Proceedings of the
44th IEEE Conference on Decision and Control, and
the European Control Conference 2005
Seville, Spain, December 12-15, 2005

A Semi-Definite programming-based Underestimation method for global optimization in molecular docking*

Ioannis Ch. Paschalidis†, *Member, IEEE*, Yang Shen‡, Sandor Vajda§, Pirooz Vakili¶, *Member, IEEE*

***Abstract*— The paper introduces a new global optimization method that is targeted to solve molecular docking problems, an important class of problems in computational biology. The search method is based on finding general convex quadratic underestimators to the binding energy function that is funnel-like. Finding the optimum underestimator requires solving a semi-definite programming problem, hence the name Semi-Definite programming based Underestimation (SDU). The optimal underestimator is used to bias sampling in the search region. A detailed comparison of SDU with a related method of Convex Global Underestimator (CGU), a discussion of the convergence properties of SDU, and computational results of the application of SDU to a number of rigid protein-protein docking problems are provided.**

***Index Terms*— Computational biology, Global optimization, Semi-definite programming, Molecular docking.**

I. Introduction

THE solution of a number of important problems in computational biology rests on finding global minima of energy functions that are funnel-like. These are functions with multiple non-convex funnels and a huge number of local minima of less depth that are spread over the domain of the function. For example, protein folding is the problem of predicting the 3-dimensional native structure (or "conformation") of proteins from their 1-dimensional amino acid sequences. It is known that proteins when they fold can follow multiple paths on the energy landscape [1] which is funnel-like shaped. Similar energy funnels are also found in other problems such as protein-protein docking [2].

Global optimization methods such as simulated annealing and genetic algorithms have been applied in some of these areas but they are very slow and easily trapped in kinetic moves. A number of recent approaches have attempted, with some success, to use the funnel-like shape to guide the global search to the vicinity of the global minimum. For example, the Semi-Global Simplex (SGS) algorithm uses simplex moves on surfaces spanned by the local minima rather than on the free energy itself [3]. Or, the SmoothDock approach [4] uses the strategy of descending on the "smooth" components of the energy function to which one slowly adds higher frequency components. Of most relevance to this paper is the Convex Global Underestimator (CGU) method where convex quadratic underestimators are used to approximate the envelope spanned by the local minima of the energy function [5]. The vicinity of the minimizer of the underestimator is viewed as the potential location of the global minimum of the energy function. The problem of finding the optimal underestimator is formulated and solved as a Linear Programming (LP) problem.

It has been shown that CGU does not perform well in some cases and that its performance deteriorates as the dimension of the search space increases [3]. We contend that a critical reason for this poor performance is the restricted class of underestimators used in CGU. This restriction amounts to a lack of flexibility in capturing the overall shape of the energy funnels and hence an inability to locate promising regions to search for the global minimum.

We use the same strategy of using quadratic convex functions to underestimate the envelope spanned by the local minima of the energy function. However, we consider the class of general convex quadratic functions for underestimation. In this case, given a finite set of local minima, finding the optimal underestimator amounts to solving a Semi-Definite Programming (SDP) problem, hence the term Semi-Definite programming-based Underestimation (SDU). We show, theoretically as well as experimentally, that SDU outperforms CGU, often significantly. Using some preliminary experimental results, we show that SDU is a promising method for solving molecular docking problems.

The rest of the paper is organized as follows. Sec. II presents some background material on molecular docking. Sec. III presents our SDU method. Comparisons with CGU are in Sec. IV. SDU's convergence properties are discussed in Sec. V. Some results on docking proteins are discussed in Sec. VI. We end with conclusions in Sec. VII.

II. Preliminaries

Next we review key properties of the free energy functions that are to be minimized in molecular docking problems. We start with their biophysical properties and then abstract characteristic mathematical properties that are important in the development of appropriate optimization strategies.

* Research partially supported by the NSF under a CAREER award ANI-9983221 and grants DMI-0330171, ECS-0426453, CNS-0435312, DMI-0300359, and EEC-0088073, and by the ARO under the ODDR&E MURI2001 Program Grant DAAD19-01-1-0465 to the Center for Networked Communicating Control Systems.

† Center for Information & Systems Eng., and Dept. of Manufacturing Eng., Boston University, 15 St. Mary's St., Brookline, MA 02446, e-mail: yannisp@bu.edu, url: http://ionia.bu.edu/.

‡ Center for Information & Systems Engineering, and Dept. of Manufacturing Eng., Boston University, e-mail: yangshen@bu.edu.

§ Department of Biomedical Engineering, Boston University, e-mail: vajda@bu.edu.

¶ Center for Information & Systems Engineering, and Dept. of Manufacturing Eng., Boston University, e-mail: vakili@bu.edu.

0-7803-9568-9/05/$20.00 ©2005 IEEE

A. Biophysical Origin

Free energy evaluation models: At fixed temperature and pressure, a complex of two molecules adopts the conformation that corresponds to the lowest Gibbs free energy of the system that includes the component molecules and the solvent – usually water – surrounding them. Thus, in docking calculations the natural target function to minimize is an approximation of the Gibbs free energy, G^{RL}, of the receptor-ligand complex, or that of the binding free energy, ΔG [6]. In particular, $\Delta G = G^{RL} - G^R - G^L$, where G^R and G^L are the free energies of the (free) receptor and ligand, respectively, and both G^R and G^L are independent of the conformation of the complex; hence, minimizing G^{RL} is equivalent to minimizing ΔG.

