $\beta = 1/T$) , $binCapacity = 100$, $HD_{MAX} = 0.8$, $HD_{MIN} = 0.01$, $noImprRetrieve = 2\,000\,000$ steps. These settings were determined in a series of experiments in which we studied the influence of different parameter settings; these will be further discussed in Section 5.2.3.

The best solution quality for the homopolymer of length 64 found by our bin framework is -391 (shown in Figure 5.7), which is lower than the energy of any conformation previously reported in the literature, as we have seen, the previous best energy was -387 [154]. We found conformations with energies -391 twice, after 47 hrs and 55 hrs. The two conformations found were mirror reflections of each other, and one of them is shown in Figure 5.7. We also show examples of other low-energy conformations found by our bin framework for the 64 amino acid homopolymer in Figures 5.8, and 5.9. Conformations with energies of -389, -388, -387, some of which are displayed in these figures, were found multiple times by BINMC within the CPU time cut-off of 10 hrs on our reference machines.

5.2.2 Further Comparison for Homopolymers of Length $12, 24, 32,$ and 64

To perform further comparison of the re-implemented methods from the literature (MC, REMC, PHAT) with BINMC, we tested the methods on instances of length $N = 12, 24, 32$ by performing 10 independent runs on each protein sequence. We reported the mean solution quality reached and the standard deviation observed within the 1 hr time cut-off. We used the following set of parameters for the bin framework for the homopolymers of length 12 and 24: $\Delta E = 20$, $\Delta E_i = 5$, $T_{MC} = 1.25$, $T_{bin} = 4.344$, $binCapacity = 100$, $HD_{MAX} = 0.6$, $HD_{MIN} = 0.01$, $noImprRetrieve = 100\,000$ steps. The bin framework parameters for the homopolymer of length 32 were set to: $\Delta E = 20$, $\Delta E_i = 5$, $T_{MC} = 1.25$, $T_{bin} = 4.344$, $binCapacity = 100$, $HD_{MAX} = 0.6$, $HD_{MIN} = 0.01$, $noImprRetrieve = 1\,000\,000$ steps. These particular parameter settings are discussed in Section 5.2.3.

As can be seen from our results presented in Table 5.2, all methods find what is probably the lowest energy (-39) for the homopolymer of length 12 under 1 sec CPU time on our reference machine. For the homopolymer of length 24, we are starting to see differences among the methods: BINMC slightly outperforms all other methods in terms of CPU time needed to reach what is probably also the lowest energy for the homopolymer of length 24 (-109). MC is the next best method in terms of performance, then PHAT, followed by REMC. The performance results for REMC and PHAT are worse than for MC because the polymer is too short, and simple MC methods that do not invest in exchanges between replicas perform better. For the homopolymer of length 32, BINMC outperforms other methods by

Method	Temperature set	$Time_{cut-off}$	$Energy_{avg} \pm sd$	$Energy_{min}$	$p-value$
MCSA [52]	annealed from 2.75 to 1.25	24 min (approx)	−349.3 (±2.1)	−362	
REMC [52]	linear 1.25 to 2.75	28 min (approx)	−368.2 (±0.8)	−373	
PHAT [154]	linear 1.25 to 2.75	1 hr 25 min (approx)	**-380.4** (±1.9)	−387	
our MC	1.25	24 min	−367.2 (±1.7)	−370	
our MC	1.25	28 min	−367.4 (±2.7)	−371	0.1367
our REMC	linear 1.25 to 2.75	28 min	−368.5 (±2.1)	−373	0.3425
our PHAT	linear 1.3 to 2.75	28 min	−367.5 (±3.3)	−372	0.1599
our BINMC	T_{MC} = 1.25, T_{bin}=6.521	28 min	**-370.3** (±4.3)	−379	
our MC	1.25	1 hr 25 min	−368.2 (±4.6)	−374	0.0006*
our REMC	linear 1.25 to 2.75	1 hr 25 min	−369.4 (±3.0)	−376	0.0008*
our PHAT	linear 1.3 to 2.75	1 hr 25 min	−369.5 (±3.2)	−376	0.0023*
our BINMC	T_{MC} = 1.25, T_{bin}=6.521	1 hr 25 min	**-375.7** (±3.8)	−383	

Table 5.1: Comparison of the solution quality obtained for the homopolymer of length $N = 64$ by the Monte Carlo Simulated Annealing (MCSA) [52], the Replica Exchange Monte Carlo (REMC) with a linear set of temperatures [52] and the Parallel-hat Tempering algorithm (PHAT) [154] with our implementation of Monte Carlo (MC), REMC and PHAT and our new Bin Framework Monte Carlo (BINMC). The time reported for MCSA and REMC from [52] is the estimated time required to run on 2.4 GHz machines (in the original paper authors used 500 MHz processor, therefore, reported times in [52] are conservatively divided by a factor of 4.8). The time reported for Parallel-hat tempering [154] is the estimated time to run on 2.4 GHz as well (in the original paper [154] CPU of 750 MHz was used, therefore we conservatively applied a factor of 3.2). The authors in [52] also implemented REMC with an exponential set of temperatures. The exact temperatures were not specified in the paper [52], however, and the authors could not recall it in personal communication (the lowest energy observed in this case was −374). The number of runs used for comparison was 10 for all algorithms. In the last column of the table, we report p-values indicating the probability that the null hypothesis of no difference between the mean energies reached over 10 runs for BINMC and a particular algorithm listed (within the same CPU cut-off time) is true; we used the Mann-Whitney U test to calculate p-values [59]; * indicates that p-values are below the significance level of 0.05 of wrongly rejecting the null hypothesis.

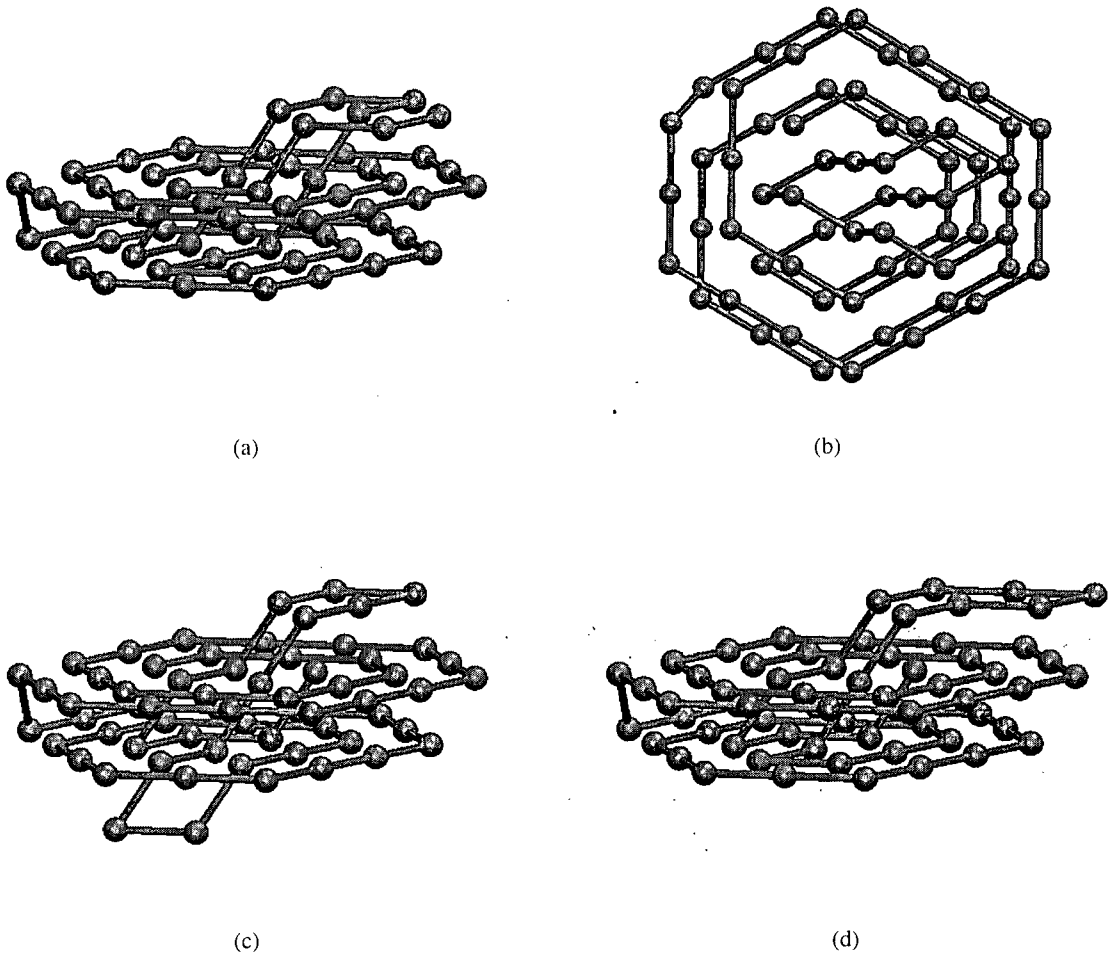


Figure 5.7: (a) The lowest energy conformation of the FCC 64 amino acids homopolymer found by our Bin Framework Monte Carlo method (energy -391, short-range energy is -212, long-range energy is -179). The detailed description of this conformation is also found in Appendix B; (b) same conformation, view from above; (c) a low-energy conformation of the FCC 64 amino acids homopolymer found by BINMC (energy -387, short-range energy is -212, long-range energy is -175); this was found in the same run that lead to the conformation with the best energy of -391; (d) another low-energy conformation of the FCC 64 amino acids homopolymer found in the same run of BINMC (energy -388, short-range energy is -212, long-range energy is -176).

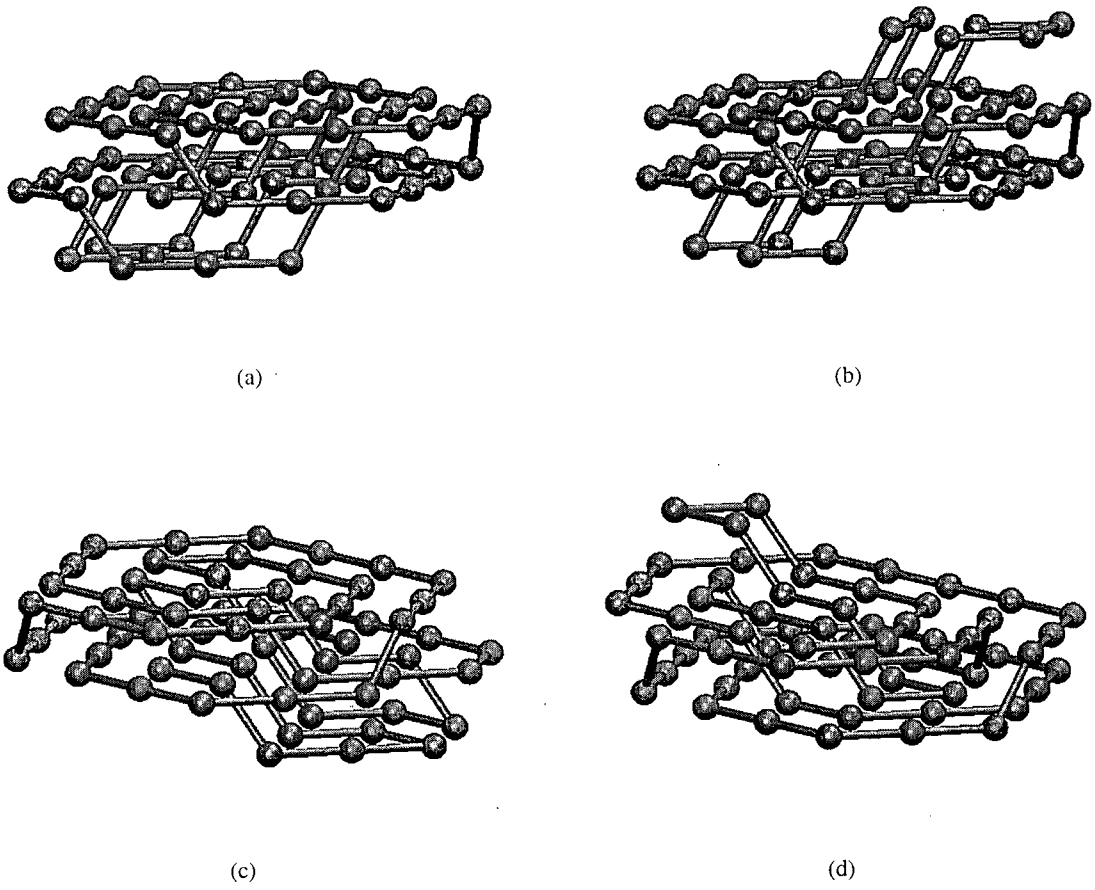
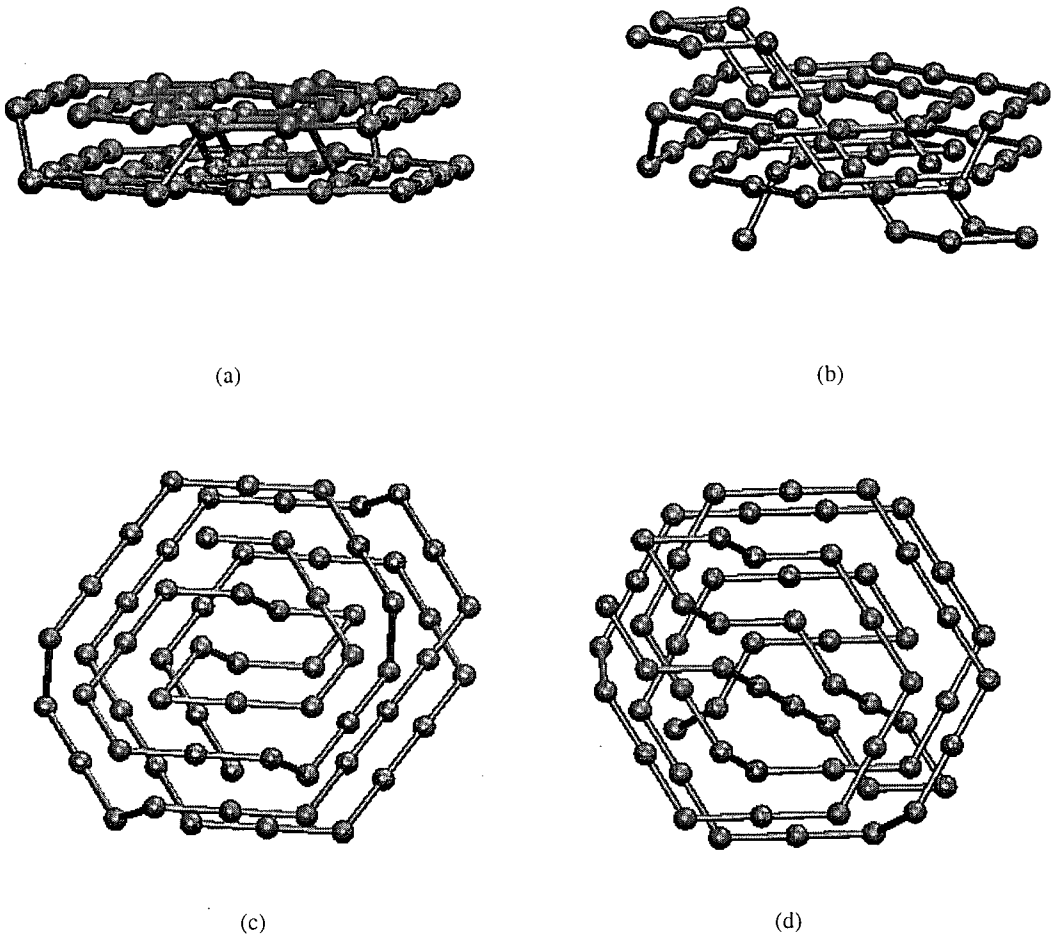


Figure 5.8: (a) A low-energy conformation of the FCC 64 amino acids homopolymer found by BINMC (energy -387, short-range energy is -208, long-range energy is -179); (b) another low-energy conformation of the FCC 64 amino acids homopolymer found in the same run of BINMC (energy -389, short-range energy is -212, long-range energy is -177); (c) a low-energy conformation of the FCC 64 amino acids homopolymer found by BINMC (energy -387, short-range energy is -208, long-range energy is -179); (d) another low-energy conformation of the FCC 64 amino acids homopolymer found in the same run of BINMC (energy -389, short-range energy is -220, long-range energy is -169).



(a) (b)

(c) (d)

Figure 5.9: (a) The low-energy conformation of the FCC 64 amino acids homopolymer found by our bin framework (energy -387, short-range energy is -220, long-range energy is -167); (b) another low-energy conformation of the FCC 64 amino acids homopolymer found by our bin framework (energy -387, short-range energy is -220, long-range energy is -167); (c) the same conformation as in part (a), view from above; (d) the same conformation as in part (b), view from above.

obtaining lower average energy (finding lower energy states such as -161 more often), next is PHAT, then REMC, followed by MC. We believe that the authors in [52] found only the sub-optimal solution quality of -158 for the 32 amino acid polymer, since energies lower than -158 are more difficult to obtain and methods start to vary in their ability to reach such energies. Unfortunately, they did not report the precise energy and could not reproduce it in personal communications. We show the best solution qualities found for polymers of length 12, 24, and 32 in Figure 5.10. These solutions seem to be unique in terms of short-range and long-range energy values, since all of the conformations found by different methods multiple times have the same short- *vs.* long-range energy interplay.

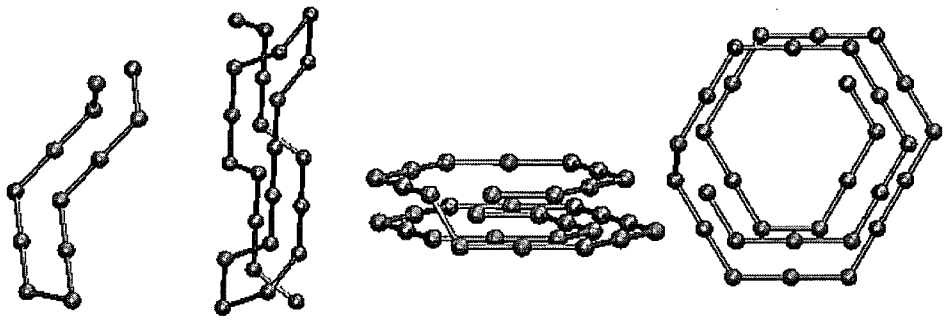


Figure 5.10: The lowest energy conformation of the FCC homopolymers of 12, 24, and 32 (same conformation from different point of view) amino acids (from left to right) found by all of the algorithms (corresponding energies are: -39, short-range energy is -28, long-range energy is -11 for the 12 amino acid polymer; -109, short-range energy is -68, long-range energy is -41 for the 24 amino acid polymer; and -161, short-range energy is -112, long-range energy is -49 for the 32 amino acid polymer). The detailed description of these conformations is also found in Appendix B.

Additionally, we carried out 10 independent long runs for the homopolymer of length 64 and 32 to further investigate behavior of different algorithms over an extended period of time. We used a longer cut-off time of 10 hrs on our reference machine, since the solutions reported in Table 5.2 are sub-optimal. As seen from the results presented in Table 5.3, BINMC outperforms our implementation of the state-of-the-art REMC and PHAT methods and the canonical MC method in terms of the solution quality reached. REMC was second best, then PHAT, then MC.

Method	*Length*	$Energy_{avg} \pm sd$	$Energy_{min}$	$CPU\ Time_{avg}$	$Time_{med}$	$Time\ q_{75}$	$Time\ q_{25}$	$p-value$
MC	12	−39 (±0)	−39	< 1 sec	< 1 sec	< 1 sec	< 1 sec	
REMC	12	−39 (±0)	−39	< 1 sec	< 1 sec	< 1 sec	< 1 sec	
PHAT	12	−39 (±0)	−39	< 1 sec	< 1 sec	< 1 sec	< 1 sec	
BINMC	12	−39 (±0)	−39	< 1 sec	< 1 sec	< 1 sec	< 1 sec	
MC	24	−109 (±0)	−109	5.0 min (±4.1 min)	5.5 min	7.2 min	1.5 min	0.1230
REMC	24	−109 (±0)	−109	18.3 min (±18.0 min)	16.3 min	19.4 min	4.3 min	0.0015*
PHAT	24	−109 (±0)	−109	8.7 min (±8.2 min)	6.6 min	11.7 min	2.9 min	0.0039*
BINMC	24	−109 (±0)	−109	**1.7** min (±1.2 min)	1.8 min	2.7 min	0.5 min	
Method	*Length*	$Energy_{avg} \pm sd$	*Lowest Energy*	$CPU\ Time_{avg}$	$Energy_{med}$	$Energy\ q_{75}$	$Energy\ q_{25}$	$p-value$
MC	32	−158.1 (±0.9)	−161	4.3 min (±8.4 min)	−158	−158	−158	0.0155*
REMC	32	−158.2 (±0.7)	−161	4.5 min (±8.6 min)	−158	−158	−158	0.0185*
PHAT	32	−158.3 (±0.9)	−161	5.8 min (±8.0 min)	−158	−158	−158	0.0214*
BINMC	32	**-158.9** (±0.6)	−161	23.0 min (±20.1 min)	−159	−159	−158	

Table 5.2: Comparison of the average solution quality obtained and the average time required for the homopolymers of lengths $N = 12, 24, 32$ for the re-implemented MC, REMC, PHAT, and BINMC. The time cut-off used was 1 hr on 2.4 GHz reference machine, and the averages were calculated from 10 independent runs. Temperature sets used for the re-implemented algorithms are the same as in Table 5.1. In the last column of the table we report p-values indicating the probability that the null hypothesis of no difference between the mean CPU run-time (for the homopolymer of length 24) or the mean energies reached over 10 runs (for the homopolymer of length 32) for BINMC and a particular algorithm listed is true; we used the Mann-Whitney U test to calculate p-values [59]; * indicates that p-values are below the significance level of 0.05 of wrongly rejecting the null hypothesis.

