Fast Stochastic Global Optimization Methods and Their Applications to Cluster Crystallization and Protein Folding

by

Lixin Zhan

A thesis
presented to the University of Waterloo
in fulfilment of the
thesis requirement for the degree of
Doctor of Philosophy
in
Physics

Waterloo, Ontario, Canada, 2005

AUTHOR'S DECLARATION FOR ELECTRONIC SUBMISSION OF A THESIS

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Abstract

Two global optimization methods are proposed in this thesis. They are the multicanonical basin hopping (MUBH) method and the basin paving (BP) method.

The MUBH method combines the basin hopping (BH) method, which can be used to efficiently map out an energy landscape associated with local minima, with the multicanonical Monte Carlo (MUCA) method, which encourages the system to move out of energy traps during the computation. It is found to be more efficient than the original BH method when applied to the Lennard-Jones systems containing 150–185 particles.

The asynchronous multicanonical basin hopping (AMUBH) method, a parallelization of the MUBH method, is also implemented using the message passing interface (MPI) to take advantage of the full usage of multiprocessors in either a homogeneous or a heterogeneous computational environment. AMUBH, MUBH and BH are used together to find the global minimum structures for Co nanoclusters with system size $N \leq 200$.

The BP method is based on the BH method and the idea of the energy landscape paving (ELP) strategy. In comparison with the acceptance scheme of the ELP method, moving towards the low energy region is enhanced and no low energy configuration may be missed during the simulation. The applications to both the pentapeptide Met-enkephalin and the villin subdomain HP-36 locate new configurations having energies lower than those determined previously.

The MUBH, BP and BH methods are further employed to search for the global minimum structures of several proteins/peptides using the ECEPP/2 and ECEPP/3 force fields. These two force fields may produce global minima with different structures. The present study indicates that the global minimum determination from ECEPP/3 prefers helical structures. Also discussed in this thesis is the effect of the environment on the formation of beta hairpins.

Acknowledgements

I wish to express my deepest gratitude to my supervisor, Professor Wing-Ki Liu. His patience, encouragement, and enthusiasm make my Ph.D. experience enjoyable. I am grateful for his suggestion to study the global minimization of cluster systems, which ultimately leads to this thesis. His insight into physical problems has enhanced my understanding of this work. Most importantly, he helped me develop the correct attitude towards scientific research. I benefit much in both spoken and written English from constant interaction with him in different manners for all these four years.

I would like to express my sincere thanks to Professor Jeff Z. Y. Chen as well. His guidance led me to the study of Monte Carlo simulation and protein folding. Discussion with him always gave me a deeper understanding of physics. His help in the development of my scientific attitude and the improvement of my English is also greatly appreciated. Further thanks go to my advisory committee, Professors F. McCourt and G. Tenti, for their insightful suggestions.

My friends in both the Department of Physics and the Department of Electrical and Computer Engineering have contributed much to made my life wonderful during the graduate period. I am indebted to them.

Financial support in the form of teaching assistantship from the Department of Physics, University of Waterloo, and research assistantships from NSERC is gratefully acknowledged. Computational time provided by the Hydra beowulf cluster and SHARCNET is greatly appreciated.

Finally, I want to thank Professors Wing-Ki Liu and Jeff Z. Y. Chen again for their critical reading of the manuscript of this thesis.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Global Optimization Methods

Global optimization of a multi-variable problem has long been an intensive subject of research in many fields and is an important issue in the characterization of complex systems [1, 2]. Obvious applications with significant importance include the design of integrated circuits [3] such as microprocessors, the prediction of protein structures [1, 4, 5], *ab initio* computation of nano-size atomic structures [1, 2, 6, 7] and optimization in transportation systems [8]. From a physical point of view, global optimization can be interpreted as a global minimization procedure of physical systems, when they can be effectively described by multivariable functions, even though these functions may have very complex forms. The global optimization will then be performed on such functions. By treating the targeted function as an effective "potential energy", and the dependent variables as the coordinates of particles, searching the global optimum is equivalent to locating the classical ground state of a physical system.

For a physical system, the potential energy function is often very complicated, making it impossible to obtain the global minimum analytically. Numerical solution of the system will then be the only feasible way thanks to the power and availability of modern computers. Since most of the global minimization procedures are time consuming, efficient algorithms are required in practical applications. The develop-

ment of efficient algorithms for global minimization is still quite active because of this requirement. The algorithms of optimization can be classified into four overlapping categories [9, 10, 11]: (1) deterministic methods, (2) stochastic methods, (3) heuristic methods, and (4) smoothing methods. Most of the techniques exhibit varying degrees of success in applying to the corresponding physical systems, for which the numerical techniques are specifically designed. The main difficulty in global minimization is associated with the fact that multiple local minima may exist, with locations separated from each other by high energy barriers. Most numerical procedures are effective in finding a minimum; however, not all numerical procedures are efficient in finding the global minimum.

Methods that employ the Monte Carlo procedure, which belong to the second category, are based on the statistical aspect of a physical problem and have become one of the major branches in the field of optimization. For complex systems where a large number of energy minima exist, these methods are effective. Since the first appearance of the Monte Carlo method half a century ago [12], there have been many different implementations for various fields of applications. Of central importance to all the Monte Carlo methods is the statistical weight associated with each point of the system coordinate, depending on the potential energy of the system. The simulated annealing (SA) method [3] is the most commonly used algorithm in which a Boltzmann weight has been adopted, where temperature plays a pivotal role in determining the thermodynamic properties. Initially, system configurations are generated at high temperature, simulating the melted state of the system. Following a prescribed cooling schedule, the temperature is lowered in stages until the system freezes at a low-temperature solid state, corresponding to an energy minimum. A well chosen cooling strategy encourages the simulation to yield a crystalline state close to the global energy minimum. However, because of the fluctuation nature of the thermodynamics, SA cannot give the precise value of the global energy minimum. Furthermore, it is also possible that the system can be trapped in an undesirable local energy minimum.

The introduction of non-Boltzmann weighting schemes into Monte Carlo methods by Torrie and Valleau [13] offered a new strategy in the traditional MC method, which partially resolved the trapping issue. The basic idea behind a re-weighting

scheme is to introduce a statistical weight other than the canonical Boltzmann one, to ensure a more extensive searching of the low energy space to improve the chance of reaching the global minimum. An ideal choice of the weight is the inverse of the density of states (DoS) of energy, which would lead to the desired uniform random walk trajectory over the entire energy space. However, the density of states is unknown *a priori*. First generating an approximation to the weight, and hence the density of states, the multicanonical Monte Carlo (MUCA) method improves the estimate iteratively using statistics accumulated during a pre-defined number of MC steps [14, 15, 16, 17, 18, 19, 20, 21, 22, 23]. The approximate location of the energy minimum will then be determined by further MC iterations based on the multicanonical weight just obtained.

There are other Monte Carlo based methods that can be used to pin down the precise value of the energy minima. The basin hopping (BH) method [2], which is in principle the same as the "Monte Carlo Minimization" (MCM) method of Li and Scheraga [24], is the combination of a Monte Carlo method based on the Boltzmann weight with a deterministic local minimization procedure. The original configuration is replaced by its nearest local minimum configuration, precisely computed from a deterministic procedure. Using the local minimum rather than the original energy potential, such a scheme maps the energy landscape into a staircase form where plateaus are the local energy minima. A Boltzmann weight is then used to move the system from plateau to plateau, in the transformed energy space.

Both MUCA and BH have been shown to be successful in their applications to finding stable structures of crystalline clusters [1, 2, 25, 26, 27, 28, 29] and predicting protein native structures [1, 19, 21, 23, 24, 30]. MUCA, inasmuch as SA, has some difficulties in locating the energy minimum accurately because of the thermodynamic nature of the methods. In BH, though each new energy minimum is precisely determined, the hopping between the energy minima can still be trapped in a deep minimum area for a long time, which is an inherited problem of the finite-temperature Monte Carlo method. When the system size is large, it becomes difficult to reach the global minimum within a reasonable computational time.

To overcome this deficiency, we introduce in Chapter 2 a new optimization algorithm, the multicanonical basin hopping (MUBH) method [6], that incorporates

a multicanonical weight into the basin-hopping method. Instead of using the Boltzmann weight that relies on a fixed temperature, MUBH takes the main idea from MUCA by using the multicanonical weight but this weight, as in BH, is based on the nearby local minimum of each configuration visited, calculated deterministically. In other words, MUBH is a MUCA method based on the reduced energy landscape of BH. As will be demonstrated in Chapter 2, this new algorithm shows substantial improvement in efficiency over BH for relatively large clusters when applied to the Lennard-Jones (LJ) systems. Success is mainly the result of avoiding the potential pitfalls in the original MUCA and BH methods.

Beyond the improvement of computational algorithms, we can also take advantage of the distributed computation environment that can carry out a single computational task on multiple processors. A typical Monte Carlo algorithm is a Markov-chain procedure in which every new step is generated from the previous step. The acceptance or rejection of a step is determined by the relative weight generated from the current step in comparison with that from the previous step, according to a selection rule, such as the Metropolis criterion. Our approach is based on several different simultaneous Markov-chains [31, 32, 33]. Starting with different initial (random) conditions for multiple Markov-chains, each Markov-chain runs independently on one processor. Because of the stochastic nature of these optimization methods, this would multiplicatively increase the probability of finding the final result; hence the computational time would be shortened. The replica exchange method (REM) [32], also referred to as the parallel tempering method [34], is a good example of the multiple Markov-chain application. We will present the asynchronous multicanonical basin hopping (AMUBH) method [7] in Chapter 3, which is a parallel implementation of the MUBH method. A single computation is carried out over multiple processors, each carrying out one independent computation starting from a different initial condition. AMUBH combines the statistical histograms collected from all processors for occasional update of the multicanonical weight, which is then distributed to each processor for continuing calculations. Running threads are not required to finish synchronously for the update.

Recently, the Energy Landscape Paving (ELP) method, a novel approach to the global optimization problem that combines ideas from tabu search [35, 36, 37] and

energy landscape deformation [38, 39] was proposed by Hansmann and Wille [40]. ELP has very general applicability. The central idea is to perform low-temperature Monte Carlo simulations, but with a modified energy expression designed to steer the search away from regions that have already been visited. When applied to the X-ray structure determination of organic molecules [41], it was shown that ELP outperforms the simulated annealing method, and when it was utilized for the simulation of the heptapeptide deltorphin [42], the ELP method was proven to be more effective in sampling the low energy region of conformational space than the multicanonical (MUCA) method. In fact, any Monte Carlo method that updates its weight based on the collected histograms in previous steps can be considered as a typical implementation of the ELP method. Hence, the generalized ensemble methods (MUCA, entropic sampling, histogram reweighting, etc.) are all different implementations of the ELP method. Hansmann's implementation in Ref. [40] used the simplest functional form of the energy dependent histogram. Just as the SA method and the MUCA method discussed earlier, ELP cannot obtain the exact minima value as well, since the temperature cannot be set to absolute zero to eliminate all the thermal fluctuations. Similar to the MUBH method, we will introduce the local minimization procedure into the ELP method, i.e., combine BH with ELP. This combination is also expected to have good efficiency for global optimization problems. In our actual implementation of the combination procedure, a modification is introduced to the acceptance criteria for a new MC step so that no lower energy configurations would be missed. Its implementation will be discussed in detail in Chapter 4 as the basin paving (BP) method. Being a more general global optimization method, BP can be applied to both cluster crystallization and protein folding problems.

1.2 Cluster Crystallization

The study of nanoclusters composed of either metallic atoms, nonmetallic atoms or their mixtures is of fundamental importance [2, 26, 27, 43, 44, 45, 46, 47, 48, 49] to nanotechnology where unusual physical and chemical properties depend strongly on cluster size N. Even changing a single atom in the cluster may dramatically

alter the nanocluster properties, such as the specific heat and the magnetic susceptibility [44, 46, 47]. Cluster systems have been widely studied both theoretically and experimentally. The crystal structure of a cluster depends on the interaction between particles. For inert gas atoms, e.g. Ar, He and Ne, a Lennard-Jones potential is often employed to describe the interaction between any two atoms of the system. We call a system described solely by Lennard-Jones interactions between any pair of particles the Lennard-Jones cluster. For Lennard-Jones clusters, global minima have been computationally obtained with size N up to 1 000 [2, 50, 51, 52, 53, 54, 55, 56]. However, only Ref. [2] performed an unbiased configuration search utilizing the basin hopping algorithm to system size up to 150. The other studies are based on lattice models, which are only fast in determining configurations based on lattices already constructed. If the global minimum configuration of a cluster is different from all the known lattice structures, it may never be located using lattice based methods. For example, by using the unbiased BH method, a new Leary's tetrahedral structure was recently determined for the Lennard-Jones cluster of size $N = 98$, which had never been found by any lattice-based method previously. With the new proposed unbiased MUBH method, we were able to perform large system simulations within a reasonable time. This is illustrated by the comparison of MUBH and BH in Chapter 2. Other than the Lennard-Jones potential, there are other potentials, such as the Morse potential and the Gupta potential, that are often used for cluster system studies. The Gupta potential [57, 58], which was first proposed for studying metal surface relaxation, is now mainly adopted to study metal clusters [26, 45, 48, 52, 58]. In Chapter 3, cobalt nanoclusters will be studied using the Gupta potential. This study is a general purpose global minimum searching procedure based on BH, MUBH and AMUBH. For small size clusters ($2 \leq N \leq 150$), the BH method is efficient enough; MUBH is applied when the system size is in the range of $150 < N \leq 180$, since in this range, the efficiency of MUBH is dramatically improved in comparison with BH alone; for systems with much larger sizes ($180 < N \leq 200$), AMUBH, the asynchronous parallelized version of MUBH, is adopted for the purpose of saving computational time with the usage of multiple processors in a computer network.

1.3 Protein Folding

Proteins are one of the most important and common macromolecules that make up the primary constituents of life. From a chemical point of view, they are unbranched chains joined by some or all of the twenty naturally occurring amino acids. A protein is only functional when it folds into a typical spatial structure, which is called the native state. The three dimensional (3D) configuration of a protein is solely determined by its amino acid sequence. Protein folding problems, which predict the spatial structure and the corresponding function of a protein from its amino acid sequence alone, are then critical in biological studies. When proteins cannot fold correctly, there can be serious diseases caused. Alzheimer's, Parkinson's, mad cow diseases and many cancers are believed to be caused by the misfolding of proteins [59, 60].

Experimental techniques often used in determining a protein's native structure are X-ray crystallography [61, 62] and nuclear magnetic resonance (NMR) [61, 63]. Although these experimental methods can provide high-resolution structural information about some proteins, computer simulations can be used to obtain valuable information that cannot be obtained experimentally. Furthermore, the folding dynamical, kinetic and stochastic properties of the folding procedure can be studied as well.

Comparative modeling methods [5, 64, 65] predict the protein structures based on the fact that there are obvious similarities between the 3D structures of some proteins. It consists of four sequential steps to predict the structure [65]: (1) Templates, which are the known structures related to the target sequence, must be first found usually from a database in this step. (2) The target sequence is aligned with the templates. (3) Based on the alignment, a three dimensional (3D) model for the target protein will be constructed. (4) The model will be evaluated for its folding correctness and overall model accuracy. If the model is not satisfactory, these four steps are repeated until an acceptable model is obtained. Comparative modeling can generate structures with high accuracy. However, at least one known structure is required for its successful application.

Calculations from "first principles" can also predict the native structure of a

protein, although its accuracy and reliability are inferior to comparative methods. According to the thermodynamic hypothesis [66, 67], the native structure of a protein lies at the global minimum of free energy, which can further be approximated by the global energy minimum. Predicting the native state of a protein can then be interpreted as a global minimization procedure. Most of the existing global optimization techniques can be used for the protein folding problem. In this thesis, the recently proposed MUBH and BP method will be applied to search for the lowest energy structures of different proteins/peptides.

For the present simulations, two empirical force fields ECEPP/2 [68, 69] and ECEPP/3 [70] are employed to describe the interatomic interactions of the system studied. After the introduction of the BP algorithm in Chapter 4, we will apply it to locate the lowest energy structures of the pentapeptide Met-enkephalin and the villin subdomain HP-36 systems. MUBH, BP, and BH methods are used in Chapter 5 to study several proteins/peptides. The study covers the formation of alpha helix, beta hairpin and random coil structures. As we will notice later in the chapter, simulations using the ECEPP/2 and ECEPP/3 force fields generate different structures even for the same system. By comparing the difference between the structures located, the effects introduced by the difference between the potentials to the folding mechanism of the proteins will be discussed. Environmental interactions have close relationships with the final folded structures. The formation of a beta sheet often requires the presence of solvent or "background" proteins. Such environmental effects will also be studied in Chapter 5 of this thesis.

Finally, in Chapter 6, we summarize the results obtained in this thesis and provide a list of suggestions for future work.

We include in Appendix A a brief review on protein folding and list the 20 amino acids that occur in proteins in Appendix B. The acronyms used in this thesis are summarized in Appendix C.

Chapter 2

Multicanonical Basin Hopping Method

As mentioned in Chapter 1, global optimization plays a very important role in many fields. A large number of computational algorithms have been developed to perform the optimization in the past years. A practical system can often be described by a "potential energy" function, which could be very complicated in general. A typical searching strategy will be applied to the complex "potential energy surface" (PES) with the expectation of locating the global "energy minimum" during a reasonable computational time. From the PES's point of view, a searching procedure can be based on the original energy surface or on a transformed energy surface for the purpose of simplifying the original one, which may result in the improvement of efficiency to reach the global minimum (optimum). From the searching procedure's point of view, the simulation often follows two directions. One uses the molecular dynamics (MD) method. By solving the equations of motion for each particle in a classical system according to Newton's second law $\mathbf{F} = m\frac{d^2\mathbf{r}}{dt^2}$, MD simulation computes the trajectory that the system follows starting from given initial conditions. The MD method is a good choice for investigating the kinetic properties of the system studied. However, MD is only applicable to systems of small size at the present stage because of the intensive computational resources it requires. If only the global minimum is of interest, it is generally not the best choice. The

other direction uses the Monte Carlo (MC) method. For the MC method, instead of following the real trajectory as MD does, it relies on the statistical aspect of a physical system. By assigning a statistical weight to each state visited, for example the Boltzmann weight $\exp(-E/k_BT)$ with E the energy of the state, the acceptance of a new step is decided by the comparison of the weight with that of its previous step. Even though the trajectory for a MC simulation is not the dynamical one that the system will follow, the statistical quantities can still be extracted when the simulation is performed long enough. This method cannot give as much information as MD can. However, it is generally more efficient than the MD simulation. When obtaining the global minimum is the main purpose, the MC method is often the first choice.

The Metropolis Monte Carlo method [12] is the most common MC method used in obtaining the statistical quantities of the system studied. For this algorithm, the simulation starts from a randomly generated initial configuration $\mathbf{r}_0$, with energy $E(\mathbf{r}_0)$ and the Boltzmann weight $w(\mathbf{r}_0) = \exp[-E(\mathbf{r}_0)/k_BT]$, where k_B is the Boltzmann constant and T is the simulation temperature. A random deviation $\Delta\mathbf{r}$ is then performed so that the system reaches a new configuration $\mathbf{r}_1 = \mathbf{r}_0 + \Delta\mathbf{r}$, with the energy $E(\mathbf{r}_1)$ and the weight $w(\mathbf{r}_1) = \exp[-E(\mathbf{r}_1)/k_BT]$. The acceptance of the new configuration is determined by comparing the weight of the two configurations,

$$\frac{w(\mathbf{r}_1)}{w(\mathbf{r}_0)} = e^{-(E(\mathbf{r}_1)-E(\mathbf{r}_0))/k_BT} = e^{-\Delta E/k_BT}, \tag{2.1}$$

with $\Delta E = E(\mathbf{r}_1) - E(\mathbf{r}_0)$. If $\Delta E < 0$, the new configuration $\mathbf{r}_1$ is accepted unconditionally; if $\Delta E > 0$, $\mathbf{r}_1$ is accepted with probability $P = \exp(-\Delta E/k_BT)$. In the latter case, a random number ξ, where $0 \leq \xi < 1$, will be generated: if $\xi \leq P$, the trial configuration is accepted; otherwise, it is rejected and the simulation keeps the old configuration. Since the probability of finding ξ having a value less than P is P, this procedure guarantees that an appropriate probability is generated. The state in which the system now resides will be treated as the initial configuration for the next MC step. The procedure is repeated iteratively until the pre-set terminating conditions are satisfied. In a practical simulation, however, the Metropolis MC method is not efficient in finding the global minimum because its population distribution is bell-shaped, as will be mentioned in Section 2.2. This

kind of distribution discourages the simulation from going to the lower energy region, which results in the insufficient sampling of lower energy configurations, and the difficulty of surpassing high energy barriers, which are often the main problems for most Monte Carlo methods to overcome.

The simulated annealing (SA) method is probably the most widely used MC method for global minimization since it was first proposed in 1983 by Kirkpatrick *et al.* [3]. SA is based on the Metropolis MC method by introducing a variable temperature to simulate the slow cooling (annealing) procedure of a physical system. A practical example is the crystal growing process. The system is first equilibrated at a high temperature, which corresponds to the gaseous state, and then slowly cooled down to a very low temperature close to zero, in analogy with the transition from the gaseous state to the liquid state and then to the solid state. If the cooling scheme is well chosen, this procedure will ensure that the system cools down to the crystal state, which gives the global minimum. Otherwise, the system may be stuck in a glassy state, a local minimum, and will never be able to reach the crystal state. The annealing scheme affects the success of locating the global minimum and the convergence speed directly. Other than the conventional annealing schedule, in which the temperature is controlled simply by $T_{k+1} = cT_k$ with k the annealing step and c the annealing ratio satisfying $0 < c < 1$, there are some other schedules proposed as well, for instance, Cauchy annealing, Boltzmann annealing, and adaptive simulated annealing (ASA) [71, 72]. Amongst them, ASA draws much attention due to the fact that it is suitable for less known systems and has been proven to be more robust than other annealing techniques when applied to complex problems with multiple local minima.

As just illustrated, the efficiency of a MC method when applied to the global minimization problem is mainly determined by its ability of surpassing high energy barriers and visiting lower energy regions in more detail. Different MC implementations adopt different techniques to solve this problem, which results in different application efficiencies. The basin hopping (BH) method [2] achieves this goal by removing the energy peaks via a local minimization procedure in each MC step, while generalized ensemble methods overcome the difficulties by setting different weights for different energy regions. We will give a detailed review of these two methods

in the following sections, since our new proposed method, the multicanonical basin hopping (MUBH) method, is based on them.

2.1 Review of Basin Hopping Method

Consider the potential energy of a system, $E(\mathbf{r})$, which is a function of all molecular coordinates, $\mathbf{r}$. The dotted curve in Fig. 2.1 schematically shows the presence of multiple minima of the system. Initially, a random configuration $\mathbf{r}$ is chosen as the starting point, from which the configuration of the local minimum, $\mathbf{r}_{\min}$, is determined numerically from a minimization procedure, such as the conjugate gradient method [73]. Then $\mathbf{r}_{\min}$ is given a small trial "move" to a new configuration $\mathbf{r}'$ not far from $\mathbf{r}_{\min}$, and a new local minimum $\mathbf{r}'_{\min}$ is again obtained from the minimization procedure. One determines the acceptance of this new minimum configuration according to the Metropolis scheme [12] by considering a Boltzmann weight $\exp(-\Delta E_{\min}/k_B T)$, where $\Delta E_{\min}$ is the energy difference between the new minimum and the minimum already found at the previous step. This procedure is repeated to search for the next local minimum. Computationally the local minimization procedure is the bottleneck of the BH method. We found that the limited-memory quasi-Newton optimization method [74] shows better performance in comparison with other techniques discussed in Ref. [73]. Any small improvement at this step would see repeated saving of the multi-iteration computational time.

Conceptually, the local-minimum searching step in the above algorithm is equivalent to transforming the energy landscape, represented by the dotted curve in Fig. 2.1, to a new reduced energy landscape, which consists of plateaus of energy minima only:

$$\tilde{E}(\mathbf{r}) = \min\{E(\mathbf{r})\} = E(\mathbf{r}_{\min}) \tag{2.2}$$

where $\min\{...\}$ represents an energy minimization process by using $\mathbf{r}$ as the initial condition and $\mathbf{r}_{\min}$ is the configuration of the local minimum obtained from $\mathbf{r}$ [2]. The energy maxima of the original function are discarded in the reduced energy landscape and are no longer of concern, as only the structure of the "basins", represented by the solid curve in Fig. 2.1, is examined.

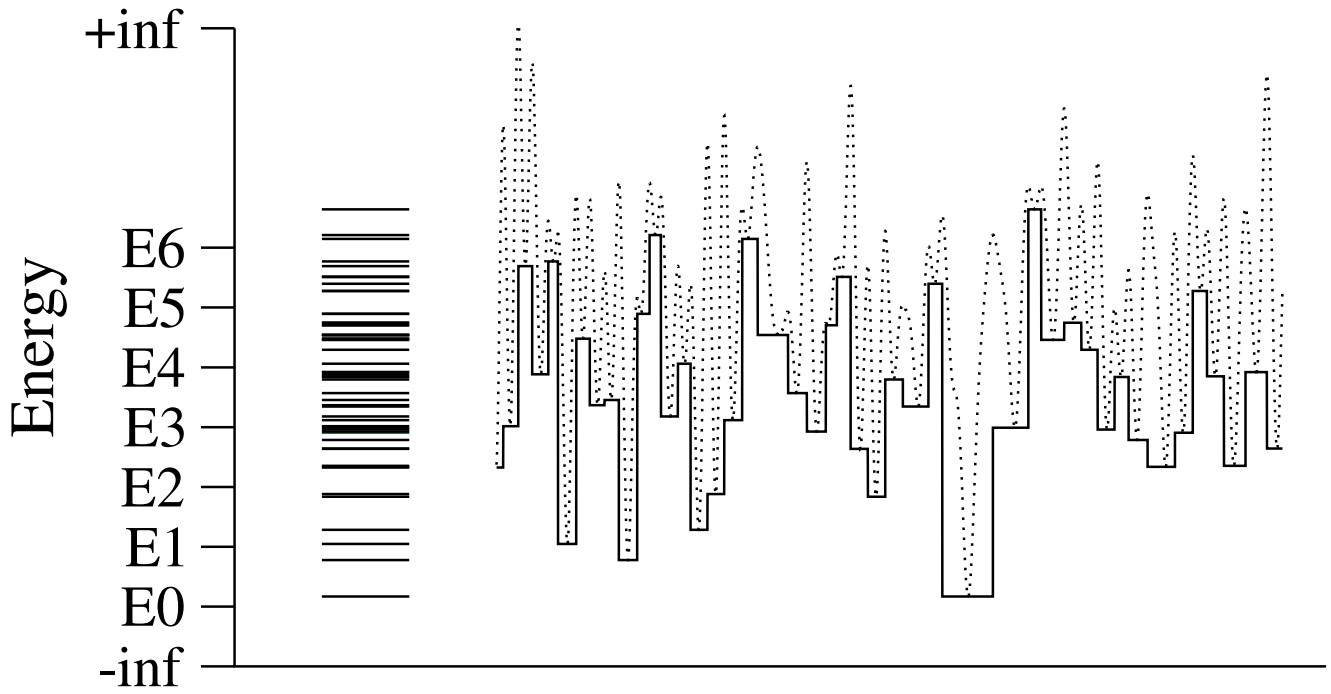


Figure 2.1: A schematic illustration of the concept of the basin-hopping energy transformation in one-dimension. The dotted curve shows an original function under consideration and the steps drawn by the solid line represent the local minima of the original curve, under the transformation in Eq. (2.2). The transformed energy landscape can be described by the energy spectrum to the left. Also shown is the division of the energy axis into various bins considered in the multicanonical procedure.

The reduced energy landscape is now a multi-step function that has values of the local energy minima only. As can be seen in Fig. 2.1, there still exist energy barriers in the new landscape, where the new maxima, which are actually the local minima of the original function, separate deeper minimum wells. The Monte Carlo part of the BH algorithm is used to handle the hopping of the system from one plateau to another under a thermal energy k_BT. The hopping probability depends highly on the reduced-energy difference between the plateaus of the two consecutive steps and the choice of k_BT. Hence, simulations can still be trapped in the reduced energy landscape.

The basin-hopping method has been demonstrated to be superior to other techniques when applied to small size Lennard-Jones systems [2]. However, as remarked in [2], the efficiency of the approach "could doubtless be improved by combining it with various other techniques". One such improvement would be to use an annealing schedule to induce a temperature reduction process — a simulated annealing and BH method; the other possibility would be to employ a non-Boltzmann weight in the Monte Carlo part of the simulation, as will be shown below.

2.2 Multicanonical Monte Carlo Method

In a canonical ensemble, configurations at temperature T are weighted by the Boltzmann factor

$$w_B(E) = e^{-\beta E} \tag{2.3}$$

where $\beta = (k_BT)^{-1}$. The resulting probability distribution of the energy, experienced by the system in the simulation, takes the form

$$P_B(E,T) \propto \rho(E) w_B(E) \tag{2.4}$$

where $\rho(E)$ is the density of states, which increases with energy rapidly. In the low energy harmonic region and for a single minimum, $\rho(E) \propto (E - E_0)^{n_F/2}$, where n_F is the system degrees of freedom and E_0 its ground state energy [75]. The Boltzmann factor decreases with energy exponentially for a given temperature. Thus $P_B(E,T)$ is strongly peaked at the average energy of the system corresponding

to that temperature. Ideally, one uses a very low T to search for configurations near the ground state. However, a low T prevents the actual simulation from moving efficiently between the energy wells and barriers. Berg [16] has shown that $P_B(E,T)$ cannot be computed accurately due to the poor sampling in a canonical system.

The multicanonical ensemble, on the other hand, is designed differently, where the weight function used in the Monte Carlo simulation is directly related to the density of states by

$$w_{\rm mu}(E) \propto 1/\rho(E)\ . \tag{2.5}$$

The resulting probability distribution of the energy is

$$P_{mu}(E) \propto \rho(E) w_{\rm mu}(E) = {\rm const}, \tag{2.6}$$

i.e., the system is expected to move throughout the entire energy space by a random walk.

Without loss of generality, we simply set the constant of proportionality to unity in Eq. (2.5) because only relative probabilities are required in the Metropolis scheme. However, the density of states of a physical system, and thus the multicanonical weight, is unknown *a priori*, and needs to be estimated via iterated numerical simulations. The starting iteration, the zeroth iteration, is usually performed by adopting the Boltzmann weight as the initial guess for $w_{\rm mu}(E)$, $w_{\rm mu}^{(0)}(E) \equiv w_B = \exp(-\beta^{(0)}E)$ where $\beta^{(0)} = (k_B T^{(0)})^{-1}$ with $T^{(0)}$ a given initial temperature, and an energy histogram $H^{(1)}(E)$ is constructed from the Monte Carlo sampling. Because we expect that $H^{(1)}(E) \propto w_{\rm mu}^{(0)}(E)\rho(E)$ so that $\rho(E) \propto H^{(1)}(E)/w_{\rm mu}^{(0)}(E)$, Eq. (2.5) shows that an improved estimate of the multicanonical weight can be obtained from $w_{\rm mu}^{(1)}(E) = 1/\rho(E) \approx w_{\rm mu}^{(0)}(E)/H^{(1)}(E)$. This procedure is repeated so that at the nth iteration, the simulation is carried out with the estimated weight $w_{\rm mu}^{(n)}(E)$, which yields a distribution $H^{(n+1)}(E)$ for E, collected within the nth iteration. A new estimate for the statistical weight

$$w_{\rm mu}^{(n+1)}(E) \approx w_{\rm mu}^{(n)}(E)/H^{(n+1)}(E) \tag{2.7}$$

is then used in the $(n+1)$th iteration.

In practice, one estimates an energy lower bound E_0 and an energy upper bound E_L from the first MC run, and divides the energy region of interest into L bins

having a bin width Δ. Each bin carries a label i and is characterized by its upper energy E_i. We further define the zeroth bin for energy $E \leq E_0$ and the $(L+1)$th bin for energy $E > E_L$. The statistics can now be collected for each bin, and $H^{(n+1)}(E_i)$ can be numerically defined as the number of states appearing in the ith bin during the nth iteration.

To use the histogram effectively in the next iteration in which a smooth $w_{\rm mu}^{(n+1)}(E)$ is required, the system entropy $S(E) \equiv -\ln w_{\rm mu}(E)$ in the ith bin can be parameterized, following Berg's scheme [17, 18, 19], as

$$S_i(E) = \beta_i E - \alpha_i \qquad \text{for} \qquad E_{i-1} < E \leq E_i, \tag{2.8}$$

where β is the derivative of entropy $S(E)$ with respect to energy E, with discrete expression

$$\beta_i = \frac{S_{i+1}(E_{i+1}) - S_i(E_i)}{\Delta}\,. \tag{2.9}$$

β_i and α_i in Eqs. (2.8) and (2.9) stand for $\beta(E_i)$ and $\alpha(E_i)$, respectively, in bin i.

By inserting the entropy equivalent of Eq. (2.7), i.e. $S^{(n+1)}(E) = S^{(n)}(E) + \ln H^{(n+1)}(E)$, into Eq. (2.9), we will be able to obtain an iterative expression for β_i at the $(n+1)$th step,

$$\begin{aligned}
\beta_{i,0}^{(n+1)} &= \frac{S_{i+1}^{(n+1)}(E_{i+1}) - S_i^{(n+1)}(E_i)}{\Delta} \\
&= \frac{S_{i+1}^{(n)}(E_{i+1}) - S_i^{(n)}(E_i)}{\Delta} + \frac{\ln H_{i+1}^{(n+1)} - \ln H_i^{(n+1)}}{\Delta} \\
&= \beta_i^{(n)} + \frac{\ln H_{i+1}^{(n+1)} - \ln H_i^{(n+1)}}{\Delta}
\end{aligned} \tag{2.10}$$

Due to the statistical uncertainty and fluctuation of the collected histograms, a subscript "$_0$" has been used in the previous equation to denote that $\beta_{i,0}^{(n+1)}$ is not the final estimator yet. A correction needs to be performed to obtain the final $\beta_i^{(n+1)}$.