We use free energy evaluation models that combine molecular mechanics with continuum electrostatics and empirical solvation terms. In the most general case the binding free energy is decomposed according to

$$\Delta G = \Delta E_{elec} + \Delta E_{vdw} + \Delta E_{int} + \Delta G^*_{des} - T\Delta S_{sc} + \Delta G_o, \quad (1)$$

where ΔE_{elec} is the change in electrostatic energy upon binding, ΔG^*_{des} is the desolvation free energy, ΔE_{vdw} is the change in van der Waals energy, and ΔE_{int} is the change in internal energy due to any flexing/straining of the backbone and side chains. The entropy term, $-T\Delta S_{sc}$, accounts for the decrease in entropy experienced by the interface side chains upon binding. The term, ΔG_o, accounts for all other changes in the binding free energy that occur upon binding, which is considered to depend weakly on the conformation and will be treated as a constant (15 kcal/mol in this work). The internal (bonded) energy, ΔE_{int}, is the sum of bond stretching, angle bending, torsional, and improper terms.

B. Mathematical Properties

Multi-frequency behavior: The free energy function can be regarded as the sum of three components with different frequencies. First, the sum of electrostatic, desolvation, and entropic terms changes relatively slowly along any reaction path, and hence we define the "smooth" free energy, or the smooth component of the free energy by

$$\Delta G_s = \Delta E_{elec} + \Delta G_{des} - T\Delta S_{sc} + \Delta G_o \quad (2)$$

where the desolvation free energy ΔG_{des} does not include the solvent-solute van der Waals term. ΔG_s is much less sensitive to structural perturbations than the terms ΔE_{vdw} and ΔE_{int}. The internal energy ΔE_{int} changes with an intermediate frequency, and the frequency of change is very high for ΔG_{vdw}.

In local minima in which the internal and van der Waals terms are close to zero, the free energy surface is essentially determined by the "smooth" free energy component ΔG_s. However, an arbitrary pathway in the conformational space goes through non-native states at which the ΔE_{vdw} and ΔE_{int} are high, resulting in the funnel-like shape shown in Fig. 1.

III. SDU: The Semi-Definite Underestimator Method

In this section we introduce the SDU method. We first introduce some notational conventions we will be using.

Notational Conventions: All vectors are assumed to be column vectors. We use lower case boldface letters to denote vectors and for economy of space we write $\mathbf{x} = (x_1, \ldots, x_n)$ for the column vector $\mathbf{x}$. $\mathbf{x}'$ denotes the transpose of $\mathbf{x}$, $\mathbf{0}$ the vector of all zeroes, $\mathbf{e}$ the vector of all ones, and $\mathbf{e}_i$ the ith unit vector. For any vector $\mathbf{x}$ we write $\|x\|_1$ for the L1 norm, i.e., $\|x\|_1 = \sum_{i=1}^n |x_i|$, and $\|x\|$ for the Euclidean norm. We use upper case boldface letters to denote matrices. Specifically, we write $\mathbf{A} = (A_{i,j})_{i,j=1}^n$ for the matrix with $(i,j)th$ element equal to $A_{i,j}$. We denote by $\text{diag}(\mathbf{x})$ the diagonal matrix with elements $x_1, \ldots, x_n$ in the main diagonal and zeroes elsewhere. We also denote by $\text{diag}(\mathbf{A}, \mathbf{B})$ the block diagonal matrix with matrices $\mathbf{A}$ and $\mathbf{B}$ in the main diagonal and zeroes elsewhere. We define

$$\mathbf{F} \bullet \mathbf{Y} \stackrel{\triangle}{=} \sum_{i=1}^n \sum_{j=1}^n F_{i,j} Y_{i,j}. \quad (3)$$

Finally, for any event $\mathscr{S}$ we use $1\{\mathscr{S}\}$ to denote the indicator function of this event, that is, $1\{\mathscr{S}\}$ equals one when the event occurs and zero otherwise.

We are now prepared to describe the two key components of the SDU algorithm.

A. Constructing an Underestimator

Let us denote by $f : \mathbb{R}^n \to \mathbb{R}$ the function we seek to minimize and assume we have obtained a set of K local minima $\boldsymbol{\phi}^1, \ldots, \boldsymbol{\phi}^K$ of $f(\cdot)$. Let the underestimator be defined by,

$$U(\boldsymbol{\phi}) \stackrel{\triangle}{=} \boldsymbol{\phi}'\mathbf{Q}\boldsymbol{\phi} + \mathbf{b}'\boldsymbol{\phi} + c, \quad (4)$$

where $\mathbf{Q} \in \mathbb{R}^{n \times n}$ is a positive semi-definite matrix, $\mathbf{b} \in \mathbb{R}^n$, and c is a scalar. The positive semi-definiteness of $\mathbf{Q}$ guarantees the convexity of $U(\cdot)$.

Using an L1 norm as a distance metric the problem of finding the tightest possible such underestimator $U(\cdot)$ can be formulated as follows:

$$\begin{array}{ll} \min & \sum_{j=1}^K \left(f(\boldsymbol{\phi}^j) - c - \boldsymbol{\phi}^{j\prime}\mathbf{Q}\boldsymbol{\phi}^j - \mathbf{b}'\boldsymbol{\phi}^j\right) \\ \text{s.t.} & f(\boldsymbol{\phi}^j) \geq c + \boldsymbol{\phi}^{j\prime}\mathbf{Q}\boldsymbol{\phi}^j + \mathbf{b}'\boldsymbol{\phi}^j,\ j = 1, \ldots, K, \\ & \mathbf{Q} \succeq 0, \end{array} \quad (5)$$

where the decision variables are $\mathbf{Q}$, $\mathbf{b}$, and c, and "$\succeq 0$" denotes positive semi-definiteness.

