Iterative Convex Quadratic Approximation for Global Optimization in Protein Docking

Roummel F. Marcia[1], Julie Mitchell[2], and J. Ben Rosen[3]

Abstract. An algorithm for finding an approximate global minimum of a funnel shaped function with many local minima is described. It is applied to compute the minimum energy docking position of a ligand with respect to a protein molecule. The method is based on the iterative use of a convex, general quadratic approximation that underestimates a set of local minima, where the error in the approximation is minimized in the L^1 norm. The quadratic approximation is used to generate a reduced domain, which is assumed to contain the global minimum of the funnel shaped function. Additional local minima are computed in this reduced domain, and an improved approximation is computed. This process is iterated until a convergence tolerance is satisfied. The algorithm has been applied to find the global minimum of the energy function generated by the Docking Mesh Evaluator program. Results for three different protein docking examples are presented. Each of these energy functions has thousands of local minima. Convergence of the algorithm to an approximate global minimum is shown for all three examples.

1 Introduction

Two important problems in computational biology can be formulated as that of finding the global minimum of a funnel (or basin) shaped function with many local minima. The function to be minimized represents the potential energy of a single molecule, or of two molecules joined together. In the first case, we wish to approximate the native structure of a protein molecule, given the amino acid sequence which describes the protein (the protein folding problem). In the second case we wish to determine computationally the correct location on a protein molecule where a ligand (drug) will attach to it, as well as the correct orientation of the ligand with respect to the protein molecule. This is called the docking problem. For both of these cases the basic assumption is that with a reliable potential energy function, the correct geometric structure corresponds to the global minimum of the energy function.

The Convex Global Underestimator (CGU) was developed for finding the global minimum of a function of the type described above. Initially the method was developed for, and applied to, the protein folding problem for a set of small model molecules [2,10]. Briefly CGU first computes a set of local minima to the energy function $f(x)$, in the space x in $\mathbb{R}^n$, of coordinates which determine the protein molecule structure (conformation space). A strictly convex, quadratic function $q(p, x)$, with s parameters p, is then computed, which underestimates all the local minima, and also minimizes the approximation error in the L^1 norm. The local minimum x^*_{pred} corresponding to the unique minimum x_{pred} of $q(p, x)$, with respect to x, is then an estimate of the location of the global minimum of the funnel shaped energy surface. A reduced rectangular domain in $\mathbb{R}^n$ is then determined, which contains x^*_{pred} and the local minimum $x_{\min}$ with the minimum function value, and additional local minima are computed in this domain. A convex quadratic underestimator is again computed, and the process iterated until $x^*_{\text{pred}} \approx x_{\min}$. For this application to small molecules, the quadratic function $q(p, x)$ was limited to a diagonal Hessian matrix. In this paper we apply a similar method to the protein-ligand docking problem. For this problem we need to use a convex, general quadratic underestimating function, in which the Hessian of $q(p, x)$ may contain nonzero, off-diagonal terms. The manner in which such an approximating function is computed has been described in [11]. The algorithm is described in Section 3, the

docking problem is summarized in Section 4, and the results of applying the algorithm to three different docking problems are given in Section 5.

2 General Convex Quadratic Approximation

The General Convex Quadratic Approximation (GCQA) algorithm determines a general quadratic function which is strictly convex, and underestimates a set of m data points in $I\!R^{n+1}$, $\{(x_1, f(x_1)), (x_2, f(x_2)), \cdots, (x_m, f(x_m))\}$, where $x_k \in I\!R^n$ for $k = 1, \cdots, m$, and $f\colon I\!R^n \to I\!R$. The $s = (n+1)(n+2)/2$ parameters in the quadratic function are determined so that the function is strictly convex, underestimates the data points, and minimizes the error in the approximation to the data in the L^1 norm. The number of data points required is $m \geq s$, usually $m = 4s$. This underestimator to the data points is used iteratively to attempt to find a good approximation to the global minimum of a function with many local minima. The potential energy function for the docking of a ligand on a protein is a function of this type as described in Section 4. The energy function is defined in a six-dimensional space ($n = 6$), where the six coordinates determine the relative positions of the protein and the ligand.

