ABSTRACT

Title of Dissertation: HOMOTOPY OPTIMIZATION METHODS AND PROTEIN STRUCTURE PREDICTION

Daniel Michael Dunlavy, Doctor of Philosophy, 2005

Dissertation directed by: Professor Dianne P. O'Leary
Applied Mathematics and
Scientific Computation Program

The focus of this dissertation is a new method for solving unconstrained minimization problems—*homotopy optimization using perturbations and ensembles* (HOPE). HOPE is a homotopy optimization method that finds a sequence of minimizers of a homotopy function that maps a template function to the target function, the function from our minimization problem. To increase the likelihood of finding a global minimizer, points in the sequence are perturbed and used as starting points to find other minimizers. Points in the resulting ensemble of minimizers are used as starting points to find minimizers of the homotopy function as it deforms the template function into the target function.

We show that certain choices of the parameters used in HOPE lead to instances of existing methods: probability-one homotopy methods, stochastic search

methods, and simulated annealing. We use these relations and further analysis to demonstrate the convergence properties of HOPE.

The development of HOPE was motivated by the protein folding problem, the problem of predicting the structure of a protein as it exists in nature, given its amino acid sequence. However, we demonstrate that HOPE is also successful as a general purpose minimization method for nonconvex functions.

Numerical experiments performed to test HOPE include solving several standard test problems and the protein folding problem using two different protein models. In the first model, proteins are modeled as chains of charged particles in two dimensions. The second is a backbone protein model, where the particles represent amino acids, each corresponding to a hydrophobic, hydrophilic, or neutral residue. In most of these experiments, standard homotopy functions are used in HOPE. Additionally, several new homotopy functions are introduced for solving the protein folding problems to demonstrate how HOPE can be used to exploit the properties or structure of particular problems.

Results of experiments demonstrate that HOPE outperforms several methods often used for solving unconstrained minimization problems—a quasi-Newton method with BFGS Hessian update, a globally convergent variant of Newton's method, and ensemble-based simulated annealing.

HOMOTOPY OPTIMIZATION METHODS AND PROTEIN STRUCTURE PREDICTION

by

Daniel Michael Dunlavy

Dissertation submitted to the Faculty of the Graduate School of the
University of Maryland, College Park in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
2005

Advisory Committee:

Professor Dianne P. O'Leary, Chair
Professor Marco Colombini
Professor William D. Dorland
Professor Howard C. Elman
Professor David Fushman

DEDICATION

To Nancy, Maisy, and Beckett with love and gratitude.

May there always be hope in your lives, as there is in mine.

ACKNOWLEDGEMENTS

I would like to acknowledge the support of my mentors, friends and family throughout the work on my dissertation.

First and foremost, I would like to thank my advisor and mentor, Dianne O'Leary. Her patience and encouragement helped guide me through this challenging work, and the dedication of her time and energy was one of the main reasons that I was able to finish this dissertation. I have learned much from her about mentoring, collaboration, leadership, and friendship, and I hope I can pass this knowledge on to others I encounter in my work.

I would like to acknowledge other mentors that have played important roles throughout my graduate work: Tammy Kolda, for teaching me about large-scale implementation of optimization algorithms; Nil Mackey, for introducing me to homotopy methods and numerical linear algebra; and John Conroy, for always taking the time to listen and help me develop my ideas on projects in many different areas.

I thank Ron Unger, Dave Thirumalai, and Dmitri Klimov for their help in developing the models for protein structure. I also thank Eric Slud for advice on the probabilistic analysis in Chapter 3.

I am extremely grateful for financial support from the National Library of Medicine at the National Institutes of Health through Bioinformatics Individual Fellowship F37LM008162 and from the Department of Computer Science at the University of Maryland, College Park (UMCP). I also thank UMIACS and CSCAMM at UMCP for the use of their resources.

I would like to thank the members of my dissertation committee for their service: Marco Colombini, Bill Dorland, Howard Elman, and David Fushman.

I appreciate the many hours of stimulating discussion with other graduate students at UMCP; particularly Bita Khoshvaghti, Darran Furnival, and Bob Shuttleworth.

I am indebted to my parents, Jim and Peg, and my siblings, Colleen, Tim, and Kevin, for their unwavering support and encouragement throughout all of my graduate work. I also thank Bob and Ann Beebe for all of their support and encouragement.

Lastly, this work would not be possible without the support, encouragement, dedication, sacrifice, and love of my wife, Nancy, and two children, Maisy and Beckett. There are no words to describe what they have given to me, what they mean to me.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

Chapter 1

Introduction

The focus of this dissertation is a new method for solving unconstrained minimization problems—*homotopy optimization using perturbations and ensembles* (HOPE). The development of HOPE was motivated by the protein folding problem, the problem of predicting the structure of a protein as it exists in nature, given its amino acid sequence. However, we demonstrate that HOPE is also successful as a general purpose minimization method for nonconvex functions.

HOPE is a homotopy optimization method that finds a sequence of minimizers of a homotopy function mapping a template function to the target function, the original function being minimized. To increase the likelihood of finding a global minimizer, points in the sequence are perturbed and used as starting points to find other minimizers. Points in the resulting ensemble of minimizers are used as starting points to find minimizers of the homotopy function as it deforms the template function into the target function.

Results of experiments demonstrate that HOPE outperforms several methods often used for solving unconstrained minimization problems—a quasi-Newton method with BFGS Hessian update, a globally convergent variant of Newton's method, and ensemble-based simulated annealing. These experiments include

solving several standard test problems and the protein folding problem using different protein models.

The dissertation is divided into two parts. The first focuses on HOPE's mathematical and algorithmic details and its relation to other optimization methods (Chapters 2–4). The second focuses on the application of HOPE to the protein folding problem (Chapters 5–7).

Existing methods for unconstrained minimization are presented in Chapter 2, where the relationship of HOPE to these other methods is discussed. In the beginning of Chapter 3, the general framework of homotopy methods—the class of methods to which HOPE belongs—is presented. The remainder of the chapter includes a description of the HOPE algorithm and analysis of its convergence properties for problems in one dimension. Results of HOPE applied to several standard test problems are presented in Chapter 4. These results suggest that HOPE can be used for solving general unconstrained minimization problems.

Chapter 5 provides an introduction to the protein folding problem and includes a discussion of existing numerical methods for solving the problem. A general discussion of the application of HOPE to the protein folding problem can be found at the end of Chapter 5. Results of HOPE applied to the protein folding problem for different protein models are discussed in Chapters 6 and 7.

The results of the experiments presented throughout this dissertation demonstrate that HOPE is effective in solving difficult unconstrained minimization problems. In some experiments, no knowledge of the structure of the problems was assumed (Chapter 4), whereas in other experiments a thorough understanding of the structure and properties of the problem was incorporated into the minimization procedure via the homotopy function (Chapter 7). In both situations,

HOPE was more successful at finding global minimizers that the standard methods tested.

In Chapter 8 we present conclusions and directions for future work.

Chapter 2

Methods for Unconstrained Minimization

We are interested in solving the *unconstrained minimization* problem, stated as

$$\begin{aligned} &\text{Given } f : \mathbb{R}^n \to \mathbb{R} \,, \\ &\text{find } x^* \in \mathbb{R}^n \text{ such that } f(x^*) = \min_{x \in \mathbb{R}^n} f(x) \,. \end{aligned} \tag{2.1}$$

If such a point exists, we call it a *global minimizer* and $f(x^*)$ the *global minimum* of f on $\mathbb{R}^n$. To simplify the discussion, we assume that at least one global minimizer exists for all functions considered in this work. Note that the global minimum is unique, but global minimizers are not necessarily unique. For example, the function $f(x) = \cos(x)$ has the unique global minimum of -1 but an infinite number of minimizers at the points $x = (2k+1)\pi$, where $k \in \mathbb{Z}$.

We confine our discussion in this work to minimization problems, since "there is no fundamental difference between minimization and maximization problems" [80]. The reason for this is that

$$\max_{x \in \mathbb{R}^n} f(x) = - \min_{x \in \mathbb{R}^n} -f(x) \,. \tag{2.2}$$

Thus, the ideas and methods in this work associated with minimization, minimizers and minima apply as well to maximization, maximizers, and maxima,

or more generally to optimization, optimizers, and optima. We also note that unconstrained optimization is sometimes referred to as *global optimization*.

In some applications, it may be of interest to find a *local minimizer* of f, a point $x^\dagger$ such that $f(x^\dagger) \leq f(x)$ for all x in a neighborhood of $x^\dagger$. We reiterate that the goal in this work is to find global minimizers, but note that methods for accurately finding local minimizers are often incorporated into methods for finding global minimizers. As Schoen writes in [99],

> Virtually all methods for global optimization consist of two phases: a *global phase*, aimed at thorough exploration of the feasible region or subsets of the feasible region where it is known the global optimum will be found, and a *local phase* aimed at locally improving the approximation to some local optima. Often these two phases are blended into the same algorithm, which automatically switches between exploration and refinement.

In this dissertation we present a new *minimization method*, a numerical procedure designed to solve the minimization problem in (2.1). We say that a minimization method solves the problem if it generates a point, x^1, such that

$$|f(x^1) - f(x^*)| \leq \epsilon \, . \tag{2.3}$$

for some user-defined criterion, $\epsilon \geq 0$, where x^* is a solution to (2.1). Similarly, we say that a minimization method finds a local minimizer of f if it generates a point, x^1, such that

$$|f(x^1) - f(x^\dagger)| \leq \epsilon \, . \tag{2.4}$$

where $x^\dagger$ is a local minimizer of f.

In this chapter, we describe several existing methods for solving unconstrained minimization problems. The goal in presenting these methods here is not to present an exhaustive review. Rather we present several methods that are often used in practice and those that are related to or are of theoretical use in discussing and analyzing HOPE.

There are several ways to classify optimization methods; Torn and Žilinskas present a review in [112] of several classifications found in the literature through 1989 and Pintér presents in [91] a more modern classification, reflecting the additions of methods developed after that date. Although these classifications may be useful for understanding the broad range of methods available for solving global optimization problems, we will take a slightly different approach to classification, dividing the methods into two general classes: local and global methods.

Local optimization methods, or local methods for optimization, are those methods whose goal is to find a local minimizer of the function f. In contrast, the goal of *global optimization methods* is to find a global minimizer (or the global minimum) of f. We note that local methods are sometimes called global methods in the optimization literature, but this is in the sense of global convergence [27, p. 5]; "globally convergent methods" are methods that are guaranteed to converge to a local minimizer from almost every starting point.

2.1 Local Methods

One important difference between local and global optimization methods is that it is possible to verify that a point is a local minimizer of a smooth function whereas more information is required to verify that a point is a global minimizer.

Theorem 1 *Let $f : \mathbb{R}^n \to \mathbb{R}$ be twice continuously differentiable and $x^\dagger$ be a local minimizer of f. Then*

$$(i) \quad \nabla f(x^\dagger) = 0 \ , \ \textit{and} \tag{2.5}$$

$$(ii) \quad \nabla^2 f(x^\dagger) \ \textit{is positive semidefinite} \ . \tag{2.6}$$

For a proof of Theorem 1, see [27, p. 82]. The condition in (2.5) is referred to as the *first order necessary condition* for a local minimizer and the condition in (2.6) is referred to as the *second order necessary condition* for a local minimizer. Note that the condition $\nabla f(x^\dagger) = 0$ holds for all stationary points (minimizers, maximizers and saddle points), so it alone cannot be used to verify that a point is a local minimizer.

Theorem 2 *Let $f : \mathbb{R}^n \to \mathbb{R}$ be twice continuously differentiable. If*

$$\nabla f(x^\dagger) = 0 \quad \textit{and} \quad \nabla^2 f(x^\dagger) \ \textit{is positive definite} \tag{2.7}$$

then $x^\dagger$ is a local minimizer.

For a proof of Theorem 2, see [80, p. 298]. The condition in (2.7) is referred to as the *second order sufficient condition* for a local minimizer.

We now present three local optimization methods: Newton's method, quasi-Newton methods, and the method of steepest descent. The first two are general methods that form the basis of many methods used in practice today, and the third method is of theoretical use but rarely used in practice due to its poor convergence rate.

These methods are iterative methods that generate a sequence of points starting at $x^{(0)} = x^0 \in \mathbb{R}^n$, a point provided as input to the method. The remaining

points in the sequence are generated using the following iterative scheme:

$$x^{(k+1)} = x^{(k)} + \alpha^{(k)} p^{(k)} \ , \tag{2.8}$$

where $x^{(k)}$ is the *current iterate*, $\alpha^{(k)} \in \mathbb{R}$ is the *step length* and $p^{(k)} \in \mathbb{R}^n$ is the *search direction* during the k^{th} iteration of the method. The three methods differ in how $\alpha^{(k)}$ and $p^{(k)}$ are determined.

Newton's Method. In Newton's method, $\alpha^{(k)} = 1$ and $p^{(k)}$ is determined by solving the *Newton equations*,

$$\nabla^2 f(x^{(k)}) p = -\nabla f(x^{(k)}) \ , \tag{2.9}$$

for p. If $\nabla^2 f(x^{(k)})$ is nonsingular, then the solution is unique.

Newton's method is typically derived by approximating the equation $\nabla f(x) = 0$ using the first two terms of the Taylor series of ∇f about the point $x^{(k)}$,

$$\nabla f(x^{(k)} + p) \approx \nabla f(x^{(k)}) + \nabla^2 f(x^{(k)}) p \ . \tag{2.10}$$

Since we want $\nabla f(x^{(k)} + p) = 0$, we find p by solving

$$\nabla f(x^{\dagger}) \approx \nabla f(x^{(k)}) + \nabla^2 f(x^{(k)}) p = 0 \ . \tag{2.11}$$

Another interpretation of Newton's method, and one that highlights the relationship between the three local methods presented here, is that $p^{(k)}$ is chosen as the minimizer of the quadratic function,

$$Q_{newt}(p) = f(x^{(k)}) + \nabla f(x^{(k)})^T p + \frac{1}{2} p^T \nabla^2 f(x^{(k)}) p \ , \tag{2.12}$$

and thus solves the Newton equations. This follows from Theorem 1, since a local minimizer of Q_{newt} satisfies

$$\nabla Q_{newt}(p) = \nabla f(x^{(k)}) + \nabla^2 f(x^{(k)}) p = 0 \tag{2.13}$$

Note that Q_{newt} corresponds to the first three terms of the Taylor series expansion of f around the point $x^{(k)}$ and can be thought of as a quadratic model approximating f at the point $x^{(k)}$. If $\nabla^2 f(x^{(k)})$ is not positive definite, choosing $p^{(k)}$ as the solution of the Newton equations may not lead to a descent direction. In such cases, a diagonal matrix D is often added to $\nabla^2 f(x^{(k)})$ so that $(\nabla^2 f(x^{(k)}) + D)$ is positive definitive, and then $p^{(k)}$ is found by solving $\left(\nabla^2 f(x^{(k)}) + D\right) p = -\nabla f(x^{(k)})$. Choices for D are discussed in [44].

The main drawback of Newton's method is that in general it is only locally convergent, i.e., the method is not guaranteed to converge to a solution from all starting points. However, additional conditions can be placed on search directions and step lengths that lead to globally convergent variants of Newton's method.

Step Length Restrictions. Some globally convergent methods use searches along $p^{(k)}$ to determine suitable step lengths. These *line searches* often assume that $p^{(k)}$ satisfies the following conditions:

$$\textit{Decent direction:} \qquad \nabla f(x^{(k)})^T p^{(k)} < 0 \,, \tag{2.14}$$

$$\textit{Sufficient descent:} \qquad -\frac{\nabla f(x^{(k)})^T p^{(k)}}{\|\nabla f(x^{(k)})\| \cdot \|p^{(k)}\|} \geq \epsilon > 0 \,, \tag{2.15}$$

$$\textit{Gradient related:} \qquad \|p^{(k)}\| \geq c\|\nabla f(x^{(k)})\|,\ c > 0 \,. \tag{2.16}$$

The step length is then determined such that one of the following conditions hold:

$$\textit{Armijo-Goldstein:} \quad f(x^{(k)} + \alpha^{(k)} p^{(k)}) \leq f(x^{(k)}) + \mu \alpha^{(k)} \nabla f(x^{(k)})^T p^{(k)} \,, \tag{2.17}$$

$$\textit{Wolfe:} \qquad |\nabla f(x^{(k)} + \alpha^{(k)} p^{(k)})^T p^{(k)}| \leq \nu |\nabla f(x^{(k)})^T p^{(k)}| \,, \tag{2.18}$$

where $0 < \mu < 1$ and $0 \leq \nu < 1$. *Backtracking* is an example of a line search that starts at $\alpha^{(k)} = 1$ (the full Newton step length) and if necessary, decreases $\alpha^{(k)}$

slowly until one of the the step length conditions above is satisfied. Candidate values of $\alpha^{(k)}$ might be generated, for example, by fitting a cubic polynomial to $f(x^{(k)} + \alpha p^{(k)})$. Details of the derivation of the conditions above and other line searches can be found in [64].

The result of satisfying these conditions on the search directions and step lengths at every iteration is that for any convergent subsequence of the iterates produced, we have

$$\lim_{k \to \infty} \|\nabla f(x^{(k)})\| = 0 \tag{2.19}$$

for problems where f has a bounded level set and ∇f is Lipshitz continuous [80].

Another approach for guaranteeing convergence uses *trust region methods* for determining the search directions and step lengths [107]. In such methods, $p^{(k)}$ is determined by minimizing Q_{newt} subject to the constraint $\|p\| < \Delta_k$, where Δ_k is the radius of the region around $x^{(k)}$ where we "trust" that $Q_{newt}(p)$ sufficiently approximates $f(x^{(k)} + p)$. Often, updates of Δ_k to Δ_{k+1} depend on the ratio of actual decrease (in f) to predicted decrease (in Q_{newt}). This ratio reflects how well the quadratic model can predict function decreases within the trust region. If the ratio is large, Δ_k is typically increased by a constant factor up to some maximum value. If the ratio is small, though, Δ_k is decreased; otherwise, it remains unchanged in the next iteration.

For problems where f has a bounded level set and f, ∇f and $\nabla^2 f$ are all continuous, the use of a trust region method also guarantees that $\|\nabla f(x^{(k)})\| \to 0$ as $k \to \infty$ [80].

Steepest Descent Method. In the steepest descent method, $\nabla^2 f(x^{(k)})$ is approximated using the $n \times n$ identity matrix, I_n, in defining the quadratic model.

Thus, $p^{(k)}$ is chosen to minimize

$$Q_{sd}(p) = f(x^{(k)}) + \nabla f(x^{(k)})^T p + \frac{1}{2} p^T I_n p \ . \tag{2.20}$$

Note that this means that $p^{(k)}$ satisfies $\nabla Q_{sd}(p^{(k)}) = 0$ and that

$$p^{(k)} = -\nabla f(x^{(k)}) \ . \tag{2.21}$$

Thus, we take a step in the direction of steepest descent. The step length, $\alpha^{(k)}$, is usually determined using a line search.

The infinitely small step steepest descent method (i.e., where $\alpha^{(k)}$ is an infinitely small positive number) is often used to define basins of attraction of local minimizers [112].

Definition 1 (Basin of attraction of a local minimizer [112]) *The basin of attraction of $x^\dagger$, a local minimizer of f, is denoted by $\mathcal{R}^\dagger$ and is the largest set of points such that the infinitely small step steepest descent method started at any point $x \in \mathcal{R}^\dagger$ will converge to $x^\dagger$.*

Basins of attraction, useful in characterizing behavior of optimization algorithms around local minimizers, will be used in analyzing the convergence properties of HOPE in Chapter 3.

Quasi-Newton Methods. A quasi-Newton method can be viewed as Newton's method where $\nabla^2 f(x^{(k)})$ is approximated by a positive definite matrix, B_k, in the definition of the quadratic model. Thus, $p^{(k)}$ is chosen to minimize

$$Q_{qn}(p) = f(x^{(k)}) + \nabla f(x^{(k)})^T p + \frac{1}{2} p^T B_k p \ . \tag{2.22}$$

Note that this means that $p^{(k)}$ satisfies $\nabla Q_{qn}(p^{(k)}) = 0$ and thus is a solution to

$$B_k p^{(k)} = -\nabla f(x^{(k)}) \ . \tag{2.23}$$

Different procedures for determining B_k lead to different quasi-Newton methods. Typically, $B_0 = I$, and B_{k+1} is defined using an *update formula* given by

$$B_{k+1} = B_k + U_k \ , \tag{2.24}$$

where U_k is an update, or correction, term.

The BFGS update formula, for example, attributed to Broyden [17], Fletcher [38], Goldfarb [45], and Shanno [103], is given by

$$B_{k+1} = B_k + \frac{(B_k s_k)(B_k s_k)^T}{s_k^T B_k s_k} + \frac{y_k y_k^T}{y_k^T s_k} \tag{2.25}$$

where $s_k = x^{(k+1)} - x^{(k)}$ and $y_k = \nabla f(x^{(k+1)}) - \nabla f(x^{(k)})$.

If B_k is positive definite, then B_{k+1} is positive definite when $y_k^T s_k > 0$ [80]. This condition is typically enforced using a line search along $p^{(k)}$ to determine a value of $\alpha^{(k)}$ such that the condition holds.

Note that the steepest descent method can be viewed as a quasi-Newton method with $B_k = I_n$.

Homotopy Optimization Methods. Another class of local methods, homotopy methods (also referred to as continuation, deformation, or embedding methods), have been effectively used for solving systems of nonlinear equations [5] and nonlinear optimization problems [117, 119]. These methods are characterized by the use of a homotopy—a continuous transformation from one function to another—to solve such problems. Homotopy methods are often used when 1) existing methods only converge from starting points that are close to a solution; or 2) the function being minimized is very complicated. Typically, a homotopy function maps a less complicated, well-understood, or trivial function to the more complicated function of interest. The goal is to choose a homotopy containing a

path from a solution to the easier problem to that of the more difficult problem and then to trace this path numerically.

We define *homotopy optimization methods* as homotopy methods designed to solve optimization problems. These methods are discussed in detail in Chapter 3.

2.2 Global Methods

In this section we present several global methods directly related to HOPE; interested readers are directed to Neumaier's survey [84] and web site [82] for an introduction to global optimization and to several texts [56, 112] and volumes of papers [31, 41, 55, 77, 86] for more details of specific methods.

As was mentioned in the previous section, there are no general sufficient optimality conditions for global minimizers. Conditions have been presented for some classes of global optimization problems [52, 53], but these conditions rely on the structure of the problem (e.g., convexity) to guarantee that a point is a global minimizer.

Stochastic Search Methods. These methods are based on a random search of the function domain and are typically applied to problems on a bounded domain. In this class of methods, the random search ranges from independent sampling from the entire function domain to stochastic perturbations of current iterates. Methods employing the latter most resemble HOPE.

Pure random search (PRS) [16] is the simplest variant of this class of methods. It generates a sequence of points sampled from a distribution on the domain until a point, x, is found with an acceptably low function value, i.e., $f(x) \leq y$ for some $y \in \mathbb{R}$. The expected number of points to sample before this is satisfied is an

exponential function of the dimension of the problem [124]; thus, methods based on PRS are not often used in practice. However, PRS does provide a benchmark for other methods in this class.

Pure adaptive search (PAS) [88, 127] is similar to PRS, although the assumption is made that the next point is always sampled from a region that contains points with strictly lower function values. Under mild assumptions on the optimization problem being solved, the complexity of PAS in terms of iteration count is linear in the dimension of the problem [127]. Different choices of point generation methods and the regions to be explored in subsequent steps lead to different practical methods.

Often used in implementations of PRS and PAS methods, the *Hit-and-Run* (HR) algorithm [13, 106] is a method of generating a sequence of points by taking steps of random lengths in random directions starting from current iterates. The *Improving Hit-and-Run* (IHR) algorithm [128], for example, is an implementation of PAS that uses HR for generating points. The iterates in IHR are chosen as follows

$$x^{(k+1)} = \begin{cases} x^{(k)} + \alpha^{(k)} p^{(k)} & \text{if } f(x^{(k)} + \alpha^{(k)} p^{(k)}) < f(x^{(k)}) \\ x^{(k)} & \text{otherwise} \end{cases} \tag{2.26}$$

where $\alpha^{(k)}$ is a random step length and $p^{(k)}$ is a random direction. Thus, IHR uses stochastic perturbations of the current iterate to find the next iterate. HR is also used in some of the numerical experiments of HOPE presented in Chapter 4.

Simulated Annealing. *Simulated annealing* (SA) methods are based on the analogy defined in [66] between simulating the annealing process of a collection of atoms [75] and solving combinatorial optimization problems.

In each iteration of SA methods 1) a candidate point is generated, 2) the point is tested based on an acceptance criterion dependent on a nonnegative parameter T (representing temperature in the annealing process), and 3) the parameter T is updated according to a cooling schedule.

The procedure for generating candidate points is called the *move class*. Many move classes involve randomly changing one or more of the variables of the current iterate, and the amount of change is often a function of T. For example, *Hide-and-Seek* (HAS) is an SA method that uses HR with step lengths proportional to T for its move class. The move classes of *basin hopping* [116] and *fast annealing* [59] are examples of other types of methods for generating candidates. In basin hopping, candidates are local minimizers near the current iterate, and in fast annealing, candidates are generated by moving the current iterate in a random direction with a step length chosen from a Cauchy distribution. An important feature common to both these move classes is the ability to generate candidate points far from the current iterate, even when T is close to zero. For some problems, this feature helps the SA method avoid getting trapped in regions of attraction of local minimizers [116].

The most often used criterion for accepting a move to a candidate point is the Metropolis criterion [75]. Using this acceptance criterion, the next iterate is chosen with probability

$$P_T(x, x_c) = \begin{cases} 1 & \text{if } f(x_c) < f(x) \\ e^{\frac{f(x)-f(x_c)}{T}} & \text{otherwise} \quad . \end{cases} \tag{2.27}$$

where x is the current iterate and x_c is the candidate point. The benefit of using this acceptance probability is that for constant T the iterates converge to a Boltzmann distribution. Thus, the sampling of candidates points is biased toward

global minimizers, and as $T \to 0$, the distribution concentrates more around the global minimizers. Other acceptance probabilities that have the Boltzmann distribution as their stationary distributions for fixed T are used in some SA methods, e.g. *fast annealing* [94].

The *cooling schedule* of an SA method is the procedure for determining the sequence of temperatures to be used. An exponential cooling schedule is often used in practice; implementation is straightforward and theoretical convergence properties of SA methods employing such a schedule are well documented [114]. In this schedule, the temperature during the k^{th} iteration is given by

$$T^{(k)} = T^0 a^k \ , \tag{2.28}$$

where T^0 is the initial temperature and $a \in (0, 1)$ (with typical values close to 1). Other commonly used schedules include adaptive cooling and nonmonotonic cooling [94]. Adaptive cooling schedules incorporate system information (e.g., amount of decrease in function value between successive iterates) into the choice of $T^{(k)}$. In nonmonotonic cooling, values of $T^{(k)}$ are generated and accepted in a process similar to candidate point generation and acceptance; thus, the sequence of temperatures is not guaranteed to be decreasing as k increases.

SA methods converge to a global minimizer with probability one (i.e., almost surely) [74, 114]. However, convergence results found in the literature often assume extremely slow cooling schedules or place requirements on the move class that are prohibitive for practical implementation. Nevertheless, SA methods have proven successful in solving a wide range of large-scale optimization problems in circuit design, combinatorics, data analysis, imaging, neural networks, biology, physics, geophysics, finance, and military applications [59].

Evolutionary Algorithms. A class of global methods with which simulated annealing is sometimes associated is called *evolutionary algorithms* (EA). This class also includes genetic algorithms, genetic programming, evolutionary programming, and evolutionary strategies [9]. These methods start with an ensemble, or population, of points and use *mutation* and *selection* procedures to move closer to a global minimizer. In SA methods, the mutation procedure is the move class and the selection procedure is the Metropolis criterion. The main difference between EA and SA is that EA are applied to ensembles of points.

The connection between EA and HOPE is the use of ensembles of points that are mutated (i.e., perturbed) and the use of a selection procedure for, among other purposes, controlling the amount of computation performed.

Smoothing Methods. Smoothing methods are typically used for finding global minimizers of continuously differentiable functions. In general, these methods start by finding a global minimizer of a less complicated approximation of the original function—less complicated in the sense of fewer local minimizers, larger basins of attraction of minimizers, etc. The approximation function is then deformed into the original function in a series of steps, where a minimizer of the deformed function at one step is found starting from the minimizer found at the previous step. The goal is to choose an approximation function that is easy to minimize yet captures the global, or macroscopic, features that most influence the location of the global minimizer of the original function.

Many smoothing methods for global optimization were originally developed for minimizing potential energy functions associated with atomic clusters or small molecules [98]. However, recent work shows that such methods can be extended to solve more general optimization problems [3, 4].

Homotopy Optimization Methods. Global homotopy optimization methods have been developed that find all local minimizers (or stationary points) of a function [29, 111]. However, due to the amount of computation required in these methods, they are typically only applicable to problems with a small number of local minimizers.

Homotopy optimization methods are often referred to as smoothing methods as many of the deformation functions used in smoothing methods are homotopies. The main difference between these two methods—one that is not readily found in the literature—is that a homotopy function does not necessarily begin from a less complicated approximation of the original function. In general, homotopy functions, or homotopies, are continuous functions mapping a *template* function to a *target* function, the original function to be minimized; and no assumption is made about the template approximating the target. We make this distinction because HOPE uses homotopies between template and target functions that are both very complicated in some of the experiments presented in Chapters 4 and 7. Note, though, that the use of such homotopies should be used only when a minimizer of the template function is known or trivial to compute. Otherwise too much work may be required just to start the method.

Chapter 3

HOPE

In this chapter, we introduce HOPE, a new homotopy method for solving unconstrained minimization problems. We start the discussion of HOPE with a review of homotopy methods for solving systems of equations. We then discuss homotopy methods for solving unconstrained optimization problems. The HOPE algorithm is then presented, followed by an analysis of its performance using several special cases.

Unless otherwise noted, all functions presented in this chapter are assumed to be sufficiently smooth. Specifically, we assume that the first and second derivatives of scalar functions and the first derivatives of vector functions exist and are continuous.

3.1 Homotopy Methods for Solving Nonlinear Equations

We briefly present here the framework of homotopy methods for solving systems of nonlinear equations. Interested readers should consult [5] for more details on the background, analysis, and implementation of such methods.

Consider the following problem:

$$\text{Find } x \in \mathbb{R}^n, \text{ such that } F^1(x) = 0 \, . \tag{3.1}$$

where $F^1 : \mathbb{R}^n \to \mathbb{R}^n$. A solution to this problem is assumed to exist and is denoted by x^*. In general, more than one solution may exist and F^1 may be a very complicated function.

The general framework of a homotopy method for finding a solution of (3.1) can be described as follows:

(i) Define a function, $F^0 : \mathbb{R}^n \to \mathbb{R}^n$, where a solution, denoted by x^0, to the problem

$$F^0(x) = 0$$

exists and is either known or easy to compute.

(ii) Define a continuous *homotopy function*, $h : \mathbb{R}^{n+1} \to \mathbb{R}^n$, a function of the original variables, $x \in \mathbb{R}^n$, plus a *homotopy parameter*, $\lambda \in \mathbb{R}$, such that

$$h(x, \lambda) = \begin{cases} F^0(x), & \text{if } \lambda = 0 \text{ and} \\ F^1(x), & \text{if } \lambda = 1 \, . \end{cases} \tag{3.2}$$

It follows from the definitions of x^0 and x^1 that

$$h(x^0, 0) = F^0(x^0) = 0 \text{ and}$$
$$h(x^*, 1) = F^1(x^*) = 0 \, .$$

Examples of homotopy functions typically used in practice are the following:

$$\begin{aligned} &\text{Convex :} && h(x, \lambda) = (1 - \lambda)F^0(x) + \lambda F^1(x) \\ &\text{Fixed point :} && h(x, \lambda) = F^1(x) - (1 - \lambda)F^1(a) \end{aligned}$$

where $a \in \mathbb{R}^n$ is fixed, but arbitrary. Note that the fixed point homotopy can also be viewed as a convex homotopy with $F^0(x) = F^1(x) - F^1(a)$.

(iii) Attempt to trace an equilibrium curve $u(\tau) = u(x(\tau), \lambda(\tau)) \in h^{-1}(0)$ from the starting point $u(0) = (x^0, 0)$ to the point $(x^*, 1)$. In other words, starting at $u(0)$, attempt to follow $u(\tau)$ such that $h(u(\tau)) = 0$, as τ varies until $\lambda(\tau) = 1$. In general, there is no restriction on how τ is allowed to vary; however, this may not be practical and some restrictions may be needed to guarantee convergence of the method in a reasonable amount of time. The choice of $\tau = \lambda$ is convenient, but different parameterizations may be more suitable depending on the homotopy function used.

In summary, homotopy methods for solving (3.1) require definitions of the functions $F^0(x)$ and $h(x, \lambda)$, determination of the point x^0, and a method for numerically tracing the curve $u(\tau)$. Publicly available implementations of homotopy methods for solving (3.1) include LOCA [95], HOMPACK [118, 120], AUTO [32, 33, 34], and CONTENT/MATCONT [28, 68]. These implementations include several standard curve tracing algorithms and homotopy functions.

A *continuation method* is an instance of a homotopy method in which the parameter of the equilibrium curve (τ) increases monotonically as the equilibrium curve is traced. An example of a continuation method (CONT), adapted from [5], is presented in Figure 3.1. CONT attempts to trace an equilibrium curve of h, parameterized using $\tau = \lambda$, from $(x^0, 0)$ to $(x^1, 1)$, where x^1 is an approximation of x^*. CONT generates a sequence of points, $(x^{(k)}, \lambda^{(k)})$, satisfying $h(x^{(k)}, \lambda^{(k)}) = 0$. In practice only an estimate of a point on the equilibrium curve is found, i.e., $\|h(x^{(k)}, \lambda^{(k)})\| \leq \epsilon$ for some acceptable tolerance $\epsilon > 0$, and this is typically accomplished using an iterative method starting at the point $(x^{(k-1)}, \lambda^{(k)})$.

The goal in using CONT is to choose $\Delta\lambda$ to be small enough to guarantee the convergence of an iterative process solving $h(x, \lambda) = 0$ in Step 5 but large

1. Input: $x^{(0)} = x^0$ where $F(x^0) = 0$; $m \geq 1$
2. Initialize: $\lambda^{(0)} = 0$; $\Delta\lambda = 1/m$
3. **for** $k = 1, \ldots, m$
4. $\quad \lambda^{(k)} = \lambda^{(k-1)} + \Delta\lambda$
5. $\quad$ solve $h(x, \lambda^{(k)}) = 0$, starting with $x^{(k-1)}$, obtaining $x^{(k)}$
6. **end**
7. Output: $x^1 = x^{(m)}$

Figure 3.1: CONT Algorithm.

enough to converge to $(x^1, 1)$ in a reasonable amount of time (and using a reasonable amount of computation). The use of a fixed value for $\Delta\lambda$ in CONT is for illustrative purposes only and $\Delta\lambda$ is often determined adaptively in continuation methods used in practice.

One potential drawback in using CONT is that the equilibrium curve being traced may contain turning points with respect to λ, points where the curve is not defined locally for $\lambda + \Delta\lambda$ for any positive $\Delta\lambda$. Since λ increases monotonically, the path cannot be traced past the turning point when λ is used as the parameter of the equilibrium curve, and CONT may fail to converge to a solution. A reparameterization of the equilibrium curve by its arc length is often used in practice to avoid this problem with turning points [5].

3.2 Homotopy Optimization Methods

We now describe homotopy methods for solving unconstrained optimization problems. We also discuss the connection of these methods to the homotopy methods presented in the previous section.

Given a function, $f^1 : \mathbb{R}^n \to \mathbb{R}$, we are interested in solving the following unconstrained minimization problem:

$$\min_{x \in \mathbb{R}^n} f^1(x) \, . \tag{3.3}$$

We assume that a solution to this problem exists and denote it x^*. In general, more than one solution may exist and f^1 can be a very complicated function.

By defining a homotopy function between another function, $F^0 : \mathbb{R}^n \to \mathbb{R}$, and ∇f^1, we could consider solving this problem using the homotopy methods discussed in the previous section. By doing so, we would find a point, x^1, such that $\nabla f^1(x^1) = 0$. However, x^1 is not guaranteed to be a local minimizer of f^1, since $\nabla f^1(x^1) = 0$ when x^1 is a local maximizer or saddle point as well. In order to guarantee that x^1 is a local minimizer of f^1, the Hessian matrix, $\nabla^2 f^1(x^1)$, must be positive definite. Depending on the choice of the homotopy function and the method used for solving the system of equations, though, the solution produced by these homotopy methods may not satisfy this condition. However, this approach has proven successful for convex functions [119].

In this thesis, we take a slightly different approach in developing a homotopy optimization method, where the general framework is as follows:

(i) Define a function, $f^0 : \mathbb{R}^n \to \mathbb{R}$, for which a *local minimizer*, denoted by x^0, exists and is either known or trivial to compute.