Let vectors $\mathbf{b}^+, \mathbf{b}^- \geq \mathbf{0}$ and scalars $c^+, c^- \geq 0$ satisfying $\mathbf{b} = \mathbf{b}^+ - \mathbf{b}^-$ and $c = c^+ - c^-$. Let $\mathbf{s} = (s_1, \ldots, s_K)$ and $\mathbf{Y}$ be the block diagonal matrix given by

$$\mathbf{Y} \stackrel{\triangle}{=} \text{diag}(\mathbf{Q}, \text{diag}(\mathbf{b}^+), \text{diag}(\mathbf{b}^-), c^+, c^-, \text{diag}(\mathbf{s})). \quad (6)$$

Note that $\mathbf{Y} \in \mathbb{R}^{(3n+K+2)\times(3n+K+2)}$. Let $\mathbf{F}_0 \stackrel{\triangle}{=} \text{diag}(\text{diag}(\mathbf{0}), -\text{diag}(\mathbf{e}))$, where $\mathbf{0}$ is the $(3n+2)$-dimensional zero vector, and $\mathbf{e}$ is the K-dimensional vector of ones. Also, for $j = 1, \ldots, K$ we define

$$\mathbf{F}_j \stackrel{\triangle}{=} \text{diag}(\boldsymbol{\phi}^j\boldsymbol{\phi}^{j\prime}, \text{diag}(\boldsymbol{\phi}^j), -\text{diag}(\boldsymbol{\phi}^j), 1, -1, \text{diag}(\mathbf{e}_j)).$$

In addition, let $\mathbf{E}_{i,j}$ denote the $(3n+K+2) \times (3n+K+2)$ matrix with all elements equal to zero except the (i,j)th

element which equals 1. Then, (5) can be written as follows:

$$\begin{aligned} \textbf{(SDP-P)} \quad \max \quad & \mathbf{F}_0 \bullet \mathbf{Y} \\ \text{s.t.} \quad & \mathbf{F}_j \bullet \mathbf{Y} = f(\boldsymbol{\phi}^j),\ j = 1, \ldots, K, \\ & \mathbf{E}_{i,j} \bullet \mathbf{Y} = 0,\ j = 1, \ldots, i-1, \\ & \qquad i = n+1, \ldots, 3n+K+2, \\ & \mathbf{Y} \succeq 0, \end{aligned} \tag{7}$$

where the decision variable is the matrix $\mathbf{Y}$. Problem (**SDP-P**) in (7) is a *Semi-Definite Programming (SDP)* problem [7]. SDP problems can be solved efficiently using interior-point methods [7] (in polynomial time).

The dual to (**SDP-P**) in (7) is the problem

$$\begin{aligned} \textbf{(LMI-D)} \quad \min \quad & \sum_{j=1}^{K} x_j f(\boldsymbol{\phi}^j) \\ \text{s.t.} \quad & \mathbf{Z} = \sum_{j=1}^{K} x_j \mathbf{F}_j \\ & + \sum_{i=n+1}^{3n+K+2} \sum_{j=1}^{i-1} w_{i,j} \mathbf{E}_{i,j} - \mathbf{F}_0, \\ & \mathbf{Z} \succeq 0, \end{aligned} \tag{8}$$

where the decision variables are x_j's and $w_{i,j}$'s. Problem (**LMI-D**) can be seen as the problem of minimizing a linear function subject to the *Linear Matrix Inequality (LMI)*

$$\sum_{j=1}^{K} x_j \mathbf{F}_j + \sum_{i=n+1}^{3n+K+2} \sum_{j=1}^{i-1} w_{i,j} \mathbf{E}_{i,j} - \mathbf{F}_0 \succeq 0.$$

Our main result on underestimating a set of local minima is summarized in the following theorem.

Theorem III.1 *Consider a function $f : \mathbb{R}^n \to \mathbb{R}$ and a set of local minima $\boldsymbol{\phi}^1, \ldots, \boldsymbol{\phi}^K$ of $f(\cdot)$. Let $(\mathbf{Q}, \mathbf{b}^+, \mathbf{b}^-, c^+, c^-, \mathbf{s})$ form an optimal solution $\mathbf{Y}$ of Problem (**SDP-P**) in (7), where $\mathbf{Y}$ is defined as in (6). Let $U(\boldsymbol{\phi}) \stackrel{\triangle}{=} \boldsymbol{\phi}'\mathbf{Q}\boldsymbol{\phi} + (\mathbf{b}^+ - \mathbf{b}^-)'\boldsymbol{\phi} + (c^+ - c^-)$. Then $U(\cdot)$ satisfies $f(\boldsymbol{\phi}^j) \geq U(\boldsymbol{\phi}^j)$ for all $j = 1, \ldots, K$ while minimizing $\|(f(\boldsymbol{\phi}^1), \ldots, f(\boldsymbol{\phi}^K)) - (U(\boldsymbol{\phi}^1), \ldots, U(\boldsymbol{\phi}^K))\|_1$. Moreover, the dual to Problem (**SDP-P**) is the LMI problem (**LMI-D**) in (8).*

Hereafter, we will say that a function $U(\cdot)$ satisfying the statement of Theorem III.1 *underestimates* $f(\cdot)$ at points $\boldsymbol{\phi}^1, \ldots, \boldsymbol{\phi}^K$. Figure 1 illustrates such an underestimator.