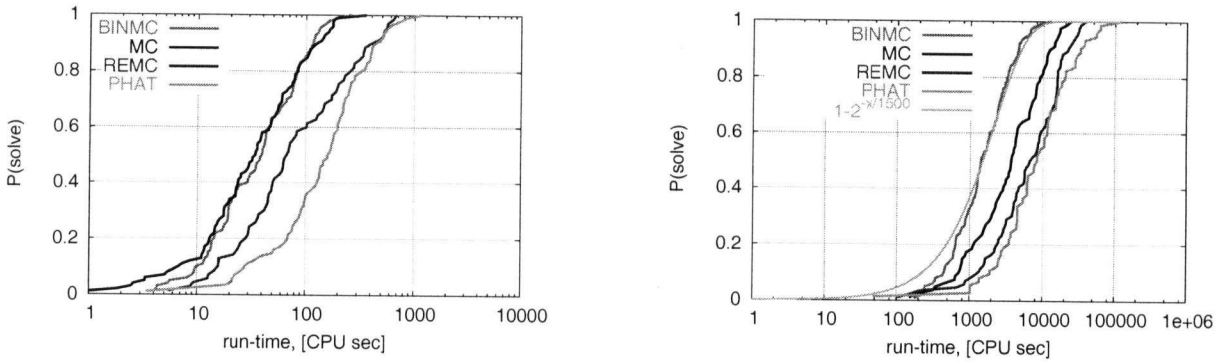
Method	Temperature set	*Length*	$Energy_{avg} \pm sd$	$Energy_{med}$	*Energy* q_{75}	*Energy* q_{25}	$Energy_{min}$	$p - value$
our MC	1.25	32	−158.7 (±1.9)	−159	−159	−158	−161	0.0271*
our REMC	linear 1.25 to 2.75	32	−159.6 (±1.3)	−160	−161	−158	−161	0.5471
our PHAT	linear 1.3 to 2.75	32	−158.9 (±1.4)	−159	−159	−158	−161	0.0638
our BINMC	T_{MC} = 1.25, T_{bin}=6.521	32	**-160.1** (±0.9)	−161	−161	−159	−161	
our MC	1.25	64	−372.2 (±2.3)	−372	−373	−371	−377	0.0005*
our REMC	linear 1.25 to 2.75	64	−376.1 (±3.5)	−376	−378	−373	−382	0.0521
our PHAT	linear 1.3 to 2.75	64	−374.1 (±3.8)	−374	−377	−371	−383	0.0120*
our BINMC	T_{MC} = 1.25, T_{bin}=6.521	64	**-379.5** (±3.3)	−381	−382	−376	−389	

Table 5.3: Comparison of the solution quality obtained for the homopolymers of length $N = 64$ and $N = 32$ by re-implemented MC, REMC with the linear set of temperatures, PHAT, and our new BINMC on 2.4 GHz reference machine with a time cut-off of 10 hrs over 10 independent runs. In the last column of the table, we report p-values indicating the probability that the null hypothesis of no difference between the mean energies reached over 10 runs for BINMC and a particular algorithm listed (within the same CPU cut-off time) is true; we used the Mann-Whitney U test to calculate p-values [59]; * indicates that p-values are below the significance level of 0.05 of wrongly rejecting the null hypothesis.

Next we conducted a more thorough comparison of the methods by comparing run-time distributions (RTDs) for all of the methods on problem instances of length 32 and 64. For the homopolymer of length 32, we used the sub-optimal solution quality of -158, since, as shown in Table 5.3, the time required to obtain the best solution quality of -161 is computationally prohibitive to conduct 100 runs for some of the methods. For the homopolymer of length 64, we also used the sub-optimal solution quality of -370 in order to make comparison computationally feasible for all of the methods. We conducted 100 independent runs on 2.4 GHz reference machine. Our results are presented in Figure 5.11. For the homopolymer of length 32, our BINMC and MC outperform all of the algorithms in terms of the time required to find the sub-optimal solution quality of -158. The next best algorithm was REMC, followed by PHAT. For the homopolymer of length 64 our BINMC outperforms all of the algorithms in terms of the time required to find the sub-optimal solution quality of -370. The next best algorithm was MC, followed by REMC and PHAT. It should be noted that for both the 32 and 64 amino acid homopolymer, the RTD for PHAT has a longer right tail, indicating that the probability of longer runs is higher. A somewhat unexpected result is that MC performs better than REMC and PHAT. However, we have to remember that the RTDs reported in Figure 5.11 are for sub-optimal qualities only. Since MC at a low temperature is "greedier" and does not run multiple chains at different temperatures nor attempts exchanges between them, it gets to sub-optimal energies faster. After reaching them, however, it gets stuck. The solution quality does not improve when running MC for a long time (10 hrs or more on our 2.4 GHz reference machine) for homopolymers of length 32 and 64, as shown in Table 5.3.

We also looked at the scaling behaviour of REMC and BINMC with homopolymer length. We measured the median run-time to reach the global minimum over 20 runs for sequences of length 12, 24, and 32 amino acids (the sequence of length 64 was not used since only BINMC reaches the lowest energy known). When three data points available are fitted with the line on a semi-logarithmic plot, the scaling behaviour appears to be exponential for both REMC and BINMC; the median run-time scales as $10^{0.34*N-5.2}$ for REMC and $10^{0.28*N-4.7}$ for BINMC.

Finally, we inspected the distribution of energies sampled by each method for the long homopolymer of length 64, based on approximately 5×10^9 conformations each. As seen in Figure 5.12 part (a) and (b), REMC and PHAT have typical energy distributions for each replica, as reported in [154]. In the case of PHAT, probabilities of low and high energies are enhanced, as expected. Interesting differences are observed when we examine the distribution of energies visited by MC and BINMC (see Figure 5.13). The Bin Framework Monte Carlo method samples energies according to the Boltzmann distribution, but this distribution is shifted towards lower energies.

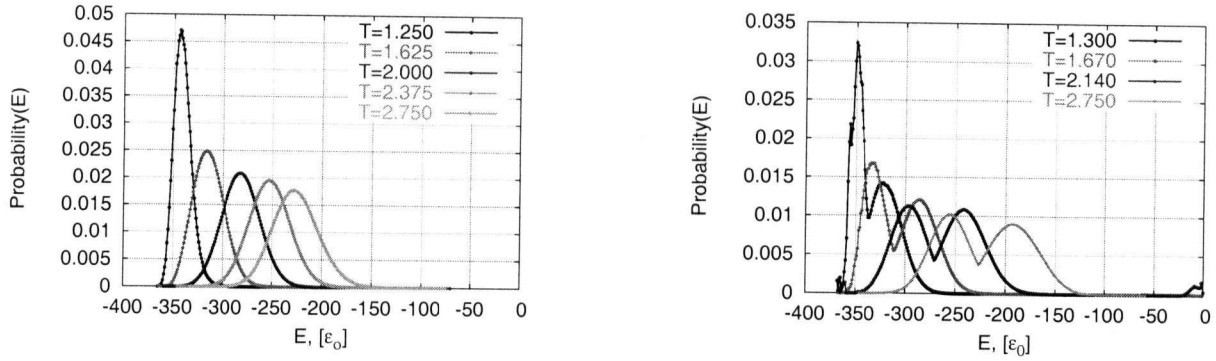


(a) (b)

Figure 5.11: Run-time distributions of CPU times on our 2.4 GHz reference machine to obtain a sub-optimal solution quality of -158 for the homopolymer of length 32 (part (a)) and to obtain sub-optimal solution quality of -370 for the homopolymer of length 64 (part (b)) using Monte Carlo (MC), Replica Exchange Monte Carlo (REMC), Parallel-hat Tempering (PHAT), and the Bin Framework Monte Carlo (BINMC). We fit the RTD of BINMC for the homopolymer of length 64 with exponential distribution, to show by example that RTDs for all algorithms are exponential. For all algorithms, 100 independent runs were performed. In all of them, the target energy was reached.

5.2.3 Discussion of the Bin Framework

We now turn our attention to the study of the parameter settings for the Bin Framework Monte Carlo method. We are interested in knowing what the reasonable values are for all of the parameters used. In this study, we performed 20 runs for the polymer of length 32, and record the time required to reach the best solution quality of -161. We chose this particular homopolymer to study differences in parameter settings since the solution quality of -161 is most likely the optimum, and the instance of length 32 is both challenging and still computationally feasible for performing multiple runs of our algorithm. To study the influence of different parameters, we vary one (or in some cases two parameters when they are closely related), and keep all of the other parameters of the algorithm constant. Unless indicated otherwise, parameters for the homopolymer of length 32 were



(a) (b)

Figure 5.12: Distributions of energies visited by different replicas in a representative run of the Replica Exchange Monte Carlo (REMC) (part (a)) and in a representative run of the Parallel-hat Tempering Monte Carlo (PHAT) (part (b)) for the homopolymer of length 64. The time cut-off used was 28 min on our reference machine.

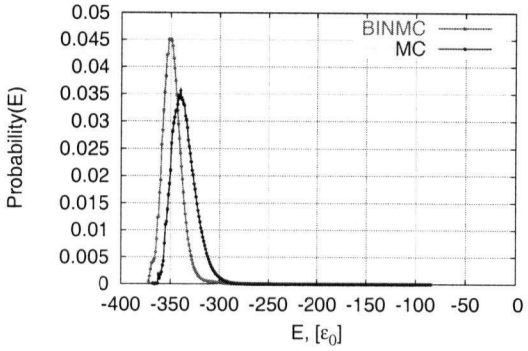


Figure 5.13: Distributions of energies visited by the Monte Carlo (MC) and our Bin Framework Monte Carlo (BINMC) for the homopolymer of length 64. The time cut-off used was 28 min on our reference machine.

fixed at the following values: $\Delta E = 20$, $\Delta E_i = 5$, $T_{MC} = 1.25$, $T_{bin} = 4.344$, $binCapacity = 100$, $HD_{MAX} = 0.6$, $HD_{MIN} = 0.1$, $noImprRetrieve = 100\,000$ steps. We plot the average CPU time required to reach the optimum over 20 independent runs on the 2.4 GHz reference machine.

The Number of $noImprRetrieve$ Steps

An important parameter of our Bin Framework Monte Carlo method is the $noImprRetrieve$ number of steps. It characterizes how often we reach into a bin and retrieve a conformation based on the indication that our search stagnated. This parameter determines a cut-off for a number steps performed without seeing an improvement over the best energy before reaching into bins to retrieve a new conformation. To understand the influence of the $noImprRetrieve$ step cut-off, we conducted 20 independent runs for the homopolymer of length 32 to reach -161 while varying $noImprRetrieve$ from 1 000 to 2 000 000 steps. As can be seen from the results shown in Figure 5.14, the $noImprRetrieve$ parameter significantly affects performance of BINMC. For the homopolymer of length 32, a value close to 1 000 000 steps gives the best performance. If we retrieve conformations too often, we prevent the search from getting to the promising local optima by restarting it with a new conformation too early. If we reach into the bin framework very infrequently, the simulation becomes equivalent to a pure Monte Carlo run at a constant T_{MC} temperature. We also found evidence in informal experiments that the best value of $noImprRetrieve$ steps increases with chain length; for the homopolymer of 64 amino acids $noImprRetrieve = 2\,000\,000$ steps results in good performance. This observation is consistent with the intuition that for longer chains, longer time is required to reach promising local optima.

Additionally, we compared BINMC (performing an adaptive retrieval of diverse conformations) with a simple restart strategy. If there was no improvement on the best energy observed for $noImprRetrieve = 1\,000\,000$ steps, instead of retrieving a conformation from bins we restarted the search with a newly constructed conformation. In this case, the simulation failed to reach the energy of -161 within one week of CPU time on our 2.4 GHz reference machine (only the energy of -159 was reached). Similarly, in the case of the homopolymer of length 64, only an energy of -365 was reached within one week of CPU time.

The Energy Range ΔE and the Bin's Temperature T_{bin}

The performance of our bin framework is also dependent on the ratio and the individual settings of the energy range of interest, ΔE, (conformations with energy within this range are binned) and the temperature that controls the probability of

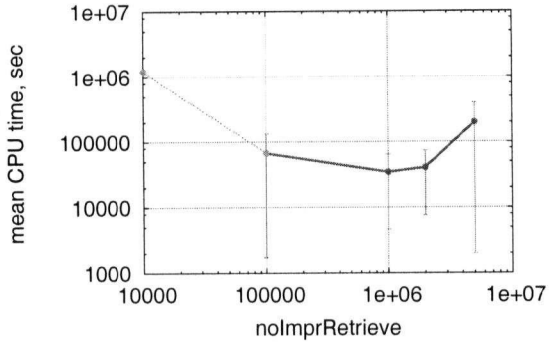


Figure 5.14: Mean CPU time required for finding minimum energy conformations (-161) of the 32 amino acid homopolymer over 20 independent runs on our reference machine as a function of the number of non-improving steps (*noImprRetrieve*) before retrieving a conformation from bins. Error bars indicate standard deviation observed. Dashed line indicates that energy of -161 was not found after 2 weeks' CPU time on our reference machine.

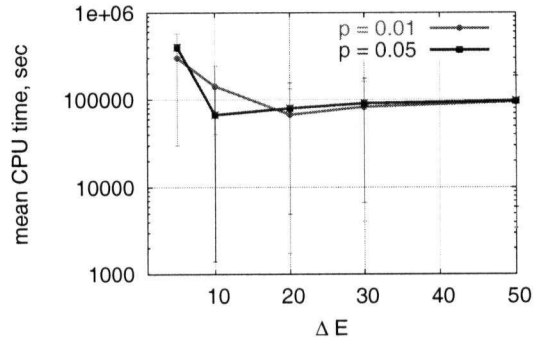


Figure 5.15: Mean CPU time required for finding minimum energy conformations (-161) of the 32 amino acid homopolymer over 20 independent runs on our reference machine as a function of combination of parameters: the energy range of interest (ΔE) and the bin's temperature (T_{bin}), $p = e^{-\Delta E/T_{bin}}$. Error bars indicate standard deviation observed.

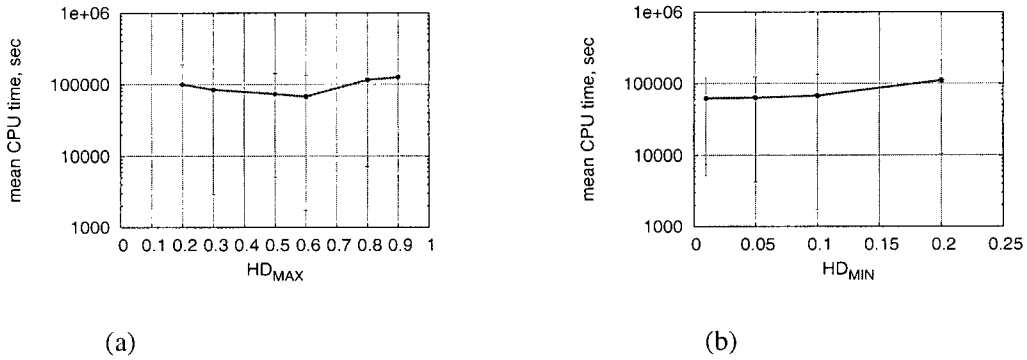
retrieving conformations of various energies, T_{bin}. Since it is not efficient to place conformations that will later have a very low probability of being retrieved, the relative settings of these two parameters are important. The probability of retrieval of the conformation with the highest energy ($E_0 + \Delta E$) from the bin system is equal to $p = e^{-\beta_{bin}\Delta E}$. Thus, for example, if we would like to have at least a 1% chance of retrieving the highest energy conformation from the bin system ($p = e^{-\beta_{bin}\Delta E} \geq 0.01$), the following condition must hold: $\frac{\Delta E}{T_{bin}} \geq 4.605$. Therefore, if we have fixed the desired probability of retrieval (p) and one other parameter (either ΔE or T_{bin}; we chose ΔE), we can calculate the value of another parameter (T_{bin}). We fixed p at 0.01 and 0.05 and varied ΔE from 10 to 50 $k_B T$ (T_{bin} was calculated as described previously). Our results presented in Figure 5.15 indicate that BINMC shows a rather robust performance for various combinations of ΔE and T_{bin}. The performance worsens when the ΔE range is too small and we are only storing a few of the best conformations encountered. The search time increases slowly as ΔE becomes too large and we are storing too many conformations. Each probability value (we tested $p = 0.01$ and $p = 0.05$) seems to have its optimal ΔE range, and as p increases the optimal ΔE seems to decrease. This observation is consistent with the intuition that if the retrieval probability is increased, the simulation can perform well with a smaller energy range of interest, since the binned conformations have a higher chance of being retrieved. From experiments with the longer homopolymer of length 64 (not reported here) larger ΔE results in good performance (particularly $\Delta E = 30$ and $T_{bin} = 6.522$, which renders p equal to 0.01, works well).

As for the relationship between T_{MC} and T_{bin}, here we only investigated the case when MC is run at a low temperature and the bin framework uses a high temperature for conformation retrieval. This scenario seems most promising, given the intuition that the search has to be efficient in minimizing conformational energy and the bin framework can provide diversification and intensification at the same time. We found that running MC at high temperatures, for example, $T_{MC} = 2.0$ for both 32 and 64 amino acid polymer, does not yield optimal results. Therefore, T_{MC} was kept constant at 1.25, which is below the transition temperature of 1.8 $[\varepsilon_0/k_B]$ reported for the homopolymer of length 64 [52, 154].

The Hamming Distance Criterion (HD_{MAX} and HD_{MIN})

The efficiency of our bin framework also depends on the diversity of conformations binned at each energy level. In this experiment, we investigated the role the Hamming distance criterion (controlled by HD_{MAX} and HD_{MIN}) plays in the overall optimization. First, we fixed HD_{MIN} (the fraction of residues that have to be different for low-energy conformations), and varied HD_{MAX} (the fraction of

residues that have to be different for high-energy conformations) from 0.2 to 0.9. Note that setting the Hamming distance criteria to 0 does not make sense, since the same conformation will be binned over and over. As seen in Figure 5.16 part (a), HD_{MAX} of about 0.6 seems to result in the best performance. As expected, if HD_{MAX} is set too low, the diversity of the set stored decreases and this impairs performance. If HD_{MAX} is set too high, very few conformations may exist to satisfy this stringent criterion.



(a) (b)

Figure 5.16: Mean CPU time required for finding minimum energy conformations (-161) of the 32 amino acid homopolymer over 20 independent runs on our reference machine as a function of the Hamming distance criterion (HD_{MAX} is varied, $HD_{MIN} = 0.1$ is kept constant, part (a) and HD_{MIN} is varied, $HD_{MAX} = 0.6$ is kept constant, part (b)). Error bars indicate standard deviation observed.

Next, we fixed HD_{MAX}, and varied HD_{MIN} from 0.01 to 0.9. As seen from Figure 5.16 part (b), if $HD_{MIN} > 0.1$ is set too high, the binning process becomes less efficient, since we are not binning all promising low-energy conformations. This happens because there are fewer conformations at low-energy levels and they are more similar to each other, compared with higher-energy level conformations.

In our informal experiments, both HD_{MAX} and HD_{MIN} criteria do not differ significantly for a longer polymer of length 64 ($HD_{MAX} = 0.8 - 0.6$ and $HD_{MIN} = 0.01$ seem to work well).

The Energy Window Width ΔE_i

In this experiment, we studied the performance of the Bin Framework Monte Carlo method as a function of the energy window width ΔE_i. Intuitively, ΔE_i controls the level of coarse-graining during the process of memorization of promising conformations. For simplicity, it is constant for all bins i in our implementation. As seen in Figure 5.17, we varied ΔE_i from 1 (when every single energy level is stored in its own bin) to 10 ε_0 (when 10 different energy levels are grouped in a single bin). We observed that our bin framework is quite robust with respect to the width of the energy window and that the optimal value appears to be around 5 ε_0. For the homopolymer of length 64, both $\Delta E_i = 5$ and $\Delta E_i = 10$ seem to perform well.

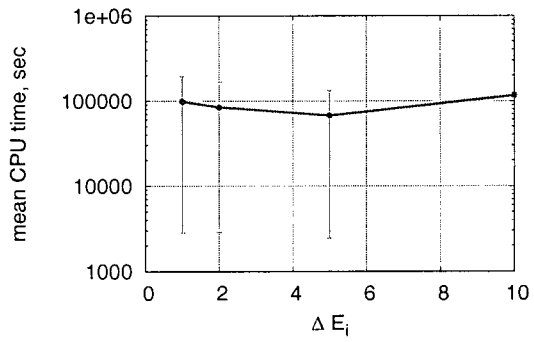


Figure 5.17: Mean CPU time required for finding minimum energy conformations (-161) of the 32 amino acid homopolymer over 20 independent runs on our reference machine as a function of the energy window width ΔE_i. Error bars indicate standard deviation observed.

The Capacity of Bins

In our final experiment, we studied the impact of the capacity of bins. The capacity of bins is responsible for determining the number of conformations stored and used for later retrieval. As can be seen in Figure 5.18, good performance is generally obtained for a capacity value of about 100 conformations per bin. Storing too few or too many conformations results in less efficient searching, due to the inability of the bin system to provide an effective diversification mechanism when too few

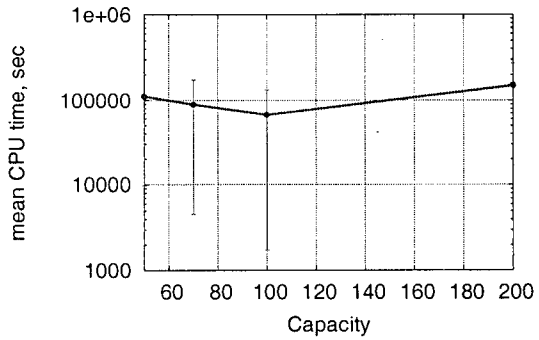


Figure 5.18: Mean CPU time required for finding minimum energy conformations (-161) of the 32 amino acid homopolymer over 20 independent runs on our reference machine as a function of the bin's capacity. Error bars indicate standard deviation observed.

conformations are stored, or due to the computational overhead encountered when the search has to memorize too many conformations. This observation is also consistent for the homopolymer of length 64, where a capacity of 100 seems to work well.

5.3 Summary

During our study of different parameter settings, we determined that the following parameters, in the order of their decreasing relevance, play an important role in the performance of the BINMC algorithm: (1) the number of non-improving steps (over the best energy) that are performed before reaching into a bin and replacing the current conformation in the Monte Carlo run; (2) the ratio between the width of the energy of interest considered for binning and the bin temperature; (3) the diversity criteria used during binning high- and low-energy conformations (the Hamming distance limits); (4) the width of the energy window considered by bins; and (5) the capacity of bins. This empirical study further shows that the bin framework performs better than the simple restart strategy and than the pure Monte Carlo algorithm, which it is based on. We also determined that all components of our algorithm are important for its efficiency.

The bin framework introduced in this chapter belongs to the class of model-

based search (MBS) methods, since it uses a non-parametric model represented by conformations stored in the bin framework. The model is updated during the search and influences the choice of a new candidate solution every $noImprRetrieve$ steps, when search stagnation is detected. In MBS for the discrete off-lattice model with structural fragment insertion as described in [20], the choice of a new candidate solution is influenced at every step by the pool of conformations stored. Thus, MBS exploration of the search space is only dependent on conformations stored [20]. Therefore, regions that are pruned from the model are eliminated and not explored any further. Since the bin framework provides only a subsidiary mechanism for generating candidate solutions when search stagnation is detected, it does not completely eliminate unexplored regions of the search space. This is achieved by running a non-model-based search (canonical MC) for a sufficiently long time to allow it to explore other regions of the search space. Additionally, MBS does not have a mechanism comparable to our diversity criteria between stored conformations. In MBS, N best conformations, the quality of which is based on the score described in Chapter 3, are stored.

The bin framework sorts conformations into bins representing different energy levels to make sure that the model contains as many energy levels of interest as possible while still reducing the search space. This aspect of the search is somewhat conceptually related to histogram-based sampling and search methods such as MUCA and ELP. It should be noted, however, that bin framework is a non-parametric model of the space composed of the actual subset of diverse promising candidate solutions. MUCA, on the other hand, and to some degree ELP, are based on a parametric model of sampling all energy levels with the same probability without emphasis on low energies. In addition, our bin framework applies different Hamming distance criteria based on the energy level of the bin to ensure that conformations stored are diverse and reflect the overall structure of the landscape, which is believed to be funnel-like for proteins.

Finally, the bin framework proposed here suggests a very general model that can be used adaptively to provide both intensification and diversification during the search. As has been shown in this work, even the most simple design choices resulted in very good performance of the algorithm and improved on the best-known solution quality for the homopolymer of length 64. Development of other adaptive and possibly more complex mechanisms therefore holds much promise.

Chapter 6

Construction Search for Identifying Folding Pathways

If the path be beautiful, let us not ask where it leads.

Anatole France

In this chapter, we propose a novel and simple construction-based approach to address the problem of identifying folding pathways as follows: given the topology of the native state, identify native contacts that form folding nuclei based on a graph theoretical approach that considers effective contact order (effective loop closure) as its objective function.