The protein-ligand energy function is computed by the program Docking Mesh Evaluator (DoME), as summarized in Section 4. An adaptive grid consisting of three-dimensional simplices is generated using Adaptive Poisson-Boltzmann Solver (APBS) [1]. DoME computes the energy function at each of the four vertices of each simplex, and linearly interpolates to get the energy at any interior point of a simplex. Thus once the function values at the vertices are computed, it is a relatively fast calculation to compute the function value for any specified orientation of the ligand with respect to the protein molecule. This energy function is a six-dimensional piecewise linear function. A local minimization therefore consists of minimizing this 6-D piecewise linear function.

The global optimization algorithm consists of a sequence of major iterations. The first major iteration consists of a coupled use of scanning and optimization. Initially, DoME scans the energy landscape for favorable configurations. The coordinates for these configurations are then used as starting points for local optimization. (This coupled optimization can be done once as a preprocessing step and need not be done again.) The best $m = 4s$ local minima are chosen as initial points to underestimate. The vertex with the minimum function value is denoted by x^*_{init}. The function value $f(x^*_{\text{init}})$ is the minimum value that is known from the adaptive grid calculation only. An initial six-dimensional hyperrectangular domain is constructed so as to include all $4s$ vertices. This is the initial domain $\mathcal{D}$. The major iterations of the global optimization algorithm are then carried out, as summarized next.

Each major iteration consists of two phases. Phase I of the first major iteration consists of the coupled optimization described previously. Phase I of subsequent major iterations consists of defining a hyperrectangular domain $\mathcal{D}$ in $I\!R^n$, and randomly generating m ($\geq s$) data points $\{x_1, \cdots, x_m\}$, and computing a corresponding local minimum from each of these points, to give x^*_k, and $f^*_k = f(x^*_k)$, $k = 1, \cdots, m$. Let $\mathcal{S}_{k_b}$ denote the set of k_b points with the smallest function values. (A typical value for k_b is 3.) In Phase II of each major iteration the data points (x^*_k, f^*_k), $k = 1, \cdots, m$, are underestimated by a general convex quadratic function, with an easily computed global minimum x_{pred}, which is a predicted value for the global minimum of the energy function $f(x)$. A corresponding local minimum x^*_{pred}, is computed, and is added to the set of data points. This new point x^*_{pred}, may replace one of the points in $\mathcal{S}_{k_b}$. The new domain $\mathcal{D}$ is then defined as the smallest hyperrectangle containing the points in $\mathcal{S}_{k_b}$. The next major iteration is started with this domain.

The domain size is typically reduced at each major iteration, and the iterations are terminated when the predicted global minimum x_{pred} is close to the local minimum with the smallest function value. Details of this algorithm are given in the next section. At

termination, as shown by the computational results in Section 5, many of the data points are interpolated by the convex quadratic approximation, so that this quadratic function, and its eigenvalues and eigenvectors give a good representation of the energy surface in the neighborhood of the presumed global minimum of the energy function.

The Phase II of the global optimization algorithm has been described in detail in an earlier paper [11] where the theory and computational results are given for fitting data points in $\mathbb{R}^n$ with a convex quadratic function. It is shown there that the convex quadratic approximation gives a good approximation to a nonconvex function. Specifically, the data points x_i are selected randomly in a hyperrectangle in $\mathbb{R}^n$, with n as large as 15, and the function values $f(x_i)$ are obtained from a convex quadratic function, with a positive or negative random perturbation of each function value. The results given in [11] show that the approximation essentially recovers the original unperturbed convex quadratic function. The computational method used to determine the convex quadratic approximation in Phase II of the global optimization algorithm is the large-scale constrained optimization package NPSOL [3].

An earlier version of the convex quadratic approximation has been used successfully to determine the global minimum energy conformation of small protein models [2,10]. This earlier version, called the Convex Global Underestimator (CGU) algorithm limited the quadratic function to diagonal terms only. Using diagonal terms only, CGU was also used for docking small ligands. This problem required only good convex approximation in the three space variables [9]. This limited its usefulness if the best convex quadratic approximation actually included off-diagonal terms. The protein-ligand docking energy functions generated by DoME all require significant off-diagonal terms to get a good approximation, as described in Section 4.