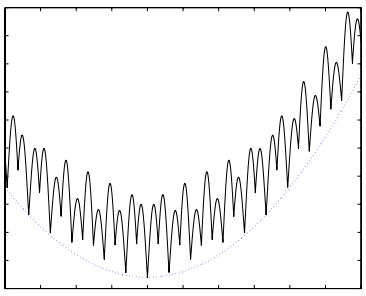


Fig. 1. A funnel-like shaped function and a quadratic function underestimating the surface spanned by the local minima.

B. Sampling

Suppose we are seeking the native conformation in some region $\mathscr{B}$. Using a set of local minima $\boldsymbol{\phi}^1, \ldots, \boldsymbol{\phi}^K \in \mathscr{B}$ of $f(\cdot)$ we construct an underestimator $U(\cdot)$ as described in Section III-A. Let $\boldsymbol{\phi}^P$ the minimizer of $U(\cdot)$. Notice that the underestimator contains information on the location of the near-native energy valley. We are interested in sampling conformations such that conformations close to $\boldsymbol{\phi}^P$ are more likely to be selected. In addition, conformations with high enough energy can be ignored. This can be achieved by using the following probability density function (pdf) in $\mathscr{B}$:

$$g(\boldsymbol{\phi}) = \frac{U(\boldsymbol{\phi}) - U_{max}}{\int_{\mathscr{B}} (U(\boldsymbol{\phi}) - U_{max})\, d\boldsymbol{\phi}} \stackrel{\triangle}{=} \frac{U(\boldsymbol{\phi}) - U_{max}}{A}. \tag{9}$$

In the expression above $U_{max} = \max_{\mathscr{B}} U(\boldsymbol{\phi})$ and we introduced the normalizing constant A to denote the integral in the denominator.

To generate random samples in $\mathscr{B}$ using the above pdf we will use the so called *rejection method*. In particular let $h(\phi) = 1/V$ be the uniform pdf in $\mathscr{B}$ where $V = \int_{\mathscr{B}} d\boldsymbol{\phi}$ is the volume of $\mathscr{B}$. Notice that

$$g(\boldsymbol{\phi}) \leq \frac{V(U(\boldsymbol{\phi}^P) - U_{max})}{A} h(\phi), \qquad \forall \boldsymbol{\phi} \in \mathscr{B}$$

and set $R(\boldsymbol{\phi})$ equal to the ratio of the left hand side over the right hand side of the above, that is,

$$R(\boldsymbol{\phi}) \stackrel{\triangle}{=} \frac{U(\boldsymbol{\phi}) - U_{max}}{U(\boldsymbol{\phi}^P) - U_{max}}. \tag{10}$$

In order to discard high energy conformations we are interested in sampling points in $\mathscr{B}$ with associated probability density in some interval $[\zeta, 1]$, where $\zeta \in [0, 1)$. The algorithm in Fig. 2 outputs such a sample point. To see that notice that in Step 1 we generate uniformly distributed samples in the set $\{(\mathbf{x}, y) \mid \mathbf{x} \in \mathscr{B}, y \in [\zeta, 1]\}$. The rejection rule of Step 2 accepts samples that are uniformly distributed in $\{(\mathbf{x}, y) \mid \mathbf{x} \in \mathscr{B}, \zeta \leq y \leq g(\mathbf{x})\}$. Thus, the output $\boldsymbol{\phi}$ of the algorithm is distributed in $\mathscr{B}$ according to $g(\cdot)$.

1) Generate a uniformly distributed random variables $\mathbf{x}_1 \in \mathscr{B}$ and $\mathbf{x}_2 \in [\zeta, 1]$.
2) If $\mathbf{x}_2 \leq R(\mathbf{x}_1)$, stop and output $\boldsymbol{\phi} = \mathbf{x}_1$; otherwise, return to Step 1.

Fig. 2. An algorithm generating a sample in $\mathscr{B}$ drawn from $g(\cdot)$ with associated density in $[\zeta, 1]$.

We finally note that the algorithm in Fig. 2 requires knowing $U_{\max}$. In many cases this is straightforward to obtain. Consider for instance the case where $\mathscr{B}$ is a polyhedron. Then, since $U(\cdot)$ is convex it achieves its maximum at an extreme point of the polyhedron $\mathscr{B}$. Hence, it suffices to search over all extreme points which in low-dimensional problems (e.g., rigid docking) are not that many. Alternatively, one can use an estimate of $U_{\max}$, e.g., $\max_i U(\boldsymbol{\phi}^i)$.