A number of computational methods for the prediction of folding nuclei already exist in the literature, but most of them rely on restrictive assumptions about the nature of nuclei or the process of folding. Our motivation was to develop a simple, efficient and robust algorithm to find an ensemble of pathways with the lowest effective contact order and to identify contacts that are crucial for folding.

Our approach is different from previously used methods in that it uses efficient graph algorithms and does not rely on restrictive assumptions about the structure, location and process of folding nuclei formation. Our predictions provide more detail concerning the protein folding pathway than most other methods in the literature. We demonstrate the success of our approach by predicting folding nuclei for a data set of proteins for which experimental kinetic data is available. We show that our method compares favorably with other methods in the literature and that its results agree with experimental results[1].

6.1 Description of the Problem

The Problem of identifying folding pathways (a set of native contacts that play an important role in folding along with identifying a time-ordered sequence of their

[1] A version of this chapter has been published. A. Shmygelska, (2005) Search for Folding Nuclei in Native Protein Structures, Bioinformatics, 2005, 21: i394-i402.

formation) of proteins can be formally defined as follows: Given a target backbone conformation (the native state) c represented as the set of pairwise native contacts $c = \{e_1, e_2, \ldots, e_m\}$ (where m is the total number of pairwise contacts present in the native state) and a factor $\delta \in (0, 1]$, find an ensemble of folding pathways $o = \{o_i \mid o_i \in O(c)\}$, where $O(c)$ is the set of all possible folding pathways for conformation c and each folding pathway $o_i \in O(c)$ consists of a sequence of pairwise contacts in relative order of their formation $o_i = o_{i,1}o_{i,2}\ldots o_{i,m}$ (each $o_{i,t}$ is a contact between two amino acids present in the native state that was formed in the folding pathway i as the t^{th} contact, $o_{i,t} \in \{e_1, e_2, \ldots, e_m\}$, $t \in [1, m]$). Each pathway o_i is a pathway satisfying the following condition: the value of an objective function $E(c, o_i)$ is within a factor δ of the minimal objective function value $E(c, o^*) = \min\{E(c, o_j) \mid o_j \in O(c)\}$, *i.e.* $E(c, o_i) \leq \delta E(c, o^*)$. Usually, the objective function only considers the entropy of contact formation, since the precise energetics of the protein folding process are still unknown.

The problem of identifying folding nuclei imposes an additional constraint on folding pathways o_i: In this case, each folding pathway o_i consists of a sequence of pairwise contacts in relative order of their formation $o_i = o_{i,1}o_{i,2}\ldots o_{i,k_i}$, such that when all k_i contacts have been formed in the pathway o_i, the process of folding results in the downhill rapid assembly of the native state. Thus, the objective function (energy) of adding contacts $o_{i,k_i+1}\ldots o_{i,k_m}$ is smaller than the specified threshold, *i. e.*, $E(c, o_{i,k_i+1}\ldots o_{i,k_m}) < E_{threshold}$.

6.2 Description of the Algorithm

The proposed algorithm for identifying folding nuclei (given the native state, determine the order of contact formation and identify those that are critical for the process of folding) is based on the notion of effective contact order (ECO) first introduced by Dill [144]. As described in Chapter 2, effective contact order of a newly added contact is defined as the effective loop closure size (the number of steps, *i.e.*, covalent and non-covalent links, taken along the shortest path on the polymer graph) given that other contacts have been formed.

Our approach consists of the following three steps, which we describe in detail in subsequent subsections: (1) construction of the polymer graph (a graph of contacts present in the native state of a protein); (2) sampling of the space of folding pathways to identify an ensemble of low cumulative effective contact order pathways; (3) analysis of the ensemble obtained in the previous step to extract contacts that belong to folding nuclei, and analysis of the order of their relative formation.

6.2.1 Generation of the Polymer Graph

The choice of model for a protein representation, discussed in Chapter 2, was guided by two somewhat opposing objectives: reduction of the search space and accurate geometry of protein structure. A protein is represented as a connected undirected weighted graph with one node per residue. Two residues are connected by an edge, if they are in contact in the native state, as described previously in Chapter 2. Every edge of the polymer graph is augmented by the value of ECO – loop closure size required to form a given contact or edge. Initially, we consider an extended polymer where no non-covalent contacts are present; thus the weight of any edge (i, j) is set to $CO(i, j) = |i - j|$. Figure 6.1 shows an example of the polymer graph generated for the chymotrypsin inhibitor 2 (CI2) protein from Table 6.1. Generally, such polymer graphs, as has been previously observed [51], are very well connected and exhibit a so-called small-world property, whereby most nodes in the graph are also neighbors of one another, and every node can be reached from every other node by a small number of hops or steps.

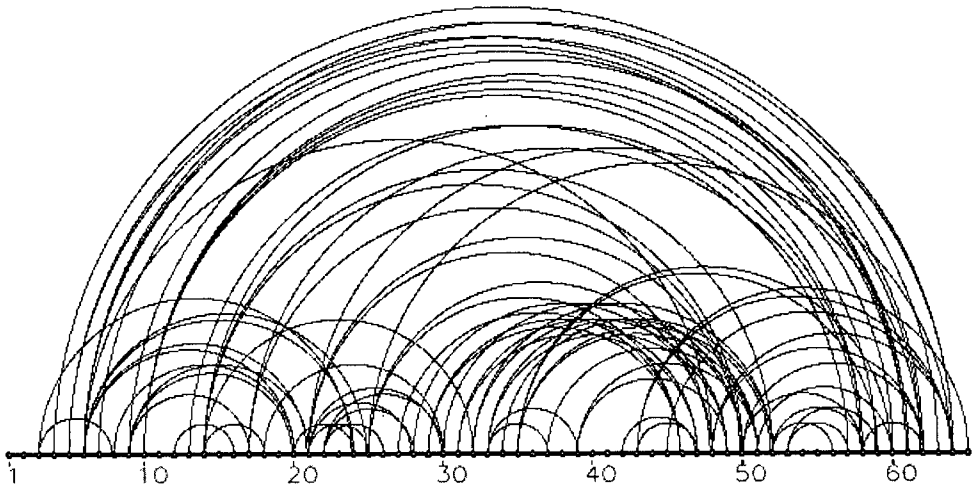


Figure 6.1: Polymer graph generated in phase one of our algorithm for the chymotrypsin inhibitor 2 (CI2) protein containing 65 nodes (n) and 82 edges (m).

6.2.2 Sampling of Folding Pathways

Since dominant protein folding pathways are those that maximize the formation of favourable native interactions while minimizing the loss of configurational entropy,

low effective order contacts are more readily made as compared to high effective contact order contacts during the process of folding. Thus, we are interested in finding an ensemble of folding pathways such that the cumulative effective contact order:

$$ECO_{total} = \sum_{i,j \in \{e_1,\ldots,e_m\}, i \leq j+3} ECO(i,j,A_t) \tag{6.1}$$

for all of the contacts added is minimized, where m is the total number of edges $\{e_1, ..., e_m\}$ in the polymer graph, and i, j iterate only through the set of non-covalent native contacts. As previously mentioned, individual values of $ECO(i,j,A_t)$ are affected by what other contacts have been formed previously – the set of previously-formed contacts A_t, $t \in [0, m-1]$, $A_0 = \emptyset$, and $A_{t+1} = A_t \cup \{(i,j)\}$. Therefore maximizing the cumulative effective contact order is not trivial.

To solve this problem, we use probabilistic constructive local search, as illustrated in Figure 6.2. Our algorithm, outlined in Figure 6.3, works as follows: we start with an extended polymer graph that represents a fully extended polymer chain; at each step we add one edge (i,j) (native contact) to the polymer graph probabilistically (with probability p) based on the effective contact order:

$$p = \frac{1}{ECO(i,j,A_t)^{3/2+k/m}}. \tag{6.2}$$

We used a power of $3/2$ based on the theory of the random Gaussian polymer, where the probability of a random contact is proportional to the separation along the chain raised to the power of $3/2$ [48]. This local search algorithm is quite "greedy" and will generally tend to pick edges with low ECO, following the intuition that during the process of folding, low-ECO contacts have a higher probability of formation. As non-covalent edges are added to the polymer graph, local search should become even greedier, since the probability that there will be subsequent edges whose ECO will depend on the edges just added will decrease. The goal here is to minimize the sum of ECO of individual contacts during their formation. Thus, we added an additional factor k/m to the power, where k is the number of non-covalent edges already added to the polymer graph, and m is the total number of non-covalent edges in the polymer graph. Thus, this factor reflects the proportion of non-covalent edges already added to the graph.

The weights of edges that are still left to be added are updated according to an efficient graph algorithm for the dynamic all pairs shortest path problem by Ramalingam and Reps [111]. This algorithm maintains information about the shortest paths by keeping a subtree of potentially affected nodes for each source – a destination pair containing all edges that belong to at least one shortest path from a source node to a destination node. Distances are updated by running a Dijkstra-like procedure on the affected nodes. The time complexity of this algorithm is

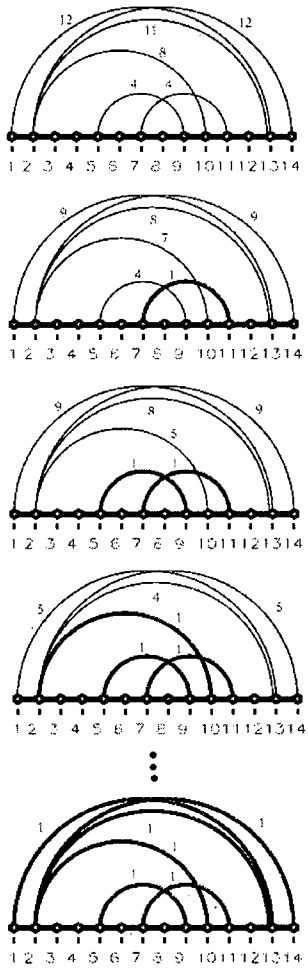


Figure 6.2: Illustration of the sampling phase of the algorithm. Edges that are added to the polymer graph are highlighted in bold, while edges that still have to be added are represented in non-bold. Initially, we start with a fully extended polymer, and edges that are to be added are weighted by their chain separation. At each step one edge is probabilistically added based on its ECO weight, and the ECO weights of the edges that are still to be added have to be revised at each step. The process of addition of edges stops when all of the edges are added to the polymer graph and their respective ECOs shrink to 1.

procedure *ProbabilisticConstructiveLocalSearch*
  input: polymer graph
  output: folding pathway with low ECO
  initialize weights of all edges (i, j) to $|i - j|$;
  add all edges with weight equal to 1 to the graph;
  k:= 0;
  while (not all edges added i.e., $k < m$) **do**
    select next edge (i, j) to add with probability $p = \frac{1}{ECO(i,j,A_t)^{3/2+k/m}}$;
    update weights for edges not yet added;
    k:=k+1;
  end
end

Figure 6.3: Outline for the probabilistic constructive local search used in the sampling of the folding pathways stage of our algorithm, where k is the number of non-covalent edges already added to the polymer graph and m is the total number of non-covalent edges in the polymer graph.

$O(mn + n^2 log(n))$, where n is the number of nodes in the graph and m is the number of edges [31].

Samples obtained and added to the ensemble of pathways that have low cumulative effective contact order should be independent of each other, thus we used the results of multiple independent runs of the constructive search to construct the low-ECO ensemble. Our simple sampling procedure is repeated multiple times until the number of low cumulative ECO pathways specified by the user is built. A particular low cumulative ECO pathway is added to the ensemble of low-ECO pathways if the cumulative effective contact order obtained is within 10% of the lowest ECO value obtained during the search.

6.2.3 Collective Analysis of Low Effective Contact Order Pathways

To identify the subset of contacts that belongs to the folding nuclei and to determine their relative formation order, we analyze a representative ensemble of a large number of low cumulative ECO pathways obtained during the previous sampling step. In order to obtain an upper bound on the subset of native contacts that can potentially participate in folding nuclei, we define a critical point in the folding pathway after which the process of subsequent contact formation becomes downhill according to our objective function of interest – the effective contact order. This critical point determines a moment in time when all folding nuclei have been

assembled. To define this critical point in terms of the effective contact order, we record the subset of native contacts (edges) that, as they are added, result in an effective contact order smaller than the given threshold for all edges still left to be added. The definition of the ECO threshold that determines when edges should no longer be added to the set of potential folding nuclei naturally follows from studies of loop lengths that have high probability of formation without involving intermediate steps [83]. We defined this threshold to be $\Delta l = 15$ residues, based on the observations in [83]. After the maximal effective contact order of edges that are left to be added drops below this specified threshold, we stop adding edges to the set of potential folding nuclei contacts.

As a result, we end up with the subset of native contacts that potentially belong to the set of folding nuclei contacts. In order to identify which of these edges are true folding nuclei contacts, we analyze relative dependencies among the edges as they were formed, that is, added to the polymer graph. We count the number of times each edge is used in the shortest path (effective contact order) events of the subsequent edges. We do so both directly *i.e.*, the edge belongs to the shortest path of another edge during growth of the polymer graph, and indirectly, the shortest path for an edge that is added during the growth process contains an edge whose shortest path directly or indirectly dependents on the current edge – a recursive step. When this recursive analysis of edge dependencies is completed, we can calculate the proportion of times an edge was used in the shortest-path events of other edges. The edges that are used most often are those that form early and are crucial for reducing the effective contact order (and entropy) for the subsequent edges. Thus, these edges comprise folding nuclei contacts. The outline for the analysis and identification of the folding nuclei phase of our algorithm is given in Figure 6.4.

6.3 Empirical Results and Discussion

To test our construction-based algorithm for identification of folding pathways, we used the set of 29 proteins listed in the paper by Rader *et al.* [107]. Only 27 were used since we had problems processing the pdb files of 1osp and 1hcb.

The output of our algorithm consists of the percent usage of each edge in the ensemble of low-ECO folding pathways. Those contacts that are used most often in the shortest path (low-ECO contact formation) represent folding nuclei, since they are crucial for subsequent native contact formation.

We display our results in two ways. The first one is more intuitive: we show the actual edges in the three-dimensional protein structures shaded according to usage in the shortest path events. The darker edges represent those contacts predicted to

```
procedure IdentifyFoldingNuclei
   input: the set of low-ECO pathways,
          the length threshold Δl
   output: the percent usage of each edge in low-ECO pathways
   for (each pathway in the low-ECO ensemble ) do
      set weights of all edges (i, j) to |i − j|;
      add all edges with weight equal to 1 to the graph;
      while (not all edges added) do
         add edge in order recorded;
         update ECO for the remaining edges;
         calculate maximum ECO among the remaining edges;
         if (maximum ECO > Δl) then
            for (each edge added) do
               get the list of edges that current edge depends on;
               recursively increment usage of the edges involved;
            end
         end
      end
   end
end
```

Figure 6.4: Outline of the analysis phase of our algorithm that identifies folding nuclei contacts in the low effective contact order pathway ensemble. Parameter Δl is the length threshold used to identify contacts whose formation proceeds rapidly.

fold earlier during the folding process – these are important for subsequent folding. Another way to represent edges is using the contact matrix of a protein shaded according to the order of contact formation. In both cases, contacts that are essential for low-ECO folding are clearly visible, and the overall folding mechanism can be deduced. The second approach is to display results using the "band" representation of a protein and to use shading to represent the usage of residues in the protein during low-ECO contact formation. We also use shading to indicate the order of becoming structured for individual residues in three-dimensional protein structures. To convert edge usage into residue usage, we sum usage numbers for each edge involving the residue of interest, and normalize it by the total usage of all amino acids in the low-ECO pathways. Using the second representation, we are able to compare our method with other methods in the literature, the majority of which produce the "band" representation of folding nuclei, even though descriptions of a folding nucleus in terms of contacts are more accurate.

First we compare our folding nuclei predictions with the experimental results

obtained from the hydrogen-deuterium slow exchange (H/X) method that was run on the set of 29 proteins by Li *et al.* and Rader *et al.*, as well as from experimental Φ-value analysis [93]. As mentioned in [107], the experimental folding core was defined in this case as the secondary structural elements (assigned using Database of Secondary Structure Assignments (DSSP)) containing residues with the largest protection factors in the H/X experiments, which indicates that these residues are structured early during the process of folding. Exchange rates are most often measured in the native structures (this does not directly represent folding events). For some proteins, exchange rates for partially folded species during the actual process of folding are available. We indicate these residues using darker shades in the experimental results, since these become structured earlier in the process of folding [76].

In Figures 6.5, 6.6, and 6.7 we show "band" representations of the previously mentioned set of 27 proteins. The top band is for the H/X experimental results [76], the middle band (if available) are the experimental Φ-values that represent the participation of a residue in the transition state based on the mutagenesis experiments [93], and the lower band represents our predictions, gray-scale-coded by the percent involvement of the residues in the low-ECO events. The results of our approach generally show good agreement with the experimental results.

Also using visual comparison of our predictions with Figure 3 from [108] with the FIRST rigid algorithm, the GNM fast mode peak and the GNM slow mode minima methods tested by Rader *et al.*, we found that the results from our method are in agreement with the previously mentioned methods and additionally provide information about the order of contact formation.

Next we examine case by case a few very well-studied proteins in our data set and compare our predictions with the experimental and computational data available about their folding pathways.

The chymotrypsin inhibitor 2 protein forms a four-stranded β-sheet packed against an α-helix, as illustrated in Figure 6.8. This is probably the best characterized protein using Φ-value analysis. Experimental Φ-values are high in the α-helix, with the highest values for the residues at the N-cap of the α-helix. These residues interact with two residues in the β-sheet to form a core, which is the most highly structured region of the protein in the transition state [93]. Our prediction captures the previously described experimental observations. Our gray-scale-coded contact matrix for CI2 (see Figure 6.8) also agrees with computational results of Weikl *et al.* establishing that α-β2, α-β1, β3-β4 form first, and β1-β4 forms after all of the local contacts have been formed.

Among the most difficult cases for our algorithm are the folding pathways of proteins G and L. Both protein G and protein L are members of the ubiquitin fold (while they have little sequence identity) and fold via helix-assisted hairpin forma-

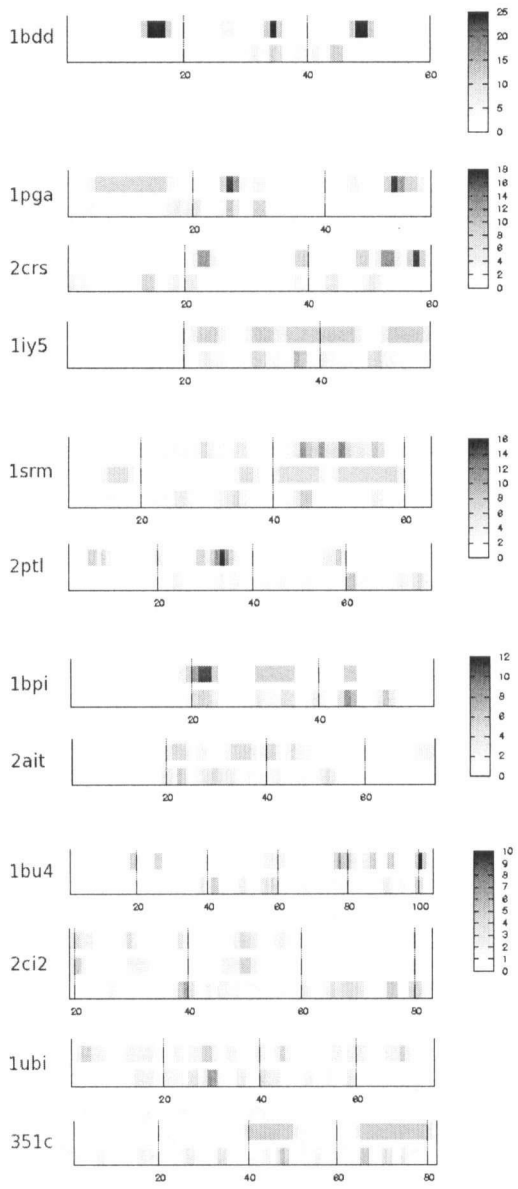


Figure 6.5: Band representation of the experimental and predicted folding nuclei for our test set of 27 proteins, Part I. The upper band represents H/X experimental data (darker-shaded residues represent those that gain protection first during folding, lighter-shaded residues represent residues that have slow exchange in the native state H/X [76]). The middle band (if available) represents experimental Φ-values (the darker the shade, the higher the Φ-value ($0 \leq \phi \leq 1$) [93]. The lower band represents our folding nuclei predictions, shaded according to percentage of involvement in low-ECO events (the exact scale in percent is given on the right side; proteins are grouped according to the maximum percentage of edge usage).

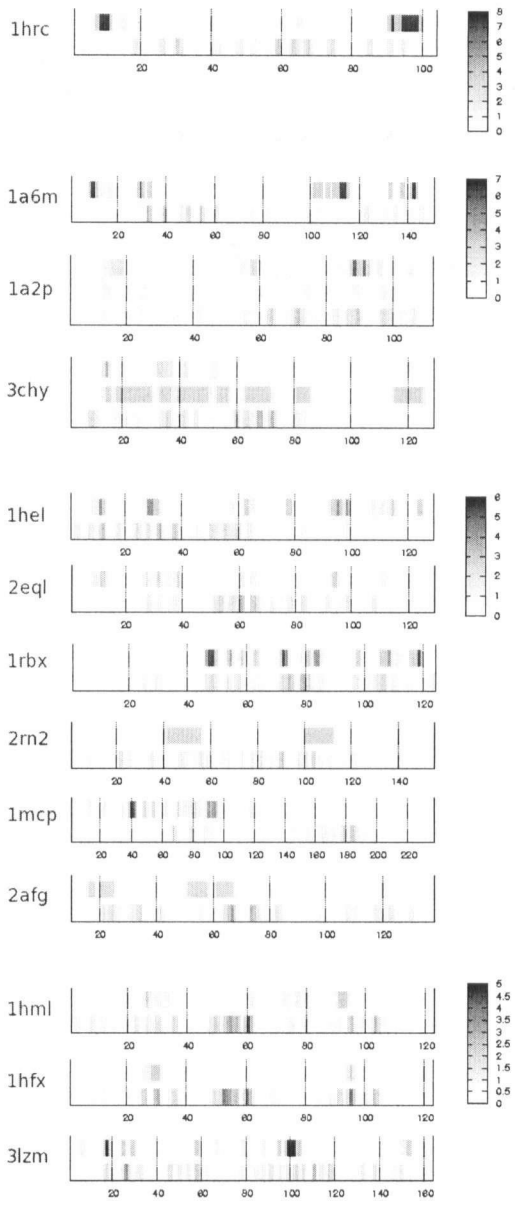


Figure 6.6: Band representation of the experimental and predicted folding nuclei for our test set of 27 proteins, Part II. The upper band represents H/X experimental data (darker-shaded residues represent those that gain protection first during folding, lighter-shaded residues represent residues that have slow exchange in the native state H/X [76]). The middle band (if available) represents experimental Φ-values (the darker the shade, the higher the Φ-value ($0 \leq \phi \leq 1$) [93]. The lower band represents our folding nuclei predictions, shaded according to percentage of involvement in low-ECO events (the exact scale in percent is given on the right side; proteins are grouped according to the maximum percentage of edge usage).