3 Algorithmic Details

3.1 Phase I: Generating Points

After the first iteration where the the initial seed points are computed by DoME, the data points used to construct the approximation are obtained as follows. Given the domain $\mathcal{D}$, m points are randomly generated such that $x_k \in \mathcal{D}$, $k = 1, \cdots, m$ and $f(x_k) \leq f_{\max}$ for some scalar $f_{\max}$. Since the points x_k are chosen at random, some function values $f(x_k)$ may be very large. Such data points can be considered as outliers in the underestimator approximation process. Since we are minimizing the error in the L^1 norm, these outliers will not affect the shape of the underestimator $q(p, x)$. In fact, only the interpolated data points will determine the s parameters in $q(p, x)$ (see for example [12]). The approximation error value will however be greatly increased by these outliers. If two (or more) random points are close to each other, only the one with the minimum $f(x)$ will be interpolated. Any random points with large function values can therefore be eliminated without changing the underestimator . Eliminating such points will significantly reduce the computation time, and also the size of the approximation error. Since $f(x_{\text{init}})$ is known from DoME to be negative, we chose the cutoff value $f_{\max} = 2.0 \times 10^2$. Increasing this value does not change the underestimator. The randomly generated points are then used as initial values for local minimization. A non-gradient-based optimization algorithm (the Hooke-Jeeves direct search algorithm [6]) is used for computing the local minima since the energy function is piecewise linear and therefore non-differentiable. The set of local minima is denoted by $\{x_1^*, \cdots, x_m^*\}$. Let $x_{\min}$ be one of the local minima with the smallest function value, i.e., $x_{\min} \in \{x_k^*\}$ with $f(x_{\min}) \leq f(x_k^*)$ for all k. While it is possible that a local minima x_k^* might lie outside of the domain, this has not occurred in any of the examples we have tested so far.

3.2 Phase II: Computing the Predicted Global Minimum

Given the m data points, a strictly convex quadratic function $q(x)$ such that $q(x_k^*) \leq f(x_k^*)$ for all k is constructed using GCQA (see Section 2). Let x_q be the unique global minimum of $q(x)$. If x_q is feasible with respect to $\mathcal{D}$, then the initial predicted global minimum x_{pred} is defined by x_q. If not, i.e., $x_q \notin \mathcal{D}$, then x_{pred} is defined as the solution of the quadratic programming problem

$$\begin{aligned} \underset{x \in \mathbb{R}^n}{\text{minimize}} \quad & q(x) \\ \text{subject to} \quad & x_L \leq x \leq x_U, \end{aligned} \tag{QP}$$

where x_L and x_U define the domain $\mathcal{D} = \{x \colon x_L \leq x \leq x_U\}$.

Using x_{pred} as an initial estimate, a local minimization is performed on $f(x)$ to obtain a local minimum x_{pred}^*. This local minimum is used as the predicted global minimum for the energy function.

3.3 Domain Trimming

The GCQA algorithm consists of a number of major iterations. At each major iteration the volume of the search domain $\mathcal{D}$ is usually reduced. Initially the search domain is chosen sufficiently large so that it will contain the global minimum x_{gmin} of $f(x)$. At each subsequent iteration it is desired to reduce the volume of $\mathcal{D}$ as much as possible, subject to its containing x_{gmin}, with high probability. This is accomplished by finding the smallest rectangular domain that contains, in its interior, both the predicted global minimum, x_{pred}^*, and the local minimum point x_{min} with the minimum function value. The assumption is that x_{gmin} is close to at least one of these points. Based on the computational results obtained (Section 5) this assumption is valid for the examples considered. Typically, the distance between the new x_{min} and x_{pred}^* will decrease with each major iteration. That is, the underestimator is an improved approximation to $f(x)$ in the neighborhood of x_{min}.