(ii) Define a continuous homotopy function, $h : \mathbb{R}^{n+1} \to \mathbb{R}$, a function of the original variables, $x \in \mathbb{R}^n$, plus a homotopy parameter, $\lambda \in \mathbb{R}$, such that

$$h(x, \lambda) = \begin{cases} f^0(x), & \text{if } \lambda = 0 \text{ and} \\ f^1(x), & \text{if } \lambda = 1 \ . \end{cases} \tag{3.4}$$

Since x^0 is a minimizer of f^0 and x^* is a minimizer of f^1, it follows that

$$\nabla_x h(x^0, 0) = \nabla f^0(x^0) = 0 \text{ and}$$
$$\nabla_x h(x^*, 1) = \nabla f^1(x^*) = 0 \ ,$$

where $\nabla_x h(x, \lambda)$ denotes the partial derivatives of $h(x, \lambda)$ with respect to x.

(iii) Starting at $(x^0, 0)$, generate a sequence of points, $\left\{(x^{(k)}, \lambda^{(k)})\right\}_{k=0}^{m}$, where each $x^{(k)}$ is a local minimizer of $h(x, \lambda^{(k)})$ with respect to x, and $\lambda^{(k)} \in [0, 1]$ with $\lambda^{(0)} = 0$ and $\lambda^{(m)} = 1$.

Each point in the sequence produced in step (iii) is a local minimizer of h. Specifically, the last point in the sequence, $(x^1, 1) = (x^{(m)}, \lambda^{(m)})$ is a local minimizer of $h(x, 1)$. Since $h(x^1, 1) = f^1(x^1)$, then x^1 is a local minimizer of f^1.

Steps (i) and (ii) are analogous to the first two steps in the framework for homotopy methods for solving systems of equations. The difference is in step (iii) of the two frameworks, where the tracing of an equilibrium curve is replaced by the generation of a sequence of minimizers of h with respect to x. The connection between the two frameworks is that all of the points generated in step (iii) above are on *some* equilibrium curve of $\nabla_x h$, since they are all minimizers of h.

An example of such a homotopy optimization method (HOM) is presented in Figure 3.2. HOM produces a sequence of points starting at $(x^0, 0)$ and ending at

1. Input: $x^{(0)} = x^0$, a local minimizer of f^0; $m \geq 1$
2. Initialize: $\lambda^{(0)} = 0$; $\Delta\lambda = 1/m$
3. **for** $k = 1, \ldots, m$
4. $\quad \lambda^{(k)} = \lambda^{(k-1)} + \Delta\lambda$
5. $\quad$ minimize $h(x, \lambda^{(k)})$ starting with $x^{(k-1)}$, obtaining $x^{(k)}$
6. **end**
7. Output: $x^1 = x^{(m)}$

Figure 3.2: HOM Algorithm.

$(x^1, 1)$, where x^1 is an approximation of x^*. Note that HOM is almost identical to CONT, with the only difference between the two algorithms being Step 5, where the iterative solution of a system of equations—which is $\nabla_x h(x, \lambda) = 0$ for the minimization problem—is replaced by a minimization of h. Furthermore, HOM and CONT may produce the same points for some problems. The difference is that HOM is allowed to jump from one equilibrium curve to another in subsequent steps in λ. For example, in problems where the equilibrium curves contain turning points, HOM will jump to another equilibrium curve past that turning point (if such a curve exists).

In the end, though, HOM is only guaranteed to find a local minimizer of f^1. Local minimizers may be of interest for some problems or in some application areas; however, we are interested in finding the *global minimizer* of f^1, a solution of the problem in (3.3).

3.3 HOPE Algorithm

HOPE is an extension of HOM that increases the likelihood of finding the global minimizer of f^1. Whereas HOM generates a sequence of points converging to a single local minimizer of f^1, HOPE generates a sequence of ensembles of points where each ensemble member is a local minimizer of the homotopy function. This sequence converges to an ensemble of local minimizers of the homotopy function at $\lambda = 1$.

In Step 5 of the HOM algorithm, the next local minimizer in the sequence, $x^{(k)}$, is found using a local minimization method starting at the previous point in the sequence, $x^{(k-1)}$. In the HOPE algorithm, the next ensemble of local minimizers is found using local minimization starting at the points in the previous ensemble in the sequence along with one or more *perturbed versions* of each of those points. A perturbed version of x is denoted by $\xi(x)$, where $\xi : \mathbb{R}^n \to \mathbb{R}^n$ is a function that stochastically perturbs one or more of the variables in x. In the end, HOPE produces an ensemble of local minimizers of f^1, from which we choose the one with the lowest function value as the best approximation to the true solution.

Due to the exponential growth in the number of points produced at each of the main iterations in HOPE, constraints on computational resources may require limiting the size of the ensemble, thus limiting the number of paths of local minimizers to be followed in the next and subsequent steps in the algorithm. Pruning duplicate points from ensembles may also help in efficiently using computational resources.

Before presenting the details of the HOPE algorithm, we first introduce the following notation. Values input to or initialized in HOPE are indexed using 0. We note that the size of the ensemble may be different at the start and end of an

iteration. Let $c^{(k-1)}$ be the number of points in the ensemble at the beginning of iteration k, with $c^{(0)} = 1$ (i.e., a single starting point is used). The j^{th} point in the ensemble at the start of iteration k is denoted by $x_j^{(k-1)}$. We use a secondary index to keep track of perturbed versions of points in the ensemble. Thus, at the end of iteration k, $x_{j,0}^{(k)}$ is the point found by minimization starting at $x_j^{(k-1)}$, and $x_{j,i}^{(k)}$ is the point found by minimization starting at the i^{th} perturbed version of $x_j^{(k-1)}$.

The HOPE algorithm is presented in Figure 3.3, where the overall structure of HOM and CONT is retained. The differences between HOPE and the previous two algorithms occur in Step 5–14, where the step of minimizing (HOM) or solving a system of equations (CONT) is replaced by the local minimization of ensemble points and their perturbed versions and the determination of the ensemble to be used in the next iteration. In terms of computational costs, HOPE requires more work than HOM at each of the m main iterations, making several calls to a local minimization method and choosing which of the local minimizers to retain for use in the next iteration.

HOPE requires two more input values than HOM: c_{max}, the maximum number of points in an ensemble, and $\hat{c}$, the number of perturbed versions of each point in the ensemble to generate. With no limit on the size of the ensemble and no pruning of duplicate points, the number of points in the ensemble at the end of iteration k is $(\hat{c}+1)^k$. With such ensemble size constraints, this number of points becomes $\min\{c^{(k-1)}(\hat{c}+1), c_{max}\}$. And with pruning for uniqueness, there may be even fewer points in the ensemble.

In Steps 13–14 of HOPE, the ensemble of local minimizers to be used in the next iteration is determined. If the number of distinct local minimizers found at

1. Input: $x_1^{(0)} = x^0$, a local minimizer of f^0; $m \geq 1$; $c_{max} \geq 1$; $\hat{c} \geq 0$
2. Initialize: $\lambda^{(0)} = 0$; $\Delta\lambda = 1/m$; $c^{(0)} = 1$
3. **for** $k = 1, \ldots, m$
4. $\quad \lambda^{(k)} = \lambda^{(k-1)} + \Delta\lambda$
5. $\quad$ **for** $j = 1, \ldots, c^{(k-1)}$
6. $\quad\quad$ minimize $h(x, \lambda^{(k)})$, starting at $x_j^{(k-1)}$, obtaining $x_{j,0}^{(k)}$
7. $\quad\quad$ **if** $\hat{c} > 0$
8. $\quad\quad\quad$ **for** $i = 1, \ldots, \hat{c}$
9. $\quad\quad\quad\quad$ minimize $h(x, \lambda^{(k)})$, starting at $\xi(x_j^{(k-1)})$, obtaining $x_{j,i}^{(k)}$
10. $\quad\quad\quad$ **end**
11. $\quad\quad$ **end**
12. $\quad$ **end**
13. $\quad c^{(k)} = \min\{c^{(k-1)}(\hat{c}+1), c_{max}\}$
14. $\quad x_1^{(k)}, \ldots, x_{c^{(k)}}^{(k)} =$ the $c^{(k)}$ "best" (unique) local minimizers among $x_{j,i}^{(k)}, j = 1, \ldots, c^{(k-1)}, i = 0, \ldots, \hat{c}$
15. **end**
16. Output: $x^1 =$ the point with lowest function value among $x_j^{(k)}, j = 1, \ldots, c^{(m)}$

Figure 3.3: HOPE Algorithm.

the current iteration is less than the maximum ensemble size, then all are used in the next iteration; otherwise we must choose the "best" subset. What constitutes

the best subset may differ with the specific problem or application area to which HOPE is applied and may depend on the iteration number (k), the values of the algorithm parameters (m, c_{max}, and $\hat{c}$), or the choice of local minimization routine (along with *its* parameterization). An obvious measure of what constitutes the best conformations—the one used in the numerical experiments presented in Chapters 4 and 7—is homotopy function value: conformations with the lowest function values are considered the best. However, there may be other suitable (or perhaps even better) measures depending on the choices for f^0 and h; for example, when the minimizers of f^0 and f^1 are related geometrically or where the homotopy function has been designed to deform f^0 into f^1 in a particular manner.

Note that different parameter choices for m and $\hat{c}$ reduce HOPE to HOM and to the local minimization method used. Table 3.1 shows the choices for these parameters that lead to the various instances of HOPE. We thus view HOPE as a method that extends a local method for solving global optimization problems. Results of numerical experiments suggesting that such an extension increase the likelihood of finding the global minimizer of f^1 are presented in Chapter 4. First, though, we gain some insight into the behavior of HOPE through analysis of some special cases.

3.4 Algorithm Analysis

We begin the analysis by showing that with proper choices for the homotopy function, perturbation function, and algorithm parameters, HOPE is equivalent to other methods for which convergence results exist. We then demonstrate the performance of HOPE for a special case, and then generalize the results.

Method	**m**	$\hat{\mathbf{c}}$
Local Method	1	0
HOM	> 1	0
HOPE	> 1	≥ 1

Table 3.1: Parameter choices for m and $\hat{c}$ reducing HOPE to a local minimization method and the HOM algorithm.

3.4.1 Asymptotic Analysis

In this section we present several theoretical results regarding the performance of HOM and HOPE. Most of these results are derived by showing that under certain conditions HOM and HOPE are equivalent to other methods for solving optimization methods. Convergence results for the equivalent methods, then, hold for HOM and HOPE under those conditions.

HOM using Probability-One Homotopy Functions. The first result involves HOM applied to a convex function and follows from the analysis of probability one homotopy methods [117]. In that work, for a convex target function, $f^1 : \mathbb{R}^n \to \mathbb{R}$ and $f^1 \in C^3$, it was shown that an equilibrium curve of $\nabla_x h = 0$, with

$$h(x, \lambda) = (1 - \lambda)\frac{1}{2}(x - x^0)^T(x - x^0) + \lambda f^1(x) \tag{3.5}$$

exists, contains both x^0 and x^*, has finite length, and contains only minimizers of h for almost every point x^0. A consequence of this is that a local minimizer of h exists at all $\lambda \in [0, 1]$. Since f^1 is convex, it has a single unique global minimizer (following from the second-order optimality sufficient conditions). Thus, HOM applied to solving this problem using a globally convergent local minimization

method (e.g., Newton's method with a trust region) will produce a sequence of minimizers of h, converging to $(x^*, 1)$ starting from almost any point x^0. Note that these results hold for any $m \geq 1$.

This result states that for well-behaving homotopy maps mapping a convex quadratic function to a general convex function f^1, HOM converges to the global minimizer of f^1 with probability one.

HOPE as a Stochastic Search Method. We next turn to HOPE and show that under mild assumptions, HOPE is an Improving Hit-and-Run (IHR) method (Section 2.2) when using the Hit-and-Run algorithm to generate perturbed versions of ensemble points. The benefit of this analysis is that we can show that for fixed m and c_{max}, HOPE will converge to a global minimizer of f^1. IHR converges with probability one as the number of points generated by HR goes to ∞; moreover, for one class of functions (Lipshitz, elliptical), the number of points required to guarantee convergence is linear in the dimension of the problem [128].

We now present the assumptions on HOPE required for HOPE to be an IHR method. First, the perturbation function used in HOPE must be HR, as in IHR. Second, to match IHR exactly, the number of iterations of the local minimization method used in HOPE must be set to 0. (We note that since the distribution of points generated by HR is independent of the the distribution of local minimizers found in HOPE, this assumption could be dropped, but we include it to match HOPE to IHR exactly.) Lastly, we assume that $c_{max} = 1$, $m = 1$ and that homotopy function value is used as the measure to constitute the best ensemble points in HOPE.

Under these conditions, HOPE is an IHR method. Thus, following the analysis of IHR in [128], we conclude that HOPE converges with probability one as $\hat{c} \to \infty$.

This result is clearly independent of c_{max}; only one point needs to be saved in the ensemble at a time, since there is no limit on the number of perturbed versions that can be generated. Furthermore, for $m > 1$, we need only consider the behavior of HOPE in the last step in λ. At the end of step $m-1$, we have an ensemble containing $c^{(m-1)}$ points. If we take any one of these points to be x^0, we can apply the previous argument and thus, for $m > 1$, the results hold as well.

We conclude this analysis by noting that using similar assumptions, HOPE can be shown to be a Pure Random Search (PRS) method as well. The convergence of HOPE under those conditions can be shown, but the complexity results of PRS show that the number of points required for this convergence is exponential in the dimension of the problem. However, these results verify that convergence can be guaranteed for HOPE using perturbation functions other than HR.

HOPE as a Simulated Annealing Method. We next discuss the conditions for which HOPE is a simulated annealing (SA) method. Theoretical convergence proofs exist for many variants of SA methods, and results of numerical experiments show that simulated annealing is an effective method for solving problems for which several standard minimization methods fail [74].

Recall that in an SA method, a move class, acceptance criteria, and cooling schedule must be defined. The convergence of SA methods make assumptions on the behavior and properties of these components, and we will need to make equivalent assumptions about the corresponding components in HOPE.

It is possible to combine all three components into the generation of perturbed

versions of ensemble points by defining the perturbation function as

$$\xi_{T(\lambda)}(x) = \begin{cases} \delta x & \text{with probability } P_{T(\lambda)}(x, \delta x) \\ x & \text{with probability } 1 - P_{T(\lambda)}(x, \delta x) \end{cases} \tag{3.6}$$

where δx is a point sampled from a distribution in which every point in the domain has a positive probability of being sampled; P_T is the Metropolis acceptance criterion often used in SA methods; and $T(\lambda)$ is a continuous function of λ defining the temperature such that $T(0) = T^0$ and $T(1) = 0$, e.g., $T(\lambda) = T^0(1 - \lambda)$.

The main difference between HOPE and SA methods is the function being minimized. Moreover, in SA methods, the acceptance criteria depend on the temperature and the values of the function at the current iterate and the candidate point generated using the move class. If we use the identity homotopy function, $h(x, \lambda) = f^1(x)$, in HOPE then the function being minimized as each step in λ is f^1, as in SA methods.

Combining the perturbation function, $\xi_{T(\lambda)}$, with the identity homotopy function, HOPE becomes an SA method. Thus, convergence results that apply to SA methods can be applied to HOPE as well. In [26] it was shown that by placing an upper bound on the rate of temperature decrease and allowing generation of points in the entire domain (or feasible region), an SA method converges to a global minimizer almost surely as $k \to \infty$. Here k is the number of iterations (steps in T) taken.

Analysis of the Parameter *m*. We conclude this section with an analysis of the behavior of HOPE as a function of m. Specifically, we show that when f^1 is well-behaved, then HOPE converges, and we can provide a bound on the number of steps in λ required before HOPE converges with some given probability. We

begin the discussion with a review of the Implicit Function Theorem, which we will use in the analysis.

Theorem 3 (Implicit Function Theorem (IFT)[129]) *Let $h : \mathbb{R}^{n+1} \to \mathbb{R}$ be twice continuously differentiable, $(\hat{x}, \hat{\lambda}) \in \nabla_x h^{-1}(0)$, and $\nabla_x^2 h(\hat{x}, \hat{\lambda})$ be invertible. Then in a neighborhood of $(\hat{x}, \hat{\lambda})$ all points (x, λ) that satisfy $\nabla_x h(x, \lambda) = 0$ are on a single continuously differentiable path through $(\hat{x}, \hat{\lambda})$.*

Next we extend the definition of a basin of attraction (Definition 1) to homotopy functions.

Definition 2 (Basin of attraction of a local minimizer of a homotopy function) *The basin of attraction of $x^\dagger$, a local minimizer of $h(x, \lambda)$ with respect to x, is denoted by $\mathcal{R}^\dagger(\lambda)$ and is the largest set of points such that the infinitely small step steepest descent method started at any point $x \in \mathcal{R}^\dagger(\lambda)$ will converge to $x^\dagger$.*

The basin of attraction of the global minimizer of $h(x, 1) = f^1(x)$ will be denoted by $\mathcal{R}^*(\lambda)$. The final definition we need is that of isolated stationary points.

Definition 3 (Isolated stationary point of a homotopy function) *A stationary point of $h(x, \lambda)$ with respect to x, denoted by x^s, is an isolated stationary point if there is a neighborhood of x^s containing no other stationary points.*

Next we make some assumptions about the functions that we will be dealing with. All functions are assumed to be defined on a bounded, compact domain $\mathcal{X} \subset \mathbb{R}^n$.

(A1) $f^0(x), f^1(x) \in C^2(\mathcal{X}, \mathbb{R})$. That is, these functions have continuous first and second derivatives.

(A2) $h(x, \lambda) \in C^2(\mathcal{X} \times [0, 1], \mathbb{R})$.

(A3) The global minimizer of f^1 is unique and is isolated.

(A4) $\xi_{\mathcal{X}}$ generates points uniformly on $\mathcal{X}$.

We will now define a region in $\mathcal{X} \times [0, 1]$ that will play a key role in proving convergence of HOPE. It follows from (A2)–(A3) and Theorem 3 that in some neighborhood of $(x^*, 1)$, there exists a unique equilibrium curve that passes through $(x^*, 1)$ and contains only minimizers of h. Furthermore, since the global minimizer of f^1 is unique, then there is some value, λ_*, such that for all $\lambda \in [\lambda_*, 1]$, the basin $\mathcal{R}^*$ around the equilibrium curve contains the global minimizers of h and $V(\mathcal{R}^*(\lambda)) > 0$, where $V(\cdot)$ denotes volume measures with respect to $\mathcal{X}$.

In Figure 3.4, we show an example homotopy function in one dimension where $\mathcal{X} = [x_l, x_u]$. The solid curves are equilibrium curves of $\nabla_x h$ and the dashed curves are the boundaries of the basins of attraction of the minimizers of h. The shaded region represents $\mathcal{R}^*$, the basin of attraction of the global minimizers of h for all $\lambda \in [\lambda_*, 1]$, and the solid curve in that region is the curve of global minimizers.

We now focus on the behavior of HOPE for $\lambda \in [\lambda_*, 1]$. First we denote

$$V_* = \min_{\lambda_* \leq \lambda \leq 1} V(\mathcal{R}^*(\lambda)) \tag{3.7}$$

as the minimum volume of the basin of attraction of the global minimizer. Using the perturbation function, $\xi_{\mathcal{X}}$, which samples uniformly from $\mathcal{X}$, the probability that a point will be sampled from $\mathcal{R}^*(\lambda)$ is the ratio of its volume to the volume

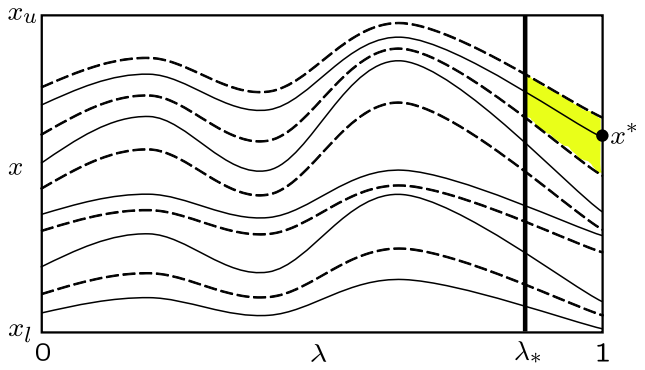


Figure 3.4: Depiction of the point λ_* for a homotopy function in one dimension. The solid lines represent the equilibrium curves of local minimizers of h and the dashed lines represent the boundaries of the basins of attraction associated with those curves. The shaded region is $\mathcal{R}^*$ containing the global minimizers of h for $\lambda \in [\lambda_*, 1]$.

of $\mathcal{X}$. Thus, we have

$$\Pr\left[\, \xi_{\mathcal{X}}(x) \in \mathcal{R}^*\left(\lambda\right) \text{ when } \lambda \in [\lambda_*, 1] \,\right] \geq V_*/V(\mathcal{X}) \,. \tag{3.8}$$

If we let $\hat{c} = 1$ and $c_{max} = 1$, we can compute a lower bound on the probability that HOPE converges as a function of the number of steps taken in λ when $\lambda \geq \lambda_*$. The probability that a single perturbed version is not in $\mathcal{R}^*\left(\lambda\right)$ can be bounded above:

$$\Pr\left[\, \xi_{\mathcal{X}}(x) \text{ is outside } \mathcal{R}^*\left(\lambda\right) \text{ when } \lambda \in [\lambda_*, 1] \,\right] < 1 - V_*/V(\mathcal{X}) \,. \tag{3.9}$$

Since only one perturbed version is generated at each step, we also have

$$\Pr\left[\begin{array}{l} \text{All } \xi_{\mathcal{X}}(x) \text{ generated in } k \text{ iterations of HOPE} \\ \text{are outside } \mathcal{R}^*\left(\lambda\right) \text{ when } \lambda \in [\lambda_*, 1] \end{array}\right] < (1 - V_*/V(\mathcal{X}))^k \tag{3.10}$$

Thus, the probability that at least one perturbed version is in the basin of attraction of the global minimizer when k iterations of HOPE are performed for

$\lambda \in [\lambda_*, 1]$ is bounded below as follows:

$$\Pr\left[\,\exists \xi_{\mathcal{X}}(x) \in \mathcal{R}^*\left(\lambda\right)\,\right] > 1 - \left(1 - V_*/V(\mathcal{X})\right)^k \,. \tag{3.11}$$

It is clear, then, that HOPE will converge in the limit as $k \to \infty$. Moreover, by deriving this bound, we have an upper bound on the amount of computation required for HOPE to converge. The ability to derive such a bound for some problems may prove necessary, especially in the situation where computing resources are limited.

The following theorem refines this bound. In this theorem, we use the convention $\prod_{k=m}^{n} a_k = 1$ if $n < m$.

Theorem 4 *Consider the HOPE algorithm applied to a minimization problem where the functions f^0, f^1, and h satisfy (A1)-(A3). Assume that a function satisfying (A4), $\xi_{\mathcal{X}}$, is used to generate perturbed versions of ensemble points and that $\hat{c} = 1$, $c_{max} = 1$, and m are the parameters used in HOPE. Let k be the smallest integer so that*

(1) $\lambda^{(k)} \geq \lambda_*$, and

(2) $(x^{(k)}, \lambda^{(k)}) \in \mathcal{R}^*\left(\lambda^{(k)}\right) \implies (x^{(k+1)}, \lambda^{(k+1)}) \in \mathcal{R}^*\left(\lambda^{(k+1)}\right)$ for $\lambda \in [\lambda_*, 1]$.

Then either

(i) $x^{(k-1)} \in \mathcal{R}^*\left(\lambda^{(k)}\right)$ and HOPE converges to the global minimizer of f^1, or

(ii) the probability that HOPE converges to the global minimizer of $f^1(x)$, i.e., the probability that at least one point generated in HOPE for $\lambda \in [\lambda_*, 1]$ is in $\mathcal{R}^*$, is given by

$$\sum_{r=k}^{m} \left\{ V_r \prod_{s=k}^{r-1} (1 - V_s) \right\} \,, \tag{3.12}$$

with

$$V_k = \frac{V(\mathcal{R}^*\left(\lambda^{(k)}\right))}{V(\mathcal{X})} \ . \tag{3.13}$$

Proof. (i) Assume $x^{(k-1)} \in \mathcal{R}^*\left(\lambda^{(k)}\right)$. Then by (2) above, all subsequent iterates are in $\mathcal{R}^*$. Thus, HOPE converges to the global minimizer of f[1].

(ii) Assume $x^{(k-1)} \notin \mathcal{R}^*\left(\lambda^{(k)}\right)$. Now the probability that a perturbed version of $x^{(k-1)}$ is in $\mathcal{R}^*\left(\lambda^{(k)}\right)$ is given by

$$\Pr\left[\, \xi_{\mathcal{X}}(x^{(k-1)}) \in \mathcal{R}^*\left(\lambda^{(k)}\right) \,\right] = \frac{V(\mathcal{R}^*\left(\lambda^{(k)}\right))}{V(\mathcal{X})} = V_k \ . \tag{3.14}$$

Thus, $1 - V_k$ is the probability that $\xi_{\mathcal{X}}(x^{(k-1)}) \notin \mathcal{R}^*\left(\lambda^{(k)}\right)$.

At step r in λ, $k \leq r \leq m$, the probability that $\xi_{\mathcal{X}}(x^{(r-1)}) \in \mathcal{R}^*\left(\lambda^{(r)}\right)$ and none of the perturbed versions in steps $k, \ldots, r-1$ are in $\mathcal{R}^*$ is given by

$$V_r \prod_{s=k}^{r-1}(1 - V_s) \ . \tag{3.15}$$

Since these events are disjoint, we can sum these probabilities over the $r = k, \ldots, m$ steps in λ to get the probability that at least one of the perturbed versions is in $\mathcal{R}^*$:

$$\sum_{r=k}^{m} \left\{ V_r \prod_{s=k}^{r-1}(1 - V_s) \right\} \ , \tag{3.16}$$

which is the result in (3.12). □

3.4.2 Analysis of N-Modal Sine Functions

In this section we analyze the behavior of HOPE on a special class of functions. These functions were designed specifically to highlight some of the properties of HOPE.

Consider the following *N-modal sine* functions

$$f^0(x) = -\sin(x) + \sin(Nx) \tag{3.17}$$

$$f^1(x) = \quad \sin(x) + \sin(Nx) \tag{3.18}$$

and the convex homotopy function

$$h(x, \lambda) = (1 - \lambda) f^0(x) + \lambda f^1(x) \tag{3.19}$$

where $f^0, f^1 : \mathbb{R} \to \mathbb{R}$ and $h : \mathbb{R} \times [0, 1] \to \mathbb{R}$. Note that the N-modal sine functions are 2π-periodic; thus we confine our analysis to $x \in [0, 2\pi]$. We also limit our analysis to the case when N is even to further simplify the discussion. (For odd N, the only difference is the presence of *two* global minimizers in each period compared with just one for even N.) Figure 3.5 shows a plot of f^0 and f^1 with $N = 10$.

The objective is to analyze the performance of HOPE used to minimize f^1. The N-modal sine functions have the following characteristics:

1. There are N local minimizers in one period.

2. A stationary point of h is either a maximizer or minimizer. There are no saddle points.

3. All stationary points of h are isolated.

4. Since $\nabla_h = N\cos(Nx) + (2\lambda - 1)\cos(x)$, the equilibrium curves of $\nabla_x h$ are linear in λ.

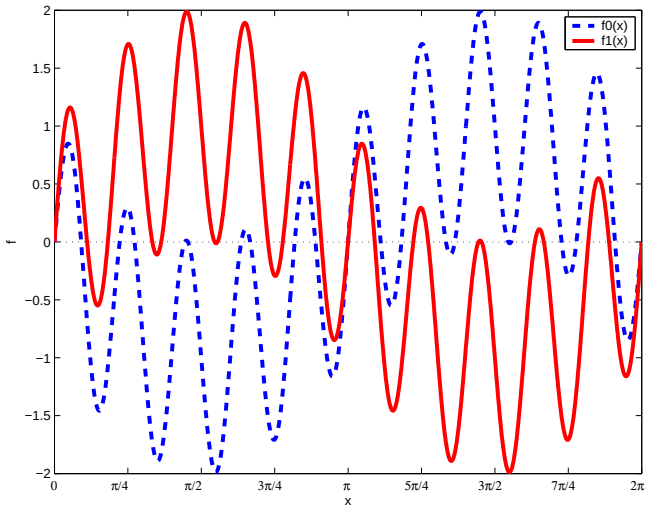


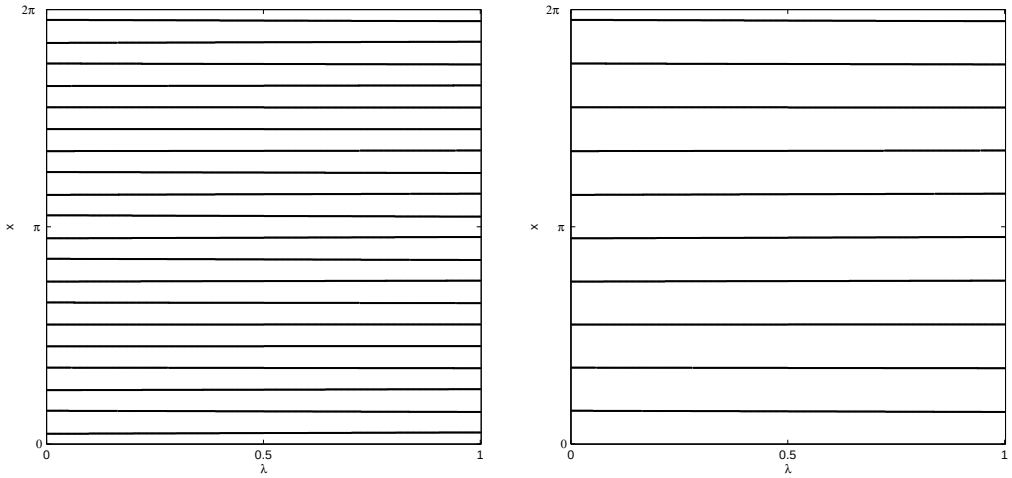
Figure 3.5: N-modal sine functions ($N = 10$) for $x \in [0, 2\pi]$.

5. The basins of attraction of all local minimizers are approximately the same width, $2\pi/N$, and are the open sets of points between local maximizers.

6. The distance between the unique global minimizers of f^0 and f^1 for even N is exactly π, or half the period.

Figure 3.6 presents the equilibrium curves of $\nabla_x h$ for $N = 10$ with (a) showing all curves and (b) showing only the curves of local minimizers. Note that the equilibrium curves are not constant with respect to x, although they appear to be constant in the figure. However, as N increases, the slope of these curves becomes negligible ($\Delta x/\Delta\lambda \to 0$ as $N \to \infty$). We will model the behavior of HOPE applied to this problem assuming the curves are constant.

We now analyze the behavior of HOPE in finding the global minimizer of f^1, starting at one of the local minimizers of f^0. We assume that the infinitely small step steepest descent method is used for local minimization of h. Furthermore, we let $\hat{c} = 1$, $c_{max} = 2^m$, and assume no pruning of duplicate points. Thus,



(a) (b)

Figure 3.6: Equilibrium curves of $\nabla_x h$ for the N-modal sine functions ($N = 10$).

each point in the ensemble will be perturbed once and no ensemble points will be discarded.

HOPE is initialized with an ensemble consisting of a single point. Let $\mathbb{S}^{(k)}$ be the set of points to be used as starting points for local minimization of h during the k^{th} iteration of HOPE. Then,

$$\begin{aligned} \mathbb{S}^{(k)} &= \mathbb{S}_{pf}^{(k)} \cup \mathbb{S}_{rw}^{(k)} \\ &= \left\{\xi(x_j^{(k-1)})\right\}_{j=1}^{2^{(k-1)}} \cup \left\{\xi(x_j^{(k-1)})\right\}_{j=1}^{2^{(k-1)}} \end{aligned}$$

where $\mathbb{S}_{pf}^{(k)}$ are the *path following* starting points and $\mathbb{S}_{rw}^{(k)}$ are the *random walk* starting points.

Since the curves of local minimizers of h are nearly constant with respect to λ for large N, we can assume that the points in $\mathbb{S}_{pf}^{(k)}$ remain on the same equilibrium curve throughout all of the iterations; they do not jump from one curve to another.

The points in $\mathbb{S}_{rw}^{(k)}$ are allowed to move from one basin of attraction to another

depending on the probability distribution function of the perturbation used. The name "random walk" reflects the analogy between them and particles in a random walk [93].

The perturbations used in the analysis are as follows:

$$\xi_N(x) = x + \delta x \ , \tag{3.20}$$

where δx is a random variable chosen from a uniform distribution on the interval $[-3\pi/N, 3\pi/N]$. The local minimizers of h are at the centers of the basins of attraction and the width of each basin is approximately $2\pi/N$. Therefore, a perturbed version of a local minimizer is equally likely to be a point in the same or an adjacent basin of attraction of that minimizer.

Recall that there are N curves of local minimizers of h for $\lambda \in [0, 1]$ and $x \in [0, 2\pi]$. Thus, there are N basins of attraction as well. Since the local minimizers are isolated, we can view the basins of attraction as states in a Markov chain, where S_j is the state that corresponds to the basin of attraction of the j^{th} local minimizer of h with respect to x. The states are assumed to be ordered such that S_N corresponds to the basin of attraction of x^*, the global minimizer of f^1. We let S_i be the state of x^0, the local minimizer of f^0 used as the starting point in HOPE. Due to the periodicity of h, we simplify the discussion of movement from S_1 to S_N and vice versa by denoting $S_0 = S_N$.

Thus, we analyze HOPE as a one-type discrete-time branching random walk (BRW) over N states [10, 51, 93]. At the k^{th} time step, the parents are the points in $\mathbb{E}^{(k-1)}$ and the offspring are the minimizers found using the perturbed versions of those parents as starting points. Since the infinitely small step steepest descent method is used for local minimization, offspring are either in the same state as their parents or in one of the two adjacent states of their parents. Note that this

BRW differs from the classical Galton-Watson process in that the parents never die off (since we have assumed $c_{max} = 2^m$).

The goal is to determine the minimum number of steps to be taken in λ such that an offspring will be in S_N given that $x^0 \in S_i$. This would mean that a point would be in the basin of attraction of the global minimizer of f^1, $\mathcal{R}^*(\lambda)$.

Let $W_{j,k}$ denote the number of ensemble points in S_j at time k (i.e. $\lambda = k/m$). Thus, our task is to determine k_m, the smallest value of k such that

$$W_{j,0} = \begin{cases} 1 & \text{if } j = i \\ 0 & \text{if } j \neq i \end{cases} \qquad \text{and} \tag{3.21}$$

$$\mathrm{E}\left[W_{N,k_m} \right] > 1 \tag{3.22}$$

where $\mathrm{E}[\, y \,]$ is the expected value of y. To compute k_m, we first note that

$$\mathrm{E}\left[W_{N,k+1} \right] = \mathrm{E}\left[W_{N,k} \right] + \frac{1}{3}\mathrm{E}\left[W^-_{1,k} + W^0_{N,k} + W^+_{N-1,k} \right] \ . \tag{3.23}$$

where $W^-_{j,k}$, $W^0_{j,k}$ and $W^+_{j,k}$ are the number of offspring of the parents in S_j that are in states S_{j-1}, S_j, and S_{j+1} at time k. Once an offspring enters S_N, HOPE is guaranteed to obtain the global minimizer. The term $\mathrm{E}\left[W_{N,k} \right]$ corresponds to the points in $\mathbb{S}^{(k)}_{pf}$, and the other terms to those in $\mathbb{S}^{(k)}_{rw}$, respectively. Using this relation, we derive the following set of difference equations for determining the expected number of points in S_j at time $k+1$ in the BRW:

$$b_{j,k+1} = \begin{cases} \frac{1}{6}b_{N,k} + \frac{2}{3}b_{1,k} + \frac{1}{6}b_{2,k} \ , & \text{if } j = 1 \\ \frac{1}{6}b_{j-1,k} + \frac{2}{3}b_{j,k} + \frac{1}{6}b_{j+1,k} \ , & \text{if } j = 2, \ldots, N-1 \\ \frac{1}{6}b_{N-1,k} + \frac{2}{3}b_{N,k} + \frac{1}{6}b_{1,k} \ , & \text{if } j = N \end{cases} \tag{3.24}$$

where $b_{j,k} = 2^{-k}\,\mathrm{E}\left[W_{j,k} \right]$. Thus, we choose as k_m the smallest k such that

$$b_{N,k} = e_i^T (2P)^k e_N \geq 1 \tag{3.25}$$

where

$$P = \frac{1}{6}\begin{bmatrix} 4 & 1 & & & 1 \\ 1 & 4 & 1 & & \\ & \ddots & \ddots & \ddots & \\ & & 1 & 4 & 1 \\ 1 & & & 1 & 4 \end{bmatrix} \tag{3.26}$$

and $e_j \in \mathbb{R}^n$ is a vector of zeros with a 1 in the j^{th} position. Note that e_i represents the starting probability vector for the ensemble of random walkers, since we have assumed that $x^0 \in S_i$. Thus, the term $e_i^T (2P)^k e_N$ represents the expected number of points in the basin of attraction of the global minimizer of f^1 after k iterations of HOPE started at the local minimizer in S_i.

