SANDIA REPORT

SAND2003-8516
Unlimited Release
Printed October 2003

Optimizing an Emperical Scoring Function for Transmembrane Protein Structure Determination

T. G. Kolda, G. A. Gray, K. L. Sale, M. M. Young

Prepared by
Sandia National Laboratories
Albuquerque, New Mexico 87185 and Livermore, California 94550

Sandia is a multiprogram laboratory operated by Sandia Corporation,
a Lockheed Martin Company, for the United States Department of Energy's
National Nuclear Security Administration under Contract DE-AC04-94AL85000.

Approved for public release; further dissemination unlimited.

Sandia National Laboratories

Issued by Sandia National Laboratories, operated for the United States Department of Energy by Sandia Corporation.

NOTICE: This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government, nor any agency thereof, nor any of their employees, nor any of their contractors, subcontractors, or their employees, make any warranty, express or implied, or assume any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represent that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government, any agency thereof, or any of their contractors or subcontractors. The views and opinions expressed herein do not necessarily state or reflect those of the United States Government, any agency thereof, or any of their contractors.

Printed in the United States of America. This report has been reproduced directly from the best available copy.

Available to DOE and DOE contractors from
U.S. Department of Energy
Office of Scientific and Technical Information
P.O. Box 62
Oak Ridge, TN 37831

Telephone: (865)576-8401
Facsimile: (865)576-5728
E-Mail: reports@adonis.osti.gov
Online ordering: http://www.osti.gov/bridge/

Available to the public from
U.S. Department of Commerce
National Technical Information Service
5285 Port Royal Rd
Springfield, VA 22161

Telephone: (800)553-6847
Facsimile: (703)605-6900
E-Mail: orders@ntis.fedworld.gov
Online order: http://www.ntis.gov/help/ordermethods.asp?loc=7-4-0#online

SAND2003-8516
Unlimited Release
Printed October 2003

Optimizing an Empirical Scoring Function for Transmembrane Protein Structure Determination

Genetha Anne Gray*
Tamara G. Kolda†
Computational Sciences and Mathematics Research Department
Sandia National Laboratories
Livermore, CA 94551-9217

Kenneth L. Sale‡
Malin M. Young§
Biosystems Research Department
Sandia National Laboratories
Livermore, CA 94551-9217

ABSTRACT

We examine the problem of transmembrane protein structure determination. Like many other questions that arise in biological research, this problem cannot be addressed by traditional laboratory experimentation alone. An approach that integrates experiment and computation is required. We investigate a procedure which states the transmembrane protein structure determination problem as a bound constrained optimization problem using a special empirical scoring function, called Bundler, as the objective function. In this paper, we describe the optimization problem and some of its mathematical properties. We compare and contrast results obtained using two different derivative free optimization algorithms.

Keywords: transmembrane protein, bound constrained optimization

*Corresponding author. Email: gagray@sandia.gov.
†Email: tgkolda@sandia.gov.
‡Email: klsale@sandia.gov.
§Email: mmyoung@sandia.gov.

This page intentionally left blank.

1 Introduction

In this study, we consider solving the bound constrained nonlinear optimization problem

$$\begin{aligned} &\min f(\mathbf{x}) \\ &\text{s.t. } \mathcal{L} \leq \mathbf{x} \leq \mathcal{U}, \end{aligned} \tag{1.1}$$

where $f : \mathbb{R}^n \to \mathbb{R}$ is a nonlinear function; $\mathbf{x}, \mathcal{L}, \mathcal{U} \in \mathbb{R}^n$; and $\mathcal{L}$ and $\mathcal{U}$ are given lower and upper bounds on $\mathbf{x}$ respectively. In particular, we are interested in an important computational biology problem, transmembrane protein structure determination, which can be formulated as (1.1). In this application, the objective function f is an empirical scoring function designed to rate the validity of proposed transmembrane protein structures. The variable $\mathbf{x} \in \mathbb{R}^n$ represents the spatial positions of certain components of the transmembrane protein, and the bounds $\mathcal{L}$ and $\mathcal{U}$ are derived using some observed properties of these components.

There is a wide variety of optimization methods available for finding a solution to (1.1). However, the effectiveness and efficiency of these algorithms can be application specific. Hence, answering the question of which to use is not easy. In this paper, we examine the transmembrane protein structure identification problem and its model formulation. We choose two different optimization algorithms are that seem to suit this application. We compare and contrast numerical results we obtained using real data for a transmembrane protein of known structure.

This paper is organized as follows: In section 2 we discuss the biological significance of transmembrane proteins and the importance of determining their structures. Then, in section 3, we describe the mathematical formulation of the transmembrane protein structure determination problem and give some details of the scoring function. We review some of the basic characteristics of the optimization methods that we applied to the problem and give the details of our implementation of these algorithms in section 4. The results of our numerical study are presented in section 5. Finally, in section 6, we summarize our work and draw some conclusions.

2 Biological Background

Approximately one-third of the proteins encoded for by a typical genome are transmembrane proteins, and they participate in many important cell processes. Some transmembrane proteins form a channel through which certain ions and molecules can enter or leave the cell. Others act as signal transduction receptors or play roles in cell recognition, senses mediation, or cell to cell communication. Many diseases are the result of transmembrane protein malfunction, absence, or mutation. Hence, these proteins are an important target of drug design. In fact, a large percentage of the current pharmaceuticals act on transmembrane proteins [55].

Like all proteins, a transmembrane protein is a macromolecule consisting of a chain of amino acids. The defining characteristic of a transmembrane protein is that this chain traverses the cell membrane one or more times. For example, a G-protein-coupled receptor, one type of transmembrane protein involved in signal transduction, spans the cell membrane 7 times. The portion of the transmembrane protein within the cell membrane consists primarily of hydrophobic amino acids, while the portion outside the cell membrane consists mainly of hydrophilic amino acids. These characteristics, in conjunction with the makeup of the cell membrane, dictate the overall structure of transmembrane proteins. In particular, due to the chemical environment of the membrane interior, the amino acids that are inside the cell membrane form

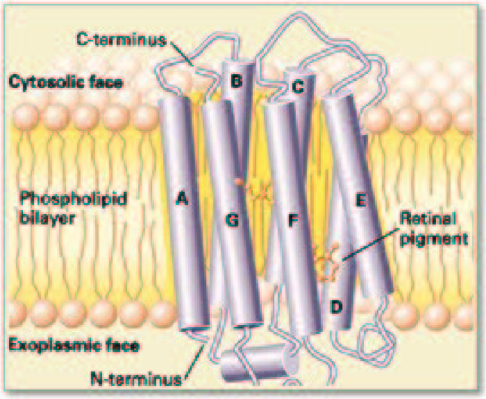


Figure 1: This is an illustration of the transmembrane protein rhodopsin in a retina cell membrane. The seven linked cylinders, labeled A through G, represent the seven α-helices that traverse the cell membrane. (This cartoon was obtained from the G-protein-coupled receptor data base [51].)

stable secondary structures including α-helices and β-sheets. To date, two major structural classes of transmambrane domains have been observed: all α-helical and all β-stranded. We will limit the subsequent discussion to the all α-helical case. Hence, for the purposes of our study, a transmembrane protein consists of a bundle of connected α-helices. Figure 1 contains an illustration of a transmembrane protein in which the α-helices are represented as cylinders.

Currently, the protein data bank (PDB) contains over 21,000 structures, and its size is increasing exponentially [5]. However, the majority of the proteins found in the PDB are soluble proteins. To date, the structures of only about 30 transmembrane proteins have been determined (see [46] and references therein). This is due to the fact that experimental structure determination methods such as X-ray crystallography and nuclear magnetic resonance (NMR) have been difficult to apply to transmembrane proteins. Furthermore, since so few transmembrane protein structures have been determined, very few suitable templates exist for homology modeling [21]. Therefore, the development of an integrated computational/experimental model to address transmembrane protein structure and function questions is an important challenge in the field of structural biology.