Given the predicted global minimum x_{pred}^*, we wish to reduce the hyperrectangle to obtain a new search domain. The reduced domain must contain a set of k_b points, $\mathcal{S}_{k_b}$, with the lowest function values and if appropriate, x_{pred}^*. In general, it is assumed that at least one of the points in $\mathcal{S}_{k_b}$ will be near the global minimum. Since the new domain must contain the global minimum, these points are required to be in the *strict* interior of the new domain. This new domain $RD(\{x_k\}; \rho)$ is obtained by computing the smallest hyperrectangle containing $\{x_k\}$ and by padding the sides by a factor of ρ. More precisely, if the smallest and largest i-th coordinate for all x_k are denoted by

$$\begin{aligned} (x)_i^{\min} &= \{(x_k)_i \colon (x_k)_i \leq (x_j)_i \text{ for all } j\} \\ (x)_i^{\max} &= \{(x_k)_i \colon (x_k)_i \geq (x_j)_i \text{ for all } j\}, \end{aligned}$$

then the bounds of $RD(\{x_k\}; \rho)$ are given by

$$\begin{aligned} (x_L)_i &= (x)_i^{\min} - \rho \left((x)_i^{\max} - (x)_i^{\min} \right) \\ (x_U)_i &= (x)_i^{\max} + \rho \left((x)_i^{\max} - (x)_i^{\min} \right). \end{aligned}$$

In our numerical experiments, the value $\rho = 0.1$ is chosen.

If the predicted global minimum x_{pred}^* is in $\mathcal{D}$, then the new domain will contain x_{pred}^* and the local minimum with the lowest function value, x_{min}, i.e., $\mathcal{D} = RD(x_{\text{pred}}^*, x_{\text{min}}; \rho)$. If $x_{\text{pred}}^* \notin \mathcal{D}$, then the new domain is defined to contain the k_b best points: $\mathcal{D} = RD(\mathcal{S}_{k_b}; \rho)$.

There are two possible situations for which the size of the domain $\mathcal{D}$ does not decrease. The first is when the local minimum values $f(x_k^*)$ and the predicted global minimum value

$f(x^*_{\text{pred}})$ are all greater than $f(x_{\min})$, where $x_{\min}$ is the previous best local minimum. The second is when x^*_{pred} is exterior to $\mathcal{D}$, and $f(x^*_{\text{pred}}) < f(x_{\min})$. In this case $\mathcal{D}$ is expanded to include x^*_{pred}. These situations only occurred in Example 1 (Table 5.1). If the termination tolerance (Section 3.4) is not satisfied when the number of major iterations (= iter_{mid}), termination is forced by reducing the length of each edge of $\mathcal{D}$ by a factor $\gamma \in (0, 1)$, so that the volume of $\mathcal{D}$ is reduced by γ^6 at each major iteration, with $\mathcal{D}$ centered at $x_{\min}$. The values $\text{iter}_{mid} = 20$ and $\gamma = 0.5$ were used to ensure that calculations would terminate within a day. This type of domain reduction was needed only for Example 1, and the maximum number of major iterations needed was always ≤ 28.

3.4 Termination

The algorithm is terminated when the predicted global minimum x_{pred} corresponds to the local minima with the lowest known function value $x_{\min}$, i.e., the relative error is smaller than some prescribed tolerance:

$$\frac{\|x_{\min} - x^*_{\text{pred}}\|_2}{\|x_{\min}\|_2} < \tau,$$

for some $\tau > 0$. In our computations, the value $\tau = 10^{-4}$ is used.

4 Docking Mesh Evaluator

The Docking Mesh Evaluator (DoME) is a program for predicting the structure of bound protein-protein or other macromolecular complexes by modeling the potential energy landscape and finding its global minima. This model is based on the effects of solvents on biological interactions and the pairwise interactions between the atoms in each molecule. The electrostatic potential is described by a second order nonlinear partial differential equation known as the Poisson-Boltzmann equation:

$$-\nabla \cdot (\epsilon \nabla u) + \bar{\kappa}^2 \sinh(u) = \rho, \tag{PBE}$$

where u is the electrostatic potential, ρ is a normalized sum of charges taken at atomic centers, ϵ describes the degree of polarizability, and $\bar{\kappa}$ indicates the impact of temperature and ionic concentration on solvent effects. This model approximates the statistical average of electrostatic effects for a macromolecule in solution. The attractive and repulsive interaction between the atoms is modeled by the Lennard-Jones (12-6) pairwise potential:

$$u_{ij} = 4\varepsilon_{ij} \left(\left(\frac{\sigma_{ij}}{r_{ij}} \right)^{12} - \left(\frac{\sigma_{ij}}{r_{ij}} \right)^{6} \right), \tag{LJ}$$

where ε_{ij} and σ_{ij} are specific Lennard-Jones parameters and r_{ij} represents the Euclidean distance between atoms i and j. Note that the $\frac{1}{r_{ij}^{12}}$ term in (LJ) describes the short-range repulsive potential while the $\frac{1}{r_{ij}^{6}}$ term describes the long-range attractive tail of the potential between two particles.