We note that the matrix P is a circulant matrix; thus we can derive a formula for $e_i^T (2P)^k e_N$. We start our discussion of this alternate formulation by presenting some properties associated with circulant matrices [25]. The eigenvalues ψ_m of an $N \times N$ circulant matrix

$$C = \text{circ}(c_0, c_1, \ldots, c_{N-1}) = \begin{bmatrix} c_0 & c_1 & \ldots & c_{N-1} \\ c_{N-1} & c_0 & \ldots & c_{N-2} \\ \vdots & \ddots & \ddots & \vdots \\ c_1 & c_2 & \ldots & c_0 \end{bmatrix} \tag{3.27}$$

are given by

$$\psi_m = \sum_{l=0}^{N-1} c_l e^{i2\pi ml/N} , \tag{3.28}$$

and its eigenvectors v_m are the columns of the $N \times N$ discrete Fourier transform

(DFT) matrix

$$F_N = \frac{1}{\sqrt{N}} \begin{bmatrix} 1 & 1 & 1 & \dots & 1 \\ 1 & \omega & \omega^2 & \dots & \omega^{N-1} \\ 1 & \omega^2 & \omega^4 & \dots & \omega^{2*(N-1)} \\ \vdots & \vdots & \vdots & & \vdots \\ 1 & \omega^{N-1} & \omega^{2(N-1)} & \dots & \omega^{(N-1)(N-1)} \end{bmatrix} \tag{3.29}$$

with $\omega = e^{i2\pi/N}$. Thus, the eigenvalue decomposition for a circulant matrix is given by

$$C = F_N^* \Psi F_N \, . \tag{3.30}$$

with

$$\Psi = \begin{bmatrix} \psi_0 & & & & \\ & \psi_1 & & & \\ & & \psi_2 & & \\ & & & \ddots & \\ & & & & \psi_{N-1} \end{bmatrix} . \tag{3.31}$$

Now we have

$$\begin{aligned} e_i^T (2P)^k e_N &= 2^k P_{i,N}^k \\ &= 2^k \sum_{l=0}^{N-1} \psi_l^k F_{(l+1),i}^* F_{N,(l+1)} \\ &= \frac{2^k}{N} \sum_{l=0}^{N-1} \psi_l^k \left(e^{i2\pi/N}\right)^{l(i-1)} \left(e^{i2\pi/N}\right)^{l(N-1)} \\ &= \frac{2^k}{N} \sum_{l=0}^{N-1} \psi_l^k e^{i2\pi(i-N)l/N} \, . \end{aligned}$$

"Worst-case" performance of HOPE applied to this problem can be estimated by substituting $i = N/2$ in the expression for $P_{i,N}^k$. This corresponds to starting

HOPE from the global minimizer of f^0 since the global minimizers of f^0 and f^1 are a distance of π away from each other. This also corresponds to starting in the state furthest from S_N. Doing so, we get

$$\begin{aligned} 2^k P^k_{N/2,N} &= \frac{2^k}{N} \sum_{l=0}^{N-1} \psi_l^k e^{i2\pi(N/2-N)l/N} \\ &= \frac{2^k}{N} \sum_{l=0}^{N-1} \psi_l^k e^{-i\pi l} \\ &= \frac{2^k}{N} \sum_{l=0}^{N-1} (-1)^l \psi_l^k \\ &= \left(\frac{2}{3}\right)^k \frac{1}{N} \sum_{l=0}^{N-1} (-1)^l \left(2 + \cos(2\pi j/N)\right)^k \ . \end{aligned}$$

Thus, we have derived a simple expression for the expected number of ensemble points in S_N after k iterations of HOPE started at a point in $S_{N/2}$:

$$e_{N/2} \left(2P\right)^k e_N = 2^k P^k_{N/2,N} = \left(\frac{2}{3}\right)^k \frac{1}{N} \sum_{l=0}^{N-1} (-1)^l \left(2 + \cos(2\pi j/N)\right)^k \ .$$

Even for very large N, then, we can easily compute an expected number of iterations that HOPE would need to converge to the global optimizer of this N-modal sine function. We will see next that simulations of HOPE match this predicted behavior quite well.

Simulating HOPE. To conclude our analysis of HOPE applied to the N-modal sine functions, we present the results of simulations of HOPE solving these problems. We compare the number of steps taken in the simulation of HOPE to the predicted number of steps required. In the simulations of HOPE, we keep track of the number of particles in the N different states (basins of attraction) during time (iteration) k and do not actually run the HOPE algorithm to solve the problem. In this way, we can work with much larger ensembles efficiently.

We performed two sets of simulations of HOPE for $N = 5, 6, \ldots, 30$. In the first set of simulations, we start in state $i = N/2$. As mentioned above, this corresponds to simulating the worst-case performance of HOPE. In the second set of simulations, we start in state $i = N/4$. This state is at a distance from S_N equal to the average distance to S_i from all other states. Thus, we attempted to simulate the worst case and average case behaviors of HOPE.

For each N, 1000 simulations were run starting at both $i = N/2$ and $i = N/4$. Figure 3.7 presents the results of the simulations. The lines labelled "$k_m(i = N/2)$" and "$k_m(i = N/4)$" contain the predicted values of k_m for $i = N/2$ and $i = N/4$, respectively. The line labelled "Sim(max)" is the maximum number of steps taken in the simulation before at least one point was in S_N, and the line labelled "Sim(mean)" is the corresponding average number of steps taken. We conclude that the simulations follow closely the behavior predicted by the analysis, off roughly by a constant factor.

To conclude this section, we note that the predicted and simulated behaviors of HOPE presented above match well. For the N-modal sine function, we were able to compute estimates for the number of steps that need to be taken before an ensemble point was in the basin of attraction of the global minimizer. We will see in the next section that in general more assumptions about general homotopy functions may lead to comparable analysis.

3.4.3 Analysis of Problems in One Dimension

We now extend the analysis of HOPE to more general f^0, f^1, and h with $x \in \mathbb{R}$ and $\lambda \in [0, 1]$. We also allow more general perturbation functions to be used. We begin the discussion by assuming the following properties hold for the homotopy

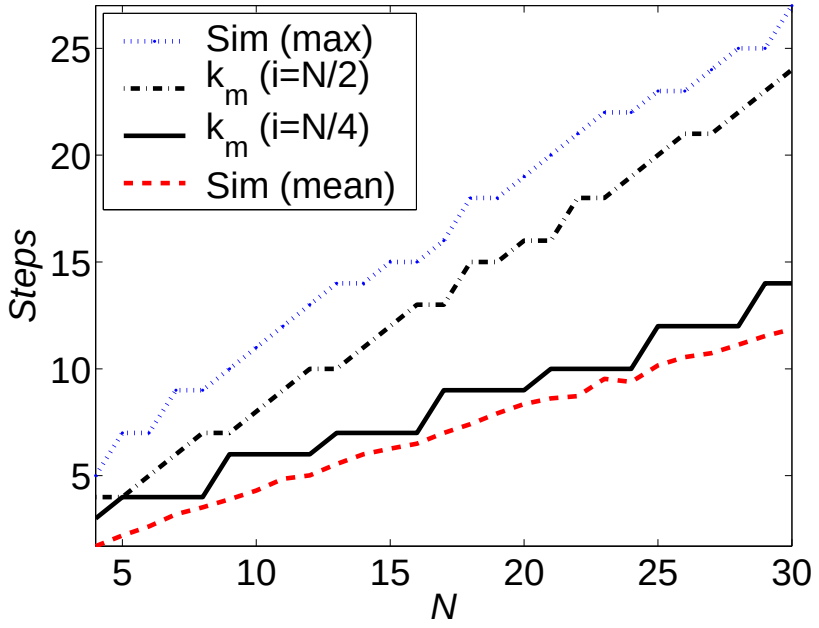


Figure 3.7: Expected number of steps in λ before HOPE generates a point in the basin of attraction of the global minimizer of f^1.

function, $h(x, \lambda)$:

(B1) h is periodic for fixed λ in x with period $x_u - x_l$.

(B2) All stationary points of h are isolated.

(B3) $\nabla_x^2 h(x, \lambda)$ is invertible for all $(x, \lambda) \in \mathbb{R} \times [0, 1]$.

It follows from (B2) and (B3) that there is a constant number of local minimizers, of h for $\lambda \in [0, 1]$. Let n be the number of local minimizers of h.

Figure 3.8 shows the plot of the equilibrium curves of $\nabla_x h$ for an example homotopy function satisfying (B1)–(B3). The solid curves contain local minimizers of h, and the dashed lines represent the boundaries of the basins of attraction of each of the local minimizer curves, as defined in Definition 2.

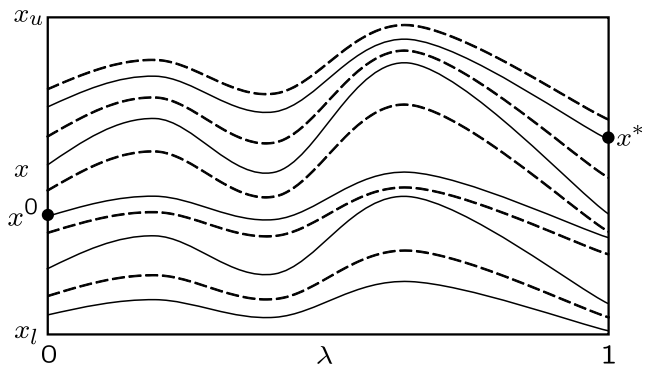


Figure 3.8: Equilibrium curves of $\nabla_x h$ and the basins of attraction of a periodic homotopy function h (with period $x_u - x_l$) whose stationary points with respect to x are isolated. The solid lines represent the equilibrium curves of local minimizers of h and the dashed lines represent the boundaries of the basins of attraction associated with those curves.

Since there are n curves of local minimizers of h for $\lambda \in [0,1]$, there are n distinct basins of attraction for $\lambda \in [0,1]$. Moreover, since the local minimizers are assumed to be isolated, $|\mathcal{R}_j^\dagger(\lambda)| > 0$ for $\lambda \in [0,1]$, $j = 1, \ldots, n$, where $\mathcal{R}_j^\dagger(\lambda)$ is the basin of attraction of the j^{th} local minimizer, $x^\dagger$.

We again view the basins of attraction as states in a Markov chain, where S_j is the state that corresponds to the basin of attraction of the j^{th} local minimizer of h. The states are assumed to be ordered such that S_n corresponds to the basin of attraction of x^*, the global minimizer of f^1. Figure 3.9 shows the enumeration of the states of the homotopy function from Figure 3.8.

We make the same assumptions about the parameterization of HOPE as were made in the previous section. The difference in the more general analysis is that the parent points in the BRW, i.e., the points in $\mathbb{S}_{pf}^{(k)}$, are not guaranteed to remain in the same state for all values of $\Delta\lambda$. However, for sufficiently large values of m we can assume that this is the case. See [5] for a detailed treatment

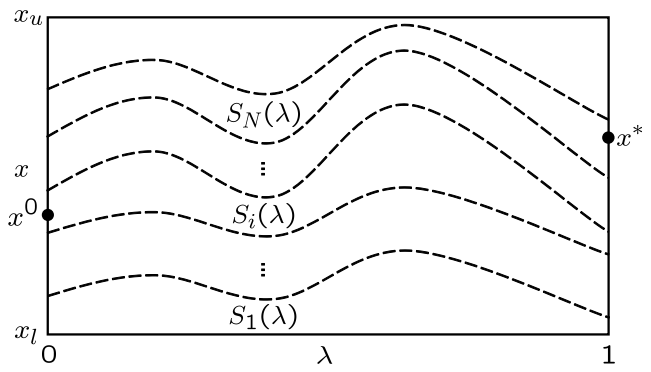


Figure 3.9: The states of the Markov chain associated with the basins of attraction of a periodic homotopy function h (with period $x_u - x_l$) whose stationary points with respect to x are isolated. States i and n correspond to the basins of attraction of local minimizers h containing the points x^0 and x^1*, respectively.

of the requirements on the step size in λ that will guarantee that such a condition is satisfied. Let M_h be the smallest value of m for which this condition is satisfied for a homotopy function, h, satisfying (B1)–(B2).

The analysis of the points in $\mathbb{S}_{rw}^{(k)}$ follows closely the one presented in the previous section. The difference is that a generic homotopy function is now used, which leads to a dependency on λ of the probabilities of transition from one state (basin of attraction) to another. For $\lambda \in [0, 1]$, we denote $P(\lambda)$ as the dynamic transition matrix of the BRW over the n states, $S_1, \ldots, S_n$. The entries of this matrix are determined as follows:

$$P_{i,j}(\lambda) = \Pr\left[\, \xi_\lambda(x) \in S_j : x \in S_i \,\right] \tag{3.32}$$

In the previous section, the number of points in S_N after k steps was determined by looking at an entry in the k^{th} power of the matrix P. However, in that analysis, the transition matrix was constant for all $\lambda \in [0, 1]$. To determine the number of steps to take in λ, we can replace the powers of P with the product

of $P(\lambda)$ for all values of λ used in HOPE. Thus, assuming that all points in $\mathbb{S}_{pf}^{(k)}$ remain in the same state in the next iteration of HOPE, the goal is to determine the smallest m such that

$$e_i^T \left(2^m \prod_{k=1}^{m} P(\lambda_k) \right) e_N = e_i^T \Big(2^m P(\lambda_1) P(\lambda_2) \cdots P(\lambda_m) \Big) e_N \geq 1 \ , \tag{3.33}$$

where $\lambda_k = k/m$. This can be computed for each instance of h. However, because of the dynamic nature of $P(\lambda)$ no general formula for computing such an m exists. The sets of matrices $\{P(\lambda)\}_{k=0}^{m}$ will in general be different for different values of m. Nevertheless, simulations similar to those for the N-modal sine functions in the previous section can be performed for each instance of h to be used in HOPE in order to compute a value for m. Note that the factor of 2 comes from $\hat{c} = 1$, since at each iteration in HOPE, there are $(\hat{c} + 1)$ times as many points in the ensemble as there were in the previous iteration. Thus, the factor of 2 in (3.33) should be replaced by $(\hat{c} + 1)$ when $\hat{c} \neq 1$.

We note that the value of m chosen to satisfy (3.33) must be greater than M_h or there will be no guarantee that the points in $\mathbb{S}_{pf}^{(k)}$ will behave according to this analysis above. However, as we show in the next chapter, in practice HOPE converges for relatively small values of m compared to those predicted by the analysis presented here. Therefore, we recommend such analysis only for determining loose upper bounds for m.

In this section, we have presented several frameworks for which HOPE can be analyzed. For this analysis to be useful for specific problems, information about the properties of the homotopy function must either be assumed or verified. We turn to numerical experiments involving HOM and HOPE in the next chapter. We will see that the performance of HOPE far exceeds that predicted by the analysis from this chapter.

Chapter 4

Numerical Experiments: Unconstrained Minimization

We apply HOPE and HOM to several standard test problems found in the unconstrained optimization literature. The purpose of these experiments is to highlight some of the advantages of using HOPE and HOM over local methods, and to show that HOPE is an effective method for solving general unconstrained minimization problems.

The discussion begins with information about the the methods being tested. Details of the functions that will be minimized will then be presented. Descriptions of the experiments and a discussion of the results are then presented.

4.1 Numerical Methods

Local minimization in HOPE and HOM was performed by a quasi-Newton method that uses a cubic line search and the BFGS update formula, which we denote `QNewton-BFGS`. In Matlab, `QNewton-BFGS` is implemented in the routine `fminusub` and is accessed from the unconstrained minimization driver, `fminunc`. Note that `QNewton-BFGS` requires only first derivative information.

Experiments using `QNewton-BFGS` by itself will be performed to provide a benchmark to which we will compare the results using HOPE and HOM. Thus, we

describe here the differences in the use of `QNewton-BFGS` on its own and in HOPE and HOM. `QNewton-BFGS` terminates when the change in function values between iterates drops below `TolFun`, the maximum change in any of the variables in x between iterates drops below `TolX`, the number of iterates reaches `MaxIter`, or the number of function evaluations reaches `MaxFunEval`. The default values used in the experiments in this chapter are `TolFun` $= 10^{-6}$, `TolX` $= 10^{-12}$, `MaxIter` $=$ 400, and `MaxFunEval` $= 800$. When `QNewton-BFGS` is used for minimization in HOPE and HOM, these defaults are used, with the exception of `MaxIter`. Since HOM makes m calls and HOPE makes at most $m \times c_{max}$ calls to `QNewton-BFGS` in each run of an experiment, `MaxIter` will be significantly lower than the default value shown above.

Two types of perturbations were used in the experiments testing HOPE. The first, denoted by ξ_{hr}, uses the Hit-and-Run algorithm (Section 2.2) with a uniform distribution of perturbation lengths between 0 and a fixed maximum perturbation length p_{max}. The other, denoted by ξ_{pct}, is a variant of Hit-and-Run where the maximum perturbation length is a percentage of the $\|x\|_2$, where x is the point being perturbed. The percentage is fixed throughout each run of an experiment and is denoted by $\tilde{p}_{max}$.

All of the experiments were run under Linux on a 2.5 GHz Intel Pentium 4 processor using Matlab 6.5 and the Optimization Toolbox 2.2 from Mathworks, Inc.

4.2 Test Problems

The problems used in testing HOPE and HOM include the N-modal sine functions designed for the analysis of HOPE, several examples from a set of test

problems often used in testing new minimization methods, and a problem that was designed to pose difficulties for global optimization methods that rely heavily on local methods.

All functions that are minimized in the experiments are denoted by $f^1(x)$ to emphasize their role as target functions in HOPE and HOM. Each function is given a label to distinguish it from the others, and all discussions will refer to functions and associated experiments by these labels.

4.2.1 N-Modal Sine Function (`Nmod`)

The N-modal sine function introduced in (3.18) will be used in the experiments presented in this chapter:

$$f^1(x) = \sin(x) + \sin(Nx) \ ,$$

where $f^1 : \mathbb{R} \rightarrow \mathbb{R}$ and $N \in \mathbb{Z}$. The label for this function is `Nmod`.

4.2.2 Moré, Garbow, and Hillstrom Test Functions

The problems in this section are a subset of the test functions in [78]. Using the starting points reported in that paper, `QNewton-BFGS` converged to the correct solution for all but 5 of the 35 test problems. Those 5 problems are presented below and were used in testing HOPE and HOM.

All of the functions in this set are the sums of squares of m functions of n variables:

$$f^1(x) = \sum_{i=1}^{m} \left[g_i(x)\right]^2 \ . \tag{4.1}$$

The dimensions of the problems in terms of m and n and a description of the functions g_i are given in the following sections. The starting points and global minimizers as reported in [78] are also presented for each problem.

The label for each problem is shown in the title of each problem section. A source reference for each problem is also provided.

Freudenstein and Roth Function (`Freu`) [43]

We spend some time discussing the background of this function, as it was designed to test a homotopy method called "Parameter-Perturbation Procedure" [43]. This procedure is a homotopy method for finding the roots of a system of equations by deforming the parameters of a derived set of equations. However, the authors did not refer to their procedure as a homotopy method and did not provide details of how to deform the parameters systematically. It is interesting to note that in the original description of the Parameter-Perturbation Procedure the parameters of the problem (polynomial coefficients) were deformed at different rates. We follow this same idea in the homotopy functions used for solving the protein structure prediction problems in Chapters 6–7.

Size of problem: $n = 2$, $m = 2$.

Functions used in $f^1(x)$*:*

$$g_1(x) = -13 + x_1 + ((5 - x_2)x_2 - 2)x^2 \ , \text{ and}$$
$$g_2(x) = -29 + x_1 + ((1 + x_2)x_2 - 14)x^2 \ .$$

Starting point: $x_0 = (0.5, -2)^T$.

Global minimum: $f^1(x^*) = 0$ at $x^* = (5, 4)^T$.

Jennrich and Sampson Function (`Jenn`) [61]

Size of problem: $n = 2$, $m = 10$.

Functions used in $f^1(x)$*:*

$$g_i(x) = 2 + 2i - (e^{ix_1} + e^{ix_2}) \ .$$

Starting point: $x_0 = (0.3, 0.4)^T$.

Global minimum: $f(x^*) \approx 124.3622$ at $x^* \approx (0.2578, 0.2578)^T$.

Meyer Function (`Mey`) [76]

Size of problem: $n = 3, m = 16$.

Functions used in $f^1(x)$*:*

$$g_i(x) = x_1 e^{\left(\frac{x_2}{t_i + x_3}\right)} - y_i \ ,$$

Function parameters: $t_i = 45 + 5i$ and

i	y_i	i	y_i
1	34780	9	8261
2	28610	10	7030
3	23650	11	6005
4	19630	12	5147
5	16370	13	4427
6	13720	14	3820
7	11540	15	3307
8	9744	16	2872

Starting point: $x_0 = (0.02, 4000, 250)^T$.

Global minimum: $f^1(x^*) \approx 87.9459$ at $x^* \approx (0.0056, 6181.3464, 345.2236)^T$.

Biggs EXP6 Function (`Be6`) [12]

Size of problem: $n = 6, m = 13$.

Functions used in $f^1(x)$*:*

$$g_i(x) = x_3 e^{-t_i x_1} - x_4 e^{t_i x_2} + x_6 e^{t_i x_5} - y_i \ .$$

Function parameters: $t_i = 0.1i$, $y_i = e^{-t_i} - 5e^{-10t_i} + 3e^{-4t_i}$.

Starting point: $x_0 = (1, 2, 1, 1, 1, 1)^T$.

Global minimum: $f^1(x^*) = 0$ at $x^* = (1, 10, 1, 5, 4, 3)^T$.

Trigonometric Function (`Trig`) [108]

Size of problem: $n = 10, m = 10$.

Functions used in $f^1(x)$*:*

$$g_i(x) = n - \sum_{j=1}^{n} \cos x_j + i(1 - \cos x_i) - \sin x_i \ .$$

Starting point: $x_0 = 1/n * (1, 1, 1, 1, 1, 1, 1, 1, 1, 1)^T$.

Global minimum: $f^1(x^*) = 0$ at $x^* \approx (0.0430, 0.0440, 0.0451, 0.0463, 0.0477,$
$0.0494, 0.0512, 0.1952, 0.1650, 0.0601)^T$.

4.2.3 Pintér Test Function (`Pint`) [91]

In [91], Pintér advocates using "randomized test functions that will have a randomly selected unique global solution" in testing global optimization methods. He argues that this reduces the ability to tune a method's performance to the test problems. We follow this advice and test HOPE and HOM on the functions described by him in that work. The general form of the test problem is

$$f(x) = s \sum_{i=1}^{n} (x_i - x_i^*)^2 + \sum_{k=1}^{k_{max}} a_k \sin^2[f_k P_k (x - x^*)] \ , \tag{4.2}$$

where x^* is the unique global solution, $s > 0$ is a scaling factor, $a_k > 0$ are amplitude scaling factors, f_k are (integer) frequency multipliers, and $P_k(\cdot)$ are polynomial noise terms that vanish at the zero vector. Note that this defines a class of functions where each instance is specified by the choice of x^*. Thus, we can create a set of random test functions by randomly choosing the elements of x^*. The label for these functions is `Pint`.

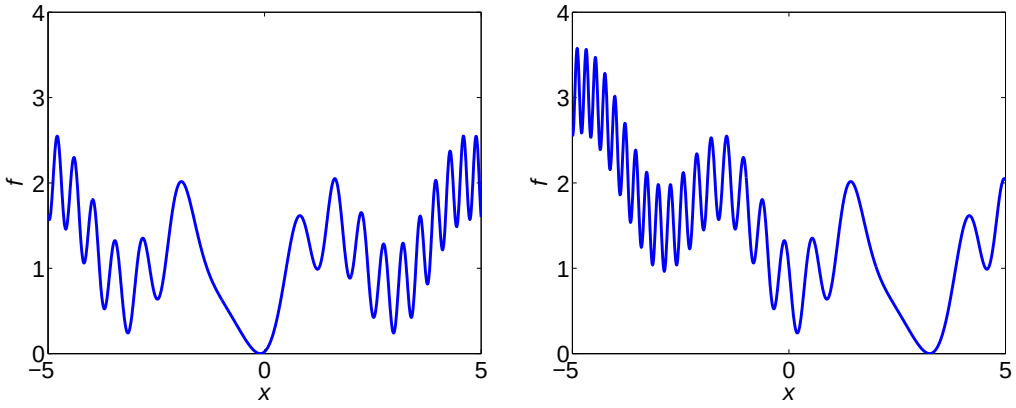
Values of the parameters and the functions in $f(x)$ used in the experiments are defined in Table 4.1. Note that the elements of the global solutions, x_i^* $(i = 1, \ldots, n)$, for each problem instance were chosen from a uniform distribution on $[-5, 5]$.

Parameter	**Value**
s	$0.025n$
k_{max}	2
a_k	1
f_k	1
$P_1(x - x^*)$	$\sum_{i=1}^{n}(x_i - x_i^*) + \sum_{i=1}^{n}(x_i - x_i^*)^2$
$P_2(x - x^*)$	$\sum_{i=1}^{n}(x_i - x_i^*)$

Table 4.1: Parameters and functions used to define the `Pint` functions.

As an illustration of these functions, two functions in one dimension are presented in Figure 4.1 with unique global minimizers at (a) $x^* = -0.1$ and (b) $x^* = 3.25$. Also, surface and contour plots of a function in two dimension with $x^* = [-4.42, -1.47]^T$ are shown in Figures 4.2 and 4.3. The functions have the property that the region of attraction of the global minimizer is relatively

large compared to the basins of other local minimizers. Also, far from the global minimizers, high frequency oscillation terms create local minimizers with deep narrow basins of attraction. Minimization methods that rely on local methods for searching for a global minimizer will likely converge to a local minimizer unless the method starts at a good approximation of the solution. Thus, this class of functions will be used to demonstrate the value of using HOPE and HOM over a local method.



(a) (b)

Figure 4.1: Plots of `Pint` functions in one dimension with (a) $x^* = -0.1$ and (b) $x^* = 3.25$.

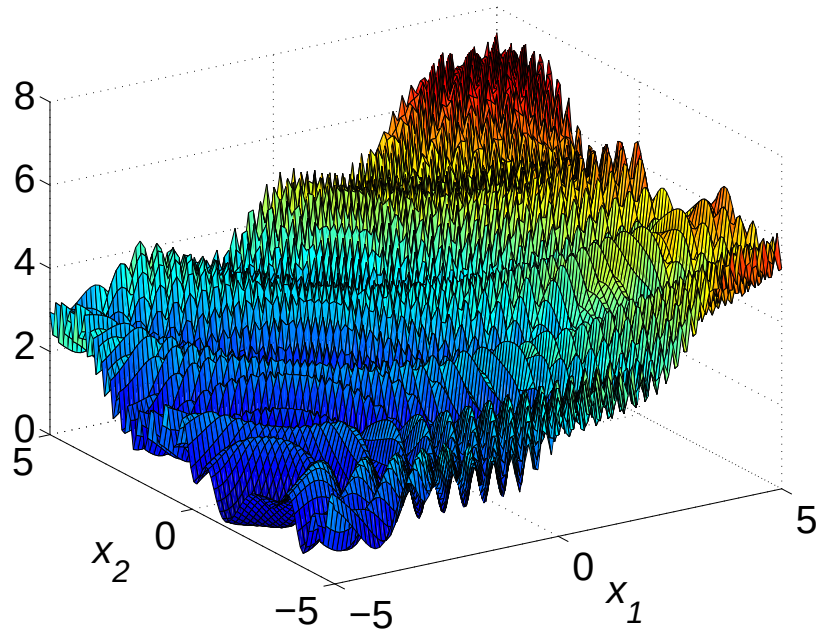


Figure 4.2: Surface of a `Pint` function with $x^* = [-4.42, -1.47]^T$.

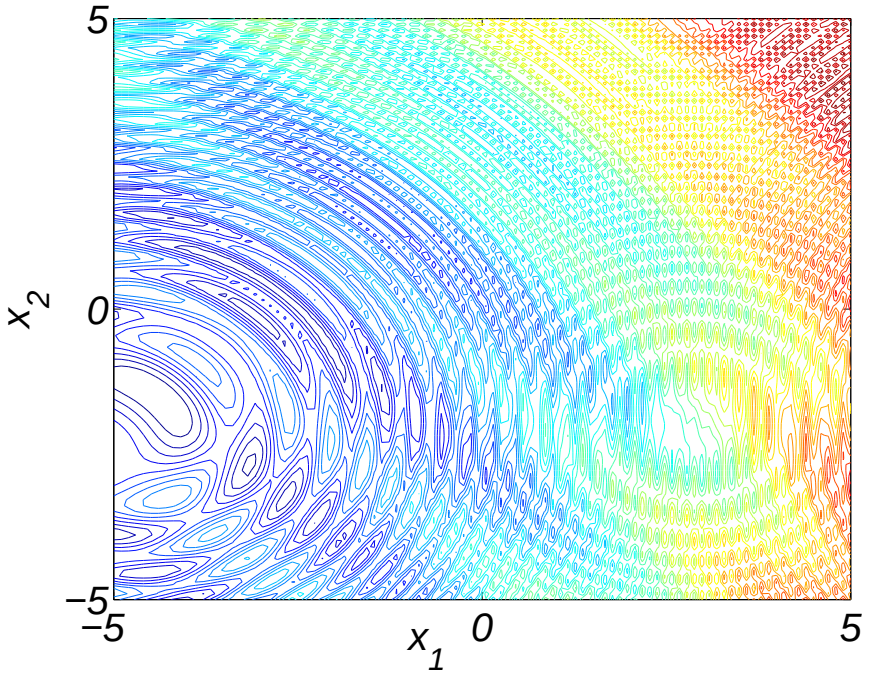


Figure 4.3: Contours of a `Pint` function $x^* = [-4.42, -1.47]^T$.

4.3 Results

4.3.1 N-Modal Sine Function

Several experiments were performed on the `Nmod` functions using two different homotopy functions. The purpose of the experiments was to illustrate the influence of the homotopy function on the performance of HOPE and HOM. The two homotopies used have the same general (convex homotopy) form:

$$h(x, \lambda) = (1 - \lambda) f^0(x) + \lambda f^1(x) \tag{4.3}$$

but differ in the function f^0 used. One homotopy function uses the N-modal sine function introduced in (3.17):

$$f^0(x) = -\sin(x) + \sin(Nx) \ .$$

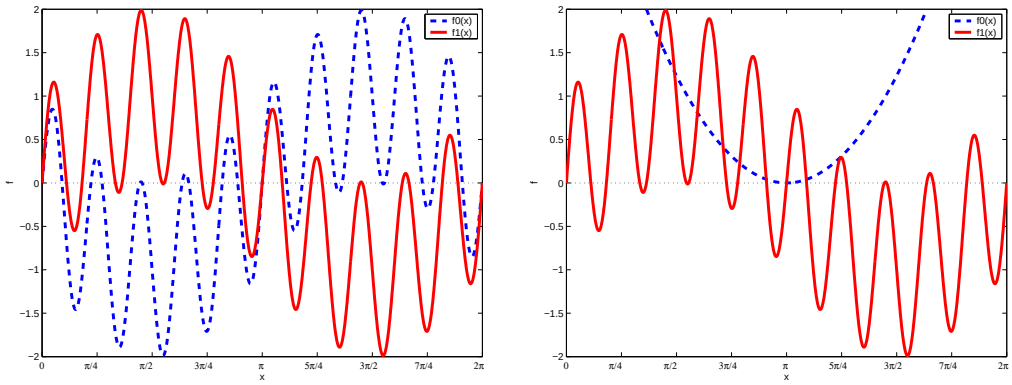
The other homotopy function uses the quadratic function

$$f^0(x) = \frac{1}{2}(x - \pi)^2 \tag{4.4}$$

where $f^0 : \mathbb{R} \to \mathbb{R}$. Note that the quadratic f^0 differs from f^0 in (3.17) in two important ways: it is not periodic and it is strictly convex.

Figure 4.4 shows plots of the two f^0 functions and f^1 with $N = 10$. Figure 4.5 shows the equilibrium curves of $\nabla_x h$ (on the left) and the curves of local minimizers of $\nabla_x h$ (on the right) for the different homotopy functions used in the experiments. Recall that the curves of local minimizers are a subset of the equilibrium curves.

Two features of the quadratic f^0 make it useful for testing HOPE: the function is strictly convex and thus has a unique global minimizer, and determining



(a) $f^0(x) = -\sin(x) + \sin(Nx)$ (b) $f^0(x) = \frac{1}{2}(x-\pi)^2$

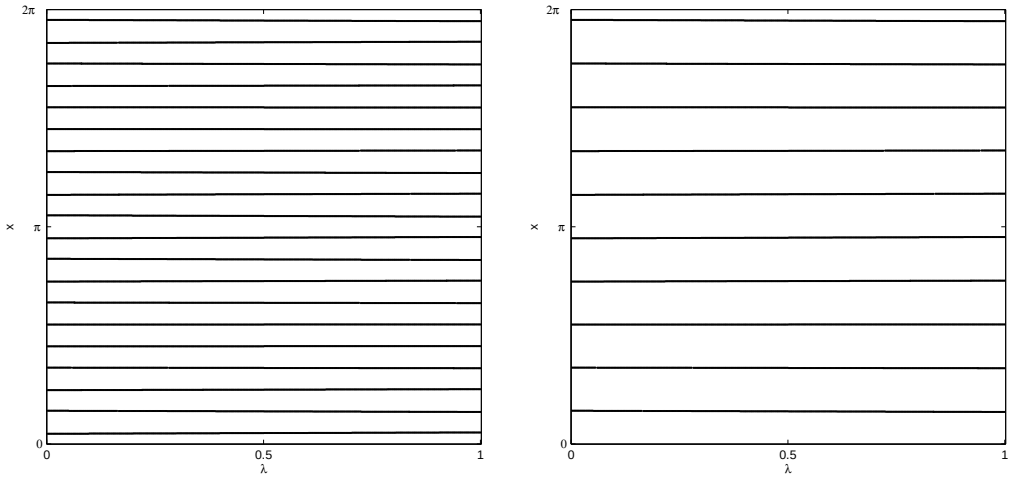
Figure 4.4: Plots of f^0 and f^1 for the convex homotopies used in the N-modal sine function experiments.

the global minimizer and derivatives of the function does not require any computation. Specifically, $x^* = \pi$ and $\nabla f^0(x) = x - \pi$.

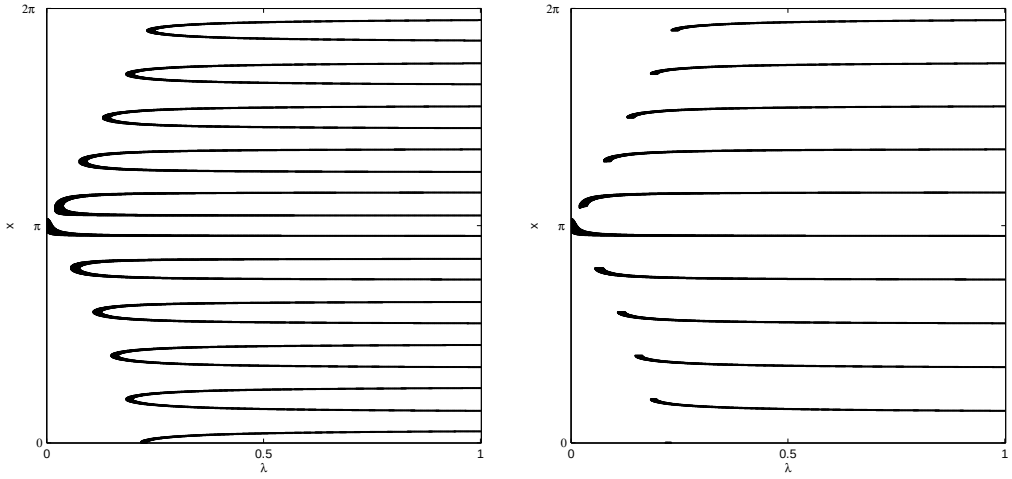
The experiments consisted of 1000 runs of HOPE using the following parameters: $\hat{c} = 1$, $c_{max} = 8$, and $m = kN/5, k = 1, \ldots, 5$. `QNewton-BFGS` was used for performing local minimization in HOPE and `MaxIter` $= 10$ for all values of m.

Figure 4.6 shows plots of the percentage of successful runs versus m, the number of steps in λ, for $N = 10, 20, 30, 40, 50, 60$. The solid and dashed lines in the figure are the results using the N-modal sine and quadratic f^0 functions, respectively. For smaller values of N, there is little difference in the performance of HOPE using the different homotopies. As N increases, there is a definite advantage in using the quadratic over the N-modal sine function for f^0.