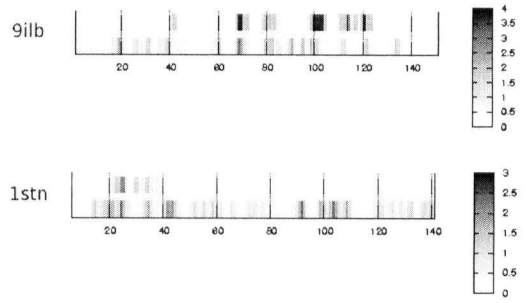


Figure 6.7: Band representation of the experimental and predicted folding nuclei for our test set of 27 proteins, Part III. The upper band represents H/X experimental data (darker-shaded residues represent those that gain protection first during folding, lighter-shaded residues represent residues that have slow exchange in the native state H/X [76]). The middle band (if available) represents experimental Φ-values (the darker the shade, the higher the Φ-value ($0 \leq \phi \leq 1$) [93]. The lower band represents our folding nuclei predictions, shaded according to percentage of involvement in low-ECO events (the exact scale in percent is given on the right side; proteins are grouped according to the maximum percentage of edge usage).

tion. Their topology consists of a central α-helix packed against a β-sheet composed of two anti-parallel β-hairpins, as shown in Figure 6.9. According to H/X experimental data, protein G folds through a transition-state ensemble with a well-formed β-hairpin 2 (formed by $\beta 3$ - $\beta 4$), while protein L forms β-hairpin 1 (formed by $\beta 1$ - $\beta 2$) first [76]. Our approach captures helix-assisted folding, but since the objective function only considers topological effects and ignores energetics, it does not correctly predict the hairpin formation order. The importance of energetics in the folding of proteins G and L has been shown previously in the literature [23].

The enzyme barnase folds into a five-stranded anti-parallel β-sheet and two α-helices, see Figure 6.10. The Φ-value analysis of barnase shows that major secondary structures are formed but the loops are unfolded [93]; this observation is consistent with our predicted results. It has also been shown experimentally (by H/X [139]) and in molecular dynamic simulations [28] that the folding pathway of barnase is dominated by β-sheet structures, particularly by the formation of the $\beta 3$-$\beta 4$ hairpin, and as seen from our predictions this hairpin forms early.

The chicken src SH3 domain folds into a five-stranded orthogonal β-sheet structure, see Figure 6.11. Both H/X [76] and Φ-value [93] experimental results show that the $\beta 2$-$\beta 3$ hairpin forms first. Our predictions are consistent with these results.

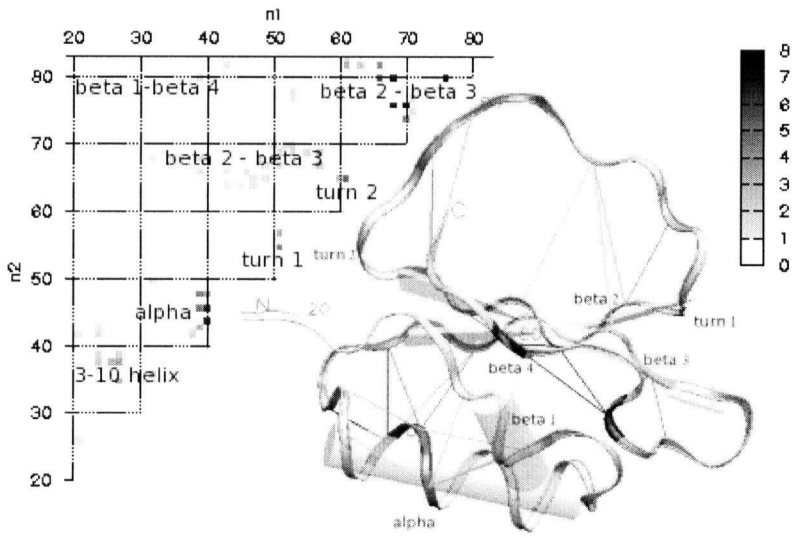


Figure 6.8: Predicted order of contact formation for the chymotrypsin inhibitor 2 protein (pdb id 2ci2). Contact matrix and three-dimensional structure representations are gray-scale-coded according to percent contact usage (the scale is given on the right). Indexes n_1 and n_2 represent indexes of the residues along the backbone. Darker-shaded edges (contacts) are predicted to form first.

The CheY protein is composed of five parallel β-sheets and five α-helices (with two α-helices, A and E, located on one face of the β-sheet and three other α-helices, B, C, and D, on the other side), see Figure 6.12. H/X protein stability is highest in the hydrophobic core formed by the side chains of the residues in β-strands 1, 2, and 3 and helices A and E. The α-helix A, together with β-strands 1 and 3, is involved in the formation of the folding nucleus [70]. Φ-value analysis shows that only residues in the first sub-domain (1 to 61) exhibit some degree of native-like structure, while the second sub-domain is essentially unstructured (residues 62-129). Our predictions correspond to experimental results: the first sub-domain, particularly β-strands 1, 2, and 3 as well as helices A, B, and C, forms early.

The ubiquitin protein (see Figure 6.13) forms a molten globule intermediate that has partially folded $\beta 1$ and $\beta 2$ strands of the five-strand β-sheet, part of the $\beta 3$ strand, and a partially structured α-helix packed on the hydrophobic side of the sheet (H/X experiments) [19]. Our predictions are consistent with these experimental observations.

The T4 lysozyme protein forms a molten globule that involves helices E, H,

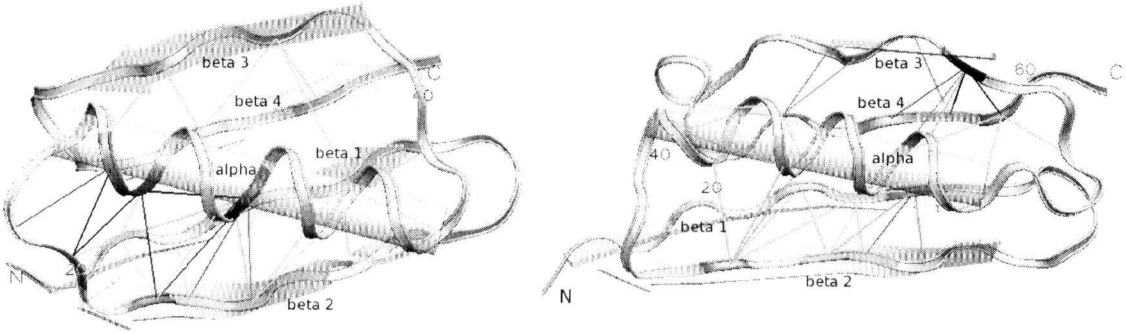


Figure 6.9: Predicted order of contact formation for the B1 immunoglobulin-binding domain of protein G (left) and protein L (right), (pdb ids 1pga and 2ptl). Darker shaded edges (contacts) are predicted to form first.

and A and the C-terminal of helix C (H/X analysis) [80], as shown in Figure 6.14. Our predictions also show that interactions between helices E, H and C-terminal of helix C are established early in the process of folding.

According to experimental H/X data, the bovine pancreatic trypsin inhibitor protein (see Figure 6.15) forms α-β1 and β2 - β3 structures that come together [76]. Our predictions are exactly supported by these experimental observations.

The B domain of protein A is composed of three helices, forming a three helix-bundle, as shown in Figure 6.16. It has been demonstrated experimentally, using H/X, that Protein A forms a marginally stable intermediate consisting of helices α2 and α3 [76]; according to our approach, these two helices should also come into contact with each other during the early stages of the folding process.

6.4 Summary

Our graph-theoretical construction method for identifying folding nuclei is in agreement with the experimental data obtained using Φ-value analysis and the H/X method for different structural classes of proteins for the data set used. It is able to conduct multiple runs of the probabilistic constructive search on sequences of significant lengths (sequences of length $54 - 237$ amino acids were used here). Since the average length of a protein domain (the part of the protein sequence that folds largely independently of the rest of the sequence) is between $200 - 300$ amino acids [92], our approach is an attractive tool for studying folding of individual domains in different structural classes of proteins.

Our method has a number of advantages compared with the methods used to

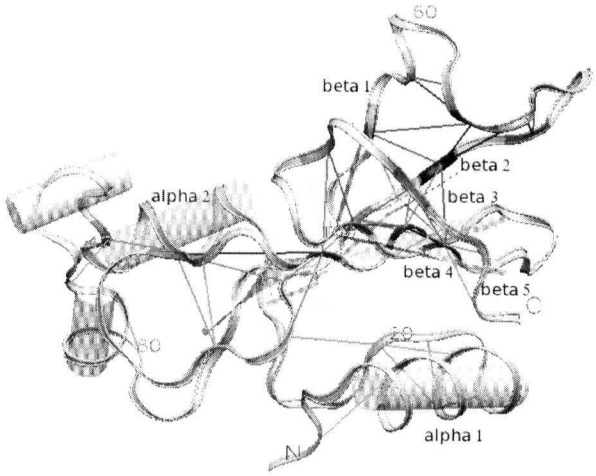


Figure 6.10: Predicted order of contact formation for the the barnase protein (pdb id 1a2p). Darker shaded edges (contacts) are predicted to form first.

address this problem in the literature (described in Chapter 3). In particular, it is conceptually simple; it has a reasonable and precise definition of a folding nucleus based on the entropy of the chain; and it places no restrictions on structure (as compared with clustering of native contacts and the use of low effective contact order search [144]), location or other properties (such as evolutionary conservation) of folding nuclei. Additionally, it provides information about the folding pathway, unlike other methods (for example, GNM [32], FIRST [107], and amino acid conservation within a family/super-family [112]) that provide only the set of residues predicted to participate in formation of folding nuclei. Unfortunately, performance data for some of the methods mentioned in Chapter 3 (for example, time-intensive molecular dynamic unfolding of the native structures [36], the probabilistic roadmap method [2], minimum cuts unfolding of native structures [151], amino acid conservation within a family/super-family [112]) are unavailable for the comprehensive data set [76, 107] chosen for this work. Therefore, we could not directly compare the performance of these methods and our new algorithm.

One shortcoming of our method is that, similarly to many other methods in the literature, it considers only native contacts, and even though it scales quite well with the length of the protein, there is a limitation on length.

Our method, while still having conceptual simplicity, can be extended to use more realistic energy functions that in addition will consider energetic contributions. It can thus be used as a tool for obtaining further insight into the process of folding.

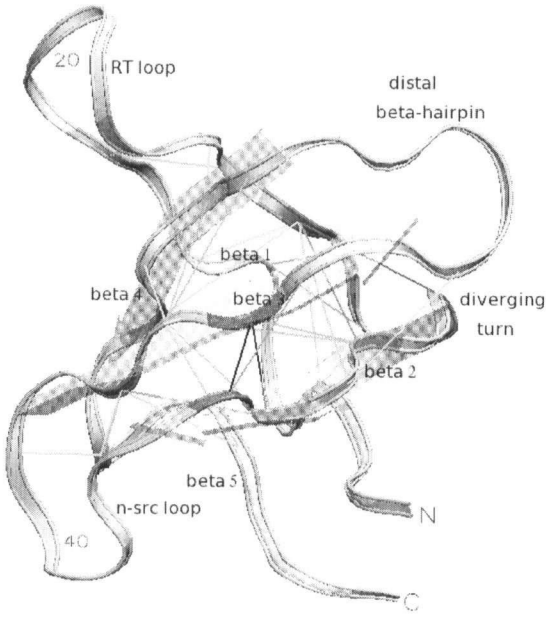


Figure 6.11: Predicted order of contact formation for the chicken src SH3 domain (pdb id 1srm). Darker shaded edges (contacts) are predicted to form first.

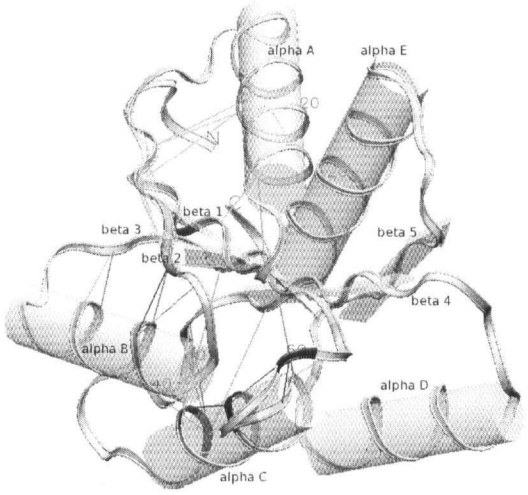


Figure 6.12: Predicted order of contact formation for the CheY protein (pdb id 3chy). Darker shaded edges (contacts) are predicted to form first.

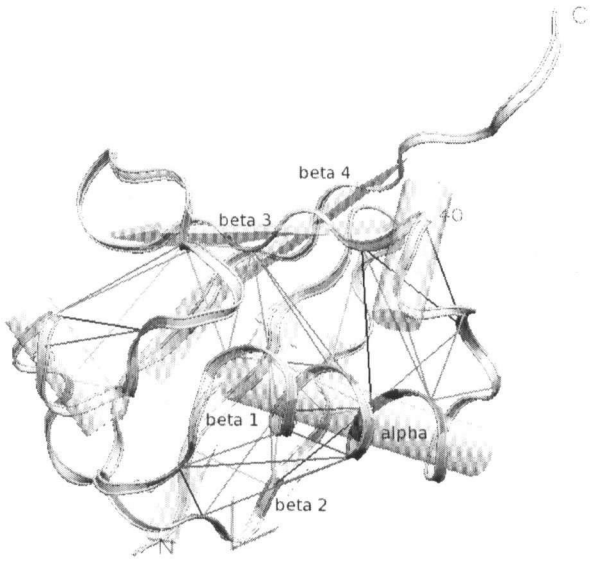


Figure 6.13: Predicted order of contact formation for the ubiquitin protein (pdb id 1ubi). Darker shaded edges (contacts) are predicted to form first.

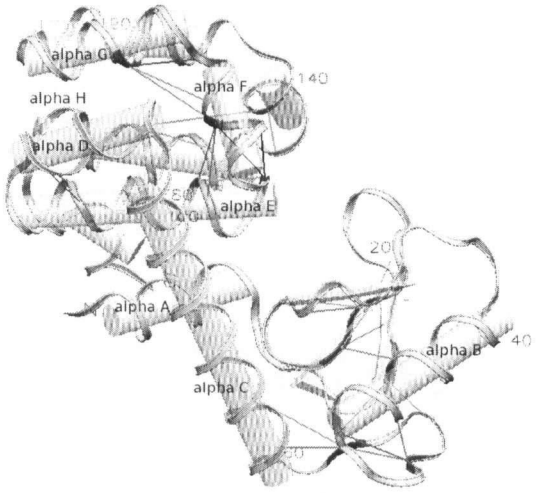


Figure 6.14: Predicted order of contact formation for the T4 lysozyme protein (pdb id 3lzm). Darker shaded edges (contacts) are predicted to form first.

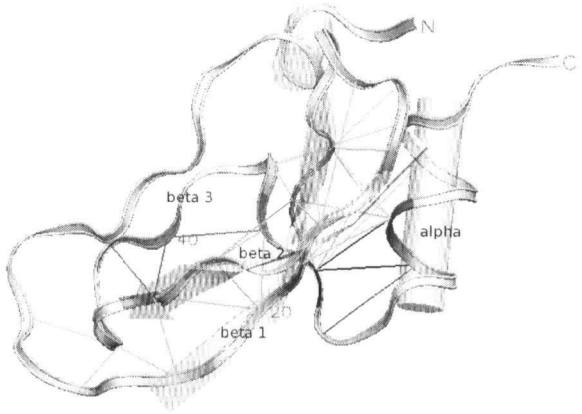


Figure 6.15: Predicted order of contact formation for the bovine pancreatic trypsin inhibitor protein (pdb id 1bpi). Darker shaded edges (contacts) are predicted to form first.

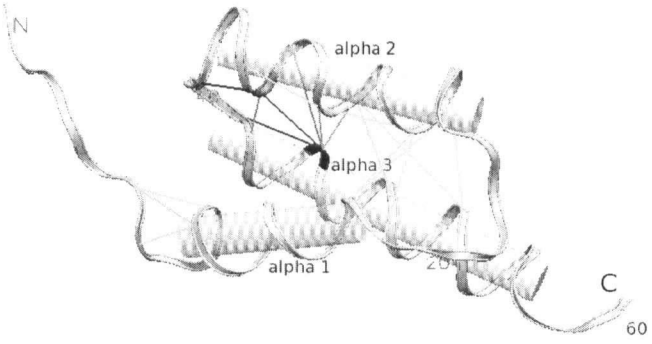


Figure 6.16: Predicted order of contact formation for the B domain of protein A (pdb id 1bdd). Darker shaded edges (contacts) are predicted to form first.

	Protein	PDB ID	length
1	Apo-myoglobin	1a6m	151
2	Barnase	1a2p	108
3	Cytochrome c	1hrc	104
4	T4 lysozyme	3lzm	164
5	Ribonuclease T1	1bu4	104
6	α-Lactalbumin	1hml	123
7	Chymotrypsin inhibitor 2	2ci2	64
8	Ubiquitin	1ubi	76
9	Bovine pancreatic trypsin inhibitor	1bpi	58
10	Interleukin-1β	9ilb	151
11	Hen egg-white lysozyme	1hel	129
12	Equine lysozyme	2eql	129
13	Protein A, B-domain	1bdd	60
14	Staphylococccal nuclease	1stn	136
15	Ribonuclease A	1rbx	124
16	Ribonuclease H	2rn2	155
17	Guinea pig α-lactalbumin	1hfx	123
18	B1 immunoglobulin-binding domain protein G	1pga	56
19	B1 immunoglobulin-binding domain protein L	2ptl	78
20	Cardiotoxin analog III	2crs	60
21	Tendamistat	2ait	74
22	Single chain antibody fragment	1mcp	237
23	Human acidic fibroblast growth factor-1	2afg	127
24	Cytochrome c551	351c	82
25	Ovomucoid third domain	1iy5	54
26	Chicken src SH3 domain	1srm	56
27	CheY	3chy	128

Table 6.1: Set of proteins used in this study. The kinetic data about folding pathways of these proteins is available in [76] and [107].

Chapter 7

Conclusions and Future Directions

May I never use my reason against the truth.

Hasidic prayer

In this chapter, we discuss how the approaches introduced in this thesis relate to each other and outline scientific contributions made towards developing more efficient search methods for protein folding problems. We also draw conclusions based on the observed performance of our new algorithms and outline directions for future research.

7.1 ACO for 2D and 3D Hydrophobic Polar Folding

In this work, we have shown that Ant Colony Optimization (ACO) can be applied in a rather straight forward way to the 2D and 3D HP protein folding problems. Even though our ACO algorithm is based on very simple structural components (single relative directions) and a simple subsidiary local search procedure (iterative first improvement), it performs fairly well compared with other algorithms and specialized heuristics on the benchmark instances considered here, particularly in 2D. The only non-specialized algorithm that typically performs better than our ACO algorithm, both in 2D and 3D, is PERM. We observed that, particularly in 3D, the run-time required by ACO for finding minimum (or best-known) energy conformations scales worse than PERM with sequence length. However, our results show that our ACO algorithm finds a different ensemble of native conformations compared to PERM, and has less difficulty folding sequences whose native states contain structural nuclei located in the middle rather than at the ends of a given sequence, as well as sequences with structures in which the ends interact. Because of its ability to find more balanced ensembles of minimum (or close to minimum) energy conformations, our new ACO algorithm can greatly facilitate investigations of the topology and location of structural nuclei based on the sequence alone (when

the native state of a protein is not necessarily known, which is the requirement for the graph-theoretical construction search introduced in Chapter 6).

We found that two major components of ACO — the pheromone values, which capture experience accumulated over multiple iterations of the search process from multiple conformations, as well as the heuristic information that provides myopic guidance to the folding process — play a significant role for longer 2D sequences and, to a lesser extent, for 3D sequences. This observation suggests that in 3D, it may be preferable to associate pheromone values with more complex solution components.

In future work, it will be interesting to investigate the use of more complex and informative solution components. These can be obtained, for example, based on reinforcement of contacts, similarly to the contact representation used for the graph-theoretical construction search for folding nuclei presented in Chapter 6. Another promising direction is to consider different heuristic functions that take into account certain properties of intermediate and complete conformations, similar to the more specialized HZ, CHCC, CG, and CI methods. It would also be worthwhile to consider the use of subsidiary local search procedures that are more powerful than the iterative first improvement algorithm used in this work (for example, the Bin Framework Monte Carlo method introduced in Chapter 5). Furthermore, we believe that better techniques for avoiding the attrition problem during the construction process (particularly in 2D) should be developed (for example, a look-ahead strategy could be used).

Finally, while HP protein folding problems are of considerable interest because of their conceptual simplicity, ultimately, most applications of protein folding algorithms require the use of more realistic models of protein structure. Our ACO algorithm does not rely on heuristics and properties that are specific to the HP model, yet it performs very well on this restrictive, but not entirely unrealistic abstract model. We therefore believe that relatively straight forward extension of our ACO algorithm to more complex and realistic models of protein structure, including other lattice and discrete off-lattice models, holds significant promise.

7.2 Adaptive Bin Framework for the FCC $\beta-$ Sheet Model

Our bin framework generalizes different adaptive strategies and provides a very rich framework for establishing interactions between the search process and the search landscape. The adaptive component of this framework is very important for the overall efficiency of the search. Here we introduce adaptive mechanisms for both choosing which conformations should be stored, based on the conformations

already stored in memory, and biasing preferences of retrieving particular conformations, based on the past history when the search stagnates. This is generally absent from other methods and therefore, even when running other search methods for an extensive time, the best solution quality found may not improve.

Since we are not simply performing a memory-less search by keeping only the current conformation or a set of conformations at different temperatures as current state-of-the-art methods for protein folding do, but keep the k best ones encountered at any time during the search (where k is determined by the diversity of a set), we can search multiple regions containing promising conformations and possibly sample compact conformations more efficiently.

Similar to the Model-based Search (MBS) method [20], our bin framework is used to store promising candidate solutions for future reuse. However, unlike MBS, we developed and tested an adaptive diversification mechanism that varies based on the energy level considered and takes into account how different a conformation is from other conformations of the same energy. Additionally, the energy level of interest, which determines what is the highest energy that conformations are allowed to have and still be memorized in the bin framework, and individual Hamming distance criteria used for each bin, are adapted according to the estimate of the ground state energy.

One of the important questions we address in this part of our work is: how do adaptive strategies perform for the complex energy landscape compared with heuristic methods that do not adapt based on the landscape being searched (such as Replica Exchange Monte Carlo and Parallel-hat Tempering) for the protein folding problem? As shown by our analysis of the empirical performance of our novel Bin Framework Monte Carlo method (a combination of the bin framework and a simple Monte Carlo algorithm), we were able to develop more efficient search methods that rely on the bin framework and use an adaptive component. These methods can outperform both canonical Monte Carlo, Replica Exchange Monte Carlo, and its heuristic variant Parallel-hat Tempering when it comes to the search for a global optimum, as shown for the FCC β-sheet protein folding model.

Some of the promising directions for future work that we are planning to consider include more advanced adaptive strategies that extract other features of the search landscape and determine what precisely is needed – either diversification or intensification – and adjust the search accordingly. Combination of other stochastic local search methods, such as Ant Colony Optimization introduced in Chapter 4 with the bin framework introduced here can be considered. Additionally, we are planning to generalize our bin framework further to work on partial as well as complete conformations, producing an efficient generalized framework that combines two distinct search strategies. Finally, we would like to extend our bin framework to address other protein folding models that use more complex energy potentials,

such as the FCC model, with a more general energy function and discrete off-lattice models.