DoME evaluates the total energy function by using the numerical solution of the Poisson-Boltzmann Equation generated by the Adaptive Poisson-Boltzmann Solver (APBS) and the Finite Element Toolkit (FEtk) [1,5] on a high-resolution adaptive mesh and interpolating at the four vertices of each simplex. The Lennard-Jones potential function values are then computed at all vertices of the adaptive mesh, using the van der Waals radii for target

molecule atoms and assuming a probe atom the size of carbon. The Lennard-Jones potential can be interpolated nearly identically to the electrostatic potential. This approach to computing interaction energies is similar to that of other fast molecular docking methods, such as AutoDock[4] and DOCK[8]. These methods interpolate precomputed potential function values in order to lessen the computational expense. The major difference is that DoME interpolates against adaptive meshes rather than regular grids, and electrostatic potentials are defined using solutions to the Poisson-Boltzmann equation.

The energy function generated by DoME represents a typical landscape for docking problems. Since the correct docking position is known for each of the three examples used here, we can compare the energy function value at this docking position with the minimum function value computed for the same function by the global optimization algorithm. Based on this information, we are aware that the energy function needs to be further improved. This is based on the fact that, for all three examples, the value of the energy function at the known docking position is significantly higher (by as much as 60%) than the global minimum value obtained by Algorithm 5.1. Clearly this means that the current energy function does not attain its global minimum at the correct docking position, and therefore needs to be suitably modified. Work on this is currently in progress, however we are confident that with a better energy model, the predicted global minima identified by GCQA will give biologically realistic results.

5 Numerical Results

We first present the algorithm for the global optimization method implemented in DoME.

5.1 The algorithm

Algorithm 5.1. Global optimization algorithm in DoME

Define γ $(0 < \gamma < 1)$; τ; $\text{iter}_{\max}$; $f_{\max}$; k_b; iter_{mid};
Initialize x_U, x_L;
Define $\mathcal{D} = \{x\colon x_L \le x \le x_U\}$;
Initialize iter = 1;
while $(\|x_{\min} - x^*_{\text{pred}}\|_2/\|x_{\min}\|_2 > \tau)$ and $(\text{iter} \le \text{iter}_{\max})$ **do**
    Generate m points $\{x_k\}$ such that $x_k \in \mathcal{D}$ and $f(x_k) \le f_{\max}$;
    Perform local minimization on $\{x_k\}$ to obtain $\{x^*_k\}$;
    if $(x^*_k \notin \mathcal{D})$ **then**
        if $(f(x^*_k) \le f(x_{\min}))$ **then**
            Expand $\mathcal{D}$ such that $x^*_k \in \mathcal{D}$;
        else
            $x^*_k \leftarrow x_k$;
    Define $x_{\min} \in \{x^*_k\}$ such that $f(x_{\min}) \le f(x^*_k)$ for all k;
    GCQA: Compute and minimize approximation $q(p, x)$ subject to $x \in \mathcal{D}$
        to obtain predicted global minimum x_{pred};
    Perform local minimization on x_{pred} to obtain x^*_{pred};
    if $(x^*_{\text{pred}} \notin \mathcal{D})$ **then**
        if $(f(x^*_{\text{pred}}) \le f(x_{\min}))$ **then**
            Expand $\mathcal{D}$ such that $x^*_{\text{pred}} \in \mathcal{D}$;
        else
            Let $\mathcal{S}_{k_b}$ be the set of k_b best points.