These results suggest that the choice of homotopy affects the performance of HOPE. In these experiments, the homotopy that led to better performance was the one with the quadratic f^0, which has little connection to the f^1 in terms of location and number of local minimizers. For some problems, though, such a generic homotopy may not always perform best. The goal, therefore, is to balance



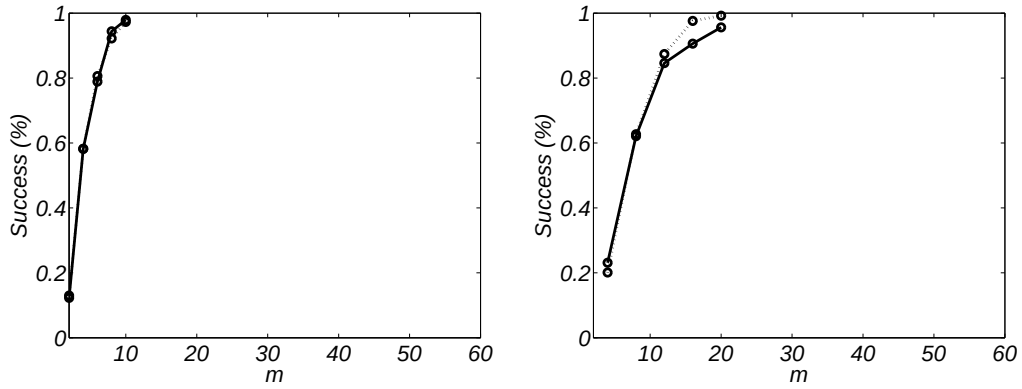
(a) $h(x,0) = -\sin(x) + \sin(Nx)$



(b) $h(x,0) = \frac{1}{2}(x-\pi)^2$

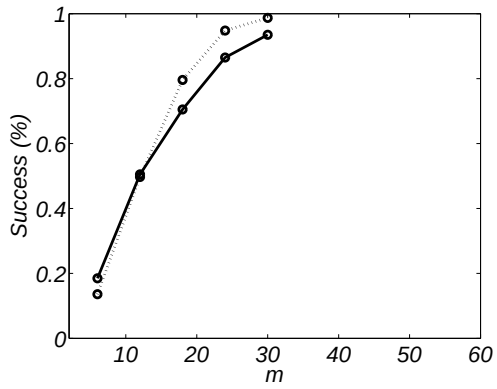
Figure 4.5: Equilibrium curves $\nabla_x h$ for the different homotopy functions used in the N-modal sine function experiments. All equilibrium curves are shown on the left, and only the curves of minimizers are shown on the right.

the amount of effort required in creating a customized homotopy function with the increase in performance gained in using such a homotopy in HOPE.

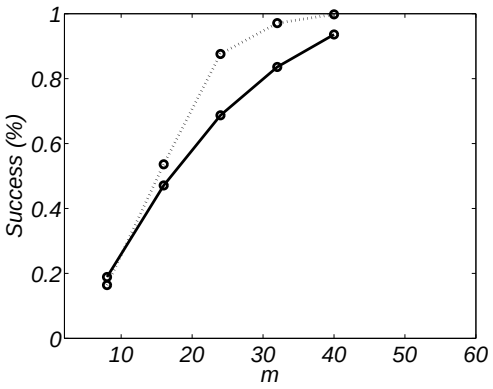


(a) $N = 10$

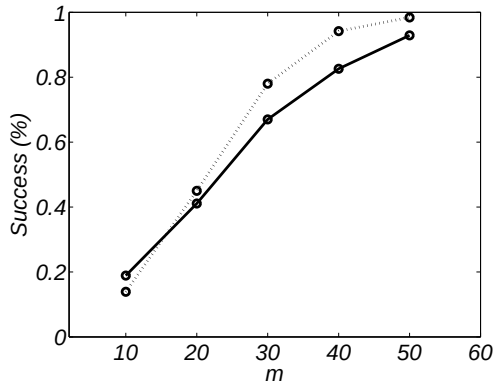
(b) $N = 20$



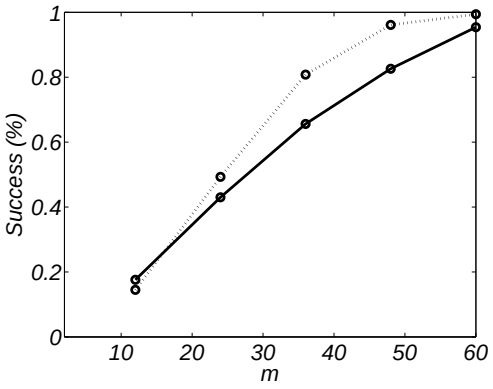
(c) $N = 30$



(d) $N = 40$



(e) $N = 50$



(f) $N = 60$

Figure 4.6: Results of 1000 runs of HOPE applied to the N-modal sine function f^1 using convex homotopies with $f^0(x) = -\sin(x) + \sin(Nx)$ (solid lines) and $f^0(x) = \frac{1}{2}(x - \pi)^2$ (dashed lines).

4.3.2 Moré, Garbow, and Hillstrom Test Functions

In this section, we present the results of experiments using HOPE, HOM, and `QNewton-BFGS` to minimize the `Freu`, `Jenn`, `Mey`, `Be6`, and `Trig` functions. The results of the first set of experiments shows that HOPE outperforms both HOM and `QNewton-BFGS`. In the second set of experiments, HOPE is used to minimize the `Freu` function using combinations of different parameters. The results of these experiments illustrate the interplay between the parameters used in HOPE and their effect on performance.

The homotopy function used in HOPE and HOM was

$$h(x, \lambda) = (1 - \lambda)f^0(x) + \lambda f^1(x) \tag{4.5}$$

where the target function, f^1, is the function being minimized, and the template function is

$$f^0(x) = \frac{1}{2}(x - x^0)^T(x - x^0) \ . \tag{4.6}$$

This homotopy function is often associated with probability-one homotopy methods used to solve optimization problems [117, 119]. It is a generic homotopy, however, and does not take advantage of any structure inherent in the target functions, since x^0 is chosen arbitrarily.

Experiment 1. Table 4.2 presents the results of minimizing these functions using HOPE, HOM, and `QNewton-BFGS`. The default parameters were used for `QNewton-BFGS` when run on its own. In HOPE and HOM, `QNewton-BFGS` was used for local minimization with `MaxIter` = 20. The first two columns show the function name and value of the global minimum, $f^1(x^*)$. The third column shows the function value at the point x^1 output by `QNewton-BFGS`; recall that

for these problems `QNewton-BFGS` was not able to find a global minimizer. The points produced by `QNewton-BFGS` for `Be6` and `Trig` have function values close to the global minimum. However, these are documented local minima [67].

For the experiments using HOM to minimize these functions, values of $m = 1, \ldots, 100$ were used. The goal in using a range of values of m was to demonstrate the increase in success of using HOM over `QNewton-BFGS` and the extent that more steps in λ affects this increase. However, the results of these experiments show only marginal improvement. Moreover, the small increase in performance came at a disproportionately large cost in computation. Columns 4 and 5 in the table show the lowest function value attained for one of the minimizers generated using HOM and the value of m at which that value was attained (with "—" signifying that no improvement was made in using HOM over `QNewton-BFGS` for any value of $m = 1, \ldots, 100$).

HOM correctly predicted only one global minimizer (`Jenn`) and only one other local minimizer (`Mey`) with a significantly lower function value than the one found using `QNewton-BFGS`. In the latter case, the improvement was dramatic, as measured by relative function value error:

$$relerr_{f^1} = \frac{|f^1(x^1) - f^1(x^*)|}{|f^1(x^*)|} \ . \tag{4.7}$$

For the `Mey` function, HOM produced a local minimizer many orders of magnitude better in relative function error than the one generated using `QNewton-BFGS` ($relerr_{f^1} \approx 6.62 \times 10^{-1}$ for HOM versus $relerr_{f^1} \approx 5.73 \times 10^{7}$ for `QNewton-BFGS`). However, this improvement came at the cost of $m = 81$ steps in the homotopy parameter and a correspondingly large number of function evaluations: a total of 7675 function evaluations for HOM versus 45 for `QNewton-BFGS` (almost 171 times as many). In general, such an increase in the number of function evaluations may

		QNewton-BFGS	HOM		HOPE	
	$\mathbf{f^1(x^*)}$	$\mathbf{f^1(x^1)}$	$\mathbf{f^1(x^1)}$	**m**	$\mathbf{f^1(x^1)}$	**m**
Freu	0	48.98	48.98	—	0	9
Jenn	124.36	2020	124.36	2	124.36	2
Mey	87.95	5.0×10^{8}	146.18	81	87.95	3†
Be6	0	5.7×10^{-5}	5.7×10^{-5}	—	10^{-14}	1
Trig	0	2.8×10^{-5}	2.8×10^{-5}	—	10^{-14}	5

†*For $1 \leq m \leq 10$, there was only a single successful run, and it was when $m = 3$.*

Table 4.2: Results of QNewton-BFGS HOM, and HOPE applied to Moré test functions. The lowest function values found and the fewest number of steps in λ (HOM and HOPE) to find the corresponding minimizer are presented.

not produce as significant an improvement.

The results of experiments using HOPE to minimize these functions are presented in the remaining columns of the table. In the experiments using HOPE, $\hat{c} = 1$ and $c_{max} = 2^m$, and perturbed versions were generated using ξ_{hr} with $p_{max} = 10^{-3}$ (i.e., very local perturbations). The column labelled $\mathbf{f^1(x^1)}$ shows the lowest function value attained in the experiments using HOPE, where 10 runs were performed for each of the values $m = 1, \ldots, 10$. Since no points were discarded from the ensembles in HOPE, the runs at $m = 10$ require a considerable amount of computation compared to those for HOM and QNewton-BFGS. The last column shows the lowest value of m ($1 \leq m \leq 10$) for which all 10 runs were successful (i.e., 100% success rate). The best results were for Be6 and Jenn, where only 1 and 2 steps, respectively, were required. Compared to the results for these problems where one fewer step in λ was taken, the increases in success rates were dramatic (0% at $m = 0$ for Be6 and 30% at $m = 1$ for Jenn). This suggests

that for these problems the perturbations were most responsible for the success of HOPE. For `Trig` and `Freu`, more steps in λ ($m = 5$ and $m = 9$, respectively) were required before achieving a perfect set of runs. Furthermore, the increases in success rates were more gradual for these two problems (80% successes for `Trig` at $m = 4$ and for `Freu`at $m = 8$), suggesting that the performance of HOPE depends on more than perturbations alone. However, it is unclear from these few examples which parameter, if any, plays the most important role in determining the success of HOPE.

For `Mey`, HOPE was not able to predict the global minimizer for all 10 runs for any value $m \leq 10$. The only successful result is shown in the table, where the global minimizer was found in 1 of the 10 runs for $m = 3$ steps in λ. For `Mey`, there are several orders of magnitude difference in x_1^* and x_2^*, the first two elements of the global minimizer (see the problem description in Section 4.2.2). The amount of perturbation ($p_{max} = 10^{-3}$) used in these experiments was not enough to lead HOPE to success for $m \leq 10$. In followup experiments using larger perturbations ($p_{max} = 100$), though, HOPE was successful in finding the global minimizer of `Mey` in 100% of the runs for each $m = 1, \ldots, 10$.

We conclude that for these problems HOPE outperformed `QNewton-BFGS` and HOM in terms of successfully finding the global minimizer of a function. Moreover, the results suggest that a small amount of perturbation can dramatically increase the performance of HOPE over HOM. In the next experiment, however, we see that when we limit the ensemble size, a larger amount of perturbation may be required to produce comparable results.

Experiment 2. In the second set of experiments we applied HOPE to the `Freu` problem using various amounts of perturbation, maximum ensemble sizes,

and numbers of steps in λ to illustrate the impact of the algorithm parameters on performance. We chose to focus on the `Freu` function because there is a relatively high function barrier between the global minimizer and the standard starting point that is used. Moreover, there is a local minimizer on the same side of the barrier as the standard starting point, and this is the point to which most local methods (and many global methods) converge. Thus, the goal of this experiment is to demonstrate the effects of parameter choices in HOPE on solving problems where the path to the solution is uphill and very steep.

`QNewton-BFGS` was used for local minimization with `MaxIter` $= 60$. Perturbed versions of ensemble members were generated using ξ_{hr} and $\hat{c} = 1$ in these experiments. A total of 100 runs were performed for each combination of the following parameter values: $p_{max} = 1, 2, 4, 8$; $c_{max} = 2, 4, 8, 16$; and $m = 1, 2, 4, 8$.

Table 4.3 presents the results of these experiments. The first two columns show the amount of perturbation and the maximum ensemble size. The next four columns show the number of runs where HOPE correctly predicted the global minimizer for $m = 1, 2, 4, 8$, respectively. The last four columns show the ratio of successful runs to total number of calls to `QNewton-BFGS` over the 100 runs for $m = 1, 2, 4, 8$, respectively.

The general trend of these results show that as the amount of perturbation and the amount of computational effort increases (as controlled by c_{max} and m) the chances of correctly predicting the global minimizer increases as well. However, the amount of perturbation appears to be the most important parameter affecting the success of HOPE. This was expected for the `Freu` problem since there is a relatively high barrier between the standard starting point and the global minimizer that prevents many methods from converging to the correct so-

$\mathbf{p_{max}}$	$\mathbf{c_{max}}$	successes when m =				successes/min when m =			
		1	**2**	**4**	**8**	**1**	**2**	**4**	**8**
1	2	7	1	1	3	0.0233	0.0014	0.0008	0.0012
	4	5	0	1	3	0.0167	0	0.0006	0.0009
	8	6	1	4	8	0.0200	0.0014	0.0013	0.0013
	16	8	0	5	8	0.0267	0	0.0016	0.0010
2	2	12	13	24	36	0.0400	0.0186	0.0185	0.0144
	4	7	19	32	41	0.0233	0.0271	0.0178	0.0111
	8	9	24	54	65	0.0300	0.0343	0.0174	0.0105
	16	4	23	56	72	0.0133	0.0329	0.0181	0.0083
4	2	9	27	48	60	0.0300	0.0386	0.0343	0.0222
	4	5	27	52	74	0.0167	0.0386	0.0289	0.0185
	8	7	25	75	95	0.0233	0.0357	0.0242	0.0151
	16	8	21	77	98	0.0267	0.0300	0.0248	0.0107
8	2	22	48	73	95	0.0733	0.0686	0.0521	0.0328
	4	15	49	79	98	0.0500	0.0700	0.0416	0.0228
	8	20	54	94	100	0.0667	0.0771	0.0303	0.0159
	16	17	59	95	100	0.0567	0.0843	0.0306	0.0106

Table 4.3: Results of HOPE applied to `Freu` for different amounts of perturbation, maximum ensemble sizes and steps in λ.

lution. Many methods converge to a local minimizer located on the same side of the barrier as the starting point (as was the case for `QNewton-BFGS` and for HOM in the experiments above). Increasing the amount of perturbation may not

always be the most effective use of resources. In cases where f^0 and f^1 are closely related (e.g., in terms of location of their global minimizers), less perturbation may lead to better results, with larger perturbations unnecessarily searching the domain in unpromising areas.

The ratios of success to effort (last four columns) show that effective use of computational resources may prove challenging when using HOPE. For example, for $p_{max} = 8$ and $m = 8$, the number of successful runs was $95, 98, 100, 100$ for $c_{max} = 2, 4, 8, 16$, respectively, suggesting that larger ensemble sizes lead to more successful runs. An important question then is whether this increase in success justifies the corresponding increase in the amount of computational effort. In this case, there is a downward trend in the corresponding ratios: $0.0328, 0.0228, 0.0159, 0.0106$. For $p_{max} = 8$ and $c_{max} = 16$, the number of successful runs was $17, 59, 95, 100$ for $m = 1, 2, 4, 8$, respectively. The corresponding ratios are $0.0567, 0.0843, 0.0306, 0.0106$, suggesting another downward trend for values of $m > 1$. In general, it is unclear whether larger ensembles with fewer homotopy steps or smaller ensembles with more homotopy steps will lead to better results.

Another presentation of the results for $p_{max} = 8$ is given in Figure 4.7, where the number of successful predictions of the global minimizer is plotted against the average number of calls to `QNewton-BFGS`, the average total number of iterations of `QNewton-BFGS` for all steps in λ, and the average number of function evaluations for each run. The four points for the different number of steps in λ correspond to the results for $c_{max} = 1, 2, 4, 8$. Again, the general trend is that HOPE is more successful when more steps in λ are taken. Also, as the maximum ensemble size increases more computation was performed and this led to slightly better results.

However, the increase in the amount of computation did not lead to significant increases in the number of successful predictions. This suggests that the number of steps in λ may impact performance of HOPE more than ensemble size.

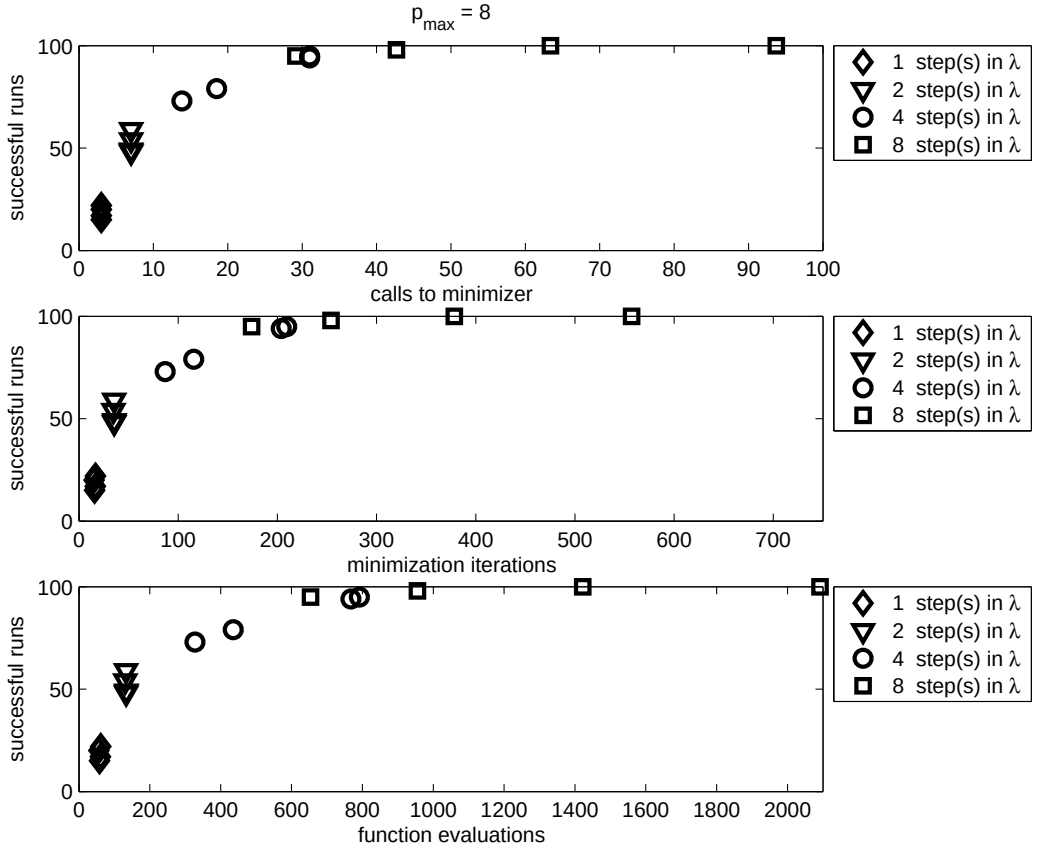


Figure 4.7: Results of HOPE applied to `Freu` for $p_{max} = 8$. The four markers for each value of m correspond to the four values of $c_{max} = 2, 4, 8, 16$.

4.3.3 Pintér Test Function

In these experiments, we focused on the the interplay between the number of steps taken in λ and the dimension of the function being minimized. We applied HOPE to `Pint` functions of dimension $n = 1, \ldots, 10$. The `Pint` function and

corresponding homotopy function for $n = 10$ is defined using

$$x^* = (-3.0173, -4.4483, \quad 4.6930, -4.7538, \quad 1.5104, \\ -3.9100, -4.3961, -1.4326, -0.3789, \quad 1.4885)^T \tag{4.8}$$

and

$$x^0 = (\quad 1.4127, \quad 4.3035, -4.1816, -0.8379, \quad 3.5322, \\ 3.1757, \quad 2.9291, \quad 0.1542, \quad 3.2336, \quad 3.0290)^T . \tag{4.9}$$

These values were samples from a uniform distribution on $[-5, 5]$. The problems with $n < 10$ are defined using the first n elements of each of x^* and x^0 above.

For each value of n, `QNewton-BFGS` was run using default parameters and starting from 30,000 random points. The elements of each of these starting points were chosen from a uniform distribution on $[-5, 5]$. HOPE was run 10 times using `QNewton-BFGS` for local minimization with `MaxIter` $= 10$; the HOPE parameters used were $\hat{c} = 1$, $c_{max} = 8$, and $m = 1, 2, 4, 8$. Perturbed versions of ensemble points in HOPE were generated using ξ_{pct} with a maximum of 10% perturbation ($\tilde{p}_{max} = 0.10$).

Table 4.4 presents the results of these experiments. Column 1 shows the dimension of the problem and column 2 shows percentage of the runs where `QNewton-BFGS` successfully predicted the global minimizer. `QNewton-BFGS` was most successful for the problem with $n = 1$, but its performance seems to depend on more than just the dimension of the problem. As n increases the performance of `QNewton-BFGS` is fairly constant for these problems. The last four columns of the table show the percentage of the 10 runs where HOPE successfully predicted the global minimizer for $m = 1, 2, 4, 8$, respectively. For all n, the success of

HOPE increased monotonically with m, and for all $n > 1$ all of the runs predicted the global minimizer when $m = 8$.

		HOPE			
	`QNewton-BFGS`	**success (%), when m =**			
n	**success (%)**	**1**	**2**	**4**	**8**
1	29	0	0	10	50
2	9	0	10	40	100
3	9	0	30	100	100
4	10	30	20	60	100
5	12	0	20	100	100
6	12	20	40	100	100
7	11	20	20	100	100
8	12	0	50	70	100
9	9	10	10	100	100
10	8	0	20	100	100

Table 4.4: Results of `QNewton-BFGS` and HOPE applied to `Pint` problems of dimensions $n = 1, \ldots, 10$ using ξ_{pct} for perturbations with $\tilde{p}_{max} = 0.10$.

These results further illustrate the use of HOPE in extending local minimization methods for use in finding global minimizers. Surprisingly, the most difficult problem for HOPE was the one with $n = 1$, the same problem for which `QNewton-BFGS` had most success. This is an indication that perturbation alone may lead to a degradation of performance of a local method extended using HOPE. However, with more steps in the homotopy, and thus a more gradual deformation of f^0 into f^1, HOPE eventually outperforms the local method: for

$n = 1$ and $m = 8$, the success rate of HOPE is 50% compared to 29% for `QNewton-BFGS`.

In another set of experiments, `QNewton-BFGS`, HOM, and HOPE were used to solve the `Pint` function with $n = 100$ defined in [91]. For these methods, 100 runs were performed starting at random points (chosen as for `QNewton-BFGS` in the experiments above). `QNewton-BFGS` and HOPE used the same parameters as above, except for $c_{max} = 4$. HOM was run using the same parameters as HOPE.

Table 4.5 presents the results of these experiments. The columns of the table show the method used, number of steps in λ (m), percentage of successful runs, average number of function evaluations per run ($\overline{N}_f$), ratio of successes to average number of function evaluations, and average function value of the predicted points ($\overline{f^1(x^1)}$), respectively. The results present further evidence that HOPE outperforms HOM and that HOM outperforms `QNewton-BFGS`. Furthermore, the ratio of successes to average number of function evaluations allows for comparison of the three methods. When this ratio is approximately 0.6, HOPE is about twice as effective as the other methods (HOPE has 44 successes when the ratio is 0.62, HOM has 22 when the ratio is 0.60, and `QNewton-BFGS` has 20 when the ratio is 0.57). Results when the ratios are around 0.36 and 0.15 for HOPE and HOM illustrate similar increases—when the ratio is 0.36, HOPE has 70 successes to HOM's 43 (approximately 1.63 times more); and when the ratios are around 0.15, HOPE has 94 successes to HOM's 36 (approximately 2.61 times more).

4.4 Summary

In this chapter, we have demonstrated that HOPE and HOM are more effective than a quasi-Newton method in solving general unconstrained minimization

Method	m	Success (%)	$\overline{\mathbf{N_f}}$	Success per $\overline{\mathbf{N_f}}$	$\overline{\mathbf{f^1(x^1)}}$
`QNewton-BFGS`	0	20	35	.57	1.431
HOM	1	22	37	.60	1.028
	2	30	64	.47	1.021
	4	43	119	.36	0.283
	8	36	227	.16	0.221
HOPE	1	44	71	.62	0.436
	2	70	196	.36	0.085
	4	94	695	.14	0.012
	8	98	1739	.06	10^{-11}

Table 4.5: Results of `QNewton-BFGS`, HOM, and HOPE applied to the `Pint` function with $n = 100$.

problems. Results of several experiments suggest that as more steps in λ are taken and larger perturbations are used, the performance of HOPE improves. By taking more steps in λ, the template function is deformed more gradually into the target function. Such gradual change may be necessary for some problems where the template and target functions behave very differently. The use of perturbations allows searching of the function domain in areas that may not be reachable by following curves of minimizers of the homotopy function. We suspect that larger perturbations will be more useful when little is known about the relationship between the template and target functions and a generic homotopy function is used.

In the next few chapters we will see that time and effort spent in design-

ing homotopy functions that can take advantage of the problem structure and relationships between the template and target function leads to significant performance increases. In these upcoming chapters, HOPE is applied to the protein folding problem, a problem with a great deal of structure.

Chapter 5

Protein Structure Prediction

A central challenge in biochemistry today is the development of reliable, efficient computational methods for solving the *protein folding problem*, the problem of predicting the tertiary structure (three-dimensional conformation) of a protein in its native state given its primary structure (amino acid sequence). Several surveys and collections are available that contain information about the history of the protein folding problem and methods that have been developed for solving this problem [35, 40, 42, 83, 87]. Readers interested in the energetic interactions involved in the folding of a protein should consult [22, 101].

5.1 Background

Milestones in experimental research in protein structure include the sequencing of insulin [96, 97], determination of the structure of myoglobin via X-ray crystallography [65], and determination that the *native conformation* of ribonuclease, i.e. the shape in which it performs its function properly, is the one in which the Gibbs free energy [1] is lowest [8]. The results of these experiments led researchers

[1]The Gibbs free energy of a system is the maximum amount of work it can do at a constant temperature in a constant volume.

to embark on using computation and simulation to predict the native conformation of a protein from its amino acid sequence. After four decades of such work by biologists, chemists, mathematicians, statisticians, and computer scientists, the goal of those first computations—to solve the protein folding problem—still eludes researchers, despite the prominence and importance of the problem in the field of computational biology.

Computational methods for solving the protein folding problem fall into three categories:

- *Molecular dynamics* (MD) simulations concentrate on the force balance of the atoms within a protein and the resulting (Langevin) dynamics, which are approximated by a stochastic differential equation [50]. The main drawback of using MD for protein structure prediction is that the time steps required for accurate simulations are many orders of magnitude smaller than the time scale on which a protein folds into its native conformation.

- *Bioinformatics* algorithms use experimental structure data to predict conformations for which no experimental results exist. Comparative modeling methods (e.g., threading, homology modeling) [37] have been the most successful and widely implemented of the bioinformatics approaches for protein structure prediction to date. These methods determine the most probable native (found in nature) conformation by statistically matching the sequence of a protein to that of one or more template proteins—proteins whose native conformations have been determined experimentally.

- *Energy minimization* methods are global optimization methods designed to predict the native conformation of a protein by minimizing an energy

function associated with that protein.

We concentrate in this thesis on energy minimization. In this chapter we consider the energy models and the minimization methods that have been applied to them.

5.2 Protein Structure

Proteins, or peptides, are molecules consisting of amino acids (aa) bonded together. Amino acids are organic compounds consisting of carbon (C), nitrogen (N), oxygen (O), hydrogen (H), and sulphur (S) atoms. Figure 5.1 depicts the topology of a single amino acid, where lines denote chemical bonds between atoms (two lines denote a double bond) and the symbol **R** denotes the residue, or side chain, of the amino acid. To distinguish the two carbon atoms, the one to which the residue is bonded is given the subscript α. The dashed lines in the figure show a bond to another amino acid (or H at the N-terminus or OH at the C-terminus). There are 20 amino acids and they differ in the type and number $(1 - 18)$ of atoms that comprise the residue. Smaller proteins are sometimes referred to as *oligopeptides* (< 10 aa) or *polypeptides* ($\sim 10 - 100$ aa).

The *backbone* of an amino acid refers to the linear sequence of bonded atoms N–C_α–C, and the backbone of a protein refers to the full sequence of its amino acid backbones. *Bond lengths* are the distances between two bonded atoms and *bond angles* are the angles between three consecutive, bonded atoms. A *dihedral angle* is one defined for chains of four consecutive bonded atoms and is the angle between the vector normal to the plane of the first three atoms and the vector normal to the plane of the last three atoms.

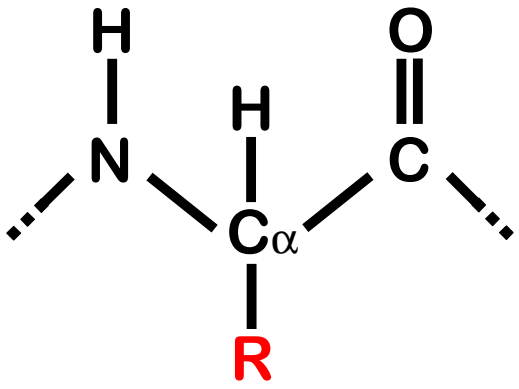


Figure 5.1: The topology of an amino acid in a protein.

Protein structure is classified into one of four categories: primary, secondary, tertiary, and quaternary structure. The *primary structure* of a protein is its amino acid sequence. *Secondary structures* are distinctive substructures found in native conformations. Two of the most common secondary structures are the α-helix, where the backbone is wrapped into a coil-like structure, and the β-sheet, where several portions of the backbone are aligned parallel (or antiparallel) to one another in a single plane. Repeated patterns of secondary structures that appear in many proteins are often referred to as *motifs*. *Tertiary* structure is the conformation of the atoms of a protein in three dimensions. Finally, *quaternary* structure refers to the association of two or more polypeptide chains into a structure stabilized mainly by noncovalent interactions (hydrogen bonding, van der Waals interactions, etc.).

The Protein Data Bank (PDB)[2] [11] is a database of experimentally determined native conformations of proteins (and nucleic acids). As of June 2005, the PDB contained more than 31,000 structures (∼28,000 proteins).

[2] `http://www.pdb.org`

5.3 Protein Models

According to Anfinsen's thermodynamic hypothesis [7], the native conformation of a protein is the one for which its Gibbs free energy is at a minimum. Although the Gibbs free energy of a protein in a particular conformation may be computed using quantum mechanical calculations, the computational demands for such calculations are too great for existing resources.

Several empirical potential energy functions, or force fields, have been developed for approximating the Gibbs free energy of a protein [92]. These force fields are typically parameterized using experimentally obtained data to approximate the Gibbs free energy or only compute the enthalpy, or internal energy, of the protein, neglecting any entropic contributions. Despite the imperfect approximations, many researchers agree that these models can be used to gain insight and understanding into the native states and folding dynamics of proteins.

A typical empirical potential energy function is a sum of five terms:

$$E(X) = E_{bl} + E_{ba} + E_{dih} + E_{vdw} + E_{el} \tag{5.1}$$

where the terms are specified in Table 5.1. The variables $X \in \mathbb{R}^{3n}$ are the Cartesian coordinates of the protein's n atoms, r are bond lengths, θ are bond angles, ϕ are dihedral angles, r_{ij} are distances between nonbonded atoms i and j, and q_i, q_j are (partial) charges on atoms i and j. The dielectric constant of the medium in which the analysis takes place is denoted by ϵ. The remaining parameters are chosen to fit experimentally determined native conformations of several small peptides or other organic compounds and vary depending on the atoms involved in each interaction. The numbers k_r, k_θ, and $k_{n\phi}$ are force constants; $\bar{r}$ is the average bond length; $\bar{\theta}$ is the average bond angle; and a_{ij} and b_{ij}

Term	*Interaction Type*	*Standard Form*
E_{bl}	bond length	$\sum_{\text{bonds}} \frac{k_r}{2}(r - \bar{r})^2$
E_{ba}	bond angle	$\sum_{\text{angles}} \frac{k_\theta}{2}(\theta - \bar{\theta})^2$
E_{dih}	dihedral angle	$\sum_{\text{dihedrals}} k_{n\phi}\left[1 + \cos(n\phi)\right]$
E_{vdw}	van der Waals	$\sum_{i,\,j} \left\{ \frac{a_{ij}}{r_{ij}^{12}} - \frac{b_{ij}}{r_{ij}^{6}} \right\}$
E_{el}	electrostatic	$\sum_{i,\,j} \left\{ \frac{q_i q_j}{\epsilon r_{ij}} \right\}$

Table 5.1: Standard terms found in empirical potential energy functions associated with protein models.

determine the distance between two atoms at which the van der Waals potential is at a minimum and the value of that minimum energy. In some force fields all terms modeling interactions between atoms that are not bonded (E_{vdw}, E_{el}) are combined into a single expression and called the *nonbonded potential*. Nonbonded potentials typically include sums of pairwise interactions taken over pairs of atoms separated by at least 2, 3, or 4 bonds; the choice for this number varies in existing force fields and is still a subject of debate. Other terms appearing in force fields include hydrogen bonding and solvation interaction terms, but there is little agreement among researchers as to the most suitable formulations of these potentials. Commonly used force fields include AMBER [21], CHARMM [15], ECEPP/3 [81], GROMOS [113], OPLS [62], and MM3 [6].

Protein structure prediction by energy minimization typically makes use of

one of these force fields or a simplified version of one of them. The force fields above are referred to as *all-atom* force fields, models capable of and parameterized for calculating the energy of a protein using all of its constituent atoms.

Residue-level models simplify energy calculations by including only backbone atoms and a single particle modeling the collective properties of the atoms in each residue. The UNRES model is an example of a residue-level model in which the C_α atoms and *united residues* are included in the energy calculations [72].

Backbone models include only the atoms that make up the backbone of the protein. To simplify calculations, the potential energy in some backbone models is a function of the angles defining the geometry of the backbone. Such a model is used in the experiments presented in Chapter 6. Another example of a backbone model is presented in Chapter 7, where only the C_α atoms are included and the properties of an entire amino acid residue are modelled in the parameterization of its corresponding C_α. The term *chain* is often used to refer to an instance of a protein derived using a backbone model.

The use of simplified models reduces the complexity of the interactions and hence reduces the amount of computation involved in energy calculations. Software tools for reconstructing all-atom structures from backbone structures (e.g., PHOENIX[3] [73], BB[4] [2], and MaxSprout[5] [54]) are often employed when simplified models are used for structure prediction.

Other simplified models developed for protein analysis via energy minimization include lattice models [30], statistical potentials [79, 105], and pairwise inter-

[3] `http://cbsu.tc.cornell.edu/software/protarch/index.htm`

[4] `http://mccammon.ucsd.edu/~adcock/bb.html`

[5] `http://www.ebi.ac.uk/maxsprout/`

action models [87]. Energy calculations in lattice models are extremely efficient as the positions of atoms included in the models are restricted to points on a finite lattice in three dimensions. The drawback of lattice models is their inability to model or predict secondary structures (especially helices) accurately. Statistical potentials, sometimes called knowledge-based potentials or scoring functions, are derived using experimentally determined native structures of proteins (typically from the PDB). These potentials are based on the correlation of the observed frequency of a structural feature with its associated free energy. Thus, these potentials have a global minimum corresponding to the most frequently observed native conformations (or collections of substructures found most often in native conformations). Pairwise interaction models are often used in testing computational methods for solving the protein folding problem. One commonly used instance is the Lennard-Jones 6–12 potential, which consists solely of the van der Waals term, E_{vdw}, with $a_{ij} = 2$ and $b_{ij} = 1$.

The potential energy functions used for protein analysis typically have a large number of local minima, many of which are close in function value to the global minimum. Moreover, it is estimated that the number of local minima increases exponentially with the number of atoms in a protein [71, 123]. Results show that minimizing an energy function of the form in (5.1) is NP-hard [85], prompting the authors of those results to suggest that "function-minimization algorithms can be efficient for protein structure prediction only if they exploit protein-specific properties."

5.4 Existing Methods for Energy Minimization

Over the past four decades, many approaches have been developed for determining the native conformation of a protein via minimization of a potential energy function. We list some of the more effective methods in this section. However, few of these methods make use of protein-specific properties, and none has been designed to take advantage of structural similarities of sequence-related pairs of proteins.

Several variants of the local and global optimization methods presented in Chapter 2 have shown promise for accurately predicting native conformations for small proteins ($\sim$ 50–200 aa). Local methods include the truncated Newton method [126] and a hybrid limited memory BFGS quasi-Newton/Hessian-free Newton method [24]. Global methods include stochastic search methods [23], simulated annealing [63, 104, 122, 123], evolutionary algorithms [14, 70, 69], and smoothing methods [48, 49, 71, 90, 98, 110, 125]. Other methods outside of these general method classes include convex global underestimation [89], derivative-free pattern search [36, 46], stochastic tunneling [100, 121], branch-and-bound [39, Chapters 15–16], and packet annealing [102].

5.5 HOPE

We now present the use of HOPE for minimizing potential energy function associated with protein models. The homotopy functions used in HOPE are denoted by $H(X, \lambda)$ and map the potential energy functions of a template protein, $E^0(X)$, to that of target protein, $E^1(X)$. The native conformation of the template protein is assumed to be known and is used as the starting conformation in HOPE.