The variance of Eq. (2.10) can be estimated as

$$\sigma^2[\beta_{i,0}^{(n+1)}] = \sigma^2[\beta_i^{(n)}] + \frac{\sigma^2[\ln H_{i+1}^{(n+1)}] + \sigma^2[\ln H_i^{(n+1)}]}{\Delta}\,. \tag{2.11}$$

Since $\beta_i^{(n)}$ is a fixed quantity obtained from the nth step and it does not fluctuate, we have $\sigma^2[\beta_i^{(n)}] = 0$. The fluctuation of $\beta_{i,0}^{(n+1)}$ is then only governed by the sampled histograms. Using the fact that $\sigma^2[H_i^{(n+1)}] \sim H_i^{(n+1)}$ [76], and further $\sigma^2[\ln H_i^{(n+1)}] \sim \sigma^2[H_i^{(n+1)}]/(H_i^{(n+1)})^2 \sim 1/H_i^{(n+1)}$, Eq. (2.11) can then be written as

$$\sigma^2[\beta_{i,0}^{(n+1)}] = \frac{c}{H_{i+1}^{(n+1)}} + \frac{c}{H_i^{(n+1)}} \tag{2.12}$$

with c an unknown constant. The statistical weight of the contribution of $\beta_{i,0}^{(n+1)}$ to the final estimator $\beta_i^{(n+1)}$ is inversely proportional to the variance. By choosing a convenient factor of proportionality, $c = 1$, we obtain the statistical weight

$$g_i^{(n+1)} = \frac{H_{i+1}^{(n+1)} H_i^{(n+1)}}{H_{i+1}^{(n+1)} + H_i^{(n+1)}} \, . \tag{2.13}$$

Now, the final estimator for the $(n+1)$th step can be obtained as a weighted average, which is based on $\beta_{i,0}^{(n+1)}$ together with the accumulated weights that were involved in calculating $\beta_i^{(n)}$ in the previous n iterations,

$$\beta_i^{(n+1)} = (1 - \hat{g}_i^{(n+1)})\beta_i^{(n)} + \hat{g}_i^{(n+1)} \beta_{i,0}^{(n+1)} \tag{2.14}$$

with $\hat{g}_i^{(n+1)}$ the accumulated variance weight

$$\hat{g}_i^{(n+1)} = \frac{g_i^{(n+1)}}{\sum_{k=1}^{n+1} g_i^{(k)}} \, . \tag{2.15}$$

Retaining the constant c in Eq. (2.13) would yield the same $\hat{g}_i^{(n+1)}$. Combining Eqs. (2.10) and (2.14) together, we are able to write the final estimator as a function of the weighted histograms,

$$\beta_i^{(n+1)} = \beta_i^{(n)} + \hat{g}_i^{(n+1)} \frac{\ln H_{i+1}^{(n+1)} - \ln H_i^{(n+1)}}{\Delta} \tag{2.16}$$

The major advantage of the improved parameter determination in the above equation, compared to Eq. (2.10), is that it leads to more stable simulating results.

Once $\beta_i^{(n+1)}$ is determined, we will be able to obtain the parameter α_i in Eq. (2.8) for the $(n+1)$th step. Considering two neighboring bins $i+1$ and i, the entropy at their shared boundary energy E_i should be continuous at each step, i.e.,

$S_{i+1}^{(n+1)}(E_i) = S_i^{(n+1)}(E_i)$ for step $n+1$. It can be further expressed in the parameterized form as $\beta_{i+1}^{(n+1)}E_i - \alpha_{i+1}^{(n+1)} = \beta_i^{(n+1)}E_i - \alpha_i^{(n+1)}$ based on Eq. (2.8). Hence, α_i can be obtained iteratively

$$\alpha_i^{(n+1)} = \alpha_{i+1}^{(n+1)} + [\beta_i^{(n+1)} - \beta_{i+1}^{(n+1)}]E_i \tag{2.17}$$

with the definition of $\alpha_L = 0$ in every step.

We have used the improved expression of β_i, Eq. (2.16), together with Eq. (2.17), for $i = 0, 1, ..., L-1$ to determine the multicanonical weight in this work. For $i \geq L$, we always let $\beta_i^{(n+1)} = \beta^{(0)}$ and $\alpha_i^{(n+1)} = 0$. Initially, $\beta_i^{(0)} = \beta^{(0)}$ and $\alpha_i^{(0)} = 0$ for all i. The updating procedure proceeds from the high energy bins towards the low energy bins, and if either $H^{(n+1)}(E_{i+1}) = 0$ or $H^{(n+1)}(E_i) = 0$, $g_i^{(n+1)} = 0$ from Eq. (2.13) and we set $\beta_i^{(n+1)} = \beta_i^{(n)}$. The weight for the $(n+1)$th iteration is then calculated according to

$$w_{\mathrm{mu}}^{(n+1)}(E) = e^{-\beta_i^{(n+1)}E + \alpha_i^{(n+1)}} \tag{2.18}$$

for E belonging to the ith bin.

2.3 Multicanonical Basin Hopping Method

MUCA is a very promising method that has particularly facilitated the exploration of the low energy landscape of various complex systems in the calculation of thermodynamic properties at low temperatures [14, 15, 17, 19, 20, 21, 23]. Because MUCA contains an effective temperature $T_i = (k_B\beta_i)^{-1}$ which is *not* identically 0 for the lowest energy bin, thermal fluctuations deter the system from descending into the global energy minimum as many low energy configurations still contribute prominently. To this extent, it suffers from the same problem as the simulated annealing method, and can only be used to give an estimate, but not the exact value, of the ground state energy. Considering the BH's merit in rapidly locating a minimum precisely and the MUCA's ability to surmount high energy barriers, these two methods are complementary to each other.

The main idea of MUBH is to handle the hopping between the plateaus in the reduced energy landscape with a probability function determined by the energy

spectrum that contains all reduced-energy plateaus. The targeted multicanonical weight, $w_{\rm mu}(E)$, in the original MUCA is replaced by $w_{\rm mu}(\tilde{E}) = 1/\rho(\tilde{E})$ with $\rho(\tilde{E})$ the density of minimal states, where $\tilde{E}$ is defined by Eq. (2.2).

MUBH contains the following steps. The first iteration is identical to the original BH procedure by carrying out a limited run that contains M canonical Monte Carlo steps for the transformed energy landscape, as described in Sec. 2.1. To do so, an initial temperature $T^{(0)}$ needs to be selected, and the sensitivity of this selection on the efficiency of the algorithm will be addressed below. Upon finishing the first iteration, we collect the histogram of the reduced energy defined in Eq. (2.2), and start to consider a multicanonical Monte Carlo procedure for the acceptance of a new hopping. Most equations in Sec. 2.2 remain applicable, provided that the energies E in these equations are replaced by the reduced counterparts $\tilde{E}$.

For a given system, we first select a lower bound E_0, chosen close to the best estimate of the lowest energy, and establish an upper bound E_L by identifying it with a value close to which the histogram attains its maximum in a limited MC run with the initial temperature $T^{(0)}$. The range $[E_0,\ E_L]$ is then divided into L equal segments with an increment Δ. Because we are only concerned about the energy plateaus in Fig. 2.1, in which the energy spectrum is represented schematically on the left hand side, the statistics collected during the simulation represents the frequency of visited plateaus. The statistical weight is characterized by the values of α_i and β_i obtained for the ith bin. For $\tilde{E} > E_L$, the Boltzmann weight $w_B(\tilde{E}) = \exp(-\tilde{E}/k_B T^{(0)})$ is used. In this weighting strategy, the final histogram distribution is expected to be smooth, if not completely flat, within $[E_0,\ E_L]$. The program flow chart for the MUBH method is shown in Fig. 2.2.

2.4 Application to Lennard-Jones Clusters

The Lennard-Jones cluster consisting of N particles (LJ_N) forms a crystal structure corresponding to an energy minimum. The LJ_N cluster was originally selected to demonstrate the effectiveness of the BH method. Further, the global energy minimum for each given N has been relatively well determined [2, 52, 53]. Even

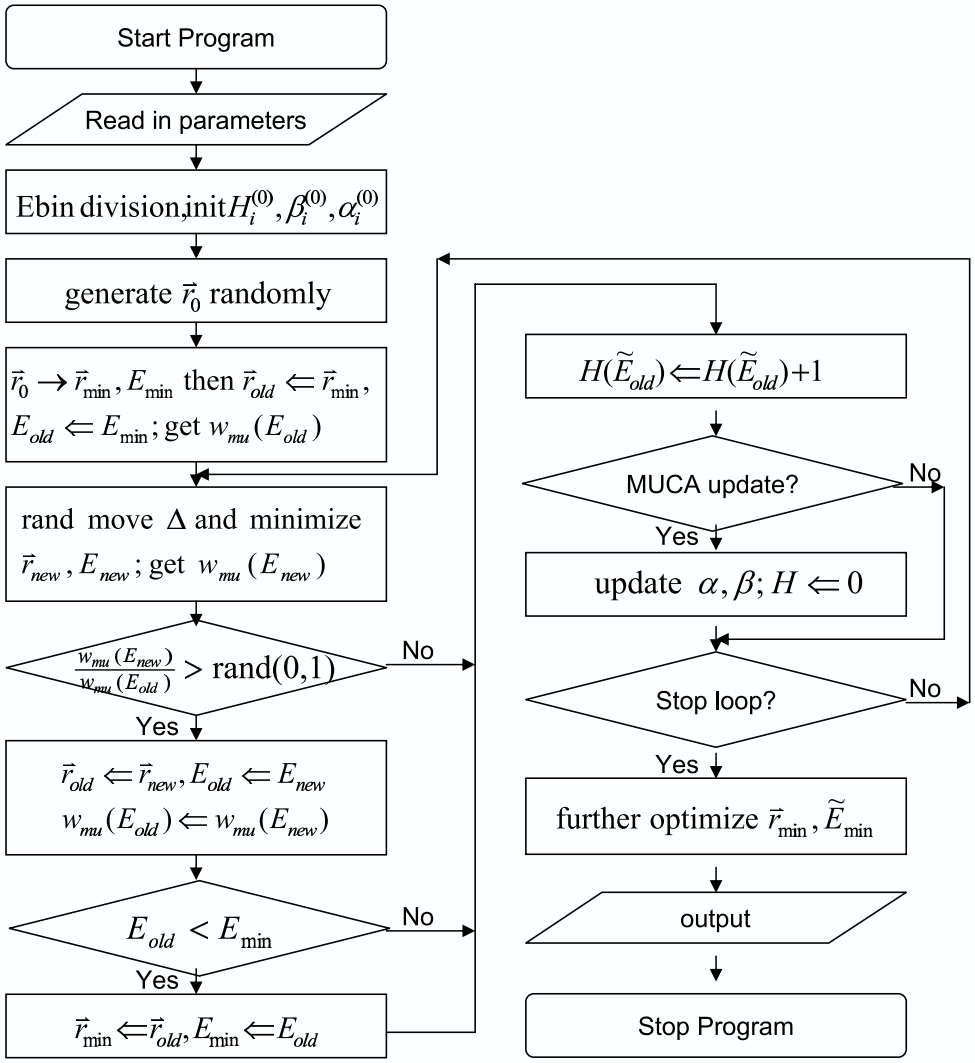


Figure 2.2: The flow chart for the multicanonical basin hopping method.

for a system with cluster size as small as $N = 98$ [25], there are of the order of 10^{40} minima, so that Lennard-Jones clusters are sophisticated enough for testing algorithms. Actually, the Lennard-Jones system has recently become a benchmark for checking the efficiency and accuracy of global optimization methods.

The energy of the LJ_N system is given by

$$E = 4\epsilon \sum_{i>j}^{N} [(\frac{\sigma}{r_{ij}})^{12} - (\frac{\sigma}{r_{ij}})^{6}] \tag{2.19}$$

where r_{ij} is the distance between particles i and j, ϵ and $2^{1/6}\sigma$ are the equilibrium well depth and separation, respectively, for a pair of particles. Hereafter, we employ reduced units such that $\epsilon = \sigma = k_{\mathrm{B}} = 1$. We performed both BH and MUBH simulations on various LJ_N systems for comparison.

For the BH method, all of the calculations were performed with the temperature $T = 0.8$. The MC displacement step size was initially set to 0.4, and was adjusted automatically during the simulations to maintain an acceptance ratio of 50%, averaged over every 100 MC steps. As for the MUBH calculations, two groups of runs based on different initial temperatures were performed. For $T^{(0)} = 2.0$, the multicanonical update was performed after every $M = 2\,000$ basin-hopping MC steps; while for $T^{(0)} = 5.0$, the update was after every $M = 1\,000$ basin-hopping MC steps. In a typical MUBH run, the entire energy range was divided into 12 bins — 10 bins between E_0 and E_L, together with two additional bins $[-\infty,\ E_0]$ and $[E_L,\ +\infty]$. E_0 was chosen to be close to the global energies provided in [53]. For the lower initial temperature runs, $T^{(0)} = 2.0$, the value of E_L will be smaller than that of the higher initial temperature runs, $T^{(0)} = 5.0$. Runs with a lower initial temperature had a narrower MUBH energy range.

Table 2.1 shows the results of BH and MUBH simulations on LJ_N. In the table, the number of runs failed ($\mathcal{F}$) to find the global minimum after a given number of steps, and the total number of runs ($\mathcal{T}$) are both shown. For any given system size N, $L_{\max}$ represents the maximal MC steps permitted in the simulation in case of any failed runs, or the actual maximal MC steps used to reach the global minimum if all runs are successful. For all the successful runs, the global energy minima obtained in our BH and MUBH calculations are identical to those reported

	BH($T^{(0)} = 0.8$)				MUBH($T^{(0)} = 2.0$)				MUBH($T^{(0)} = 5.0$)			
N	Ave	S.D.	$L_{\max}$	$\mathcal{F}/\mathcal{T}$	Ave	S.D.	$L_{\max}$	$\mathcal{F}/\mathcal{T}$	Ave	S.D.	$L_{\max}$	$\mathcal{F}/\mathcal{T}$
150	20,114	4,522	100,000	1/15	13,967	4,394	69,166	0/15	9,862	1,573	22,641	0/15
155	62,674	17,964	500,000	1/16	35,027	8,004	500,000	1/15	22,799	4,789	73,832	0/20
160	157,940	80,003	1,500,000	5/16	58,327	15,871	253,582	0/16	39,294	8,149	200,000	1/16
165	635,236	124,619	2,000,000	4/16	92,362	18,704	300,000	1/16	141,610	25,654	500,000	1/18
170	333,800	85,854	1,400,000	1/16	119,980	32,125	1,300,000	2/16	51,200	8,180	200,000	2/20
175	290,970	106,290	1,268,539	0/16	81,277	29,282	500,000	1/16	52,306	15,685	1,400,000	1/14
180	234,310	77,099	2,000,000	1/15	127,800	27,984	400,000	1/15	47,353	9,478	200,000	1/15
181	259,620	54,799	816,121	0/16	132,740	27,636	2,000,000	1/16	54,762	6,342	92,671	0/14
182	265,280	59,801	1,500,000	2/16	137,510	28,219	2,000,000	4/16	64,606	15,709	600,000	1/18
183	1,000,972	206,894	2,603,728	0/16	217,200	54,835	1,000,000	1/16	207,690	49,353	650,000	2/18
185	1,194,140	170,688	4,000,000	1/16	357,920	92,595	1,594,614	0/18	419,380	91,730	2,000,000	2/19

Table 2.1: The average number of MC steps to reach the global minimum for each N (Ave) and their standard deviation (S.D.) in both BH and MUBH methods. $\mathcal{F}$ gives the number of runs failed to reach the global minimum after $L_{\max}$ steps and $\mathcal{T}$ is the total number of runs conducted. If $\mathcal{F} = 0$, $L_{\max}$ represents the maximal number of steps to reach the global minimum among the $\mathcal{T}$ runs.

in Ref. [53]. The average MC steps to reach the global minimum listed in the table are plotted in Fig. 2.3 as a function of N. Compared to BH, MUBH requires less MC steps by factors of 2 to 5 in the global minimum exploring process. The figure also demonstrates that as N increases the reduction in computational time for finding the global minimum becomes more significant. The exact improvement, however, depends on the physical systems and hence cannot be quantified easily. The improvement of MUBH over BH for systems of size $N < 150$ is not noteworthy, as for smaller systems, the energy landscape is relatively simple. In some cases, MUBH can even give worse results than BH, because BH is already a very efficient method especially for low barrier cases, and it takes only several thousand MC steps to find the global minima for these systems, long before MUBH can reach a stable multicanonical weight. For example, we have tested both methods for systems of size $N = 100$ to 150, and could not find any significant saving in computational time in MUBH in comparison with BH.

The intrinsic capability of overcoming energy barriers in MUBH is the reason for the improvement in computational time. Figures 2.4 (a) and (b) show the time trajectories of $\tilde{E}$ for both BH and MUBH, respectively. It is visible from Fig. 2.4 (a) that the BH run can be trapped in a local minimum for a long computational time. In comparison, Fig. 2.4 (b) shows that the computation in MUBH proceeds very differently. The trajectory covers a much wider energy range and the MUBH run rarely gets trapped in a local energy minimum. Hence, in comparison with BH, MUBH encourages better navigation in the reduced energy landscape in search of the global energy minimum.

One important parameter that influences the MUBH search efficiency is the initial temperature $T^{(0)}$. This is already apparent in Fig. 2.3 for the two choices of $T^{(0)}$. Using $N = 150$ and $N = 170$ as examples, we have further performed independent MUBH runs for a number of different choices of $T^{(0)}$. The average number of MC steps for finding the global minimum as a function of $T^{(0)}$ are listed in Table 2.2 and plotted in Fig. 2.5. Also shown in Table 2.2 are the corresponding values of E_L. From both the table and the figures, it is clear that when the initial temperature is low, it requires more MC loops on the average for systems to reach their global minima. When $T^{(0)}$ is high, less MC loops will be needed. A low

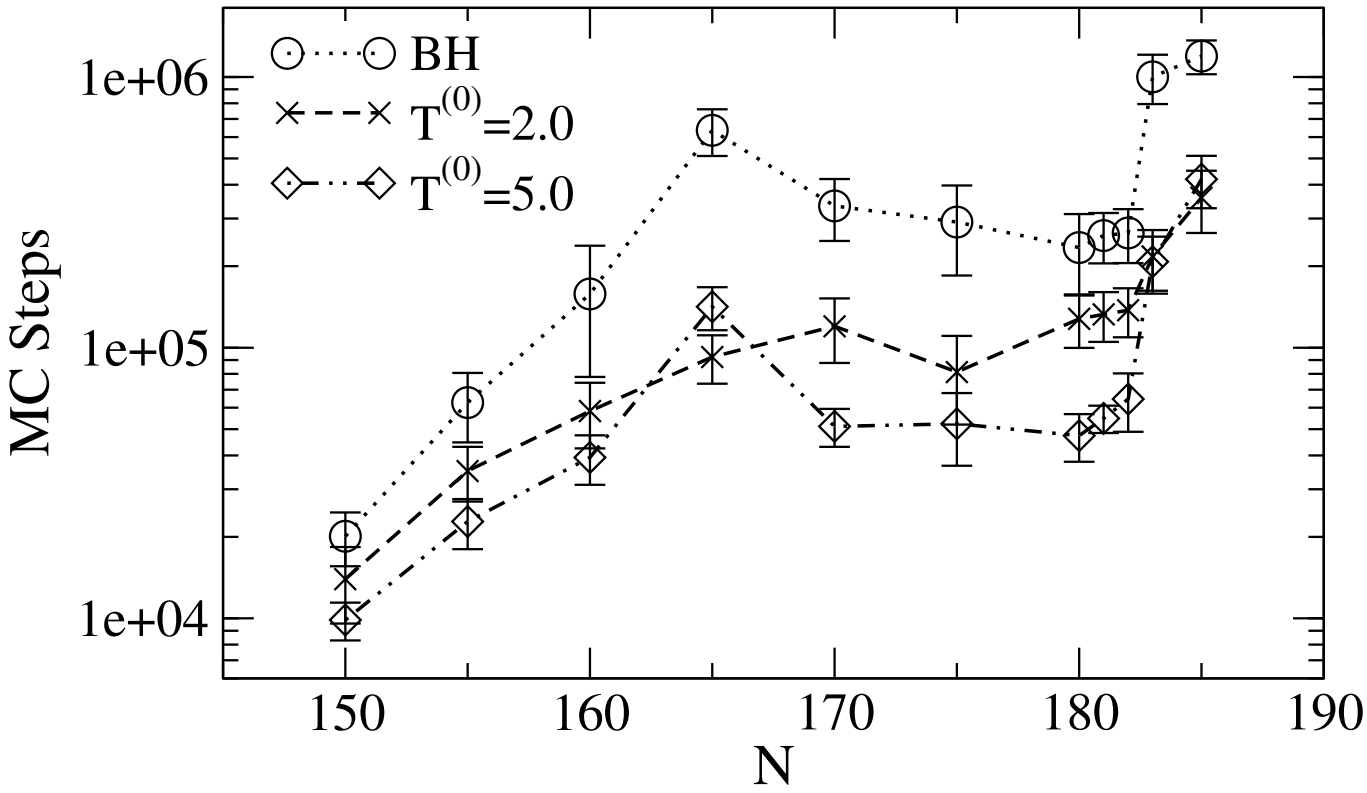


Figure 2.3: The average Monte Carlo steps of the BH and MUBH methods in finding global minima. The circles represent BH results, while the crosses and diamonds are MUBH results for $T^{(0)} = 2.0$ and 5.0, respectively. The error bar represents the standard deviation of the $\mathcal{T}$ runs for each N.

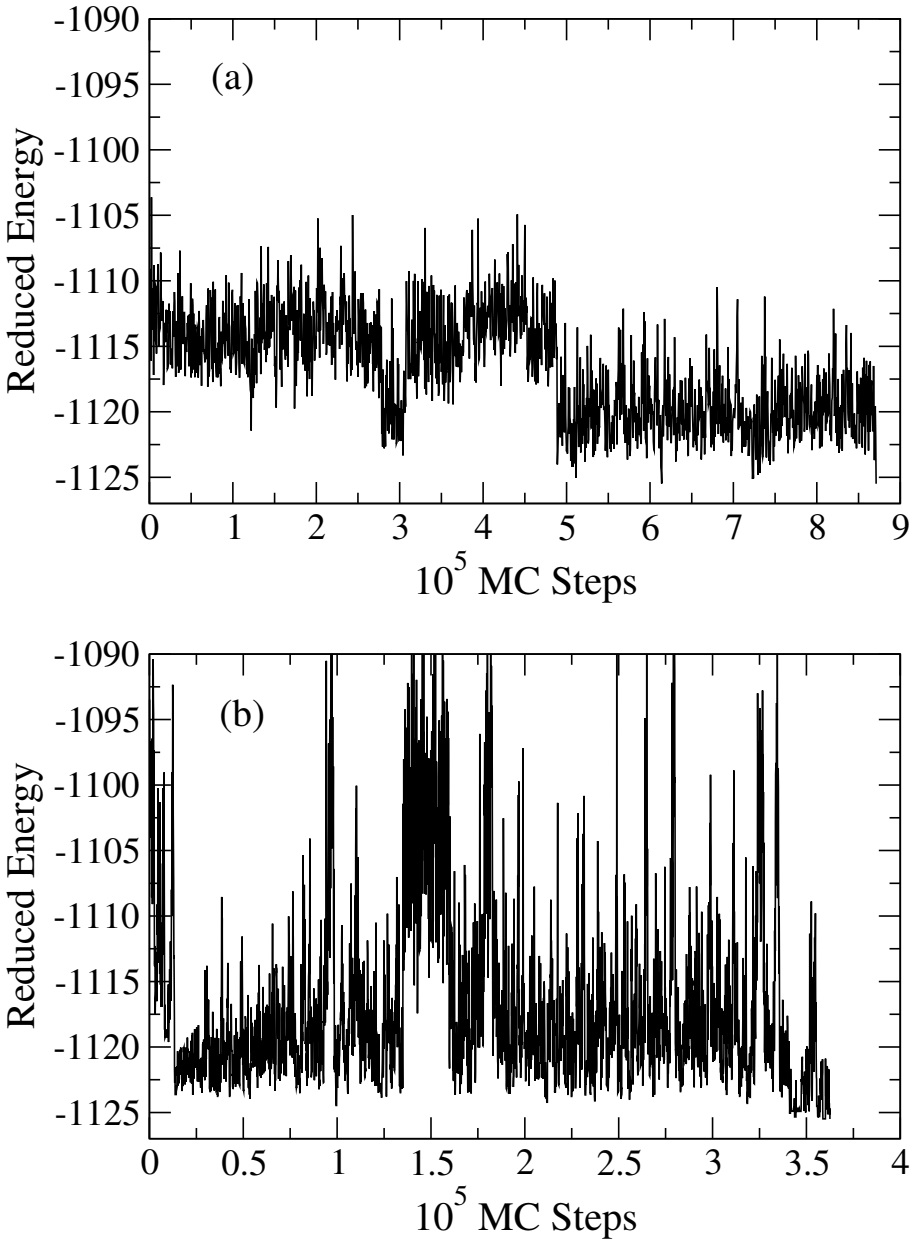


Figure 2.4: Typical searching trajectories of (a) BH with $T = 0.8$ and (b) MUBH with $T^{(0)} = 5.0$ for $N = 185$.

$T^{(0)}$	$N = 150$					$N = 170$				
	E_L	Ave	S.D.	$L_{\max}$	$\mathcal{F}/\mathcal{T}$	E_L	Ave	S.D.	$L_{\max}$	$\mathcal{F}/\mathcal{T}$
1.0	-870.4	13,765	4,053	64,361	0/15	-1,006.8	216,214	96,763	1,300,000	2/16
1.5	-870.4	13,880	3,544	53,988	0/15	-1,004.8	89,855	17,056	300,000	2/16
2.0	-870.4	13,967	4,394	69,166	0/15	-1,002.8	119,980	32,125	1,300,000	2/16
2.5	-868.4	11,714	2,158	29,893	0/15	-1,001.8	92,336	18,558	500,000	1/16
3.0	-868.4	13,881	3,022	41,495	0/15	-1,000.8	84,459	22,656	380,530	0/16
3.5	-865.4	10,410	1,459	26,616	0/15	-999.8	66,016	9,763	140,733	0/16
4.0	-865.4	10,324	1,258	21,957	0/15	-998.8	62,312	12,321	202,025	0/16
4.5	-864.4	12,529	1,950	27,077	0/15	-997.8	80,160	24,820	416,724	0/16
5.0	-863.4	9,862	1,573	22,641	0/15	-995.8	51,200	8,180	200,000	2/20
0.8(BH)	——	20,114	4,522	100,000	1/15	——	333,800	85,854	1,400,000	1/16

Table 2.2: The average number of MC steps correspond to different initial temperatures for the LJ clusters with $N = 150$ and $N = 170$. Ave, S.D., $L_{\max}$ and $\mathcal{F}/\mathcal{T}$ are the same as those of Table 2.1. E_L stands for the multicanonical upper bound energy with the unit of ϵ.

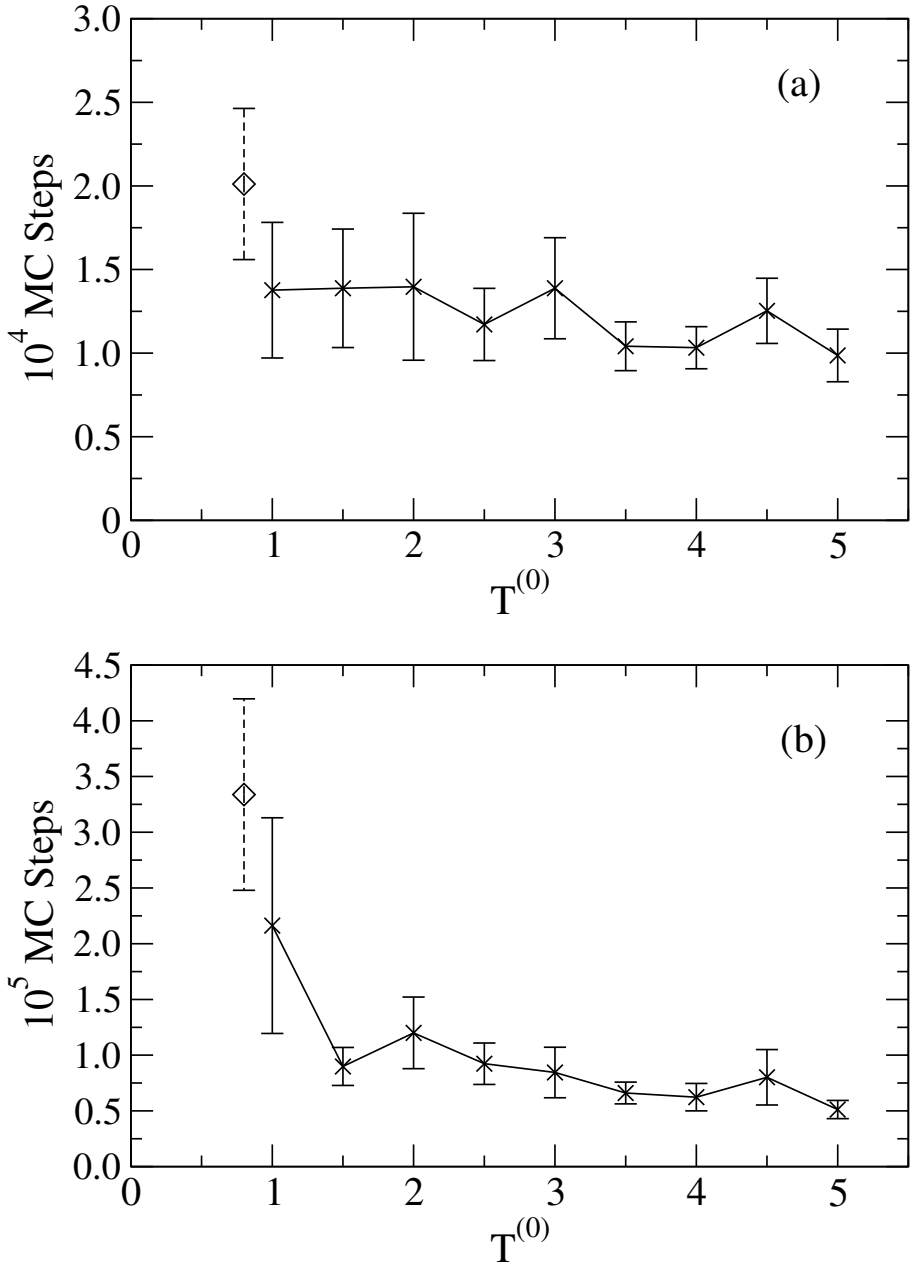


Figure 2.5: Initial temperature dependence of the average number of MC steps to locate the global minimum using MUBH for (a) $N = 150$ and (b) $N = 170$. The points represented by diamonds at $T^{(0)} = 0.8$ are the BH results. The error bar corresponds to the standard deviation for each run.

$T^{(0)}$ produces a low E_L which discourages the anticipated navigation to the high energy region. This potential pitfall becomes less serious as we use a higher $T^{(0)}$, resulting in a higher E_L. However, computationally, it is not always true that a higher $T^{(0)}$ necessarily leads to better efficiency. Indeed, the energy range is widened for the search with a high initial $T^{(0)}$; consequently it also requires a larger MC displacement step. We find that such increase would demand a longer computational time for the convergence of the local minimization procedure, in every MC step in MUBH. The ideal case would be to select a $T^{(0)}$ that corresponds to an E_L just above the energy barriers in the system — a largely unknown factor for any given complex system. Empirically, we find that $T^{(0)}$ in the range of 3.0 to 4.0 is probably optimal for conducting MUBH simulations for the $N = 150$ and 170 LJ_N systems, as can be seen from Fig. 2.5. We have also developed the following strategy in selecting $T^{(0)}$. Before performing a production MUBH run, we try several short BH runs of one or two thousand MC steps for each of the several values of $T^{(0)}$. If all these runs for a $T^{(0)}$ give similar histogram distributions, this $T^{(0)}$ become one of the candidates. After we obtain several candidates of $T^{(0)}$, we choose the one with lowest value.

Yet another important parameter that influences the MUBH efficiency is the total MC steps, M, between each multicanonical update for α and β. A reduction of M would effectively reduce multiple steps of the local minimization implementation in our algorithm. On the other hand, enough statistics for the energy bins should be accumulated before each update. For our study of the LJ_N systems, 1 000–2 000 MC steps between multicanonical update samplings on the 10+2 bins gave satisfactory results. In general, the number of bins and the number of MC sweeps in each MUCA iteration should be carefully chosen.

2.5 Summary

A new Monte Carlo method, the multicanonical basin-hopping (MUBH) method, is developed as a practical global optimization approach. This method is a combination of the multicanonical Monte Carlo method and the basin hopping method in order to make use of the advantages of both of them. To ascertain its efficiency, we

have implemented it on benchmark systems of Lennard-Jones clusters. For small systems, $N < 150$, the MUBH method gives no obvious improvement over the BH method because the reduced potential energy surface is relatively simple, so that BH could locate the global energy minimum before MUBH becomes effective. When the system size is increased to $N > 150$, the improvement of MUBH over BH is dramatic. These observations suggest that MUBH is suitable for large systems. The efficiency of MUBH comes from the fact that not only can it "hop" between the local minima directly, an advantage from the basin-hopping method, but it can also easily overcome the energy barriers in the transformed energy landscape using the non-Boltzmann scheme of the multicanonical method. It thus solves the problem of energy barriers in the reduced energy landscape of basin-hopping, and the insufficient sampling of the low energy landscape of the multicanonical method. The simulation results also show that the initial temperature setting is very important for the method. A suitable initial temperature will result in much better performance.