7.3 Construction Search for Identification of Folding Pathways

Our graph-theoretical construction method for identifying folding pathways, while conceptually very simple, has been shown to produce results that agree with the experimental data available for folding nuclei as well as with a number of more complex computational methods for finding folding nuclei. The presented method has a number of advantages: it represents a natural way to bound and identify the subset of native contacts important for low-ECO folding pathways; it provides additional information on the order of native contact formation and their relative dependencies; it does not require, but can potentially be augmented by, experimental results; and it does not impose restrictions on the complete formation of certain elements (clusters) of structure.

Future work will include a more extensive comparison of different computational methods for finding folding nuclei in different structural classes of proteins as well as possible augmentation of the method with physically based energy functions, particularly those that consider not only entropic but also energetic contributions. It would also be interesting to incorporate insights gained from the folding nuclei study to design more efficient search methods for protein folding. Specifically, if certain regularities for a specific structural class of proteins are discovered through a computational study of folding pathways (for example, using the graph-theoretical method proposed here), they could be incorporated into search methods for protein folding such as those introduced in Chapters 4 and 5, in the form of initial probabilities during the search.

7.4 Summary

The protein folding problem is at the center of this thesis. Within this vast problem we focused on two major sub-problems, with the emphasis on the first: (1) conformational searching for the protein folding problem and (2) the problem of identifying folding nuclei. In the context of search problems associated with the process of protein folding, we explored promising search strategies that result in more efficient searching for problems with complex energy landscapes, such as protein folding and the closely related problem of identifying folding pathways.

We designed and evaluated the performance of our new algorithms using em-

pirical evaluation techniques. Additionally, we attempted to understand the behaviour and properties of our algorithms by studying the influence of various parameters on their empirical performance for problems of interest. Whenever possible, we also determined the theoretical guarantees an algorithm was retaining or giving up. In this thesis, we considered three classes of promising algorithms that have not been developed and studied previously for these problems:

1. Biologically inspired self-organizing algorithms based on the notion of *stigmergy*. The phenomenon of *stigmergy* arises in a multi-agent system, where a collection of agents modifies the environment. These changes in turn affect the decision process of each agent. A well-known algorithm in this category is Ant Colony Optimization that along with other approaches based on *stigmergy*, has not been previously implemented and tested for this problem. In this work, we developed and tested the Ant Colony Optimization method for the problem of protein folding under the widely-studied 2D and 3D HP models.

2. The second class of algorithms we have presented in this work explore reactive (adaptive) search, which reacts to the progress made and adjusts the search strategy accordingly. We introduced a new framework – the bin framework, which stores a diverse set of promising conformations encountered during the search. The storage and retrieval of solutions is guided by the search progress made, and provides the diversification or intensification of the search.

3. The third class of algorithms studied in this work is based on efficient repetitive construction. We developed a construction search method that relies on efficient, dynamic, graph-theoretical methods and avoids forming restrictive assumptions about the process of folding. The Ant Colony Optimization method introduced for the protein folding problem is also a construction-based search, in which partial solution components are reinforced indirectly by means of a pheromone matrix.

One of the questions that was not answered explicitly in this thesis is, which of the proposed novel heuristic methods is the most promising for protein folding and closely-related problems? As seen from our empirical evaluations of the proposed methods, usually construction-based methods (such as Ant Colony Optimization) benefit from further local search optimization on complete candidate solutions. Local search methods relying on complete candidate solutions can benefit from occasional restart with a promising initial solution, as we showed using our novel bin framework. This candidate solution can also come from an efficient

construction heuristic transferring the search into a different part of the landscape when diversification is required, or from the previously memorized solution that resulted from the search on a complete conformation. The latter was done in our bin framework, and is a more intensification-directed strategy.

Thus, the real question is: How can these strategies be combined to result into a more efficient adaptive search method? This is one of the primary directions planned for our future research – when the need for a significant diversification is detected, based on the search progress made, a construction-based search could improve performance; when intensification is preferred, an efficient search based on complete candidate solutions could be more promising.

As seen from the individual conclusions and proposed future work described previously for the novel methods introduced in this thesis, all three methods will benefit from future integration of strategies found to work well in the two other parts of this work. Specifically, Ant Colony Optimization (probabilistic constructive search) will benefit from incorporation of a more powerful local search on complete candidate solutions, such as the Bin Framework Monte Carlo method. Additionally, ACO can be further improved, particularly when used on more realistic protein models, incorporating insights that can be derived using an algorithm for identification of folding nuclei. In its turn, our Bin Framework Monte Carlo method can be enhanced by a more powerful diversification mechanism that uses a construction-based search, such as ACO. As in the case of ACO, it can also benefit from additional insights derived from the search for folding nuclei. Finally, the graph-theoretical search for folding nuclei can be extended not to rely on the knowledge of the native state, by employing efficient search methods for protein folding, such as ACO and BINMC.

In the course of this work, we have identified several promising general areas for future research. We have shown that biologically inspired, adaptive search models based on the concept of importance sampling, and construction-based search methods provide an alternative to the current algorithms used for these problems that are primarily based on Monte Carlo and generalized Monte Carlo methods. In the previous literature on protein folding, these directions were either not proposed at all (as in the case of the biologically inspired algorithms) or are proposed in a limited setting (as in the case of adaptive strategies and the focus on construction-based search). We have shown that our novel methods have the ability to search the landscape efficiently, and in some cases outperform known algorithms.

Ultimately, in our future work we plan to address the problem of efficient searching from the joint perspective of: (1) optimization of energy potentials to provide more guidance during the search; and (2) the development of search methods that can be undertaken in parallel with the process of energy optimization, to further address the limitations of existing methods. The ultimate goal of this two-

fold process of optimization, which considers the two most important components of the problem, is to be able to provide more accurate protein structure prediction using simplified models for protein folding that allow to search effectively.

Bibliography

[1] P. Aloy, A. Stark, C. Hadley, and R.B. Russel, "Predictions without Templates: New Folds, Secondary Structure, and Contacts in CASP5," Proteins: Structure, Function, and Genetics, Vol. 53, 2003, pp. 436-456.

[2] N.M. Amato and G. Song, "Using Motion Planning to Study Protein Folding Pathways," Journal of Computational Biology, Vol. 9, 2002, pp. 149-168.

[3] C. Andrieu, N. de Freitas, A. Doucet, and M.I. Jordan, "An Introduction to MCMC for Machine Learning," Machine Learning, Vol. 50, 2003, pp. 5-43.

[4] C. Anfinsen, "Principles that Govern the Folding of Protein Chains," Science, Vol. 181, 1973, pp. 223-230.

[5] R. Backofen and S. Will, "A Constraint-Based Approach to Structure Prediction for Simplified Protein Models that Outperforms Other Existing Methods," Proceedings of the XIX International Conference on Logic Programming, 2003, pp. 49-71.

[6] R. Backofen, S. Will, and P. Clote, "Algorithmic Approach to Quantifying the Hydrophobic Force Contribution in Protein Folding," R.B. Altman, A.K. Dunker, L. Hunter, T.E. Klein, Proceedings of the 5^{th} Pacific Symposium on Biocomputing, 2000, pp. 92-103.

[7] Z. Bagci, R. L. Jernigan, and I. Bahar, "Residue Coordination in Proteins Conforms to the Closest Packing of Spheres," Polymer, Vol. 43, 2002, pp. 451-459.

[8] U. Bastolla, H. Fravenkron, E. Gestner, P. Grassberger, and W. Nadler, "Testing New Monte Carlo Algorithm for the Protein Folding Problem," Proteins: Structure, Function, and Genetics, Vol. 32, No. 1, 1998, pp. 52-66.

[9] R. Battiti and G. Tecchiolli, "The Reactive Tabu Search," ORSA Journal on Computing, Vol. 6, No. 2, 1994, pp. 126-140.

[10] B.A. Berg and T. Neuhaus, "Multicanonical Ensemble: A New Approach to Simulate First-Order Phase Transitions," Physical Review Letters, Vol. 68, No. 1, 1992, pp. 9-12.

[11] B.J. Berne and J.E. Straub, "Novel Methods of Sampling Phase Space in the Simulation of Biological Systems," Current Opinion in Structural Biology, Vol. 7, 1997, pp. 181-189.

[12] F.C. Bernstein, T.F. Koetzle, G.J.B. Williams, E.F. Meyer Jr.,M.D. Brice, J.R. Rodgers, O. Kennard, T. Shimanouchi, and M. Tasumi, "The Protein Data Bank: A Computer-based Archival File for Macromolecular Structures," Journal of Molecular Biology, Vol. 112, 1977, pp. 535-542.

[13] T. Beutler and K. Dill, "A Fast Conformational Search Strategy for Finding Low Energy Structures of Model Proteins," Protein Science, Vol. 5, 1996, pp. 2037-2043.

[14] E. Bonabeau, M. Dorigo, and G. Theraulaz, "Swarm Intelligence: From Natural to Artificial Systems," Oxford University Press, 1999.

[15] R. Bonneau and D. Baker, "Ab Initio Protein Structure Prediction: Progress and Prospects," Annual Review of Biophysics and Biomolecular Structure, Vol. 30, 2001, pp. 173-189.

[16] R. Bonneau, C.E.M. Strauss, C.A. Rohl, D. Chivian, P. Bradley, L. Malmstorm, T. Robertson, and D. Baker, "*De Novo* Prediction of Three-Dimensional Structures for Major Protein Families," Journal of Molecular Biology, Vol. 322, 2002, pp. 65-78.

[17] P.E. Bourne and H. Weissig "Structural Bioinformatics," John Wiley & Sons Publishing, NJ, USA 2003, p. 17.

[18] J.U. Bowie and D. Eisenberg, "An Evolutionary Approach to Folding Small α-helical Proteins that Use Sequence Information and an Empirical Guiding Fitness Function," Proceedings of the National Academy of Sciences of the USA, Vol. 91, No. 10, 1994, pp. 4436-4440.

[19] M.S. Briggs and H. Roder, "Early Hydrogen-Bonding Events in the Folding Reaction of Ubiquitin," Proceedings of National Academy of Sciences of the USA, Vol. 89, 1992, pp. 2017-2021.

[20] T.J. Brunette and O. Brock, "Improving Protein Structure Prediction with Model-based Search," Bioinformatics, Vol. 21 (Suppl. 1), 2005, pp. i66-i74.

[21] H.S. Chan and K.A. Dill, "Protein Folding in the Landscape Perspective: Chevron Plots and Non-Arrhenius Kinetics," Proteins: Structure, Function, and Genetics, Vol. 30, 1998, pp. 2-33.

[22] H.S. Chan and K.A. Dill, "The Protein Folding Problem," Physics Today, Vol. 46, 1993, pp. 24-32.

[23] M.S. Cheung, L.L. Chavez, and J.N. Onuchic, "The Energy Landscape for Protein Folding and Possible Connections to Function," Polymer, Vol. 45, 2004, pp. 547-555.

[24] G. Chikenji, M. Kikuchi, and Y. Iba, "Multi-Self-Overlap Ensemble for Protein Folding: Ground State Search and Thermodynamics,"' Condensed Materials, Vol. 27, 1999.

[25] T. Chiu and R.A. Goldstein, "Optimizing Energy Potentials for Success in Protein Tertiary Structure Prediction," Folding and Design, Vol. 3, 1998, pp. 223-228.

[26] C. Clementi, A.E. Garcia, and J.N. Onuchic, "Interplay among Tertiary Contacts, Secondary Structure Formation and Side-Chain Packing in the Protein Folding Mechanism: An All-Atom Representation Study," Journal of Molecular Biology, Vol. 326, 2003, pp. 933-954.

[27] T.E. Creighton, "Protein Folding," W. H. Freeman and Company, NY, USA, 1992 pp. 1-55.

[28] V. Daggett and A. Fersht, "The Present View of the Mechanism of Protein Folding," Nature, Vol. 4, 2003,pp. 497-502.

[29] T. Dandekar and P. Argos, "Folding the Main Chain of Small Proteins with the Genetic Algorithm," Journal of Molecular Biology, Vol. 236, 1994, pp. 844-861.

[30] P.M.C. de Oliveira, T.J.P. Penna, and H.J. Herrmann, "Broad Histogram Method," Brazilian Journal Of Physics, Vol. 26, 1996, 677-706.

[31] C. Demetrescu, S. Emiliozzi, and G.F. Italiano, "Experimental Analysis of Dynamic All Pairs Shortest Path Algorithms," Technical Report ALCOM-FT, ALCOMFT-TR-03-9, Vol. 8, 2003, pp. 1571-1591.

[32] M.C. Demirel, A.R. Atilgan, R.L. Jernigan, B. Erman, and I. Bahar, "Identification of Kinetically Hot Residues in Proteins," Protein Science, Vol. 7, 1998, pp. 2522-2532.

[33] K.A. Dill, K.M. Fiebig, and H.S. Chan, "Cooperativity in Protein-Folding Kinetics," Proceedings of National Academy of Sciences of the USA, Vol. 90, 1993, pp. 1942-1946.

[34] A.R. Dinner and M. Karplus, "The Role of Stability and Contact Order in Determining Protein Folding Rates," Nature Structural Biology, Vol. 8, 2001, pp. 21-22.

[35] C.M. Dobson, "Protein Folding and Misfolding," Nature, Vol. 426, 2003, pp. 884-890.

[36] N. Dokholyan, S. Buldyrev, H. Stanley, and E. Shakhnovich, "Identifying the Protein Folding Nucleus Using Molecular Dynamics," Journal of Molecular Biology, Vol. 296, 2000, pp. 1183-1188.

[37] M. Dorigo and G. Di Caro, "New Ideas in Optimization," In New Ideas in Optimization, eds. D. Corne, M. Dorigo, F. Glover, McGraw-Hill, 1999, pp. 11-32.

[38] M. Dorigo, G. Di Caro, and L.M. Gambardella, "Ant Algorithms for Discrete Optimization," Artificial Life, Vol. 5, No. 2, 1999, pp. 137-172.

[39] M. Dorigo and L.M. Gambardella, "Ant Colonies for the Traveling Salesman Problem," Biosystems, Vol. 43, 1997, pp. 73-81.

[40] M. Dorigo, V. Maniezzo, and A. Colorni, "Positive Feedback as a Search Strategy," Technical Report, pp. 91-016, Dip. Electronica, Politecnico di Milano, Italy, 1991.

[41] M. Dorigo, V. Maniezzo, and A. Colorni, "The Ant System: Optimization by a Colony of Cooperating Agents," IEEE Transactions on Systems, Man, and Cybernetics-Part B, Vol. 26, No. 1, 1996, pp. 29-41.

[42] M. Dorigo and T. Stützle, "Ant Colony Optimization," The MIT Press, 2004.

[43] R.L. Dunbrack, "Rotamer Libraries in the 21st Century," Current Opinion in Structural Biology, Vol. 12, 2002, pp. 431-440.

[44] A. Elofsson, S.M. Le Grand, and D. Eisenberg, "Local Moves: An Efficient Algorithm for Simulation of Protein Folding," Proteins: Structure, Function, and Genetics, Vol. 23, 1995, pp. 73-82.

[45] M. Feig, J. Karanicolas, and C.L. Brooks III, "MMTSB Tool Set: Enhanced Sampling and Multiscale Modeling Methods for Applications in Structural Biology," Journal of Molecular Graphics and Modeling, Vol. 22, 2004, pp. 377-395.

[46] A.R. Fersht, A. Matouschek, and L. Serrano, "The Folding of an Enzyme : I. Theory of Protein Engineering Analysis of Stability and Pathway of Protein Folding," Journal of Molecular Biology, Vol. 224, 1992, pp. 771-782.

[47] A.V. Finkelstein and A.Y. Badretdinov, "Rate of Protein Folding near the Point of Thermodynamic Equilibrium between the Coil and the Most Stable Chain Fold," Folding Design, Vol. 2, 1997, pp. 115-121.

[48] P.J. Flory, "Principles of Polymer Chemistry," Ithaca, New York: Cornell University Press, 1953, pp. 402-413.

[49] S.O. Garbuzynskiy, A.V. Finkelstein, and O.V. Galzitskaya, "Outlining Folding Nuclei in Globular Proteins," Journal of Molecular Biology, Vol. 336, 2004, pp. 509-525.

[50] F. Glover and M. Laguna, "Tabu Search," Kluwer, Norwell, MA, 1997.

[51] L.H. Greene and V.A. Higman, "Uncovering Network Systems within Protein Structures," Journal of Molecular Biology, Vol. 334, 2003, pp. 781-791.

[52] D. Gront, A. Kolinski, and J. Skolnick, "Comparison of Three Monte Carlo Conformational Search Strategies for a Protein-like Homopolymer Model: Folding Thermodynamics and Identification of Low-Energy Structures," Journal of Chemical Physics, Vol. 13, No. 12, 2000, pp. 5065-5071.

[53] U.H.E. Hansmann, "Protein Folding Simulations in a Deformed Energy Landscape," The European Physical Journal B, Vol. 12, 1999, pp. 607-611.

[54] U.H.E. Hansmann, "Simulated Annealing with Tsallis Weights – A Numerical Comparison," Physica A, Vol. 242, 1997, pp. 250-257.

[55] M. Hao and H.A. Scheraga, "Designing Potential Energy Functions for Protein Folding," Current Opinion in Structural Biology, Vol. 9, 1999, pp. 184-188.

[56] T.X. Hoang, F. Seno, J.R. Banavar, M. Cieplak, and A. Maritan, "Assembly of Protein Tertiary Structures from Secondary Structures Using Optimized Potentials," Proteins: Structure, Function, and Genetics, Vol. 52, 2003, pp. 155-165.

[57] J. Holland, "Adaptation in Natural and Artificial Systems," University of Michigan Press, Ann Arbor, MI, 1975.

[58] H.H. Hoos and T. Stützle, "On the Empirical Evaluation of Las Vegas Algorithms," Proceedings of the 14th Conference on Uncertainty in Artificial Intelligence, Morgan Kaufmann Publishers, 1998, pp. 238-245.

[59] H.H. Hoos and T. Stützle, "Stochastic Local Search: Foundations and Applications," Morgan Kaufmann Publishers / Elsevier, 2004.

[60] H.P. Hsu, V. Mehra, W. Nadler, and P. Grassberger, "Growth Algorithm for Lattice Heteropolymers at Low Temperatures," Journal of Chemical Physics, Vol. 118, 2003, pp. 444-451.

[61] H.P. Hsu, V. Mehra, W. Nadler, and P. Grassberger, "Growth-based Optimisation Algorithm for Lattice Heteropolymers," Physical Review E, Vol. 68, 2003, pp. 021113-1 – 021113-4.

[62] A. Irbäck, "Dynamic-Parameter Algorithms for Protein Folding," Monte Carlo Approach to Biopolymers and Protein Folding, eds. P. Grassberger, G.T. Barkema and W. Nadler, World Scientific, Singapore, 1998, pp. 98-109.

[63] D.T. Jones, "Predicting Novel Protein Folds by Using FRAGFOLD," Proteins: Structure, Function, and Genetics Suppl., Vol. 5, 2001, pp. 127-132.

[64] W.L. Jorgensen and J. Tirado-Rives, "The OPLS Potential Functions for Proteins. Energy Minimizations for Crystals of Cyclic Peptides and Crambin," Journal of American Chemistry Society, Vol. 110, 1988, pp. 1657-1666.

[65] D. Klimov and D. Thirumalai, "Lattice Models for Proteins Reveal Multiple Folding Nuclei for Nucleation-Collapse Mechanism," Journal of Molecular Biology, Vol. 282, 1998, pp. 471-492.

[66] A. Kolinski and J. Skolnick, "Reduced Models of Proteins and their Applications," Polymer, Vol. 45, 2004, pp. 511–524.

[67] R. Konig and T. Dandekar, "Improving Genetic Algorithms for Protein Folding Simulations by Systematic Crossover," Biosystems, Vol. 50, 1999, pp. 17–25.

[68] N. Krasnogor, W.E. Hart. J. Smith, and D.A. Pelta, "Protein Structure Prediction with Evolutionary Algorithms," Proceedings of the Genetic and Evolutionary Computation Conference, 1999.

[69] N. Krasnogor, D. Pelta, P. M. Lopez, P. Mocciola, and E. de la Canal, "Genetic Algorithms for the Protein Folding Problem: A Critical View," In C.F.E. Alpaydin, ed., Proceedings of Engineering of Intelligent Systems. ICSC Academic Press, 1998.

[70] E. Lacroix, M. Bruix, E. Lopez-Hernandez, L. Serrano, and M. Rico, "Amide Hydrogen Exchange and Internal Dynamics the Chemotactic Protein CheY from *Escherichia coli*," Journal of Molecular Biology, Vol. 271, 1997, pp. 472-487.

[71] D.P. Landau and K. Binder, "A Guide to Monte Carlo Simulations in Statistical Physics," New York, Academic Press.

[72] K.F. Lau and K.A. Dill, "A Lattice Statistical Mechanics Model of the Conformation and Sequence Space of Proteins," Macromolecules, Vol. 22, 1989, pp. 3986–3997.

[73] N. Lesh, M. Mitzenmacher, and S. Whitesides, "A Complete and Effective Move Set for Simplified Protein Folding," International Conference on Research in Computational Molecular Biology (RECOMB), 2003, pp. 188-195.

[74] A.M. Lesk, "Introduction to Protein Architecture," Oxford University Press, Oxford, 2001.

[75] M. Levitt, "A Simplified Representation of Protein Conformations for Rapid Simulation of Protein Folding," Journal of Molecular Biology, Vol. 104, 1976, pp. 59-107.

[76] R. Li and C. Woodward, "The Hydrogen Exchange Core and Protein Folding," Protein Science, Vol. 8, 1999, pp. 1571-1591.

[77] F. Liang and W.H. Wong, "Evolutionary Monte Carlo for Protein Folding Simulations," Journal of Chemical Physics, Vol. 115, No. 7, 2001, pp. 3374–3380.

[78] A.R. Lima, P.M.C. de Oliveira, and T.J.P. Penna, "A Comparison Between Broad Histogram and Multicanonical Methods," Condensed Materials Archive, No. 0002176v1, 2000.

[79] J.S. Liu, "Monte Carlo Strategies in Scientific Computing," Springer, 2001.

[80] M. Llinas, B. Gillespie, F.W. Dahlquist, and S. Marqusee, "The Energetics of T4 Lysozyme Reveal a Hierarchy of Conformations," Nature Structural Biology, Vol. 6, No. 11, 1999, pp. 1072-1078.

[81] A.D. MacKerrel, D. Bashford, M. Bellott, R.L. Dunbrack, J.D. Evanseck, M.J. Field, S. Fisher et al., "All-atom empirical potential for molecular modeling and dynamics studies of proteins," Journal of Physical Chemistry B, Vol. 102, 1998, pp. 3586-3616.

[82] N. Metropolis, A.W. Rosenbluth, M.N. Rosenbluth, and A.H. Teller, "Equation of State Calculations by Fast Computer Machines," Journal of Chemical Physics, Vol. 21, 1953, pp. 1087-1092.

[83] E. Michalsky, A. Goede, and R. Preissner, "Loops in Proteins (LIP) a Comprehensive Loop Database for Homology Modeling," Protein Engineering, Vol. 16, 2003, pp. 979-985.

[84] L. Mirny and E. Shakhnovich, "Protein Folding Theory: From Lattice to All-Atom Models," Annual Review in Biophysics and Biomolecular Structure, Vol. 30, 2001, pp. 361-396.

[85] L. Mirny and E. Shakhnovich, "Universally conserved residues in protein folds. Reading evolutionary signals about protein function, stability and folding kinetics," Journal of Molecular Biology, Vol. 291, 1999, pp. 177-196.