The modeling of transmembrane proteins can be broken up into separate tasks of defining the transmembrane helices and determining the relative orientation of these helices. A process known as sliding-window hydrophobicity is an accurate and well established method of predicting transmembrane helices given their amino acid sequences [24, 25, 43]. As of yet, no widely accepted method has emerged to subsequently ascertain the spatial locations of these helices. Because the cell membrane does impose certain structural constraints on the positions of the helices and thus limits the number of possible structures, several ab-initio computational approaches have been proposed [7, 35, 52]. One such procedure is based on the fact that the conformational space of membrane proteins can be effectively sampled and gives a technique for enumerating all the possible helical bundles [7]. However, this method neglects the orientations of the individual helices around their respective axes. Several other methods seem promising

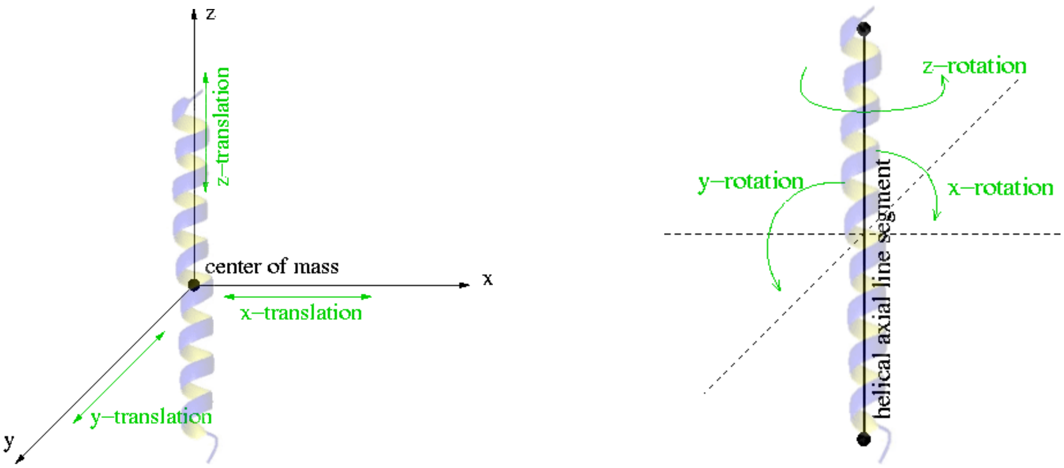


Figure 2: These pictures depict the six positional variables associated with each helix. The image on the left shows the x, y and z translations of the helix which are defined in terms of the center of mass of the helix in its initial placement. On the right, the x, y and z rotations are illustrated. We use the initial placement of the helix to define an axial line segment from two points centered in the terminal turn at each of the helix.

but are biased toward the structures of specific transmembrane proteins and have yet to be validated for other transmembrane proteins [13, 35, 52].

3 Transmembrane Protein Structure Determination

In [17, 45], a new method is proposed for determining the spatial location of the transmembrane protein helices. This method focuses on finding a solution to the optimization problem (1.1) where the objective function f assigns a score to each helical arrangement which is a measure of how similar it is to the actual structure.

3.1 Mathematical Description of the Problem

In this study, determining the structure of the transmembrane protein is reduced to describing the relative orientation of the helices, or how they "bundle." Each helix is assumed to be a rigid body, and thus we describe its position in space using its center of mass and a line segment defined by the two points centered in the terminal turns of the helical ends. We define a three-dimensional reference space for each helix using its initial center of mass and initial helix axial line segment. In other words, the position of each helix is defined in terms of its original location. Then, the variables in (1.1) are merely the x, y, and z translations from the original centers of mass of each helix and the x, y, and z rotations about the initial helix axial line segment for each helix as illustrated in figure 2. Hence, a transmembrane protein with m helices has $6m$ variables. At this time, we do not consider the loops that connect the helices as part of the structure determination but note that they can be added using existing techniques after the helical positions have been established [54, 56].

All $6m$ variables have simple bounds, most of which derive from the fact that transmembrane proteins reside in the cell membrane. The restrictions on the x and y rotations of each helix are the result of a survey of helix tilt angles described in [6]. The z rotational variables have no such limitations and are allowed to vary in the entire z-rotational space. Both the x and the y translations are confined to a space that is approximately one-third of the total radius of the membrane protein as suggested by a study of helix packing behavior found in [6]. The z translation variables have the tightest bounds. Their movement is limited by the membrane itself since the helical portions of the transmembrane protein must remain inside the cell membrane.

We now need a way to compare possible structures and decide which one best approximates the transmembrane protein in question. If the structure were known, such comparisons could be made simply using root mean square deviation (RMSD)[1]. However, the overall goal of this work is to identify unknown transmembrane protein structures, so we must develop another technique. We use a penalty scoring function, known as Bundler, to rate each structure [45]. Bundler measures how well a structure conforms to specific criteria based on experimental data and helix bundling features described in the literature, and it does not require any *a priori* knowledge of the location of the helices. The Bundler score is smallest for those structures that most closely meet the specified criteria. Thus, we define an objective function f for problem (1.1) using Bundler to give this structure a score. Therefore, minimizing f is the computational tool for determining the structure of a transmembrane protein.

3.2 The Scoring Function: Bundler

As previously stated, the Bundler scoring function combines experimental data and topological models created from a survey of known transmembrane helix packing interactions. For each structure, the score is calculated as the sum

$$P = P_E + P_I, \tag{3.2}$$

where P_E quantifies the structure's violation of a set of experimental distance constraints and P_I quantifies how well the structure satisfies some helix packing parameters determined by analyzing a set of 16 nonredundant membrane proteins. In this paper, we are interested in the details of optimizing such a function. Hence, we give only a basic description of the Bundler scoring function. We direct the reader to [45] for further details and more specific explanations of the function's development, including the results of a study that show correlation between scores and RMSDs.

It has been shown that distance constraints are an important aspect in determining transmembrane protein structure. In fact, the number of possible structures decreases exponentially with the number of distance constraints and increases exponentially with the error on the distance measures [17]. Hence, Bundler incorporates experimental distance constraints in the term

$$P_E = \sum_{(i,j)\in\Omega} K_E * \begin{cases} (d_{ij} - \ell_{ij})^2, & r_{ij} < \ell_{ij}, \\ 0, & \ell_{ij} \le d_{ij} \le u_{ij}, \\ (u_{ij} - d_{ij})^2, & d_{ij} > u_{ij}, \end{cases} \tag{3.3}$$

[1] RMSD is a way of comparing two protein structures by calculating the sum of the distances of comparable atoms. See, for example, [30] for more details.

where ℓ_{ij} and u_{ij} are predetermined upper and lower bounds on the distance between atoms i and j, respectively; d_{ij} is the distance between atoms i and j in the current structure; Ω is a subset of atom pairs; and K_E is a force constant. The distance constraints ℓ_{ij} and u_{ij} are obtained from experimental methods such as chemical crosslinking, dipolar electron paramagnetic resonance (dipolar EPR), fluorescence resonance energy transfer (FRET), or NMR for assembling transmembrane helical proteins. Note that these constraints are not procurable for every pair of atoms in the structure. Instead, experimental distance constraints are only available for a small subset, Ω, of all atom pairs.

Obtaining enough distance constraints to uniquely determine a structure is difficult, particularly for transmembrane proteins [16, 17]. Furthermore, these distances are never error free. Hence, Bundler also includes a term to distinguish between structures that meet observed helix packing properties (determined from an analysis of known structures) and those that do not. This term, P_I, is actually a sum of 6 different terms, i.e.,

$$P_I = P_\delta + P_\theta + P_\phi + P_{sc} + P_{vdw} + P_c. \tag{3.4}$$

Each term checks a different helical bundling property.