C. The SDU Algorithm

We now have all the ingredients to present our SDU algorithm. The algorithm seeks a global minimum of the free energy function $f(\cdot)$ in some region $\mathscr{B}$ of the conformational space; it is presented in Figure 3. Throughout the algorithm we maintain a set $\mathscr{L}$ of interesting local minima obtained so far as well as the best such local minimum denoted by

1) **Initialization**: Starting from K ($K \geq 2n+1$) random points in $\mathscr{B}$ perform local minimization to obtain K local minima $\boldsymbol{\phi}^1, \ldots, \boldsymbol{\phi}^K$ of $f(\cdot)$. Set $\mathscr{L} = \{\boldsymbol{\phi}^1, \ldots, \boldsymbol{\phi}^K\}$ and $\boldsymbol{\phi}^G = \arg\min_{i=1,\ldots,n} f(\boldsymbol{\phi}^i)$.
2) **Underestimation**: Solve Problem (**SDP-P**) in (7) to obtain the underestimator $U(\boldsymbol{\phi})$. Set the predictive conformation equal to a minimizer of $U(\cdot)$, that is, when $\mathbf{Q}$ is invertible $\boldsymbol{\phi}^P = -\frac{1}{2}\mathbf{Q}^{-1}\mathbf{b}$.
3) **Elimination**: Discard unfavorable local minima from $\mathscr{L}$; more specifically, set $\mathscr{L} := \mathscr{L} \backslash \{\boldsymbol{\phi} \in \mathscr{L} \mid R(\boldsymbol{\phi}) < \zeta \text{ and } \boldsymbol{\phi} \neq \boldsymbol{\phi}^G\}$.
4) **Focalization**: Define a neighborhood $\mathscr{N}(\boldsymbol{\phi}^P) \subseteq \mathscr{B}$ of $\boldsymbol{\phi}^P$. Set $\mathscr{B} := \mathscr{N}(\boldsymbol{\phi}^P)$.
5) **Exploration**:
   a) Start from $\boldsymbol{\phi}^P$ and use local minimization to obtain a local minimum $\hat{\boldsymbol{\phi}}^P$ of $f(\cdot)$. If $\hat{\boldsymbol{\phi}}^P \in \mathscr{B}$ set $\mathscr{L} := \mathscr{L} \cup \{\hat{\boldsymbol{\phi}}^P\}$ and $\boldsymbol{\phi}^G := \arg\min\{f(\boldsymbol{\phi}^G), f(\hat{\boldsymbol{\phi}}^P)\}$.
   b) Obtain m samples from the sampling algorithm of Fig. 2. Using these samples as starting points perform local minimization to obtain m local minima $\mathbf{x}^1, \ldots, \mathbf{x}^m$ of $f(\cdot)$. Set $\mathscr{L} := \mathscr{L} \cup \{\mathbf{x} \mid \mathbf{x} = \mathbf{x}^1, \ldots, \mathbf{x}^m, \mathbf{x} \in \mathscr{B}\}$ and
$$\boldsymbol{\phi}^G := \arg \min_{\substack{\boldsymbol{\phi}=\boldsymbol{\phi}^G, \mathbf{x}^1, \ldots, \mathbf{x}^m \\ \boldsymbol{\phi} \in \mathscr{B}}} f(\boldsymbol{\phi}).$$
6) **Termination**: If $\|\boldsymbol{\phi}^G - \boldsymbol{\phi}^P\| < \epsilon$ then stop; otherwise go to Step 2.

Fig. 3. The SDU algorithm.

$\boldsymbol{\phi}^G$. The evolution of the algorithm in Fig. 3 depends on the parameters $K, \zeta \in [0,1], m$ and ϵ, as well as the way we define the neighborhood $\mathscr{N}(\boldsymbol{\phi}^P)$ in Step 5. These will be appropriately tuned in every problem instance.

A couple of remarks on the proposed SDU algorithm are in order. The algorithm combines exploration with focalization in energy favorable regions of the conformational space (energy funnels). The exploration is in fact biased towards these energy favorable funnels. This is motivated by the desire to avoid computationally expensive exploration in areas of the conformational space that are not likely to contain the native structure.

We should point out that we make no claims that the SDU algorithm will converge to the global minimum of $f(\cdot)$. In fact, it is straightforward to see that it will not find the global minimum if we do not use enough local minima to determine the underestimator or when $f(\cdot)$ is arbitrary and does not have a funnel-like shape. However, later in the paper we will provide arguments that guarantee convergence for funnel-like shaped functions under a suitable set of conditions.

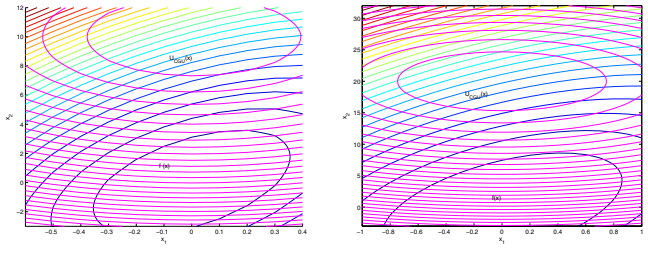
IV. CGU and and its Limitations

The CGU algorithm [5] can be viewed as a special case of the SDU algorithm under the following key modifications:

1) **Underestimation**. In deriving the underestimator $U(\cdot)$ impose the constraint that the matrix $\mathbf{Q}$ is a diagonal positive semi-definite matrix. Then the semi-definite constraint can be replaced by a non-negativity constraint for all diagonal entries. It follows that Problem (**SDP-P**) can be reformulated as a *linear programming problem (LP)* which can be easily solved.
2) **Sampling**. Replace our biased sampling method with random (uniform) sampling in the neighborhood $\mathscr{N}(\boldsymbol{\phi}^P) \subseteq \mathscr{B}$ of $\boldsymbol{\phi}^P$.