The goal in using HOPE for protein structure prediction is to take advantage of the protein-specific properties of the energy function used and any relationship between the native conformations of the template and target proteins. Moreover, we would like to determine to what extent the amino acid sequences of the two proteins must match in order to produce accurate structural predictions using HOPE.

HOPE is similar to comparative modeling methods in that it uses the properties of a template protein to help predict the native conformation of a target protein. In contrast, HOPE in the current implementation uses a single template protein, whereas many comparative modeling methods use pieces of one or more template proteins to help predict the native conformation of a target protein. HOPE is also related to smoothing methods for energy minimization. However, in such methods the deformation starts with a smooth approximation of the template's potential energy function and the output is often only a single local minimizer of the target energy function. Finally, HOPE can be viewed as a simulated annealing method on an evolving energy landscape defined by the homotopy function, using a constant temperature ($T = 0$) in its annealing schedule and a move class that includes only local minimizers as candidate conformations.

Homotopy methods have been used previously for exploring potential energy surfaces and computing stationary points of energy functions [1] and for computing optimal configurations of atomic and molecular clusters [20, 58]. In both instances standard homotopy functions (convex, fixed point, etc.) were employed, and such functions do not exploit the protein-specific features of potential energy functions—e.g., electrostatic interactions between charged particles and torsional energy dependent on properties of the particles defining each dihedral angle in

a polypeptide chain. Moreover, the homotopy functions used were smoothing functions for the potentials of a single cluster of atoms. In contrast, we use a homotopy function that deforms the potential energy function of a template protein into that of a target protein. Thus, HOPE takes advantage of the sequence-based and/or structural relationships between proteins in predicting the native conformation of the target protein.

In the next two chapters, we demonstrate the use of HOPE in solving the protein folding problem. Two different protein models are introduced and HOPE is applied to minimizing the potential energy functions associated with the models. Several different homotopy functions and perturbations are used to demonstrate the flexibility of HOPE in taking advantage of model-specific properties.

Chapter 6

Numerical Experiments: Chains of Charged Particles

In this chapter, we present the results of HOPE and HOM applied to a problem involving chains of charged particles. These results demonstrate that HOPE is more successful than HOM at predicting native conformations using this model. Also, these two methods were more successful than a standard quasi-Newton method at minimizing the potential energy of the model, with HOPE showing significantly superior performance and HOM only marginally better performance.

In this model, two types of amino acids are allowed, and each amino acid is represented by its C_α atom. Thus, a polypeptide chain is modeled as a chain of particles in two-dimensional space, where each particle corresponds to a C_α atom and contains a charge of ± 1, depending on which type of residue it models. The value of models using a limited set of amino acids and restricted to two-dimensional space is discussed in [109] and the use of such simple models in testing a new method for solving the protein folding problem is presented in [90].

6.1 The Potential Energy Function

Let $X \in \mathbb{R}^{2n}$ denote the Cartesian coordinates of a chain of n particles in two dimensions, with $X_k \in \mathbb{R}^2$ containing the coordinates of the k^{th} particle. We also

use the notation X_k to denote the k^{th} particle in the chain. Figure 6.1 shows the geometry of the chain of charged particles. Four particles, $X_i, \ldots, X_{i+3}$, are depicted as nodes in the figure, with lines between nodes representing the bonds between particles.

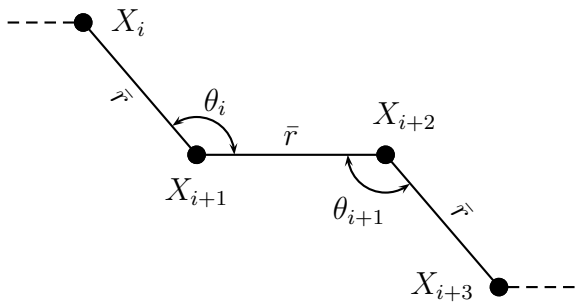


Figure 6.1: Geometry of the chain of charged particles.

The distance between particles X_i and X_j is denoted by $r_{ij} = \|X_i - X_j\|_2$, with bond lengths fixed at a distance $\bar{r}$. The angle $\theta_i \in [0, 2\pi)$, formed between particles X_i, X_{i+1}, and X_{i+2}, is the bond angle. To remove the rotational and translation freedom from each conformation, we fix the centers of the first two particles in the chain at the points $(\bar{r}, 0)$ and $(0, 0)$. The coordinates of the particles in the remainder of the chain are computed as

$$X_k = X_{k-1} + \begin{pmatrix} \cos(\theta_{k-2}) & -\sin(\theta_{k-2}) \\ \sin(\theta_{k-2}) & \cos(\theta_{k-2}) \end{pmatrix} (X_{k-2} - X_{k-1}) \tag{6.1}$$

where X_k is taken to be a column-vector. Thus, the $n-2$ bond angles fully determine a conformation.

The potential energy function, $E : \mathbb{R}^{n-2} \to \mathbb{R}$, is given by:

$$E(\theta) = E_{vdw}(\theta) + E_{el}(\theta) \tag{6.2}$$

where E_{vdw} and E_{el} are the van der Waals and electrostatic potentials, respectively. The van der Waals potential is a sum of pairwise interaction terms as in Table 5.1:

$$E_{vdw}(\theta) = \sum_{i=1}^{n-3} \sum_{j=i+3}^{n} \varepsilon \left(\left(\frac{\sigma}{r_{ij}} \right)^{12} - 2 \left(\frac{\sigma}{r_{ij}} \right)^{6} \right) \tag{6.3}$$

where ε and σ determine the minimum energy value and the distance at which that minimum occurs for pairwise particle interactions. Figure 6.2(a) shows a plot of the pairwise interactions in E_{vdw} of particles X_i and X_j as a function of the distance between them. The values of $\varepsilon = 0.4$ and $\sigma = 3.6$ are used.

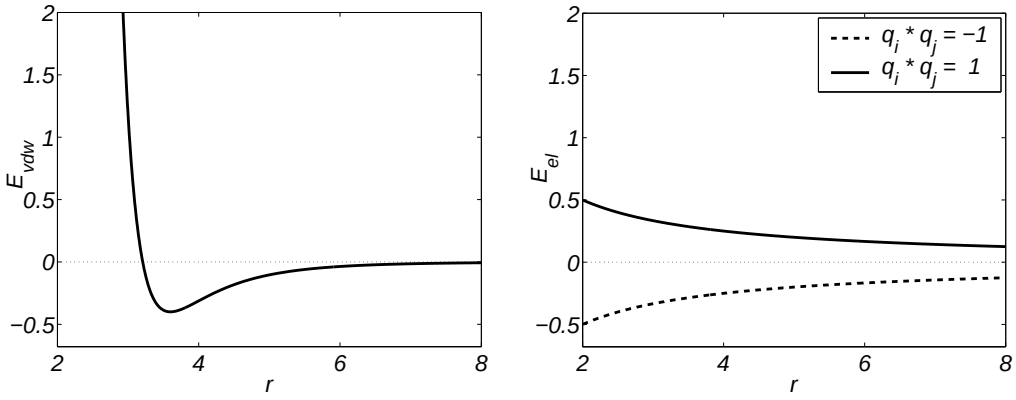
The electrostatic potential, as in Table 5.1, is given by:

$$E_{el}(\theta) = \sum_{i=1}^{n-2} \sum_{j=i+2}^{n} \frac{q_i q_j}{r_{ij}} \tag{6.4}$$

where q_k denotes the charge on particle X_k. Figure 6.2(b) shows a plot of the pairwise interaction in E_{el} between particles X_i and X_j as a function of the distance between them. The dashed curve is the interaction energy between particles with charges of opposite sign (i.e., $q_i q_j = -1$), and the solid curve is the energy between particles having charges of the same sign (i.e., $q_i q_j = 1$).

6.2 The Homotopy Function

We now define the homotopy between the potential energy functions of the template chain, $E^0(\theta)$, and the target chain, $E^1(\theta)$. We assume that the template and target chains contain the same number of particles. The homotopy is designed to deform the charges of the template chain (q^0) into those of the target chain (q^1). The energy terms in E_{vdw} depend on the position of particles and thus their contribution in the homotopy function is independent of the homotopy parameter.



(a) (b)

Figure 6.2: Plots of (a) van der Waals ($\varepsilon = 0.4$ and $\sigma = 3.6$) and (b) electrostatic potentials terms.

The homotopy function is given as:

$$H(\theta, \lambda) = \sum_{i=1}^{n-2} \sum_{j=i+2}^{n} \frac{q_i(\lambda) q_j(\lambda)}{r_{ij}} + E_{vdw}(\theta) \tag{6.5}$$

where $q_k(\lambda)$ is a continuous function mapping q_k^0 to q_k^1 such that

$$q_k(0) = q_k^0 \ , \quad \text{and} \tag{6.6}$$

$$q_k(1) = q_k^1 \ . \tag{6.7}$$

The functions $q_k(\lambda)$ depend on $\tilde{n}$, the number of corresponding particles in the template and target chains that do not match in sign. These functions are defined as

$$q_k(\lambda) = \begin{cases} q_k^1, & \text{if } q_k^0 = q_k^1 \ \text{ and} \\ \rho_j(\lambda) q_k^1 + (1 - \rho_j(\lambda)) q_k^0, & \text{if } q_k^0 \neq q_k^1 \ , \end{cases} \tag{6.8}$$

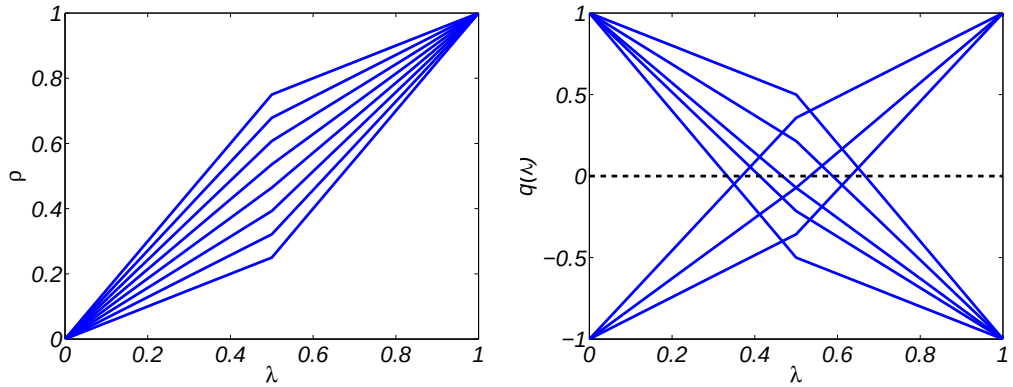
where

$$\rho_j(\lambda) = \begin{cases} \left(\frac{1}{2} + \frac{j-1}{\tilde{n}}\right) \lambda, & \text{if } \lambda \leq 0.5 \ \text{ and} \\ \left(\frac{1}{2} + \frac{\tilde{n}-(j-1)}{\tilde{n}}\right) \lambda - \frac{1}{2} + \frac{j-1}{\tilde{n}}, & \text{if } \lambda > 0.5 \ , \end{cases} \tag{6.9}$$

for $j = 2, ..., \tilde{n}$ when $\tilde{n} > 1$ and

$$\rho_1(\lambda) = \lambda \tag{6.10}$$

when $\tilde{n} = 1$. Figure 6.3 shows the plots of $\rho(\lambda)$ and the corresponding $q(\lambda)$ where the charges of $\tilde{n} = 8$ corresponding particles in the template and target chains do not match. The functions $\rho_j(\lambda)$ all change at different rates, which means that $q_k(\lambda) = 0$ for at most one value of k for any particular value of λ (as can be seen in Figure 6.3(b)). We found that allowing several charges to pass through 0 at the same point led to poor performance of HOPE and HOM.



(a) (b)

Figure 6.3: Examples of (a) $\rho(\lambda)$ and (b) corresponding $q(\lambda)$ when $\tilde{n} = 8$ particles differ in charge in the template and target chains.

6.3 Results

We performed three sets of experiments using this model. In the first set, we performed an exhaustive search for the native conformations of small chains ($n = 4, 5, 6$). The results of these experiments included the creation of a database of native structures that can be used in testing minimization methods. In the second set of experiments, HOPE, HOM and `QNewton-BFGS` were used to predict

the native structures in that database. Therefore, in these experiments, exact solutions were available and the efficacy of the methods could be measured accurately. In the third set of experiments, HOPE, HOM, and `QNewton-BFGS` were applied to problems involving larger chains ($n = 20$). For these experiments, native conformations were not available.

6.3.1 Computations using an Exhaustive Search

In this section we present the results of performing an exhaustive search to find the native conformations of small chains ($n = 4, 5, 6$). The search was implemented in C (using the GNU C compiler, `gcc` v3.2.2) and performed in Linux on a 2.2 GHz Pentium IV processor with 512 Mb of RAM. Our conclusion is that using this hardware, an exhaustive search is too computationally expensive for chains with $n > 7$ particles. However, for chains of size $n = 4, 5, 6$ we were able to produce a database of native conformations that can be used to test other minimization methods applied to this problem.

For each n, there are 2^n distinct chains of charged particles. For example, for $n = 4$, the possible chains (denoted here by the charges on the particles) are

```
+1,+1,+1,+1       -1,-1,-1,-1
-1,+1,+1,+1       +1,-1,-1,-1
+1,-1,+1,+1       -1,+1,-1,-1
-1,-1,+1,+1       +1,+1,-1,-1
+1,+1,-1,+1       -1,-1,+1,-1
-1,+1,-1,+1       +1,-1,+1,-1
+1,-1,-1,+1       -1,+1,+1,-1
-1,-1,-1,+1       +1,+1,+1,-1
```

However, due to symmetry, we need only consider the chains where the last charge is positive, i.e. $q_n = +1$. This reduces the number of unique chains to 2^{n-1}.

We searched for the lowest energy conformation using a grid on $[0, 2\pi)$ with 1° (≈ 0.017453 radians) increments, so that there were 360 possible angles for each θ_k ($k = 1, \ldots, n-2$). Using this grid, 360^{n-2} potential energy function evaluations must be performed for each chain to determine which conformation has the lowest energy. We can reduce this number by a factor of 2 by restricting the first angle to $\theta_1 \in [0, \pi]$. Table 6.1 presents an estimate of N_f, the number of function evaluations required for all 2^{n-1} chains, for several values of n.

n	**4**	**5**	**6**	**7**	**8**	**9**	**10**
$\log_{\mathbf{10}}(\mathbf{N_f})$	5.41	8.27	11.13	13.99	16.84	19.70	22.56

Table 6.1: Function evaluations required for the exhaustive search of native conformations of chains of charged particles.

Since each function evaluation consists of translating bond angles into particle coordinates, computing distances between all pairs of non-bonded particles, and

computing and summing energies due to pairwise interactions, the number of floating point operations quickly becomes too great for available computational resources.

In order to reduce the amount of computation, we did not perform function evaluations for a conformation for which the distance between any pair of non-bonded particles in the chain was less than or equal to the bond lengths (which is fixed at $\bar{r}$ for all bonded particles in the chain). We need only the first k angles to compute the pairwise distances between particles $X_1, X_2, \ldots, X_{k+2}$. If two of these particles are too close together when we increment angle θ_k, we need not perform the search for any values of the angles $\theta_{k+1}, \theta_{k+2}, \ldots, \theta_{n-2}$ for this value of θ_k. This leads to a further reduction in computation by 1) not computing the coordinates of and distance between particles later in a chain when two particles earlier in the chain are too close, and 2) not performing a function evaluation of those conformations.

After reducing the computation involved in the exhaustive search using these measures, the computing required for finding native conformations was still too great for chains with $n > 6$ particles. A search for the lowest energy conformation of all the chains of $n = 4, 5, 6$ particles was performed. Also, a partial search was conducted for $n = 7, 8$ particles for estimating the time needed to carry out the full search for chains of these sizes. The partial search consisted of fixing the first angle and performing an exhaustive search of the lowest energy conformation using the remaining $n - 3$ angles. Since there are many infeasible conformations due to the minimum distance requirements described above, the number of different values possible for the first angle is 103 (out of 180). The wall-clock times for searching for the lowest energy conformations are presented

in Table 6.2. The estimated times for $n = 7, 8$ are computed by multiplying the time required for the fixed first angle search by 103.

n	**4**	**5**	**6**	**7**	**8**
Time	$<$ .1 seconds	17 seconds	2 hours	$>$ 37 days*	$>$ 1.75 years*

** estimated*

Table 6.2: Amount of wall clock time required to perform the exhaustive search of native conformations of chains of charged particles.

We conclude that the amount of time required in performing an exhaustive search to determine the lowest energy conformation of a chain of charged particles is too great to be of practical use. Presented in Appendix A are the lowest energy conformations for chains of size $n = 4, 5, 6$, and we can see that there is not much variation in the conformations for chains with different sets of charges.

6.3.2 Computations using HOPE/HOM on Small Chains

In this section, we present the results of HOPE, HOM, and `QNewton-BFGS` used for predicting the native conformations of chains containing $n = 4, 5, 6$ particles. The lowest energy conformations found by the exhaustive search method described in the previous section were taken to be the native conformations. The results show that all three methods are able to predict the native conformations effectively and efficiently for all but a few template-target pairs.

In this set of experiments, each conformation was used as a template (starting point) to predict the native conformations of the remaining targets. `QNewton-BFGS` was applied to the problems using its default parameters. In HOM and HOPE $m = 4$ steps in λ were taken and `MaxIter` $= 20$ iterations of `QNewton-BFGS` were

allowed for each local minimization performed. Finally, the remaining parameters in HOPE were set to $\hat{c} = 1$ and $c_{max} = 4$. Perturbations used in HOPE were performed using ξ_{pct} (Section 4.1) with maximum perturbations of $0.1 \cdot \|\theta^0\|_2$ (i.e. 10% of the the 2-norm of the the angles defining the native conformation of the template).

Table 6.3 presents the results of these experiments. The first two columns show the sizes of the chains and the corresponding number of pairs of templates and targets ($= n^2 - n$) comprising the experiments. The remaining columns show the percentage of experiments where `QNewton-BFGS`, HOM, and HOPE successfully predicted the native conformations, where a success is measured using $relerr_{f^1}$ as in (4.7). Specifically, a successful prediction is one satisfying

$$\frac{|E^1(\theta^1) - E^1(\theta^*)|}{|E^1(\theta^*)|} \leq 10^{-3} \,, \tag{6.11}$$

where θ^1 is a conformation predicted using one of the three methods and θ^* is the corresponding native conformation. We used `QNewton-BFGS` to minimize the energy of each chain, starting from the native conformations produced by the exhaustive search. We found that the difference in energies between the native conformations and those found using `QNewton-BFGS`, as measured using (6.11) was always below 10^{-3}. Thus, we used this number in the success measure.

We see that all three methods successfully predicted most native conformations, with HOPE and HOM predicting all of the native conformations for $n = 4, 6$. This was to be expected, since many of the native conformations of these small chains are very similar (see Appendix A).

We next focused on a single template-target pair for which all three methods were unsuccessful. For this pair the native conformation is very similar to a conformation of the target chain corresponding to a *local* minimizer of E.

n	Pairs	QNewton-BFGS Success (%)	HOM Success (%)	HOPE Success (%)
4	56	93	100	100
5	240	97	97	97
6	992	99	100	100

Table 6.3: Comparison of prediction results for QNewton-BFGS, HOM, and HOPE for small chains ($n = 4, 5, 6$).

The template and target chains are

$$q^0 = \{+1, -1, +1, -1, +1\} \tag{6.12}$$

$$q^* = \{-1, -1, +1, +1, +1\} \tag{6.13}$$

where q^0 and q^* are the charges on the particles in the template and target chains, respectively. Figure 6.4 shows the native conformations for (a) the template chain and (b) the target chain. In these figures, each particle in the chain is depicted using a circle with radius 0.8 that is centered at the particle's coordinates. Blue circles (with a "+" at their centers) and red circles (with no symbols at their centers) correspond to particles with charges of $+1$ and -1, respectively. All bond lengths are fixed at $\bar{r} = 1.5$; thus the overlap of circles highlights the particles that are bonded.

The conformation predicted by all three methods is shown in Figure 6.5. This conformation is almost identical to the native conformation of the template, which is used as the starting conformation in the minimization methods.

We ran experiments with HOPE and HOM on this template-target pair using different parameter choices in the methods. The goals in performing these experiments were to determine if the homotopy optimization methods could solve this

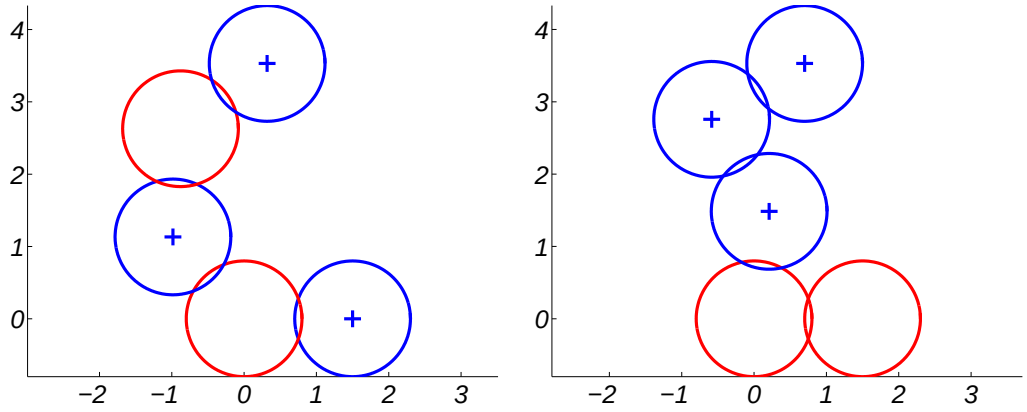


Figure 6.4: Native conformations of the (a) template chain, q^0, and (b) target chain, q^* used in the extended experiments involving small chains.

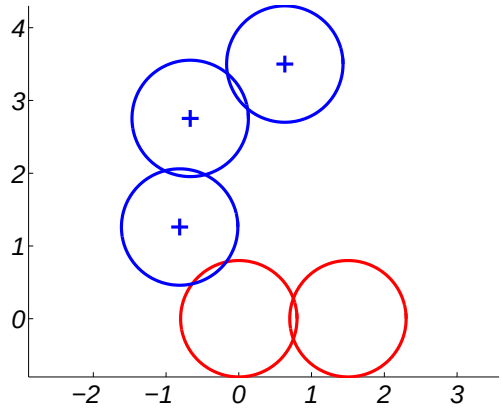


Figure 6.5: Conformation predicted by `QNewton-BFGS`, HOM, and HOPE for the target chain q^*.

problem and to identify which parameters influence the success if possible.

HOM was run using this template-target pair with $m = 2, 4, 8, 10, 20, 30$ steps in λ. Unfortunately, HOM was not able to find the native conformation in any of those runs.

HOPE was run using this template-target pair with $\hat{c} = 1$ and all combinations of $m = 2, 4, 8$ steps in λ and maximum perturbations of 25%, 50%, 75%, and 100% of $\|\theta^0\|_2$. For each combination of parameters, 100 runs of HOPE were performed.

Table 6.4 presents the number of runs for each parameter combination where HOPE successfully predicted the native conformation of the target chain. The general trend in these results is that more perturbation and more steps in λ leads to better performance of HOPE. As with results on the standard test problems in Chapter 4, though, it is not clear which parameters influence HOPE the most.

	Maximum Perturbation			
m	**25**%	**50**%	**75**%	**100**%
2	6	24	37	36
4	28	74	78	85
8	39	82	83	87

Table 6.4: Successful predictions of the native conformation of the template chain (out of 100) using HOPE with several combinations of algorithm parameters.

In this section, we presented results of two experiments using HOPE, HOM, and `QNewton-BFGS` to predict the native conformations of small chains. The results of the first experiments showed that all three methods were very successful. One of the problems which all three methods failed to solve in the first set of experiments was used to further test of HOPE. The results of those experiments show that the use of perturbation and the number of steps taken in λ led to increases in the number of successful predictions of the native conformations using HOPE. In the next section, we see that HOPE outperforms HOM and `QNewton-BFGS` on larger chains as well.

6.3.3 Computations using HOPE/HOM on Larger Chains

In the third set of experiments, we applied HOPE, HOM, and `QNewton-BFGS` to chains of $n = 20$ particles. These experiments differ from the previous experiments in that the native conformations of the template and target chains were not known. Therefore, we were not able to verify whether these methods are capable of predicting the native structure, and conformations used as starting points corresponded to *local* minimizers of E^0 rather than global minimizers. Nevertheless, the experiments show that HOPE outperforms HOM and `QNewton-BFGS` by predicting lower energy conformations for most template-target pairs; in a small number of experiments HOM and `QNewton-BFGS` predict the same conformation as HOPE.

The parameters used in the three methods were the same as in the previous experiments, except $m = 8$ steps were taken in λ in HOM and HOPE and the maximum amount of perturbation allowed in HOPE was 20% of $\|\theta^0\|_2$. A total of 10 template chains was used in the experiments, where the sequence of charges for each chain was chosen randomly. For each of these template chains, target chains were generated by changing $\tilde{n} = 2, \ldots, 20$ charges. The positions in which the changes in the charges took place were chosen randomly for each value of $\tilde{n}$. In the experiments, only one run of HOPE was performed for each template-target pair.

The results of the experiments demonstrate that HOPE clearly outperforms both HOM and `QNewton-BFGS` in predicting the lowest energy conformations of the three methods. In all of the experiments, HOPE predicted the lowest energy conformation of any of the three methods, with HOM and `QNewton-BFGS` producing the same conformation in several of the experiments.

The results for one of the template chains is presented in Table 6.5. We present these results as they are an example where HOM and `QNewton-BFGS` performed the best, producing the same conformations as HOPE for 4 target chains. The first column shows the number of charges that are different in the template and target chains. The next three columns show the energies of the conformations generated by `QNewton-BFGS`, HOM and HOPE, respectively.

The remaining columns in the table show how much more computation is involved for HOM over `QNewton-BFGS` and HOPE over `QNewton-BFGS`, respectively, where the ratios of the number of function evaluations for the methods are given. Here, N_f^{qn}, N_f^{hom}, and N_f^{hope} are the numbers of function evaluations performed in `QNewton-BFGS`, HOM, and HOPE, respectively. On average, HOM required about 14.5 times as many function evaluations as `QNewton-BFGS`, and for HOPE about 49.6 times as many were required. Although there is more work performed by HOPE, we see that it consistently outperforms the other two methods.

Not only does HOPE produce lower energy conformations than the other two methods, it produces ensembles of minimizers. Often more than one conformation in the ensembles generated by HOPE had lower energy than the minimizers produced by HOM and `QNewton-BFGS`. The average number of unique ensemble conformations produced by HOPE that had lower energy than the corresponding conformations generated by HOM and `QNewton-BFGS` was 4.74 out of total of 8 ensemble members. (Recall from Section 3.3 that the final ensembles contain $(\hat{c}+1) \times c_{max}$ conformations.)

Figure 6.6 presents the conformations of the template used as the starting point and two examples of the conformations predicted by HOPE, HOM, and `QNewton-BFGS` ($\tilde{n} = 4, 19$) in the same experiments. We see that there is little

difference between the template and the conformations predicted by HOM and `QNewton-BFGS`. For HOPE, the differences between the template conformation and the predicted conformations were minor in some experiments ($\tilde{n} = 4$) and in others very dramatic ($\tilde{n} = 19$).

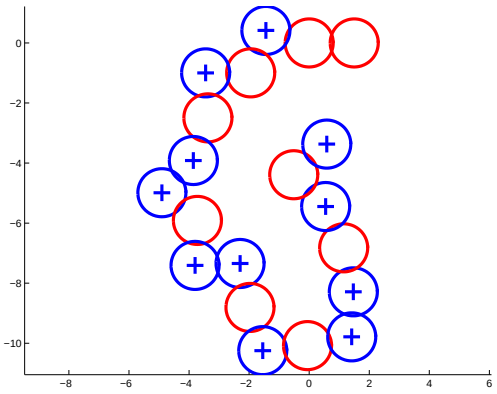
In this chapter, we presented the results of experiments on chains of charged particles. This experiments were our first using HOM and HOPE for protein structure prediction. As with the results on the standard test problems presented in Chapter 4, HOPE outperformed HOM and `QNewton-BFGS`. We will see in the next chapter that when applied to a more realistic protein model, HOPE is again successful at predicting low energy conformations.

$\tilde{\mathbf{n}}$	QNewton-BFGS $\mathbf{E}(\theta^{\mathbf{1}})$	**HOM** $\mathbf{E}(\theta^{\mathbf{1}})$	**HOPE** $\mathbf{E}(\theta^{\mathbf{1}})$	$\mathbf{N_f^{hom}/N_f^{qn}}$	$\mathbf{N_f^{hope}/N_f^{qn}}$
2	-18.596	-18.596	$-20.291^{\ddagger}$	18.4	61.1
3	-19.683	-19.683	$-21.685^{\ddagger}$	17.0	59.0
4	-19.389	-19.389	$-21.154^{\ddagger}$	17.9	61.1
5	$-19.429^{\dagger}$	-19.269	$-19.748^{\ddagger}$	5.8	16.5
6	-17.204	-17.204	$-18.788^{\ddagger}$	7.4	24.9
7	-20.825	-20.825	-20.825	18.9	73.6
8	-18.317	-18.317	$-19.870^{\ddagger}$	10.3	37.3
9	-17.780	-17.780	-17.780	7.1	23.4
10	-19.377	-19.377	-19.377	11.8	40.1
11	-21.293	-21.293	$-23.578^{\ddagger}$	18.0	60.1
12	-12.363	-12.363	$-13.844^{\ddagger}$	8.6	35.0
13	-17.933	-17.933	-17.933	6.5	22.2
14	-18.393	-18.393	$-20.589^{\ddagger}$	11.4	44.1
15	-14.990	$-15.017^{\dagger}$	$-18.238^{\ddagger}$	18.7	65.0
16	-18.746	-18.746	$-20.090^{\ddagger}$	18.4	58.5
17	-17.608	-17.608	$-18.475^{\ddagger}$	18.4	54.0
18	-18.850	-18.850	$-19.191^{\ddagger}$	19.5	67.7
19	-17.848	-17.848	$-20.271^{\ddagger}$	19.6	72.0
20	-17.037	-17.037	$-19.220^{\ddagger}$	21.5	66.8

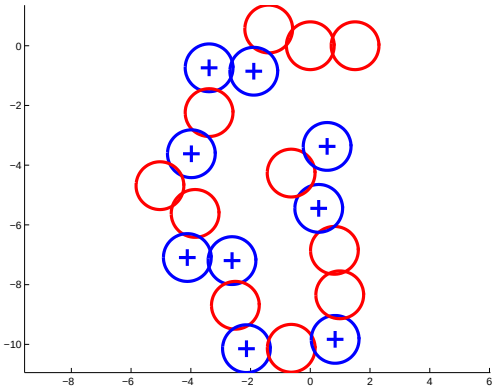
†*Lowest energy conformation between* QNewton-BFGS *and HOM.*

‡*Lowest energy conformation among all methods.*

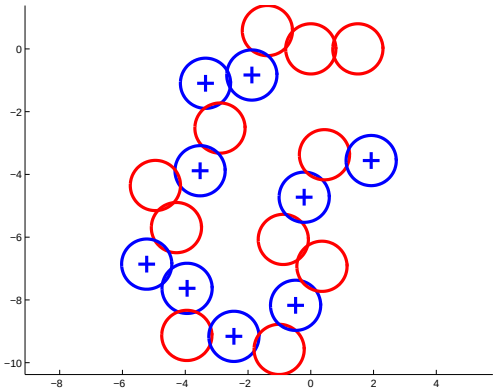
Table 6.5: Results of predictions using QNewton-BFGS, HOM, and HOPE on a chain containing $n = 20$ particles, where N_f^{qn}, N_f^{hom}, and N_f^{hope} are the numbers of function evaluations performed in QNewton-BFGS, HOM, and HOPE, respectively.



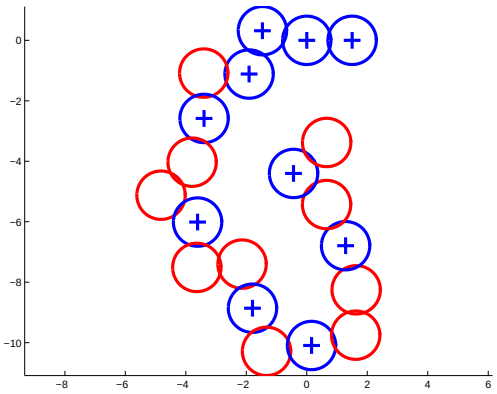
Template



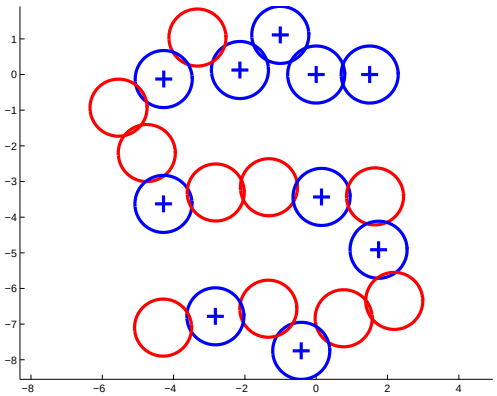
HOM and `QNewton-BFGS` ($\tilde{n} = 4$)



HOPE ($\tilde{n} = 4$)



HOM and `QNewton-BFGS` ($\tilde{n} = 19$)



HOPE ($\tilde{n} = 19$)

Figure 6.6: The template and predicted conformations of chains with $n = 20$ particles using HOPE, HOM, and `QNewton-BFGS`.

Chapter 7

Numerical Experiments: A Backbone Model of Proteins

In this chapter we present the results of numerical experiments using HOPE to predict the native conformations of proteins. Simulated annealing, parameterized to use more than twice the amount of computational resources to solve the same problem, was unsuccessful. We also show that HOM is more successful at predicting the native conformations than a globally convergent variant of Newton's method.

We use the previously introduced coarse-grained protein model of Veitshans, Klimov and Thirumalai [115] to test the efficacy of HOPE. In the coarse-grained protein model each amino acid is represented by its C_α atom. Thus, a polypeptide chain is modeled as a chain of particles, where each particle corresponds to an C_α atom and models one of three types of residues in terms of affinity for neighboring molecules: hydrophobic, hydrophilic, or neutral. The features of proteins that are most responsible for structural stability are included in the model—hydrophobic forces, van der Waals interactions, and torsional strain—and both bond lengths and bond angles are allowed to be variable. The diversity of hydrophobic species in real proteins is modeled in the interactions between the particles corresponding to hydrophobic residues.

7.1 The Potential Energy Function

Let $X \in \mathbb{R}^{3n}$ denote the Cartesian coordinates of a chain of n particles in three dimensions, with $X_k \in \mathbb{R}^3$ containing the coordinates of the k^{th} particle in the chain. We also use the notation X_k to denote the k^{th} particle. The geometry of a chain is presented in Figure 7.1. Four particles, $X_i, \ldots, X_{i+3}$, are depicted as nodes in the figure, with lines between nodes representing the bonds between particles. These bonds do not model true chemical bonds in the protein; rather they reflect the rigid distances between consecutive C_α atoms observed in native conformations of proteins (~ 3.8Å).

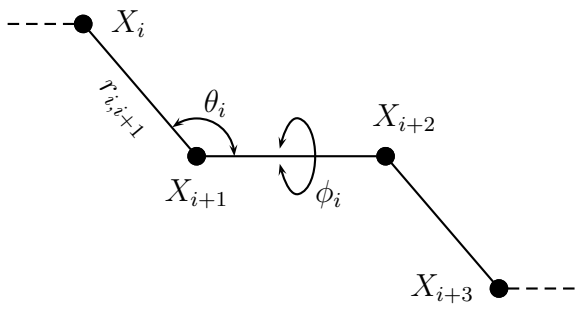


Figure 7.1: Geometry of the model protein.

The distance between particles X_i and X_j is denoted by $r_{ij} = \|X_j - X_i\|_2$; thus, $r_{i,i+1}$ is the bond length between consecutive particles X_i and X_{i+1}. The angle $\theta_i \in [0, \pi]$, formed between the three consecutive particles, X_i, X_{i+1}, and X_{i+2} is the bond angle. The angle $\phi_i \in [-\pi, \pi]$, formed between the vectors normal to the plane defined by particles X_i, X_{i+1}, and X_{i+2} and that defined by particles X_{i+1}, X_{i+2}, and X_{i+3} is the dihedral angle. The choice of sign for the dihedral angles conforms to the rules set forth in [60].

Each particle is assigned a particle type, p, depending on the type of residue

to which it corresponds: hydrophobic (B), hydrophilic (L), or neutral (N). For example, if particle X_k represents a hydrophobic residue then $p_k = B$.

The total potential energy of a chain of particles, $E : \mathbb{R}^{3n} \to \mathbb{R}$, is given by:

$$E(X) = E_{bl}(X) + E_{ba}(X) + E_{dih}(X) + E_{non}(X) \tag{7.1}$$

where E_{bl}, E_{ba}, E_{dih} and E_{non} correspond to the bond length, bond angle, dihedral angle, and nonbonded potentials, respectively. The bond length and bond angle potentials, which depend only on the coordinates of the particles, are as in Table 5.1:

$$E_{bl}(X) = \sum_{i=1}^{n-1} \frac{k_r}{2} \left(r_{i,i+1} - \bar{r}\right)^2 \tag{7.2}$$

$$E_{ba}(X) = \sum_{i=1}^{n-2} \frac{k_\theta}{2} \left(\theta_i - \bar{\theta}\right)^2 \tag{7.3}$$

with k_r, $\bar{r}$, k_θ, and $\bar{\theta}$ given.