Chapter 3

Asynchronous Multicanonical Basin Hopping

Computers are becoming more and more importance in scientific research recently. Numerical simulation, in which global optimization is included, is one of the fields that relies dramatically on the development of computing abilities. Time spent in simulation is always the biggest problem that scientists and engineers have to face in their research. Much effort has been spent in saving computational time since the appearance of the first computer. The efforts can mainly be classified into several categories. One direct way is to improve the computational power of the CPU, which approximately obeys Moore's law. In the past years, we have been fortunate enough to witness the progress of information technology, which includes the development of CPUs. However, progress far less meets the increasing computational power required for computer simulations. Further, it is confined by the present manufacture science and technology and out of control of the most majority of scientists.

Another way is to simplify the description of the target system, which includes many techniques, such as getting rid of all the non-essential interactions between system components, using a simplified functional expression to approximate the original complicated interactions, and combining system components together into units while ignoring the internal interactions in each unit. Such approximation

techniques are widely used in nearly all the simulations and some well-known approximations are even commonly accepted as standards in solving related problems. By assuming that the electronic motion and the nuclear motion in molecules can be separated, and hence that the electron needs no reaction time to follow the movement of the nuclei, based on the fact that the nuclear mass is far greater than the electronic mass, the Born-Oppenheimer approximation becomes nearly the only feasible way in simulating quantum molecular interactions. The Lennard-Jones potential is a simple functional expression developed originally for approximating the van der Waals interactions between inert gas atoms. Even the hard sphere approximation for atoms can generate satisfactory results in some situations. There are some other empirical potentials, such as the Morse potential and the Gupta potential, that are used for different systems to study their physical properties. The results are often in good agreement with experiment. An example given in this chapter is the simulation results of cobalt clusters obtained using the Gupta potential. In the protein folding study, the empirical force fields adopted, which include the Amber [77, 78, 79, 80, 81, 82], CHARMM [83, 84], ECEPP [85, 68, 69, 70], OPLS [86, 87, 88] force fields etc., are all approximations to the atomic interactions of the protein or peptide system based on the experimental observations and the *ab initio* calculations of small systems, which in turn is based on the Born-Oppenheimer approximation. All these force fields have proven to be good descriptions at different levels and in different situations.Researchers are studying system dynamical and/or stochastic properties by adopting molecular dynamics or Monte Carlo simulations based on them. An even much simpler approximation, the Gō model [89], is often used to study the dynamical properties of protein systems.

Once the system to be studied, which includes the approximate or exact description of the system, and the computational resources available are determined, an inevitable step is to select or develop the right computational algorithm for the study. Different algorithms may result in dramatic differences in performance and accuracy. One of the best algorithms ever proposed is the fast fourier transform (FFT), which improves the computational efficiency from order $\mathcal{O}(N^3)$ for the discrete fourier transform (DFT) to order $\mathcal{O}(N \log N)$. When only Monte Carlo simulations are considered, we have also seen some milestones in algorithm development.

The Metropolis algorithm [12] proposed in 1953 introduced importance sampling into computer simulation. Simulated annealing (SA), in 1983 by Kirkpatrick *et al.* [3], started the wide application of Monte Carlo methods in solving global optimization problems. The umbrella sampling method proposed in 1977 by Torrie and Valleau [13] introduced a non-Boltzmann weight and the reweighting technique to MC. Further improvements of the histogram technique, the reweighting technique and simulation efficiency were realized with the proposing of the histogram reweight method in 1988 by Ferrenberg and Swendsen [90], the multicanonical Monte Carlo method in 1991 by Berg and Neuhaus [14, 15] and the Entropic sampling method in 1993 by Lee [22]. The recently appeared Wang-Landau algorithm, in 2001 [91, 92], was claimed to be more efficient than the multicanonical Monte Carlo method, and is becoming more and more extensively applied in many fields.

The work we did in the previous chapter also represents algorithm development. The MUBH method we proposed shows obvious efficiency improvements in applications to the global minimization of large systems. However, when the system size becomes much larger, MUBH requires very long computational times to perform the simulation as well. Because no algorithm with much greater efficiency improvement has appeared recently and the computational power of single CPU is limited, we had to seek a different solution, namely parallel computing, which organizes the CPUs in a computer network to solve the same problem simultaneously by distributing parts of the work to different CPUs for computing and re-organizing them when finished. This is a CPU for time strategy. When all other efforts are not applicable, parallel computing will be left as the only feasible way to save computational time.

After a brief review of parallel techniques for Monte Carlo methods in the next section, we shall discuss in Sec. 3.2 the parallelization of the MUBH method introduced in the previous chapter to AMUBH, which stands for *asynchronous multicanonical basin hopping*. AMUBH, together with BH and MUBH, will then be applied to Co nanoclusters to obtain their global minimum structures in Sec. 3.3, and a fast global minimum locating approach, the *structure mapping* method, will be proposed in Sec. 3.4 when analyzing the the structures of the Co nanoclusters.

3.1 Parallel Computing and Monte Carlo Method

For a typical Monte Carlo procedure, each new step is generated by making a small trial move from the present state. The acceptance of the current move is determined by the relative weight of the new state in comparison with that of the present one, according to a selection rule, such as the Metropolis criterion. This procedure is performed iteratively, thereby generating a typical Markov-chain. According to how many Markov-chains are generated in a simulation, there are two main directions in which parallelization of the Monte Carlo method can be implemented: single Markov-chain parallelization or multiple Markov-chain parallelization.

Intrinsically, a Markov-chain procedure is a serial task that is mostly suitable for linear processing. However, depending on the physical systems studied, the calculation could still be parallelized for single Markov-chain computing. For systems with very short-range potentials, e.g. the hard sphere model as a limiting case, or with nearest neighbor interactions only, e.g. the Ising model, it is possible to parallelize the simulation procedure by dividing the system into subdomains. Thus, one can simulate each subdomain on separate CPUs and then collect the calculated results together to determine the configuration weight of this step. Due to the system interacting characteristic, there are few interactions between the divided subdomains, and hence communications between the CPUs will not be significant: this is important for improving the efficiency of CPU usage. Simulations performed on lattice particles ("spins") by Pawley *et al.* [93] was one of the first successful applications of domain decomposition to parallelize the Monte Carlo method. Once long-range interactions exist between system particles, the domain decomposition method will no longer be applicable.

When it is possible to divide the system energy, including the long-range interactions, into different parts, parallel energy calculation seems to be the only method that intrinsically speeds up every Monte Carlo move [94]. In this case, the system energy can be divided into different parts, and each part can assigned to a CPU. The total energy is obtained by collecting and summing up all the partial energies from the individual CPUs. The system weight will then be determined by the total

energy. In Ref. [95], Jones and Goodfellow discussed the energy parallel approach and proposed an improved scheme for a better arrangement of communications in order to minimize CPU idle time. Using the PVM (Parallel Virtual Machine) package, Hayryan *et al.* [94] parallelized the ECEPP [85, 68, 69, 70] force fields implemented in the program package SMMP [96], and performed MUCA simulations to small peptides.

Since the working objective of the parallelization approaches just discussed is the simulated system itself, rather than the Monte Carlo algorithm, parallelization using either the domain decomposition method or the parallel energy calculation scheme depends only on the system studied. Once the system is decomposable, no matter by domain or by energy, most algorithms, MC included, can be adopted to perform the distributed computing. While this kind of MC parallelization is not easy to implement, it follows nearly the same trajectory as its corresponding sequential code. The efficiency speedup is also strongly system dependent.

There is still another Monte Carlo method, the hybrid Monte Carlo (HMC) method [97], that uses the parallel techniques developed for MD simulations. Each MC iteration in HMC contains four steps: (1) random velocity selection for each particle according to Gaussian distribution; (2) a short MD simulation; (3) calculation of the Boltzmann factor for the new configuration; and (4) acceptance criterion checking for the MD move just performed. All the parallel algorithms developed for MD simulation can be applied straightforwardly in step (2).

Perhaps the most effective way of parallelizing the MC method is by adopting the multiple Markov-chain scheme. In fact, multiple Markov-chain implementation of the MC method was quite active over the past few years and several algorithms based on this technique have been proposed. By setting the initial conditions the same in every aspect except for seeds of the random number generator, such as the implementation in Ref. [31], ensemble data from the parallel MC simulation can be averaged over all processors, which yields better statistics than a single simulation of the same duration. This is the most obvious way to parallelize the MC methods. However, it can simply be replaced by running several sequential simulations and accumulating all the data obtained.

The replica exchange method (REM) [98] (also referred to as parallel temper-

ing [34], replica Monte Carlo method [99], or multiple Markov-chain method [100]), simulates several, say M, *noninteracting* copies (or replicas, in the language of REM) of a system at M different temperatures. Each copy corresponds to one temperature and *vice versa*. Simulations are performed in the canonical ensemble which means that the Boltzmann weight is applied directly and no weight determination step as in the generalized ensemble is required. After every pre-set number of MC steps, pairs of replicas (or equivalently pairs of simulation temperatures) are exchanged with a specified transition probability. The exchanged process undergoes a random walk in the temperature domain, which in turn induces a random walk in the energy domain. Hence a wide configuration space can be sampled during the simulation. By assigning each replica to a separate CPU of a computer cluster and switching the replicas between CPUs when required, REM can easily be parallelized with no additional effort. A sample implementation can be found in Ref. [101], which maintained 20 copies on 20 nodes in simulating the protein folding problem. REM has been extended to the replica exchange multicanonical (REMUCA) method [102], the replica exchange simulated tempering (REST) method [103], and the multicanonical replica exchange method (MUCAREM) [102].

In the parallelization of the MUBH method, we shall use the multiple Markov-chain technique to minimize the CPU idle time, which is inevitable for parallel computing. Moreover, it is easier to implement than other techniques.

3.2 Asynchronous Multicanonical Basin Hopping

The multicanonical basin hopping (MUBH) method contains two key points, as illustrated in detail in Section 2.3: the local minimization for each trial move, and the new state selection criterion based on the multicanonical weight, processes inherited from the basin hopping method and the multicanonical Monte Carlo method, respectively.

In each MC step, a small trial move $\Delta\mathbf{r}$ is performed on the original configuration $\mathbf{r}_{\min}$, which is a local minimum obtained from $\mathbf{r}$ in the previous step, and the

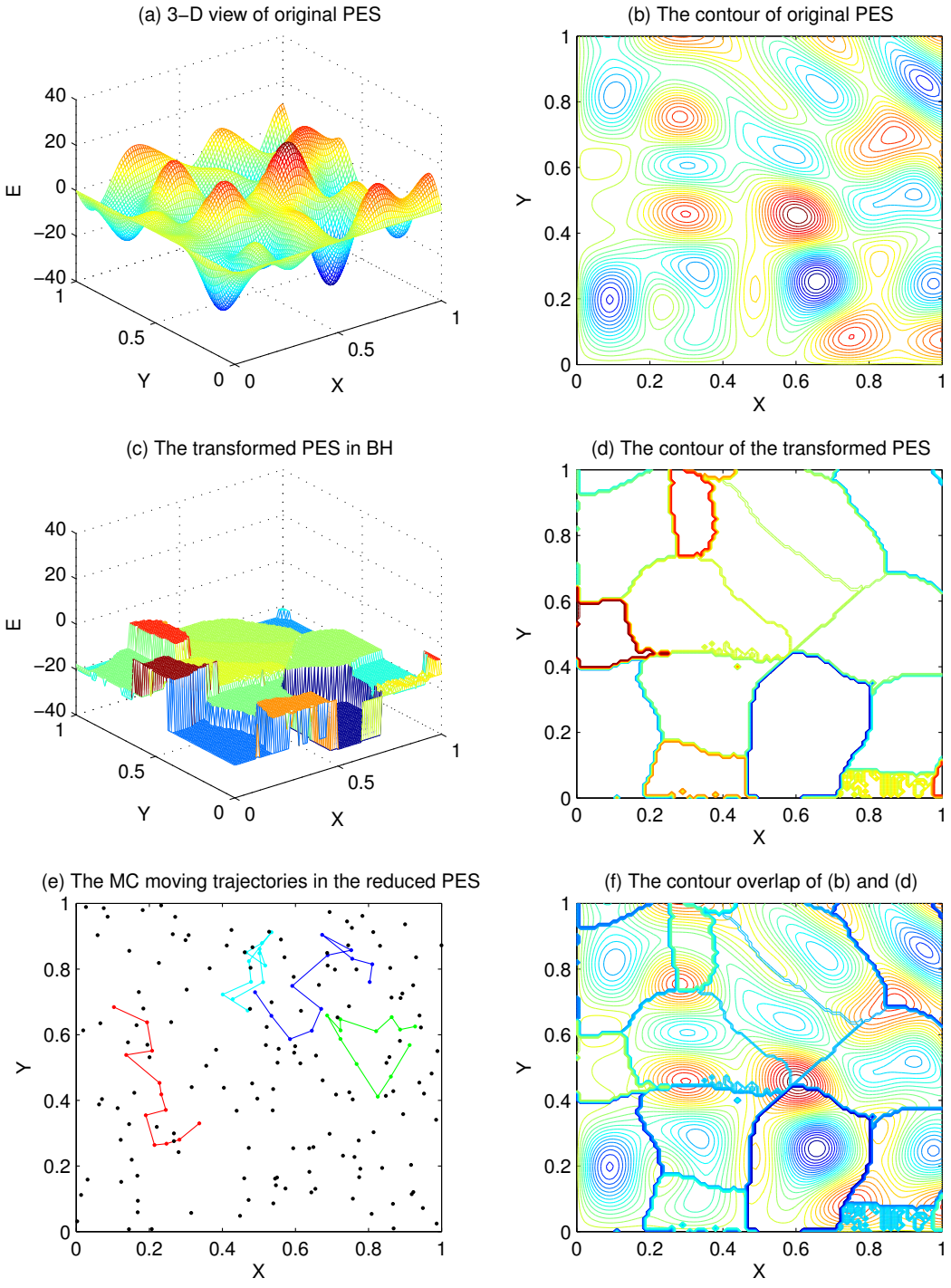


Figure 3.1: A schematic illustration of energy transformation in two dimensions for a basin-hopping-related method.

system reaches a new configuration $\mathbf{r}' = \mathbf{r}_{\min} + \Delta\mathbf{r}$. A local minimization procedure started from $\mathbf{r}'$ is then performed to determine the new local minimum, $\mathbf{r}'_{\min}$. The acceptance of $\mathbf{r}'_{\min}$ is determined by the multicanonical weights of state $\mathbf{r}'_{\min}$ and the previous state $\mathbf{r}_{\min}$. This iteration procedure only ends when the pre-set terminal condition is satisfied. The minimization procedure is equivalent to transforming the original energy landscape, denoted by $E(\mathbf{r})$, to a reduced one, denoted by $\tilde{E}(\mathbf{r})$, which contains only the local minima of the original one,

$$\tilde{E}(\mathbf{r}) = \min\{E(\mathbf{r})\} = E(\mathbf{r}_{\min})\,. \tag{3.1}$$

Figure 3.1 shows a two dimensional (2D) illustration of the energy landscape transformation: panel (a) illustrates the original potential energy surface (PES), and panel (b) its contour map. After the energy transformation by determining all of the local minima of the system configurations, the original PES will be transformed into staircase-like plateaus as shown in panel (c), with its contour map shown in panel (d). Panel 3.1 (f) shows the regions of the contour map that were converted to different plateaus by overlapping of the contour maps shown in panels 3.1 (b) and (d). The parallelization can be performed in this step. However, as the parallel energy calculation method we mentioned in the last section is system dependent, its application to a new system will possibly require new coding. Further, it is hard to balance the data distribution for the best CPU usage in order to reduce the CPU idle time during message communication.

The Monte Carlo weight determination step in MUBH is the same as in the MUCA method, except that the energy $\tilde{E}(\mathbf{r})$ used here is the reduced energy instead of the original energy $E(\mathbf{r})$. To determine the multicanonical weight $w_{\mathrm{mu}}(\tilde{E})$, we adopted the recursive scheme of Berg and Neuhaus [14, 15] and Berg [17, 16, 18], which is a stable method for determining $w_{\mathrm{mu}}(\tilde{E})$ iteratively. When the system entropy $S(\tilde{E}) \equiv \ln \rho(\tilde{E}) = -\ln w_{\mathrm{mu}}(\tilde{E})$ is parameterized as

$$S_i(\tilde{E}) = \beta_i \tilde{E} - \alpha_i\ , \qquad \text{for} \quad E_{i-1} < \tilde{E} \le E_i\ , \tag{3.2}$$

then the main iteration equations that are used for determining the system entropy, and hence the multicanonical weight, are

$$\beta_i^{(n+1)} = \beta_i^{(n)} + \hat{g}_i^{(n+1)}\, \frac{\ln H_{i+1}^{(n+1)} - \ln H_i^{(n+1)}}{\Delta}\,, \tag{3.3}$$

and

$$\alpha_i^{(n+1)} = \alpha_{i+1}^{(n+1)} + [\beta_i^{(n+1)} - \beta_{i+1}^{(n+1)}]\, E_i\,. \tag{3.4}$$

Details of the definition of the parameters, the division of the energy bins and the derivation of the equations can be found in Secs. 2.2 and 2.3.

After each iteration update of α_i and β_i, the multicanonical weight within bin i is calculated from

$$w_{\mathrm{mu},i}(\tilde{E}) = e^{-S_i(\tilde{E})} = e^{-(\beta_i \tilde{E} - \alpha_i)}, \qquad \text{for} \quad E_{i-1} < \tilde{E} \leq E_i\,. \tag{3.5}$$

Once the multicanonical weight has been determined, the normal Monte Carlo iterations can then be performed using this weight. The final histogram distribution is expected to be flat in the energy range that uses the multicanonical weight.

To this end, we propose the asynchronous multicanonical basin hopping (AMUBH) method, which is a parallel implementation of the MUBH method utilizing the multiple Markov-chain technique. A single computation is carried out over multiple processors, with each processor carrying out one independent computation starting from a different initial condition, which is the initial configuration in our implementation. Panel (e) of Fig. 3.1 illustrates several Markov-chains in searching a 2D configuration space. Additional searching routes along different Markov-chains at the same time will allow more space to be visited during a limited simulation time. AMUBH combines the statistical histograms collected from all processors for occasional update of the multicanonical weight, which is then distributed to each processor for continuing calculations. Because running threads are not required to finish synchronously for the update, the CPU idle time is minimized.

Another crucial aspect of any algorithm is its portability to variable computational environments, in particular, to differently structured computer clusters. The present program has been implemented using the Message Passing Interface (MPI) with C bounding, since MPI is accepted as the future message passing standard [104] and is widely available. There is no need to modify the source parallel codes when porting from one platform to another, so long as it supports MPI. To ensure that the code can make full usage of various types of processors in a cluster, we have designed the code such that faster CPUs would not need to wait for slower CPUs to finish a certain segment of computation.

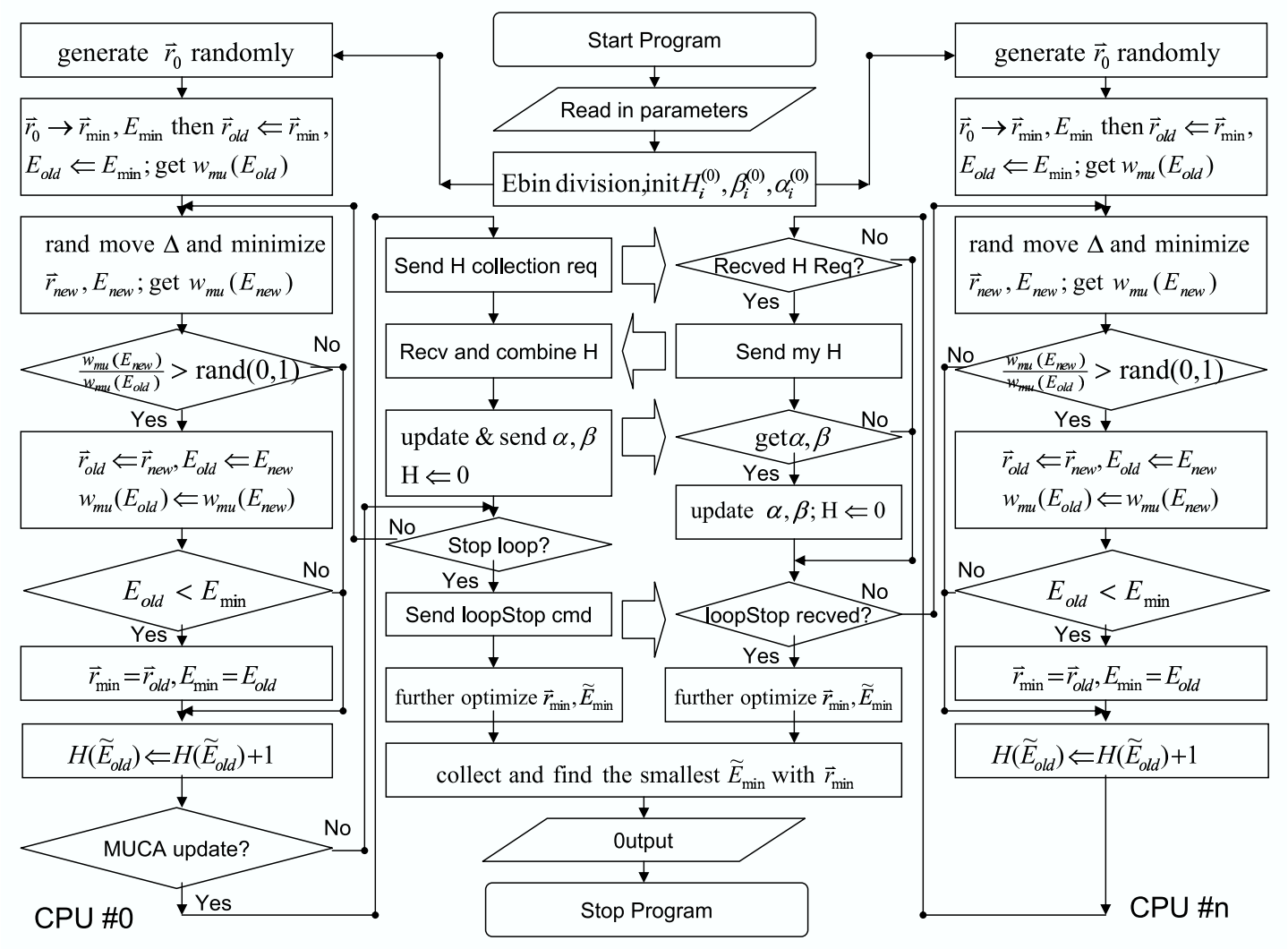


Figure 3.2: The flow chart for the asynchronous multicanonical basin hopping method.

Figure 3.2 shows the flow chart for the AMUBH algorithm. Suppose there are N_p processors. After the input of all the parameters for the system in the initialization step, jobs are distributed to each of the N_p processors. A simulation begins in a processor with a random configuration obtained by providing the random number generator a distinct seed, which results in uncorrelated random number sequences across the cluster. From these initial conditions, N_p different Markov-chains are generated simultaneously in the simulation. Except for the weight updating step, all other steps in each processor are carried out identically to those in a stand-alone sequential MUBH calculation. When the weight updating step (MUCA update) is reached, the main node, which we call CPU0, sends a request to all the other CPUs to ask for their collected histograms and waits for their answers. Upon receiving such a request, each CPU sends its histogram collection to CPU0 once the current Monte Carlo step has been completed — a minimal amount of delay for CPU0 because each Monte Carlo step takes only a short CPU time. After the arrival of all histograms, CPU0 combines them with its own and uses the resulting tally to calculate β_i and α_i according to Eqs. (3.3) and (3.4). Then the updated β_i and α_i are sent back to the other CPUs to continue on the next iteration using the multicanonical weight of Eq. (3.5) until another request for histogram update arrives. In implementing this scheme, the only idle time occurs at CPU0 during the period between sending out the MUBH update request and receiving the histograms from the last CPU. All other CPUs keep on running after sending out the histograms to CPU0, in case a new global minimum is found before the updated β_i and α_i from CPU0 arrive. Of course the histogram information collected during this "waiting period" by these CPUs cannot be used for the next update since the old weight is still being used. For a large system, the idle time of CPU0 is inconsequential compared to the total time required for the computation.

The main feature of AMUBH is that the convergence in estimating β and α can be obtained approximately N_p times faster than that in a sequential code. If we need M Monte Carlo steps in sequential MUCA to update the weight, we may need only $M_p = M/N_p$ Monte Carlo steps on average to obtain the new β_i and α_i. The sooner the multicanonical weight converges, the earlier AMUBH offers the probability for the system to visit the rare configurations, although the real time

N	MUBH	AMUBH	N_p	$\mathcal{F}/\mathcal{T}$	MUBH/AMUBH
165	92 362	43 667	4	2/8	2.12
170	119 980	30 047	6	1/8	3.99
185	357 920	106 416	8	2/8	3.36

Table 3.1: The speedup table for AMUBH compared with MUBH when applied to the Lennard-Jones clusters. Columns MUBH and AMUBH are the average MC steps for the simulations to reach the global minima. N_p is the number of CPUs used in each AMUBH simulation. $\mathcal{F}$ gives the number of runs that failed to reach the global minimum in the pre-defined number of steps for each system size for a total of $\mathcal{T}$ simulations performed.

required to locate the global minimum may not scale linearly with the number of processors.

To demonstrate the improvement, AMUBH was applied to finding the minimum energy configurations of Lennard-Jones particles of size $N = 165$, 170 and 180, with initial temperature $T^{(0)} = 2.0$. The results were compared with those obtained from sequential MUBH presented in Sec. 2.4, as shown in Table 3.1. For each AMUBH run, the job was stopped when a global minimum obtained in Sec. 2.4 was found. The tests were run on a homogeneous computer cluster, and we took the MC steps of the CPU that found the global minimum as the MC steps of the entire run. In Table 3.1, we have listed the average AMUBH steps needed to find the minimum in successful runs. To present the statistics more faithfully, we have also listed $\mathcal{F}$, the number of runs in which the job met a pre-set upper limit of MC steps without finding the global minimum, and $\mathcal{T}$, the total number of runs conducted. The upper limit of total MC steps was set at 8×10^4, 1.5×10^5, and 2×10^5 for $N = 165, 170$ and 180, respectively. We can see from the table that there is substantial improvement of the performance of AMUBH over MUBH.

3.3 Crystal structure of the Cobalt Nanoclusters

Recently, the size distribution of monodispersed Co nanoclusters on a single crystal Si_3N_4 film at room temperature has been experimentally determined [43]. Thus it is of interest to determine theoretically the energy and structure of the global minima of Co clusters and compare them with experimental results.

Empirical many-body potentials have played an important role in computer simulations of the thermodynamic and structural properties of physical clusters. The Gupta potential [57] has been successfully applied to metal clusters [26, 48, 27, 45], even though it was originally proposed for studying lattice relaxation at a metal surface. The N-body Gupta potential energy is given by

$$E = \sum_{i}^{N} [V_r(i) + V_d(i)] \; . \tag{3.6}$$

where, $V_d(i)$ is a many-body potential for particle i based on the tight-binding model, which has the form

$$V_d(i) = -\zeta \left\{ \sum_{j \neq i} \exp[-2q(r_{ij}/r_0 - 1)] \right\}^{1/2} \; , \tag{3.7}$$

with r_0 the equilibrium nearest-neighbor inter-atomic distance in the bulk. The excluded-volume nature of the cluster is represented by a short range repulsive potential $V_r(i)$,

$$V_r(i) = \xi \sum_{j \neq i} \exp[-p(r_{ij}/r_0 - 1)] \; , \tag{3.8}$$

where ζ, q in Eq. (3.7) and ξ, p in Eq. (3.8) are parameters for different metallic clusters. For Co clusters, $\zeta = 1.4880$ eV, $q = 2.286$, $\xi = 0.0950$ eV, and $p = 11.604$ [58]. Using data from Ref. [57], r_0 is determined to be 2.497 Å even though this parameter is not needed in our simulation, as interatomic distances can be measured in units of r_0, which is what we used in our calculation.

Using the BH, MUBH and AMUBH methods, we are able to locate the global minima of Co clusters with system size N up to 200. As discussed in Chapt. 2,

BH is sufficiently efficient for small clusters, so that for cluster size $N \leq 150$, we simply used BH to calculate the global minima. For $150 < N \leq 180$, MUBH is utilized to improve the sampling efficiency. When the system becomes even larger, $180 < N \leq 200$, MUBH will take too long to locate the global minimum in a single simulation, and only AMUBH, the asynchronous parallel version of MUBH, is capable of locating the global minima within a reasonable computational time by utilizing several processors. In most of our runs, we used 8 processors.

We summarize the results of the global minima of Co clusters with N up to 200 in Table 3.2, obtained from the BH, MUBH, or AMUBH methods as discussed above. It is possible that there may still exist global minima which we are unable to locate. Even for a "small" Lennard-Jones system of $N = 98$, it has been estimated that there are of the order of 10^{40} local energy minimum states [25]. It is only recently that a new configuration, Leary's tetrahedron structure [25], was discovered with lower energy minimum than previously found. Technically, it is impossible for Monte Carlo methods, no matter how efficient they are, to browse over all the system configurations in a reasonable time, especially for large systems. Nonetheless, we believe that the great majority of the energy minima listed in Table 3.2 are true global minima.

3.4 Analysis and Discussion

3.4.1 Most Stable Structures

The global minimum energies by themselves do not provide too much information about the structural changes in the system. To observe how particularly stable clusters stand out from the average trend, we first fitted the energies in Table 3.2 to a smooth background

$$E_{fit}(N) = aN + bN^{2/3} + cN^{1/3} + d. \tag{3.9}$$

The following parameters have been numerically obtained: $a = -4.439$ eV, $b = 2.966$ eV, $c = -0.314$ eV, and $d = 1.069$ eV. Then, the differences between the

N	Sym*	E(eV)	N	Sym*	E(eV)	N	Sym*	E(eV)	N	Sym*	E(eV)	N	Sym*	E(eV)
2	$D_{\infty h}$	-3.15065179	42	C_s	-150.3537320	82	C_1	-307.8073682	122	C_1	-468.7835333	162	C_s	-631.4892354
3	D_{3h}	-6.13875890	43	C_s	-154.3956285	83	C_{2v}	-311.8078868	123	C_s	-472.9077116	163	C_s	-635.6974681
4	T_d	-9.53815918	44	C_1	-158.0644979	84	C_1	-315.5917003	124	C_s	-477.1200398	164	C_1	-639.6651744
5	D_{3h}	-12.80265501	45	C_s	-162.0863246	85	C_1	-319.6053737	125	C_s	-481.1192842	165	C_s	-643.6312979
6	O_h	-16.34438847	46	C_{2v}	-166.2913580	86	C_3	-323.7154810	126	C_s	-484.8936404	166	C_s	-647.4269926
7	D_{5h}	-19.72589196	47	C_1	-169.9535267	87	C_s	-327.6080333	127	C_{2v}	-489.1059691	167	C_s	-651.4852810
8	D_{2d}	-23.02759620	48	C_s	-173.9503669	88	C_s	-331.8011735	128	C_s	-493.2284031	168	C_{3v}	-655.6716875
9	C_{2v}	-26.51110143	49	C_{3v}	-178.1544651	89	C_{3v}	-335.9802365	129	C_s	-497.4405058	169	C_2	-659.7610012
10	C_{3v}	-30.07287786	50	C_s	-181.8220828	90	C_s	-339.9817382	130	C_s	-501.5615395	170	C_{2v}	-663.9368866
11	C_{2v}	-33.61576383	51	C_{2v}	-185.9350749	91	C_s	-343.9809747	131	C_{2v}	-505.7734388	171	C_s	-668.0163456
12	C_{5v}	-37.49841062	52	C_{3v}	-190.1403866	92	T_d	-347.9778980	132	C_1	-509.4321859	172	C_{5v}	-672.2252676
13	I_h	-41.81920702	53	C_{2v}	-194.3327986	93	C_1	-351.7684938	133	C_s	-513.5564800	173	C_{5v}	-676.4334607
14	C_{3v}	-44.93586980	54	C_{5v}	-198.5240012	94	C_1	-355.7718467	134	C_{3v}	-517.7680506	174	C_s	-680.4455188
15	C_{2v}	-48.52567444	55	I_h	-202.7136225	95	C_1	-359.7710265	135	C_s	-521.9794548	175	C_s	-684.4569924
16	C_s	-52.06297312	56	C_{3v}	-206.0150172	96	C_1	-363.7165452	136	C_s	-526.1908129	176	C_s	-688.4678220
17	C_2	-55.60429192	57	C_s	-209.5442434	97	C_{2v}	-367.7878938	137	C_{3v}	-530.4021224	177	C_{5v}	-692.4785721
18	C_s	-59.31543520	58	C_{3v}	-213.5703311	98	C_s	-371.9770018	138	C_{3v}	-534.5374680	178	C_{5v}	-696.4876286
19	D_{5h}	-63.55030713	59	C_1	-217.2824524	99	C_{2v}	-376.1518497	139	C_{2v}	-538.7473477	179	C_s	-700.5051534
20	C_{2v}	-67.02968998	60	C_s	-221.2802030	100	C_s	-380.1500888	140	C_1	-542.9571615	180	C_s	-704.7117734
21	C_1	-70.52209797	61	C_{2v}	-225.2668703	101	C_{2v}	-384.1472457	141	C_{5v}	-547.1669337	181	C_1	-708.8456888
22	C_s	-74.18441246	62	C_s	-228.9027576	102	C_{2v}	-388.0998555	142	C_1	-551.3764942	182	C_s	-713.0250346
23	D_{3h}	-78.29682896	63	C_1	-232.8947610	103	C_s	-392.1510857†	143	C_{2v}	-555.5860117	183	C_{2v}	-717.2330494
24	C_{2v}	-81.76435353	64	C_s	-236.8781886	104	C_{2v}	-396.1596140†	144	C_{3v}	-559.7954535	184	C_s	-721.0009837
25	C_3	-85.47990015	65	C_2	-240.5201760	105	C_2	-399.9681259	145	C_{2v}	-564.0048189	185	C_1	-725.0514853
26	T_d	-89.36830977	66	C_1	-244.5103568	106	C_1	-403.9771201	146	C_{5v}	-568.2141412	186	C_1	-729.1129869
27	C_{2v}	-92.98960595	67	C_2	-248.4964196	107	C_s	-408.1348622	147	I_h	-572.4234175	187	C_s	-733.3760918
28	T	-96.92113165	68	C_1	-252.2877445	108	C_s	-412.3059832	148	C_1	-575.7201251	188	C_s	-737.5849134
29	C_3	-100.4998608	69	C_1	-256.3507509	109	C_1	-416.3057218	149	C_s	-579.4663803	189	C_s	-741.7933500
30	C_s	-104.1809983	70	C_{5v}	-260.5151508	110	C_s	-420.3045440	150	C_{3v}	-583.5955478	190	D_{5h}	-746.1251704
31	C_3	-107.9263613	71	C_5	-264.6524350	111	C_s	-424.2282150	151	C_1	-587.4533875	191	C_{5v}	-750.3333012
32	D_3	-111.9588267	72	C_s	-268.2437968	112	C_s	-428.3068888	152	C_s	-591.4342513	192	D_{5h}	-754.5413203
33	C_1	-115.7226628	73	C_s	-272.2334494	113	C_s	-432.4790340	153	C_{3v}	-595.3992438	193	C_{2v}	-758.3111302
34	C_s	-119.4265968	74	C_{5v}	-276.3986522	114	C_s	-436.6473857	154	C_{2v}	-599.2114317	194	C_s	-762.3625498
35	C_2	-123.2227072	75	D_{5h}	-280.6061534	115	C_{5v}	-440.8130358	155	C_s	-603.2419487	195	C_{2v}	-766.3703440
36	C_s	-127.2020270	76	C_s	-284.2813802	116	C_{5v}	-444.9823350	156	C_s	-607.4238848	196	C_{2v}	-770.3240955
37	C_1	-131.0021454	77	C_{2v}	-288.2933742	117	C_1	-448.6415848	157	C_{2v}	-611.6331535	197	C_s	-774.3435029
38	O_h	-135.2399864	78	C_1	-291.9663639	118	C_s	-452.6837657	158	C_s	-615.6036022	198	C_2	-778.3467808
39	C_{5v}	-139.1262062	79	C_{2v}	-295.9790500	119	C_s	-456.7668376	159	C_{2v}	-619.5716762	199	C_1	-782.4498541
40	C_s	-142.8153124	80	C_1	-299.7861918	120	C_1	-460.4312295	160	C_1	-623.3105715	200	C_1	-786.4906393
41	C_s	-146.4701197	81	C_2	-303.8053980	121	C_1	-464.5711039	161	C_1	-627.3149921			

Table 3.2: The lowest minimum energies found for Co clusters. * Symmetry class of the clusters determined from Ref. [105]. † Results from the structure mapping of the corresponding Lennard-Jones configurations; BH gives -391.9370759 eV for $N = 103$ and -395.9359667 eV for $N = 104$.

energy minima and the empirical fitting as a function of system size N are shown in Fig. 3.3 (dots connected by a black solid line). It is clear from the figure that the clusters with $N = 13$, $N = 55$ and $N = 147$ have the lowest relative energies compared to their neighbors, which is a signature that they have the most stable crystal structures. This corresponds to the fact that 13, 55 and 147 are the magic numbers for systems which have closed-shell icosahedral structures. In addition, there are other structures which are relatively more stable than their close neighbors. It is easy to see that $N = 19$, 75, 116 are amongst them.