[86] A. Mitsutake, Y. Sugita, and Y. Okamoto, "Generalized-Ensemble Algorithms for Molecular Simulations of Biopolymers," Biopolymers (Peptide Science), Vol. 60, 2001, pp. 96-123.

[87] S. Miyazava and R. Jernigan, "Residue-Residue Potentials with a Favorable Contact Pair Term and Unfavorable High Packing Density Term for Simulation and Threading," Journal of Molecular Biology, Vol. 256, 1996, pp. 623-644.

[88] A. Monge, R.A. Friesner, and B. Honig, "An Algorithm to Generate Low-Resolution Protein Tertiary Structures from Knowledge of Secondary Structure," Proceedings of the National Academy of Sciences of the USA, Vol. 91, No. 11, 1994, pp. 5027-5029.

[89] J. Moult, K. Fidelis, A. Zemla, and T. Hubbard, "Critical Assessment of Methods of Protein Structure Prediction (CASP): Round IV," Proteins: Structure, Function, and Genetics Suppl., Vol. 5, 2001, pp. 2-7.

[90] T. Nandi, C.B. Rao, and S. Ramachandran, "Comparative Genomics Using Data Mining Tools," Journal of Bioscience, Vol. 27, No. 1, 2002, pp. 15-25.

[91] J.T. Ngo, J. Marks, and M. Karplus, "Computational Complexity: Protein Structure Prediction and the Levinthal Paradox," Protein Engineering, Vol. 5, No. 4, 1992, pp. 313-321.

[92] B. Nölting, "Protein Folding Kinetics," Springer, 1993.

[93] B. Nölting and K. Andert, "Mechanism of protein folding," Proteins, Vol. 41, 2000, pp. 288-298.

[94] Y. Okamoto, "Generalized-Ensemble Algorithms: Enhanced Sampling Techniques for Monte Carlo and Molecular Dynamic Simulations," Journal of Molecular Graphics and Modeling, Vol. 22, 2004, pp. 425-439.

[95] A.R. Ortiz, A. Kolinski, P. Rotkiewicz, B. Ilkowski, and J. Skolnick, "Ab Initio Folding of Proteins Using Restrains Derived from Evolutionary Information," Proteins: Structure, Function, and Genetics Suppl., Vol. 3, 1999, pp. 177-185.

[96] E.M. O'Toole and A.Z. Panagiotopoulos, "Monte Carlo Simulation of Folding Transitions of Simple Model Proteins Using a Chain Growth Algorithm," Journal of Chemical Physics, Vol. 97, No. 11, 1992, pp. 8644-8652.

[97] E. Paci, M. Vendruscolo, C.M. Dobson, and M. Karplus, "Determination of a Transition State at Atomic Resolution from Protein Engineering Data," Journal of Molecular Biology, Vol. 324, 2002, pp. 151-163.

[98] B.H. Park, E.S. Huang, and M. Levitt, "Factors Affecting the Ability of Energy Functions to Discriminate Correct from Incorrect Folds," Journal of Molecular Biology, Vol. 266, 1997, pp. 831-846.

[99] B.H. Park and M. Levitt, "Energy Functions that Discriminate X-ray and Near-Native Folds from Well-Constructed Decoys," Journal of Molecular Biology, Vol. 258, 1996, pp. 367-392.

[100] B.H. Park and M. Levitt, "The Complexity and Accuracy of Discrete State Models of Protein Structure," Journal of Molecular Biology, Vol. 249, 1995, pp. 493-507.

[101] A. Parkes and J.P. Walser, "Tuning Local Search for Satisfiability Testing," Proceedings of the Applications of Artificial Intelligence Conference, MIT Press, 1996, pp. 356-362.

[102] A.W.P. III. Patton and E. Goldman, "A Standard GA Approach to Native Protein Conformation Prediction," In Proceedings of the 6^{th} International Conference on Genetic Algorithms, Morgan Kauffman, 1995, pp. 574–581.

[103] J.T. Pedersen and J. Moult, "Protein Folding Simulations with Genetic Algorithms and a Detailed Molecular Description," Journal of Molecular Biology, Vol. 269, 1997, pp. 240-259.

[104] K. Plaxco, K. Simons, and D. Baker, "Contact Order, Transition State Placement and the Refolding Rates of Single Domain Proteins," Journal of Molecular Biology, Vol. 277, 1998, pp. 985-994.

[105] S.S. Plotkin and J.N. Onuchic, "Understanding Protein Folding with Energy Landscape Theory Part I: Basic Concepts," Quarterly Reviews of Biophysics, Vol. 35, No. 2, 2002, pp. 111-167.

[106] P. Pokarowski, A. Kolinski, and J. Skolnick, "A Minimal Physically Realistic Protein-Like Lattice Model: Designing and Energy Landscape that Ensures All-Or-None Folding to a Unique Native State," Biophysical Journal, Vol. 84, 2003, pp. 1518-1526.

[107] A.J. Rader and I. Bahar, "Folding Core Predictions from Network Models of Proteins," Polymer, Vol. 45, 2004, pp. 659-668.

[108] A.J. Rader, B.M. Hespenheide, L.A. Kuhn, and M.F. Thorpe, "Protein Unfolding: Rigidity lost," Proceedings of National Academy of Sciences of the USA, Vol. 99, 2002, pp. 3540-3545.

[109] G. Ramachandran and V. Sasisekharan, "Conformation of Polypeptides and Proteins," Advances in Protein Chemistry, Vol. 23, 1968, pp. 283-437.

[110] R. Ramakrishnan, B. Ramachandran, and J.F. Pekny, "A Dynamic Monte Carlo Algorithm for Exploration of Dense Conformational Spaces in Heteropolymers," Journal of Chemical Physics, Vol. 106, No. 6, 1997, pp. 2418-2424.

[111] G. Ramalingam and T. Reps, "An Incremental Algorithm for a Generalization of the Shortest-Path Problem," Technical Report TR-1087, Computer Sciences Department, University of Wisconsin, Madison, WI, 1992, pp. 1-26.

[112] B.V. Reddy, W.W. Li, I.N. Shindyalow, and P.E. Bourne, "Conserved Key Amino Acid Positions (CKAAPs) Derived from the Analysis of Common Substructures in Proteins," Proteins, Vol. 42, 2001, pp. 148-163.

[113] M.J. Rooman, J.A. Kocher, and S.J. Wodak, "Prediction of Protein Backbone Conformation Based on Seven Structure Assignments," Journal of Molecular Biology, Vol. 221, 1991, pp. 961-979.

[114] A. Sali, E. Shakhnovich, and M. Karplus, "Kinetics of Protein Folding – A Lattice Model Study of the Requirements for Folding to the Native State," Journal of Molecular Biology, Vol. 235, 1994, pp. 1614-1636.

[115] A. Sali, E. Shakhnovich, and M. Karplus, "How Does a Protein Fold?" Nature, Vol. 369, 1994, pp. 248-251.

[116] R. Sayle and E.J. Milner-White, "RASMOL - Biomolecular Graphics for All," Trends in Biochemical Science, Vol. 20, No. 9, 1995, pp. 374-376.

[117] G.E. Schultz and H.R. Schirmer, "Principles of Protein Structure," Springer Verlag, NY, 1979.

[118] A. Shmygelska, "Search for Folding Nuclei in Native Protein Structures," Bioinformatics, Vol. 21, 2005, pp. i394 - i402.

[119] A. Shmygelska, R. Hernandez, and H.H. Hoos, "An Ant Colony Algorithm for the 2D HP Protein Folding Problem," Third International Workshop, ANTS 2002, Proceedings. Springer's Lecture Notes in Computer Science (LNCS) series, Vol. 2463, pp. 40-53.

[120] A. Shmygelska and H.H. Hoos, "An Ant Colony Optimisation Algorithm for the 2D and 3D Hydrophobic Polar Protein Folding Problem," BMC Bioinformatics, Vol. 6, No. 30, 2005.

[121] A. Shmygelska and H.H. Hoos, "An Improved Ant Colony Optimisation Algorithm for the 2D HP Protein Folding Problem," Advances in Artificial Intelligence: 16th Conference of the Canadian Society for Computational Studies of Intelligence, AI 2003. Springer's Lecture Notes in Computer Science (LNCS) series, Vol. 2671, pp. 400-417.

[122] K. Simons, C. Kooperberg, E. Huang, and D. Baker, "Assembly of Protein Tertiary Structures from Fragments with Similar Local Sequences Using Simulated Annealing and Bayesian Scoring Function," Journal of Molecular Biology, Vol. 268, 1997, pp. 209-225.

[123] M.J. Sippl, "Calculation of Conformotional Ensembles from Potentials of Mean Force: An Approach to the Knowledge-based Prediction of Local Structures in Globular Proteins," Journal of Molecular Biology, Vol. 213, 1990, pp. 859-883.

[124] M.J. Sippl, "Knowledge-based Potentials for Proteins," Current Opinion in Structural Biology, Vol. 5, 1995, pp. 229-235.

[125] J. Skolnick, A. Kolinski, D. Kihara, M. Betancourt, P. Rotkiewicz, and M. Boniecki, "Ab Initio Protein Structure Prediction via a Combination of Threading, Lattice Folding, Clustering, and Structure Refinement," Proteins: Structure, Function, and Genetics Suppl., Vol. 5, 2001, pp. 149-156.

[126] J. Skolnick, Y. Zhang, A.K. Arakaki, A. Kolinski, M. Boniecki, A. Szilagyi, and D. Kihara, "TOUCHSTONE: A Unified Approach to Protein Structure Prediction," Proteins: Structure, Function, and Genetics, Vol. 53, 2003, pp. 469-479.

[127] R. Srinivasan and G.D. Rose, "Ab Initio Prediction of Protein Structure Using LINUS," Proteins: Structure, Function, and Genetics, Vol. 47, 2002, pp. 489-495.

[128] T. Stützle, "Local Search Algorithms for Combinatorial Problems – Analysis, Improvements, and New Applications," PhD thesis, FB Informatik, TU Darmstadt, 1998, pp. 134-135.

[129] T. Stützle and H.H. Hoos, "MAX-MIN Ant System," Future Generation Computer Systems, Vol. 16, No. 8, 2000, pp. 889-914.

[130] Y. Sugita and Y. Okamoto, "Replica-Exchange Multicanonical Algorithm and Multicanonical Replica-Exchange Method for Simulating Systems with Rough Energy Landscape," Chemical Physics Letters, Vol. 329, 2000, pp. 261-270.

[131] S. Sun, "Reduced Representation Model of Protein Structure Prediction: Statistical Potential and Genetic Algorithms," Protein Science, Vol. 2, 1993, pp. 762-785.

[132] L. Toma and S. Toma, "Contact Interactions Method: A New Algorithm for Protein Folding Simulations," Protein Science, Vol. 5, 1996, pp. 147-153.

[133] G.M. Torrie and J.P. Valleau, "Nonphysical Sampling Distributions in MC Free Energy Estimation: Umbrella Sampling," Journal of Computational Physics, Vol. 23, 1977, pp. 187-199.

[134] R. Unger and J. Moult, "A Genetic Algorithm for Three Dimensional Protein Folding Simulations," In Proceedings of the 5^{th} International Conference on Genetic Algorithms, Morgan Kaufmann, 1993, pp. 581-588.

[135] R. Unger and J. Moult, "Genetic Algorithms for Protein Folding Simulations," Journal of Molecular Biology, Vol. 231, No. 1, 1993, pp. 75-81.

[136] R. Unger and J. Moult, "Finding the Lowest Free Energy Conformation of a Protein is a NP-hard Problem: Proof and Implications," Bulletin of Mathematical Biology, Vol. 55, No. 6, 1993, pp. 1183-1198.

[137] S. Vajda, M. Sippl, and J. Novotny, "Empirical Potentials and Functions for Protein Folding and Binding," Current Opinion in Structural Biology, Vol. 7, 1997, pp. 222-228.

[138] C.A. Voigt, D.B. Gordon, and S.L. Mayo, "Trading Accuracy for Speed: A Quantitative Comparison of Search Algorithms in Protein Sequence Design," Journal of Molecular Biology, Vol. 299, 2000, pp. 789-803.

[139] N. Vu, H. Feng, and Y. Bai, "The Folding Pathway of Barnase: The Rate-Limiting Transition State and a Hidden Intermediate under Native Conditions," Biochemistry, Vol. 43, 2004, pp. 3346-3356.

[140] D.J. Wales and J.P.K. Doye, "Global Optimization by Basin-Hopping and the Lowest Energy Structures of Lennard-Jones Clusters Containing up to 110 Atoms," Journal of Physical Chemistry, A, Vol. 101, 1997, pp. 5111-5116.

[141] B. Wallner and A. Elofsson, "Can Correct Protein Models Be Identified?" Protein Science, Vol. 12, 2003, pp. 1073-1086.

[142] A. Wallqvist and M. Ullner, "A Simplified Amino Acid Potential for Use in Structure Predictions of Proteins," Proteins: Structure, Function, and Genetics, Vol. 18, 1994, pp. 267-280.

[143] R.F. Weaver, "Molecular Biology," WCB McGraw Hill, 1999.

[144] T.R. Weikl and K.A. Dill, "Folding Rates and Low-Entropy-Loss Routes of Two-State Proteins," Journal of Molecular Biology, Vol. 329, 2003, pp. 585-598.

[145] W. Wenzel and K. Hamacher, "Stochastic Tunneling Approach for Global Minimization of Complex Potential Energy Landscapes," Physical Review Letters, Vol. 82, 1999, pp. 3003-3007.

[146] C. Wilson and S. Doniach, "A Computer Model to Dynamically Simulate Protein Folding: Studies with Crambin," Proteins: Structure, Function, and Genetics, Vol. 6, 1989, pp. 193-209.

[147] P.G. Wolynes, "Three Paradoxes of Protein Folding," In Protein Folds, A Distance-Based Approach, eds. H. Bohr and S. Brunak, CRC Press, FL, USA 1996, pp. 3-17.

[148] H. Xu and B.J. Berne, "Multicanonical Jump Walk Annealing: An Efficient Method for Geometric Optimization," Journal of Chemical Physics, Vol. 112, 2000, pp. 2701-2708.

[149] H. Xu and B.J. Berne, "Multicanonical jump walking: A method for efficiency sampling rough energy landscapes," Journal of Chemical Physics, Vol. 110, 1999, pp. 10299-10306.

[150] K. Yue and K.A. Dill, "Forces of Tertiary Structural Organization in Globular Proteins," Proceedings of National Academy of Sciences of the USA, Vol. 92, 1992, pp. 146-150.

[151] M.J. Zaki, V. Nadimpally, D. Bardhan, and C. Bystroff, "Predicting Protein Folding Pathways," Bioinformatics, Vol. 20, 2004, Suppl. 1, pp. i386-i393.

[152] J.L. Zhang and J.S. Liu, "A New Sequential Importance Sampling Method and its Application to the Two-Dimensional Hydrophobic-Hydrophilic Model," Journal of Chemical Physics, Vol. 117, No. 7, 2002, pp. 3492-3498.

[153] Y. Zhang, D. Kihara, and J. Skolnick, "Local Energy Landscape Flattening: Parallel Hyperbolic Monte Carlo Sampling of Protein Folding," Proteins: Structure, Function, and Genetics, Vol. 48, 2002, pp. 192-201.

[154] Y. Zhang and J. Skolnick, "Parallel-Hat Tempering: A Monte Carlo Search Scheme for the Identification of Low-Energy Structures," Journal of Chemical Physics, Vol. 115, No. 11, 2001, pp. 5027-5032.

[155] Y.Q. Zhou and M. Karplus, "Folding Thermodynamics of a Model Three-Helix-Bundle Protein," Proceedings of the National Academy of Sciences of the USA, Vol. 94, 1997, pp. 14429-14432.

[156] M. Zlochin, M. Birattari, N. Meuleau, and M. Dorigo, "Model-based Search for Combinatorial Optimization," Technical Report, IRIDIA, Brussels, 2001.

Appendix A

Ant Colony Optimization

In this appendix, we include supplemenary information for our Ant Colony Optimization (ACO) for the 2D and 3D Hydrophobic Polar (HP) protein folding problems. The following tables provide additional information on our new test sets of biological and randomly generated HP sequences and the results from our computational experiment with ACO and PERM. For the biological sequences, we show the PDB IDs of the original protein sequences. Note that in our translation from protein sequences into HP strings, non-standard amino acid symbols, such as X and Z, were skipped; consequently, some of our HP strings differ in length from the respective PDBSELECT sequences.

ID	PDB ID	HP sequence	Length
B30-1	1HA9:A	phpphhhphphhpppppppppphhhphpphphhph	34
B30-2	1CE4:A	ppphpppppppphphhhphphhhppphhhhphpphpp	35
B30-3	1FWO:A	pphpppphpphhhhpppppphhhhppphphhhhppphhp	35
B30-4	1ZTA	hppphpphpppphpphhppphpppphhphppphhhpp	35
B30-5	1G1Z:A	pppphphhhppphhhppphhppphhhphphp	31
B30-6	1BZG	hhpppphhpppphpphpphpppphhhppphhhphpph	34
B30-7	1BH7	hphhppphhhhhphhphphphhhhppphppphphh	33
B30-8	1B4G	hhppphphphpppphpphhpppphppphh	29
B30-9	1BNX:A	phhhhhppphppphhhhhhhhhhphhhphhhhhhpp	33
B30-10	1FCT	hhhhhppphhhphhhphhhphphhppphpphh	32

Table A.1: Biological sequences of length ≈ 30

ID	E_{min} (2D)	PERM t_1	PERM t_2	PERM t_{exp}	ACO t_{avg}
B30-1	-12	0.43	0.56	0.49	0.71
B30-2	-12	0.15	4.09	0.29	1.08
B30-3	-14	0.15	0.02	0.04	1.19
B30-4	-10	2.5	0.3	0.54	3.64
B30-5	-13	0.10	0.17	0.12	0.22
B30-6	-13	0.84	10.40	1.55	70.94
B30-7	-16	0.02	0.02	0.02	0.19
B30-8	-8	0.09	0.003	0.006	0.17
B30-9	-18	0.002	0.008	0.003	0.06
B30-10	-16	0.003	12.50	0.006	4.08

Table A.2: Performance comparison of PERM and ACO on biological sequences of length ≈ 30 in 2D

ID	E_{min} (3D)	PERM t_1	PERM t_2	PERM t_{exp}	ACO t_{avg}
B30-1	-17	1.14	1.03	1.08	1.19
B30-2	-16	1.06	1.3	1.17	2.29
B30-3	-18	1.20	1.20	1.20	2.34
B30-4	-16	3.4	1.2	1.77	20.18
B30-5	-20	2.2	3.04	2.55	65.71
B30-6	-19	1.20	1.50	1.33	23.18
B30-7	-25	1.10	2.40	1.51	62.36
B30-8	-13	1.10	1.07	1.09	1.74
B30-9	-26	1.20	1.09	1.14	1.57
B30-10	-24	1.8	3291	3.60	163.23

Table A.3: Performance comparison of PERM and ACO on biological sequences of length ≈ 30 in 3D

ID	PDB ID	HP sequence	Length
B50-1	1KBF:A	hphpppphpppphhpphpphpppphhhhhpppppphpppppppppphhhpp	49
B50-2	1KBH:A	phpppppphhhpphpphhpppphphhpphpphhhhhphhpphphhphp	47
B50-3	1G9P:A	hhhphhhphpppppppppphphhpphpphpphhhhhhhhhhhhhh	45
B50-4	1YUJ:A	hphpphpphhhpppphpppppppphhpphhphhhhpppppphpphpphpphh	50
B50-5	1GAB	phpphhhpphpppphhhpphphhpppphhhphhpphpphpphphhpppphhphph	53
B50-6	2BRZ	pppppphhpphhhpppphhpppphpppphpppphpphpphhpppppphpphpphpph	53
B50-7	1VPC	pphphhphhhphhpphhhhphphhpppppphhhhpppppphhppp	45
B50-8	1VIB	hphphhhhhhpppppppphphphpppphphhpphpphhpphhpppppphppppp	54
B50-9	1CEU:A	hppphhpppphhpppphhpphphphhpphpppphhpphphhhpphhpphhpphh	51
B50-10	1CFH	hpphphpphhphphpppphpppppphpphpphhpppppppphhpphhp	47

Table A.4: Biological sequences of length ≈ 50

ID	E_{min} (2D)	PERM			ACO
		t_1	t_2	t_{exp}	t_{avg}
B50-1	-12	5.60	3.90	4.60	39.5
B50-2	-19	650.40	0.90	1.80	283.31
B50-3	-20	3.80	0.04	0.08	2.47
B50-4	-17	3.33	2.24	2.68	184.5
B50-5	-22	4.90	117.80	9.41	820.37
B50-6	-14	0.43	0.74	0.55	2.08
B50-7	-17	270.70	298.80	284.06	130.35
B50-8	-14	0.04	2.16	0.07	0.29
B50-9	-21	0.40	0.23	0.29	55.32
B50 -10	-14	0.27	0.3	0.28	3.22

Table A.5: Performance comparison of PERM and ACO on biological sequences of length ≈ 50 in 2D

ID	E_{min} (3D)	PERM t_1	PERM t_2	PERM t_{exp}	ACO t_{avg}
B50-1	-18	2.08	1.50	1.74	14.65
B50-2	-29	43.60	18.20	25.68	701.84
B50-3	-28	1.40	1.30	1.35	5.52
B50-4	-28	2.29	1 822.2	117.90	(-27)
B50-5	-37	18.2	1 235.9	164.87	(-36)
B50-6	-25	2.80	10.7	4.44	1 638.29
B50-7	-26	406.90	2.70	5.36	814.11
B50-8	-25	10.05	(-24)	-	1026.12
B50-9	-36	3 492.6	4.03	16.10	(-35)
B50 -10	-22	2.50	1.40	1.80	200.68

Table A.6: Performance comparison of PERM and ACO on biological sequences of length $\approx$ 50 in 3D

ID	HP sequence	Length	H fraction
R30-1	pphphpphpphphpppppphphhpphhhpph	30	0.40
R30-2	hpppppphhphhhpphpppphhhhhhhhpph	30	0.53
R30-3	pphpphphpppphhpppphhpphphphhhhh	30	0.47
R30-4	pphhhhpppppphphhpphpppppphphh	30	0.40
R30-5	hhpppphhphpphhphhhpphphpphphpph	30	0.50
R30-6	phhpphphpppppphhhphpppphpphhhh	30	0.43
R30-7	phhhphhhpppphhphphphhhpppppph	30	0.47
R30-8	ppphpphhphphhpphpphphhhphhphph	30	0.50
R30-9	hhphhpppphpppphphpphpphphhphph	30	0.43
R30-10	hphphpphpphphhpppphphhhphhphhp	30	0.53

Table A.7: Random sequences of length 30

ID	E_{min} (2D)	PERM t_1	PERM t_2	PERM t_{exp}	ACO t_{avg}
R30-1	-90	0.4	0.001	0.002	0.27
R30-2	-13	1.60	0.011	0.022	0.58
R30-3	-10	0.14	0.001	0.002	0.17
R30-4	-9	0.04	0.07	0.05	0.26
R30-5	-13	0.01	0.038	0.02	0.44
R30-6	-11	86.3	0.33	0.65	12.85
R30-7	-8	0.007	0.03	0.01	0.10
R30-8	-12	0.033	0.017	0.045	0.38
R30-9	-9	0.002	0.11	0.004	0.14
R30-10	-12	0.007	0.001	0.002	0.34

Table A.8: Performance comparison of PERM and ACO on random sequences of length 30 in 2D