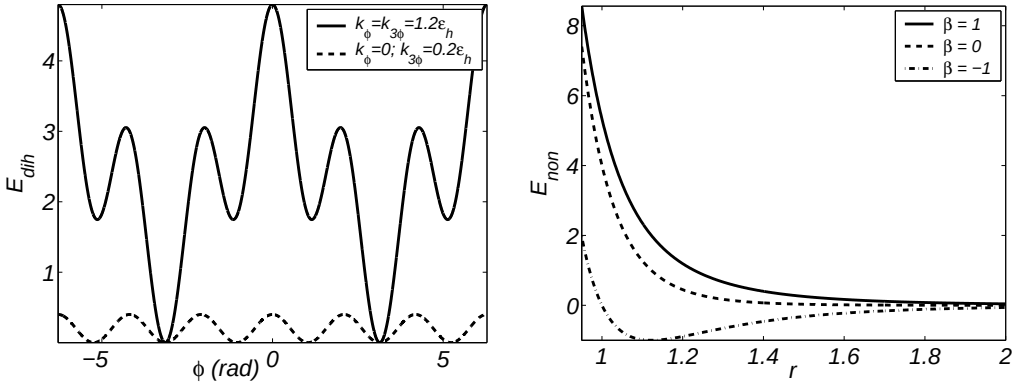
The dihedral angle potential, which does depend on the properties of particles in a particular chain, is as in Table 5.1 using $n = 1, 3$:

$$E_{dih}(X) = \sum_{i=1}^{n-3} \left[k_\phi(1 + \cos\phi_i) + k_{3\phi}(1 + \cos 3\phi_i)\right] \tag{7.4}$$

where k_ϕ and $k_{3\phi}$ take on one of two values depending on P_i, the number of neutral (N) particles in the subchain forming the dihedral angle ϕ_i. Specifically, the values of these parameters are given by

P_i	k_ϕ	$k_{3\phi}$
< 2	$1.2\varepsilon_h$	$1.2\varepsilon_h$
≥ 2	0	$0.2\varepsilon_h$

where ε_h is the average strength of hydrophobic interactions. Figure 7.2(a) shows a plot of the two curves for the possible sets of parameters as a function of the dihedral angle, ϕ. Note that there is a significant difference between the energy barriers of the two curves. Conformations of subchains with two or more neutral particles are more flexible (due to the lower energy barriers in the dashed curve of Figure 7.2(a)) than those of subchains with fewer than two neutral particles (with large energy barriers at $\phi = 0$ and $\phi \approx \pm 2$ in the solid curve).



(a) (b)

Figure 7.2: Plots of (a) dihedral and (b) nonbonded potential energy terms.

The nonbonded potential also depends on the types of particles in the chain, and is a form of the Lennard-Jones 6–12 potential (similar to E_{vdw} in Table 5.1):

$$E_{non}(X) = \sum_{i=1}^{n-3} \sum_{j=i+3}^{n} \gamma_{ij} \left\{ \alpha_{ij} \left(\frac{\bar{r}}{r_{ij}} \right)^{12} - \beta_{ij} \left(\frac{\bar{r}}{r_{ij}} \right)^{6} \right\} . \tag{7.5}$$

The parameters used in each pairwise interaction of particles i and j are

p_i	p_j	α_{ij}	β_{ij}	γ_{ij}
L	L,B	1	-1	$4\varepsilon_L$
N	L,N,B	1	0	$4\varepsilon_L$
B	B	1	1	$4\nu\varepsilon_h$

where $\varepsilon_L = \frac{2}{3}\varepsilon_h$, and ν is a dimensionless parameter assumed to have a Gaussian distribution with a mean value of 1 and a standard deviation of σ. The diversity in the hydrophobic residues is controlled by σ, with $\sigma = 0$ leading to no diversity (i.e., all hydrophobic residues are identical). We assume that interactions between hydrophobic residues are attractive; thus, only positive values of ν are used. (More details for the choice of ν are given in [115].) Specific values of σ for the chains used in computations are presented in Section 7.4. Figure 7.2(b) shows a plot of the three curves for the possible sets of parameters as a function of r, the distance between a given pair of particles. Note that the plot of the curve for $\beta = -1$ shows only the average interaction between the pair of particles (i.e., it is the curve for $\nu = 1$).

7.2 The Homotopy Function

We now define the homotopy between the potential energy functions of the template chain, $E^0(X)$, and the target chain, $E^1(X)$. This homotopy is designed to reduce high energy barriers that are due to the dihedral potential and allow for conformational changes to be driven solely by nonbonded interaction at some points during the deformation of E^0 into E^1. The reduction of energy barriers is necessary in situations where a dihedral angle in the native conformation of the template is of opposite sign to the corresponding angle in the native conformation of the template. Once nonbonded interaction becomes the most influential factor driving conformational changes, larger changes are allowed. This is necessary if the native conformations of the template and target proteins differ in structure by a significant amount.

First, the dihedral potentials are partitioned into two terms containing low

frequency ($\cos\phi_i$) and high frequency ($\cos 3\phi_i$) terms:

$$E_{dih}(X) = E_{dih1}(X) + E_{dih2}(X) \tag{7.6}$$

with

$$E_{dih1} = \sum_{i=1}^{n-3} k_\phi(1 + \cos\phi_i) \ , \text{ and} \tag{7.7}$$

$$E_{dih2} = \sum_{i=1}^{n-3} k_{3\phi}(1 + \cos 3\phi_i) \ . \tag{7.8}$$

Using these two dihedral terms, we define the homotopy function as

$$\begin{aligned} H(X,\lambda) = {} & E^0_{bl}(X) + E^0_{ba}(X) \\ & + (1-\rho_1(\lambda))E^0_{dih1}(X) + \rho_4(\lambda)E^1_{dih1}(X) \\ & + (1-\rho_2(\lambda))E^0_{dih2}(X) + \rho_5(\lambda)E^1_{dih2}(X) \\ & + (1-\rho_3(\lambda))E^0_{non}(X) + \rho_6(\lambda)E^1_{non}(X) \end{aligned} \tag{7.9}$$

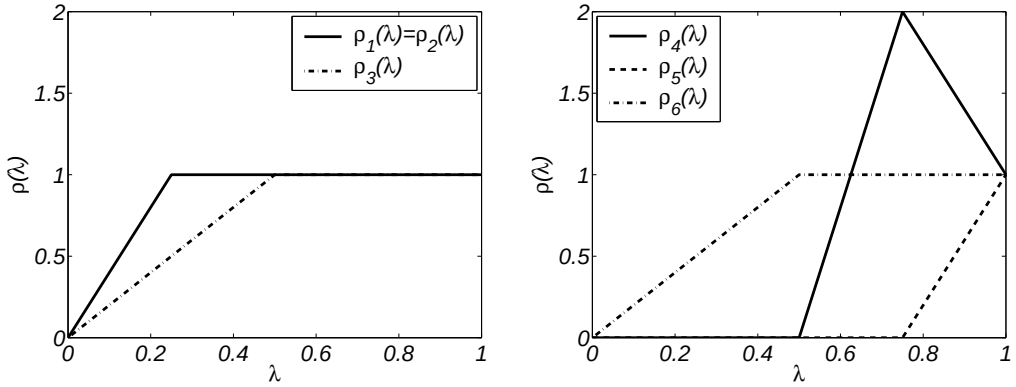
where $\rho_i(\lambda)$, $i = 1, \ldots, 6$, are continuous weighting functions dependent on the homotopy parameter, λ. In order to satisfy the conditions that $H(X,0) = E^0(X)$ and $H(X,1) = E^1(X)$, these functions must satisfy the following:

$$\rho_i(\lambda) = \begin{cases} 0, & \text{if } \lambda = 0 \\ 1, & \text{if } \lambda = 1 \end{cases} \ , \ i = 1, \ldots, 6 \ . \tag{7.10}$$

The convex homotopy, defined using $\rho_i(\lambda) = \lambda$, did not yield good results but was used as a starting point for developing a more useful homotopy. Specifically, we performed computations with HOM to identify modifications to the convex homotopy that increased the success rate of predicting the correct conformations for the target chains.

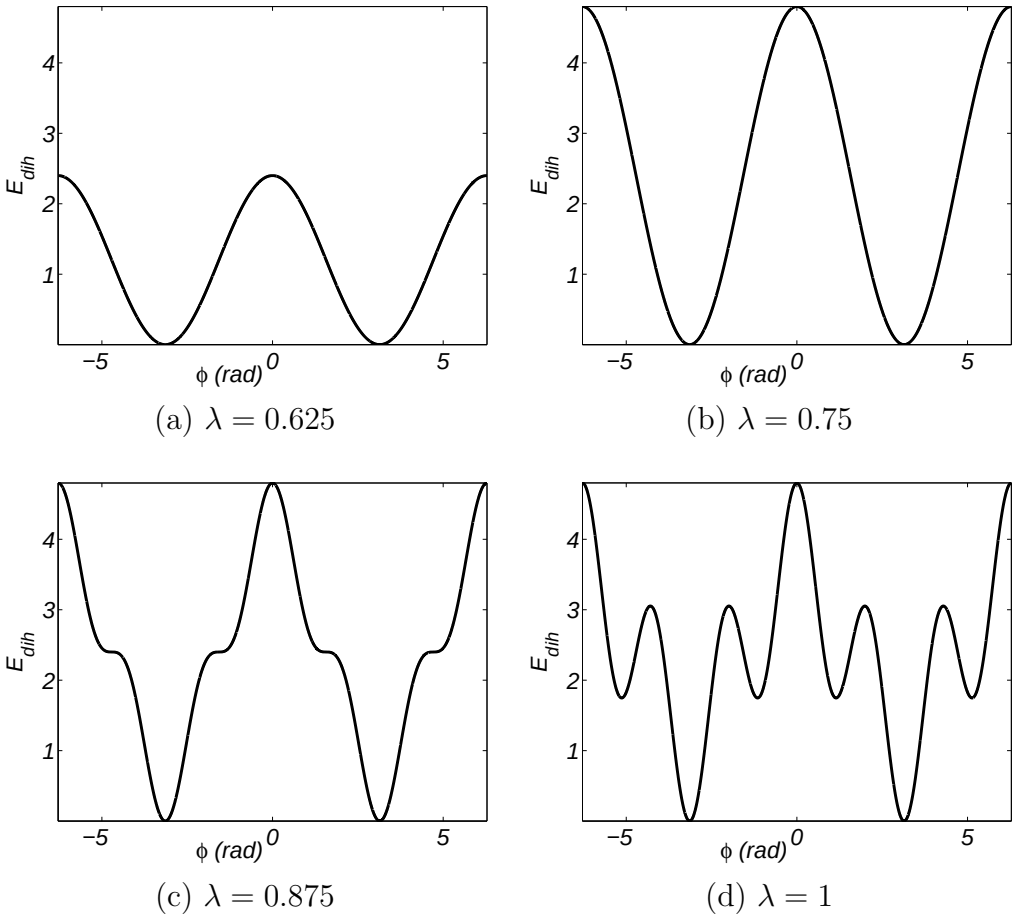
Figure 7.3 shows plots of the weighting functions, $\rho_i(\lambda)$, used in the computations. A convex homotopy deforms E^0_{non} into E^1_{non} in the first half of the homotopy ($\lambda \in [0.0.5]$) so that in the second half of the homotopy ($\lambda \in [0.5, 1]$), E^1_{non} is the only nonbonded potential contribution in H. The template dihedral terms (E^0_{dih1} and E^0_{dih2}) are driven to zero during the first quarter of the homotopy, and the target dihedral terms are not included until the second half of the homotopy. Thus, during the second quarter of the homotopy ($\lambda \in [0.25, 0.5]$) there are no dihedral angle potential contributions in H. This allows the nonbonded interactions to determine all conformational stability. We found this necessary for overcoming the large energy barriers in the dihedral potentials for template-target pairs whose lowest energy conformations contain dihedral angles of opposite sign corresponding to the same subchain. In other words, if a dihedral angle differs in sign in the lowest energy conformations for the template and target chains, there exists a large energy barrier (see Figure 7.2(a)) between the two conformations. The homotopy function, H, has been designed to allow for the required conformational changes for such cases.

In the second half of the homotopy, contributions from E^1_{dih1} and E^1_{dih2} are introduced into H, but at different rates. We do this because for subchains containing at most one neutral particle, E_{dih} has two local minima with high energy values; we would like to avoid such minima. Figure 7.4 shows the contributions of E^1_{dih} and E^1_{dih2} to H for several values of λ. At $\lambda = 0.5$, the potential is zero. As λ is increased from 0.5 to 0.75 the contribution of the low frequency dihedral terms is increased to $2 \times E^1_{dih1}$. This helps bias towards conformations with $\phi = \pm\pi$, avoiding the local minima of E^1_{dih}. As λ increases to 1 the high frequency terms are gradually included, leading to the true dihedral potential for the target chain.



(a) (b)

Figure 7.3: Plots of the weighting functions used in the homotopy function, $H(X, \lambda)$, for (a) template and (b) target terms.



(a) $\lambda = 0.625$ (b) $\lambda = 0.75$

(c) $\lambda = 0.875$ (d) $\lambda = 1$

Figure 7.4: Plots of the dihedral potential for subchains with fewer than two neutral particles for several values of λ in the second half of the homotopy.

7.3 Perturbations

In this section, we describe perturbations based on bond length, bond angle, and individual particle adjustments that have shown promise in our computations. Recall that during an iteration of HOPE, each conformation carried over from the previous iteration is perturbed to produce one or more new conformations. We have attempted to create perturbations that are related to the properties of the particles in the proteins. Specifically, we concentrate on perturbing the particles whose types in the template (p^0) and target (p^1) do not match.

7.3.1 Bond Length Perturbations

We first present a perturbation method based on adjustments in bond lengths. In this method, we start at one end of the chain and visit each particle, perturbing the bond length between particles X_k and X_{k+1} if $p^0_{k+1} \neq p^1_{k+1}$. The perturbed bond length between particles X_k and X_{k+1} becomes

$$\hat{r}_{k,k+1} = r_{k,k+1} + \delta_r \ , \tag{7.11}$$

where δ_r is taken from a uniform distribution on the interval $[-a_r, a_r]$. The choice of values for a_r is discussed in Section 7.4 in the context of specific computations.

Once particle X_k has been perturbed, the particles in the remainder of the chain, $X_{k+2}, \ldots, X_n$, are then shifted by δ_r in the same direction of change as X_{k+1}. Figure 7.5(a) depicts a perturbation of particle X_k for $\delta_r > 0$, where particles $X_{k+1}, X_{k+2}, \ldots, X_n$ are shifted to $\hat{X}_{k+1}, \hat{X}_{k+2}, \ldots, \hat{X}_n$. If another bond length is perturbed later in the chain, say between particles X_j and X_{j+1} for some $k < j < n-1$, then $\hat{X}_{j+1}$ is perturbed (instead of the original X_{j+1}) and $\hat{X}_{j+2}, \ldots, \hat{X}_n$ are shifted (instead of the original $X_{j+2}, \ldots, X_n$).

Figure 7.5(b) depicts a perturbation of particle X_k for $\delta_r > 0$, where the remainder of the chain is not shifted (i.e., particles $X_{k+2}, \ldots, X_n$ remain in place). Results of computations using bond length perturbations where the remainder of the chain was shifted were significantly better than for those using this type of perturbation, however. Therefore, we will present results only for the bond length perturbations with shifts.

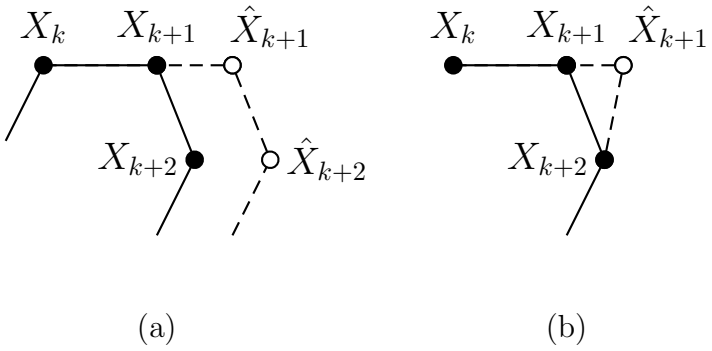


Figure 7.5: Perturbations based on bond length adjustments. In (a) X_{k+1} and the remainder of the particles in the chain are shifted and in (b) only X_{k+1} is shifted.

7.3.2 Bond Angle Perturbations

In this section, we present a perturbation methods based on adjustments in bond angles. For these perturbations, we once again start at one end of the chain, but visit each *angle*, perturbing θ_k if $p^0_{k+1} \neq p^1_{k+1}$ or $p^0_{k+2} \neq p^1_{k+2}$. Recall that θ_k is the bond angle between particles X_k, X_{k+1}, and X_{k+2}. The new bond angle is given by

$$\hat{\theta}_k = \theta_k + \delta_\theta \ . \tag{7.12}$$

where δ_θ is taken from a uniform distribution on the interval $[-a_\theta, a_\theta]$.

The coordinates of particle X_{k+2} can be adjusted to satisfy (7.12). As in the bond length perturbations, the particles in the remainder of the chain, particles $X_{k+3}, \ldots, X_n$, are rotated as well. Specifically, when θ_k is perturbed, particles $X_{k+2}, \ldots, X_n$ are rotated by δ_θ around the normal at the point X_{k+1} to the plane defined by particles X_k, X_{k+1}, and X_{k+2}. Figure 7.6(a) depicts such a perturbation of θ_k for $\delta_\theta > 0$, with the coordinates of particles $X_{k+2}, \ldots, X_n$ rotated to $\hat{X}_{k+2}, \ldots, \hat{X}_n$. If another angle is perturbed later in the chain, say angle θ_j for some $k < j < n-2$, then particles $\hat{X}_{j+2}, \ldots, \hat{X}_n$ are rotated (instead of the original $X_{j+2}, \ldots, X_n$).

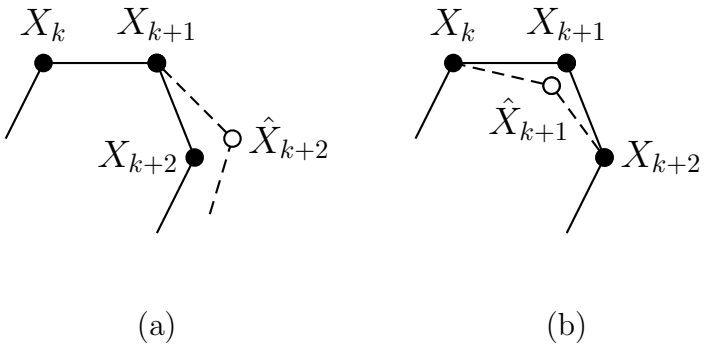


Figure 7.6: Perturbations based on bond angle adjustments. In (a), the particles in the remainder of the chain are shifted when the bond angle centered at particle X_{k+1} is adjusted, and in (b) only particle X_{k+1} is shifted to produce an adjustment in the bond angle.

Figure 7.6(b) depicts a perturbation of θ_k in which only the position of particle X_{k+1} is changed. By shifting particle X_{k+1} along a ray that passes through X_{k+1} and bisects the angle θ_k in the plane defined by particles X_k, X_{k+1}, and X_{k+2}, the bond angle can be adjusted to satisfy (7.12). Such a perturbation does not affect the positions of any other particles. As was the case for the bond length perturbations, rotating particles in the remainder of the chains produced better

results than this perturbation, so we will present results only for bond angle perturbations with shifts.

7.3.3 Individual Particle Perturbations

The final perturbation method we present here is based on adjustments of individual particles. Similar to the bond length perturbations, we start at one end of the chain and visit each particle, perturbing the coordinates of particle X_k if $p_k^0 \neq p_k^1$. The coordinates of particle X_k in the perturbed version are given by

$$\hat{X}_k = X_k + \delta_X \tag{7.13}$$

where $\delta_X \in \mathbb{R}$ and each element is taken from a uniform distribution on the interval $[-a_{ip}, a_{ip}]$.

This perturbation of particle X_k alters the bond lengths $r_{k-1,k}$ and $r_{k,k+1}$ as well as the angle θ_{k-1}. This type of perturbation may more realistically reflect the dynamics of individual atoms in protein, although we only perturb particles whose type differs in the template and target chains.

A possible alternative to perturbing individual particles in this way is to perturb an individual particle as well as those particles sharing a bond with it. The particles sharing a bond could be moved in the same direction as the perturbed particle but to a lesser extent. This could be carried on to subsequent particles to which those particles are bonded, damping the amount of position change for particles further away in the chain. In this way, the perturbations would more closely reflect the dynamics of chemically bonded atoms. However, further development and testing of this type of perturbation is needed to investigate the effect of using perturbations in HOPE that are more closely linked to the dynamics of real macromolecules.

7.4 Results

We present the results of two sets of computations to show the effectiveness of both HOM and HOPE in predicting the lowest energy conformations of the model proteins. These results highlight the usefulness of using the homotopy methods presented in this paper compared with some standard methods used to solve unconstrained minimization problems. Furthermore, the results show that the use of perturbations in HOPE helps increase the probability of predicting a correct target conformation over HOM.

The computations were performed using Matlab[6] under Linux on a 2.5 GHz Intel Pentium 4 processor. The potential energy function in (7.1) and the homotopy function in (7.9) were implemented in C so that the first and second derivatives could be produced using the automatic differentiation tool ADOL-C v1.8 [47]. The energy functions and their derivatives were accessed in Matlab via the MEX interface. The parameters used for the potential energy function (7.1) are given in Table 7.1. Note that distances have been scaled so that $\bar{r} = 1$ (instead of 3.8 to reflect true distances between Cα atoms). All force constants thus are scaled to reflect this choice for the unit distance.

We used the 9 chains described in the experiments of [115] for our test data. The lowest energy conformations of these chains were determined by a process combining slow cooling and simulated annealing. The sequences of the chains are shown in Table 7.2, along with the standard deviations, σ, of the hydrophobic diversity parameter, ν, and energy values of the lowest energy conformations. Recall that higher values of σ lead to greater diversity in the possible interactions between hydrophobic particles. Even though the sequences for chains F, G, and

[6]Matlab 6.5 and the Optimization Toolbox 2.2 from Mathworks, Inc.

Parameter	Description	Value
k_r	Bond length force constant	100
$\bar{r}$	Average bond length	1
k_θ	Bond angle force constant	$20/(rad)^2$
$\bar{\theta}$	Average bond angle	1.8326 rad (105°)
ε_h	Average strength of hydrophobic interactions	1
ε_L	Strength of L–L/L–B interactions	2/3

Table 7.1: Potential energy function parameters used in computations.

H are identical in terms of residues, the chains are in fact different due to the diversity in the hydrophobic-hydrophobic interactions. (Note the differences in the energies of the lowest energy conformations.) All of the chains contain $n = 22$ particles.

In order to remove the rotational and translational freedom from each conformation, we use a restricted set of $3n - 6$ variables so that

$$X = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & z_2 \\ 0 & y_3 & z_3 \\ x_4 & y_4 & z_4 \\ \vdots & \vdots & \vdots \\ x_n & y_n & z_n \end{pmatrix} \tag{7.14}$$

where $z_2, y_3, z_3 \geq 0$ and the k^{th} row of X contains the Cartesian coordinates of particle X_k. Note that all chains of particles can be rotated and translated to this coordinate system. Furthermore, the implementations of the local minimization methods tested require variables to be either scalar or vector variables, so the

Label	Sequence	σ	Minimum Energy
A	BBBBBBBBBNNNLBLBLBLBLB	0.00	-10.6509
B	BBBBBBBBNLNLNBLBBBLBLB	0.00	-10.9834
C	BBBBBBBBNLNNLBLBBBLBLB	0.30	-14.0423
D	BBBBBBBBBNNLNBLBLBLBLB	0.17	-11.8696
E	LBBBBBBBNBNLNBLBLBBBLB	0.10	-11.2465
F	LBBBBBBBBBNLNLNBLBBBLB	0.30	-15.7288
G	LBBBBBBBBBNLNLNBLBBBLB	0.30	-16.2159
H	LBBBBBBBBBNLNLNBLBBBLB	0.30	-16.3866
I	LBNBBBLBBBNNBBLBLBBBLB	0.30	-11.8513

Table 7.2: Sequences and energy values of lowest energy conformations of chains used in computations: B, hydrophobic; L, hydrophilic; and N, neutral.

$n \times 3$ matrix X above was stored in a vector of size $3n - 6$, with the elements ordered as

$$(z_2, y_3, z_3, x_4, y_4, z_4, \ldots, x_n, y_n, z_n) \ .$$

Each chain was used as a template protein (starting conformation) to predict the lowest energy conformation of the remaining 8 target proteins, yielding a total of 72 experiments. However, the lowest energy conformations match in structure for each of the following 5 template-target pairs: A–D, B–C, F–G, F–H, and G–H. Thus, there are a total of 62 experiments where the lowest energy conformation of the template chain differs from that of the target chain. Since all of the algorithms tested in our experiments were able to correctly predict the native conformations for the target chains for the 10 matching template-target pairs, we present the results for the remaining 62 experiments to highlight the

differences between the various methods.

Success of the methods was measured for each predicted conformation, X^1, against the lowest energy conformation, X^*, using two metrics—the structural overlap function from [115] and root mean-squared distance. The structural overlap function is the percentage of inter-particle distances between nonbonded particles that differ in X^1 and X^* by more than 20% of the average bond length, $\bar{r}$. It is computed as

$$\chi(X^1) = 1 - \frac{2}{n^2 - 5n + 6} \sum_{i=1}^{n-3} \sum_{j=i+3}^{n} \Theta\left(0.2\bar{r} - \left|r_{ij}^1 - r_{ij}^*\right|\right) \tag{7.15}$$

where $\Theta(\cdot)$ is the Heavyside function and r^1 and r^* are distances between particles in X^1 and X^*, respectively. Note that $\chi(X^1) \in [0, 1]$, with $\chi(X^1) = 0$ meaning that X^1 is structurally equivalent to X^*.

The root mean-squared distance is a standard metric in the protein structure prediction literature for measuring the structural similarity between two protein conformations. It is computed as

$$RMSD(X^1) = \min_{S(X^1)} \sqrt{\frac{1}{n} \sum_{i=1}^{n} \|X_i^1 - X_i^*\|^2} \tag{7.16}$$

where $S(X^1)$ is a rotation and translation of X^1. Thus, $RMSD(X^1)$ measures the distance between corresponding particles in the predicted and lowest energy conformations when they are optimally superimposed. For exact conformational matches, $RMSD(X^1) = 0$, and the value increases as the two conformations differ more in structure.

7.4.1 Computations using HOM

In the first set of experiments we compared HOM (Section 3.2), a variant of Newton's method that uses a trust region to guarantee convergence (`Newton-TR`)

[18, 19], and `QNewton-BFGS`(Section 4.1). In Matlab, the latter two methods are implemented in `sfminbx` and `fminusub`, respectively, and are called from the unconstrained minimization driver, `fminunc`. Although the only differences in performance between `Newton-TR` and `QNewton-BFGS` were in the amount of work performed by each method to predict a target conformation, the results of both methods are presented for completeness. Within HOM, `Newton-TR` was used to find local minimizers of H (Figure 3.2, Step 5).

For each computation using HOM, we set the number of steps in λ to be $m = 10$, making $\Delta\lambda = 0.1$. The minimization routine was stopped when either the change in function value between iterates dropped below 10^{-6}, the maximum change in any of the variables in X between successive iterates dropped below 10^{-12}, or the number of iterates reached the maximum number of iterations allowed (1000 for `Newton-TR` and `QNewton-BFGS` and 60 for each minimization performed in HOM).

Table 7.3 shows the results of the 62 experiments for the different methods. The first column lists the method. The second column shows the number of computations in which $\chi(X^1) = 0$ i.e., the method predicted the correct conformation. HOM has a success rate of 24% (column 3), almost four times better than `Newton-TR` and `QNewton-BFGS`. Columns 4 and 5 present the average structural overlap, $\overline{\chi}$, and root mean-squared distance, $\overline{RMSD}$, respectively, of the 62 computations. These results show that HOM predicts better structures than `Newton-TR` and `QNewton-BFGS` on average, even when an exact match of the lowest energy conformation was not predicted. The main drawback for HOM, at least with respect to `Newton-TR`, is that it is more computationally expensive. The last three columns in the table show $\overline{t}$, the average clock time used in seconds;

Method	$\chi = 0$	Success Rate (%)	$\overline{\chi}$	$\overline{RMSD}$	$\overline{t}$	$\overline{N}_f$	$\overline{N}_{cg}$
HOM	15	24	0.36	0.38	10	152	4809
`Newton-TR`	4	6	0.45	0.55	1	20	523
`QNewton-BFGS`	4	6	0.45	0.55	13	362	—

Table 7.3: Comparison of prediction results using HOM, `Newton-TR`, and `QNewton-BFGS`.

$\overline{N}_f$, the average number of function evaluations; and $\overline{N}_{cg}$, the average number of conjugate gradient iterations per computation. Note that `QNewton-BFGS` does not use the conjugate gradient method in determining search directions. Clearly, `Newton-TR` performs the least amount of work in all three of these measures, with HOM falling in between `Newton-TR` and `QNewton-BFGS`. Each run of `QNewton-BFGS` first computes an approximation of the inverse of the Hessian of the potential energy function using finite differences of gradients, and thus is more expensive than the other methods. Even though HOM requires more work than `Newton-TR`, the trade-off in success rate shows the benefit of using a homotopy method over one of the more common minimization algorithms.

7.4.2 Computations using HOPE

In the second set of computations we compared HOPE and ensemble-based, basin-hopping simulated annealing (`Basin-SA`)—a combination of the methods of [116] and [94]. `Basin-SA` was implemented using SA Tools v1.03 [94].

In `Basin-SA` the move class, the set of possible conformations produced by perturbing a given conformation, consists solely of local minimizers of $E^1(X)$.

Specifically, candidate conformations are found using `Newton-TR` started at a perturbed version of each conformation in the ensemble. This allowed us to use the same perturbations specified in Section 7.3 and used in HOPE.

We chose to allow both methods to compute an equivalent number of local minimizers in the course of each computation. Specifically, we tested HOPE using $m = 10$ steps in λ and maximum ensemble sizes of $c_{max} = 2, 4, 8,$ and 16. Also, we set $\hat{c} = 1$, allowing only one perturbed version to be generated for each ensemble conformation. This yielded upper limits of $20, 40, 80,$ and 160 local minimizers to be computed for the values of c_{max}, respectively. Recall that there may be fewer local minimizers computed than this upper limit, as only unique conformations are carried from one iteration of HOPE to the next. `Newton-TR` as parameterized in the previous section was used for local minimization in HOPE.

In `Basin-SA`, we used ensembles of size c_{max} at each of $m = 10$ steps of a constant speed annealing schedule[7] [57], starting at $T = 10^5$. Therefore, the number of calls to `Newton-TR` in `Basin-SA` matched the corresponding upper limit of those allowed in HOPE. The main difference between the methods is the function being minimized—the homotopy function in HOPE and the potential energy function of the target chain in `Basin-SA`.

Computations were performed for both methods using the perturbations in Section 7.3, with maximum amounts of perturbation $a_r = 1$, $a_\theta = 40°$, and $a_{ip} = 1$. It is typical in simulated annealing methods that the perturbations allowed in the move class be functions of the temperature, T, defined by the annealing schedule. Thus, we designed a function for the maximum perturbation

[7]The `berkeley` schedule in SA Tools implements this schedule (all default values were used).

satisfying

$$\lim_{T \to 0} p_{max}(T) = 0 \ , \tag{7.17}$$

$$\lim_{T \to \infty} p_{max}(T) = \{a_r, a_\theta, a_{ip}\} \ . \tag{7.18}$$

The following function was used in the `Basin-SA` computations:

$$p_{max} = \alpha e^{-\frac{1}{\gamma T}}$$

where $\gamma \in (0, 1)$ and $\alpha = a_r, a_\theta$ or a_{ip}, depending on the perturbation method used. Note that this function passes through the point $(\hat{T}, \hat{p})$ when

$$\gamma = -\frac{1}{\hat{T} \ln(\hat{p}/\alpha)} \ .$$

The point $(\hat{T}, \hat{p}) = (1, 0.9)$ is used in `Basin-SA` so that the amount of maximum perturbation allowed in `Basin-SA` is equivalent to that allowed in HOPE (either a_r, a_θ, or a_{ip}) for almost all values of T except those very close to zero.

Table 7.4 shows the results of the 62 experiments for HOPE and `Basin-SA` using the bond length perturbations with shifts and individual particle perturbations. Results for the two methods using bond angle perturbations were very similar to those using the bond length perturbations. Thus, results for those experiments are not presented here, so as to highlight the differences between the other types of perturbations.

Results for the computations using bond length perturbations appear in the top half of the table and those for the individual particle perturbations appear in the bottom half of the table. Due to the use of the stochastic perturbations, each experiment was performed 10 times, and the results presented are the averages over these 10 runs.

Computations using Bond Length Perturbations

Method	c_{max}	$\chi = 0$	Success Rate (%)	$\overline{\chi}$	$\overline{RMSD}$	$\overline{t}$	$\overline{N}_f$	$\overline{N}_{cg}$
HOPE	2	33.4	54	0.14	0.17	35	539	12369
	4	43.1	70	0.08	0.11	65	992	22678
	8	54.6	88	0.03	0.04	115	1732	39642
	16	**59.0**	**95**	**0.01**	**0.02**	**200**	**2981**	**68381**
`Basin-SA`	2	13.1	21	0.27	0.36	52	753	17170
	4	20.8	34	0.19	0.26	107	1576	35528
	8	28.5	46	0.13	0.19	229	3174	71893
	16	**40.2**	**65**	**0.08**	**0.12**	**434**	**6358**	**143660**

Computations using Individual Particle Perturbations

HOPE	2	18.4	30	0.32	0.34	26	391	10284
	4	22.3	36	0.30	0.31	36	541	14163
	8	22.4	36	0.29	0.30	43	629	16513
	16	23.5	38	0.28	0.30	46	679	17733
`Basin-SA`	2	8.9	14	0.41	0.48	27	348	9309
	4	11.8	19	0.38	0.43	57	744	19843
	8	13.5	22	0.35	0.38	116	1546	40995
	16	19.3	31	0.30	0.31	236	3167	83724

Table 7.4: Comparison of prediction results using HOPE and `Basin-SA` (averaged over 10 runs). The lines in bold highlight the best prediction results for each of the algorithms.

Presenting the results for increasing values of c_{max} in Table 7.4 shows the trends of the success of and computational effort required by each method as

more local minimizers are generated. Note that the best results for HOPE and `Basin-SA` (shown in bold) are those that use bond length perturbations and the largest ensemble size, $c_{max} = 16$. Figure 7.7 shows plots of the success rates as a function of c_{max}. Clearly, HOPE using bond length perturbations outperforms all of the other methods in terms of success for each value of c_{max}, monotonically increasing to an average success rate of 95% at $c_{max} = 16$. We can also see that the computations using bond length perturbations perform better than the individual particle perturbation. Note that the success rates as functions of c_{max} for the two methods using bond length perturbations appear to be converging to a different value than those for the methods using individual particle perturbations. This suggests that qualitative differences in perturbation methods may lead to quantitatively different upper bounds on success rates. Such results lead us to believe that future work should include a more detailed analysis of the effects of perturbations in HOPE.

Figure 7.8 plots $\overline{RMSD}$, the average $RMSD(X^1)$, showing that HOPE using bond length perturbations clearly outperforms all other methods. Even though the average success rate tops out at 95%, HOPE performs very well even in the cases where the lowest energy conformation was not found, with $\overline{RMSD} = 0.02$ (and $\overline{\chi} = 0.01$) for $c_{max} = 16$. Note that the results for the different types of perturbations tend to be converging to different values, as was the case for the success rates. The success rates and these average measures of structural similarity results are not necessarily dependent on one another (e.g., it is possible to consistently perform extremely poorly in one experiment and produce increasingly better overall success rates while $\overline{RMSD}$ remains relatively high due to that one experiment). However the results show that these measures are correlated,

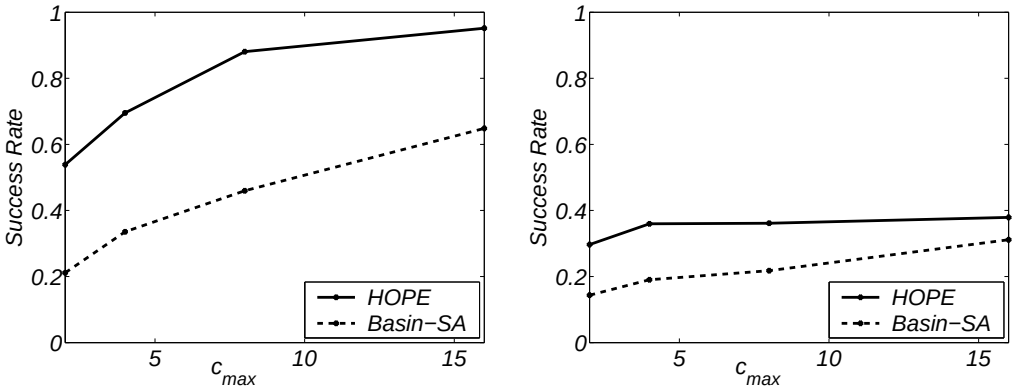


Figure 7.7: Success rates using (a) bond length perturbations and (b) individual particle perturbations.