We will argue that these two differences between CGU and SDU drastically affect the performance of the CGU algorithm. In particular, limiting the underestimator search to the class of canonical parabolas (diagonal $\mathbf{Q}$) substantially reduces the efficiency and accuracy of CGU for general problems where the surface spanned by the local minima is not typically aligned with the canonical coordinates defining the underestimating parabola. [3] reports many such cases where CGU performs poorly. Some attempts in addressing this limitation have been made in [8] but they are only able to handle very special cases.

We start our study of CGU limitations by providing a simple example where CGU fails. Consider the function $f(\boldsymbol{\phi}) = 100\phi_1^2 - 10\phi_1\phi_2 + \phi_2^2$ whose global minimum is at the origin. We use CGU to underestimate this function. In Fig. 4 we plot contours of the function $f(\cdot)$ and its resulting CGU underestimator $U_{CGU}(\cdot)$. More specifically,



(a) (b)

Fig. 4. CGU yields different results depending on the sample region.

in Fig. 4(a) we randomly (and uniformly) sampled the region $[-1,10] \times [-1,10]$ to obtain a large set of points which we used to construct the CGU underestimator. The underestimator $U_{CGU}(\cdot)$ has a global minimum (to be referred to as *prediction*) at $(0,10)$. Notice that CGU constrains its prediction within the sampling region. In Fig. 4(b) we performed the same experiment but used $[-1,20] \times [-1,20]$ as the sampling region and CGU's prediction was $(0,20)$. In both cases, CGU's prediction is at the boundary because the minimization of $U_{CGU}(\cdot)$ is constrained within the sampling region; unconstrained minimization produces an even worse result. It is evident that the prediction heavily depends on the initial sampling region which in most cases is set arbitrarily. In the next subsection we analyze the CGU underestimating approach and compare to the one we employ in SDU.

A. Comparing the CGU and SDU underestimators

As we discussed in Section III a quadratic underestimator will not be informative if either (*i*) $f(\cdot)$ is not funnel-like and the envelope of local minima can not be well approximated by a convex quadratic, and (*ii*) if we do not use a rich enough set of local minima in constructing $U(\cdot)$. In the following we wish to remove these two potential sources of poor performance in order to better assess the underestimating power of CGU and SDU. More specifically, we consider the "ideal" case of underestimating a convex quadratic given by $f(\mathbf{t}) = \mathbf{t}'\bar{\mathbf{Q}}\mathbf{t}+\bar{\mathbf{b}}'\mathbf{t}+\bar{c}$, where $\bar{\mathbf{Q}} \succeq 0$. Further, we assume that an infinite number of sample points of $f(\cdot)$ in some compact sampling region $\mathscr{B}$ is at our disposal when we construct the underestimator. The construction of the underestimator based on utilizing all sample points in $\mathscr{B}$ can be formulated as the following (infinite dimensional) optimization problem:

$$\begin{array}{ll} \text{minimize} & \int_{\mathbf{t}\in\mathscr{B}}(f(\mathbf{t}) - U(\mathbf{t}))\ d\mathbf{t} \\ \text{subject to} & f(\mathbf{t}) \geq U(\mathbf{t}), \qquad \mathbf{t} \in \mathscr{B}, \end{array} \tag{11}$$

where the decision variables are the (yet unspecified) parameters defining $U(\mathbf{t})$.

Suppose first that we use the SDU underestimating approach and seek to construct a function $U(\mathbf{t}) = \mathbf{t}'\mathbf{Q}\mathbf{t} + \mathbf{b}'\mathbf{t} + c$, where $\mathbf{Q} \succeq 0$. Consider the problem in (11) for such a $U(\mathbf{t})$. The next proposition is immediate.

Proposition IV.1 *SDU can underestimate $f(\cdot)$ exactly, in particular, $(\mathbf{Q}, \mathbf{b}, c) = (\bar{\mathbf{Q}}, \bar{\mathbf{b}}, \bar{c})$ is an optimal solution of (11).*

We next consider the CGU underestimation approach. Specifically, we seek to construct a function $U(\mathbf{t}) = \mathbf{t}'\mathbf{D}\mathbf{t} + \mathbf{b}'\mathbf{t} + c$, where $\mathbf{D}$ is diagonal positive semi-definite matrix. Namely, $\mathbf{D} = \text{diag}(d_1, \ldots, d_n)$ where $d_i \geq 0$ for $i = 1, \ldots, n$. For simplicity of the exposition $\mathscr{B} = \mathscr{B}_1 \times \cdots \times \mathscr{B}_n$ where $\mathscr{B}_i = [l_i, u_i]$ and $u_i - l_i = T$ for all $i = 1, \ldots, n$. We denote $\mathbf{a}(\mathbf{t}) = (t_1^2, \ldots, t_n^2, t_1, \ldots, t_n, 1)$, $\mathbf{h} = (\int_{t_1\in\mathscr{B}_1} t_1^2 dt_1/T, \ldots, \int_{t_n\in\mathscr{B}_n} t_n^2 dt_n/T, \int_{t_1\in\mathscr{B}_1} t_1 dt_1/T, \ldots, \int_{t_n\in\mathscr{B}_n} t_n dt_n/T, 1)$, and $\mathbf{z} = (d_1, \ldots, d_n, b_1, \ldots, b_n, c)$.