The packing distance score, P_δ, and packing angle score, P_ϕ, consider all the helical pairs in the bundle and penalize them if they are too far apart or too close together. Let Γ denote the set of $m(m-1)/2$ distinct helical pairs (i, j). Then the packing distance score is defined as

$$P_\delta = \sum_{(i,j)\in\Gamma} K_\delta * \begin{cases} (\delta_{ij} - \delta_l)^2, & \delta_{ij} < \delta_l, \\ 0, & \delta_l \le \delta_{ij} \le \delta_u, \\ (\delta_u - \delta_{ij})^2, & \delta_{ij} > \delta_u. \end{cases} \tag{3.5}$$

Here, $\delta_l = \overline{\delta} - 1.5s_\delta$ and $\delta_u = \overline{\delta} + 1.5s_\delta$, where $\overline{\delta}$ and s_δ are the mean and standard deviation of the interhelical distances, respectively, which are calculated using a set of 16 known structures; δ_{ij} is the distance between the centers of mass of helices i and j in the current structure; and K_δ is a given force constant. Similarly, the packing angle score is defined as

$$P_\theta = \sum_{(i,j)\in\Gamma} K_\theta * \begin{cases} (\theta_{ij} - \theta_l)^2, & \theta_{ij} < \theta_l, \\ 0, & \theta_l \le \theta_{ij} \le \theta_u, \\ (\theta_u - \theta_{ij})^2, & \theta_{ij} > \theta_u, \end{cases} \tag{3.6}$$

where $\theta_l = \overline{\theta} - 1.5s_\theta$ and $\theta_u = \overline{\theta} + 1.5s_\theta$, and $\overline{\theta}$ and s_θ are the mean and standard deviation of the interhelical packing angles; δ_{ij} is the interhelical packing angle between helices i and j in the current structure; and K_θ is a given force constant.

The packing density is defined as the ratio of atomic volume to solvent accessible volume [42]. It gages how efficiently a protein folds together or equivalently how much interior space is left unused. The packing density score is defined as

$$P_\phi = K_\phi * \begin{cases} (\phi - \phi_l)^2, & \phi < \phi_l, \\ 0, & \phi_l \le \phi \le \phi_u, \\ (\phi_u - \phi)^2, & \phi > \phi_u, \end{cases} \tag{3.7}$$

where $\phi_l = \overline{\phi} - 1.5s_\phi$ and $\phi_u = \overline{\phi} + 1.5s_\phi$, and $\overline{\delta}$ and s_δ are the mean and standard deviation of the observed packing density; ϕ is the packing density of the current structure; and K_ϕ is a given force constant. It penalizes those structures which are packed too tightly or too loosely.

In transmembrane proteins, it has been observed that amino acids have a preference for which amino acids they interact with on neighboring helices [2, 3, 35]. The side-chain interaction propensity score, P_{sc}, incorporates this into Bundler. It is based on the membrane helical interfacial pairwise (MHIP) amino acid interaction propensity table proposed in [3], and it penalizes structures containing amino acid pairs that are in contact contrary to their normal observed behavior. Let Λ_i be the set of Cβ atoms in helix i and Υ be the set of m consecutive helical pairs. Then, the side-chain propensity score is defined as

$$P_{sc} = \sum_{(i,j)\in\Upsilon} \left[\sum_{a\in\Lambda_i, b\in\Lambda_j} K_{sc} * (p - p_{ab}) \right], \tag{3.8}$$

where p is the maximum propensity score in the MHIP table; p_{ab} is the MHIP propensity value of atoms a and b; and K_{sc} is a constant. Note because we are using the MHIP table, the side-chain propensity score introduces discontinuities in Bundler.

To prevent interhelical clashes, Bundler includes the van der Waals repulsive function [8]

$$P_{vdw} = \sum_{(i,j)\in\Lambda} K_{vdw} * \begin{cases} 0, & r_{ij} \geq sR_{ij}, \\ (s^2R_{ij}^2 - r_{ij}^2)^2, & r_{ij} < sR_{ij}. \end{cases} \tag{3.9}$$

Here, Λ is the set of all pairs of Cβ atoms; r_{ij} is the distance between Cβ atoms i and j in the current structure; R_{ij} is the observed distance at which atoms i and j interact or repulse; s is a predetermined van der Waals scaling factor; and K_{vdw} is a given constant.

Finally, to ensure that each helix has at least two neighboring helices, Bundler includes a contact score. This piece of the scoring function guarantees that the helices are packed tightly and prevents any one helix from being excluded from the bundle. It is defined as

$$P_c = \sum_{i\in\Delta} K_c * \begin{cases} 0, & c_i \geq 2 \\ (2 - c_i), & c_i < 2, \end{cases} \tag{3.10}$$

where Δ is the set of helices; c_i is the number of helices that helix i is in contact with; and K_c is a given constant. Two helices are defined to be in contact if their centers of mass are within a given distance of one another. This distance bound is calculated using the analysis of the 16 known structures.

Note that all the pieces of the Bundler scoring function contain at least one constant as well as some predetermined bounds. Setting these parameters is an important component of the transmembrane protein structure determination problem. We do not explicitly give their values in this paper, but instead direct the reader to [45] for more specific details.

4 Optimization Methods

Because the Bundler scoring function does contain discontinuities, we have chosen two derivative free methods to obtain a solution to (1.1). Although we focus on two particular methods here, there are many other derivative free methods (see for example [28, 40] and references therein). Moreover, finite differencing could be used to approximate the gradient so that we could utilize any number of derivative based methods. However, because Bundler incorporates noisy experimental data, such approximations may contain too much error to be useful. We are actively pursuing this research direction and hope to communicate our findings in a future publication. In this paper, we present results using simulated annealing and parallel pattern search, described below.

4.1 Simulated Annealing

Simulated annealing (SA) is an optimization method that is often applied to molecular conformation problems. For a few of the many examples of its use in computational biology, see [9, 10, 19, 20, 37] and references therein. The SA algorithm is a computational analogue to the industrial annealing process in which metal alloys are slowly cooled to obtain an optimal molecular configuration. This controlled cooling process is very important since a less stable configuration is obtained when the alloy is cooled too quickly. Computationally, annealing is implemented by allowing optimization steps that do not necessarily reduce the objective function. The idea is that a few bad steps can be accepted in order to get on the best path to the solution.

The SA algorithm is based on the Metropolis method [33] of obtaining the equilibrium configuration of a group of atoms at a given temperature. A connection between the Metropolis method and Monte Carlo simulation was first described in [39]. Then in [26], Kirkpatrick and his colleagues propose the simulated annealing optimization technique that is used today. It begins with a Metropolis Monte Carlo simulation at a high temperature. After a sufficient number of Monte Carlo steps have been taken, the temperature is reduced and the Metropolis Monte Carlo is continued. This process is repeated until a specified final temperature is reached. At high temperatures, a relatively large number of the random steps will be accepted, and as the temperature decreases, fewer steps are accepted.

The main advantage of SA over other optimization methods is that it is a global method. It can avoid becoming trapped in bad local minima regardless of the starting point. Furthermore, SA is easy to implement. Unfortunately, SA also has many well-documented disadvantages. It requires extensive computational work [1, 15, 34, 53]. Furthermore, SA is sensitive to the choice of its many parameters which can be difficult to fine tune [1, 15, 38, 41, 47, 53]. For example, there are at least a dozen different temperature cooling schedule from which to choose [18, 26, 44, 53]. Finally, because the steps in SA are taken randomly, the algorithm does not employ any knowledge gained in previous iterations [4].

In our implementation of simulated annealing, we use the geometric annealing schedule,

$$T_{new} = \alpha * T_{old}, \tag{4.11}$$

where $\alpha = 0.95$. This parameter was determined using numerical experiments. Each temperature cycle is terminated after either 1000 structures are generated or after 100 structrues are accepted. The initial temperature and the number of temperature cycles are determined independently for each numerical test. The SA algorithm is implemented in C and uses the PDB Record I/O Libraries to read and write Brookhaven PDB formatted files [11]. Our implementation of SA is a serial version. Although some parallelized versions do exist [27, 31, 48], none are compatible with MPI libraries such as MPICH-1.2.4.