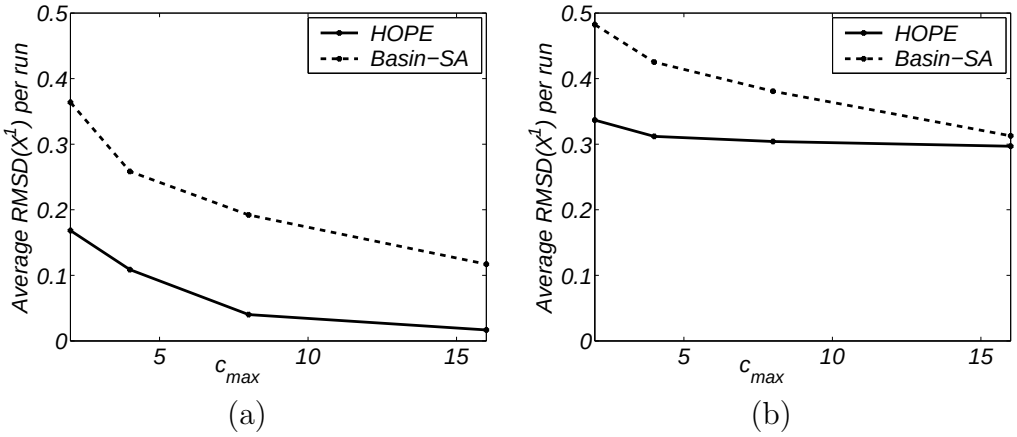


Figure 7.8: Average $RMSD(X^1)$ value per experiment using (a) bond length perturbations and (b) individual particle perturbations.

leading us to believe that our methods are robust.

The computational effort required by the different methods is presented in the last three columns of Table 7.4 showing that HOPE required less time and computational effort than `Basin-SA` to produce better results. Furthermore, we can see in Figures 7.9 and 7.10 that the amount of computational time follows the same trend as the number of function evaluations, with the increase in cost as c_{max} increases being greater for `Basin-SA` than for HOPE. These differences

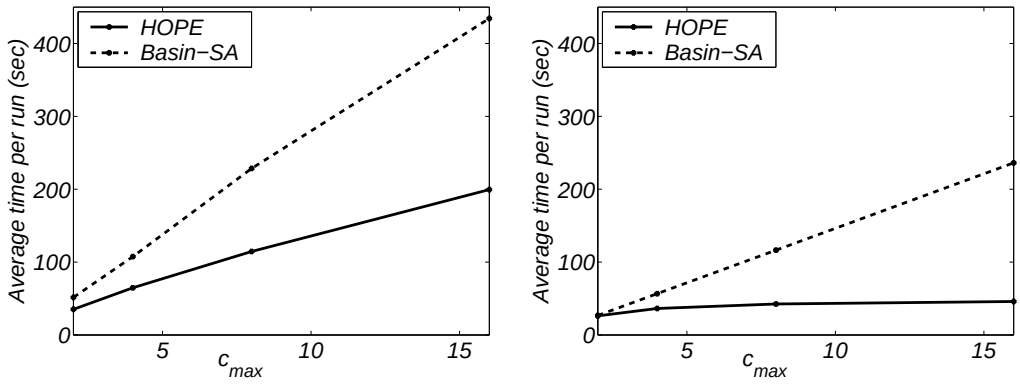


Figure 7.9: Average wall clock time in seconds per experiment using (a) bond length perturbations and (b) individual particle perturbations.

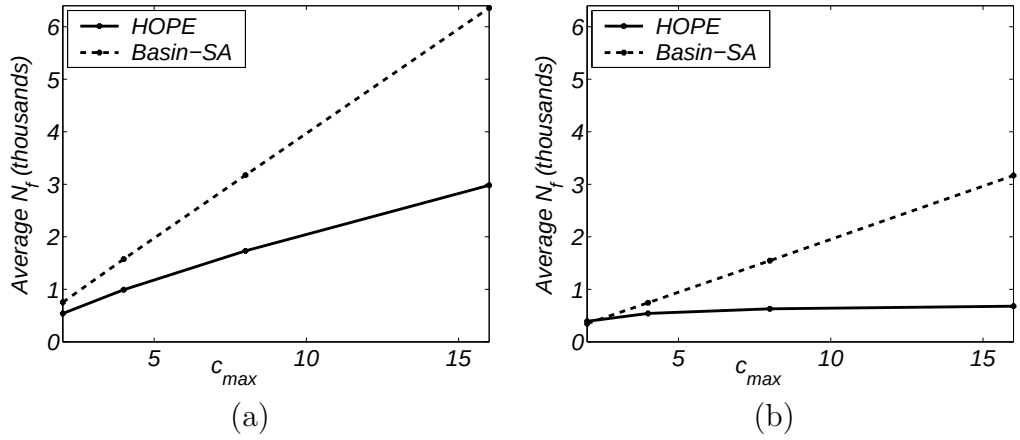


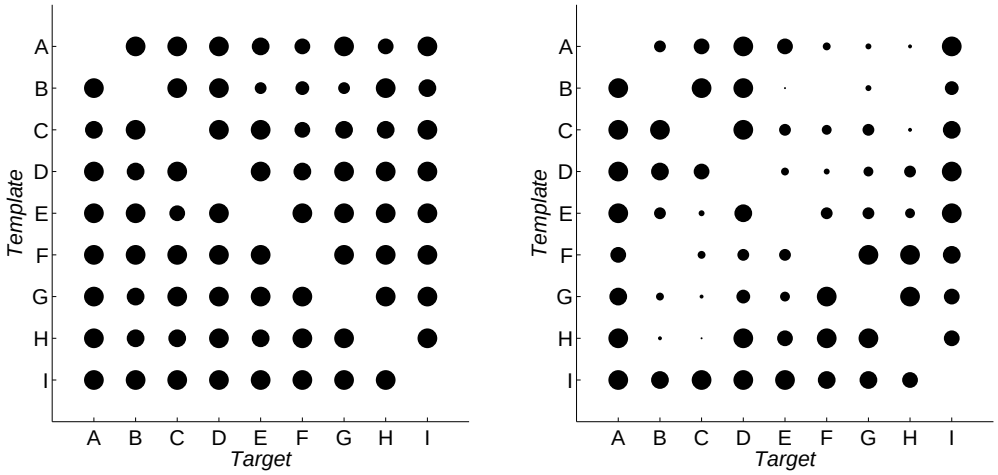
Figure 7.10: Average number of function evaluations per experiment using (a) bond length perturbations and (b) individual particle perturbations.

are most likely due to the fact that in the earliest iterations in HOPE, fewer than c_{max} conformations are used (recall that $\hat{c} = 1$). Note that for `Basin-SA` using individual particle perturbations there is not much increase in the average amount of effort required to complete the experiments. This suggests that the amount of perturbation used for these computations, $a_{ip} = 1$, did not produce enough change between the original conformations and their perturbed versions to lead to unique local minimizers.

HOPE was the only method that successfully predicted the native conformations in at least one of the 10 runs performed for all of the 62 computations. Figure 7.11 presents the results with $c_{max} = 16$ for each of the template-target pairs using (a) HOPE and (b) `Basin-SA`. The size of each circle represents the percentage of runs where a target was successfully predicted starting at a given template. HOPE predicted the correct target conformations in all 10 runs for 44/62 (71%) template-target pairs. Moreover, HOPE predicted the correct target conformation in 60% or more of the runs for each template-target. In contrast, `Basin-SA` predicted all 10 target conformations correctly for only 14/62 (23%) pairs. More importantly, `Basin-SA` was not able to correctly predict the target conformations for 3 pairs (B–F, B–H, and F–B) in any of the 10 runs.

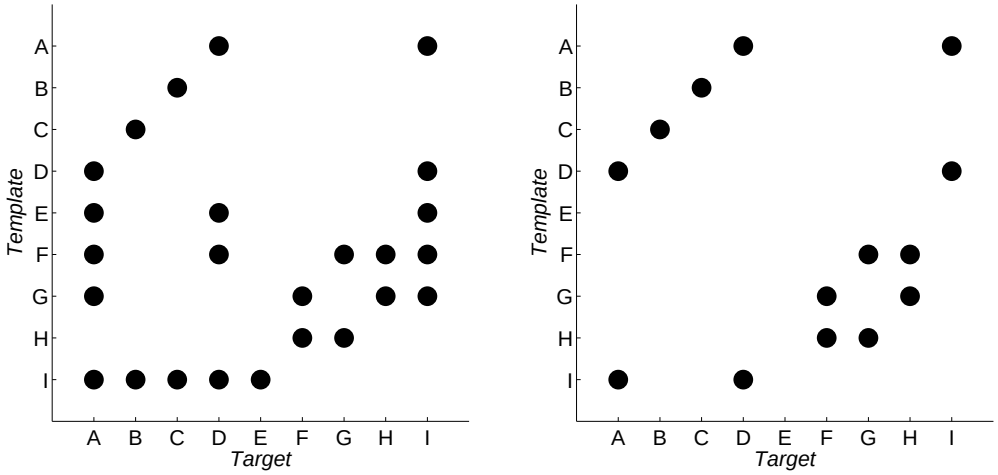
HOPE was also more successful than HOM (and thus `Newton-TR`) in predicting native conformations. Figure 7.12 presents the results using HOM and `Newton-TR` of all 72 experiments, where a circle represents a successful prediction of a native conformation of the target starting from that of the template. Counting only the template-target pairs for which HOPE predicted the correct target conformations (44/62), HOPE was almost three times effective as HOM (15/62). However, the most significant advantage of HOPE over HOM is HOPE's ability to correctly predict the native conformations for all template-target pairs.

HOPE is computationally more demanding that HOM, but the prediction results justify the use of HOPE for this protein model. HOPE correctly predicted all of the native target conformation in at least 60% of the runs for each template-target. However, this improvement required almost 20 times more computational effort, with ratios of 200/10 seconds in computation time and 2981/152 function evaluations required for a run of HOPE compared to HOM. The ratio of success



(a) (b)

Figure 7.11: Success of (a) HOPE and (b) `Basin-SA` using bond length perturbations with $c_{max} = 16$ for each template-target pair. The size of each circle represents the percentage of successful predictions over 10 runs.



(a) (b)

Figure 7.12: Success of (a) HOM and (b) `Newton-TR`.

of HOPE versus HOM to the amount of computation required to achieve that success will most likely differ from problem to problem. The results here, though demonstrate that the use of perturbations and ensembles increase the effectiveness of these homotopy optimization methods.

Chapter 8

Conclusions and Further Study

We have presented HOPE, a new method for solving unconstrained minimization problems. The results of experiments on standard test problems and in solving the protein folding problem using simplified protein models demonstrates that HOPE is an effective method for finding global minimizers. Although this performance comes at a greater computational cost than is typical for local methods and for the related HOM algorithm, there are several advantages in using HOPE over these other methods. The generation of ensembles of local minimizers of a function, the flexibility to control the amount of computation using several different parameters, and the ability to take advantage of structure inherent in the minimization problem are three important benefits.

We demonstrated the flexibility of HOPE in solving several different types of problems—ranging from simple test problems in one or two dimensions (`Nmod`), to more challenging randomly generated test functions (`Pint`), to complicated application-driven functions (the potential energy functions in the protein models). A wide variety of homotopy functions were used in HOPE, with some very generic and others designed to take advantage of specific features of the problem. The results, though, show that HOPE is effective even when very little domain-

specific knowledge of the problem is known or incorporated in the homotopy function.

We have shown that for simplified models of proteins HOPE is successful at predicting native conformations. Moreover, we demonstrated that HOPE was more successful is solving these problems than several standard minimization methods—`QNewton-BFGS`, `Newton-TR`, and simulated annealing. We conjecture that the success of HOPE in solving these problems is due to the combined use of homotopy functions, perturbations, and ensembles of points. The homotopy functions used in the experiments are capable of exploiting the possible structural similarities of sequence-related proteins, and as noted earlier, exploiting such properties is necessary for solving the protein folding problem efficiently. We used perturbations for the exploration of structurally similar conformations and to avoid conformations corresponding to local minimizers of the potential energy function. The use of ensembles of conformations allowed for parallel searches of the conformational space, sometimes in very distant regions (in terms of structure). We note that there is an ongoing debate in the scientific community whether native conformations of proteins correspond to minimal energy structures or structures with relatively low energy that are kinetically the most favorable. Using HOPE to generate ensembles of low energy candidate structures may prove useful for investigating these claims.

Future work in developing suitable homotopies for other application areas would be an important step in demonstrating the power of using a combination of function/problem deformation, stochastic perturbations, and working with ensembles, or populations, of possible solutions. Applying HOPE in another application area would help demonstrate its use as a general-purpose minimization

method as well.

For most of the experiments, HOPE performed considerably more computation than other methods. In most cases, the increase in cost was matched to some extent with an increase in performance. An important feature of HOPE is that it is inherently parallelizable and communication between processors would only be required in initializing the iterations for each new value of λ. Thus, another goal would be to implement HOPE as a parallel method, with parallelization focused on balancing the distribution of ensemble members across computational nodes. As even small proteins have a large number of degrees of freedom, implementing HOPE for large-scale problems would allow for work on more detailed models of the protein folding problem.

Several of the parameters used in HOPE could be determined adaptively, leading to more efficient use of computational resources and faster convergence for some problems. In the experiments described in this dissertation, each step in λ was fixed at $1/m$, where m was the number of steps to be taken. The value of $\Delta\lambda$ could be determined adaptively—e.g. as a function of the number of local minimization iterations required before convergence or based on the changes in homotopy function values from one step to the next. This may allow for larger (and thus fewer) steps in λ with no significant decrease in the success rate. Investigation into the use of adaptive steps in λ (as described in [5]), as well as into adaptively determining the amount of perturbation, number of perturbed versions of ensemble members to generate, and the maximum ensemble size may lead to more optimal implementations of HOPE.

Another consideration in using HOPE is the choice of starting points for the local minimization method. In this work, the ensemble members carried forward

from the previous iteration were used as starting points. However, we could extrapolate from these starting points using derivative information to produce starting points that may lead to fewer iterations in the local minimization [5].

There is great potential for HOPE in other areas of optimization as well. The HOPE algorithm presented here was designed to solve unconstrained minimization problems. However, by replacing the local minimization method with a method for solving a different problem, HOPE can be extended for solving other optimization problems. For example, the local minimization method could be replaced by a method for constrained minimization, leading to a variant of HOPE for solving minimization problems that are subject to constraints.

A major challenge is developing useful homotopies for template and target functions whose domains differ. (In all of the experiments presented in this dissertation, the dimensions of the template and target function domains were the same.) The availability of such homotopies would dramatically increase the number of problems to which HOPE could be applied. For example, there are many families of proteins whose members have structurally similar native conformations but whose amino acid sequences differ in length. A homotopy designed to work with template and target proteins with different numbers of constituent atoms (or just amino acids when using a backbone model of proteins) would allow HOPE to be applied to pairs of proteins from such families.

HOPE is a promising method for unconstrained minimization. There is much work to be done in terms of performance analysis, optimal parameter choices, and applicability to other application areas. However, based on our results we expect that HOPE can be used to lend insight on minimization problems in general and the protein folding problem in particular.

Appendix A

Native Conformations of Chains of Charged Particles

Charges	θ_1	θ_2	$\mathbf{E}(\theta)$
$++++$	$+2.3911$	$+2.3736$	$+0.5954$
$-+++$	$+1.4835$	$+3.7176$	-0.8177
$+-++$	$+2.6005$	$+4.7473$	-0.2500
$--++$	$+1.6755$	$+4.6251$	-1.5677
$++-+$	$+1.5359$	$+3.6826$	-0.2500
$-+-+$	$+2.3562$	$+2.3387$	$+0.0454$
$+--+$	$+1.6930$	$+4.5902$	-0.9957
$---+$	$+2.5656$	$+4.7997$	-0.8177

Table A.1: Charges on the $n = 4$ chains from Table 6.3.

Charges	θ_1	θ_2	θ_3	$\mathbf{E}(\theta)$
$+\,+\,+\,+\,+$	$+2.2340$	$+2.4435$	$+2.2340$	$+0.7412$
$-\,+\,+\,+\,+$	$+1.6057$	$+3.2987$	$+1.7802$	-1.1694
$+\,-\,+\,+\,+$	$+2.5482$	$+2.1118$	$+2.4958$	-0.5573
$-\,-\,+\,+\,+$	$+1.4312$	$+3.8397$	$+1.5533$	-2.4002
$+\,+\,-\,+\,+$	$+1.5184$	$+3.7874$	$+1.5184$	-0.9723
$-\,+\,-\,+\,+$	$+1.7453$	$+3.2289$	$+1.6930$	-1.1480
$+\,-\,-\,+\,+$	$+1.4661$	$+4.0492$	$+1.4312$	-2.2651
$-\,-\,-\,+\,+$	$+1.5533$	$+3.8397$	$+1.4312$	-2.4002
$+\,+\,+\,-\,+$	$+2.4958$	$+2.1118$	$+2.5482$	-0.5573
$-\,+\,+\,-\,+$	$+1.5010$	$+3.6477$	$+1.6232$	-1.2737
$+\,-\,+\,-\,+$	$+2.2864$	$+2.3562$	$+2.2864$	-0.3813
$-\,-\,+\,-\,+$	$+1.6930$	$+3.2289$	$+1.7453$	-1.1480
$+\,+\,-\,-\,+$	$+1.4312$	$+4.0492$	$+1.4661$	-2.2651
$-\,+\,-\,-\,+$	$+1.6232$	$+3.6477$	$+1.5010$	-1.2737
$+\,-\,-\,-\,+$	$+1.4661$	$+3.9095$	$+1.4661$	-2.1061
$-\,-\,-\,-\,+$	$+1.7802$	$+3.2987$	$+1.6057$	-1.1694

Table A.2: Charges on the $n = 5$ chains from Table 6.3.

Charges	θ_1	θ_2	θ_3	θ_4	$\mathbf{E}(\theta)$
$++++++$	$+2.3911$	$+2.3213$	$+2.3213$	$+2.4086$	$+0.7744$
$-+++++$	$+2.0595$	$+2.6005$	$+2.1118$	$+2.6005$	-1.6281
$+-+++++$	$+2.7227$	$+2.0071$	$+2.6005$	$+2.1817$	-1.0903
$--+++++$	$+2.4958$	$+2.1468$	$+2.4435$	$+2.3387$	-3.4178
$++-+++$	$+1.9199$	$+2.8972$	$+1.8850$	$+2.7751$	-1.3144
$-+-+++$	$+2.1118$	$+2.6354$	$+2.0246$	$+2.6529$	-2.1922
$+--+++$	$+2.2166$	$+2.5831$	$+2.0420$	$+2.6180$	-3.0511
$---+++$	$+2.4260$	$+2.2864$	$+2.2864$	$+2.4260$	-3.9775
$+++-++$	$+2.7751$	$+1.8850$	$+2.8972$	$+1.9199$	-1.3144
$-++-++$	$+2.4958$	$+2.1468$	$+2.5133$	$+2.2340$	-2.4835
$+-+-++$	$+2.6878$	$+2.0071$	$+2.6878$	$+2.0595$	-1.6246
$--+-++$	$+2.3736$	$+2.3213$	$+2.3038$	$+2.3911$	-2.8356
$++--++$	$+2.0246$	$+2.7053$	$+2.0420$	$+2.6180$	-3.2272
$-+--++$	$+2.4609$	$+2.2515$	$+2.3736$	$+2.3213$	-3.0394
$+---++$	$+2.1118$	$+2.6529$	$+2.0246$	$+2.6354$	-3.5946
$----++$	$+2.3387$	$+2.4435$	$+2.1468$	$+2.4958$	-3.4178
$+++++-+$	$+2.1817$	$+2.6005$	$+2.0071$	$+2.7227$	-1.0903
$-+++-+$	$+2.0246$	$+2.7227$	$+1.9722$	$+2.7227$	-2.4153
$+-++-+$	$+2.4435$	$+2.2864$	$+2.3038$	$+2.4260$	-1.7281
$--++-+$	$+2.3213$	$+2.3736$	$+2.2515$	$+2.4609$	-3.0394
$++-+-+$	$+2.0595$	$+2.6878$	$+2.0071$	$+2.6878$	-1.6246
$-+-+-+$	$+2.3911$	$+2.3038$	$+2.3213$	$+2.3911$	-1.4619
$+--+-+$	$+2.4435$	$+2.2689$	$+2.3562$	$+2.3562$	-2.2621
$---+-+$	$+2.6529$	$+2.0246$	$+2.6354$	$+2.1118$	-2.1922
$+++--+$	$+2.6180$	$+2.0420$	$+2.5831$	$+2.2166$	-3.0511
$-++--+$	$+2.4086$	$+2.3038$	$+2.3038$	$+2.4086$	-3.2446
$+-+--+$	$+2.3562$	$+2.3562$	$+2.2689$	$+2.4435$	-2.2621
$--+--+$	$+2.2340$	$+2.5133$	$+2.1468$	$+2.4958$	-2.4835
$++---+$	$+2.6354$	$+2.0246$	$+2.6529$	$+2.1118$	-3.5946
$-+---+$	$+2.7227$	$+1.9722$	$+2.7227$	$+2.0246$	-2.4153
$+----+$	$+2.3387$	$+2.3387$	$+2.3562$	$+2.3213$	-2.8114
$-----+$	$+2.6005$	$+2.1118$	$+2.6005$	$+2.0595$	-1.6281

Table A.3: Charges on the $n = 6$ chains from Table 6.3.

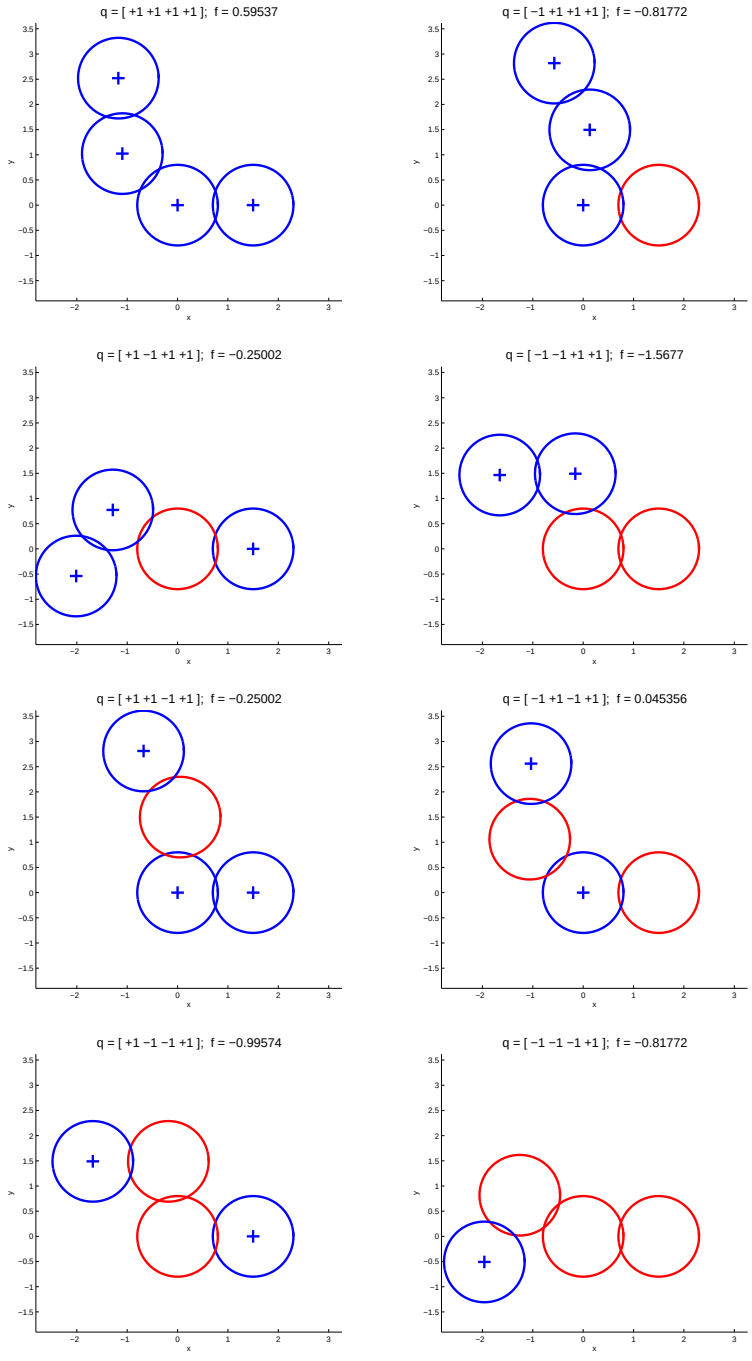


Figure A.1: Native conformations of chains of $n = 4$ particles.

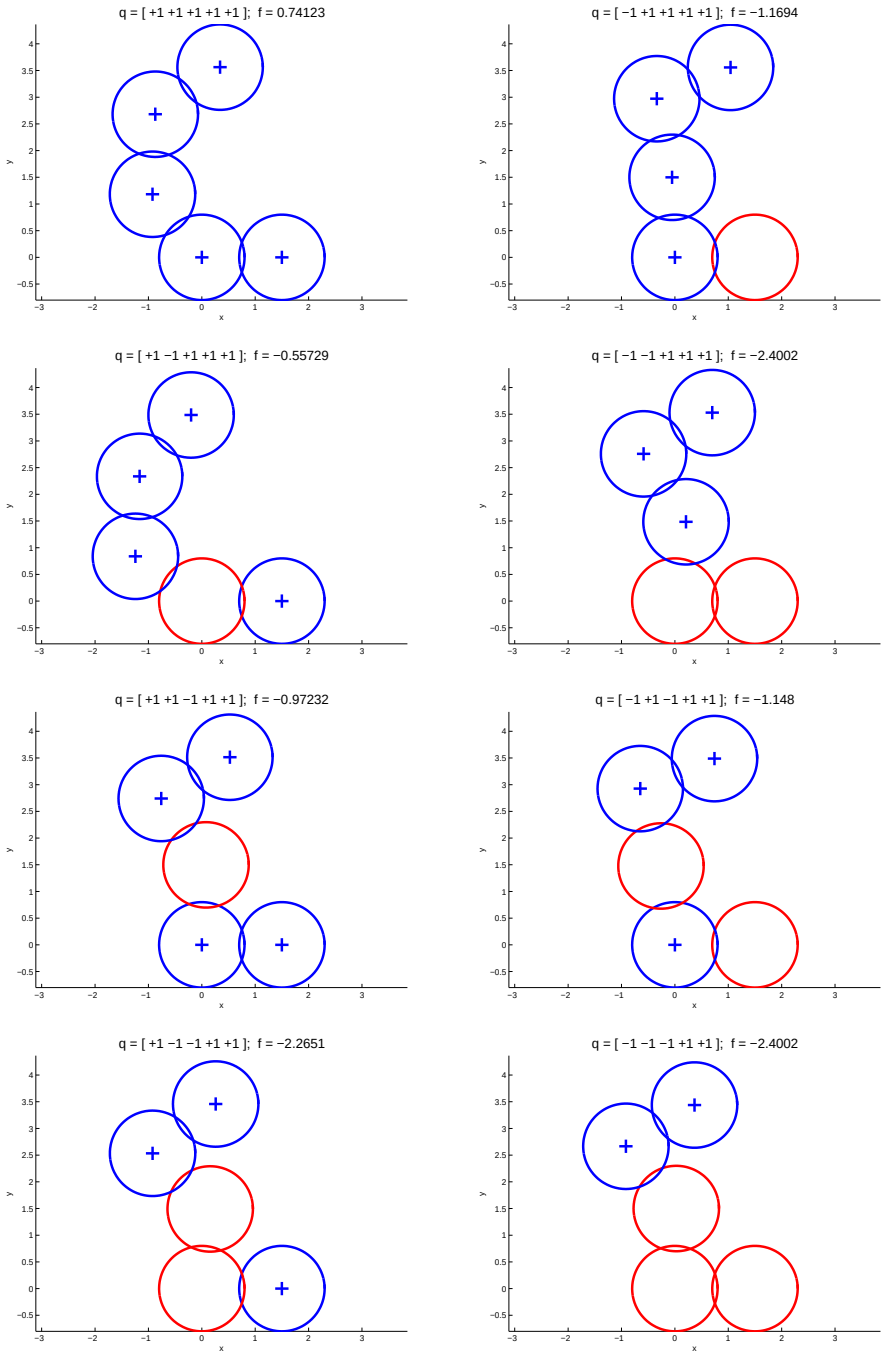


Figure A.2: Native conformations of first 8 chains of $n = 5$ particles.

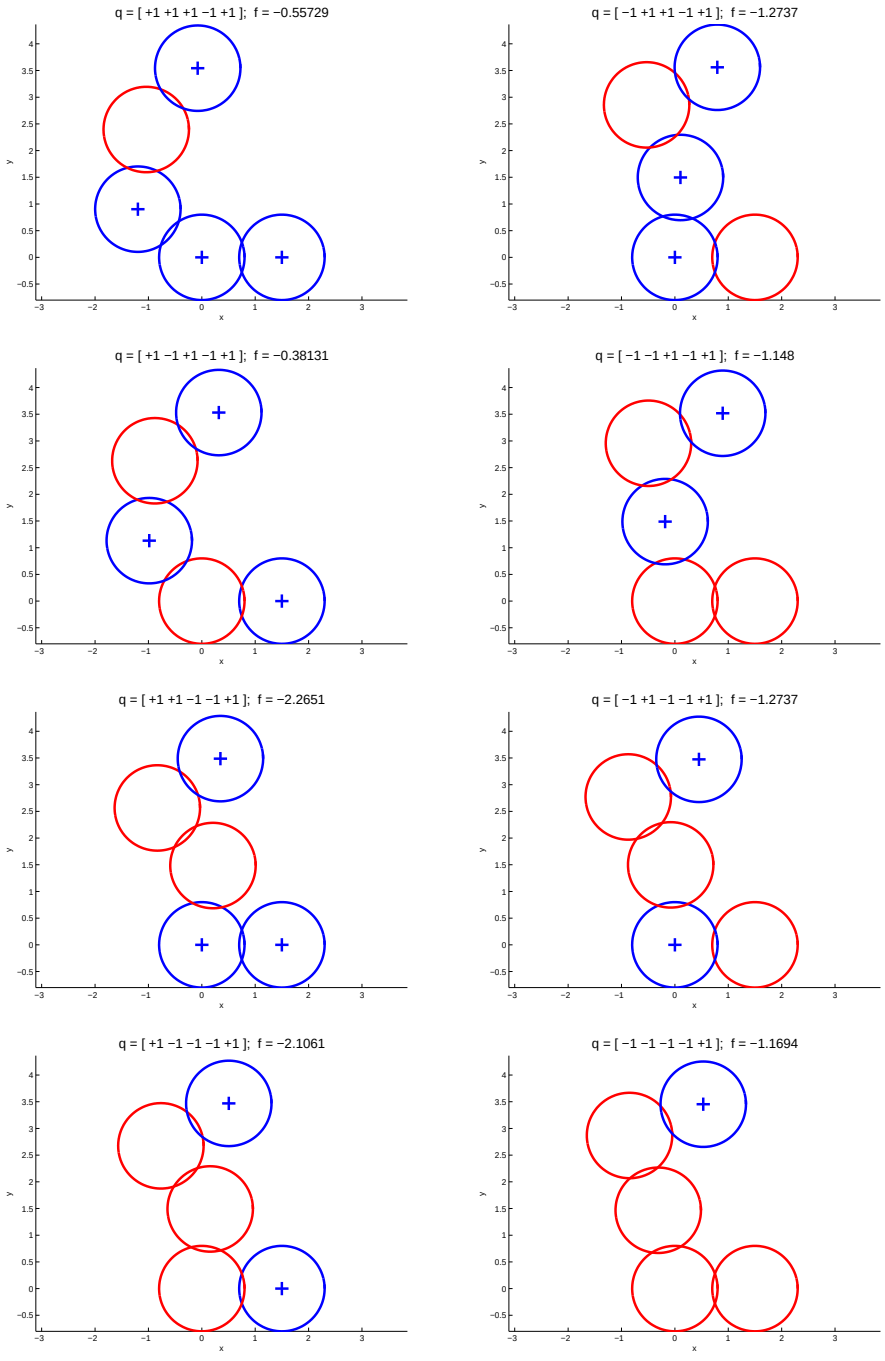


Figure A.3: Native conformations of last 8 chains of $n = 5$ particles.

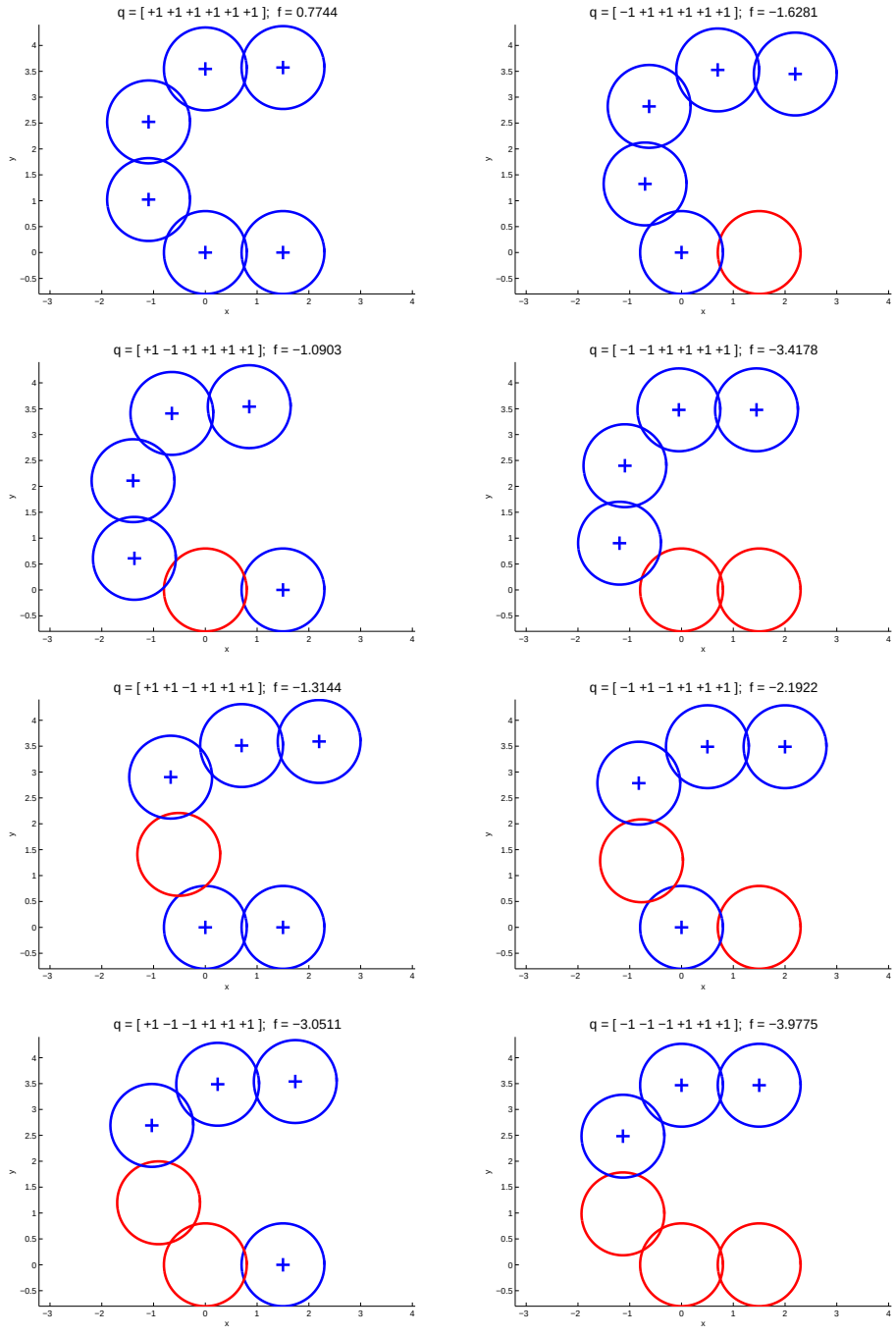


Figure A.4: Native conformations of first 8 chains of $n = 6$ particles.