$\mathcal{D} \leftarrow RD(\mathcal{S}_{k_b}; \rho)$;
else
$\mathcal{D} \leftarrow RD(x^*_{\text{pred}}, x_{\min}; \rho)$;
if (iter $\geq$ iter$_{mid}$) **then**
Reduce $\mathcal{D}$ by γ such that $x^* \in \mathcal{D}$, where $x^* = \arg\min\{f(x^*_{\text{pred}}), f(x_{\min})\}$;
iter = iter+1;
end do

5.2 Results

We now consider the application of Algorithm 5.1 to three problems in protein docking. The first two protein-protein complexes come from the CAPRI (Critical Assessment of Predicted Interactions; *http://capri.ebi.ac.uk*) blind docking trial [7]. We present results from CAPRI Targets 5 and 6 involving antibody/large-antigen complexes. The third complex involves an extracellular protein found in *bacillus amyloliquefaciens* called barnase, whose potentially lethal functions within the cell are inhibited by a polypeptide called barstar.

Examples 1 and 2: *Antibody/large-antigen complexes.* These targets call for the docking of an antibody to porcine α-amylase, a large antigen present in pancreatic secretion. These examples involve different camelid antibody heavy chain variable domains AMB7 and AMD9.

Example 3: *Barnase-barstar complex.* The barnase-barstar complex is a complex between the bacterial ribonuclease, barnase, and a Cys→Ala(40,82) double mutant of its intracellular polypeptide inhibitor, barstar. Barstar inhibits barnase by sterically blocking the active site with an α-helix and the loop segment connecting it to the adjacent helix. Thus, there is a high degree of complementarity of shape between the two structures. Also, the electrostatic interactions are stabilized at the surface interface. Thus, there is a high degree of charge complementarity as well.

The results for these examples are presented in Tables 5.1, 5.2, and 5.3. Each example was run three times, with different random points chosen in the reduced rectangular domain for each of the three runs. For each run, the same $m = 112$ initial points were used to start the process, and an equal number of points were generated at each subsequent iteration. The results for each of the three runs are given in the three columns of each table. In each column the following values are given:

$f(x^*_{\text{init}})$	Minimum function value obtained from adaptive grid calculation.
$f(x_{\min})$	Function value at final best known local minimum $x_{\min}$.
Δf_{final}	$f(x^*_{\text{pred}}) - f(x_{\min})$, where x^*_{pred} is the local minimum corresponding to the final predicted global minimum point. The value Δf_{final} is a measure of the accuracy of prediction of the global minimum by the convex quadratic approximation.
`Iter`	Number of major iterations required by the algorithm to terminate.
`Error(Init.)`	Initial L^1 approximation error between approximation and all local minima in $\mathcal{D}$.
`Error(Final)`	Final value of this error.

`Domain(Init.)`	Value of the domain size $\mathcal{D}_s = \sum_i \big((x_U)_i - (x_L)_i\big)$ for initial hyperrectangle $\mathcal{D}$.
`Domain(Final)`	$\mathcal{D}_s$ for final hyperrectangle.
`Interp. Pts.`	Number of local minima interpolated by final approximation.
$\lambda_{\min}$	Minimum eigenvalue of Hessian for final convex quadratic approximation.
$\text{cond}(H)$	Condition number of Hessian for final approximation.

	1	2	3
$f(x^*_{\text{init}})$	-44.7170	-44.7170	-44.7170
$f(x_{\min})$	-45.1937	-45.4878	-45.2417
Δf_{final}	0.2205	0.0000	0.1070
`Iter`	27	25	28
`Error(Init.)`	1894.20	1894.20	1894.20
`Error(Final)`	5.51	2.94	6.34
`Domain(Init.)`	252.5601	252.5601	252.5601
`Domain(Final)`	0.0201	0.0409	0.1013
`Interp. pts.`	10	15	15
$\lambda_{\min}$	1.6964	0.1028	0.1001
$\text{cond}(H)$	1.2×10^2	4.4×10^6	2.0×10^7

Table 5.1: Results for Example 1: α-amylase-AMB7

	1	2	3
$f(x^*_{\text{init}})$	-37.6290	-37.6290	-37.6290
$f(x_{\min})$	-64.8419	-64.3779	-64.3780
Δf_{final}	0.1139	0.0986	0.0008
`Iter`	9	23	16
`Error(Init.)`	1969.16	1969.16	1969.16
`Error(Final)`	2.21	14.06	11.66
`Domain(Init.)`	274.4603	274.4603	274.4603
`Domain(Final)`	0.0066	0.0135	0.0070
`Interp. pts.`	16	16	12
$\lambda_{\min}$	0.1182	0.1000	0.1000
$\text{cond}(H)$	1.6×10^5	1.4×10^9	2.7×10^6