4.2 Asynchronous Parallel Pattern Search

Pattern search methods are practical for solving problems such as (1.1) when the derivative of the objective function is unavailable and approximations are unreliable. They utilize a predetermined pattern of points to sample the given function space. When certain requirements on the form of the points in the pattern are followed, it can be shown that under other mild conditions, global convergence to a stationary point is guaranteed [14, 32, 50]. We also note that pattern search methods are most effective for optimization problems with less than 100 variables. Most transmembrane proteins have less than 13 helices, and we are interested in proteins that have 12 or less. Hence, the transmembrane protein structure determination problem that we consider contains at most 72 variables, and pattern search is a reasonable choice.

The majority of the computational cost of pattern search methods is the function evaluations. Hence, parallel pattern search (PPS) techniques have been derived to take advantage of parallel platforms in order to reduce the overall computational time. In particular, PPS exploits the fact that once the points in the search pattern have been defined, the function values at these points can be computed independently. PPS algorithms calculate these function values simultaneously [12, 49].

The particular implementation of PPS that we use is asynchrounous. Asynchronous parallel pattern search (APPS) retains the positive features of PPS, but it does not assume that the amount of time required for an objective function evaluation is constant or that the processors are homogeneous. It does not have any required synchronizations and thus requires less total time to return results that are comparable to those acheived by PPS [23]. Furthermore, it has been shown that APPS is globally convergent under the standard assumptions for PPS [29]. Finally, there is an existing open source version of APPS, called APPSPACK, which is easy to install and use [22].

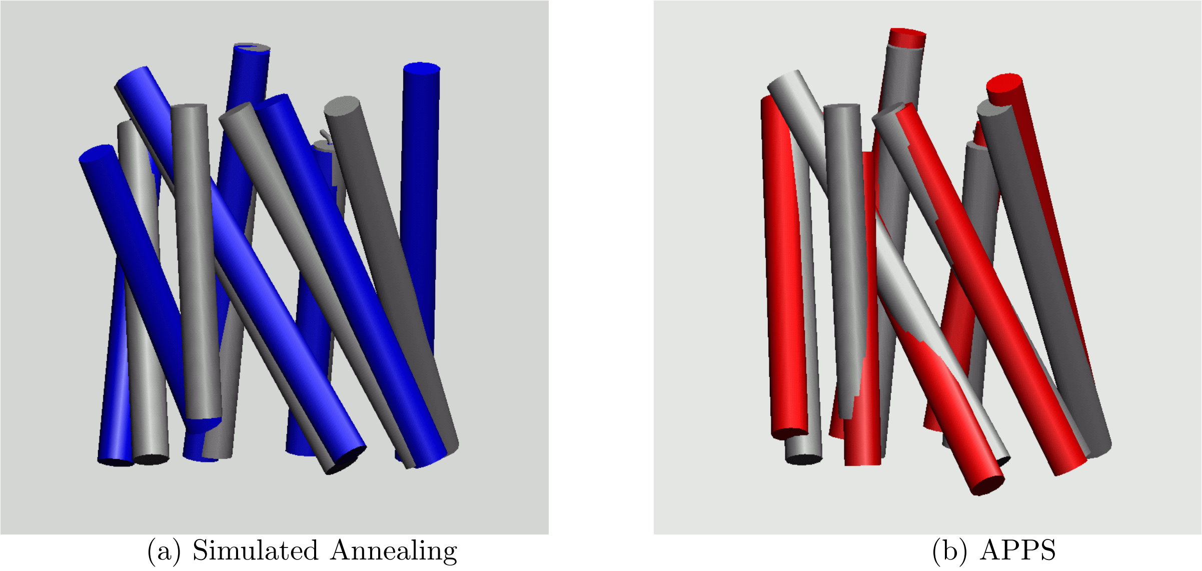
In our implementation, we opted to use the MPI mode of APPSPACK[2]. This mode requires a minimum of three processors: one master agent to coordinate the search, one cache agent to save and look up points at which the function has already been evaluated, and at least one worker to perform function evaluations. The default MPI version of APPSPACK requires that the function evaluations be run as seperate executables and communicates with the worker tasks via file input and output. In our case, the system calls and file I/O add substantial time to the overall runtime. Hence, we customized APPSPACK to avoid this overhead.

One of the main advantages of APPS is that it requires few parameters and very little tuning. We used the default values for all the parameters except the convergence tolerance which we set to be 0.01. We also note that our implementation uses the coordinate direction search pattern.

5 Numerical Results

In this section, we present some numerical results obtained using experimental distance constraints for rhodopsin. Rhodopsin is a transmembrane protein that is located in the retinal rods of the eye, and it plays a role in vision. It is a G-protein-coupled receptor and is made up of 7 transmembrane helices and thus has 42 variables in its structure determination problem. The 3-D structure of the dark-adapted form of rhodopsin is known, having been determined

[2] APPSPACK is available in MPI, PVM and serial modes.


(a) Simulated Annealing

(b) APPS

Figure 3: In both cartoons, the gray cylinders represent the α helices of dark-adapted rhodopsin. On the left, in picture (a), the blue cylinders show the locations of the helices found using simulated annealing. In picture (b) on the right, the positions of the helices as determined by APPS are depicted by the red cylinders.

using x-ray crystallography [36]. Moreover, a set of experimental distance constraints for dark-adapted rhodopsin has been compiled in [57]. Thus, dark-adapted rhodopsin is an appropriate test case for our numerical experiments. Because we are using a known structure in our tests, we can compute the difference between the true structure and any other structure using RMSD. Although we cannot use RMSD when trying to ascertain structures that have not yet been determined, we use it in our study to add clarity to some comparisons.

In our first test case, we randomized the true structure of rhodopsin to give us an initial guess. The subsequent starting structure has an initial Bundler score of $11,342.56$ and an RMSD of 15.02. We first tried optimizing this structure using SA with a starting temperature of 500 and 290 temperature cycles. After fine tuning the algorithm, the best structure we were able to produce has a score of 377.21 and an RMSD of 4.54. Next, we applied the APPS algorithm. This method required no fine tuning and on our first try, we were able to produce a structure with a score of 122.59 and an RMSD 3.41. Figure 3 shows the spatial positions of the helices relative to the known structure. Note that APPS determines the orientation of all seven helices relatively well. In contrast, two of the helices determined by SA are not a good match.

To make a more complete comparison, we also consider the computational efficiency of each method. As previously discussed, SA often requires extensive computational work. Our numerical test was no exception. The SA algorithm required $81,800$ function evaluations and 61 hours of run time on a single processor. In comparison, the APPS algorithm required only $32,458$ function evaluations and 17 minutes of run time on 86 processors. Since the two tests were run on the same machine, we can conclude that APPS was more efficient than SA as it required fewer function evaluations. Moreover, since APPS used 84 worker nodes, each processor

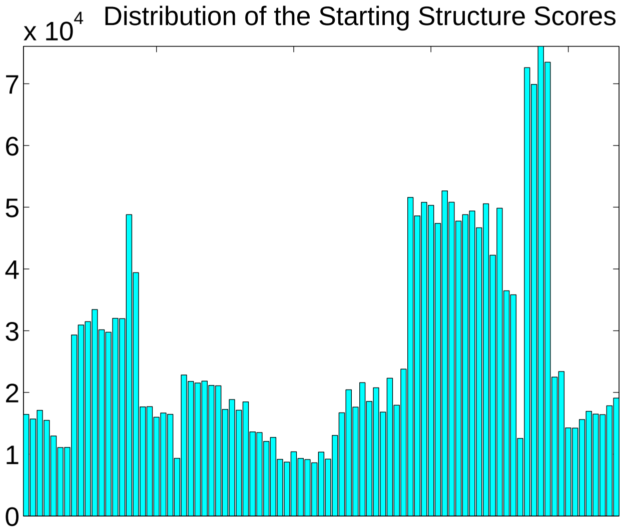


Figure 4: This graph shows the distribution of the Bundler scores for the 87 initial structures generated using the procedure outlined in [17]. The average initial score is $26,555$ with a maximum of $76,080$ and a minimum of $8,608$.

completed about 386 function evaluations. If SA were parallelized in the most efficient manor possible so that it could be executed on 86 processors, each processor would need to compute approximately 950 function evaluations, and it would still take almost 45 minutes to obtain a solution.