3.4.2 Structure Mapping

For clusters with binary interaction between particles, the relative potential range and shape determine the most stable structure for a typical cluster size N. The interaction potential between any two atoms in a cluster is repulsive at short distance due to the strong coulomb repulsion when electron clouds of the atoms overlap, and is attractive at long distance because of induction and dispersion effects, so that there is a potential well with a minimum at the two-atom equilibrium separation. Model potentials such as the Lennard-Jones potential, the Morse potential and the Gupta potential all share these general features. The similarity of the potentials will give rise to similar physical properties, which may include the lowest energy structure. In Fig. 3.4, we compared the Lennard-Jones potential with the Gupta potential for two atoms, and we can see that they are very similar to one another. Taking data from the Cambridge Cluster Database [52] and from Ref. [53], we plot in Fig. 3.5 the relative energy $E(N) - E_{fit}(N)$ for Lennard-Jones clusters up to $N = 200$. Eq. (3.9) is taken again as the fitting function, but with a new set of parameters: $a = -8.595/\epsilon$, $b = 15.15/\epsilon$, $c = -4.681/\epsilon$ and $d = -2.935/\epsilon$ with ϵ the pair equilibrium well depth of the Lennard-Jones potential. From Fig. 3.3 and Fig. 3.5, we observe that the trend of the relative energy of the Co and the Lennard-Jones clusters are similar.

Based on the LJ configurations provided by the Cambridge Cluster Database [52] for size up to 150, we first scaled the interatomic distances in the clusters by the ratio of the equilibrium distance of the Co Gupta potential to the LJ equilibrium

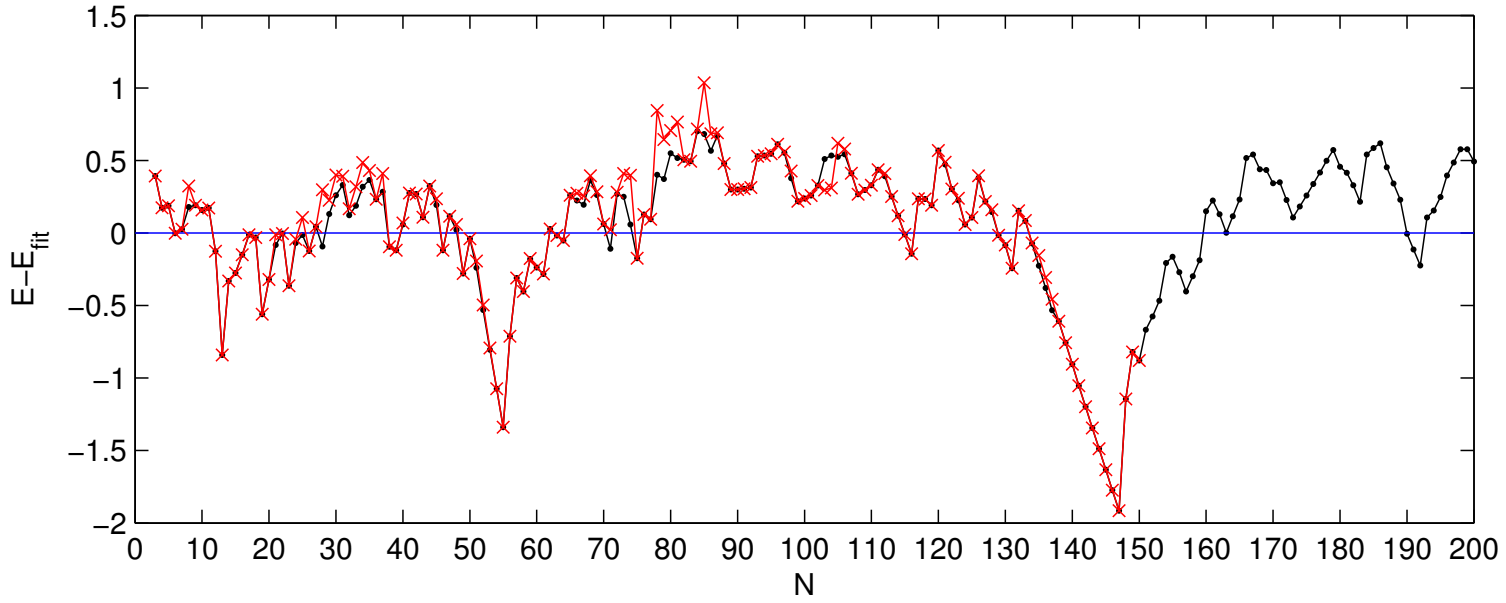


Figure 3.3: The Co cluster energy minima relative to the smooth background $E_{fit}(N)$ obtained from a four-parameter fit to the energy minima (dots connected by a black solid line). Clusters having more stable structures, i.e., lower energies, can be identified with the dips in the plot. The crosses connected by a red line denote the mapped energies from the Lennard-Jones clusters.

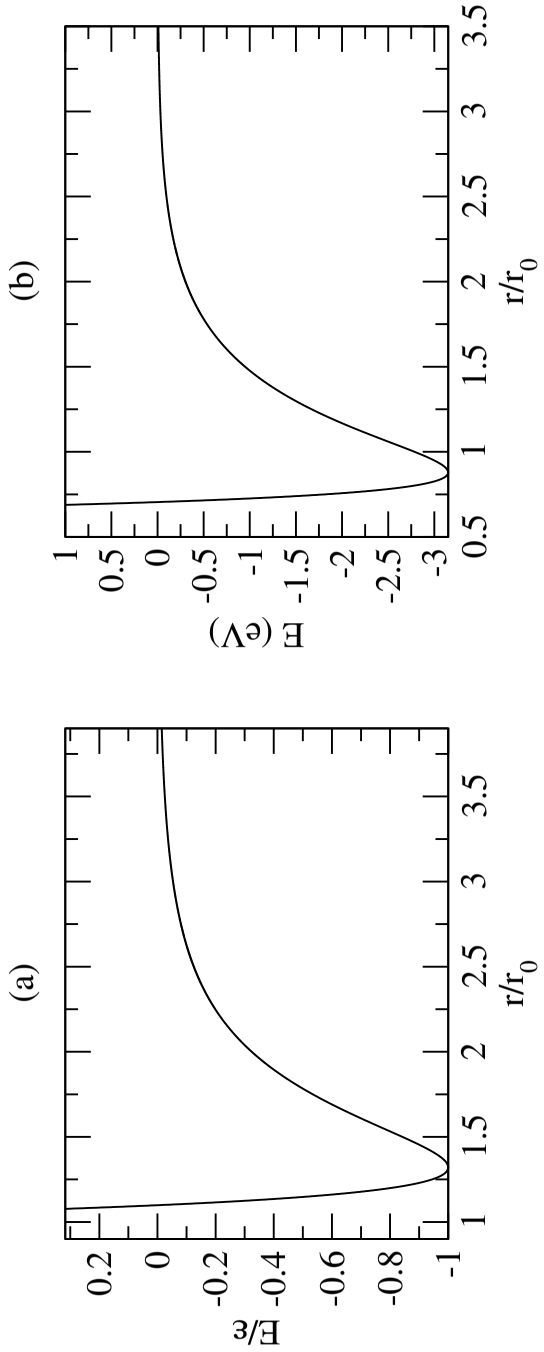


Figure 3.4: The Lennard-Jones potential (a) and the Gupta potential (b) between two atoms.

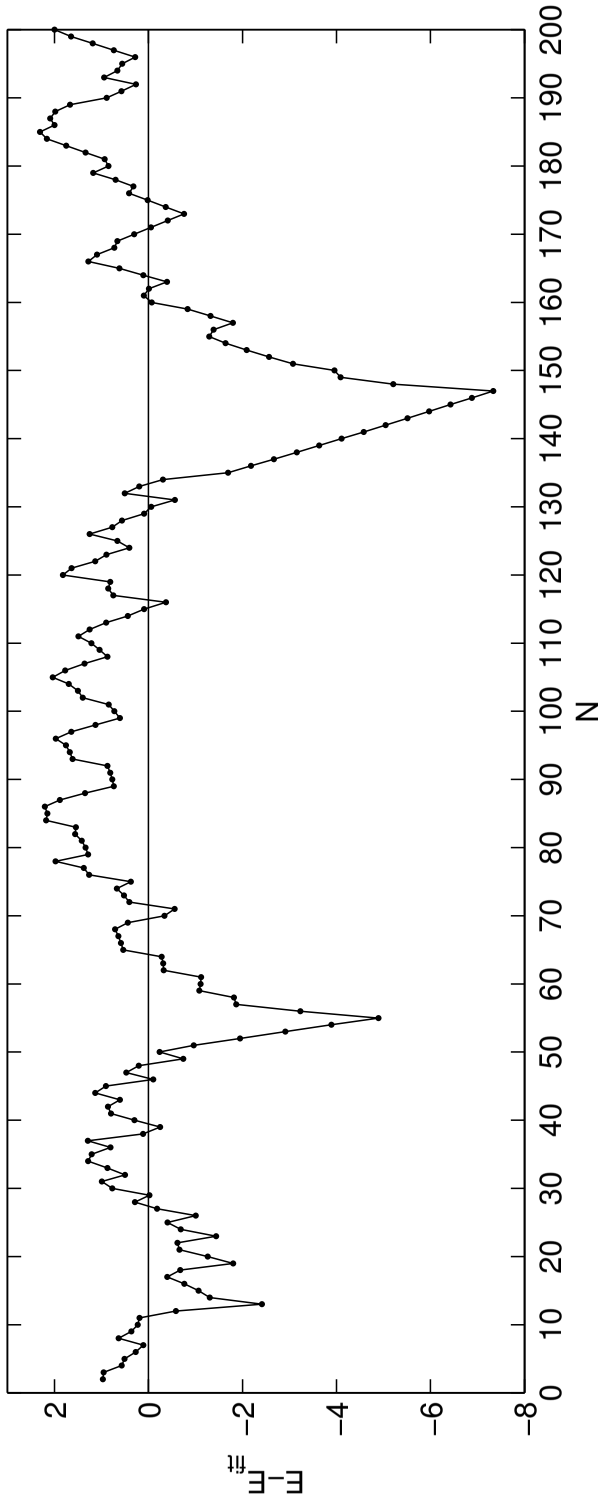
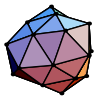
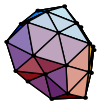


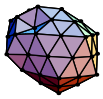
Figure 3.5: The differences of the global minima of the Lennard-Jones clusters with their corresponding fitted energies $E_{fit}^{(LJ)}(N)$.

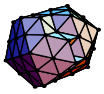
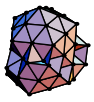
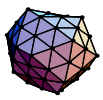
8 
18 
21 
24 
25 
28 

29 
30 
31 
32 
33 
34 

35 
37 
40 
45 
48 
51 

52 
53 
59 
66 
67 
68 

69 
71 
72 
73 
74 
78 

79 
80 
81 
84 
85 
86 

87 
97 
98 
105 
106
112

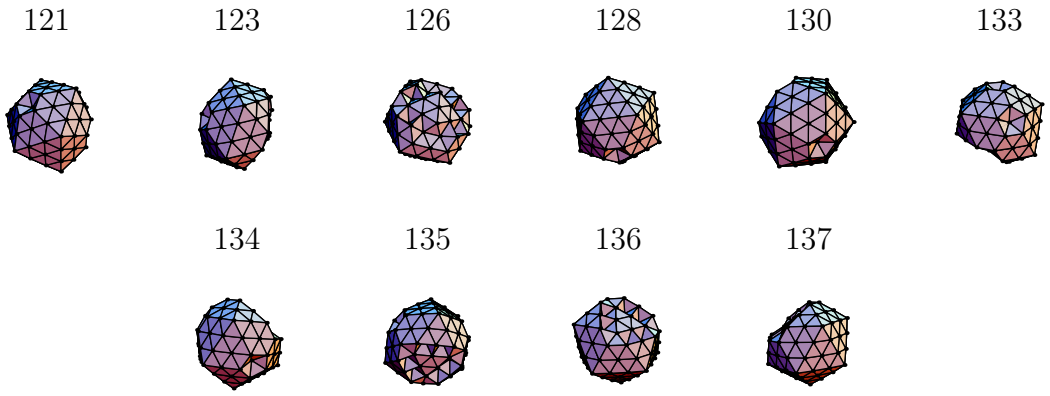


Figure 3.6: The global minimum structures which are different from their Lennard-Jones mapped siblings for $N \leq 150$.

distance and map all of the structures to the Co cluster. We then carried out a *single* energy minimization to obtain the stable structures for Co, which we called the *mapped structures*. For the mapped structures, we also plotted their energy differences with $E_{fit}(N)$ in Fig. 3.3 (crosses connected by a red solid line) for comparison with the minimum energies obtained from the BH method. Here we note that for $N = 103$ and 104, the unbiased BH method failed to locate the decahedral structure, which is the structure of the global minima for these Co clusters obtained from structure mapping. Note that the energies listed in Table 3.2 for $N = 103$ and 104 are obtained from structure mapping. For $N = 102$, we found the decahedral structure in our first BH try, but all subsequent searches (more than 10) failed. In addition to $N = 102$, 103 and 104, there are 52 clusters (for $N \leq 150$) which have minimal energy structures different from their Lennard-Jones mapped siblings. These structures are shown in Fig. 3.6. $N = 98$ is amongst them, which means that Leary's tetrahedron is no longer the global minimum for the 98-atom Co cluster. Nevertheless, a collection of all the structures obtained from simple model potentials may provide a shortcut in determining the most stable structure of any cluster system.

3.4.3 Comparison with Experiment

Recently, Gwo *et al.* reported the formation of monodispersed Co nanoclusters on a single-crystal Si_3N_4 dielectric film at room temperature [43]. Since the energy difference between the lowest energy state and the second lowest energy state (-392.151085 eV and -391.9370759 eV respectively, for $N = 103$) is far greater than $k_BT \approx 0.026$ eV at room temperature, we can compare our computed lowest energy structure with the room temperature experimental results. It is convenient to plot the discrete second derivative of $E(N)$ defined by

$$\Delta_2 E(N) = E(N+1) + E(N-1) - 2E(N) \tag{3.10}$$

as a function of cluster size N, as shown in Fig. 3.7. A large positive $\Delta_2 E(N)$ thus represents a stable cluster with size N. We also included the ranges of stable structures determined in the experiment [43] in the figure, and we found that peaks can be located within most of these ranges. There is only one exception, at $N = 55$ for which the closest experimental range is $51 - 54$. Since $N = 55$ is a magic number indicating a closed-shell structure, it should be more stable than its neighbors. The cluster size of the experimental result was obtained from the droplet volume, measured using scanning tunneling microscopy (STM), and the atom number density, which was estimated by assuming that all atoms are packed together following the hexagonal close packed (hcp) structure in the droplets [106]. Our calculation shows that most of the stable structures are icosahedral. While the density of the hcp and icosahedral structures are not the same, an icosahedron can be considered to consist of twenty tetrahedra packed around common vertex with minor distortion [107], so that the density difference between the icosahedral and the hcp structures is small. Thus comparison between the experimental and our calculated results remains valid. Our results and experiment agree very well for most of the stable structures, which in turn shows that the Gupta potential is reliable for modeling Co nanoclusters.

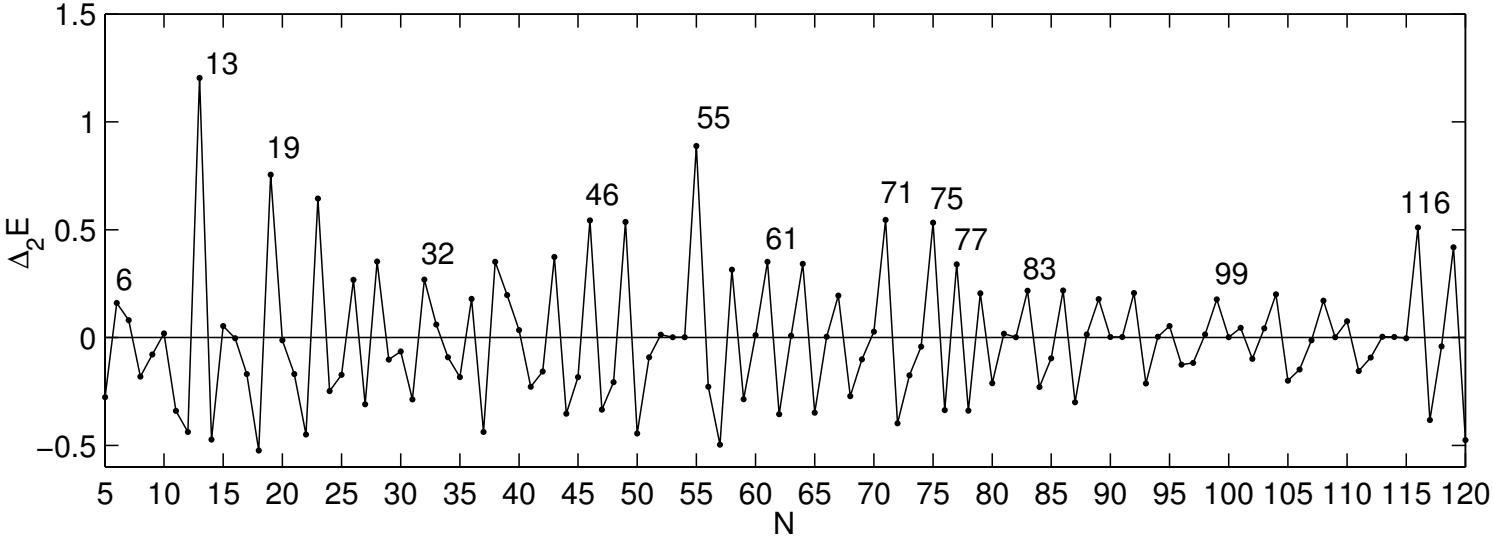


Figure 3.7: The discrete second derivative of $E(N)$ for Co nanoclusters in the range of 5–120. The numbers beside the high peaks stand for the stable structures corresponding to the experimental results of Ref. [43]. An exception here is $N = 55$.

3.5 Summary

In this chapter, we proposed the AMUBH method, an asynchronous parallelization of the multicanonical basin hopping method, by adopting the multiple Markov-chain technique. The method was found to be efficient when applied to large systems that the sequential MUBH method finds difficulty to process. The BH, MUBH and AMUBH methods were utilized to determine the structures of Co nanoclusters with system size N from 2 to 200. Most of the stable structures we found agree well with those determined experimentally. This agreement in turn illustrates that the Gupta potential we employed describes Co clusters well. Mapping the structures of the known Lennard-Jones systems to Co clusters helped us locate the real global minima for $N = 103$ and 104. Even though there remain differences in the mapped and actual lowest-energy structures, configuration mapping provides a useful method for fast global minimum determination.

Chapter 4

Basin Paving Method

Monte Carlo methods based on histogram accumulation are widely applied in computer simulations these days. By dividing the energy space, for example, into regions (bins), and accumulating the visiting frequency of each energy region, one will gain much direct knowledge of the simulation procedure. Adjustment can then be performed to guide the simulation in the desired direction. Hence, improvement of efficiency is expected from such approaches. In general, these methods often need to examine the histogram distribution in the range studied, which is closely related to the probability distribution. The desirable histograms are expected to cover a wide (energy) range and to have a relatively flat distribution. Consequently, the simulation process will be able to visit wider regions of configuration space and to overcome the (energy) barriers of a rugged potential energy surface. In practice, this is often realized by estimating the system density of states (DoS), which is unknown *a priori*. To determine DoS first, one needs to perform long pre-production MC runs, which can sometime take up to 40% of the total simulation time [20]. When global optimization is considered, the characteristic of the histogram methods may improve the chance of locating the global optimum.

Nowadays, many algorithms applying the histogram accumulation schedule have been proposed and they have proven to be powerful. The histogram method [90, 108] of Gerrenberg and Swendsen proposed in 1988 is an early introduction of the histogram schedule into the Monte Carlo method. A reweighting scheme has also

been used to extract thermodynamic information from a single MC simulation. The broad histogram method (BHM) [109, 110, 111, 112] first proposed by de Oliveira *et al.* in 1996 was believed by the authors to provide an exact determination of the density of states, even though there are systematic errors in their simulation results for a 32×32 lattice Ising model in Refs. [109] and [111]. Recently, they were able to reduce the error near T_c to a small value as shown in Ref. [113]. Questioning the correctness of the BHM random walk dynamics [114, 115], Wang and Lee [116] proposed a flat histogram method (FHM) in 2000, which is an improved histogram MC method based on the same equations and starting point as BHM. Their test on the 32×32 two dimensional Ising model showed the superiority of FHM to BHM. To our knowledge, there is no application of these methods to global optimization problems.

The multicanonical Monte Carlo (MUCA) method, proposed in 1991 by Berg and Neuhaus, is another histogram-based MC method. The detail of the method has been discussed in Chapter 2. MUCA has gained much popularity in recent years. For a MUCA simulation, the density of states is not required to be known accurately as long as the histogram distribution is relatively flat and the simulation covers a large energy region. Then one can overcome the energy barriers, and the subsequent reweighting step does not rely on the accuracy of the DoS [92]. The efficiency has been proven by its successful application to the study of first-order phase transitions [14, 15] and protein folding problems [20, 21, 117]. The entropic sampling (ES) method, proposed independently in 1993 by Lee [22], is basically equivalent to MUCA [118]. MUCA has been used in attempts to find the global energy minimum in protein folding by Hansmann and Okamoto [20, 21]. It has also been employed in the study of clusters by Bhattacharya and Sethna [119] even though their result is not favorable to MUCA. Further, it has been combined with the simulated annealing method [29], for example, for rapid location of the system global minimum.

Recently, Wang and Landau [91, 92, 120] proposed an efficient multiple-range random walk algorithm for accurate estimation of the DoS, generally known as the Wang-Landau algorithm (WLA). The WLA performs a random walk in energy space to obtain a very accurate estimate of the DoS iteratively. Starting from an

approximation of the DoS, the simulation procedure will modify the DoS at *each* step, according to a control parameter f, so that a relatively flat energy histogram distribution can be generated. By updating f according to a well chosen strategy, an improved update of the DoS will be generated when a new flat histogram distribution is obtained. The DoS will converge to the true value quickly, even for large systems, after several iterative modifications of f. Once the DoS is determined, one can estimate thermodynamic quantities at any temperature by taking canonical averages. A bonus of WLA is that one can directly estimate the Gibbs free energy and entropy through the accurately determined DoS. As the authors have claimed [91, 92, 120], "this algorithm is especially useful for complex systems with a rough landscape since all possible energy levels are visited in the same probability." What is really interesting is that, while the authors emphasized that WLA could offer substantial advantages over existing approaches, one of which is the flat histogram method, Wang and Swendsen performed a comparison between the algorithms on 2D Ising models [121], and found that more accurate results can be obtained using the FHM method with the same number of Monte Carlo steps (10^6 steps). However, with only this comparison, we cannot say that FHM is superior to WLA since the largest size of the Ising model that they employed is only 50×50, while the WLA was claimed to be efficient and accurate for large systems. Further, the control parameter f they used in their simulations may not be well chosen since f is quite flexible and the convergence speed of a simulation is sensitive to it. The WLA has been applied to optimization problems and found to be efficient [122].

The histogram approaches just mentioned above have all been developed with the initial intention of obtaining thermodynamic quantities by enhancing the visiting frequency of the extreme energies through the guidance of the collected histograms. This ability substantially enables the simulation procedure to visit lower energy regions more frequently and to overpass high energy barriers between local minima. These approaches are good candidates for global optimization problems. Due to thermal fluctuations as we mentioned in Chapter 2, however, they are not able to locate system minima precisely, which may result in failing to find the global minimum. The basin hopping method [2], on the other hand, can locate the local minima precisely with a deterministic procedure in each MC step. With transition

rates determined by the Boltzmann weight, it may still fail to overpass high reduced energy barriers (which is the *de facto* local minima of the original energy surface). Our most recently proposed multicanonical basin hopping (MUBH) method [6, 7], which has been described in detail in Chapter 2, solves this difficulty by introducing the multicanonical weight to the basin hopping method to control the acceptance of each step. Its application to Lennard-Jones clusters shows dramatic improvement over BH in obtaining the global energy minima for large systems.

Recently, Hansmann and Wille proposed a new algorithm based on histogram accumulation, which they called the energy landscape paving (ELP) method [40]. Unlike the other methods that we have mentioned, which were developed for the purpose of thermodynamic studies by approximating the DoS, ELP was designed solely for global optimization problems, which makes it substantially more efficient in such applications. Hsu *et al.* employed the energy landscape paving method and the simulated annealing (SA) method to determine the crystal structure of an organic compound from simulated X-ray diffraction data comprised of integrated intensities [41]. Their results indicated that ELP is more efficient than SA in finding the crystal structure by an order of magnitude [41]. Arkın and Çelik investigated the performance of ELP and MUCA in finding the lowest energy configurations of the heptapeptide deltorphin [42]. Their conclusion is that very long computational times are required for MUCA simulations because the probability weight factors are unknown *a priori* and have to be determined by iterations of trial simulations, as we have discussed. The ELP simulations, on the other hand, are more effective in sampling the lower energy region and studying the low energy structures.

The MUBH method simplifies the energy landscape and adopts the multicanonical weight to realize a relatively flat sampling in the reduced energy space, which also enables it to overcome the energy barriers in the transformed space. The transformation of the energy landscape sacrifices one of MUCA's attractions in obtaining thermodynamic quantities by the reweighting scheme through a single simulation. Further, the procedure for determining the density of (reduced) energy states will inevitably affect its efficiency. ELP, on the contrary, was developed solely for global optimization, and thus needs no prior running steps. The combination of ELP and BH will then have nothing to lose, but may only benefit from one another.

After the review of the ELP method in the next section, the basin paving (BP) method will be proposed in Sec. 4.2. BP is derived from BH and ELP with some modification in the weight determination procedure to catch all the lower energy samplings so that the lowest energy minimum can be sampled as soon as possible. Its application to pentapeptide Met-enkephalin and protein villin HP-36 will be presented in Sec. 4.3.

4.1 Energy Landscape Paving Method

The energy landscape paving method is designed by combining the core idea from energy space deformation [38, 39] and the tabu search [35, 36, 37] to escape entrapment in local minima, and to direct the search towards unexplored regions. The key characteristic of ELP is to perform a Monte Carlo simulation with a modified energy expression, which is updated with the simulation time to maintain a short-term memory (by histogram collecting) of the states already visited, to steer the search away from those states.

Specifically, ELP can be considered as a Monte Carlo method with the statistical weight of a state [40] given by

$$w(\varepsilon) = e^{-\varepsilon/k_B T} , \tag{4.1}$$

where T is the temperature and k_B is the Boltzmann constant. ε is the replacement of the configuration energy E

$$E \to \varepsilon = E + f(H(q,t)) , \tag{4.2}$$

where $f(H(q,t))$ is a function of the histogram $H(q,t)$ according to a pre-chosen "order parameter" q, and t is the simulation time or specifically the MC step. The histogram is updated at *each* MC step, so that $H(q,t)$ is time dependent. Consequently, the simulation procedure keeps track of the frequency of the prior exploration of a particular region described by the order parameter q. Searching is then discouraged from exploring that region again since the "memory", recorded in the histograms, will be reflected by the change of the weight.

For a ELP simulation, T is often set to a low temperature so that the sampling is biased towards the local energy region. In this case, the probability for escaping from a local minimum depends mainly on the height of the surrounding energy barriers. Within ELP, the histogram covering the region around the local minimum will be paved up, hence the weight of the states around the local minimum decreases with time. Consequently, the probability for escaping the entrapment of local minima increases. After the simulation has escaped from the local entrapment, the paved histogram discourages it from going back again but guides it to other states. For an equally paved histogram $H(q,t)$, the searching procedure will favor low energies because of the initial low temperature, and hence no unphysically high energies are sampled. The paving process deforms the original energy landscape and forces the simulation either to fall in a new local minimum or to walk through higher energy regions. In the latter case, the histograms in the high energy region will be paved up so that the simulation will be able to explore down to the lower energy region again. The histogram "memory" prevents the searching procedure from falling back to the region already explored. However, revisitation is not completely forbidden with the time evolvement when the old "memory" is mixed with new "memories". In other words, the initial low temperature ensures the low energy exploration, while the histogram paving guarantees that the simulation can escape the local entrapment.

The choice of the histogram function $f(H(q,t))$ is quite flexible. Obviously, ELP will reduce to a generalized ensemble approach when $f(H(q,t)) = f(H(q))$. For instance, $f(H(q,t)) \propto \ln(H(E))$ will reduce the ELP method to the multicanonical sampling method. Different choices of the order parameter q in setting up the histogram and different functional expressions of $f(H(q,t))$ determine the diversity of the ELP implementations. The simplest option is to choose the energy E as the "order parameter" and $f(H(q,t))$ proportional to the histogram distribution at each step directly, i.e., $f(H(q,t)) = cH(E,t)$, with c a constant having energy units. Hence, the statistical weight of a state with energy E sampled at time t is

$$w(E,t) = e^{-(E+cH(E,t))/k_BT}. \tag{4.3}$$

Hansmann and Wille performed the optimization of the pentapeptide Met-enkephalin using the above weight with the parameter c simply set to 1 kcal/mol [40]. Arkın

and Çelik applied the same weight expression to the simulation of the heptapeptide deltorphin and compared ELP with MUCA [42]. In Ref. [123], Schug *et al.* studied the sampling efficiency for different values of c and for different simulation temperatures T applied to the Trp-cage protein system. They found that the value $c = 0.05$ kcal/mol gave the best performance. The optimal temperature in their study was $T = 5$ K, and no obvious efficiency difference was found when $T > 50$ K.

Hansmann and Wille proposed another histogram function that is proportional to the histogram determined by both the configuration energy E and the helicity parameter n_H for the study of villin HP-36 protein [40]. n_H is defined as the number of residues that are part of an α helix structure in a peptide or a protein. For a residue, if the pair of backbone dihedral angles (ϕ, ψ) takes a set of values in the range of $(-70\,^\circ \pm 20\,^\circ, -37\,^\circ \pm 20\,^\circ)$, it will be considered as helical. $f(H(q,t))$ can then be replaced by $cH(E, n_H, t)$ now, so that the ELP weight will be

$$w(E, n_H, t) = e^{-(E+cH(E,n_H,t))/k_B T}. \tag{4.4}$$

In Ref. [123], they tested the efficiency for different factors c and temperatures T. Again, $c = 0.05$ kcal/mol and $T = 5$ K were found to provide the lowest energy. Further, introduction of the helicity parameter n_H leads to lower sampled energies and faster sampling of the lower energy region. However, we have to clarify here that the helicity parameter is only applicable to helix-rich secondary structures, such as the HP-36 and Trp-cage proteins. For a system with a native configuration that lacks helical content, the introduction of n_H may reduce the probability of sampling the neighborhood of the native structure, and hence worsen the convergence speed or cause failure in locating the global minimum structure. When the experimental native structure of a protein is unknown, n_H should not be adopted in simulations in order to avoid over-sampling of uninteresting regions.