ID	E_{min} (3D)	PERM t_1	PERM t_2	PERM t_{exp}	ACO t_{avg}
R30-1	-13	0.16	0.16	0.16	0.47
R30-2	-18	0.22	0.44	0.29	3.44
R30-3	-15	1.00	1.85	1.30	17.81
R30-4	-12	0.14	0.26	0.18	0.25
R30-5	-19	0.53	0.51	0.52	28.77
R30-6	-14	0.22	0.10	0.14	1.11
R30-7	-11	0.118	0.067	0.09	0.055
R30-8	-17	0.19	0.28	0.225	0.57
R30-9	-14	0.22	0.24	0.23	1.48
R30-10	-19	2.22	0.27	0.48	16.69

Table A.9: Performance comparison of PERM and ACO on random sequences of length 30 in 3D

ID	HP sequence	Length	H fraction
R50-1	ppphphphpppppphphhphhhppppphphphhpppphphphhhphpphhh	50	0.48
R50-2	hhhhhhphpppphpppphhhppphhhhhphpphhhhppphpppphpppphhppph	50	0.52
R50-3	hhhphhphphhpppphpphhhphhhhhhhhphpppphphhhphpppphphpppp	50	0.54
R50-4	hhhhppphpphhppphhhpppppphpphhpppphhhhphhhpppphpppphhh	50	0.48
R50-5	ppphhpppphpphphpphhphphhhhhhhhpphhhppphpppphpppphphp	50	0.46
R50-6	pphppphphphhhpphpppppphpphhpppphphhhhphhphhhhppphhph	50	0.48
R50-7	phhpppphphpphhpppphhhphphhhphhphhphphpphhhphphhhhhp	50	0.56
R50-8	hpppphphphpphphpphhhhhhhppphhhhhphpppphhpphphhpppphhp	50	0.50
R50-9	pphphhphhphpphpppphphphphphpphhpppppphpphhhhhppphhh	50	0.46
R50-10	hhphphpppphpphphpppppphphpppphpppppphhhphhhhhphhphph	50	0.44

Table A.10: Random sequences of length 50

ID	E_{min} (2D)	PERM t_1	PERM t_2	PERM t_{exp}	ACO t_{avg}
R50-1	-29	575.90	1.05	2.09	367.70
R50-2	-34	1.40	2.40	1.77	1 150.10
R50-3	-32	1.00	5.50	1.69	893.66
R50-4	-32	2.70	2.40	2.54	1 329.01
R50-5	-32	(-31)	5.49	-	(-31)
R50-6	-32	3.8	1.6	2.25	1 153.81
R50-7	-38	15 322.20	46.20	92.12	(-37)
R50-8	-33	1.30	0.91	1.07	837.13
R50-9	-30	9892.00	1.70	3.40	1000.49
R50-10	-27	72.80	1.80	3.51	530.84

Table A.11: Performance comparison of PERM and ACO on random sequences of length 50 in 2D

ID	E_{min} (3D)	PERM t_1	PERM t_2	PERM t_{exp}	ACO t_{avg}
R50-1	-29	575.90	1.05	2.09	367.70
R50-2	-34	1.40	2.40	1.77	1 150.10
R50-3	-32	1.00	5.50	1.69	893.66
R50-4	-32	2.70	2.40	2.54	1 329.01
R50-5	-32	(-31)	5.49	-	(-31)
R50-6	-32	3.8	1.6	2.25	1 153.81
R50-7	-38	15 322.20	46.20	92.12	(-37)
R50-8	-33	1.30	0.91	1.07	837.13
R50-9	-30	9892.00	1.70	3.40	1000.49
R50-10	-27	72.80	1.80	3.51	530.84

Table A.12: Performance comparison of PERM and ACO on random sequences of length 50 in 3D

Appendix B

Adaptive Bin Framework Monte Carlo Search

In this appendix, we report the necessary information to reconstruct the lowest energy conformations of homopolymers of length $12, 24, 32$, and 64 obtained by our adaptive bin framework search, introduced for the Face-centered Cubic (FCC) β-sheet protein folding problem.

The homopolymer of length 64

Total energy $= -391$, short-range energy $= -212$, long-range energy $= -179$.

The homopolymer of length 32

Total energy $= -161$, short-range energy $= -112$, long-range energy $= -49$.

The homopolymer of length 24

Total energy $= -109$, short-range energy $= -68$, long-range energy $= -41$.

The homopolymer of length 12

Total energy $= -39$, short-range energy $= -28$, long-range energy $= -11$.

$vector[\#]$	(x, y, z)
v[0]	$(1, 1, 0)$
v[1]	$(0, 1, -1)$
v[2]	$(0, 1, -1)$
v[3]	$(-1, 0, -1)$
v[4]	$(-1, -1, 0)$
v[5]	$(-1, -1, 0)$
v[6]	$(0, -1, 1)$
v[7]	$(0, -1, 1)$
v[8]	$(1, 0, 1)$
v[9]	$(1, 0, 1)$
v[10]	$(1, 1, 0)$
v[11]	$(1, 1, 0)$
v[12]	$(0, 1, -1)$
v[13]	$(0, 1, -1)$
v[14]	$(0, 1, -1)$
v[15]	$(-1, 0, -1)$
v[16]	$(-1, 0, -1)$
v[17]	$(-1, -1, 0)$
v[18]	$(-1, -1, 0)$
v[19]	$(-1, -1, 0)$
v[20]	$(0, -1, 1)$
v[21]	$(0, -1, 1)$
v[22]	$(0, -1, 1)$
v[23]	$(1, 0, 1)$
v[24]	$(1, 0, 1)$
v[25]	$(0, 1, 1)$
v[26]	$(1, 1, 0)$
v[27]	$(1, 1, 0)$
v[28]	$(0, 1, -1)$
v[29]	$(0, 1, -1)$
v[30]	$(0, 1, -1)$
v[31]	$(-1, 0, -1)$
v[32]	$(-1, 0, -1)$
v[33]	$(-1, -1, 0)$
v[34]	$(-1, -1, 0)$
v[35]	$(-1, -1, 0)$
v[36]	$(0, -1, 1)$
v[37]	$(0, -1, 1)$
v[38]	$(1, 0, 1)$
v[39]	$(1, 0, 1)$
v[40]	$(1, 1, 0)$
v[41]	$(1, 1, 0)$
v[42]	$(0, 1, -1)$
v[43]	$(0, 1, -1)$
v[44]	$(-1, 0, -1)$
v[45]	$(-1, -1, 0)$
v[46]	$(-1, -1, 0)$
v[47]	$(0, -1, 1)$
v[48]	$(1, 1, 0)$
v[49]	$(1, 1, 0)$
v[50]	$(1, 0, -1)$

Table B.1: Vectors for the best found conformation of 64 amino acids (total energy $= -391$, short-range energy $= -212$, long-range energy $= -179$).

vector[#]	(x, y, z)
v[51]	$(1, 0, -1)$
v[52]	$(1, 1, 0)$
v[53]	$(0, -1, 1)$
v[54]	$(-1, -1, 0)$
v[55]	$(-1, 0, 1)$
v[56]	$(-1, 0, 1)$
v[57]	$(-1, -1, 0)$
v[58]	$(1, 0, -1)$
v[59]	$(0, 1, -1)$
v[60]	$(1, 0, -1)$
v[61]	$(0, 1, -1)$
v[62]	$(1, 1, 0)$

Table B.2: Continued, vectors for the best found conformation of 64 amino acids (total energy $= -391$, short-range energy $= -212$, long-range energy $= -179$).

triplet	θ_1	θ_2	θ_3	ε_{beta}
v_0, v_1, v_2	120	180	60	−4
v_1, v_2, v_3	180	120	60	−4
v_2, v_3, v_4	120	120	120	0
v_3, v_4, v_5	120	180	60	−4
v_4, v_5, v_6	180	120	60	−4
v_5, v_6, v_7	120	180	60	−4
v_6, v_7, v_8	180	120	60	−4
v_7, v_8, v_9	120	180	60	−4
v_8, v_9, v_{10}	180	120	60	−4
v_9, v_{10}, v_{11}	120	180	60	−4
v_{10}, v_{11}, v_{12}	180	120	60	−4
v_{11}, v_{12}, v_{13}	120	180	60	−4
v_{12}, v_{13}, v_{14}	180	180	0	−4
v_{13}, v_{14}, v_{15}	180	120	60	−4
v_{14}, v_{15}, v_{16}	120	180	60	−4
v_{15}, v_{16}, v_{17}	180	120	60	−4
v_{16}, v_{17}, v_{18}	120	180	60	−4
v_{17}, v_{18}, v_{19}	180	180	0	−4
v_{18}, v_{19}, v_{20}	180	120	60	−4
v_{19}, v_{20}, v_{21}	120	180	60	−4
v_{20}, v_{21}, v_{22}	180	180	0	−4
v_{21}, v_{22}, v_{23}	180	120	60	−4
v_{22}, v_{23}, v_{24}	120	180	60	−4
v_{23}, v_{24}, v_{25}	180	120	60	−4
v_{24}, v_{25}, v_{26}	120	120	60	−4
v_{25}, v_{26}, v_{27}	120	180	60	−4
v_{26}, v_{27}, v_{28}	180	120	60	−4
v_{27}, v_{28}, v_{29}	120	180	60	−4
v_{28}, v_{29}, v_{30}	180	180	0	−4
v_{29}, v_{30}, v_{31}	180	120	60	−4
v_{30}, v_{31}, v_{32}	120	180	60	−4
v_{31}, v_{32}, v_{33}	180	120	60	−4
v_{32}, v_{33}, v_{34}	120	180	60	−4
v_{33}, v_{34}, v_{35}	180	180	0	−4
v_{34}, v_{35}, v_{36}	180	120	60	−4
v_{35}, v_{36}, v_{37}	120	180	60	−4
v_{36}, v_{37}, v_{38}	180	120	60	−4
v_{37}, v_{38}, v_{39}	120	180	60	−4
v_{38}, v_{39}, v_{40}	180	120	60	−4
v_{39}, v_{40}, v_{41}	120	180	60	−4
v_{40}, v_{41}, v_{42}	180	120	60	−4
v_{41}, v_{42}, v_{43}	120	180	60	−4
v_{42}, v_{43}, v_{44}	180	120	60	−4
v_{43}, v_{44}, v_{45}	120	120	120	0
v_{44}, v_{45}, v_{46}	120	180	60	−4
v_{45}, v_{46}, v_{47}	180	120	60	−4
v_{46}, v_{47}, v_{48}	120	60	180	0
v_{47}, v_{48}, v_{49}	60	180	120	0
v_{48}, v_{49}, v_{50}	180	120	60	−4
v_{49}, v_{50}, v_{51}	120	180	60	−4
v_{50}, v_{51}, v_{52}	180	120	60	−4

Table B.3: Triplets of vectors, angles: θ_1 (between vectors $\mathbf{v_{i-1}}$ and $\mathbf{v_i}$), θ_2 (between vectors $\mathbf{v_i}$ and $\mathbf{v_{i+1}}$), θ_3 (between vectors $\mathbf{v_{i-1}}$ and $\mathbf{v_{i+1}}$), short-range energy contributions for the best found conformation of 64 amino acids (total energy $= -391$, short-range energy $= -212$, long-range energy $= -179$).

triplet	θ_1	θ_2	θ_3	ε_{beta}
v_{51}, v_{52}, v_{53}	120	60	120	0
v_{52}, v_{53}, v_{54}	60	120	180	0
v_{53}, v_{54}, v_{55}	120	120	60	-4
v_{54}, v_{55}, v_{56}	120	180	60	-4
v_{55}, v_{56}, v_{57}	180	120	60	-4
v_{56}, v_{57}, v_{58}	120	60	180	0
v_{57}, v_{58}, v_{59}	60	120	120	0
v_{58}, v_{59}, v_{60}	120	120	0	-4
v_{59}, v_{60}, v_{61}	120	120	0	-4
v_{60}, v_{61}, v_{62}	120	120	60	-4

Table B.4: Continued, triplets of vectors, angles: θ_1 (between vectors $\mathbf{v_{i-1}}$ and $\mathbf{v_i}$), θ_2 (between vectors $\mathbf{v_i}$ and $\mathbf{v_{i+1}}$), θ_3 (between vectors $\mathbf{v_{i-1}}$ and $\mathbf{v_{i+1}}$), short-range energy contributions for the best found conformation of 64 amino acids (total energy $= -391$, short-range energy $= -212$, long-range energy $= -179$).

#	Long-range interactions
1	10 interacts with 1
2	11 interacts with 1
3	12 interacts with 1
4	12 interacts with 2
5	13 interacts with 2
6	14 interacts with 2
7	14 interacts with 3
8	15 interacts with 3
9	15 interacts with 4
10	16 interacts with 4
11	17 interacts with 5
12	17 interacts with 4
13	18 interacts with 5
14	19 interacts with 6
15	19 interacts with 5
16	20 interacts with 7
17	20 interacts with 6
18	21 interacts with 7
19	22 interacts with 8
20	22 interacts with 7
21	23 interacts with 9
22	23 interacts with 8
23	24 interacts with 9
24	25 interacts with 9
25	25 interacts with 10
26	26 interacts with 10
27	26 interacts with 11
28	27 interacts with 11
29	28 interacts with 11
30	28 interacts with 12
31	29 interacts with 12
32	29 interacts with 13
33	30 interacts with 2
34	30 interacts with 13
35	30 interacts with 14
36	31 interacts with 3
37	31 interacts with 14
38	31 interacts with 15
39	32 interacts with 4
40	32 interacts with 15
41	32 interacts with 16
42	33 interacts with 5
43	33 interacts with 4
44	33 interacts with 17
45	34 interacts with 19
46	34 interacts with 5
47	34 interacts with 18
48	35 interacts with 20
49	35 interacts with 6
50	35 interacts with 19
51	36 interacts with 21
52	36 interacts with 7
53	36 interacts with 20
54	37 interacts with 22
55	37 interacts with 21

Table B.5: Long-range interactions for the best found conformation of 64 amino acids (total energy $= -391$, short-range energy $= -212$, long-range energy $= -179$).

#	Long-range interactions
56	38 interacts with 23
57	38 interacts with 22
58	39 interacts with 24
59	39 interacts with 23
60	40 interacts with 24
61	40 interacts with 25
62	40 interacts with 9
63	41 interacts with 25
64	41 interacts with 26
65	41 interacts with 10
66	41 interacts with 27
67	42 interacts with 10
68	42 interacts with 27
69	42 interacts with 11
70	42 interacts with 1
71	42 interacts with 28
72	43 interacts with 1
73	43 interacts with 28
74	43 interacts with 12
75	43 interacts with 2
76	43 interacts with 29
77	43 interacts with 30
78	44 interacts with 2
79	44 interacts with 3
80	44 interacts with 30
81	44 interacts with 31
82	45 interacts with 3
83	45 interacts with 33
84	45 interacts with 4
85	45 interacts with 31
86	45 interacts with 32
87	46 interacts with 35
88	46 interacts with 6
89	46 interacts with 34
90	46 interacts with 5
91	46 interacts with 33
92	47 interacts with 36
93	47 interacts with 7
94	47 interacts with 35
95	47 interacts with 6
96	48 interacts with 38
97	48 interacts with 37
98	48 interacts with 22
99	48 interacts with 8
100	48 interacts with 36
101	48 interacts with 7
102	49 interacts with 39
103	49 interacts with 38
104	49 interacts with 23
105	49 interacts with 40
106	49 interacts with 9
107	49 interacts with 8
108	50 interacts with 48
109	50 interacts with 8
110	50 interacts with 47

Table B.6: Continued, long-range interactions for the best found conformation of 64 amino acids (total energy $= -391$, short-range energy $= -212$, long-range energy $= -179$).

#	Long-range interactions
111	51 interacts with 47
112	51 interacts with 46
113	51 interacts with 44
114	51 interacts with 45
115	52 interacts with 6
116	52 interacts with 46
117	52 interacts with 5
118	52 interacts with 45
119	52 interacts with 3
120	52 interacts with 4
121	53 interacts with 5
122	53 interacts with 4
123	54 interacts with 4
124	54 interacts with 17
125	54 interacts with 16
126	55 interacts with 53
127	55 interacts with 3
128	55 interacts with 4
129	55 interacts with 15
130	56 interacts with 52
131	56 interacts with 53
132	56 interacts with 3
133	57 interacts with 1
134	57 interacts with 51
135	57 interacts with 52
136	57 interacts with 44
137	57 interacts with 2
138	57 interacts with 3
139	58 interacts with 50
140	58 interacts with 42
141	58 interacts with 1
142	58 interacts with 51
143	58 interacts with 43
144	58 interacts with 44
145	59 interacts with 40
146	59 interacts with 49
147	59 interacts with 9
148	59 interacts with 41
149	59 interacts with 10
150	59 interacts with 50
151	59 interacts with 42
152	60 interacts with 9
153	60 interacts with 8
154	60 interacts with 10
155	60 interacts with 50
156	60 interacts with 58
157	60 interacts with 1
158	60 interacts with 57
159	61 interacts with 8
160	61 interacts with 7
161	61 interacts with 50
162	61 interacts with 47
163	61 interacts with 6
164	61 interacts with 51
165	61 interacts with 57

Table B.7: Continued, long-range interactions for the best found conformation of 64 amino acids (total energy $= -391$, short-range energy $= -212$, long-range energy $= -179$).

#	Long-range interactions
166	61 interacts with 52
167	62 interacts with 6
168	62 interacts with 52
169	62 interacts with 56
170	62 interacts with 53
171	63 interacts with 6
172	63 interacts with 19
173	63 interacts with 5
174	63 interacts with 53
175	64 interacts with 5
176	64 interacts with 53
177	64 interacts with 18
178	64 interacts with 17
179	64 interacts with 54

Table B.8: Continued, long-range interactions for the best found conformation of 64 amino acids (total energy $= -391$, short-range energy $= -212$, long-range energy $= -179$).

$vector[\#]$	(x, y, z)
v[0]	$(1, 0, -1)$
v[1]	$(1, -1, 0)$
v[2]	$(1, -1, 0)$
v[3]	$(0, -1, 1)$
v[4]	$(-1, 0, 1)$
v[5]	$(-1, 0, 1)$
v[6]	$(-1, 1, 0)$
v[7]	$(-1, 1, 0)$
v[8]	$(0, 1, -1)$
v[9]	$(0, 1, -1)$
v[10]	$(1, 0, -1)$
v[11]	$(1, 0, -1)$
v[12]	$(1, -1, 0)$
v[13]	$(1, -1, 0)$
v[14]	$(1, -1, 0)$
v[15]	$(0, -1, 1)$
v[16]	$(0, -1, 1)$
v[17]	$(-1, 0, 1)$
v[18]	$(-1, 0, 1)$
v[19]	$(0, 1, 1)$
v[20]	$(-1, 1, 0)$
v[21]	$(-1, 1, 0)$
v[22]	$(0, 1, -1)$
v[23]	$(0, 1, -1)$
v[24]	$(1, 0, -1)$
v[25]	$(1, 0, -1)$
v[26]	$(1, -1, 0)$
v[27]	$(1, -1, 0)$
v[28]	$(0, -1, 1)$
v[29]	$(0, -1, 1)$
v[30]	$(-1, 0, 1)$

Table B.9: Vectors for the best found conformation of 32 amino acids (total energy $= -161$, short-range energy $= -112$, long-range energy $= -49$).

triplet	θ_1	θ_2	θ_3	ε_{beta}
v_0, v_1, v_2	120	180	60	-4
v_1, v_2, v_3	180	120	60	-4
v_2, v_3, v_4	120	120	120	0
v_3, v_4, v_5	120	180	60	-4
v_4, v_5, v_6	180	120	60	-4
v_5, v_6, v_7	120	180	60	-4
v_6, v_7, v_8	180	120	60	-4
v_7, v_8, v_9	120	180	60	-4
v_8, v_9, v_{10}	180	120	60	-4
v_9, v_{10}, v_{11}	120	180	60	-4
v_{10}, v_{11}, v_{12}	180	120	60	-4
v_{11}, v_{12}, v_{13}	120	180	60	-4
v_{12}, v_{13}, v_{14}	180	180	0	-4
v_{13}, v_{14}, v_{15}	180	120	60	-4
v_{14}, v_{15}, v_{16}	120	180	60	-4
v_{15}, v_{16}, v_{17}	180	120	60	-4
v_{16}, v_{17}, v_{18}	120	180	60	-4
v_{17}, v_{18}, v_{19}	180	120	60	-4
v_{18}, v_{19}, v_{20}	120	120	60	-4
v_{19}, v_{20}, v_{21}	120	180	60	-4
v_{20}, v_{21}, v_{22}	180	120	60	-4
v_{21}, v_{22}, v_{23}	120	180	60	-4
v_{22}, v_{23}, v_{24}	180	120	60	-4
v_{23}, v_{24}, v_{25}	120	180	60	-4
v_{24}, v_{25}, v_{26}	180	120	60	-4
v_{25}, v_{26}, v_{27}	120	180	60	-4
v_{26}, v_{27}, v_{28}	180	120	60	-4
v_{27}, v_{28}, v_{29}	120	180	60	-4
v_{28}, v_{29}, v_{30}	180	120	60	-4

Table B.10: Triplets of vectors, angles: θ_1 (between vectors $\mathbf{v_{i-1}}$ and $\mathbf{v_i}$), θ_2 (between vectors $\mathbf{v_i}$ and $\mathbf{v_{i+1}}$), θ_3 (between vectors $\mathbf{v_{i-1}}$ and $\mathbf{v_{i+1}}$), short-range energy contributions for the best found conformation of 32 amino acids (total energy $= -161$, short-range energy $= -112$, long-range energy $= -49$).