For our second set of tests, 87 structures were generated using the procedure outlined in [17] and a set of 27 distance constraints, $\mathcal{D}_1$, obtained from [57]. This procedure resulted in structures that have no experimental distance penalty, i.e., $P_E = 0$, where P_E is as defined in (3.3), for each of the 87 structures with respect to $\mathcal{D}_1$. Hence, to fully test the capabilities of the optimization methods, we use a different set of distance constraints, $\mathcal{D}_2$. The set $\mathcal{D}_2$ contains upper and lower bounds for the same 27 pairs of atoms as $\mathcal{D}_1$, but the range of these bounds is tighter as detailed in [45, 57]. The average Bundler score of the starting structures is $26,555$. The distribution of these scores is shown in figure 4.

To optimize the 87 structures, we applied both APPS and SA. In this case, the SA algorithm used a starting temperature of 300 and completed 125 temperature cycles. The results of this second test set are displayed in figures 5 and 6. By using 87 starting structures, we are applying 87 different starting points. As figure 5 shows, APPS achieves a much wider variety of final scores than SA. However, this is no surprise since APPS is a local optimization method. Moreover, it appears that a few starting structures get stuck in bad local minimas. In contrast, SA is a global method. In theory, all 87 starting structures should achieve the same score. We do catch a glimpse of this in figure 5. Note that 40 of the 87 structures attain a score that is between 111 and 117 However, there are two structures with significantly lower scores and many other structures whose final scores are considerably higher.

We can conclude that overall, this implementation of SA more *effectively* reduces the Bundler score than APPS. However, some of the scores achieved by APPS are comparable. Furthermore,

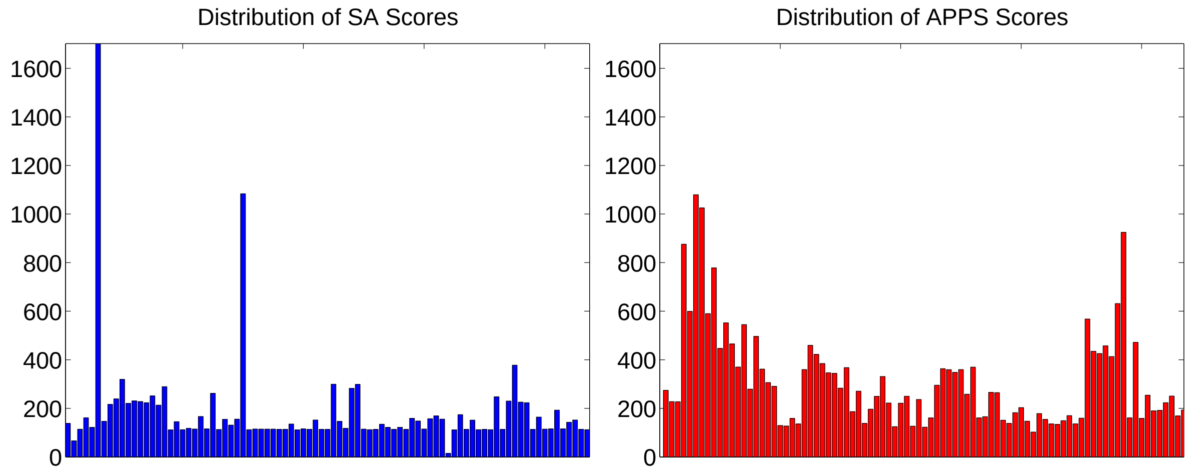


Figure 5: The graph on the left (in blue) shows the final Bundler scoring distribution for the SA algorithm. About half the structures achieve the same score. This is not unexpected since SA is a global method. However, there are some high scores which are difficult to explain. On the right (in red) is the final Bundler scoring distribution for the APPS algorithm. Despite the fact that SA is more effective overall, APPS does achieve some low scores which are comparable to those attained by SA.

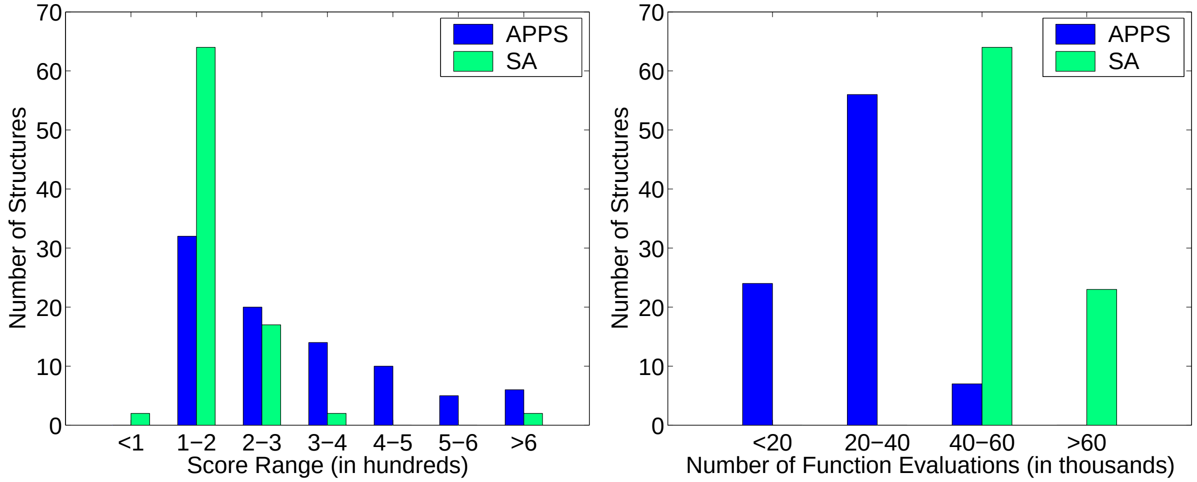


Figure 6: The bar graph on the left summarizes the final APPS and SA Bundler scoring function values (in hundreds) and the graph on the right summarizes the number of function evaluations (in thousands) required to achieve these results. This implementation of SA uses an initial temperature of 300 and completes 125 temperature cycles. Although SA more effectively reduces the Bundler score, it requires significantly more function evaluations.