The ELP method was applied by Hsu *et al.* to determine the X-ray structure of organic molecules [41], and they compared its efficiency with the SA method, as we mentioned at the beginning of this chapter. The weight comparison strategy they employed is different than that described above. Suppose that the old configuration has energy E_{old}, with collected histogram $H(E_{\text{old}})$, and the new configuration has energy E_{new} and histogram $H(E_{\text{new}})$ at step t. The acceptance probability is then

defined as

$$P \equiv \frac{w(E_{\rm new})}{w(E_{\rm old})} = e^{-\Delta\varepsilon(E_{\rm new},E_{\rm old},t)/k_BT} \tag{4.5}$$

with

$$\Delta\varepsilon(E_{\rm new},E_{\rm old},t) = (E_{\rm new}-E_{\rm old}) + c\frac{H(E_{\rm new},t)-H(E_{\rm old},t)}{H(E_{\rm new},t)+H(E_{\rm old},t)}\,. \tag{4.6}$$

In the above expression, the weight parameter c is chosen to be of order $\mathcal{O}(n_F)$, with n_F being the number of degrees of freedom of the molecules. Their results showed that ELP is more efficient than SA by an order of magnitude.

4.2 Basin Paving Method

A successful optimization technique should be able to overcome two difficulties for general simulations: the local entrapment problem and insufficient low energy exploration. Most Monte Carlo simulations may be able to solve one difficulty or another successfully, but may not be able to provide a thorough solution to both of them. Those MC methods based on histogram accumulation, including the ELP method we just discussed, are all able to move out of the local energy wells to escape from local entrapment. By the non-Boltzmann weight accumulated from histogram update, simulations are pushed to the low energy region as well. However, these approaches have boundary problems, which means that at the lower or higher energy end, simulation either provides biased statistical information or the visiting frequency is not sufficient for statistical consideration. For example, the MUCA method often has difficulty in sampling the lowest energy region enough to provide an unbiased weight for the last few energy bins. Schultz *et al.* discussed the boundary problems in the application of the Wang-Landau algorithm [124]. If only the system minimum is of interest, one may not have to worry about the upper boundary. However, the low energy region sampling must still be improved. The MUBH method provides a solution to this difficulty by adopting a local minimization procedure to pin down the bottom of an energy "basin", while adopting the multicanonical weight to avoid local entrapment. The ELP method can sample the low energy region in some detail while paving up the histogram due to the

low initial temperature setup. However, once it moves out of the local energy region, the simulation takes some time to sample the high energy region due to the recorded "memory". The simulation refuses to move back to the low energy region even when a lowest energy configuration is sampled, as it lies in a region recently visited.

To this end, we introduce a new optimization algorithm, the basin paving (BP) method, which is based on the idea of the ELP method and the energy landscape transformation by a local minimization procedure as in BH and MUBH. The local energy minimization makes sure that the movement of each step is from the bottom of one potential well to the bottom of another one. The high energy barriers of the original energy landscape are all neglected. Configuration space is divided into bins as in ELP for paving up so that sampling can leave deeper basins (low energy) to shallower basins (relatively high energy). However, once a deeper basin is located again, the simulation procedure will definitely accept it so that no lower energy configuration will be missed. By making this critical modification to the ELP method, lower energy region sampling is further enhanced.

Just as in MUBH, each Monte Carlo step in BP requires a local minimization process. Consider a physical system described by a potential function $E(\mathbf{r})$ with $\mathbf{r}$ the multidimensional coordinates. Simulation is started from a random initial configuration $\mathbf{r}$. The local minimum $\mathbf{r}_{\min}$, which have the energy $E(\mathbf{r}_{\min})$, is then determined precisely by a local minimization procedure such as the quasi-Newton optimization method [125]. Then, a small trial "move" from $\mathbf{r}_{\min}$ to a new configuration $\mathbf{r}'$ is achieved. The local minimum configuration $\mathbf{r}'_{\min}$ and its energy $E(\mathbf{r}'_{\min})$ are obtained by reperforming local minimization. Acceptance of the move from $\mathbf{r}_{\min}$ to $\mathbf{r}'_{\min}$ is determined by their weights. This procedure is performed iteratively, and is equivalent to transforming the potential energy landscape $E(\mathbf{r})$ to a new one $\tilde{E}(\mathbf{r})$ which contains only the local minima of $E(\mathbf{r})$, i.e.,

$$\tilde{E}(\mathbf{r}) = \min\{E(\mathbf{r})\} = E(\mathbf{r}_{\min}) \tag{4.7}$$

From now on, we will write $\tilde{E}(\mathbf{r})$ as $\tilde{E}$ and $\tilde{E}(\mathbf{r}')$ as $\tilde{E}'$ for simplicity. It should be understood that they are configuration dependent functions.

If the Boltzmann weight is employed to determine the acceptance, it is obviously

the BH method, while the multicanonical weight will lead to the MUBH method. The acceptance criterion for BP used here is similar to that of the ELP method, except that a modification is made to catch all the lower energy moves. If the reduced energy $\tilde{E}'$ of the new configuration $\mathbf{r}'$ is smaller than the original reduced energy $\tilde{E}$ of configuration $\mathbf{r}$, this step is surely accepted. Otherwise, the ELP weight based on the deformed energy expression will be applied. The acceptance probability of a new step can be expressed as

$$P(\tilde{E}, \tilde{E}', t) = \begin{cases} 1, & \text{if } \tilde{E}' < \tilde{E}\,, \\ \frac{w(\varepsilon(q',t))}{w(\varepsilon(q,t))}, & \text{if } \tilde{E}' \geq \tilde{E}\,, \end{cases} \tag{4.8}$$

where $\varepsilon(q,t)$ and $\varepsilon(q',t)$ are the deformed energies taking account the histogram accumulation at step t according to the order parameter q and q', respectively. The general expression of $\varepsilon(q,t)$ can be written as

$$\varepsilon(q,t) = \tilde{E} + f(H(q,t)) \tag{4.9}$$

with the corresponding weight

$$w(\varepsilon(q,t)) = e^{-\varepsilon(q,t)/k_B T}. \tag{4.10}$$

As we will show later, this bias does not overly exaggerate the sampling of the low energy simulation. Further, this small bias meets our requirement of searching the low energy space intensively while keeping the ability to surpass high energy barriers. In principle, it could improve the simulation efficiency.

Similar to the ELP method, the BP weight is quite flexible, due to the diversity of the functional expression of $f(H(q,t))$ and the order parameter q. For the BP applications discussed in the following sections, we will adopt the simplest format based on the distribution of the energy histogram, i.e.,

$$w(\varepsilon(\tilde{E},t)) = e^{-\varepsilon(\tilde{E},t)/k_B T} = e^{-(\tilde{E}+cH(\tilde{E},t))/k_B T} \tag{4.11}$$

for the weight of the simulation procedure having reduced energy $\tilde{E}$ at step t. $H(\tilde{E},t)$ stands for the value of the histogram at the location of $\tilde{E}$, with the visiting frequency accumulated from the start of the simulation to the present time t. c is the histogram weighting parameter in units of energy.

4.3 Application to Protein Molecules

We next apply the BP method to study two examples of the protein molecules: the pentapeptide Met-enkephalin and the villin subdomain HP-36. The empirical force fields ECEPP/2 [68, 69] and ECEPP/3 [70] implemented in a software package SMMP (Simple Molecular Mechanics for Protein) [96] are employed here to describe the interatomic interactions of the systems studied. More details on protein structures, protein models and empirical force fields will be given in Appendix A.

4.3.1 Application to Met-enkephalin

Met-enkephalin is an endogenous opioid pentapeptide found in the human brain, pituitary and peripheral tissues, and is involved in a variety of physiological processes. It has the residue sequence of TYR-GLY-GLY-PHE-MET. In practical simulation, NH_2 and COOH are often chosen as the N-terminus and C-terminus neutral groups, respectively. The pentapeptide consists of totally 75 atoms described by 24 independent backbone and side chain dihedral angles. Even this small peptide gives rise to a very complex conformational space and the total number of local minima was estimated to be more than 10^{11} [126]. It has been intensively studied, and the lowest energy configuration is known with both the ECEPP/2 potential [24, 126, 127, 128, 129] and the ECEPP/3 potential [128, 130]. Local minima with energies not much higher than the global minimum were sampled and classified by Freyberg and Braun [127] using the ECEPP/2 potential, and by Eisenmenger and Hansmann [128] using both the ECEPP/2 and ECEPP/3 potential, but with the peptide dihedral angle ω fixed at $180\,^\circ$. Nowadays, Met-enkephalin has become a benchmark model frequently used for testing new algorithms because of the complexity in its configuration space, yet still small enough to be studied extensively using available computational resources. We list its known global minima (or lowest energies ever found) in Table 4.1.

In our study of the Met-enkephalin, BP was employed to obtain the lowest energy configurations in all four cases: ECEPP/2 and ECEPP/3 with the peptide angle ω fixed or relaxed. For the BP simulations, the simplest weight expression of

Energy (kcal/mol)	Force Field
-12.91	ECEPP/2 [24, 126, 127, 129]
-10.72	ECEPP/2 with ω fixed at 180 ° [128]
-11.71	ECEPP/3 [130]
-10.85	ECEPP/3 with ω fixed at 180 ° [128]

Table 4.1: The lowest energies obtained in previous studies of Met-enkephalin.

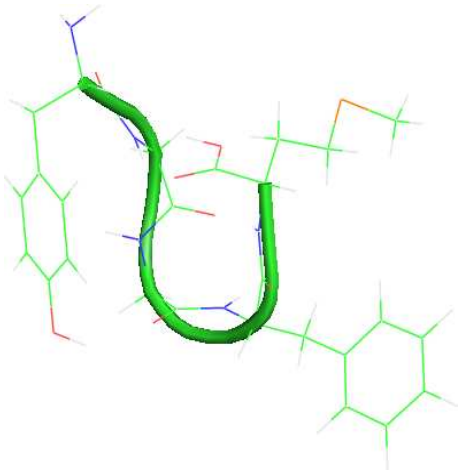
Eq. (4.11) with the factor $c = 1$ kcal/mol is used. The lowest energies we obtained and their configurations expressed in internal coordinates are listed in Table 4.2. In the table, the labels E_{II} and E_{III} are used to stand for the configurations obtained using the ECEPP/2 and ECEPP/3, respectively. If the peptide dihedral angle ω is fixed at 180 °, a prime will be added to the labels. The structures obtained are also shown in Fig. 4.1 [1] from (a) to (d) correspond to cases E'_{II}, E_{II}, E'_{III}, and E_{III}, respectively. We are able to reproduce the global minima under the ECEPP/2 potential no matter whether ω is fixed or relaxed. For case E'_{III}, i.e. when the ECEPP/3 potential with ω fixed is used, we found a new lowest energy minimum configuration, which has the energy $E = -10.90$ kcal/mol, about 0.05 kcal/mol lower than the one found in Ref. [128], denoted as $E'^{(a)}_{\text{III}}$. The difference between their structures can be obviously seen from Fig. 4.1 (c) and (e), and their internal coordinates in Table 4.2. For the ECEPP/3 potential with ω relaxed, we found the lowest energy configuration to be similar to the one found in Ref. [130], labeled as $E^{(b)}_{\text{III}}$ in Table 4.2, except that our energy value is different from theirs. The difference in energies comes from the fact that there is some difference in the force field used. For the ECEPP/3 force field used in Ref. [130], two extra terms, the cystine loop-closing term and the cystine torsional term, are included in the potential. From Table 4.2, and also from Fig. 4.1 (b) and (d), it is also obvious that the configurations of both the ECEPP/2 and ECEPP/3 potential with ω relaxed are very close to one another, even though their potential energies have some difference.

In the simulation, we tested the influence of temperature to the convergence

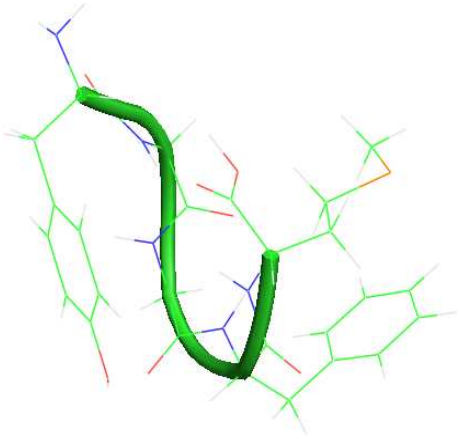
[1] All the figures of the protein/peptide structures drawn in this thesis have been generated using PyMOL [131].

	Torsion	E'_{II}	E_{II}	E'_{III}	E_{III}	$E'^{(a)}_{\text{III}}$	$E^{(b)}_{\text{III}}$
1, TYR	χ_1	-179.8	-172.6	59.9	-173.2	-174.2	-173.2
	χ_2	68.6	-101.3	94.1	-100.7	-85.2	-100.5
	χ_6	-34.7	14.1	-21.3	13.7	2.8	13.6
	ϕ	-86.3	-85.8	168.1	-83.1	-162.7	-83.5
	ψ	153.7	156.2	0.9	155.8	-41.7	155.8
	ω	180.0	-176.9	180.0	-177.1	180.0	177.2
2, GLY	ϕ	-161.5	-154.5	126.8	-154.2	65.8	-154.3
	ψ	71.1	83.7	-21.2	85.8	-87.0	86.0
	ω	180.0	168.6	180.0	168.5	180.0	168.5
3, GLY	ϕ	64.1	83.7	83.7	83.0	-157.3	83.0
	ψ	-93.5	-73.9	-61.6	-75.0	34.9	-75.1
	ω	180.0	-170.1	180.0	-170.0	180.0	-169.9
4, PHE	χ_1	179.8	58.8	58.6	58.9	52.4	58.8
	χ_2	-100.0	-85.4	92.9	-85.5	-96.0	-85.5
	ϕ	-81.7	-137.0	-128.2	-136.8	-158.8	-136.9
	ψ	-29.2	19.3	18.8	19.1	159.5	19.1
	ω	180.0	-174.1	180.0	-174.1	180.0	-174.1
5, MET	χ_1	-65.1	52.8	55.7	52.9	-66.1	52.9
	χ_2	-179.2	175.3	-178.6	175.3	-179.6	175.3
	χ_3	-179.3	-179.8	177.0	-179.9	-179.9	-179.9
	χ_4	-179.9	61.4	-179.3	-178.6	60.1	61.4
	ϕ	-80.7	-163.6	-162.1	-163.4	-82.4	-163.5
	ψ	143.5	160.4	7.5	160.8	134.1	161.0
	ω	180.0	-179.7	180.0	-179.8	180.0	-179.8
E (kcal/mol)		-10.72	-12.91	-10.90	-12.43	-10.85	-11.71

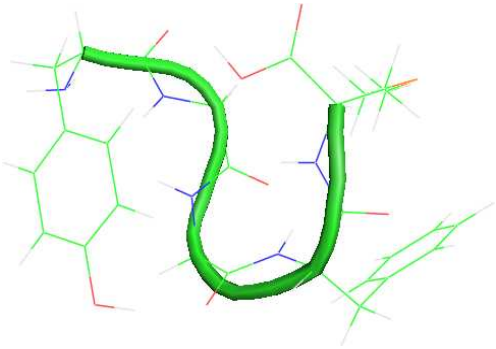
Table 4.2: The global minimum structures of Met-enkephalin in internal coordinates. The labels E_{II} and E_{III} denote that the structures are obtained using the ECEPP/2 and ECEPP/3 potentials, respectively. A prime on the label means that the peptide angles ω of the structure are fixed at 180 °. Superscripts (a) and (b) indicate results obtained in Ref. [128] and [130], respectively.



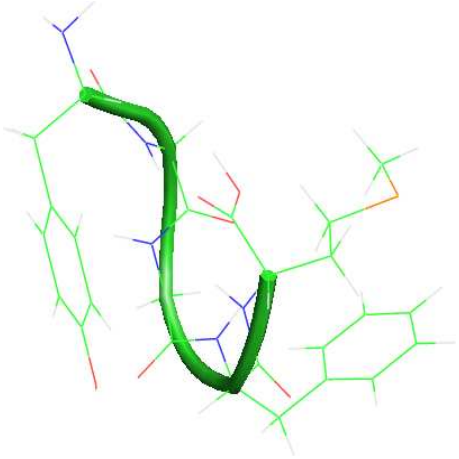
(a) ECEPP/2 with $\omega = 180\,^\circ$



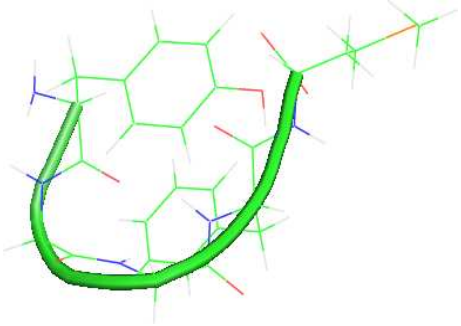
(b) ECEPP/2 with relaxed ω



(c) ECEPP/3 with $\omega = 180\,^\circ$ found using BP



(d) ECEPP/3 with relaxed ω



(e) ECEPP/3 with $\omega = 180\,^\circ$ reported in Ref. [128].

Figure 4.1: The lowest energy structures of Met-enkephalin.

speed of the BP method. In the test, we only used the ECEPP/3 potential with ω-fixed case. The temperatures we used are 5 K, 50 K, 500 K, 1 000 K and 2 000 K. For all these tests, we cannot see any obvious difference in efficiency caused by the temperature T. Hence, the simulation temperature in the BP method seems not as important as in other Monte Carlo methods. This is because that the histogram paving process can easily push the simulation from the lower energy region to the high energy region and vice versa. The initial temperature setting only dominates the first several MC steps. Figure 4.2 shows an example of the histogram distribution obtained after 20 000 Monte Carlo steps at the temperatures used. Also shown in this figure is the distribution obtained using the BH method at $T = 2\,000$ K. To demonstrate the effect introduced by accepting all the moves to the low energies as in Eq. 4.8, we combined BH and ELP together directly, which we call the BH+ELP method. It is an ELP simulation on the reduced energy surface obtained by performing a local minimization procedure at each step. The acceptance probability for BH+ELP is simply

$$P(\tilde{E}, \tilde{E}', t) \equiv \frac{w(\varepsilon(q', t))}{w(\varepsilon(q, t))} \tag{4.12}$$

for all the energies $\tilde{E}$ and $\tilde{E}'$, with the symbols here having the same meaning as those in Eq. (4.8). For the present simulation, the weight in the above expression takes the form of Eq. (4.11) with $c = 1$ kcal/mol. The histogram distribution of BH+ELP with temperature $T = 50$ K is plotted in Fig. 4.2 as well. Both the plots for BH and BH+ELP are also the snapshots after 20 000 MC steps. It is obvious from the figure that BP can "uniformly" visit the low energy region compared to BH, and has improved low energy visiting frequency compared to BH+ELP. Meanwhile, it keeps the ability of sweeping the high energy region for overcoming energy barriers, but has the trend of going back to low energy regions so that the simulation should not waste too much computational time, compared to BH+ELP, in the high energy region.

A further illustration of the ability of the BP method to cover a wide energy range is shown in Fig. 4.3. We can see from the MC trajectories that it takes nearly no time for the BP method to gain the ability of searching both the low and the high energy regions, no matter what the temperature is. Specifically, for the trajectories

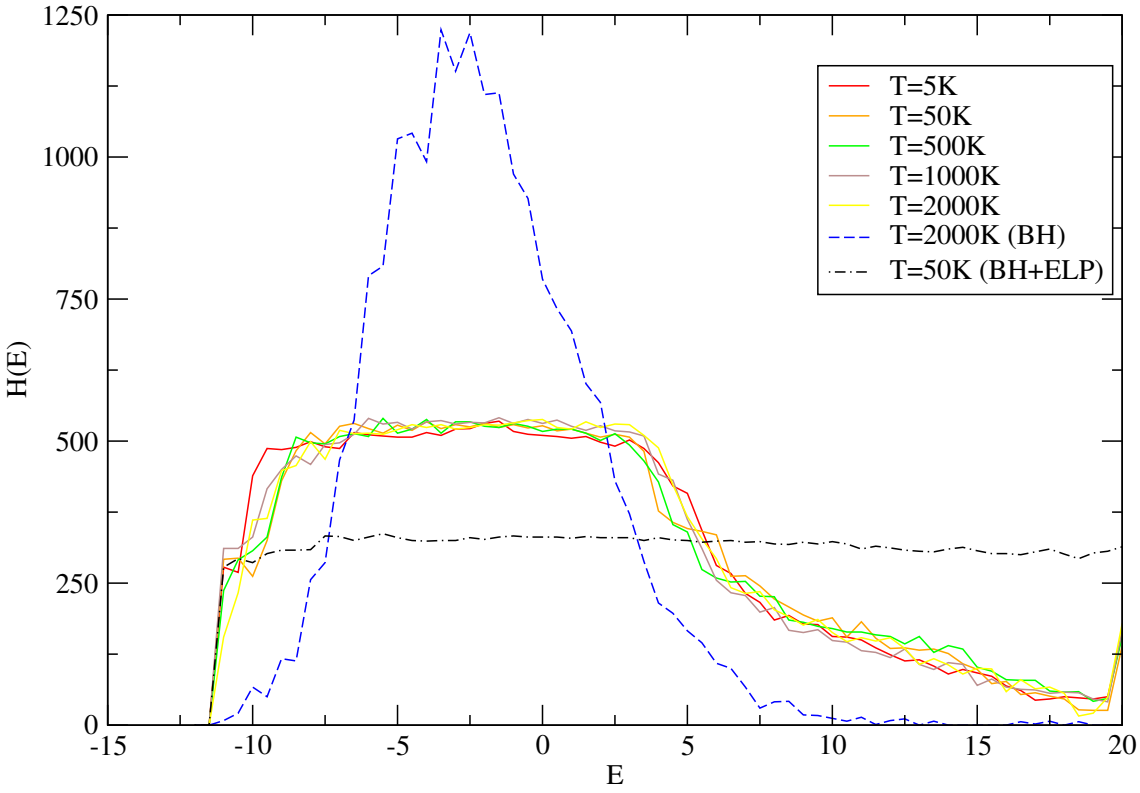


Figure 4.2: Histogram distributions obtained using BP at various temperatures, BH at $T = 2\,000$ K and BH+ELP at $T = 50$ K.

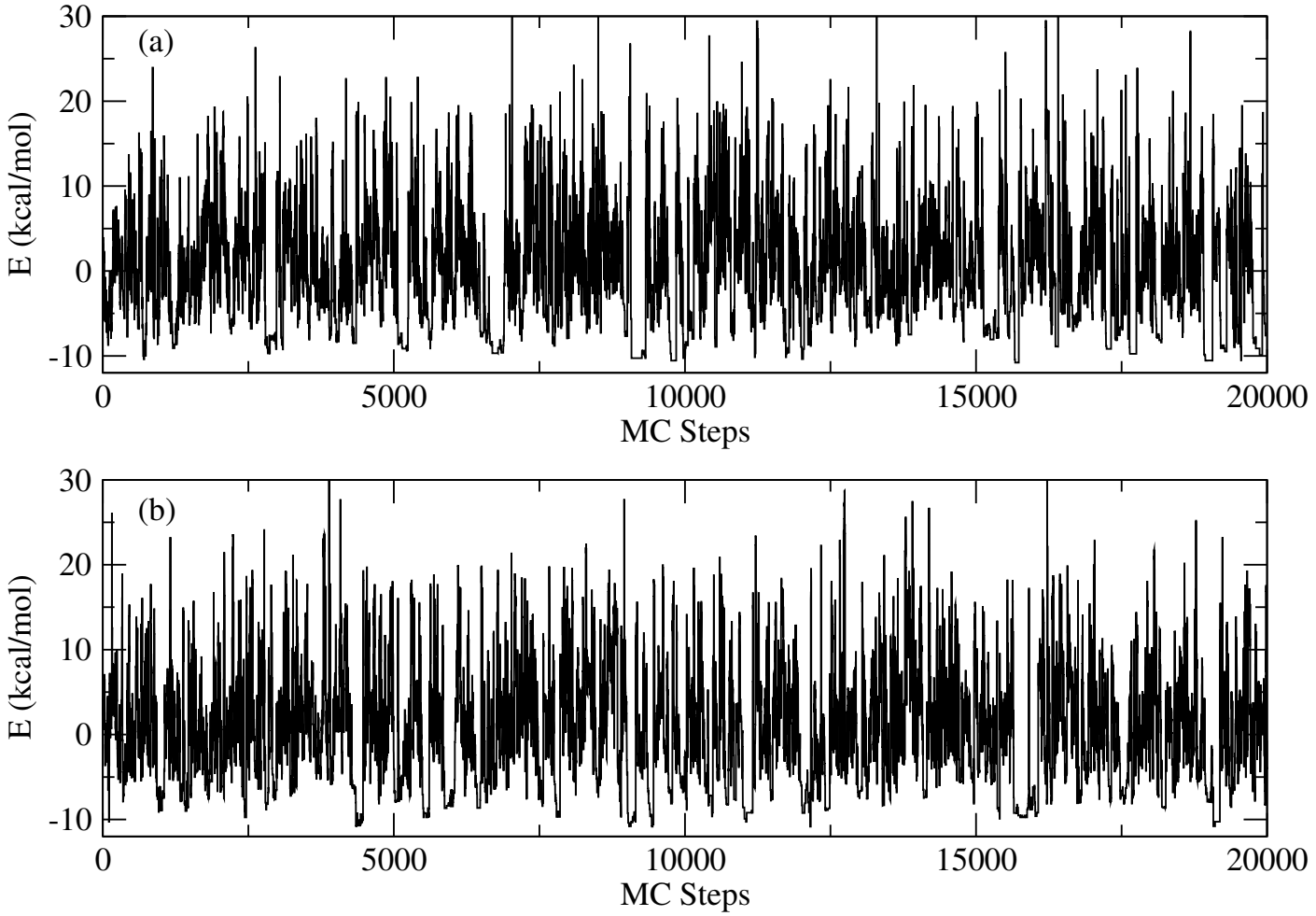


Figure 4.3: Typical searching trajectories of the BP method with temperature (a) $T = 5$ K and (b) $T = 2\,000$ K.

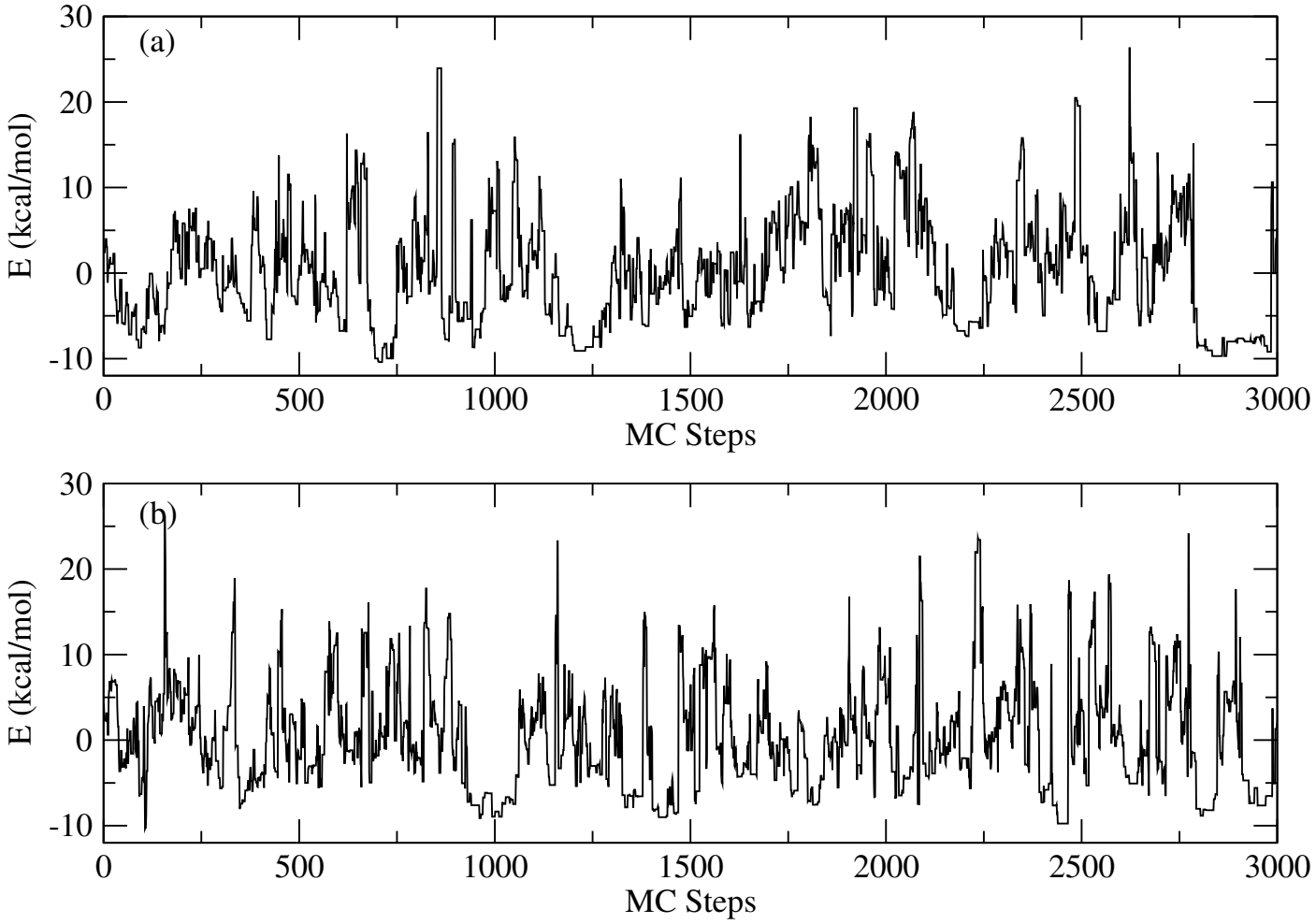


Figure 4.4: Detailed illustration of the searching trajectories at the beginning of the simulations with temperature (a) $T = 5$ K and (b) $T = 2\,000$ K.

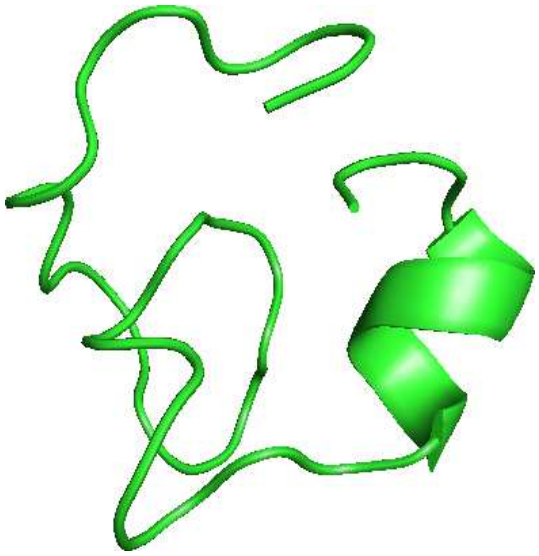
in Fig. 4.3, it takes less than 700 MC steps for the low temperature ($T = 5$ K) simulation to reach the very high energy regions. With a high temperature at $T = 2\,000$ K, the simulation immediately gains the ability of sampling both high and lower energy regions. More detail is shown in Fig. 4.4, which illustrates the trajectories of the first 3 000 MC steps.

4.3.2 Application to Villin HP-36

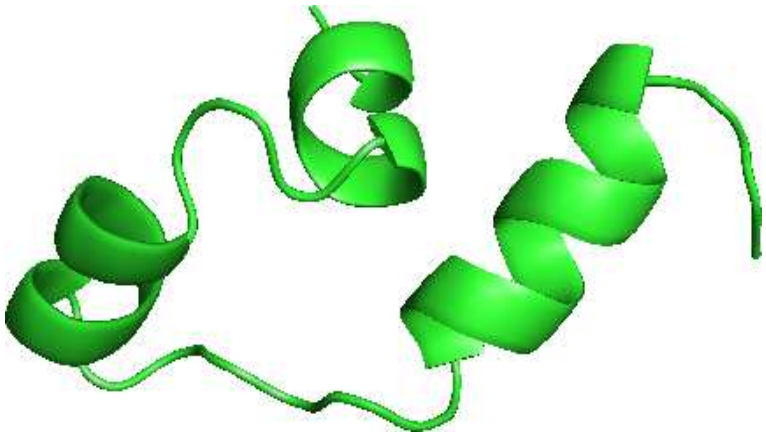
The villin subdomain HP-36, which contains 36 residues with a total of 597 atoms [40], is one of the smallest proteins that can fold automatically. It is the stable subdomain formed by the C-terminus residues of the "headpiece", which in turn is the C-terminus 76-amino acid domain of the actin-bundling protein villin from chickens [132]. The high thermal stability of the villin is largely determined by the subdomain HP-36. HP-36 contains only naturally occurring amino acids which can fold automatically into a unique and thermally stable structure and does not require disulfide bonds, oligomerization or ligand binding to retain its stable structure. Its melting temperature is about 70 °C in adequate solution at pH 7.0 [132]. It is believed to be one of the fastest folding proteins, with folding time determined to be about 4.3 μs at 300 K using the laser temperature-jump method [133], and on the time scale of 10 μs using dynamic line-shape analysis [134]. The experimental structure deposited in the Protein Data Bank [135, 136] with PDB code 1vii is obtained from nuclear magnetic resonance (NMR) studies [132], which reveal three short helices as shown in Fig. 4.5 (d). Starting from the N-terminus, residues 4 to 8 form helix 1, residues 15 to 18 form helix 2, and residues 23 to 30 form helix 3. The helices are connected by a loop from residues 9 to 14 and a turn from residues 19 to 22. Since the protein can fold fast and is small enough to be applicable to the present computers, there have been many simulations performed for HP-36 both by molecular dynamics methods as reported in Ref. [137, 138, 139], and by Monte Carlo methods as reported in Ref. [40, 101, 140, 141]. Duan and Kollman probably performed the first computational simulation of HP-36 using a parallel molecular dynamics method [137]. They simulated the protein folding procedure with explicit water which contains about 3 000 water molecules for 1 μs.