#	Long-range interactions
1	10 interacts with 1
2	11 interacts with 1
3	12 interacts with 1
4	12 interacts with 2
5	13 interacts with 2
6	14 interacts with 2
7	14 interacts with 3
8	15 interacts with 3
9	15 interacts with 4
10	16 interacts with 4
11	17 interacts with 5
12	17 interacts with 4
13	18 interacts with 5
14	19 interacts with 6
15	19 interacts with 5
16	20 interacts with 7
17	20 interacts with 6
18	21 interacts with 7
19	22 interacts with 8
20	22 interacts with 7
21	23 interacts with 9
22	23 interacts with 8
23	24 interacts with 9
24	24 interacts with 10
25	25 interacts with 10
26	25 interacts with 11
27	25 interacts with 1
28	26 interacts with 1
29	26 interacts with 12
30	26 interacts with 2
31	27 interacts with 2
32	27 interacts with 13
33	27 interacts with 14
34	28 interacts with 3
35	28 interacts with 14
36	28 interacts with 15
37	29 interacts with 4
38	29 interacts with 15
39	29 interacts with 16
40	30 interacts with 5
41	30 interacts with 4
42	30 interacts with 17
43	31 interacts with 19
44	31 interacts with 5
45	31 interacts with 18
46	32 interacts with 20
47	32 interacts with 6
48	32 interacts with 19
49	32 interacts with 21

Table B.11: Long-range interactions for the best found conformation of 32 amino acids (total energy $= -161$, short-range energy $= -112$, long-range energy $= -49$).

vector[#]	(x, y, z)
v[0]	$(1, 0, -1)$
v[1]	$(0, -1, -1)$
v[2]	$(0, -1, -1)$
v[3]	$(1, -1, 0)$
v[4]	$(0, -1, -1)$
v[5]	$(0, -1, -1)$
v[6]	$(-1, -1, 0)$
v[7]	$(-1, 0, -1)$
v[8]	$(0, 1, 1)$
v[9]	$(1, 0, 1)$
v[10]	$(0, 1, 1)$
v[11]	$(0, 1, 1)$
v[12]	$(0, 1, 1)$
v[13]	$(1, 1, 0)$
v[14]	$(0, 1, 1)$
v[15]	$(-1, -1, 0)$
v[16]	$(-1, 0, -1)$
v[17]	$(0, -1, -1)$
v[18]	$(0, -1, -1)$
v[19]	$(1, 0, -1)$
v[20]	$(0, -1, -1)$
v[21]	$(0, -1, -1)$
v[22]	$(1, -1, 0)$

Table B.12: Vectors for the best found conformation of 24 amino acids (total energy $= -109$, short-range energy $= -68$, long-range energy $= -41$).

triplet	θ_1	θ_2	θ_3	ε_{beta}
v_0, v_1, v_2	120	180	60	−4
v_1, v_2, v_3	180	120	60	−4
v_2, v_3, v_4	120	120	0	−4
v_3, v_4, v_5	120	180	60	−4
v_4, v_5, v_6	180	120	60	−4
v_5, v_6, v_7	120	120	60	−4
v_6, v_7, v_8	120	60	120	0
v_7, v_8, v_9	60	120	180	0
v_8, v_9, v_{10}	120	120	0	−4
v_9, v_{10}, v_{11}	120	180	60	−4
v_{10}, v_{11}, v_{12}	180	180	0	−4
v_{11}, v_{12}, v_{13}	180	120	60	−4
v_{12}, v_{13}, v_{14}	120	120	0	−4
v_{13}, v_{14}, v_{15}	120	60	180	0
v_{14}, v_{15}, v_{16}	60	120	120	0
v_{15}, v_{16}, v_{17}	120	120	60	−4
v_{16}, v_{17}, v_{18}	120	180	60	−4
v_{17}, v_{18}, v_{19}	180	120	60	−4
v_{18}, v_{19}, v_{20}	120	120	0	−4
v_{19}, v_{20}, v_{21}	120	180	60	−4
v_{20}, v_{21}, v_{22}	180	120	60	−4

Table B.13: Triplets of vectors, angles: θ_1 (between vectors $\mathbf{v_{i-1}}$ and $\mathbf{v_i}$), θ_2 (between vectors $\mathbf{v_i}$ and $\mathbf{v_{i+1}}$), θ_3 (between vectors $\mathbf{v_{i-1}}$ and $\mathbf{v_{i+1}}$), short-range energy contributions for the best found conformation of 24 amino acids (total energy $= -109$, short-range energy $= -68$, long-range energy $= -41$).

#	Long-range interactions
1	10 interacts with 8
2	11 interacts with 8
3	11 interacts with 7
4	11 interacts with 6
5	12 interacts with 6
6	12 interacts with 5
7	13 interacts with 4
8	13 interacts with 5
9	14 interacts with 3
10	15 interacts with 3
11	15 interacts with 2
12	16 interacts with 2
13	17 interacts with 14
14	17 interacts with 1
15	17 interacts with 2
16	17 interacts with 15
17	18 interacts with 14
18	18 interacts with 3
19	18 interacts with 1
20	18 interacts with 2
21	19 interacts with 13
22	19 interacts with 4
23	19 interacts with 14
24	19 interacts with 3
25	20 interacts with 12
26	20 interacts with 13
27	20 interacts with 4
28	21 interacts with 12
29	21 interacts with 6
30	21 interacts with 4
31	21 interacts with 5
32	22 interacts with 10
33	22 interacts with 11
34	22 interacts with 7
35	22 interacts with 6
36	23 interacts with 9
37	23 interacts with 10
38	23 interacts with 8
39	23 interacts with 7
40	24 interacts with 8
41	24 interacts with 7

Table B.14: Long-range interactions for the best found conformation of 24 amino acids (total energy $= -109$, short-range energy $= -68$, long-range energy $= -41$).

vector[#]	(x, y, z)
v[0]	$(-1, -1, 0)$
v[1]	$(0, -1, 1)$
v[2]	$(0, -1, 1)$
v[3]	$(1, -1, 0)$
v[4]	$(1, -1, 0)$
v[5]	$(1, 0, -1)$
v[6]	$(-1, 1, 0)$
v[7]	$(-1, 1, 0)$
v[8]	$(0, 1, -1)$
v[9]	$(0, 1, -1)$
v[10]	$(-1, 1, 0)$

Table B.15: Vectors for the best found conformation of 12 amino acids (total energy $= -39$, short-range energy $= -28$, long-range energy $= -11$).

triplet	θ_1	θ_2	θ_3	ε_{beta}
v_0, v_1, v_2	120	180	60	-4
v_1, v_2, v_3	180	120	60	-4
v_2, v_3, v_4	120	180	60	-4
v_3, v_4, v_5	180	120	60	-4
v_4, v_5, v_6	120	60	180	0
v_5, v_6, v_7	60	180	120	0
v_6, v_7, v_8	180	120	60	-4
v_7, v_8, v_9	120	180	60	-4
v_8, v_9, v_{10}	180	120	60	-4

Table B.16: Triplets of vectors, angles: θ_1 (between vectors $\mathbf{v_{i-1}}$ and $\mathbf{v_i}$), θ_2 (between vectors $\mathbf{v_i}$ and $\mathbf{v_{i+1}}$), θ_3 (between vectors $\mathbf{v_{i-1}}$ and $\mathbf{v_{i+1}}$), short-range energy contributions for the best found conformation of 12 amino acids (total energy $= -39$, short-range energy $= -28$, long-range energy $= -11$).

#	Long-range interactions
1	8 interacts with 5
2	8 interacts with 6
3	9 interacts with 4
4	9 interacts with 5
5	9 interacts with 3
6	10 interacts with 3
7	10 interacts with 2
8	11 interacts with 2
9	11 interacts with 1
10	12 interacts with 2
11	12 interacts with 1

Table B.17: Long-range interactions for the best found conformation of 12 amino acids (total energy $= -39$, short-range energy $= -28$, long-range energy $= -11$).

Index

Φ-value analysis an experimental technique that relies on use of a mutation as a reporter of structural change; a mutation in the protein molecule is engineered which causes a difference in stability between the mutant and wild-type; structural changes are usually measured indirectly by recording the folding speed, **24**

α-helix a common motif in the secondary structure of proteins, a right-handed coiled conformation in which hydrogen bonds are formed between the carbonyl oxygen of the amino acid residue at position n with the amide group, NH, of residue $n + 4$, **3**

β-sheet a commonly occurring form of regular secondary structure in proteins, it consists of a stretch of amino acids whose peptide backbones are almost fully extended, resulting in an elongated pleat-like structure in which the peptide carbonyls point in alternating directions relative to the plane of the sheet, **3**

ϕ dihedral angle the angle specifying rotation around the C^{α}-N axis in proteins, **2**

ψ dihedral angle the angle specifying rotation around the C^{α}-C axis in proteins, **2**

ab initio (*physics*) from first principles; a calculation is said to be *ab initio* (or from first principles) if it relies on basic and established laws of nature without additional assumptions or special models, **7**

aliphatic (*chemistry*) organic compounds in which carbon atoms form open chains (straight or branched), not aromatic rings, **2**

amino acid any molecule that contains both amine and carboxylic acid functional groups, **1**

amino group functional group composed of a nitrogen and two hydrogen atoms NH_2 covalently linked, in the process it gives the free electron pair of the nitrogen atom to a proton, and turns into positively charged NH_3^+, **1**

Ant Colony Optimization (ACO) a population-based stochastic search method inspired by real ant colonies; it is a construction-based search method based on the *pheromone information* accumulated over previous runs of the search and the *heuristic function* that provides information about the desirability of the solution component considered, **47**

aromatic (*chemistry*) having an unsaturated ring, **2**

attrition a condition when a chain runs into itself during the chain growth process, **36**

backbone (of a protein) the main chain of proteins, composed of N, C^{α}, C, O atoms, **1**

beta sheet energy potential an energy function used to fold β-sheets; according to this potential residues are classified as either being in an extended beta state or not; for every residue in the extended state there is a favourable energy contribution; additionally, contact (long-range) energy is usually present in the beta sheet potential, **18**

C - terminus the extremity of the amino acid chain terminated by a free carboxyl group, **2**

canonical simulation same as equilibrium simulation, **30**

carboxyl group functional group composed of $COOH$, in the process it dissociates into COO^{-} and a proton H^{+}, **1**

Central Dogma of genomics principle believe that asserts "sequence determines structure determines function", **6**

chaperon (*biology*) proteins whose function is to assist other proteins in achieving proper folding, **6**

coarse sampling of energy landscapes broad exploration of the search landscape that includes sampling of all of the "relevant" parts of the search space, **26**

complex search landscapes search landscapes arising in complex systems with many degrees of freedom such as spin glass and biomolecular systems such as proteins, **26**

conformational search space a set of all possible conformations for a molecule, **15**

construction-based search the search process in which the search starts from an empty candidate solution and solution components are iteratively added until a complete candidate solution is constructed, this process is usually repeated multiple times, **35**

contact order (CO) the sequence separation between residues in contact, **24**

cubic lattice a three dimensional lattice that has three base vectors perpendicular to each other, **16**

decoys (a set of) a mixture of native and non-native protein folds used for testing the ability of energy potentials to discriminate between native and non-native folds, **23**

density of states a property in statistical and condensed matter physics that quantifies how closely packed energy levels are in a particular physical system, **31**

detailed balance condition a condition that ensures that the rate at which the system transitions into and out of any state are equal; the correct distribution for a physical system in thermal equilibrium is the Boltzmann distribution, **30**

diffusion-collision model a model that proposes that local elements of native stable secondary structure form independently of tertiary structure; they collide and adhere to form tertiary interactions, **44**

diversification an important property of the search process, that concentrates on further exploration of the search space in order to avoid search stagnation (or entrapment in a local optimum), **39**

domain (*biochemistry*) a "folding unit" of a protein, the part of the protein sequence that folds largely independently of the rest of the sequence, **7**

effective contact order (ECO) effective contact order of a newly-added contact is defined as the effective loop closure size (the number of steps, covalent and non-covalent links taken along the shortest path on the polymer graph), given that other contacts have been formed, **24**

energy potential (*physics*) same as potential energy; energy which depends on mutual positions of bodies, **15**

energy barriers an increase in the objective function value that search has to overcome in order to move from the current candidate solution to another candidate solution with the same or lower (in the case of the minimization problem) objective function value, **27**

Energy Landscape Paving (ELP) adaptive stochastic local search algorithm classified as a generalized ensemble method in which the barrier height is decreased proportionally to the time the system stays in the minima (configurations are searched with time-dependent weights that take into account both the Boltzmann Probability and time spend in a particular local minima), **34**

entropy (*physics*) a measure of randomness or disorder of a system, **109**

ergodicity (or convergence) the process reaches an invariable (stationary) distribution in the limit, **29**

Evolutionary Algorithm (EA) population-based stochastic local search that is inspired by the process of evolution according to selection rules (usually has operations of mutations and recombination, **40**

face-centered cubic (FCC) lattice the usual lattice structure for the majority of crystalline metals, it has 12 base vectors, **17**

folding nucleus disjointed structure composed of protein residues that are important for the process of folding, **109**

funneled energy landscape the view of protein folding that proteins evolved to have the native state at the base of the search landscape funnel, as a consequence of the minimal frustration; it explains how most proteins fold efficiently and robustly to their functional native state and it allows robust prediction of folding kinetics, **28**

generalized ensemble Monte Carlo methods simulations in a generalized ensemble that strive to perform a random walk in potential energy space; the advantage of these methods is that from only one simulation run, one can obtain canonical ensemble averages of physical quantities as functions of temperature by the single-histogram and/or multiple-histogram re-weighting techniques, **31**

Genetic Algorithm (GA) population-based stochastic local search algorithm (a type of Evolutionary Algorithm) in which candidate solutions are encoded as genes (vectors of integers), and search strategy involves mutation (local

neighbourhood moves) and cross-over (among two or more candidate solutions) moves, **35**

global optimum a selection from a given domain which yields either the highest value or the lowest value (depending on the objective: either to minimize or to maximize), when a specific objective function is applied, **26**

globular proteins one of the three main protein classes, comprising globe-like proteins that are more or less soluble in aqueous solutions (where they form colloidal solutions); this main characteristic helps distinguishing them from fibrous and membrane proteins (the other classes), which are practically insoluble, **3**

Hamming distance the Hamming distance between two strings of equal length is the number of positions for which the corresponding symbols are different; thus, it measures the number of substitutions required to change one into the other, **79**

Hoyle paradox an inquiry asking how foldable proteins evolved within the lifetime of the universe, **5**

hydrogen bond hydrogen bonding occurs when an atom of hydrogen is attracted by strong forces to two atoms instead of only one, so that it may be considered to be acting as a bond between them; this typically occurs where the partially positively charged hydrogen atom lies between partially negatively charged oxygen and nitrogen atoms; although stronger than most other intermolecular forces, the typical hydrogen bond is much weaker than both the ionic bond and the covalent bond, **3**

hydrogen-deuterium exchange (H/X) a chemical reaction in which a covalently-bonded hydrogen atom is replaced by a deuterium atom, or vice versa; this method gives information about the solvent accessibility of various parts of the molecule, and thus the three-dimensional structure of the protein, **45**

hydrophilic having an affinity for water, **1**

hydrophobic having an aversion of water, **1**

hydrophobic collapse model a model that suggests that a protein rapidly collapses around its hydrophobic side-chains and then rearranges from the restricted conformational space, **44**

Hydrophobic Polar (HP) model a model of protein representation that classifies 20 amino acids as ether being hydrophobic or polar, **16**

Importance Sampling stochastic local search and sampling technique in which important values are emphasized by sampling more frequently; this estimator achieves reduction of variance; the basic methodology in Importance Sampling is to choose a distribution which encourages sampling of the important values more often, **38**

intensification an important property of the search process, that concentrates the search on the best candidate solutions (or solution components) found so far, and often uses "greedy" search strategies, **38**

kinetic control (*chemistry*) kinetic reaction control means that the reverse reaction does not occur or is slow; under kinetic control a product may be formed that is less stable but this product is formed faster because the activation energy for this reaction is lower, **6**

knowledge-based potential an energy potential that is based on quantities derived from a data base of known structures, **22**

lattice (*physics*) a physical model that is not defined on a continuum, but instead defined on a grid, which is a graph or an n-complex approximating space, **14**

lattice moves allowed move set on a lattice, **15**

learning-based potential an energy potential derived by maximizing the difference between correct and incorrect structures using machine learning and linear optimization techniques, **22**

Levinthal paradox an inquiry asking how can a protein find its low-energy state in time less than geological, since proteins have an astronomical number of possible conformations, **5**

local minimum a candidate solution whose objective function value is smaller or equal to the objective function values of any candidate solutions in its neighbourhood, **27**

long-range potential an energy contribution resulting from interaction of distant parts of the chain (contact-energy), **19**

marginal stability (Honig) paradox an inquiry that deals with the issue of why proteins seem to have such a delicate compensation of entropy and energy during intermediate stages of the folding process, **6**

Markov process the process in which the probability of transitioning between the current and a new state is only dependent on the current state and the energy difference between the current and the new state, **29**

model (in protein folding) the representation of a protein, **15**

model-based search search method that builds a parametric or a non-parametric model (an adaptive stochastic mechanism) of the search space that is updated during the search and the new candidate solutions are generated using the model, **36**

Molecular Dynamics (MD) molecular modeling technique that addresses numerical solutions of Newton's equations of motion on an atomistic or similar model of a molecular system to obtain information about its time-dependent properties, **26**

molten globule a stable, partially folded protein state found in mildly denaturing conditions such as low pH (generally pH = 2), mild denaturant, or high temperature; they are collapsed and generally have some native-like secondary structure but a dynamic tertiary structure; molten globules often are stable intermediate states during folding, **27**

monomer a small molecule that may become chemically bonded to other monomers to form a polymer, **16**

Monte Carlo (MC) method a widely-used class of stochastic local search algorithms for simulating the behavior of various physical systems; they are distinguished from other stochastic local search methods by making transitions between candidate solutions according to the Boltzmann probability and by the fact that the transition probability is only dependent by the energy difference between the current and a new state, **29**

move set a set of allowed moves; each move (search step) usually involves the modification, addition or removal of one or more solution components, **15**

Multicanonical Algorithm (MUCA) stochastic local search that belongs to a class of generalized ensemble methods where the transition probability between states is determined by the estimated density of states for a particular problem instance, **31**

N - terminus refers to the extremity of a protein or polypeptide terminated by an amino acid with a free amine group, **2**

native state the native state of a protein is its operative or functional form, **6**

neighbourhood relation an important component of a local search algorithm, a relationship that defines the direct neighbours of the current candidate solution, **39**

non-model-based search search method that generates new candidate solutions using solely the current solution or the current population of solutions, **36**

non-native (*biochemistry*) not a functional state of a protein, opposite to native, **15**

NP (non-deterministic polynomial time) is the set of decision problems solvable in polynomial time on a non-deterministic Turing machine; equivalently, it is the set of problems that can be verified by a deterministic Turing machine in polynomial time, **8**

NP-hard (non-deterministic polynomial-time hard) refers to the class of decision problems that contains all problems H, such that for every decision problem L in NP there exists a polynomial-time many-one reduction to H, written $L \leq_p H$; informally, this class can be described as containing the decision problems that are at least as hard as any problem in NP, **8**

nuclear magnetic resonance (NMR) spectroscopy the principal techniques used to obtain physical, chemical, electronic and structural information about a molecule in solution; it uses high magnetic fields and radio-frequency pulses to manipulate the spin states of nuclei, **9**

nucleation model a model that suggests that proteins have a small set of interactions (*folding nucleus*) common to most of the conformations in the transition state ensemble, **44**

objective function an evaluation function that assigns a numerical value to each candidate solution, **24**

off-lattice continuous model, or discrete model that does not use lattices, **15**

peptide the family of short molecules formed from the chain linking of various amino acids, **2**

peptide bond a chemical bond formed between two molecules when the carboxyl group of one molecule reacts with the amino group of the other molecule, releasing a molecule of water, **2**

physical (empirical) potential an energy potential that is based on empirical measurements of interactions between atoms and/or molecules, **22**

polar (of a compound) having an electric charge, **1**

polypeptide chain of amino acids; proteins are made up of one or more polypeptide molecules; the amino acids are linked covalently by peptide bonds, **4**

primary structure (*biochemistry*) linear sequence of amino acids in a given protein, **3**

prion an abbreviation for proteinaceous infectious particle, it is a unique type of infectious agent, made only of protein, **6**

protein chain-like polymer of small subunits (amino acids), **1**

protein folding problem problem of prediction of protein tertiary structure from its amino acid sequence for a given energy function, **14**

Pruned-Enriched Rosenbluth Method (PERM) stochastic local search that is based on Sequential Importance Sampling; additionally, it performs re-sampling after pruning and enrichment, this is a construction-based search method, **36**

quasi ergodicity a situation when trajectories between structurally diverse but statistically important energy minima have low (close to zero) transition probabilities that can result in the search or sampling process getting trapped in isolated minima; thus, the simulation may not necessarily reach the equilibrium and may not converge, **30**

quaternary structure (*biochemistry*) an arrangement of multiple folded protein molecules in a multi-subunit complex, **4**

R-group a side-chain; a part of an amino acid that is attached to a central alpha-carbon, **1**

Ramachandran plot a plot of permissible dihedral angles ϕ *vs.* ψ, **2**

random walk a process of taking successive steps, each in a random direction, **31**

random coil a polymer conformation where the monomer subunits are oriented randomly while still being bonded to adjacent units, **3**

Replica Exchange Monte Carlo (REMC) stochastic local search algorithm that belongs to a class of generalized ensemble methods; in this method a number of Monte Carlo simulations are performed at different temperatures between T_{low} to T_{high}, and simulations (sampling processes) with nearby temperatures attempt to exchange temperatures every specified number of steps; the exchange between simulations as well as within simulation is performed according to the Boltzmann probability, **33**

reptation a snake-like motion of the polymer happening by diffusion of stored length along its own contour, **38**

residue (*chemistry*) a portion of a large molecule, such as protein; also designates an amino acid in the protein folding literature, **2**

root mean square deviation (RMSD) (as applied to biological structures) the measure of the average distance between the backbones of superimposed proteins, **9**

rugged landscape (ruggedness) a property of a search landscape that describes the fluctuation of the objective function values of candidate solutions that are neighbours according to the neighbourhood relation chosen, **27**

run-time distribution probability distribution of the time required by a stochastic algorithm to solve a given problem instance, **71**

sampling a process whose goal is to generate a correct ensemble (usually defined by the Boltzmann probability in physical systems), **26**

search space a set of all candidate solutions (conformations) for a given problem instance, **20**

search landscape a mathematical concept consisting of the search space, the neighbourhood relation and the objective function for a given problem instance; a search landscape is characterized by such properties as: local minima distribution, global minima distribution, ruggedness, **26**

search methods algorithms used to search the space of all possible candidate solutions, **26**

searching a process whose goal is to find a configuration or a set of configurations with a required property (usually a global energy minimum), **26**

secondary structure (*biochemistry*) local structure defined by the hydrogen bonds of the biopolymer; in proteins, the secondary structure is defined by patterns of hydrogen bonds between backbone amide groups; in nucleic acids, the secondary structure is defined by the hydrogen bonding between the nitrogenous bases, **3**

self-avoidance (for a chain) a state of non-collision with itself, **15**

Sequential Importance Sampling stochastic local search method and sampling technique similar to Importance Sampling, the only difference being that weights that control transition probabilities between states are computed sequentially as candidate solutions are built (this is a construction-based search), **38**

short-range potential an energy contribution resulting from interaction of neighbouring amino acids, **18**

Simulated Annealing(SA) a stochastic local search algorithm that is inspired by the process of crystallization in physics; it follows the Boltzmann acceptance probability but unlike in Monte Carlo methods the temperature is lowered according to a specified cooling (annealing) schedule, **28**

square lattice one of the five two-dimensional lattice types composed of upright squares, **16**

state-of-the-art methods best performing methods for a problem of interest, **16**

steric hindrance an affect arising from the fact that each atom within a molecule occupies a certain amount of space; if atoms are brought too close together, there is an associated cost in energy due to overlapping electron clouds, and this may affect the molecule's preferred shape (conformation), **2**

stigmergy a method of communication in multi-agent systems in which the individual agents communicate with one another by modifying their local environment, **47**

stochastic local search (SLS) local search algorithm that generates and/or selects candidate solutions (or partial candidate solutions – components of a solution) using randomization techniques, **26**

Tabu Search stochastic local search method that exploits memory of the search history for guiding the search, **34**

tertiary structure (*biochemistry*) the tertiary structure of a protein is its overall shape, also known as its fold, **4**

Thermodynamic Hypothesis a hypothesis that states that the native state represents the ground state of lowest Gibbs free energy, **6**

topology of the search space a distribution of various properties for all of the possible candidate solutions of a problem (for example, the distribution of objective function values, their connectedness), **26**

transition state the transition state of a chemical reaction is a particular configuration along the reaction coordinate; it is defined as the state corresponding to the highest energy along the reaction coordinate, **24**

Umbrella Sampling stochastic local search and sampling method in which an additional biasing potential is added during the simulation to the canonical simulation (usually a particular umbrellas requires knowledge of the system or the exact energy surface in some cases, the only umbrella that does not require this is temperature), **37**

X-ray crystallography a technique in crystallography in which the pattern produced by the diffraction of X-rays through the closely spaced lattice of atoms in a crystal is recorded and then analyzed to reveal the three-dimensional shape of a molecule, **9**