In this case, the optimization problem in (11) is equivalent to the following problem:

$$\begin{array}{lll} \textbf{(LSIP-P)} & \max & \mathbf{h}'\mathbf{z} \\ & \text{s.t.} & \mathbf{a}'(\mathbf{t})\mathbf{z} \leq f(\mathbf{t}), \quad \mathbf{t} \in \mathscr{B}, \end{array} \tag{12}$$

where $\mathbf{z}$ is the decision vector. Note that it involves an infinite number of constraints. A problem with such a structure is known as the *Linear Semi-Infinite Programming (LSIP)* problem [9]. Its dual can be formulated in measure space as follows:

$$\begin{array}{lll} \textbf{(LSIP-D)} & \min & \int_{\mathscr{B}} f(\mathbf{t}) d\mu(\mathbf{t}) \\ & \text{s.t.} & \int_{\mathscr{B}} \mathbf{a}(\mathbf{t}) d\mu(\mathbf{t}) = \mathbf{h},\ \mu \in M^+(\mathscr{B}), \end{array} \tag{13}$$

where $M^+(\mathscr{B})$ denotes the set of non-negative regular Borel measures on $\mathscr{B}$.

It can be shown (we omit the details due to space limitations) that the underestimator obtained by solving **(LSIP-P)** in (12) is the limit (as $K \to \infty$) of the CGU underestimators derived based on function values $f(\mathbf{t}^1), \ldots, f(\mathbf{t}^K)$ at a set of samples $\mathbf{t}^1, \ldots, \mathbf{t}^K$ from $\mathscr{B}$. This is insightful because it suggests that when we use enough samples the quality of the CGU underestimator does not depend on sample selection but rather on the fundamental structure of the underestimator function. Our main result in this section is the following theorem; the proof is omitted in the interest of space.

Theorem IV.2 *Let $f(\mathbf{t}) = \mathbf{t}'\bar{\mathbf{Q}}\mathbf{t}$, where $\bar{\mathbf{Q}} \succeq 0$. Further, let $U^*(\mathbf{t}) = \mathbf{t}'\mathbf{D}^*\mathbf{t} + \mathbf{b}^{*'}\mathbf{t} + c^*$, be the optimal solution to (***LSIP-P***), i.e., the optimal CGU underestimator to $f(\mathbf{t})$. Then, in general, $\mathbf{b}^* \neq 0$. In other words the minimizer of the underestimator is different from the minimizer of $f(\mathbf{t})$.*

V. On SDU's Convergence

In this section we give the result that under appropriate conditions the SDU algorithm converges to the global minimum of the function $f(\cdot)$. Such (free energy) functions arising in molecular docking applications, as we have explained, possess key structural properties. Therefore, we will impose a set of structural assumptions on $f(\cdot)$ and the search region $\mathscr{B}$ that reflect the properties of the free energies functions we seek to minimize. We denote by $\text{epi}(f)$ the epigraph of $f(\cdot)$ which is defined as $\text{epi}(f) = \{(\boldsymbol{\phi}, w) \mid \boldsymbol{\phi} \in \mathscr{B}, f(\boldsymbol{\phi}) \leq w\}$. We also denote by $\text{conv}(\mathscr{S})$ the convex hull of any set $\mathscr{S}$.

Assumption A

*Assume that $f(\boldsymbol{\phi})$ satisfies the following set of conditions: (*i*) it is continuously differentiable; (*ii*) $f(\cdot)$ has a unique global minimum in $\mathscr{B}$; (*iii*) $\mathscr{B}$ is compact; (*iv*) for all local minima $\boldsymbol{\phi}$ of $f(\cdot)$ there exists an open set such that $\boldsymbol{\phi}$ is the unique minimizer of $f(\cdot)$ within this set; (*iv*) the extreme points of conv(epi(f)) lie on a quadratic function $\tilde{U}(\boldsymbol{\phi}) = \boldsymbol{\phi}'\tilde{\mathbf{Q}}\boldsymbol{\phi} + \tilde{\mathbf{b}}'\boldsymbol{\phi} + \tilde{c}$; (v) $\tilde{U}(\boldsymbol{\phi})$ has a unique global minimum which is identical with the global minimum of $f(\cdot)$ in $\mathscr{B}$.*

For functions that satisfy Assumption A we will say that they have a *funnel-like shape* (see Fig. 1 for an illustration). As we argued in Section II, Assumption A is not overly restrictive for the free energy functions we are interested in minimizing.

The following theorem establishes that given sufficient sampling of the search region $\mathscr{B}$ the SDU underestimation procedure can locate the global minimum of $f(\cdot)$ which we denote by $\boldsymbol{\phi}^*$. The proof is omitted again due to space limitation.

Theorem V.1 *Let Assumption A prevail. Consider the SDU algorithm provided in Fig. 3 and assume that $\mathscr{B}$ contains at least $\frac{(n+1)(n+2)}{2}$ local minima of $f(\cdot)$ which are extreme points of conv(epi(f)). Suppose that in Step 1 of the algorithm we obtain K uniformly distributed samples in $\mathscr{B}$ and for each one of those we perform local minimization to obtain K local minima $\boldsymbol{\phi}^1, \ldots, \phi^K$ of $f(\cdot)$. Then, the global minimum $\boldsymbol{\phi}^P$ of the underestimator $U(\cdot)$ obtained in Step 2 of the algorithm converges in probability to the global minimum $\boldsymbol{\phi}^*$ of $f(\cdot)$ as $K \to \infty$, namely, $\lim_{K\to\infty} \mathbf{P}[\boldsymbol{\phi}^P = \boldsymbol{\phi}^*] = 1$.*

We can think of Theorem V.1 as a simple sanity check of the SDU underestimation approach. It implies that given an appropriate funnel-like structure, SDU can locate the global minimum of the free energy function with probability approaching one as the number of random samples in $\mathscr{B}$ increases.