The BP approach was applied to study the folding of HP-36 in this thesis. We perform simulations *in vacuo* under the ECEPP/2 force field with the peptide angle ω fixed at 180 ° for convenience of comparing with the result from Ref. [101]. The functional expression of $f(H(q,t))$ is chosen to be the same as that used in optimizing the Met-enkephalin peptide. Note here, that unlike the simulation performed in Ref. [40], no helicity parameter n_H is included in the expression. Since n_H-dependent histogram distributions will favor α helix structures, while HP-36 has a rich helix native configuration as obtained from the experiment and previous simulations, the inclusion of n_H will improve the sampling in the helical region and bias the folding procedure toward the experimental structure. One of our purposes is to check BP's ability to study the folding of general sequences here, so that n_H is not considered in this simulation.

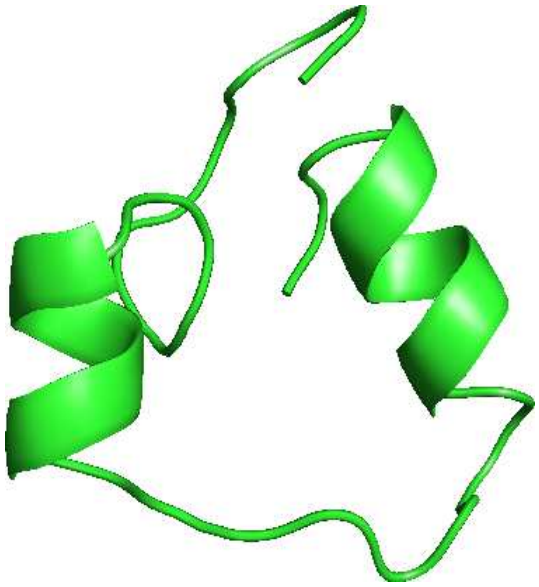
Figures 4.5 (a), (b) and (c) show the configurations obtained by BP simulation of HP-36 *in vacuo* with energy $E < -205.0$ kcal/mol. Figure 4.5 (a) has the lowest energy $E_{\rm I\!I} = -209.65$ kcal/mol. Its energy is lower than that reported in Ref. [101], -209.2 kcal/mol, even though the energy difference between them is not too large, less than 0.5 kcal/mol. For both the lowest energy structure of Fig. 4.5 (a) and the structure from Ref. [101] (refer to as PTRS, the Parallel Tempering Reference Structure), there are obvious low helicities. In fact, there is only one short helix formed by residues 28–33 for the lowest energy structure. For PTRS, which we failed to locate, the single short helix is located between helices 23 and 28. Their structures have less similarity compared to the NMR structure shown in Fig. 4.5 (d). The difference can also be reflected by their backbone root-mean-square deviation (RMSD), $R_{\rm rmsd} = 7.04$ Å for the structure of Fig. 4.5 (a) and $R_{\rm rmsd} = 7.4$ Å for PTRS. Note here that all the RMSD values we mention in this section are obtained by comparing the simulated structures with the NMR one. Figure 4.5 (b), with $E_{\rm I\!I} = -206.78$ kcal/mol, is the second lowest energy configuration that we obtained (the third lowest if the PTRS is counted), and is similar to the NMR structure. It consists of three helices as in the NMR structure although their locations are a little different. The first helix is formed by residues 2–7 and the second helix consists of residues 11–17, while in the NMR structure, the helices are formed by residues 4–8 and 15–18, respectively. The third helix is the



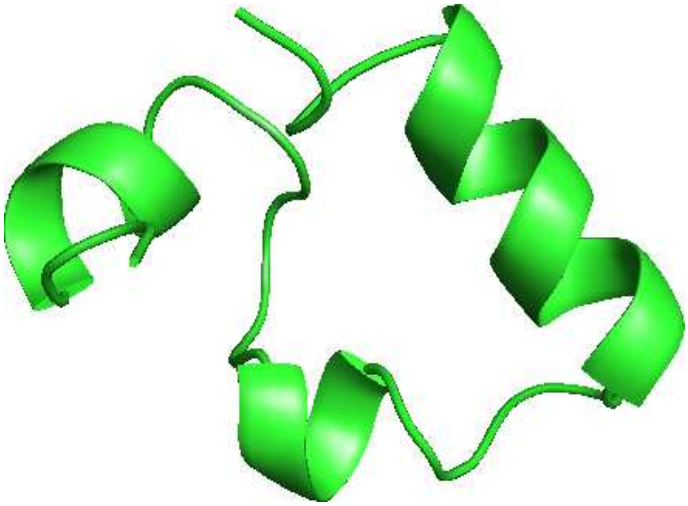
(a) $E_{\mathbb{I}} = -209.65$ kcal/mol
$R_{\text{rmsd}} = 7.04$ Å



(b) $E_{\mathbb{I}} = -206.78$ kcal/mol
$R_{\text{rmsd}} = 7.82$ Å



(c) $E_{\text{II}} = -205.82$ kcal/mol
$R_{\text{rmsd}} = 6.00$ Å



(d) $E_{\text{II}} = -176.1$ kcal/mol
NMR structure

Figure 4.5: Low energy configurations of HP-36. (a)–(c) Configurations obtained using BP simulation with $E_{\text{II}} < -205.0$ kcal/mol; (d) NMR structure with PDB code 1vii. All the configurations are drawn with the N-terminus at the left hand side.

Configuration	helix 1	helix 2	helix 3	$E_{\mathbb{I}}$ (kcal/mol)	$R_{\rm rmsd}$ (Å)
(a)	—	—	28–33	-209.65	7.04
PTRS	—	—	23–28	-209.2	7.4
(b)	2–7	11–17	24–33	-206.78	7.82
(c)	—	12–17	27–33	-205.82	6.00
NMR (d)	4–8	15–18	23–32	-176.1	—

Table 4.3: The location of the helices for the configurations shown in Fig. 4.5. PTRS stands for the configuration obtained in Ref. [101] using the parallel tempering method.

longest one, and consists of residues 24–33 for the BP simulated result, in contrast with the residues 23–32 for the NMR conformation. The most obvious difference between the two structures is the connecting turns and loops between helices, which result in a relatively large RMSD, $R_{\rm rmsd} = 7.82$Å. The structure (c) has the smallest RMSD in all the structures shown; however, its energy is higher than all the others, which include the structures (a), (b) and the PTRS. The residues forming the α helices are 12–17 and 27–33, which correspond to helix 2 and helix 3 of the NMR structure, respectively. Table 4.3 lists in detail the helix positions of the structures we just discussed.

Considering the fact that the present simulations are performed *in vacuo* while the PDB structure is determined in the solvent environment, we believe that the difference between them is inevitable. Nevertheless, by employing the BP optimization method, we have obtained some low energy configurations, and one of them shows similarities to the NMR structure. Furthermore, a structure with lower energy than the one obtained using the parallel tempering method in Ref. [101] has been located. Considering the fact that we have not located the structure of PTRS, we believe that it is quite possible that we have missed some other low energy configurations. in our limited samplings. A more detailed study of the HP-36 subdomain *in vacuo* with the ECEPP force fields, especially in the low energy region, is still required for comparison with the folding in solution to study how the solvent can affect the dynamics of protein folding.

4.4 Summary

In summary, we have introduced a new Monte Carlo global optimization method, the basin paving method, in this chapter. The basin paving method essentially performs an energy landscape paving procedure on the reduced energy landscape generated by a local minimization procedure at each MC step. However, BP specially tunes up the paving process for the purpose of fine sampling the low energy region by catching every low energy move. The paving steps deform the reduced energy landscape by keeping a short "memory" of the states just visited, so that the next sampling steps are discouraged from going back to the region just visited. The BP method retains the ability of surpassing high energy barriers, which is inherited from the ELP method. The local minimization procedure ensures that MC moves only "hop" between local minima. The exact value of the global minimum will be obtained once it is located, unlike the approximate value obtained from the canonical or generalized ensemble Monte Carlo methods due to thermal fluctuations. The BP method has been applied to the Met-enkephalin peptide, and a new configuration with lower energy has been located using the ECEPP/3 potential with the peptide angle ω fixed at 180 °. Its further application to villin subdomain HP-36 protein *in vacuo*, with the ECEPP/2 force field, found a configuration with energy lower than the one previously obtained, and a low energy structure which is similar to the experimentally determined one.

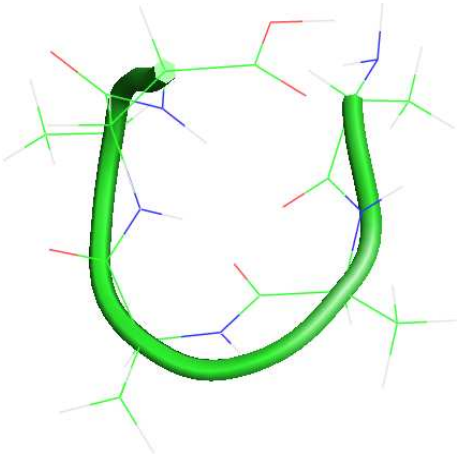
Chapter 5

Protein Folding Simulations

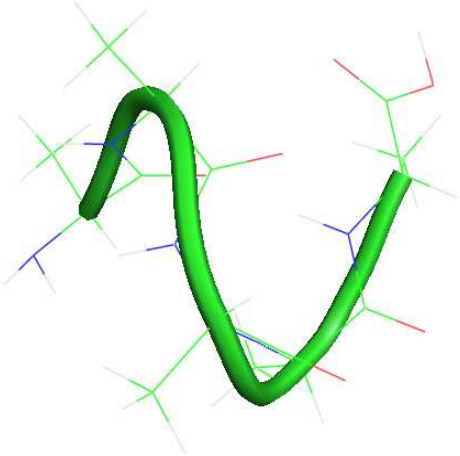
In the previous chapter, the basin paving (BP) global optimization method was applied to the Met-enkephalin peptide and the villin subdomain HP-36 protein. New lower energy minima were obtained, and hence new configurations were located, for both systems. In this chapter, the multicanonical basin hopping (MUBH) method, the basin paving method, together with the basin hopping (BH) method for small systems, will be employed to search for the global minima of several different peptides/proteins. As in Chapter 4, two empirical force fields ECEPP/2 [68, 69] and ECEPP/3 [70] implemented in the software package SMMP [96] will be used for these studies. In all the following applications, the amino group NH_2 and the carboxyl group COOH are used as the neutral N-terminus and C-terminus, respectively. Unless stated explicitly, all peptide angles ω will be fixed at $180\,^\circ$.

5.1 Polyalanine

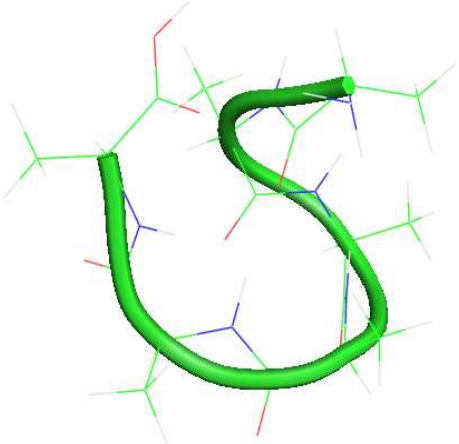
The polyalanine peptide is well-known for its helical structure and has been extensively studied by Poland and Scheraga [142] and many other groups [21, 141, 143, 144, 145, 146, 147, 148, 149]. The global minimum structures of the polyalanine peptides with different length of residues will be studied using both the ECEPP/2 and ECEPP/3 force fields and denote the energies obtained by E_{II} and E_{III}, respectively. We denote a peptide containing N alanine residues as $(\text{ALA})_N$.



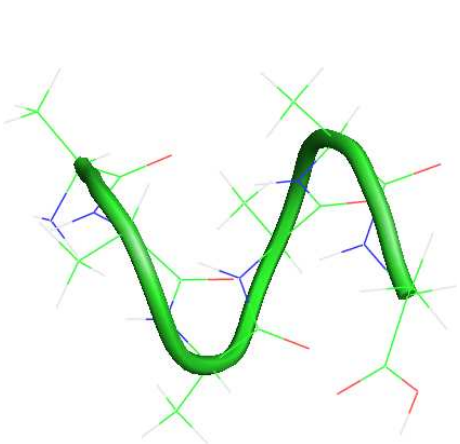
(a) $(\mathrm{ALA})_5$: $E_{\mathbb{II}} = 7.34$ kcal/mol



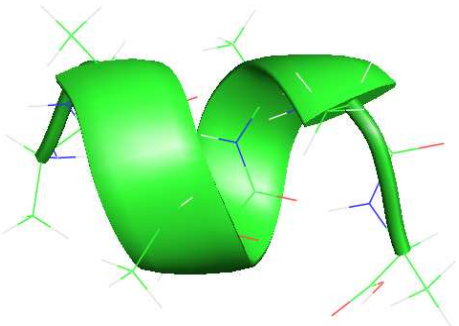
(b) $(\mathrm{ALA})_5$: $E_{\mathbb{III}} = 4.69$ kcal/mol



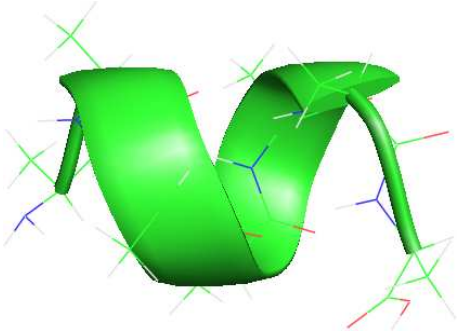
(c) $(\mathrm{ALA})_6$: $E_{\mathbb{II}} = 5.71$ kcal/mol



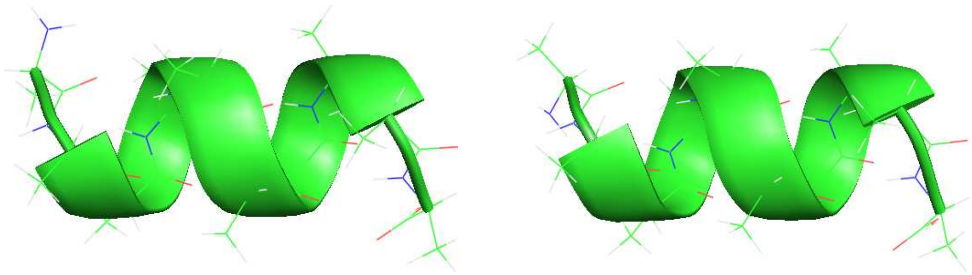
(d) $(\mathrm{ALA})_6$: $E_{\mathbb{III}} = 2.37$ kcal/mol



(e) $(\mathrm{ALA})_7$: $E_{\mathbb{II}} = 3.96$ kcal/mol



(f) $(\mathrm{ALA})_7$: $E_{\mathbb{III}} = 0.09$ kcal/mol



(g) $(ALA)_{10}$: $E_{\text{II}} = -2.64$ kcal/mol (h) $(ALA)_{10}$: $E_{\text{III}} = -7.39$ kcal/mol

Figure 5.1: The global minimum configurations of the polyalanine peptides. E_{II} and E_{III} denote the energies obtained using the ECEPP/2 and ECEPP/3 force fields, respectively.

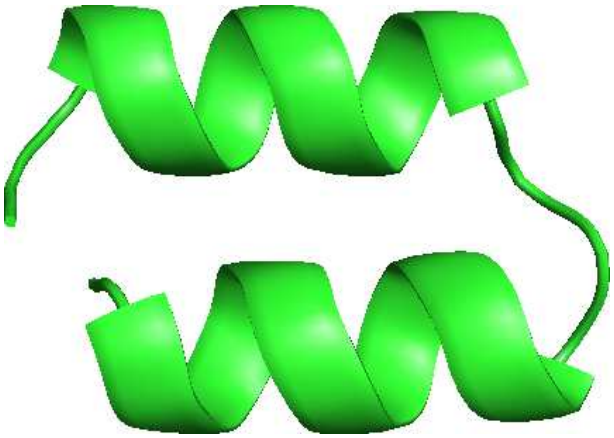
The ECEPP/2 force field was first employed to study folding of the polyalanine peptide. For $N = 5$, 6, 7 and 10, the minimum structures obtained are shown in Fig. 5.1 (a), (c), (e), and (g), respectively. It is obvious from these figures that $(ALA)_N$ begin to show the helical structure when $N \geq 7$. The ECEPP/3 potential was also applied to obtain the global minimum structures of $(ALA)_N$, shown in Fig. 5.1 (b), (d), (f) and (h) for $N = 5$, 6, 7 and 10, respectively. Although it is not as clear as the $N > 7$ cases, the ALA_N peptides begin to have the appearance of a helical structure for smaller values of N, namely, $N = 5$ and $N = 6$. The difference in the structures for small peptides of the same amino acid sequence indicates that using the ECEPP/2 and ECEPP/3 force fields may result in different low energy configurations. There are difference between the two potentials in generating the low energy landscape. From their applications in studying the polyalanine peptides, it seems that simulations using the ECEPP/3 potential prefer helical structures, which will be further studied using some other protein/peptide systems.

There were mainly three modifications introduced when updating the ECEPP/2 potential with the ECEPP/3 potential [128]: (1) The standard geometry and some energy parameters for prolyl and hydroxyprolyl have been updated with more recent experimental findings; (2) The partial atomic charges of backbone atoms have been recalculated; (3) Charges carried by the terminal groups were re-organized to avoid

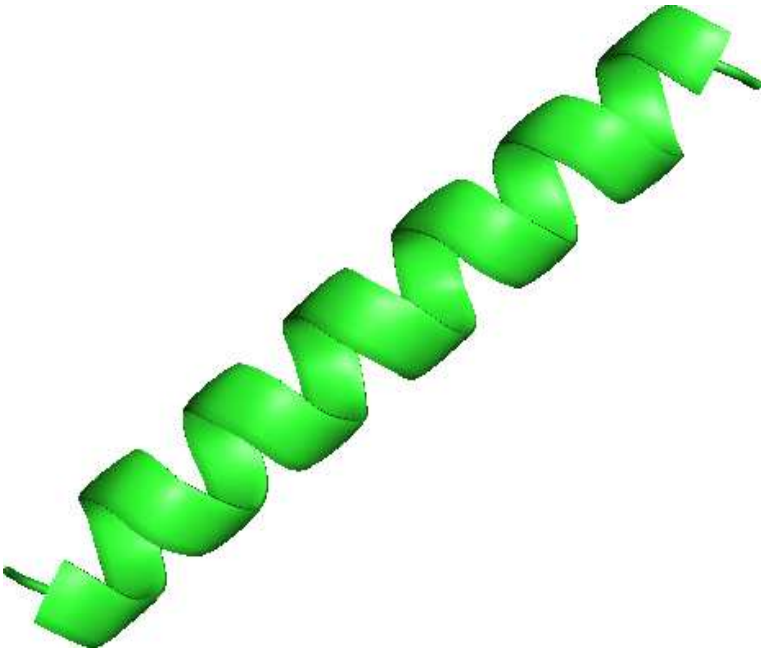
possible artifacts for different types of terminal groups. The first modification should not affect a polyalanine peptide and the third modification does not seem to contribute too much to the helical formation. Hence, the change of the backbone charge enhanced the appearance of the α helical structure when using the ECEPP/3 potential compared to the ECEPP/2 potential.

To further illustrate the minimum configuration change induced by the modification of the force field, both potentials are employed to find the lowest energy minimum of an artificial peptide, $A_{10}G_5A_{10}$, which contains two ten-alanine peptides connected by a five-glycine peptide. Its structural transition has been studied, and the lowest energy configuration was located in Ref. [141] using the ECEPP/2 potential. In this study, the lowest energy structure for both the ECEPP potentials were located, and they are shown in Fig. 5.2. The minimum structure obtained using the ECEPP/2 potential, Fig. 5.2 (a), is the same as that obtained in Ref. [141], which is an anti-parallel structure of two helices (formed by alanine residues), connected by a coil (formed by the glycine residues). When the ECEPP/3 potential is employed for the study, there is only one long helix for the minimum structure, shown in Fig. 5.2 (b), which means that the glycine residues are part of the helix now. The U-turn-like structure similar to the one obtained using ECEPP/2 (Fig. 5.2 (a)) can still be obtained; however, it has an energy higher than the long helical structure. This application further proves that ECEPP/3 favors helical structures more than ECEPP/2.

In these calculations, BH is applied to all the $(ALA)_N$ peptides, and BP is also employed in obtaining the global minimum for the peptide $(ALA)_{10}$. Both MUBH and BP are used to find the global minimum of the artificial peptide $A_{10}G_5A_{10}$. The structures shown in Fig. 5.2 have been obtained using BP with peptide dihedral angles ω fixed at 180 °. The global minima with ω relaxed are $E_{\mathrm{II}} = -47.46$ kcal/mol and $E_{\mathrm{III}} = -47.04$ kcal/mol, which are calculated using the MUBH method. Their minimum structures are similar to the ω-fixed case.



(a) $E_{\text{II}} = -45.55$ kcal/mol



(b) $E_{\text{III}} = -46.20$ kcal/mol

Figure 5.2: The minimum configurations of $A_{10}G_5A_{10}$. (a) was obtained using the ECEPP/2 force field; (b) was obtained using the ECEPP/3 force field.

5.2 Trp-cage

The twenty-residue peptide Trp-cage is by far the fastest folding protein known, which was recently designed by Neidigh *et al.* [150, 151]. It has the residue sequence of NLYIQ WLKDG GPSSG RPPPS, and its native structure contains an α helix from residues 2 to 8 and another helix from residues 11 to 14. The sixth residue, Trp6, is caged by the C-terminus polyproline stretch. Qiu *et al.* provided direct experimental evidence that the folding time of the protein in room temperature is about 4 μs [152]. Due to its small size, high thermal stability and fast folding characteristics, many numerical calculations have been performed to study this miniprotein [153, 154, 155, 156, 157, 158, 159, 160].

In this study, the BP method and the ECEPP/3 force field are employed to obtain the minimum structure of the Trp-cage protein, *in vacuo*, from a fully stretched configuration. The lowest energy configuration obtained is shown in Fig. 5.3 in green. The other one also drawn in the figure in violet is the NMR determined structure [151] deposited in the Protein Data Bank [135, 136] with PDB code 1L2Y. The NMR configuration shown in the figure is the 16th native structure which has the smallest backbone root-mean-square deviation (RMSD) from the simulated configuration, $R_{\rm rmsd} = 2.24$ Å. For the simulated structure, there is only one helix formed by residues 2 to 8. No second helix appears in the structure obtained, which results in the main difference between the present simulation results and the NMR structure.

5.3 VGV peptide

Similar to the polyalanine peptide, a polyvaline peptide *in vacuo* was believed to have a helical structure as its ground state as well [21]. However, the ground state is difficult to reach in a numerical simulation, because the large side chains induce high energy barriers around it [21]. Since the optimal number of hydrogen bonds will be established in a helical structure, the polyvaline peptide will still prefer an alpha helix structure. However, the structure can be changed if a flexible turn is introduced in the middle of the peptide.

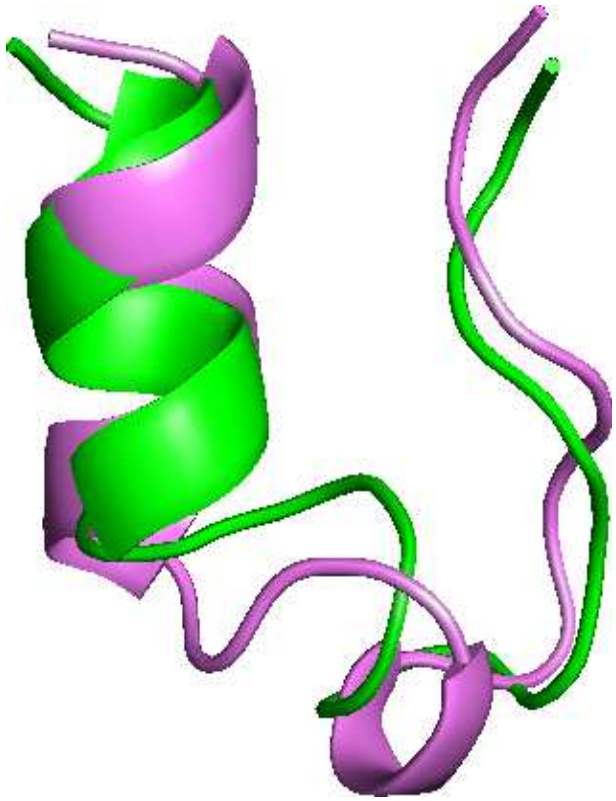
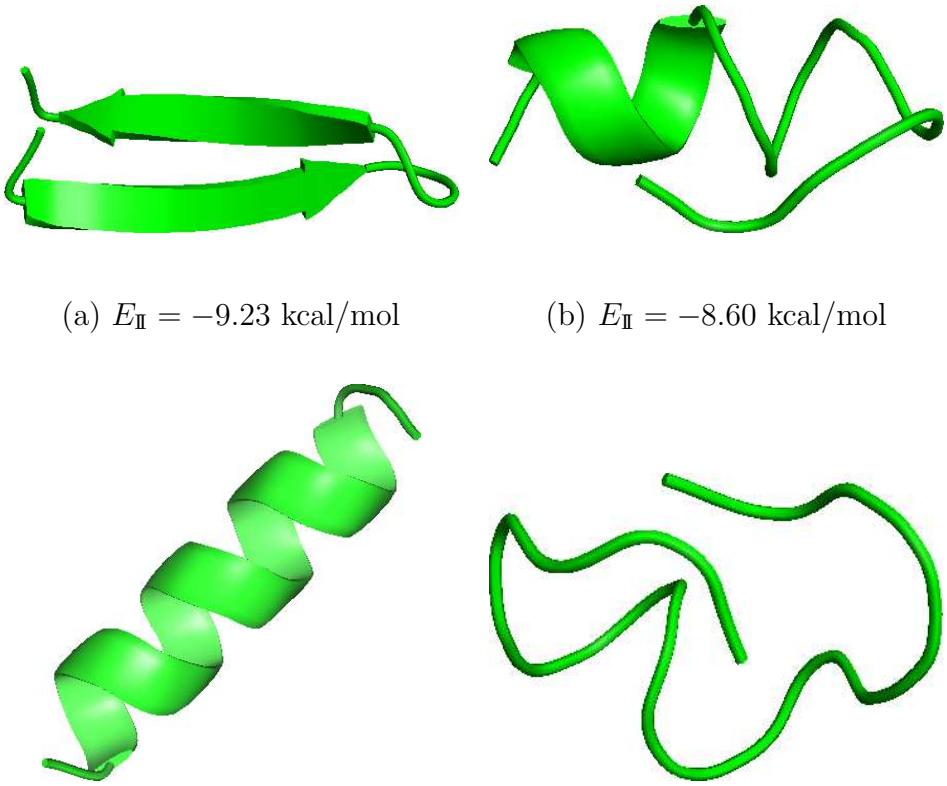


Figure 5.3: The minimum configuration of Trp-cage. The structure in green is the simulation result, and the one in violet is obtained from the Protein Data Bank with PDB code 1L2Y, which is a native state structure determined using the NMR technique [151]. Their RMSD is: $R_{\mathrm{rmsd}} = 2.24$ Å.

An artificial model peptide, the VGV peptide, which has the amino acid sequence of $(\mathrm{VAL})_7$–$(\mathrm{GLY})_2$–$(\mathrm{VAL})_7$, was studied by searching for its global optimum structure. The simulation was performed *in vacuo* using the BP method with the ECEPP/2 potential. Figure 5.4 shows four low energy configurations that stand for four kinds of different structures. Fig. 5.4 (a) shows the lowest energy configuration obtained, with $E_{\mathrm{I}} = -9.23$ kcal/mol, which has the beta hairpin structure. The VGV peptide also has the chance of having helical content, e.g. Fig. 5.4 (b) and (c), and their energies are not too much higher than that for the hairpin structure (< 0.7 kcal/mol). Random coil structures, such as the one shown in Fig. 5.4 (d) with its energy only higher than the global minimum structure by about 1.0 kcal/mol, also occur often in the low energy configurations. In a small energy range of less than 1 kcal/mol, all the three essential structures, i.e. alpha, beta, and coil, and their mixtures (Fig. 5.4 (b) is the combination of helix and coil), appear as possible low energy configurations. Hence, the VGV peptide should be quite flexible at room temperature, where $k_B T$ is about 0.6 kcal/mol.

Once again, simulations were performed using the ECEPP/3 force field as well. Just as the ECEPP/2 case, four lower energy configurations are chosen and shown in Fig. 5.5. They stand for four kinds of possible structures appeared in the simulation. Figure 5.5 (a) shows a full helical structure which has the lowest energy. The helix-coil-helix structure, Fig. 5.5 (b), has higher energy and can evolve easily to the full helix structure shown in (a). The random coil-helix structure, Fig. 5.5 (c), appears very frequently in the low energy configurations. In contrast to the situation when the ECEPP/2 force field is used, beta hairpin structures have much higher energy than the alpha and coil configurations. The configuration shown in Fig. 5.5 (d) is the lowest energy hairpin structure obtained in the simulations, and its energy $E_{\mathrm{III}} = -10.16$ kcal/mol is higher than the global minimum by more than 6 kcal/mol.

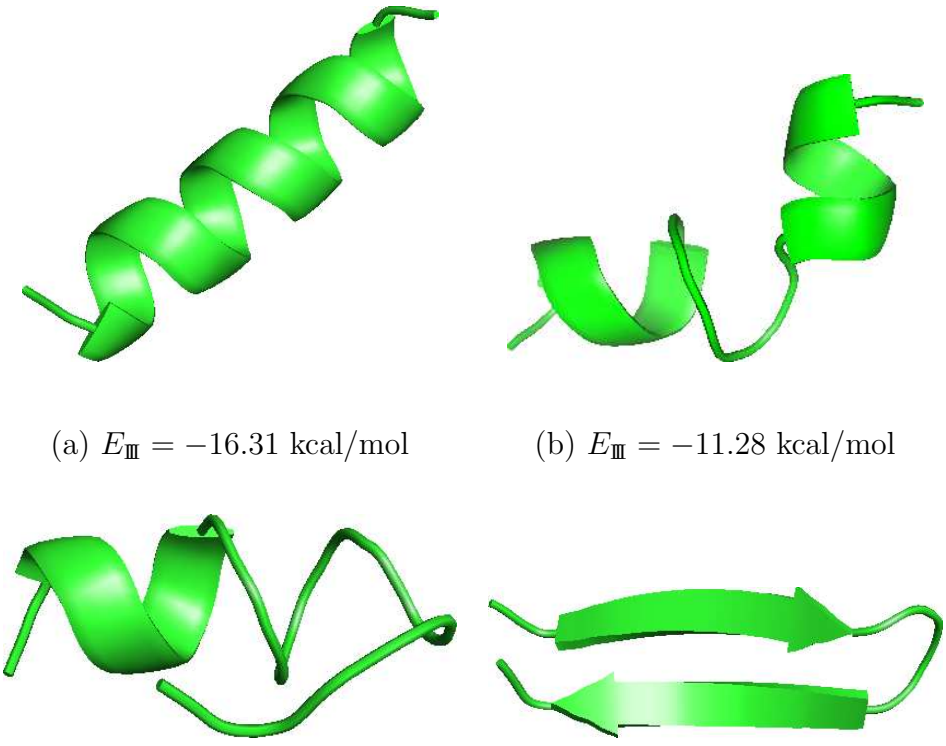
For a beta hairpin *in vacuo*, only half of the possible hydrogen bonding of the backbone can be formed, by comparison with an alpha helix structure of the same peptide. If there is solvent present, the unbonded hydrogen atoms and the oxygen atoms of the backbone can combine with the solvent molecules to form new hydrogen bonds. Hence, generally speaking, a protein/peptide *in vacuo* is not favorable to forming a beta structure, while the solvent environment will provide



(a) $E_{\mathbb{I}} = -9.23$ kcal/mol (b) $E_{\mathbb{I}} = -8.60$ kcal/mol

(c) $E_{\mathbb{I}} = -8.58$ kcal/mol (d) $E_{\mathbb{I}} = -8.29$ kcal/mol

Figure 5.4: The four lowest energy minima of the VGV peptide obtained using the BP method with the ECEPP/2 force field.



(a) $E_{\text{III}} = -16.31$ kcal/mol (b) $E_{\text{III}} = -11.28$ kcal/mol

(c) $E_{\text{III}} = -12.51$ kcal/mol (d) $E_{\text{III}} = -10.16$ kcal/mol

Figure 5.5: The four lowest energy minima of the VGV peptide obtained using the BP method with the ECEPP/3 force field.

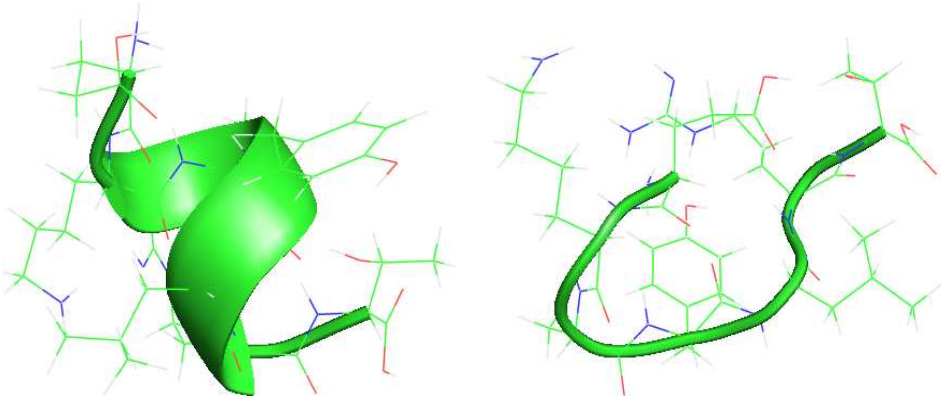
more opportunities. It is difficult to obtain beta structures without the participation of solvent molecules. However, if the energy penalty introduced by the reduction of the hydrogen bonding can be compensated by the relaxation of the backbone dihedral angles and other interactions, the formation of a relatively stable beta hairpin *in vacuo* is possible.