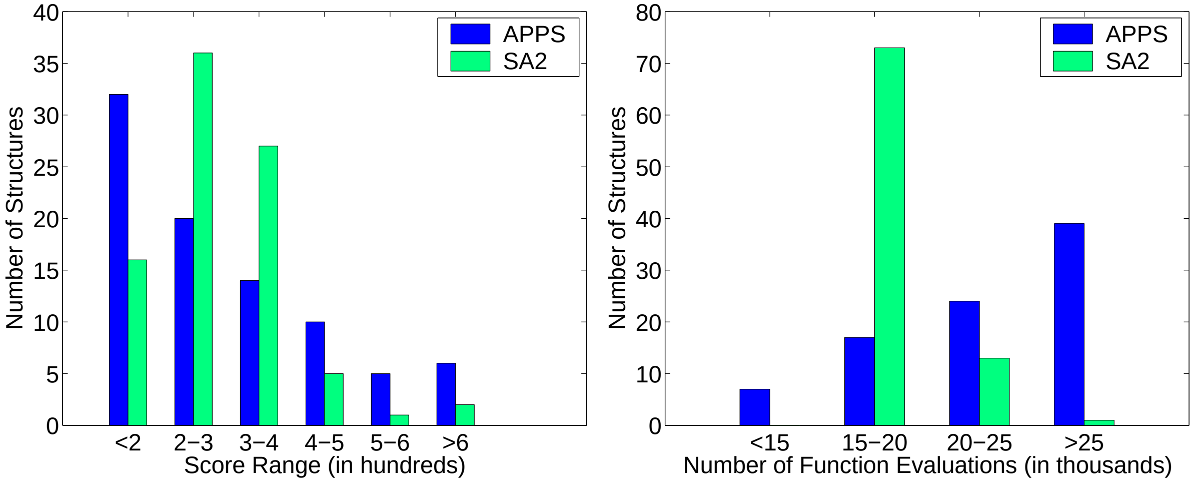


Figure 7: The bar graph on the left compares the final values of the Bundler scoring function (in hundreds) attained by APPS and SA2 and the graph on the right shows the number of function evaluations (in thousands) required to achieve the results. The simulated annealing algorithm, SA2, uses an initial temperature of 300 and does 65 temperature cycles. Note that the average number of function evaluations performed by APPS and SA2 is comparable. However, APPS appears to have more success reducing the scoring function below 200.

as figure 6 shows, the computational cost of the success of SA is quite high. It requires a minimum of $49,500$ function evaluations. In comparison, the maximum number of function evalutions needed by APPS is $48,812$ and 24 of the runs required less than $20,000$. Therefore, we conclude that APPS more *efficiently* reduces the Bundler scoring function.

We now want to more closely examine SA and make a more complete comparison of APPS and SA by reducing the number of SA function evaluations. One way this can be achieved is by reducing the number of temperature cycles. We use SA2 to denote the results of the SA procedure after only 60 temperature cycles, or approximately one-third of the number of function evaluations of the previous implementation. The results of this comparison are shown in figure 7. Here, APPS and SA now do a similar number of function evaluations. The SA algorithm is no longer more effective than APPS at reducing the scoring function. In fact, SA now attains only 16 scores below 200 while APPS achieves twice that many. Moreover, the distribution of the final SA Bundler scores is now much wider, as shown in figure 9 in the appendix.

Another way to reduce the number of SA temperature cycles is to use a lower starting temperature. To test this procedure, we use an initial temperature of 30 and do 75 temperature cycles. By beginning with a lower temperature, we will not accept as many randomized steps and thus are in effect doing a more localized search. We use SA3 to denote this test and summarize its results in figure 8. Note that we are again able to significantly reduce the number of SA function evaluations. However, the SA algorithm is no longer very successful at reducing the Bundler scoring function. Only two of the SA final scores are below 300 whereas 52 of the APPS final scores are below 300. The final Bundler scoring distribution given in figure 10 (in

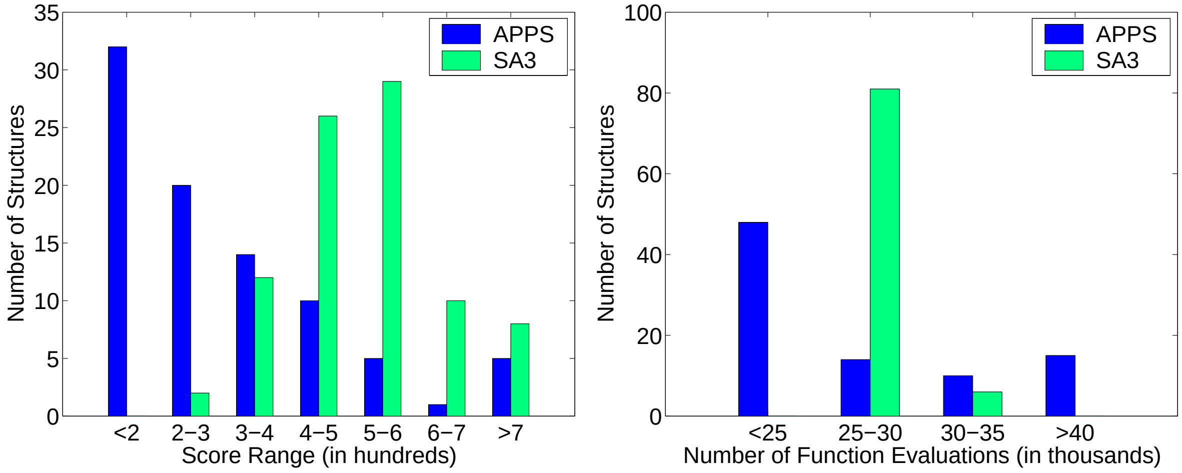


Figure 8: The bar graph on the left shows the Bundler scoring function values (in hundreds) and the graph on the right displays the number of function evaluation (in thousands) performed by the APPS and SA3 algorithms. The simulated annealing algorithm SA3 uses an initial temperature of 30 and completes 75 temperature cycles. Overall, APPS more effectively reduces the scoring function despite the fact that number of function evaluations performed by APPS and SA3 is comparable.

the appendix) further illustrates the ineffectiveness of SA. Therefore, we can conclude that the simulated annealing algorithm that uses these particular parameters, a low initial temperature and a small number of temperature cycles, is not a viable alternative for solving our problem.

6 Conclusions

Our numerical tests illustrate some of the previously mentioned characteristics of simulated annealing. First, the algorithm is slow and requires significant computational work. If too few temperature cycles are completed, the algorithm does not perform well. Secondly, the choice of SA parameters can greatly affect the outcome of the algorithm. In our tests, we varied only the number of temperature cycles and the initial temperature and obtained remarkably varied results. Finally, we did see a glimmer of the global convergence property of simulated annealing. In the test with 125 temperature cycles, 40 of the 87 different starting structures achieved essentially the same score.

Similarly, our experiments demonstrate some features of APPS. In this study, the most notable advantage of APPS is that it requires little to no parameter tuning. We opted to use a relatively large convergence tolerance. Decreasing this parameter results in an increase in the total number of function evaluations but only minor reduction in the final Bundler scores. By setting a relatively high convergence tolerance, we were able to limit the number of function evaluations and still achieve reasonable final scores. We did see some examples of structures getting stuck in local minima; however, APPS is not a global optimization method and this is to be expected. Moreover, this problem can be overcome by using multiple starting points.

Solving the transmembrane protein struture determination problem requires an optimization method which is both effective and efficient. As previously discussed, the Bundler scoring function incorporates real data obtained via laboratory experiment. Hence, there is a certain amount of noise in our objective function. At present, there is no regularization term in the Bundler scoring function to prevent fitting this noise, and hence it is not productive to apply an optimization algorithm that yields a structure with a Bundler score of zero. Moreover, we have observed that small variations in Bundler scores results in only noise level differences. Therefore, we do not require an extremely high level of accuracy in the optimization. Instead, it is to our benefit to use an algorithm which sacrafices some accuracy to improve the overall efficiency. The procedure we used in our numerical experiments to generate the 87 initial structures is part of the transmembrane protein structure determination process proposed in [17, 45]. Future projects will require optimizing thousands of structures to attain one final candidate which can be further studied in the laboratory. Hence, computational time is of the essence, and we favor optimization methods that limit the number of computations. To illustrate this point, we note that SA required approximately 5 million function evalutions to optimize the 87 structures in our test while APPS required only about 2.2 million. Moreover, using local optimization methods is not a disadvantage on such a project. Enough significantly different initial points will be used, and thus the final outcome will not be adversely affected if some of the starting points end up stuck in bad local minima.

In this paper, we discuss one particular aspect of the transmembrane protein structure identification method propsed in [17, 45], optimizing the Bundler scoring function. However, there are many other interesting and important facets of this computational biology problem. Future work includes investigating alternative optimization methods and making minor modifications to the Bundler function.