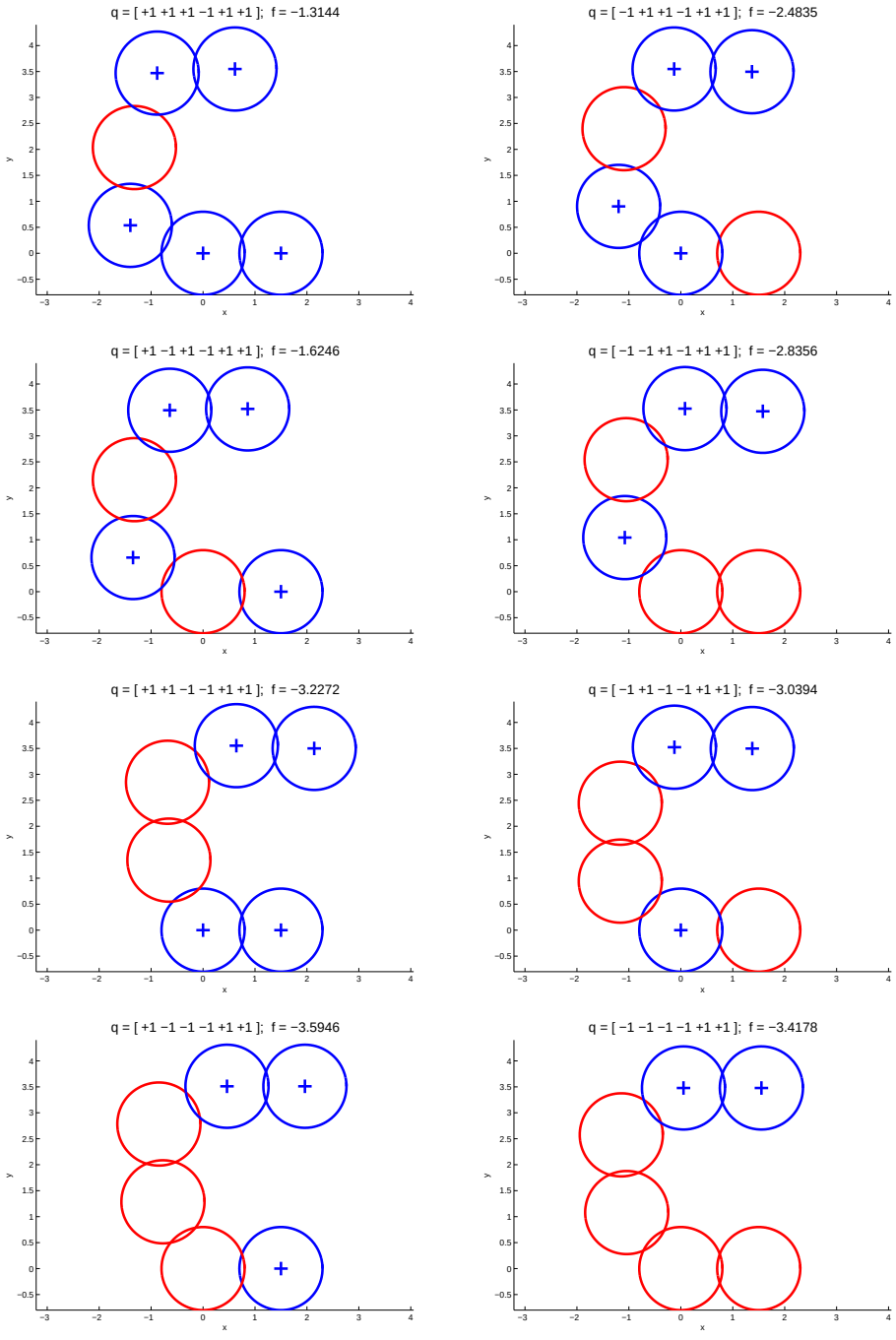


Figure A.5: Native conformations of next 8 chains of $n = 6$ particles.

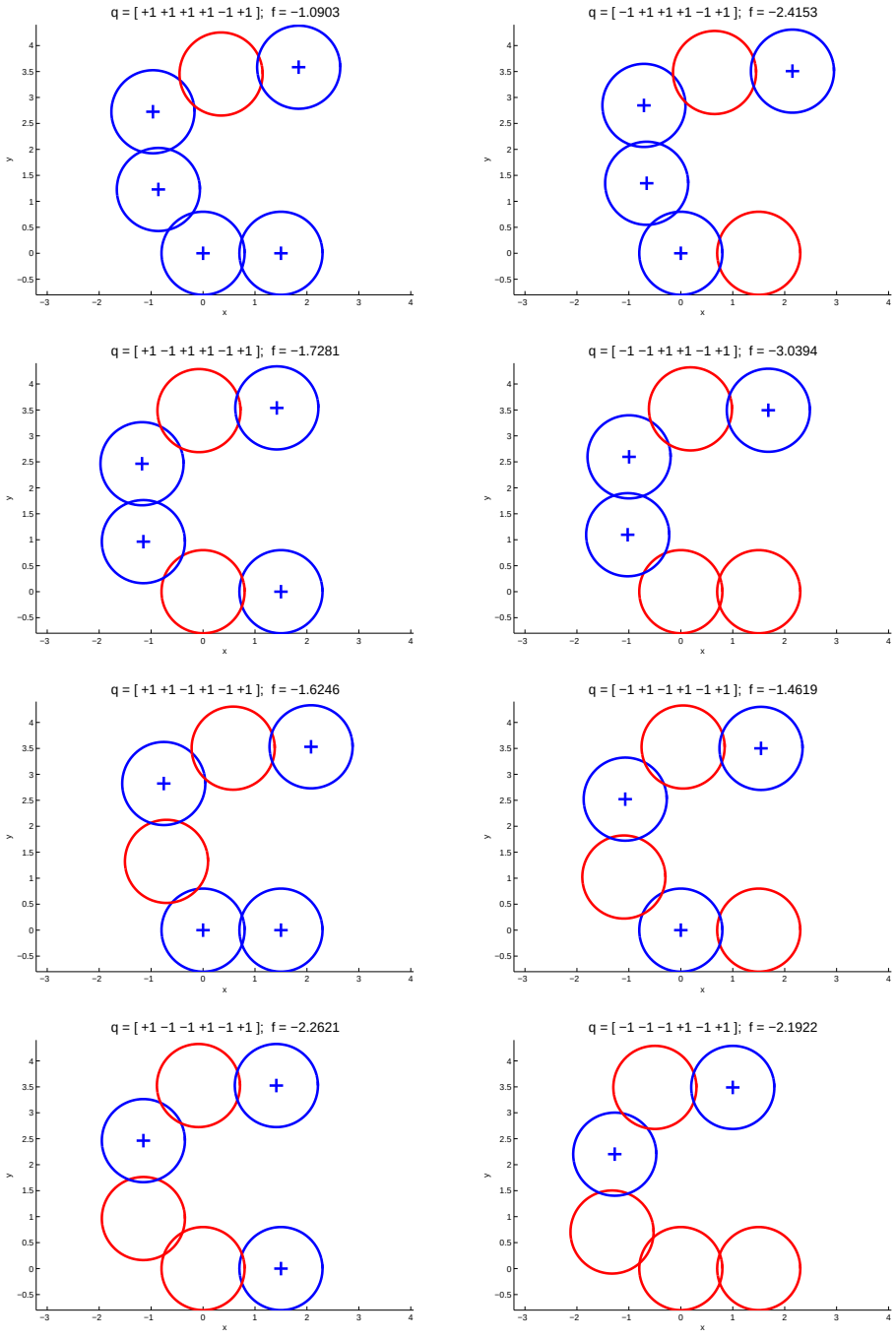


Figure A.6: Native conformations of next 8 chains of $n = 6$ particles.

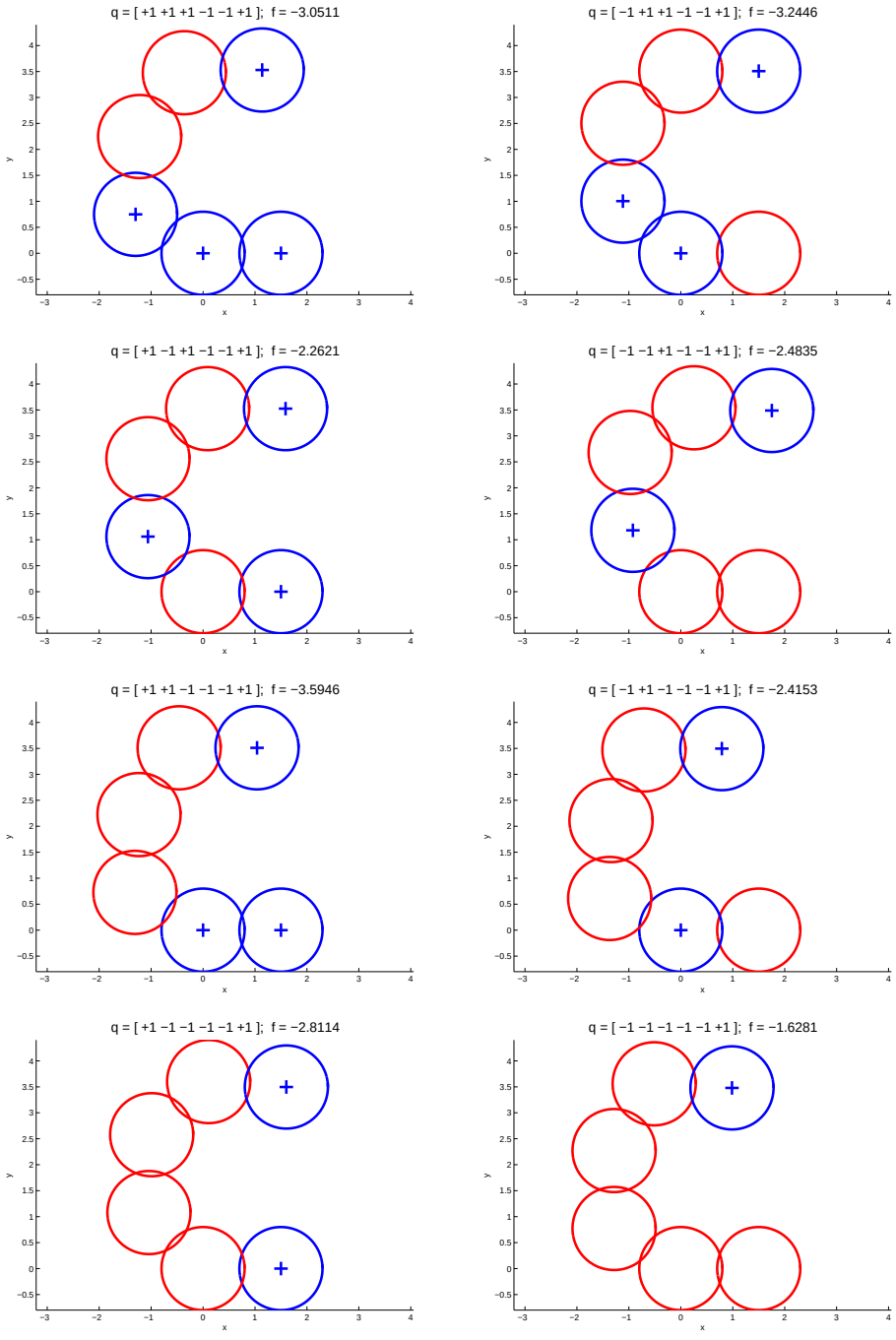


Figure A.7: Native conformations of last 8 chains of $n = 6$ particles.

Appendix B

Chains used in Experiments on Larger Chain

$\tilde{\mathbf{n}}$	Charges	HOPE $\mathbf{E}(\theta^1)$
Template	− − + − + − + + − + + − + − + + − + −+	
2	− − + − + − + + − + + − − + + + − + −+	−20.291
3	− − − − + − + + − − + − + − + + − + ++	−21.685
4	− − − + + − + − − + + − + − + − − + −+	−21.154
5	+ − + − + + + + − − + − + − + − − + ++	−19.748
6	− − + − − + − + − + − − − − + + − − −+	−18.788
7	− − + + − − + + − + − − − + + − − − −+	−20.825
8	− − − − − − + + − − + + − − + − + − −+	−19.870
9	+ − − + + + + − − + + + + + + + − + +−	−17.780
10	− − − + − − + − + + − − − − − + + − −+	−19.377
11	+ − + + − − − − + + + − + + + − + − −−	−23.578
12	+ − − − − − + − − − − − − + + − − − +−	−13.844
13	+ + − − + + − − + + − + + + + + + + +−	−17.933
14	+ + + + + + − + − − − + − − − + + − +−	−20.589
15	+ + − + − + − + − + − + − + − − + + ++	−18.238
16	+ + + + − + − − − − + + − + − − + − ++	−20.090
17	+ + − + − + − − + − − + − + + − − − −−	−18.475
18	+ + − + − + − − − − − + − + − − + + +−	−19.191
19	+ + + + − + − − + − − + − + − − + − +−	−20.271
20	+ + − + − + − − + − − + − + − − + − +−	−19.220

Table B.1: Charges on the chains of results presented in Table 6.5. The energy of the conformation predicted by HOPE is repeated here for reference.

BIBLIOGRAPHY

[1] S. Ackermann and W. Kliesch. Computation of stationary points via a homotopy method. *Theor. Chem. Acc.*, 99:255–264, 1998.

[2] S. A. Adcock. Accurate reconstruction of peptide backbone using dead-end elimination and a knowledge-based forcefield. *J. Comput. Chem.*, 25:16–27, 2004.

[3] B. Addis and S. Leyffer. A trust-region algorithm for global optimization. Technical Report MCS-P1190-0804, Argonne National Laboratory, 2004.

[4] B. Addis, M. Locatelli, and F. Schoen. Local optima smoothing for global optimization. Technical Report DSI5-2003, Dipartimento di Sistemi e Informatica, Universit degli Studi di Firenze, Firenze, Italy, 2003.

[5] E. L. Allgower and K. Georg. *Introduction to Numerical Continuation Methods*. SIAM, Philadelphia, PA, 2003.

[6] N. L. Allinger, Y. H. Yuh, and J.-H. Lii. The MM3 force field for hydrocarbons. *J. Am. Chem. Soc.*, 111:8551–8566, 1989.

[7] C. B. Anfinsen. Principles that govern the folding of protein chains. *Science*, 181(4096):223–230, 1973.

[8] C. B. Anfinsen, E. Haber, M. Sela, and F. H. White. The kinetics of formation of native ribonuclease during oxidation of the reduced polypeptide chain. *P. Natl. Acad. Sci. U.S.A.*, 47(9):1309–1314, 1961.

[9] T. Asselmeyer, W. Ebeling, and H. Rosé. Evolutionary strategies of optimization. *Phys. Rev. E*, 56(1):1171–1180, 1997.

[10] K. B. Athreya and P. E. Ney. *Branching Processes*. Springer–Verlag, Berlin, 1972.

[11] H. M. Berman, J. Westbrook, Z. Feng, G. Gilliland, T. N. Bhat, H. Weissig, I. N. Shindyalov, and P. E. Bourne. The Protein Data Bank. *Nucleic Acids Res.*, 28(1):235–242, 2000.

[12] M. C. Biggs. Minimization algorithms making use of non-quadratic properties of the objective function. *J. Inst. Math. Appl.*, 8:315–327, 1971.

[13] A. Boneh and A. Golan. Constraints' redundancy and feasible region boundedness by random feasible point generator (RFPG)., 1979. Paper presented at EURO III, Amsterdam, The Netherlands.

[14] T. Brodmeier and E. Pretsch. Application of genetic algorithms in molecular modeling. *J. Comput. Chem.*, 15:588–595, 1994.

[15] B. R. Brooks, R. E. Bruccoleri, B. D. Olafson, D. J. States, S. Swaminathan, and M. Karplus. CHARMM: A program for macromolecular energy, minimization, and dynamics calculations. *J. Comput. Chem.*, 4(2):187–217, 1983.

[16] S. H. Brooks. A discussion of random methods for seeking maxima. *Operations Research*, 6:244–251, 1958.

[17] C. G. Broyden. The convergence of a class of double-rank minimization algorithms. *J. Inst. Math. Appl.*, 6:76–90, 1970.

[18] T. F. Coleman and Y. Li. On the convergence of reflective Newton methods for large-scale nonlinear minimization subject to bounds. *Math. Program.*, 67(2):189–224, 1994.

[19] T. F. Coleman and Y. Li. An interior, trust region approach for nonlinear minimization subject to bounds. *SIAM J. Optim.*, 6:418–445, 1996.

[20] T. F. Coleman and Z. Wu. Parallel continuation-based global optimization for molecular conformation and protein folding. Technical Report MCS-P443-0694, Argonne National Laboratory, 1994.

[21] W. D. Cornell, P. Cieplak, C. L. Bayly, I. R. Gould K. M. Merz, Jr., D. M. Ferguson, D. C. Spellmeyer, T. Fox, J. W. Caldwell, and P. A. Kollman. A second generation force field for the simulation of proteins, nuclei acids, and organic molecules. *J. Am. Chem. Soc.*, 117:5179–5197, 1995.

[22] T. E. Creighton. *Proteins : Structures and Molecular Properties*. W.H. Freeman and Company, New York, NY, second edition, 1993.

[23] S. Crivelli, E. Eskow, B. Bader, V. Lamberti, R. Byrd, R. Schnabel, and T. Head-Gordon. A physical approach to protein structure prediction. *Biophys. J.*, 82:36–49, 2002.

[24] B. Das, H. Meirovitch, and I. M. Navon. Performance of enriched methods for large unconstrained optimization as applied to models of proteins. *J. Comput. Chem.*, 24:1222–1231, 2003.

[25] P. J. Davis. *Circulant Matrices*. John Wiley & Sons, New York, NY, 1979.

[26] A. Dekkers and E. Aarts. Global optimization and simulated annealing. *Math. Program.*, 50:367–393, 1991.

[27] J. E. Dennis, Jr. and R. B. Schnabel. *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*. SIAM, Philadelphia, PA, 1996. Corrected reprint of the 1983 original.

[28] A. Dhooge, W. Govaerts, and Y. A. Kuznetsov. MATCONT: A Matlab package for numerical bifurcation analysis of ODEs. *ACM Trans. Math. Software*, 29(2):141–164, 2003.

[29] I. Diener. Trajectory methods in global optimization. In R. Horst and P. M. Pardalos, editors, *Handbook of Global Optimization*, pages 649–668. Kluwer Academic Publishers, The Netherlands, 1995.

[30] K. A. Dill, S. Bromberg, K. Yue, K. M. Fiebig, D. P. Yee, P. D. Thomas, and H. S. Chan. Principles of protein folding—a perspective from simple exact models. *Protein Sci.*, 4:561–602, 1995.

[31] L. C. W. Dixon and G. P. Szegö, editors. *Towards Global Optimisation, Volumes 1–2*. North–Holland, The Netherlands, 1975, 1978.

[32] E. J. Doedel. AUTO: A program for the automatic bifurcation analysis of autonomous systems. In *Proc. 10th Manitoba Conf. on Num. Math. and Comp., Univ. of Manitoba, Winnipeg, Canada*, pages 265–284, 1981.

[33] E. J. Doedel, A. R. Champneys, T. F. Fairgrieve, Y. A. Kuznetsov, B. Sandstede, and X.-J. Wang. AUTO97: Continuation and bifurcation software for ordinary differential equations. Technical report, Department of Computer Science, Concordia University, Montreal, Canada, 1997.

[34] E. J. Doedel, R. C. Paffenroth, A. R. Champneys, T. F. Fairgrieve, Y. A. Kuznetsov, B. Sandstede, and X. Wang. AUTO 2000: Continuation and bifurcation software for ordinary differential equations (with HomCont). Technical report, Caltech, 2001.

[35] Y. Duan and P. A. Kollman. Computational protein folding: From lattice to all-atom. *IBM Systems Journal*, 40(2):297–309, 2001.

[36] E. Fan and T. Terlaky. Application of direct search methods in molecular conformation problems. Technical Report AdvOl 2002/2, Advanced Optimization Laboratory, McMaster University, 2002.

[37] A. Fiser and A. Sali. Comparative protein structure modeling. In D. Chasman, editor, *Protein Structure: Determination, Analysis, and Applications for Drug Discovery*, pages 167–206. Marcel Dekker, New York, NY, 2003.

[38] R. Fletcher. A new approach to variable metric algorithms. *Computer J.*, 13:317–322, 1970.

[39] C. A. Floudas. *Deterministic Global Optimization: Theory, Algorithms and Applications*. Kluwer Academic Publishers, The Netherlands, 2000.

[40] C. A. Floudas, J. L. Klepeis, and P. M. Pardalos. Global optimization approaches in protein folding and peptide docking. In F. Roberts, editor, *DIMACS Series in Discrete Mathematics and Theoretical Computer Science, Vol. 47*, pages 141–171. American Mathematical Society, Providence, RI, 1999.

[41] C. A. Floudas and P. M. Pardalos, editors. *Recent Advances in Global Optimization*. Princeton University Press, Princeton, NJ, 1992.

[42] C. A. Floudas and P. M. Pardalos, editors. *Optimization in Computational Chemistry and Molecular Biology*. Kluwer Academic Publishers, The Netherlands, 2000.

[43] F. Freudenstein and B. Roth. Numerical solution of systems of nonlinear equations. *J. ACM*, 10(4):550–556, October 1963.

[44] P. E. Gill and W. Murray. Newton-type methods for unconstrained and linearly constrained optimization. *Math. Program.*, 28:311–350, 1974.

[45] D. Goldfarb. A family of variable metric updates derived by variational means. *Math. Comp.*, 24:23–26, 1970.

[46] G. Gray, T. Kolda, K. Sale, and M. Young. Optimizing an empirical scoring function for transmembrane protein structure determination. *INFORMS J. Comput.*, 16(4):406–418, 2004.

[47] A. Griewank, D. Juedes, and J. Utke. Algorithm 755: ADOL-C: a package for the automatic differentiation of algorithms written in C/C++. *ACM Trans. Math. Software*, 22(2):131–167, June 1996.

[48] A. Grossfield and J. W. Ponder. Global optimization via a modified potential smoothing kernel. Technical Report CCB 2002-01, Washington University School of Medicine, 2002.

[49] U. H. E. Hansmann and L. T. Wille. Global optimization by energy landscape paving. *Phys. Rev. Lett.*, 88(6):068105, 2002.

[50] T. Hansson, C. Oostenbrink, and W. F. van Gunsteren. Molecular dynamics simulations. *Curr. Opin. Struct. Biol.*, 12(2):190–196, 2002.

[51] T. E. Harris. *The Theory of Branching Processes*. Springer, Berlin, 1963.

[52] J.-B. Hiriart-Urruty. Conditions for global optimality. In R. Horst and P. M. Pardalos, editors, *Handbook of Global Optimization*, pages 1–26. Kluwer Academic Publishers, The Netherlands, 1995.

[53] J.-B. Hiriart-Urruty. Conditions for global optimality 2. *J. Global Optim.*, 13:349–367, 1998.

[54] L. Holm and C. Sander. Database algorithm for generating protein backbone and side-chain coordinates from a C^{α} trace: Application to model building and detection of coordinate errors. *J. Mol. Biol.*, 218:183–194, 1991.

[55] R. Horst and P. M. Pardalos, editors. *Handbook of Global Optimization*. Kluwer Academic Publishers, The Netherlands, 1995.

[56] R. Horst and H. Tuy. *Global Optimization – Deterministic Approaches*. Springer–Verlag, Berlin, third edition, 1996.

[57] M. D. Huang, F. Romeo, and A. Sangiovanni-Vincentelli. An efficient general cooling schedule for simulated annealing. In *Proceedings of the International Conference on Computer Aided Design*, page 381, 1986.

[58] J. S. Hunjan, S. Sarkar, and R. Ramaswamy. Global optimization on an evolving energy landscape. *Phys. Rev. E*, 66(4):046704, 2002.

[59] L. Ingber. Simulated annealing: Practice versus theory. *Math. Comput. Model.*, 18(11):29–57, 1993.

[60] IUPAC-IUB, Commission on Biochemical Nomenclature (CBN). Abbreviations and symbols for the description of the conformation of polypeptide chains. *Biochemistry*, 9:3471–3479, 1970.

[61] R. I. Jennrich and P. F. Sampson. Application of stepwise regression to nonlinear estimation. *Technometrics*, 10:63–72, 1968.

[62] W. L. Jorgenson and J. Tirado-Rives. The OPLS potential functions for proteins. Energy minimizations for crystals of cyclic peptides and crambin. *J. Am. Chem. Soc.*, 110(6):1657–1666, 1988.

[63] H. Kawai, T. Kikuchi, and Y. Okamoto. A prediction of tertiary structures of peptides by the Monte Carlo simulated annealing method. *Protein Eng.*, 3:85–94, 1989.

[64] C. T. Kelley. *Iterative Methods for Optimization*. SIAM, Philadelphia, PA, 1999.

[65] J. C. Kendrew, G. Bodo, H. M. Dintzis, R. G. Parrish, H. Wyckoff, and D. C. Phillips. A three-dimensional model of the myoglobin molecule obtained by X-ray analysis. *Nature*, 181:662–666, 1958.

[66] S. Kirkpatrick, C. D. Gelatt Jr., and M. P. Vecchi. Optimization by simulated annealing. *Science*, 20:671–680, 1983.

[67] A. Kuntsevich and F. Kappel. Moré set of test functions. `http://www.uni-graz.at/imawww/kuntsevich/solvopt/results/moreset.html`.

[68] Y. A. Kuznetsov and V. V. Levitin. CONTENT: Integrated environment for analysis of dynamical systems. Technical report, CWI, Amsterdam, 1997.

[69] S. M. Le Grand and K. M. Merz, Jr. The application of the genetic algorithm to the minimization of potential energy functions. *J. Global Optim.*, 3:49–66, 1993.

[70] J. Lee, H. A. Scheraga, and S. Rackovsky. New optimization method for conformational energy calculations on polypeptides: Conformational space annealing. *J. Comput. Chem.*, 18:1222–1232, 1997.

[71] Z. Li and H. A. Scheraga. Monte Carlo-minimization approach to the multiple-minima problem in protein folding. *P. Natl. Acad. Sci. U.S.A.*, 84:6611–6615, 1987.

[72] A. Liwo, J. Pillardy, C. Czaplewski, J. Lee, D. R. Ripoll, M. Groth, S. Rodziewicz-Motowidlo, R. Kamierkiewicz, R. J. Wawak, S. Oldziej, and H. A. Scheraga. UNRES: A united-residue force field for energy-based prediction of protein structure—origin and significance of multibody terms. In *RECOMB '00: Proceedings of the Fourth Annual International Conference on Computational Molecular Biology*, pages 193–200, New York, NY, 2000. ACM Press.

[73] A. Liwo, M. R. Pincus, R. J. Wawak, S. Rackovsky, and H. A. Scheraga. Calculation of protein backbone geometry from α-carbon coordinates based on peptide-group dipole alignment. *Protein Sci.*, 2:1697–1714, 1993.

[74] M. Locatelli. Simulated annealing algorithms for continuous global optimization. In P. M. Pardalos and H. E. Romeijn, editors, *Handbook of Global Optimization, Volume 2*, pages 179–229. Kluwer Academic Publishers, The Netherlands, 2002.

[75] N. Metropolis, A. Rosenbluth, M. Rosenbluth, A. Teller, and E. Teller. Equations of state calculations by fast computing machines. *J. Chem. Phys.*, 21:1087–1090, 1953.

[76] R. R. Meyer. Theoretical and computational aspects of nonlinear regression. In J. B. Rosen, O. L. Mangasarian, and K. Ritter, editors, *Nonlinear Programming*, pages 465–496, New York, NY, 1970. Academic Press.

[77] A. Migdalas, P. M. Pardalos, and P. Värbrand, editors. *From Local to Global Optimization*. Kluwer Academic Publishers, The Netherlands, 2001.

[78] J. J. Moré, B. S. Garbow, and K. E. Hillstrom. Testing unconstrained optimization software. *ACM Trans. Math. Software*, 7(1):17–41, March 1981.

[79] J. Moult. Comparison of database potentials and molecular mechanics force fields. *Curr. Opin. Struct. Biol.*, 7(2):194–199, 1997.

[80] S. G. Nash and A. Sofer. *Linear and Nonlinear Programming*. McGraw–Hill, New York, NY, 1996.

[81] G. Nemethy, K. D. Gibson, K. A. Palmer, C. N. Yoon, G. Paterlini, A. Zagari, S. Rumsey, and H. A. Scheraga. Energy parameters in polypeptides. 10. Improved geometrical parameters and nonbonded interactions for use in the ECEPP/3 algorithm with application to proline-containing peptides. *J. Phys. Chem.*, 96:6472–6484, 1992.

[82] A. Neumaier. Global optimization. `http://www.mat.univie.ac.at/~neum/glopt.html`.

[83] A. Neumaier. Molecular modeling and mathematical prediction of protein structure. *SIAM Rev.*, 39:407–460, 1997.

[84] A. Neumaier. Complete search in continuous global optimization and constraint satisfaction. *Acta Numerica*, pages 271–369, 2004.

[85] J. T. Ngo and J. Marks. Computational complexity of a problem in molecular structure prediction. *Protein Eng.*, 5:313–321, 1992.

[86] P. M. Pardalos and H. E. Romeijn, editors. *Handbook of Global Optimization, Volume 2*. Kluwer Academic Publishers, The Netherlands, 2002.

[87] P. M. Pardalos, D. Shalloway, and G. Xue. Optimization methods for computing global minima of nonconvex potential energy functions. *J. Global Optim.*, 4:117–133, 1994.

[88] N. R. Patel, R. L. Smith, and Z. B. Zabinsky. Pure adaptive search in Monte Carlo optimization. *Math. Program.*, 43:317–328, 1988.

[89] A. T. Phillips, J. B. Rosen, and K. A. Dill. Convex global underestimation for molecular structure prediction. In A. Migdalas, P. M. Pardalos, and P. Värbrand, editors, *From Local to Global Optimization*. Kluwer Academic Publishers, The Netherlands, 2001.

[90] L. Piela, J. Kostrowicki, and H. A. Scheraga. The multiple minima problem in the conformational analysis of molecules. Deformation of the protein energy hypersurface by the diffusion equation method. *J. Phys. Chem.*, 93:3339–3346, 1989.

[91] J. D. Pintér. Global optimization: Software, test problems, and applications. In P. M. Pardalos and H. E. Romeijn, editors, *Handbook of Global Optimization, Volume 2*, pages 515–569. Kluwer Academic Publishers, The Netherlands, 2002.

[92] J. W. Ponder and D. A. Case. Force fields for protein simulations. *Adv. Protein Chem.*, 66:27–85, 2003.

[93] P. Révész. *Random Walks of Infinitely Many Particles*. World Scientific, Singapore, 1994.

[94] P. Salamon, P. Sibani, and R. Frost. *Facts, Conjectures, and Improvements for Simulated Annealing*. SIAM, Philadelphia, PA, 2002.

[95] A. G. Salinger, N. M. Bou-Rabee, R. P. Pawlowski, E. D. Wilkes, E. A. Burroughs, R. B. Lehoucq, and L. A. Romero. LOCA 1.1, a library of continuation algorithms: Theory and implementation manual. Technical report, Sandia National Laboratories, Albuquerque, NM, 2002.

[96] F. Sanger and E. O. Thompson. The amino-acid sequence in the glycyl chain of insulin. I. The identification of lower peptides from partial hydrolysates. *Biochemistry*, 53:353–366, 1953.

[97] F. Sanger and E. O. Thompson. The amino-acid sequence in the glycyl chain of insulin. II. The investigation of peptides from enzymic hydrolysates. *Biochemistry*, 53:366–374, 1953.

[98] S. Schelstraete, W. Schepens, and H. Verschelde. Energy minimization by smoothing techniques: A survey. In P. Balbuena and J. Seminario, editors, *Molecular Dynamics. From Classical to Quantum Methods*, pages 129–185. Elsevier, 1998.

[99] F. Schoen. Two-phase methods for global optimization. In P. M. Pardalos and H. E. Romeijn, editors, *Handbook of Global Optimization, Volume 2*, pages 151–177. Kluwer Academic Publishers, The Netherlands, 2002.

[100] A. Schug, T. Herges, and W. Wenzel. Reproducible protein folding with the stochastic tunneling method. *Phys. Rev. Lett.*, 91(15):158102, 2003.

[101] G. E. Schulz and R. H. Schirmer. *Principles of Protein Structure*. Springer–Verlag, New York, NY, 1979.

[102] D. Shalloway. Packet annealing: A deterministic method for global minimization. application to molecular conformation. In C. A. Floudas and P. M. Pardalos, editors, *Recent Advances in Global Optimization*, pages 561–592. Princeton University Press, Princeton, NJ, 1992.

[103] D. F. Shanno. Conditioning of quasi-Newton methods for function minimization. *Math. Comp.*, 24:647–656, 1970.

[104] J. K. Shin and M. S. Jhon. High directional Monte Carlo procedure coupled with the temperature heating and annealing as a method to obtain the global energy minimum structure of polypeptides and proteins. *Biopolymers*, 31:177–185, 1991.

[105] M. J. Sippl. Knowledge-based potentials for proteins. *Curr. Opin. Struct. Biol.*, 5(2):229–235, 1995.

[106] R. L. Smith. Monte Carlo procedures for generating random feasible solutions to mathematical programs, 1980. Paper presented at ORSA/TIMS Conference, Washington, D.C.

[107] D. C. Sorensen. Newton's method with a model trust region modification. *SIAM J. Numer. Anal.*, 19:409–426, 1982.

[108] E. Spedicato. Computational experience with quasi-Newton algorithms for minimization problems. Technical Report CISE-N-175, Segrate, Milano, Italy, 1975.

[109] F. H. Stillinger, T. Head-Gordon, and C. L. Hirshfield. Toy model for protein folding. *Phys. Rev. E*, 48(2):1469–1477, 1993.

[110] J. E. Straub. Optimization techniques with applications to proteins. In R. Elber, editor, *Recent Developments in Theoretical Studies of Proteins*, pages 137–196. World Scientific, Singapore, 1996.

[111] A. C. Sun and W. D. Seider. Homotopy-continuation algorithm for global optimization. In C. A. Floudas and P. M. Pardalos, editors, *Recent Advances in Global Optimization*, pages 561–592. Princeton University Press, Princeton, NJ, 1992.

[112] A. Törn and A. Žilinskas. *Global Optimization*. Springer–Verlag, Berlin, 1989.

[113] W. F. van Gunsteren and H. J. C. Berendsen. *GROMOS-87 Manual*. Laboratory of Physical Chemistry, University of Gröningen, The Netherlands, 1987.

[114] P. J. M. van Laarhoven and E. H. L. Aarts. *Simulated Annealing: Theory and Applications*. D. Reidel Publishing Company, Dordrecht, The Netherlands, 1987.

[115] T. Veitshans, D. Klimov, and D. Thirumalai. Protein folding kinetics: Timescales, pathways, and energy landscapes in terms of sequence-dependent properties. *Fold. Des.*, 2:1–22, January 1997.

[116] D. J. Wales and H. A. Scheraga. Global optimization of clusters, crystals, and biomolecules. *Science*, 284:1368–1372, August 1999.

[117] L. T. Watson. Theory of globally convergent probability-one homotopies for nonlinear programming. *SIAM J. Optim.*, 11(3):761–780, 2000.

[118] L. T. Watson, S. C. Billups, and A. P. Morgan. Algorithm 653: HOMPACK: A suite of codes for globally convergent homotopy algorithms. *ACM Trans. Math. Software*, 13:281–310, September 1987.

[119] L. T. Watson and R. T. Haftka. Modern homotopy methods in optimization. *Comput. Methods Appl. Mech. Engrg.*, 74:289–304, 1989.

[120] L. T. Watson, M. Sosonkina, R. C. Melville, A. P. Morgan, and H. F. Walker. Algorithm 777: HOMPACK90: A suite of Fortran 90 codes for globally convergent homotopy algorithms. *ACM Trans. Math. Software*, 23:514–549, December 1997.

[121] W. Wenzel and K. Hamacher. Stochastic tunneling approach for global minimization of complex potential energy landscapes. *Phys. Rev. Lett.*, 82(15):3003–3007, 1999.

[122] S. R. Wilson and W. Cui. Applications of simulated annealing to peptides. *Biopolymers*, 29:225–235, 1990.

[123] S. R. Wilson, W. Cui, J. W. Moskowitz, and K. E. Schmidt. Conformational analysis of flexible molecules: Location of the global minimum energy conformation by the simulated annealing method. *Tetrahedron Lett.*, 29(35):4373–4376, 1988.

[124] G. R. Wood and Z. B. Zabinsky. Stochastic adaptive search. In P. M. Pardalos and H. E. Romeijn, editors, *Handbook of Global Optimization, Volume 2*, pages 231–249. Kluwer Academic Publishers, The Netherlands, 2002.

[125] Z. Wu. The effective energy transformation scheme as a special continuation approach to global optimization with application to molecular conformation. *SIAM J. Optim.*, 6(3):748–768, 1996.

[126] D. Xie and T. Schlick. Efficient implementation of the truncated-Newton algorithm for large-scale chemistry applications. *SIAM J. Optim.*, 10(1):132–154, 1999.

[127] Z. B. Zabinsky and R. L. Smith. Pure adaptive search in global optimization. *Math. Program.*, 53:323–338, 1992.

[128] Z. B. Zabinsky, R. L. Smith, J. F. McDonald, H. E. Romeijn, and D. E. Kaufman. Improving hit-and-run for global optimization. *J. Global Optim.*, 3:171–192, 1993.

[129] W. I. Zangwill and C. B. Garcia. *Pathways to Solutions, Fixed Points, and Equilibria*. Prentice–Hall, Englewood Cliffs, NJ, 1981.