5.4 EKAYLRT peptide

EKAYLRT is a sequence of amino acids that often appears in naturally occurring proteins [161, 162]. It can be involved in both the alpha helix and the beta sheet formation. This peptide may provide a platform for studying the mechanism of structural transitions between the alpha helix and beta sheet structures. For convenience, the peptide will be referred to as Ekay in this section.

The present study is performed with no solvent present. For the Ekay peptide itself, its most stable structure is an alpha helix when using the ECEPP/3 potential, as shown in Fig. 5.6 (a), while just a random coil when using the ECEPP/2 potential, as shown in Fig. 5.6 (b). Their corresponding energies are presented below the figures. The difference shows again that the ECEPP/3 potential prefers alpha helix structures than ECEPP/2.

In general, the formation of a stable beta structure often occurs in the case that the protein has contact interaction with the environment, so that the unbonded atoms can interact with the environmental atoms to lower the free energy of the system. Otherwise, beta structures would not be stable, or even no beta structure can be formed. The VGV artificial peptide just discussed is an exception. The solvent effect and the interactions with other surrounding proteins/peptides are the main sources of environmental interactions. The solvation effect in protein structure formation has attracted a lot of interest in past years, and further studies are still required. Interactions introduced by other environmental contacts are relatively not so widely studied. In studying the prion-like folding pathways, Chen *et al.* [59] introduced a flat hard wall to simulate the "background" structures, which interact with the front structure. Hansmann and co-workers [161, 162] studied the



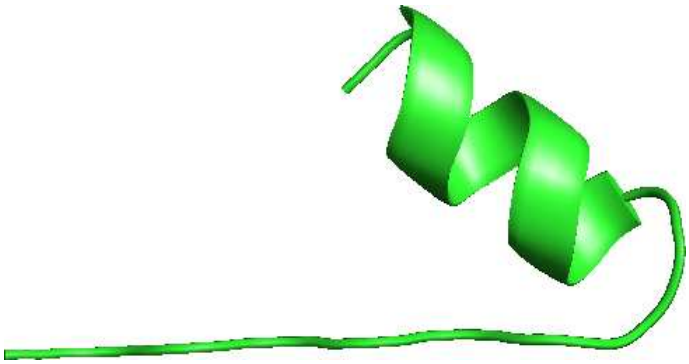
(a) $E_{\text{III}} = -28.05$ kcal/mol (b) $E_{\text{II}} = -24.69$ kcal/mol

Figure 5.6: The lowest energy configurations of the Ekay peptide.

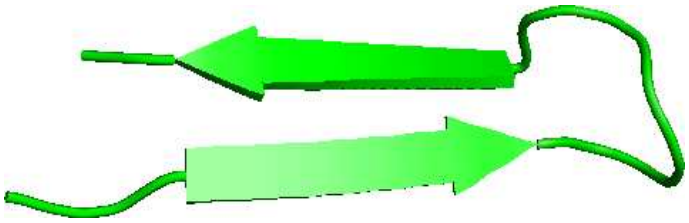
Ekay peptide folding by introducing another Ekay peptide fixed as a beta strand, and connecting it to the C-terminus of the original Ekay peptide by a four-glycine peptide chain. The rigid beta strand introduced was to simulate the environmental interactions.

In this study, we follow Hansmann's approach by introducing an extra Ekay peptide for the background interaction. Hence, the long peptide considered now is NH_2–EKAYLRT–GGGG–EKAYLRT–COOH, denoted as peptide "EGE" below. The N-terminus Ekay peptide will be chosen as the rigid beta strand in this study by fixing its backbone dihedral angles at $\psi = 140\,^\circ$ and $\phi = -140\,^\circ$. The dihedral angle ω for the whole peptide is fixed at $180\,^\circ$ to reduce the dimensionality for fast simulation. Simulations are performed *in vacuo* with both the ECEPP/2 and ECEPP/3 force fields.

Figure 5.7 shows three low energy configurations obtained using the BP method with the ECEPP/3 force field. For the global minimum structure, Fig. 5.7 (a), the free folding part at the C-terminus formed an alpha helix similar to the global minimum structure of the Ekay peptide as shown in Fig. 5.6 (a). However, there are low energy configurations having the structures as beta hairpins. Figure 5.7 (b) has the lowest energy among the beta hairpin structures obtained, while Fig. 5.7 (c) has



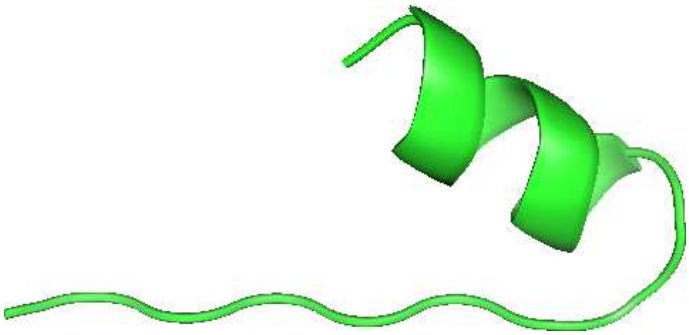
(a) $E_{\text{III}} = -61.99$ kcal/mol



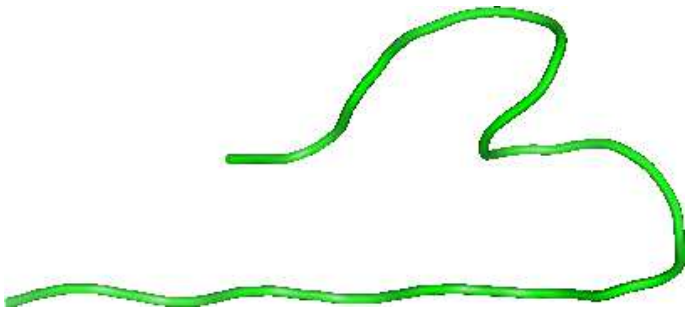
(b) $E_{\text{III}} = -58.77$ kcal/mol

(c) $E_{\text{III}} = -56.96$ kcal/mol

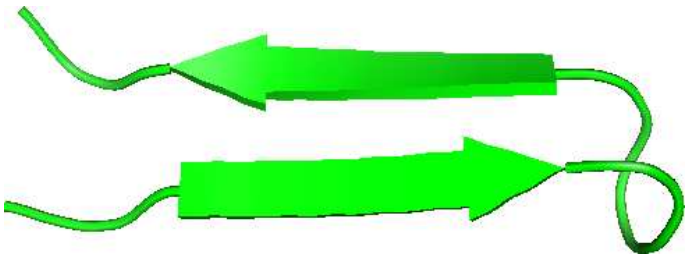
Figure 5.7: The low energy configurations of the "EGE" peptide obtained using the ECEPP/3 potential.



(a) $E_{\text{II}} = -56.73$ kcal/mol



(b) $E_{\text{II}} = -55.42$ kcal/mol



(c) $E_{\text{II}} = -54.89$ kcal/mol

Figure 5.8: The low energy configurations of the "EGE" peptide obtained using the ECEPP/2 potential.

the most residues contribute to the beta strands. Both the alpha and beta structures have high population during the simulation, even though beta conformations are not as energy favorable as the alpha helical structures.

When the ECEPP/2 is used to model the peptide, we obtained both alpha helix and beta hairpin structures at low energy as well. Random coil structures also appear often in the simulation. Figure 5.8 (a) is the lowest energy structure obtained, which contains the helix conformation, and (c) has the lowest energy among all the beta hairpin structures. For the structure shown in Figure 5.8 (b), the freely moving part of the "EGE" peptide condensed to a random coil structure, which has energy even lower than the beta hairpin shape. The low energy conformations contain all the three structures, i.e. alpha, beta and coil, in high population.

From the simulation we can see that the introduction of the rigid beta strand improves the formation of beta conformations. Without the rigid strand, the Ekay peptide can only be stable in the appearance of an alpha helix, under the ECEPP/3 potential, or a random coil, under the ECEPP/2 potential. When the rigid beta strand is introduced as the environment interactions, high populations of beta hairpin structures appear. Further, since there are low energy coil structures when using the ECEPP/2 force field while only alpha helix and beta hairpin structures can have low energies when using the ECEPP/3 force field, the result agrees well with our analysis in the previous applications that ECEPP/3 enhanced the hydrogen bonding interactions when the protein chain coiled together. Hence, the energy landscape under ECEPP/3 will have higher energy barriers and be more rugged than ECEPP/2. Simulations using ECEPP/3 may require more effort to escape local entrapment.

5.5 Summary

In this chapter, the MUBH and BP methods, together with the BH method, are applied to study the protein folding problems. For the polyalanine systems, we have been successful in locating the helical structures. The $A_{10}G_5A_{10}$ peptide has different structures with the ECEPP/2 and ECEPP/3 force fields. Another artificial peptide, the VGV peptide, shows a beta hairpin structure as the lowest energy

conformation when using the ECEPP/2 force field, while an alpha helix structure when using the ECEPP/3 force field. Further, for the "EGE" peptide with its N-terminus fixed as a beta strand, lower energy configurations obtained using the ECEPP/2 potential have high population appearing as coils, while those obtained using the ECEPP/3 potential can only take the form of either the helix or the beta hairpin structures. All the studies suggest that the ECEPP/3 potential, as an improvement of the ECEPP/2 potential, enhanced the hydrogen bonding interactions, which dominates the formation of the alpha helix and the beta sheet structures. However, present studies imply that the enhancement might be too much. Simulations might be biased towards the helical structures when using the ECEPP/3 force field. Furthermore, we obtained the minimum structure of the miniprotein Trp-cage, which is one of the fastest folding proteins known. The lowest energy configuration obtained has the RMSD of 2.24 Å when compared with the NMR structure with PDB code 1L2Y. The agreement of our simulated results with previous studies proves that the MUBH and BP methods we recently proposed are effective in studying protein folding problems.

Chapter 6

Conclusions and Future Work

6.1 Review and Conclusions

In this thesis, we have focused mainly on the development of global optimization methods and their applications to atomic cluster crystallization and protein folding problems. The two global optimization methods presented in this thesis are the multicanonical basin hopping (MUBH) method and the basin paving (BP) method.

The MUBH method, described in Chapter 2, is a combination of the multicanonical Monte Carlo (MUCA) method and the basin hopping (BH) method. During a simulation, the local minimum of a system configuration just reached is located precisely by a local minimization procedure at each step, as in the BH method. Acceptance of the present local minimum configuration is determined by comparison of its multicanonical weight with that of the last local minimum. In other words, the MUBH method is equivalent to a multicanonical method applied on a reduced energy surface that contains only plateaus of the local minima of the original energy surface. For the procedure of determining the multicanonical weight, Berg's iteration scheme [17, 18, 19] was employed in order to obtain more stable simulation results.

To ascertain its efficiency, MUBH was applied to the benchmark system of Lennard-Jones clusters. When the system is small, $N < 150$, MUBH shows no

obvious improvement in efficiency in comparison with the BH method. This is because the reduced energy surface of a small system is relatively simple, and the BH method could locate the global energy minimum before the MUBH method is able to obtain a stable multicanonical weight. When the system size is increased to $N > 150$, the MUBH method begins to show its effectiveness. The improvement in efficiency of MUBH over BH is dramatic when we compare the average number of MC steps taken to attain the global minimum. The improvement comes from the intrinsic capability that MUBH inherited from the MUCA method. When "hopping" between the plateaus of the reduced energy surface, the simulation procedure may still meet high energy barriers that BH fails to overcome, since it uses the Boltzmann weighting scheme. By contrast, a MUBH simulation will not be entrapped in a local minimum because of the multicanonical weight it adopts. Further, the local minimization procedure in each MC step is able to locate the precise value of the local minimum and thereby avoid thermal fluctuations, which solves the difficulty that MUCA will meet when used as the global optimization schedule. In the applications, the influence of the initial temperature was also tested. The results indicate that a proper initial temperature setting is important to the performance of the MUBH method.

Beyond the improvement of computational algorithms, we can also take advantage of distributed computing by carrying out parallel calculations on a single problem to save the simulation time. The MUBH method was further parallelized to the asynchronous multicanonical basin hopping method (AMUBH) in Chapter 3 using the message passing interface (MPI) for the study of large systems. In the implementation, a message is sent out by the main node to ask for the histograms collected at all the involved nodes. After receiving the required data, the new multicanonical weight is calculated and sent back to the slave nodes. After answering the main node's request by sending out their collected data, the slave nodes will continue to run using the old weight until the new weight is received for the next iteration. The parallelization improves the multicanonical weight updating speed. Further, having different nodes work on different regions of the configuration space dramatically improves the chances of finding the global minimum.

BH, MUBH and AMUBH methods were employed together to determine the

crystalline structure of Cobalt nanoclusters with system size N from 2 to 200. For small size clusters, $N \leq 150$, the BH method was applied to calculate the global minima. For $150 < N \leq 180$, the MUBH method was utilized to improve the sampling efficiency. When the system becomes even larger, $180 < N \leq 200$, both BH and MUBH would take too long to locate the global minima; hence AMUBH was applied to save computational time by using multiple CPUs for a single run. The global minima obtained agrees well with the experimental results.

Based on the similarity of the binary potentials of different clusters, a *structure mapping* method was also proposed in Chapter 3 for rapid determination of the unknown minimum structure of a cluster from the existing database of cluster configurations. By mapping the global minimum configurations of the Lennard-Jones clusters to the Cobalt nanocluster systems for $N < 150$, we were able to find two global minimum structures, $N = 103$ and 104, that the *ab initio* calculations failed to locate. If a configuration database can be setup, the structure mapping approach will provide a useful method for rapid determination of the global minimum.

Another global optimization approach, described in Chapter 4 of this thesis, is the basin paving (BP) method, which comes from the combination of the basin hopping method and the core idea of the energy landscape paving (ELP) method. As in the BH and MUBH methods, the Monte Carlo (MC) simulation is performed on the reduced energy landscape, which is obtained by performing a local minimization procedure at each MC step. The acceptance of a new state is governed by the weight based on a function of the histogram accumulated. The histogram can be updated after every MC step. By choosing a suitable functional form of the histograms, the simulation will be able to steer away from the state already visited, by keeping a "memory", to less frequently visited states so as to avoid local entrapment problems. The simulation procedure will still be able to go back to the state already visited once the old "memories" are mixed with the new "memories". If a new state has a lower energy than the present state, the new state will be surely accepted to avoid missing any low energy configurations. The bias towards the lower energy configuration is the main difference between the BP method and the ELP method in the acceptance determination procedure. This bias enhances sampling of the low energy regions and avoids the sampling of the unphysically high

energy regions, while still retaining the ability to move out of any local entrapment due to the weighting scheme based on the accumulated histograms.

When the BP method was applied to locate the global minimum structure of the pentapeptide Met-enkephalin, a new configuration with lower energy than the previous reported result was found when the ECEPP/3 force field was employed with the peptide dihedral angle ω fixed at 180°. The simulation can also reproduce all the other local minimum structures of the peptide under different conditions. When further applied to the villin subdomain HP-36 protein *in vacuo*, BP was able to locate a structure having an appearance similar to the NMR structure. A configuration having an energy lower than the previously reported minimum was also located.

Several protein/peptide systems based on the ECEPP/2 and ECEPP/3 force fields were studied in Chapter 5. Small size polyalanines and the Ekay peptide can form helix structures as lowest energy configurations when the ECEPP/3 potential is used, while only a random coil structure appears when the ECEPP/2 potential is employed. The lowest energy configuration of the $A_{10}G_5A_{10}$ peptide has the structure of a U-turn-like shape formed by two helices connected by a turn when using the ECEPP/2 force field. When ECEPP/3 is used, the corresponding structure is simply a long helix. Further, the VGV peptide can form a beta-hairpin as the global minimum configuration when the ECEPP/2 potential is employed, but prefers helical structures when the ECEPP/3 potential is employed. The difference between the low energy configurations derives from the fact that the ECEPP/3 potential uses a different parameter set than that associated with the ECEPP/2 potential. The change of the partial charges carried on by the backbone atoms is the main source causing the difference, resulting in the bias towards the helical structure when the ECEPP/3 is employed in simulations.

Environmental interactions can affect the folding mechanics of a protein. The artificial peptide "EGE" has been studied in this thesis by fixing its N-terminus Ekay peptide to a beta strand to mimic the "background" proteins. For both the ECEPP/2 and ECEPP/3 potentials used in the simulation, the lowest energy structures of "EGE" are all combination of a beta strand, which is the fixed part, and an alpha helix, which is the freely moving part. However, beta hairpin structures

are formed in both cases with high populations, even though a single Ekay peptide cannot have a stable beta strand structure. The "background" interactions help the forming of beta structures.

The lowest energy structure of the fastest folding miniprotein Trp-cage was also obtained using the ECEPP/3 potential in Chapter 5. Compared with the NMR structure with PDB code 1L2Y, its RMSD is $R_{\mathrm{rmsd}} = 2.24$ Å. The simulated result was able to reproduce the first helix of the N-terminus and the polyproline stretch of the C-terminus successfully as in the NMR structure. The residues that connect them together provides the main structure difference.

From the studies of cluster crystallization and protein folding problems, both the recently proposed MUBH and BP methods have been shown to be efficient and effective. They will be useful tools for studying global optimization problems numerically.

6.2 Future Work

In following up the present work, there are several future directions worth investigating:

1. When using the structure mapping method for rapid determination of the cluster structures with binary interactions between particles, a database that contains known structures as completely as possible will be required. Without a good database, the structure mapping method may not be able to locate the lowest energy structure of a cluster. Collecting all the known structures and constructing a database based of them is one of the projects to be carried out in order to make the structure mapping method a truly powerful tool.

2. The Lennard-Jones and Cobalt cluster systems studied in this work contain only one kind of atom. Systems containing two or more kinds of atoms, such as the nano-alloys, which are bimetallic nanoclusters, have a large variety of potential applications. Such systems can certainly be studied using the global optimization algorithms described in this thesis.

3. At the present stage, the protein folding simulations performed have all based on the ECEPP/2 and ECEPP/3 force fields. Even though they are effective most of the time, we cannot ignore that fact that they were built more than 10 years ago (for ECEPP/2, it is more than 20 years). New experimental results and computational results are not included, which makes them inaccurate in some cases. The use of new force fields, such as Amber and CHARMM, for future studies should be considered.

4. All-atom force fields can give the best description of the system studied. However, the computational time it takes to calculate the interactions with all atoms included is by no means an easy task. For some systems, a coarse-grained model can also provide a good approximation. In future studies on relatively large systems, a united-atom or bead model may also be considered.

5. In this work, all the protein foldings were performed *in vacuo*. Environmental interactions play important roles in forming a stable structure. Solvents will be introduced in future studies of protein folding simulations.

Appendix A

Protein Folding

Proteins are one of the most important and common macromolecules that make up the primary constituents of life. They are essential to both the structure and the function of cells, the building blocks of living systems. From a chemical point of view, proteins are unbranched necklace chains joined by some or all of the twenty naturally occurring amino acids in their typical sequence that can fold into unique three dimensional (3D) structures. These amino acids are joined by the peptide bonds between them. The amino acid linear sequence of a protein is always referred to as the primary structure. It reveals no direct information about how the amino acids are arranged spatially. Different regions of the sequence will form a locally defined secondary structure as an alpha (α) helix or a beta (β) strand, which is highly patterned, or a coil, which assumes no stable shape. The tertiary structure refers to the 3D conformation of the entire polypeptide chain or a domain of it. The final protein complex may contain more than one peptide chain arranged in 3D to form a quaternary structure.

Most proteins can perform their specific functions only when they fold into typical tertiary structures. A variety of diseases, such as the prion diseases, and many cancers, are caused by the misfolding of proteins [59, 60]. The function of a protein relies on its tertiary structure, which in turn is determined by the primary structure. Hence the protein folding problem, which predicts the 3D configuration and the corresponding function of a protein solely from the knowledge of its amino

acid sequence, is critical in biochemical studies. Not surprisingly, the importance of the protein folding problem has raised considerable interest. Many promising studies have been carried out over the past years, even though there is not a complete solution yet. While it is relatively easy to determine the amino acid sequence experimentally, it is a big challenge to discover the spatial structure of a protein by X-ray crystallography or nuclear magnetic resonance (NMR) techniques. Furthermore, the kinetic, dynamical and stochastic properties of a folding procedure need to be studied as well. Computer simulation can often provide a feasible way, sometime the only way, to study the folding mechanism by adopting some suitable approximations.

There are mainly two directions to theoretically determine the 3D structure[1] and study the physical characteristics of a protein with a typical amino acid sequence from "first principles": the molecular dynamics (MD) method and the Monte Carlo (MC) method. The MD method is based on Newton's second law to solve the equations of motion for each atom in a protein. If the system configuration and the speed of each atom are known at time t, and the relevant forces that are exerted on each atom can also be obtained, the movement of atoms after a small time interval Δt will be determined by solving Newton's equations of motion numerically. The forces exerted on each atom at time $t + \Delta t$ can also be computed. This procedure is repeated iteratively, and the simulated trajectory should be able to reproduce the real one if the time interval is chosen small enough. MD is a good choice for investigating the kinetic properties of the folding problem. However, because of the small time step and the limited simulation time, MD is generally not able to sample a wide enough configuration space for calculating the thermodynamics of the folding procedure. The Monte Carlo (MC) method, on the other hand, can adopt moving schemes that allow larger step size, for instance the dihedral angle moving scheme, to make a substantially bigger change of the system conformation, so that the simulation can sample wider configuration space. MC allows bigger configuration changes because it considers only the relative weight between the old and new states, $w(\mathbf{r}_{\text{new}})/w(\mathbf{r}_{\text{old}})$. From this perspective, the MC method is more

[1]Unless stated explicitly, the words structure, conformation and configuration in this part refer to the secondary or tertiary but not the primary structure.

efficient than the MD method.

According to the thermodynamic hypothesis [66, 67], the native state, which is the operative or functional structure of a protein, lies at the global minimum of the free energy, which can be further approximated by the global energy minimum. Once the global minimum structure has been determined, the native structure of a protein will be obtained and its functionality can then be studied, which is especially useful in drug design and the study of diseases. Both MD and MC approaches can be employed to search for the global minimum, even though only MC methods have been employed in this thesis.

A.1 Protein — Polypeptide Chain

We already know that a protein is one or several unbranched sequences of amino acid chains joined together by peptide bonds. All the twenty amino acids have in common a central carbon atom, conventionally labeled as C_α, to which is attached a hydrogen atom H, a carboxyl group COOH and an amino group NH_2. What distinguishes one amino acid from another is the side chain attached to the central C_α atom. Twenty different side chains specify twenty amino acids. They are each assigned a genetic code, which can be represented by one letter or three letters. Appendix B lists the structures of the twenty amino acids and their corresponding genetic codes.

Amino acids are joined end-to-end by the formation of peptide bonds. By dropping the OH from the carboxyl group of the first amino acid, and the atom H from the amino group from the second one, a bond is formed between them. This process is repeated to form a long chain. The peptide unit, which starts from the C_α atom of the first amino acid and ends at the C_α atom of the next one with the CO and NH groups part of it, is quite rigid. The peptide dihedral angles rotating around the C—N bond, often denoted as ω, can only take values very close to $180\,^\circ$. Hence, in computer simulations, they are often fixed at $180\,^\circ$ to reduce the degrees of freedom. Only the rotations around the bond formed by atom C_α and the carbon atom of the carboxyl group, labeled as C, and the bond formed by atom

C_α and the nitrogen atom of the amino group, labeled as N, of the backbone are left free. Conventionally, the dihedral angles of rotation around the C_α—C bonds are denoted by ψ while the angles of rotation around N—C_α are denoted by ϕ. The peptide chain folded together will generate a 3D structure such as alpha helices, beta strands, random coils or their combinations connected by turns.

A.2 Protein Models

To study the properties of a protein numerically, one must first set up a model that reveals the interactions between the atoms, the components and the environment. The ideal case is to solve the many-body Schrödinger equation for the potential energy function that describes the system. However, it is impossible to solve the equation for relatively large systems. Hence, simplified models are always employed in protein folding studies.

Probably the simplest model that has played important role in theoretical studies of protein folding is the hydrophobic-polar (HP) lattice model [163, 164, 165], which was first proposed in 1985 by Dill. The HP model considers of only two kinds of amino acids, which are either hydrophobic or polar (also referred to as hydrophilic) monomers, labeled as H or P, respectively. The amino acid sequence of a protein is abstracted as a binary sequence of monomers connected by a string, which can be colored with either H or P. They can only occupy the vertices of a two or three dimensional square lattice. One vertex allows the occupation of one monomer or none at all; and the adjacent amino acids in a real protein will occupy adjacent vertices too. The quality of a folding is scored by the number of hydrophobic monomer pairs that are adjacent in lattice but not adjacent in the amino acid sequence of the protein. Hence, the effective energy of a HP model can be defined as [166]

$$E = \sum_{i<j} E_{\sigma_i \sigma_j} \Delta_{ij} \ , \tag{A.1}$$

where σ_i, σ_j stand for the monomer type, which can only be H or P. Hence we have three kinds of interactions with the contact energy $E_{\mathrm{HH}} = 1$, $E_{\mathrm{HP}} = 0$ and $E_{\mathrm{PP}} = 0$. Δ_{ij} is the contact matrix element whose value depends on the relative position of

monomer i and j. If i and j occupy two nearest neighbor vertices on the lattice but not adjacent along the monomer chain (the protein), $\Delta_{ij} = 1$, otherwise, $\Delta_{ij} = 0$.

In spite of the simplicity of the HP model, the folding process illustrated by the model has behavioral similarities with that of a real protein. There have been many successful applications of the model to predict the native conformation of proteins [167, 168, 169, 170]. The discrete property of the configurations makes it easier to sample wider conformation space, hence the main features of the protein folding problem can be better captured than the relatively insufficient sampling of the configuration space of more complex models.

Other than the HP model, there are some coarse-grained models specially designed to study the folding dynamics [171], with the Gō-like model as a representative. The Gō model [89] was first introduced to study protein folding, unfolding and fluctuations *in silico* by representing the protein as a chain of one-bead amino acids, whose structure was intentionally biased towards the native structure of the proteins studied. This was realized by adjusting the repulsive and attractive parameters for non-bonded beads so that the native-like configuration could have the lowest energy. This extreme simplified representation dramatically smoothed the energy landscape to a weakly rugged funnel pointing towards the native structure, while still keeping the ability of reproducing some thermodynamic and kinetic properties of the protein. The model has been made more sophisticated recently, so that it can be used to study the meta-stable states in the folding process, which cannot be described by the original extremely simple implementation due to the relatively smooth surface and hence the loss of some intermediate states. When more energy terms are considered, new intermediate states will appear with the funnel getting more rugged. Clementi *et al.* introduced an all atom Gō model [172] which was employed to simulate the folding of protein L and protein G. Their results showed that not only the overall folding mechanism of the proteins was qualitatively recovered, but also that the roles of some specific residues were qualitatively and correctly reproduced [172]. The solvation effect can also be considered in building the Gō model as Kaya and Chan [173] and Cheung *et al.* [174] did. The inclusion of the solvent interactions with the chain of beads (protein) resulted in the appearance of partially desolvanted-partially folded intermediate states in Ref. [173] and

a near-native intermediate with a partially solvated hydrophobic core in Ref. [174].

Unlike the Gō-like models which are specifically designed with bias towards protein native configurations, coarse-grained models for general folding dynamics have been developed as well. These models are not designed for studying a specific protein but for as many proteins as possible. Hopefully, one set of well designed parameters will be able to reproduce most of the kinetic and dynamic properties of different proteins. Even though there have been applications of one-bead models [175, 176], which adopts a single bead to represent an amino acid, using more beads in the representation is generally required to take into account the generic effect of the size, type, geometry and conformation of an amino acid. Introducing more beads in a model means more fitting parameters can be adopted and more general characteristics of proteins and amino acids can be studied. Hence a closer representation of the interactions between the beads to those between the amino acids of a practical protein will be capable of being realized.

Two-bead models have been employed to study general dynamics [177], structure prediction [178, 179] and peptide binding [180]. In these models, the C_α atom of each residue is conveniently chosen as one bead. The second bead is used to represent the side chain group, whose position is often set on the centroid of the side chain where the most distinctive interactions are subject to it [177], so that the side chain-side chain interactions and side chain-backbone interactions can be included in a simulation. Even though there is an extra "peptide" bead located midway between two C_α atoms in the united-residue (UNRES) force field proposed by Scheraga and co-workers [181, 182, 178, 179], the position of this bead is solely determined by the C_α atoms without introducing any extra degree of freedom, and it can be classified as a two-bead model [171].

For each residue, if all the heavy atoms of the backbone, which are C_α, N and the CO group, are represented by three beads explicitly, and the side chain group is represented by one bead again, a four-bead model will be constructed for protein folding studies once a proper set of parameters has been determined [183, 184, 185, 186]. With the CO group of the backbone further represented as two beads and the atom H bonded to the atom N of the backbone counted in as one bead as well, one will have a six-bead model [187, 188, 189]. The structures predicted using

these models match very well with the corresponding native structures determined experimentally [183, 184, 185, 186, 187, 188, 189, 190, 191].

Despite the success of these reduced models, they lack precision when the details of structural transitions are required. Further, the parameters they employ are often derived from a typical group of proteins, so that they are only reliable when the new protein shares the general properties of the protein groups that have provided the information for fitting the parameters. To predict a more reliable structure, obtain more detailed information about state transition, and distinguish the configurations close to the native state, a more general description, an all-atom model, of the proteins is required. Actually, some all-atom models [85, 68] appeared even earlier than many of the coarse grained models. The interactions between atoms are described by an empirical potential, commonly referred to as a force field, with parameters obtained from experiment and *ab initio* calculations of small systems by solving the many-body Schrödinger equation. With the appearance of new experimental and numerical results, the parameter set of a force field will be updated so that it can approximate the most general situations as best as it can.

A.3 Empirical Force Fields

The development and application of the empirical force field lies on the validity of several assumptions [64]. The Born-Oppenheimer approximation is probably the first and the most important one, which assumes that the electrons need no reaction time to follow the movement of the nuclei based on the fact that the nuclear mass is far greater than the electronic mass. Without this approximation, it is impossible to express the energy as a function of the nuclear coordinates. Transferability is another key issue for the application of a force field. It assumes that the force field parameters obtained from small molecules, and tested on a limited number of cases, can be applied to study much larger molecules and a much wider range of systems. Moreover, the system potential is interpreted as simple models of interactions, which can be further approximated with rather simple functions that describe the main contributions.

Most of the force fields widely used these days can be described in terms of four components: the bond stretching, angle bending, torsional, and non-bonded interactions. Several sophisticated force fields may contain more terms in addition to these four terms. The non-bonded interaction often includes the electrostatic, the van der Waals and/or the hydrogen bonding interactions. For the first three terms, they can be expressed conveniently as functions of bond lengths, bond angles and the rotation of bonds, respectively, while the non-bonded interaction part will be functions of the relative distance between atoms. Energy penalties are associated with conformational deviations from their equilibrium positions.

One commonly used functional form of the protein force field incorporates a relatively simple potential energy function [192]

$$\begin{aligned} V(\mathbf{r}) = & \sum_{\text{bonds}} k_b(b-b_0)^2 \\ & + \sum_{\text{angles}} k_\theta(\theta-\theta_0)^2 \\ & + \sum_{\substack{\text{dihedral}\\ \text{angles}}} \frac{V_n}{2}(1+\cos(n\chi-\gamma)) \\ & + \sum_{\substack{\text{non-bonded}\\ \text{pairs } (i,j)}} \left[\frac{q_i q_j}{r_{ij}} + \frac{A_{ij}}{r_{ij}^{12}} - \frac{C_{ij}}{r_{ij}^{6}}\right] \end{aligned} \tag{A.2}$$

in which the first three terms are summations over bond stretches (1–2 interaction), bond angles (1–3 interaction) and dihedral angles (1–4 interaction, the improper dihedral angles can also be included in). The fourth term includes all non-bonded interactions (all pairs but not the 1–2, 1–3 and 1–4 interactions), and sometime the 1-4 interactions with rescale factors. q_i and q_j are the partial charges on atoms i and j, respectively. The van der Waals interaction is represented by the Lennard-Jones 6-12 potential.

There are many different implementations of force fields, and they have proven to be successful in different applications. Of particular importance among them are the ECEPP potentials proposed by Scheraga's group [85, 68, 69, 70], which has provided an invaluable starting point for the development of force fields, while the most popular force fields used for organic and biomolecular systems are probably

the Amber [77, 78, 79, 80, 81, 82], CHARMM [83, 84] and OPLS [86, 87, 88] force fields.