Acknowledgments

We gratefully acknowledge the Mathematical, Information, and Computational Sciences Program of the United States Department of Energy and the Laboratory Directed Research and Development program at Sandia National Laboratories for their support of this research. Sandia National Laboratories is a multi-program laboratory operated by Sandia Corporation, a Lockheed Martin Company for the United States Department of Energy under contract DE-AC04-94AL85000.

Appendix

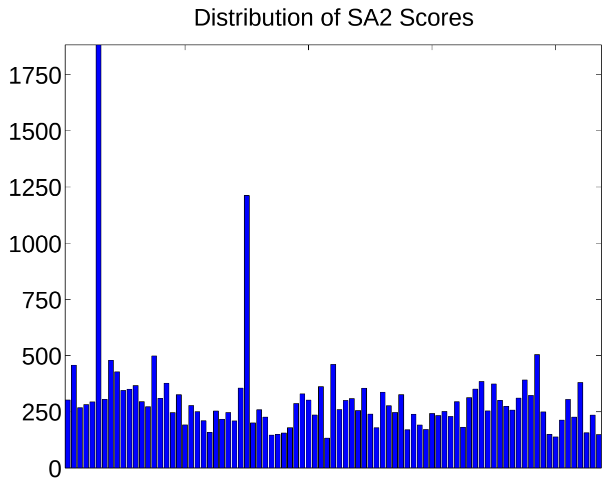


Figure 9: This graph shows the distribution of the Bundler scores achieved by the SA2 algorithm. SA2 is an implementation of simulated annealing that uses the geometric cooling schedule with $\alpha = 0.95$, an initial temperature of 300, and 60 temperature cycles. The average final Bundler score is 305.8 with a maximum of 1882.6 and a minimum of 132.4.

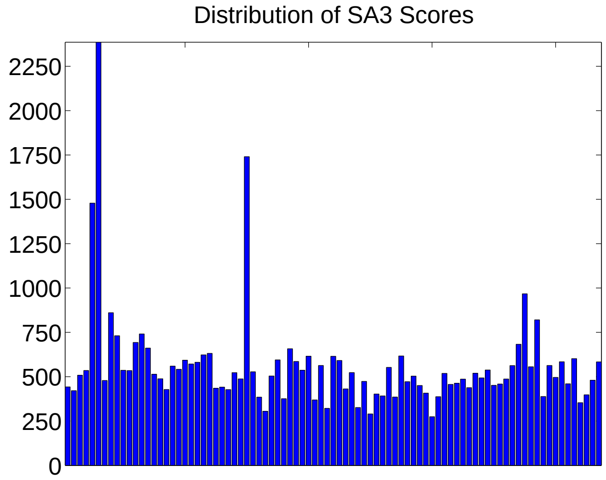


Figure 10: This is a graph of the distribution of the Bundler scores achieved by the SA3 algorithm. SA3 is an implementation of simulated annealing that uses the geometric cooling schedule with $\alpha = 0.95$, an initial temperature of 30, and 75 temperature cycles. The average final Bundler score is 561.1 with a maximum of 2385.7 and a minimum of 274.1.

References

[1] E.H. Aarts, J. Korst, and P.J. van Laarhoven. *Local Search in Combinatorial Optimization*, chapter 4, Simulated Annealing. John Wiley and Sons, 1997.

[2] L. Adamian, R. Jackups, Jr., T.A. Binkowski, and J. Liang. Higher-order interhelical spatial interactions in membrane proteins. *J. Mol. Biol.*, 327:251–272, 2003.

[3] L. Adiman and J. Liang. Helix-helix packing and interfacial pairwise interactions of residues in membrane proteins. *J. Mol. Biol.*, 311:891–907, 2001.

[4] M.M. Ali and C. Storey. Aspiration based simulated annealing algorithm. *J. Glob. Opt.*, 11:181–191, 1997.

[5] 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 Research*, 2000.

[6] J.U. Bowie. Helix packing in membrane proteins. *J. Mol. Biol.*, 272:780–789, 1997.

[7] J.U. Bowie. Helix-bundle membrane protein fold templates. *Protein Sci.*, 8:2711–2719, 1999.

[8] A.T. Brünger. *X-PLOR: A System for X-ray Crystallography and NMR, Version 3.1.* Department of Molecular Biophysics and Biochemistry, Yale University, New Haven, CT, 1992. `http://www.ocms.ox.ac.uk/mirrored/xplor/manual/htmlman/htmlman.html`.

[9] A.T. Brünger, P.D. Adams, and L.M. Rice. New applications of simulated annealing in X-ray crystallography and solution NMR. *Structure*, 5:325–336, 1997.

[10] B.J. Campbell, G. Bellussi, L. Carluccio, G. Perego, A.K. Cheetham, D.E. Cox, and R. Millin. The synthesis of the new zeolite, ERS-7, and the determination of its structure by simulated annealing and synchrotron X-ray powder diffraction. *Chem. Commun*, pages 1725–1726, 1998.

[11] G.S. Couch, E.F. Pettersen, C.C. Huang, and Ferrin T.E. Annotating PDB files with scene information. *J. Molec. Graphics*, 13:153–158, 1995.

[12] J.E. Dennis, Jr. and V. Torczon. Direct search methods on parallel machines. *SIAM J. Opt.*, 1:448–474, 1991.

[13] H. Dobbs, E. Orlandini, R. Bonaccini, and F. Seno. Optimal potentials for predicting inter-helical packing in transmembrane proteins. *Proteins*, 49:342–349, 2002.

[14] E.D. Dolan, R.M. Lewis, and V.J. Torczon. On the local convergence properties of parallel pattern search. Technical Report 2000-36, NASA Langley Research Center, Institute for Computer Applications in Science and Engineering, Hampton, VA, 2000.

[15] S. Elmohamed, G. Fox, and P. Coddington. A comparison of annealing techniques for academic course scheduling. In *2nd International Conference on the Practice and Theory of Automated Timetabling*, pages 146–166, Syracuse, NY, Apr. 1998. Practice and Theory of Automated Timetabling.

[16] J.L. Faulon, M.D. Rintoul, and M.M. Young. Constrained walks and self-avoiding walks: implications for protein structure determination. *J. Phys. A.: Math. Gen.*, 35:1–19, 2002.

[17] J.L. Faulon, K. Sale, and M.M. Young. Exploring the conformational space of membrane protein folds matching distance constraints. *Protein Sci.*, 12:1750–1761, 2003.

[18] S. Geman and D. Geman. Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images. *IEEE Trans. Patt. Anal. Mach. Intel.*, 6:721–741, 1984.

[19] A. Ghosh, R. Elber, and H.A. Scheraga. An atomically detailed study of the folding pathways of protein A with the stochastic difference equation. *PNAS*, 99:10394–10398, 2002.

[20] D. S. Goodsell and A. J. Olson. Automated docking of substrates to proteins by simulated annealing. *Proteins: Str. Func. and Genet.*, 8:195–202, 1990.

[21] P. Herzyk and R.E. Hubbard. Using experimental information to produce a model of the transmembrane domain of the ion channel phospholamban. *Biophys. J.*, 74:1203–1214, 1998.

[22] P. Hough and T.G. Kolda. *APPSPACK: Asynchronous Parallel Pattern Search.* Sandia National Laboratories. `http://software.sandia.gov/appspack/`.

[23] P.D. Hough, T.G. Kolda, and V.J. Torczon. Asynchrounous parallel pattern search for nonlinear optimization. *SIAM J. Sci. Comput.*, 23:134–156, 2001.

[24] S. Jayasinghe, K. Hristova, and S.H. White. Energetics, stability, and prediction of transmembrane helices. *J. Mol. Biol.*, 312:927–934, 2001.

[25] S. Jayasinghe, K. Hristova, and S.H. White. MPtopo: A database of membrane protein topology. *Protein Sci.*, 10:455–458, 2001.