Table 5.2: Results for Example 2: α-amylase-AMD9

	1	2	3
$f(x^*_{\text{init}})$	-35.5870	-35.5870	-35.5870
$f(x_{\min})$	-52.4205	-51.9627	-52.7510
Δf_{final}	0.0010	1.1293	0.5970
`Iter`	9	10	11
`Error(Init.)`	669.46	669.46	669.46
`Error(Final)`	9.92	95.74	11.92
`Domain(Init.)`	184.2256	184.2256	184.2256
`Domain(Final)`	0.0051	0.0165	0.0118
`Interp. pts.`	11	10	11
$\lambda_{\min}$	0.1000	0.1000	0.1000
$\text{cond}(H)$	3.0×10^3	3.8×10^4	1.2×10^3

Table 5.3: Results for Example 3: Barnase-barstar

5.3 Analysis of Results

The three examples for which results are presented show that the combination of the global optimization algorithm with DoME finds an approximate global minimum of the docking potential energy function. While the adaptive grid calculation used in DoME finds reasonably good minimum energy docking positions, significantly reduced energy values are computed by Algorithm 5.1, as shown by the difference between $f(x^*_{\text{init}})$ and $f(x_{\min})$. For the examples shown, the average decrease

$$\frac{f(x^*_{\text{init}}) - f(x_{\min})}{|f(x_{\min})|}$$

is 1.3%, 41% and 31.8% for Examples 1, 2 and 3. These results are typical of all the cases computed. For Examples 2 and 3 where a substantial decrease between $f(x^*_{\text{init}})$ and $f(x_{\min})$ is observed, the global minimum was found in the latter major iterations while the global minimum for Example 1 was obtained in the early major iterations, which suggests that the initial DoME scanning was successful in detecting local minima near the global minimum for Example 1.

The times for each run of the three examples range between 5 and 17 hours. The GCQA underestimation takes up approximately 6% of the run time, with the rest going into generating the initial points and the local optimization. Thus the additional computational time incurred by GCQA is marginal. On average, 500 function evaluations are needed to generate the initial $m = 112$ points for each major iteration. Each of the 112 generated points are used as initial points for the Hooke-Jeeves local optimization, and approximately 100 function evaluations are needed for each local optimization. Thus, for a run of 20 major iterations, the total number of function evaluations is comparable to that of the initial DoME scanning (2.0×10^6), indicating that for roughly the same amount of work, our coupled optimization approach is more effective in determining points of low energy values than by scanning or optimization alone (see [9]).

The convergence of Algorithm 5.1 to the approximate global minimum is shown by the decrease in three quantities. The first of these quantities, Δf_{final} is a measure of how accurately the final quadratic underestimator approximates the presumed global minimum of the energy function $f(x)$. In the examples shown, the average differences are 0.11, 0.03, and 0.57 for Examples 1, 2 and 3. These small values mean that the convex underestimator is an excellent approximation to the true local landscape of $f(x)$ at the approximate global minimum $x_{\min}$. The nature of this local landscape is characterized by the eigenvalues of the Hessian of the final approximation. The minimum eigenvalue $\lambda_{\min}$ represents the minimum curvature of the convex surface, and the condition number gives the ratio of the maximum to minimum curvature of this surface. The corresponding eigenvectors (not shown) also give useful information on the linear combination of the original coordinates for which the function value changes least rapidly, and most rapidly.

The second quantity showing convergence is the decrease in the approximation error between the first major iteration and the final iteration. This decrease shows that at the final iteration the approximation error has decreased substantially from its initial value, and in fact, that the final approximation actually interpolated a significant number of the local minimum. Finally, the large domain size reduction shown in the Tables means that at termination the local minimum landscape is well-represented by the final convex approximation.

The energy function $f(x)$ being minimized is a complicated function of the six coordinates, with thousands of local minima. There is no guarantee that Algorithm 5.1 actually determines the global minimum of $f(x)$. For each of the three examples used, the global minimum of the corresponding energy function is not known in advance, so we cannot be certain the $f(x_{\min})$ found in each case is in fact the desired global minimum. However, convincing evidence that $f(x_{\min})$ is at least very close to the true global minimum is given

by the fact that each of the three runs for each example gave close to the same value. In addition, the coordinates at the minima computed from the three runs were only slightly different. The protein configurations corresponding to these solutions were nearly indistinguishable. Except for the initial local minima used to start the first iteration, each of the three runs were completely different, but still found essentially the same global minimum.