In this thesis, several protein folding simulations have been performed using the ECEPP/2 [68, 69] and ECEPP/3 [70] force fields. The most obvious difference between the ECEPP force fields and the other force fields (Amber, CHARMM, OPLS and most others), is that the bond stretchings and bond angles in the ECEPP force fields are fixed at experimental values due to the fact that they are relatively rigid in protein conformations. Both ECEPP force fields contain the electrostatic term, the Lennard-Jones interaction term, and the hydrogen-bond term between pairs of atoms, together with the torsion term for all torsional angles. Their potential function can be expressed as:

$$E_{tot} = E_{ES} + E_{LJ} + E_{HB} + E_{tor}\,, \tag{A.3}$$

with

$$E_{ES} = \sum_{(i,j)} \frac{332\, q_i q_j}{\epsilon\, r_{ij}} \tag{A.4}$$

$$E_{LJ} = \sum_{(i,j)} \left(\frac{A_{ij}}{r_{ij}^{12}} - \frac{B_{ij}}{r_{ij}^{6}} \right) \tag{A.5}$$

$$E_{HB} = \sum_{(i,j)} \left(\frac{C_{ij}}{r_{ij}^{12}} - \frac{D_{ij}}{r_{ij}^{10}} \right) \tag{A.6}$$

$$E_{tor} = \sum_{l} U_l (1 \pm \cos(n_l \chi_l))\,, \tag{A.7}$$

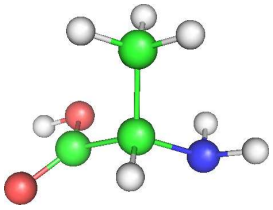
in which r_{ij} is the distance in Å between the atoms i and j. χ_l is the torsion angle with its dihedral multiplicity n_l for the chemical bond l. q_i and q_j, in units of electronic charges, are the partial charges on atoms i and j, and ϵ is the dielectric constant of the environment. The value $\epsilon = 2$ will be used in our simulations, which corresponds to the space inside the protein molecules for the case of study without solvent. The factor 332 in Eq. (A.4) is a constant used to express the electrostatic energy in kcal/mol. The parameters A_{ij}, B_{ij}, C_{ij}, D_{ij}, and U_l are calculated from the crystal structures of amino acids. As the bond lengths and bond angles are fixed, and no out-of-plane deformation of the peptide bonds is allowed,

the only independent variables remaining are the backbone dihedral angles ϕ and ψ, the peptide dihedral angles ω and the side chain dihedral angles χ. The main difference between ECEPP/2 and ECEPP/3 is that they use different parameter sets. ECEPP/3 uses a revised parameter set based on more recent experimental findings and the recalculation of partial charges.

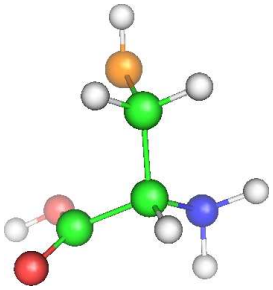
Appendix B

Twenty Amino Acids in Proteins

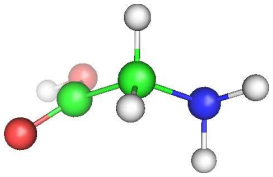
There are 20 naturally occurring amino acids in total that can be used to build proteins. They have in common a carboxyl group COOH, an amino group NH_2, a central carbon atom C_α with a hydrogen atom H attached to it, and a side chain to distinguish different amino acids. An exception is the amino acid proline, which shares the backbone atom C_α and N as part of a ring. The structures of the 20 amino acids in ball-and-stick model and their genetic codes in both three-letter and single-letter format are listed in the following figures. They have all been drawn with the backbone atoms located at the bottom of each figure. Balls in different colors stand for different atoms: green → Carbon (C), blue → Nitrogen (N), grey → Hydrogen (H), red → Oxygen (O), and orange → Sulphur (S).



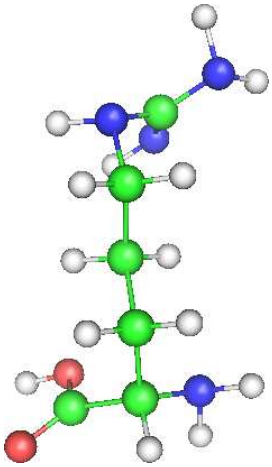
Alanine
(ALA/A)



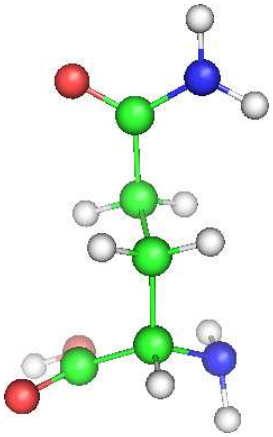
Cysteine
(CYS/C)



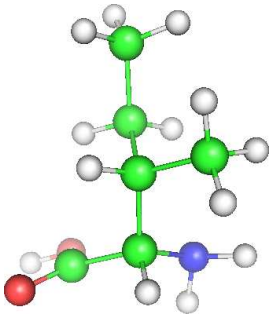
Glycine
(GLY/G)



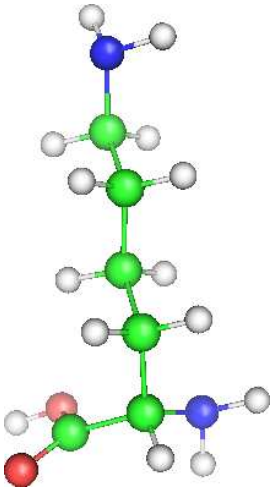
Arginine
(ARG/R)



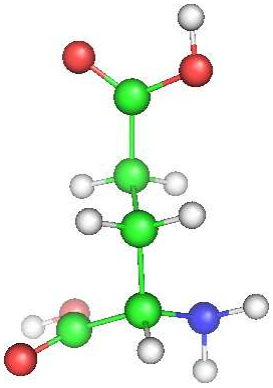
Glutamine
(GLN/Q)



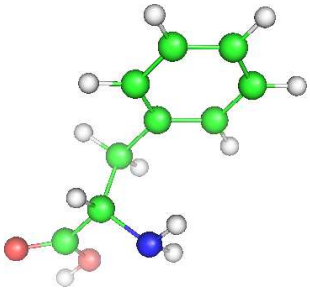
Isoleucine
(ILE/I)



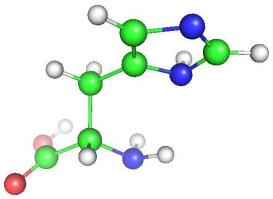
Lysine
(LYS/K)



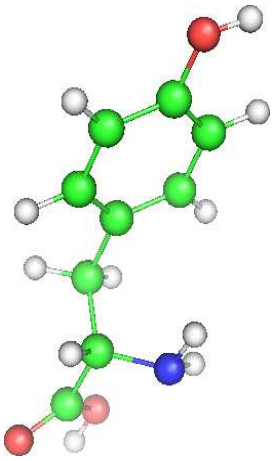
Glutamic acid
(GLU/E)



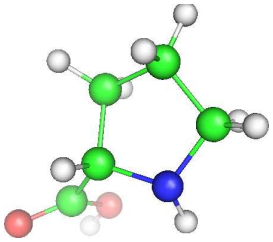
Phenylalanine
(PHE/F)



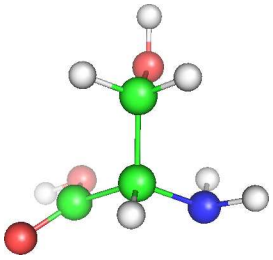
Histidine
(HIS/H)



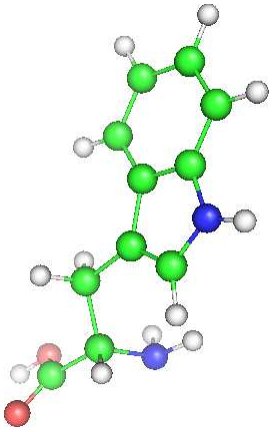
Tyrosine
(TYR/Y)



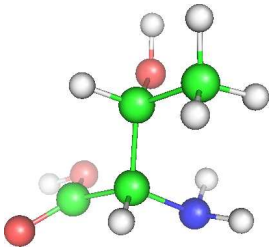
Proline
(PRO/P)



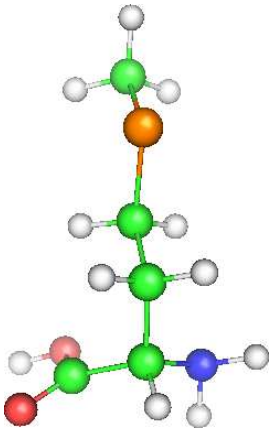
Serine
(SER/S)



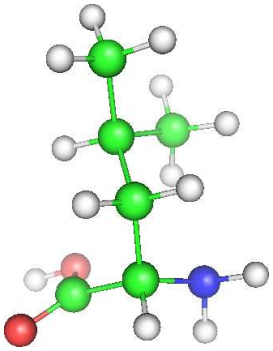
Tryptophan
(TRP/W)



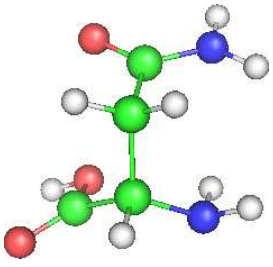
Threonine
(THR/T)



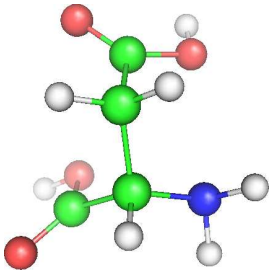
Methionini
(MET/M)



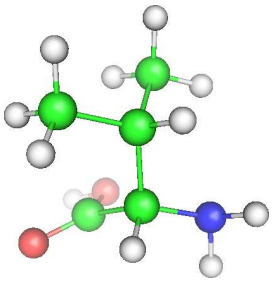
Leucine
(LEU/L)



Asparagine
(ASN/N)



Aspartic acid
(ASP/D)



Valine
(VAL/V)

Appendix C

Acronyms Used in Thesis

2D two dimensional

3D three dimensional

Amber Assisted Model Building with Energy Refinement

AMUBH Asynchronous MUlticanonical Basin Hopping

BH Basin Hopping

BHM Broad Histogram Method

BP Basin Paving

CHARMM Chemistry at HARvard Molecular Mechanics

CPU Central Processing Unit

DoS Density of States

DFT Discrete Fourier Transform

Ekay EKAYLRT

ECEPP Empirical Confirmational Energy Program for Peptides

EGE NH_2-EKAYLRT-GGGG-EKAYLRT-COOH

ELP Energy Landscape Paving

ES Entropic Sampling

FHM Flat Histogram Method

FFT Fast Fourier Transform

hcp Hexagonal-Close-Packed

HMC Hybrid Monte Carlo

HP Hydrophobic-Polar

LJ Lennard-Jones

MC Monte Carlo

MCM Monte Carlo Minimization

MD Molecular Dynamics

MPI Message Passing Interface

MUBH MUlticanonical Basin Hopping

MUCA MUltiCAnonical Monte Carlo

MUCAREM MUltiCAnonical Replica Exchange Method

NMR Nulcear Magnetic Resonance

OPLS Optimized Potentials for Liquid Simulations

PDB Protein Data Bank

PES Potential Energy Surface

PTRS Parallel Tempering Reference Structure

PVM Parallel Virtual Machine

REM Replica Exchange Method

REMUCA Replica Exchange MUltiCAnonical

REST Replica Exchange Simulated Tempering

RMSD Root-Mean-Square Deviation

SA Simulated Annealing

SMMP Simple Molecular Mechanics for Proteins

STM Scanning Tunneling Microscopy

UNRES UNited-RESidue

WLA Wang-Landau Algorithm

Bibliography

[1] D. J. Wales and H. A. Scheraga, Science **285**, 1368 (1999).

[2] D. J. Wales and J. P. K. Doye, J. Phys. Chem. A **101**, 5111 (1997).

[3] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi, Science **220**, 671 (1983).

[4] D. Baker, Nature **405**, 39 (2000).

[5] D. Baker and A. Sali, Science **294**, 93 (2001).

[6] L. Zhan, B. Piwowar, W.-K. Liu, P. J. Hsu, S. K. Lai, and J. Z. Y. Chen, J. Chem. Phys. **120**, 5536 (2004).

[7] L. Zhan, J. Z. Y. Chen, W.-K. Liu, and S. K. Lai, J. Chem. Phys. **122**, 244707 (2005).

[8] P. Matti and J. Niittymaki, editors, *Mathematical Methods on Optimization in Transportation Systems*, Klumer Academic Publishers, 2001.

[9] A. R. Leach, Rev. Comput. Chem. **2**, 1 (1991).

[10] H. A. Scheraga, Rev. Comput. Chem. **3**, 73 (1992).

[11] R. V. Pappu, R. K. Hart, and J. W. Ponder, J. Phys. Chem. B **102**, 9725 (1998).

[12] N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, and E. Teller, J. Chem. Phys. **21**, 1087 (1953).

[13] G. M. Torrie and J. P. Valleau, J. Comput. Phys. **23**, 187 (1977).

[14] B. A. Berg and T. Neuhaus, Phys. Lett. B **267**, 249 (1991).

[15] B. A. Berg and T. Neuhaus, Phys. Rev. Lett. **68**, 9 (1992).

[16] B. A. Berg, Comput. Phys. Commun. **153**, 397 (2003).

[17] B. A. Berg, Fields Inst. Commun. **26**, 1 (2000).

[18] B. A. Berg, Nucl. Phys. B (Proc. Suppl.) **63A-C**, 982 (1998).

[19] F. Yasar, T. Celik, B. A. Berg, and H. Meirovitch, J. Comput. Chem. **21**, 1251 (2000).

[20] U. H. E. Hansmann and Y. Okamoto, J. Comput. Chem. **14**, 1333 (1993).

[21] Y. Okamoto and U. H. E. Hansmann, J. Phys. Chem. **99**, 11276 (1995).

[22] J. Lee, Phys. Rev. Lett. **71**, 211, (erratum) 2353 (1993).

[23] M.-H. Hao and H. A. Scheraga, J. Phys. Chem. **98**, 4940 (1994).

[24] Z. Li and H. A. Scheraga, Proc. Natl. Acad. Sci. USA **84**, 6611 (1987).

[25] R. H. Leary and J. P. K. Doye, Phys. Rev. E **60**, R6320 (1999).

[26] S. K. Lai, P. J. Hsu, K. L. Wu, W. K. Liu, and M. Iwamatsu, J. Chem. Phys. **117**, 10715 (2002).

[27] J. P. K. Doye, Phys. Rev. B **68**, 195418 (2003).

[28] H. Xu and B. J. Berne, J. Chem. Phys. **110**, 10299 (1999).

[29] H. Xu and B. J. Berne, J. Chem. Phys. **112**, 2701 (2000).

[30] P. N. Mortenson and D. J. Wales, J. Chem. Phys. **114**, 6443 (2001).

[31] D. J. Adams, J. Comput. Phys. **75**, 138 (1988).

[32] A. Mitsutake, Y. Sugita, and Y. Okamoto, Biopolymers **60**, 96 (2001).

[33] Y. Iba, Int. J. Mod. Phys. C **12**, 623 (2001).

[34] E. Marinari, G. Parisi, and J. J. Ruiz-Lorenzo, *Spin Glasses and Random Fields*, edited by A. P. Young (World Scientific, Singpore, 1998), p. 59.

[35] F. Glover, ORSA J. Comput. **1**, 190 (1989).

[36] F. Glover, ORSA J. Comput. **2**, 4 (1990).

[37] D. Cvijovic and J. Klinowski, Science **267**, 664 (1995).

[38] G. Besold, J. Risbo, and O. G. Mouritsen, Comput. Mater. Sci. **15**, 311 (1999).

[39] W. Wenzel and K. Hamacher, Phys. Rev. Lett. **82**, 3003 (1999).

[40] U. H. E. Hansmann and L. T. Wille, Phys. Rev. Lett. **88**, 068105 (2002).

[41] H.-P. Hsu, S. C. Lin, and U. H. E. Hansmann, Acta Cryst. **A58**, 259 (2002).

[42] H. Arkın and T. Çelik, Eur. Phys. J. B **30**, 577 (2002).

[43] S. Gwo, C.-P. Chou, C.-L. Wu, Y.-J. Ye, S.-J. Tsai, W.-C. Lin, and M.-T. Lin, Phys. Rev. Lett. **90**, 185506 (1997).

[44] R. Kubo, J. Phys. Soc. Jpn. **17**, 975 (1962).

[45] J. L. Rodríguez-López, F. Aguilera-Granja, K. Michaelian, and A. Vega, Phys. Rev. B **67**, 174413 (2003).

[46] W. P. Halperin, Rev. Mod. Phys. **58**, 533 (1986).

[47] K.-H. MeiwesBroer, editor, *Metal Clusters at Surfaces: Structure, Quantum Properties, Physical Chemistry*, Springer, Berlin, 2000.

[48] Y. Li, E. Blaisten-Barojas, and D. A. Papaconstantopoulos, Phys. Rev. B **57**, 15519 (1998).

[49] J. Hernandez-Rojas and D. J. Wales, J. Chem. Phys. **119**, 7800 (2003).

[50] J. A. Northby, J. Chem. Phys. **87**, 6166 (1987).

[51] R. H. Leary, J. Global Optimization **11**, 35 (1997).

[52] D. J. Wales, J. P. K. Doye, A. Dullweber, M. P. Hodges, F. Y. Naumkin, F. Calvo, J. Hernandez-Rojas, and T. T. Middleton, The Cambridge Cluster Database, http://www-wales.ch.cam.ac.uk/CCD.html.

[53] D. Romero, C. Barron, and S. Gomez, Comput. Phys. Comm. **123**, 87 (1999).

[54] Y. Xiang, H. Jiang, W. Cai, and X. Shao, J. Phys. Chem. A **108**, 3586 (2004).

[55] Y. Xiang, L. Cheng, W. Cai, and X. Shao, J. Phys. Chem. A **108**, 9516 (2004).

[56] X. Shao, Y. Xiang, and W. Cai, J. Phys. Chem. A **109**, 5193 (2005).

[57] R. P. Gupta, Phys. Rev. B **23**, 6265 (1981).

[58] F. Cleri and V. Rosato, Phys. Rev. B **48**, 22 (1993).

[59] J. Z. Y. Chen, A. S. Lemak, J. R. Lepock, and J. P. Kemp, Proteins **51**, 283 (2003).

[60] http://folding.stanford.edu/.

[61] C. Branden and J. Tooze, *Introduction to Protein Structure*, Garland Publishing, Inc., 2nd edition, 1999.

[62] G. Rhodes, *Crystallography Made Crystal Clear*, Academic Press, 2nd edition, 2000.

[63] T. E. Creighton, *Proteins: Structures and Molecular Properties*, W. H. Freeman, 2nd edition, 1992.

[64] A. R. Leach, *Molecular Modelling Principles and Applications*, Prentice Hall, 2nd edition, 2001.

[65] M. A. Martí-Renom, A. C. Stuart, A. Fiser, R. Sánchez, F. Melo, and A. Šali, Annu. Rev. Biophys. Biomol. Struct. **29**, 291 (2000).

[66] C. B. Anfinsen, Science **181**, 223 (1973).

[67] P. L. Privalov, Adv. Protein Chem. **33**, 167 (1979).

[68] G. Nemethy, M. S. Pottle, and H. A. Scheraga, J. Phys. Chem. **87**, 1883 (1983).

[69] M. J. Sippl, G. Nemethy, and H. A. Scheraga, J. Phys. Chem. **88**, 6231 (1984).

[70] G. Nemethy, K. D. Gibson, K. A. Palmer, C. N. Yoon, G. Paterlini, A. Zagari, S. Rumsey, and H. A. Scheraga, J. Phys. Chem. **96**, 6472 (1992).

[71] Z. Wang and R. Pachter, J. Comput. Chem. **18**, 323 (1997).

[72] L. Ingber, http://www.ingber.com/ASA-README.html.

[73] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannary, *Numerical Recipes in C, 2nd Edition*, Cambridge University Press, 1992.

[74] D. Liu and J. Nocedal, Math. Program. **45**, 503 (1989).

[75] U. H. E. Hansmann and Y. Okamoto, Phys. Rev. E **56**, 2228 (1997).

[76] A. C. Melissinos, *Experiments in Modern Physics*, Academic Press, New York and London, 1966.

[77] S. J. Weiner, P. A. Kollman, D. A. Case, U. C. Singh, C. Ghio, G. Alagona, S. Profeta. Jr., and P. Weiner, J. Am. Chem. Soc. **106**, 765 (1984).

[78] S. J. Weiner, P. A. Kollman, D. T. Nguyen, and D. A. Case, J. Comput. Chem. **7**, 230 (1986).

[79] W. D. Cornell, P. Cieplak, C. I. Bayly, I. R. Gould, K. M. Merz, Jr., D. M. Ferguson, D. C. Spellmeyer, T. Fox, J. W. Caldwell, and P. A. Kollman, J. Am. Chem. Soc. **117**, 5179 (1995).

[80] P. A. Kollman, R. Dixon, W. Cornell, T. Fox, C. Chipot, and A. Pohorille, The development/application of a minimalist organic/biochemical molecular mechanic force field using a combination of ab initio calculations and experimental data, in *Computer Simulation of Biomolecular Systems*, edited by A. Wilkinson, P. Weiner, and W. van Gunsteren, volume 3, page 83, Elsevier, 1997.

[81] J. Wang, P. Cieplak, and P. A. Kollman, J. Comput. Chem. **21**, 1049 (2000).

[82] Y. Duan, C. Wu, S. Chowdhury, M. C. Lee, G. Xiong, W. Zhang, R. Yang, P. Cieplak, R. Luo, and T. Lee, J. Comput. Chem. **24**, 1999 (2003).

[83] B. R. Brooks, R. E. Bruccoleri, B. D. Olafson, D. J. States, S. Swaminathan, and M. Karplus, J. Comput. Chem. **4**, 187 (1983).

[84] A. D. MacKerell, Jr., B. Brooks, C. L. Brooks, III, L. Nilsson, B. Roux, Y. Won, and M. Karplus, Charmm: The energy function and its parameterization with an overview of the program, in *The Encyclopedia of Computational Chemistry*, edited by P. v. R. Schleyer *et al*, volume 1, page 271, John Wiley & Sons, Chichester, 1998.

[85] F. A. Momany, R. F. McGuire, A. W. Burgess, and H. A. Scheraga, J. Phys. Chem. **79**, 2361 (1975).

[86] W. L. Jorgensen and J. Tirado-Rives, J. Am. Chem. Soc. **110**, 1657 (1988).

[87] G. Kaminski, E. M. Duffy, T. Matsui, and W. L. Jorgensen, J. Phys. Chem. **98**, 13077 (1994).

[88] W. L. Jorgensen, D. Maxwell, and J. Tirado-Rives, J. Am. Chem. Soc. **118**, 11225 (1996).

[89] Y. Ueeda, H. Taketomi, and N. Gō, Biopolymers **17**, 1531 (1978).

[90] A. M. Ferrenberg and R. H. Swendsen, Phys. Rev. Lett. **61**, 2635 (1988).

[91] F. Wang and D. P. Landau, Phys. Rev. Lett. **86**, 2050 (2001).

[92] F. Wang and D. P. Landau, Phys. Rev. E **64**, 056101 (2001).

[93] G. S. Pawley, K. C. Bowler, R. D. Kenway, and D. J. Wallace, Comput. Phys. Commun. **37**, 251 (1985).

[94] S. Hayryan, C.-K. Hu, S.-Y. Hu, and R.-J. Shang, J. Comput. Chem. **22**, 1287 (2001).

[95] D. B. Jones and J. M. Goodfellow, J. Comput. Chem. **14**, 127 (1993).

[96] F. Eisenmenger, U. H. E. Hansmann, S. Hayryan, and C.-K. Hu, Comput. Phys. Commun. **138**, 192 (2001).

[97] B. Mehlig, D. Heermann, and B. Forrest, Phys. Rev. B **45**, 679 (1992).

[98] K. Hukushima and K. Nemoto, J. Phys. Soc. Jpn **65**, 1604 (1996).

[99] R. H. Swendsen and J.-S. Wang, Phys. Rev. Lett **57**, 2607 (1986).

[100] M. C. Tesi, E. J. J. van Rensburg, E. Orlandini, and S. G. Whittington, J. Stat. Phys. **82**, 155 (1996).

[101] C.-Y. Lin, C.-K. Hu, and U. H. E. Hansmann, Proteins **52**, 436 (2003).

[102] Y. Sugita and Y. Okamoto, Chem. Phys. Lett. **329**, 261 (2000).

[103] A. Mitsutake and Y. Okamoto, Chem. Phys. Lett. **332**, 131 (2000).

[104] G. Geist, J. Kohl, and P. Papadopoulous, Calculateurs Paralleles **8**, 137 (1996).

[105] http://www.colby.edu/chemistry/PChem/scripts/ABC.html.

[106] S. Gwo, Private communication.

[107] J. P. K. Doye, arXiv:cond-matt/0007338 v2 17 Nov 2000.

[108] A. M. Ferrenberg and R. H. Swendsen, Phys. Rev. Lett. **63**, 1195 (1989).

[109] P. M. C. de Oliveira, T. J. P. Penna, and H. J. Herrmann, Braz. J. Phys. **26**, 677 (1996).

[110] P. M. C. de Oliveira, T. J. P. Penna, and H. J. herrmann, Eur. Phys. J. B **1**, 205 (1998).

[111] P. M. C. de Oliveira, Eur. Phys. J. B **6**, 111 (1998).

[112] A. R. Lima, P. M. C. de Oliveira, and T. J. P. Penna, J. Stat. Phys. **99**, 691 (2000).

[113] P. M. C. de Oliveira, Braz. J. Phys. **30**, 766 (2000).

[114] B. A. Berg and U. H. E. Hansmann, Eur. Phys. J. B **6**, 395 (1998).

[115] J.-S. Wang, Eur. Phys. J. B **8**, 287 (1999).

[116] J.-S. Wang and L. W. Lee, Comput. Phys. Commun. **127**, 131 (2000).

[117] U. H. E. Hansmann and Y. Okamoto, J. Comput. Chem. **18**, 920 (1997).

[118] B. A. Berg, U. H. E. Hansmann, and Y. Okamoto, J. Phys. Chem. **99**, 2236 (1995).

[119] K. K. Bhattacharya and J. P. Sethna, Phys. Rev. E **57**, 2553 (1998).

[120] D. P. Landau and F. Wang, Braz. J. Phys. **34**, 354 (2004).

[121] J.-S. Wang and R. H. Swendsen, J. Stat. Phys. **106**, 245 (2002).

[122] M. A. de Menezes and A. Lima, Physica A **323**, 428 (2003).

[123] A. Schug, W. Wenzel, and U. H. E. Hansmann, J. Chem. Phys. **122**, 194711 (2005).

[124] B. J. Schulz, K. Binder, M. Müller, and D. P. Landau, Phys. Rev. E **67**, 067102 (2003).

[125] P. E. Gill, W. Murray, and M. H. Wright, *Practical Optimization*, Acadamic Press, 1981.

[126] Z. Li and H. A. Scheraga, J. Mol. Struct. (Theochem) **179**, 333 (1988).

[127] B. von Freyberg and W. Braun, J. Comput. Chem. **12**, 1065 (1991).

[128] F. Eisenmenger and U. H. E. Hansmann, J. Phys. Chem. B **101**, 3304 (1997).

[129] H. Meirovitch, E. Meirovitch, A. G. Michel, and M. Vásquez, J. Phys. Chem. **98**, 6241 (1994).

[130] I. P. Androulakis, C. D. Maranas, and C. A. Floudas, J. Glob. Opt. **11**, 1 (1997).

[131] W. L. DeLano, http://www.pymol.org.

[132] C. J. McKnight, D. S. Doering, P. T. Matsudaira, and P. S. Kim, J. Mol. Biol. **260**, 126 (1996).

[133] J. Kubelka, W. A. Eaton, and J. Hofrichter, J. Mol. Biol. **329**, 625 (2003).

[134] M. Wang, Y. Tang, S. Sato, L. Vugmeyster, C. J. McKnight, and D. P. Raleigh, J. Am. Chem. Soc. **125**, 6032 (2003).

[135] H. Berman, J. Westbrook, Z. Feng, G. Gilliland, T. Bhat, H. Weissig, I. Shindyalov, and P. Bourne, Nucleic Acid Research **28**, 235 (2000).

[136] http://www.rcsb.org/pdb/.

[137] Y. Duan and P. A. Kollman, Science **282**, 740 (1998).

[138] G. M. S. de Mori, G. Colombo, and C. Micheletti, Proteins **58**, 459 (2005).

[139] M.-Y. Shen and K. F. Freed, Proteins **49**, 439 (2002).

[140] B. S. Kinnear, M. F. Jarrold, and U. H. E. Hansmann, J. Mol. Graph. Model. **22**, 397 (2004).

[141] U. H. E. Hansmann, Physica A **321**, 152 (2003).

[142] D. Poland and H. A. Scheraga, *Theory of Helix-Coil Transitions in Biopolymers*, Academic Press, New York, 1970.

[143] J. P. Kemp and Z. Y. Chen, Phys. Rev. Lett. **81**, 3880 (1998).

[144] U. H. E. Hansmann and Y. Okamoto, J. Chem. Phys. **110**, 1267; **111**, 1339 (erratum) (1999).

[145] N. A. Alves and U. H. E. Hansmann, Phys. Rev. Lett. **84**, 1836 (2000).

[146] N. A. Alves and U. H. E. Hansmann, Physica A **292**, 509 (2001).

[147] Y. Peng and U. H. E. Hansmann, Biophys. J. **82**, 3269 (2002).

[148] A. E. van Giessen and J. E. Straub, J. Chem. Phys. **122**, 024904 (2005).

[149] J. Ireta, J. Neugebauer, M. Scheffler, A. Rojo, and M. Galván, (2005), Submitted to Phys. Rev. Lett.

[150] J. W. Neidigh, R. M. Fesinmeyer, kathryn S. Prickett, and N. H. Andersen, Biochemistry **40**, 13188 (2001).

[151] J. W. Neidigh, R. M. Fesinmeyer, and N. H. Andersen, Nat. Struct. Biol. **9**, 425 (2002).

[152] L. Qiu, S. A. Pabit, A. E. Roitberg, and S. J. Hagen, J. Am. Chem. Soc. **124**, 12952 (2002).

[153] F. Ding, S. V. Buldyrev, and N. V. Dokholyan, Biophys. J. **88**, 147 (2005).

[154] S. Chowdhury, M. C. Lee, G. Xiong, and Y. Duan, J. Mol. Biol. **327**, 711 (2003).

[155] J. W. Pitera and W. Swope, Proc. Natl. Acad. Sci. USA **100**, 7587 (2003).

[156] C. Simmerling, B. Strockbine, and A. E. Roitberg, J. Am. Chem. Soc. **124**, 11258 (2002).

[157] B. Zagrovic and V. Pande, J. Comput. Chem. **24**, 1432 (2003).

[158] R. Zhou, Proc. Natl. Acad. Sci. USA **100**, 13280 (2003).

[159] A. S. N. Seshasayee, Theo. Biol. Med. Model. **2**, 7 (2005).

[160] A. Schug and W. Wenzel, Europhys. Lett. **67**, 307 (2004).

[161] Y. Peng and U. H. E. Hansmann, Phys. Rev. E **68**, 041911 (2003).

[162] N. A. Alves, Y. Peng, and U. H. E. Hansmann, Braz. J. Phys. **34**, 363 (2004).

[163] K. A. Dill, Biochemistry **25**, 1501 (1985).

[164] K. F. Lau and K. A. Dill, Macromolecules **22**, 3986 (1989).

[165] K. A. Dill, Biochemistry **29**, 7133 (1990).

[166] U. H. E. Hansmann, Comput. Sci. Eng. **5(1)**, 64 (2003).

[167] A. Sali, E. Shahknovich, and M. Karplus, nature **369**, 248 (1994).

[168] R. Unger and J. Moult, J. Mol. Biol. **231**, 75 (1993).

[169] K. Yue and K. A. Dill, Proc. Natl. Acad. Sci. USA **92**, 146 (1994).

[170] W. E. Hart and S. Istrail, Fast protein folding in the hydrophobic-hydrophilic model within three-eights of optimal, in *Proceedings of the twenty-seventh annual ACM symposium on Theory of computing*, page 157, 1995.

[171] V. Tozzini, Curr. Opin. Struct. Biol. **15**, 144 (2005).

[172] C. Clementi, A. E. García, and J. N. Onuchic, J. Mol. Biol. **326**, 933 (2003).

[173] H. Kaya and H. S. Chan, J. Mol. Biol. **326**, 911 (2003).

[174] M. S. Cheung, A. E. García, and J. N. Onuchic, Proc. Natl. Acad. Sci. USA **99**, 685 (2002).

[175] V. Tozzini and J. A. McCammon, Protein Sci. **13(suppl 1)**, 194 (2004).

[176] J. Tryska, V. Tozzini, and J. A. McCammon, Protein Sci. **13(suppl 1)**, 121 (2004).

[177] I. Bahar and R. L. Jernigan, J. Mol. Biol. **266**, 195 (1997).

[178] J. A. Saunders and H. A. Scheraga, Biopolymers **68**, 300 (2003).

[179] M. Khalili, J. A. Saunders, A. Livo, S. Ołdziej, and H. A. Scharaga, Protein Sci. **13**, 2725 (2004).

[180] N. Kurt, T. Haliloglu, and C. A. Schiffer, Biophys. J. **85**, 853 (2003).

[181] A. Liwo, S. Ołdziej, M. R. Pincus, R. J. Wawak, S. Rackovsky, and H. A. Scheraga, J. Comput. Chem. **18**, 849 (1997).

[182] A. Liwo, M. R. Pincus, R. J. Wawak, , S. Rackovsky, S. Ołdziej, and H. A. Scheraga, J. Comput. Chem. **18**, 874 (1997).

[183] A. V. Smith and C. K. Hall, Proteins **44**, 344 (2001).

[184] A. V. Smith and C. K. Hall, Proteins **44**, 376 (2001).

[185] H. D. Nguyen, A. J. Marchut, and C. K. Hall, Protein Sci. **13**, 2909 (2004).

[186] A. Fernández and A. Colubri, Proteins **48**, 293 (2002).

[187] S. Takada, Z. Luthey-Schulten, and P. G. Wolynes, J. Chem. Phys. **110**, 11616 (1999).

[188] Y. Fujitsuka, S. Takada, Z. A. Luthey-Schulten, and P. G. Wolynes, Proteins **54**, 88 (2004).

[189] F. Forcellino and P. Derreumaux, Proteins **45**, 159 (2001).

[190] G. Wei, N. Mosseau, and P. Derremaux, Proteins **56**, 464 (2003).

[191] S. Lee, Y. Fujitsuka, H. Kim do, and S. Takada, Proteins **55**, 128 (2004).

[192] J. W. Ponder and D. A. Case, Force fields for protein simulations, in *Advances in Protein Chemistry*, edited by V. Daggett, volume 66, page 27, Elsevier Acadamic Press, 2003.