[26] S. Kirkpatrick, C.D. Gerlatt, Jr., and M.P. Vecchi. Optimization by simulated annealing. *Science*, 220:671–680, 1983.

[27] G. Kliewer and S. Tschöke. A general parallel simulated annealing library (parSA) and its applications in industry. PAREO 1998: First meeting of the PAREO working group on Parallel Processing in Operations Research, Versailles, France,, July 1998. `http://www.uni-paderborn.de/fachbereich/AG/monien/SOFTWARE/PARSA/`.

[28] T.G. Kolda, R.M. Lewis, and V. Torczon. Optimization by direct search: new perspectives on some classical and modern methods. *SIAM Rew.*, 45:385–482, 2003.

[29] T.G. Kolda and V.J. Torczon. On the convergence of asynchronous parallel pattern search. Technical Report SAND2001-8696, Sandia National Laboratories, 2001.

[30] A.R. Leach. *Molecular Modeling: Principles and Applications.* Prentice Hall, 2nd edition, 2001.

[31] F. H. Allisen Lee. *Parallel simulated annealing on a message-passing multi- computer*. PhD thesis, Utah State University, 1995.

[32] R.M. Lewis and V.J. Torczon. Rank ordering and positive basis in pattern search algorithms. Technical Report 96-71, NASA Langley Research Center, Institute for Computer Applications in Science and Engineering, Hampton, VA, 1996.

[33] N. Metropolis, A.W. Rosenbluth, M.N. Rosenbluth, A.H. Teller, and E. Teller. Equations of state calculations by fast computing machines. *J. Chem. Phys.*, 21:1087– 1092, 1958.

[34] B.M.E. Moret and H.D. Shapiro. *Algorithms from P to NP*, volume I. Benjamin/Cummings Publishing Company, Redwood City, CA, 1991.

[35] G.V. Nikiforovich, S. Galaktionov, J. Balodis, and G.R. Marshall. Novel approach to computer modeling of seven-helical transmembrane proteins: Current progress in the test case of bacteriorhodopsin. *Acta Biochimica Polinica*, 48:53–64, 2001.

[36] K. Palczewski, T. Kumasaka, T. Hori, C.A. Behnke, H. Motoshima, B.A. Fox, D.C. Le Trong, I.and Teller, T. Okada, R.E. Stenkamp, M. Yamamoto, and M. Miyano. Crystal structure of rhodopsin: A G protein-coupled receptor. *Science*, 289:739–745, 2000.

[37] T.D.J. Perkins and P.M. Dean. An exploration of a novel strategy for superposing several flexible molecules. *J. Comp.Aided Mol. Design*, 7:155–172, 1993.

[38] M. Piccioni. Combined multistart-annealing algorithm for continuous global optimization. Technical Report TR87-45, University of Maryland, 1987.

[39] M. Pincus. Monte Carlo method for the approximate solution of certain types of constrained optimization problems. *Oper. Res*, 18:1225–1228, 1970.

[40] M.J.D. Powell. Direct search algorithms for optimization calculations. *Acta Numer.*, 7:287–336, 1998.

[41] R.E. Randelman and G.S. Grest. N-City traveling salesman problem - Optimization by simulated annealings. *J. of Stat. Phys.*, 45:885–890, 1986.

[42] F.M. Richards. The interpretation of protein structures: total volume, group volume, distributions and packing density. *J. Mol. Biol.*, 82:1–14, 1974.

[43] G.D. Rose. Prediction of chain turns in globular proteins on a hydrophobic basis. *Nature*, 272:586–590, 1978.

[44] P. Salamon, P. Sibani, and R. Frost. *Facts, Conjectures, and Improvements for Simulated Annealing.* Monographs on Mathematical Modeling and Computation 7. SIAM, Philadelphia, PA, 2002.

[45] K.L. Sale, J.L. Faulon, G.A. Gray, and M.M. Young. Optimal bundling of the transmembrane helices of integral membrane proteins using sparse distance constraints. In preparation.

[46] Stephen White Laboratory at UC Irvine. *Membrane Proteins of Known 3D Structure.* `http://blanco.biomol.uci.edu/Membrane_Proteins_xtal.html`.

[47] G.S. Stiles. The effect of numerical precision upon simulated annealing. *Phys. Lett. A.*, 185, 1994.

[48] G.S. Stiles, K.W. Bosworth, T.W. Morgan, F.H. Lee, and R.J. Pennington. Parallel optimization of distributed database networks. In *Proc. First Int'l Conf. Applications of Transputers*, Amsterdam, 1989. IOS Press.

[49] V. Torczon. PDS: Direct search methods for unconstrained optimization on either sequential or parallel machines. Technical Report TR92-09, Rice University, Department of Computational & Applied Math, Houston, TX, 1992.

[50] V.J. Torczon. On the convergence of pattern search algorithms. *SIAM J. Opt.*, 7:1–25, 1997.

[51] University of Nijmegen, The Netherlands. *GPCRDB: Information system for G protein-coupled receptors (GPCRs).* `http://www.cmbi.kun.nl/7tm`.

[52] N. Vaidehi, W.B. Floriano, R. Trabanino, S.E. Hall, P. Freddolino, E.J. Choi, G. Zamanakos, and W.A. Goddard, III. Prediction of structure and function of G protrein-couple receptor. *PNAS*, 99:12622–12627, 2002.

[53] P.M.J. van Laarhoven and E.H.L. Aarts. *Simulated Annealing: Theory and Applications.* Dordrecht Reidel Publishing Company, Dordrecht, Holland, 1987. Republished in 1989 by Kluwer Academic.

[54] G. Vriend. WHAT IF: A molecular modeling and drug design program. *J. Mol. Graphics*, 8:52–56, 1990.

[55] S. Wilson and D. Bergsma. Orphan G-protein coupled receptors: novel drug targets for the pharmaceutical industry. *Drug Des Discov.*, 17:105–114, 2000.

[56] Z. Xiang and B. Honig. Extending the accuracy limits of prediction for side chain conformations. *J. Mol. Biol.*, 311:421–430, 2001.

[57] P.L. Yeagle, G. Choi, and A.D. Albert. Studies on the structure of the G-protein-coupled receptor rhodopsin including the putative G-protein binding site in unactivated and activated forms. *Biochemistry*, 40:11932–11937, 2001.

Distribution:

1 John Dennis
CAAM Dept, MS 134
Rice University
6100 Main St.
Houston, TX 77005-1892

1 Lisa Fauci
Tulane University
Department of Mathematics
New Orleans, LA 70118

1 Dianne O'Leary
Computer Science Department
University of Maryland
College Park, MD 20742

1 Michael Lewis
Department of Mathematics
College of William & Mary
P.O. Box 8795
Williamsburg, VA 23187-8795

1 Karin Leiderman
University of New Mexico
321 Sierra Place NE
Albuquerque, NM 87108

1 Charles Romine
U. S. Department of Energy
SC-31/GTN Bldg.
1000 Independence Avenue, SW
Washington, DC 20585-1290

1 Virginia Torczon
College of William & Mary
Department of Computer Science
P.O. Box 8795
Williamsburg, VA 23187-8795

1	MS 0310	Danny Rintoul, 9212
1	MS 0316	Chi-Chi May, 9212
1	MS 0316	Steve Plimpton, 9212
1	MS 1110	David Womble, 9214
1	MS 1111	Bruce Hendrickson, 9215
1	MS 9003	Ken Washington, 8900
1	MS 9217	Patty Hough, 8962
1	MS 9217	Steve Thomas, 8962
1	MS 9951	Len Napolitano, 8100
1	MS 9951	Joe Schroeniger, 8130
1	MS 9951	Malin Young, 8130
3	MS 9018	Central Technical Files, 8945-1
1	MS 0899	Technical Library, 9616
1	MS 9021	Classified Office, 8511/Technical Library, MS 0899, 9616
		DOE/OSTI via URL