6 Conclusions

A global optimization algorithm has been successfully used to find an approximate global minimum of the potential energy function generated by the DoME code for each of three examples of docking problems, showing that, with a correct energy function, it could be used to predict a correct docking configuration for a protein-ligand molecular pair. The typical protein-ligand docking energy function contains adjustable parameters which determine the location of its global minimum. We therefore anticipate that improved parameter values can be determined by adjusting them so as to minimize the difference between the global minimum coordinates of the docking energy function and the correct docking coordinates, for examples where these are known. This improved docking energy function can then be used to predict the docking configuration for protein-ligand pairs where the correct docking configuration is not yet known.

Acknowledgments

We thank Susan Lindsey for her contributions to DoME. The research of the authors was supported in part by DOE grant DE-FG03-01ER25497. J. B. Rosen was also supported in part by NSF/ITR grant #0082146.

References

1. N. Baker, M. Holst, and F. Wang, *Adaptive multilevel finite element solution of the Poisson-Boltzmann equation II: Refinement at solvent accessible surfaces in biomolecular systems*, J. Comput. Chem., 21, pp. 1343–1352, 2000.
2. K. A. Dill, A. T. Phillips, and J. B. Rosen, *CGU: an algorithm for molecular structure prediction*, in Large-scale optimization with applications, Part III (Minneapolis, MN, 1995), vol. 94 of IMA Vol. Math. Appl., Springer, New York, pp. 1–21, 1997.
3. P. E. Gill, W. Murray, M. A. Saunders, and M. H. Wright, *User's guide for NPSOL (version 4.0): A Fortran package for nonlinear programming*, Tech. Rep. SOL-86-2, Systems Optimization Laboratory, Stanford University, Stanford, CA, 1986.
4. D. S. Goodsell and A. J. Olson, *Automated docking of substrates to proteins by simulated annealing*, Proteins: Struct. Fun. Gen., 8, pp. 195–202, 1990.
5. M. Holst, N. Baker, and F. Wang, *Adaptive multilevel finite element solution of the Poisson-Boltzmann equation I: Algorithms and examples*, J. Comput. Chem., 21, pp. 1319–1342, 2000.
6. R. Hooke and T. A. Jeeves, *"Direct search" solution of numerical and statistical problems*, J. Assoc. Comput. Mach., 8, pp. 212–229, 1961.
7. J. Janin, *Welcome to CAPRI: A critical assessment of predicted interactions*, Proteins: Struct. Func. Genet., 47(3), p. 257, 2002.
8. E. C. Meng, B. K. Shoichet, and I. D. Kuntz, *Automated docking with grid-based energy evaluation*, J. Comp. Chem., 13, pp. 505–524, 1992.
9. J. C. Mitchell, J. B. Rosen, A. T. Phillips, and L. F. Ten Eyck, *Coupled optimization in protein docking*, in Proceedings of the third annual international conference on Computational molecular biology, ACM Press, 1999, pp. 280–284.

10. J. B. Rosen, K. A. Dill, and A. T. Phillips, *Protein structure and energy landscape dependence on sequence using a continuous energy function*, J. Comp. Biol., 4, pp. 227–239, 1997.

11. J. B. Rosen and R. F. Marcia, *Convex quadratic approximation*, Comput. Optim. Appl., 28, pp. 173–184, 2004.

12. J. B. Rosen, H. Park, J. Glick, and L. Zhang, *Accurate solution to overdetermined linear equations with errors using L_1 norm minimization*, Comput. Optim. Appl., 17, pp. 329–341, 2000.

Footnotes for Marcia, Mitchell, and Rosen paper:

[1] Department of Biochemistry, University of Wisconsin-Madison, Madison, WI
[2] Departments of Biochemistry and Mathematics, University of Wisconsin-Madison, Madison, WI
[3] Department of Computer Science and Engineering, University of California, San Diego, La Jolla, CA