VI. Applications in Rigid-body Protein Docking

In this section we report results from applying SDU to a number of protein-to-protein docking instances. We assume that both proteins are rigid. This is certainly true for their backbones and, as we will see, we will allow some flexibility in determining the position of the side chains in the interface between the two proteins. More specifically, we consider 3-dimensional problems where one protein is held fixed, both proteins are optimally oriented (based on information we obtain from the known bound structure), and we seek to determine the position of the second protein in the bound structure with respect to the first one. We will constrain these three translational variables to be in a relatively short range. This means that we have already identified the active site of the first protein and we are interested in determining the exact position of the second protein in this site. The identification of the active site is usually done by other means using clustering techniques and simple FFT-based affinity procedures based on rough energy models.

The energy functional we minimize includes van der Waals interactions (ΔE_{vdW}), the desolvation energy (ΔE_{des}) and the electrostatic energy (ΔE_{elec}). We allow side-chains to be flexible during the local minimization phase; this is critical as side-chains in the interface between the two proteins can not be considered rigid and they are packed in a way that minimizes the overall free energy of the bound complex. We report results for three complexes 1BRS (barnase/barstar), 1MLC (an monoclonal antibody and lyzozyme complex), and 2PTC (a trypsin-inhibitor complex). The bound structure for each case has the ligand centered at the origin, that is, the optimal solution is $(0, 0, 0)$. The initial search region $\mathscr{B}$ is a 10Å cube. In each iteration (indexed by i), we report the best structure found so far, $\boldsymbol{\phi}^G$, the corresponding energy $f(\boldsymbol{\phi}^G)$ (in kcal/mol), and the RMSD distance $D(\boldsymbol{\phi}^G)$ from the native structure (in Å). The primary measure of success is the proximity of $D(\boldsymbol{\phi}^G)$ to zero. In the results that follow we terminated the algorithm when no progress was made in the last three iterations.

Table I reports our results for the 1MLC complex. The evolution of SDU in all three complexes considered is depicted in Fig. 5.

i	ϕ_1^G	ϕ_2^G	ϕ_3^G	$f(\boldsymbol{\phi}^G)$	$D(\boldsymbol{\phi}^G)$
1	-0.831	-0.935	1.428	-24.30	1.90
2	-0.831	-0.935	1.428	-24.30	1.90
3	0.056	-0.444	-0.550	-64.57	0.70
4	0.056	-0.444	-0.550	-64.57	0.70
5	-0.144	0.764	0.059	-69.29	0.78
6	-0.144	0.764	0.059	-69.29	0.78
7	-0.033	0.549	0.287	-71.19	0.62

TABLE I
SDU APPLIED TO THE 1MLC COMPLEX.

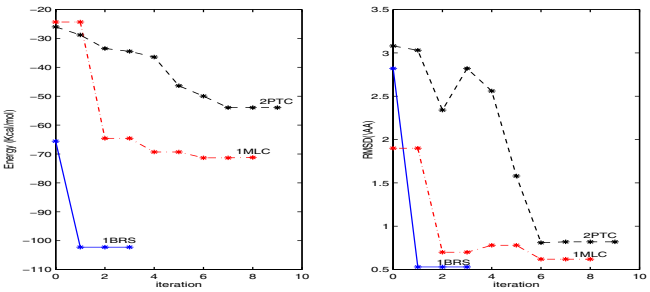


Fig. 5. The evolution of SDU for the 1BRS, 1MLC, and 2PTC complexes.

VII. Conclusions

We introduced a new method for minimizing free-energy functions appearing in molecular docking and other important problems in computational biology. These energy function are notoriously hard to minimize as they consist of terms acting in disparate space-scales and have a huge number of local minima. Yet, they exhibit a funnel-like shape which we use to our advantage.

Our method works on the surface spanned by local minima. We developed a technique based on semi-definite programming to form a general convex underestimator of the energy function. The underestimator guides random, yet biased, exploration of the energy landscape. We established, theoretically as well as numerically, that our method is superior to an alternative (CGU) method. We applied our algorithm to a number of protein-protein docking problems and showed that the resulting conformation is extremely close to the native one (within 1Å RMSD). This improves upon the 3–5Å RMSD that current methods produce.

References

[1] C. Levinthal, "Are there pathways for protein folding?" *J. Chim. Phys.*, vol. 65, pp. 44–45, 1968.

[2] C. J. Camacho, Z. Weng, S. Vajda, and C. DeLisi, "Free energy landscapes of encounter complexes in protein-protein association," *Biophys. J.*, vol. 76, pp. 1166–1178, 1999.

[3] S. Dennis and S. Vajda, "Semi-global simplex optimization and its application to determining the preferred solvation sites of proteins," *J. Comp. Chem.*, vol. 23, no. 3, pp. 319–334, 2002.

[4] C. Camacho and S. Vajda, "Protein docking along smooth association pathways," *Proc. Natl. Acad. Sci. USA*, vol. 98, pp. 10 636–10 641, 2001.

[5] A. Phillips, J. Rosen, and K. Dill, *From Local to Global Optimization (P.M. Pardalos et al., Eds.)*. Kluwer Academic Publishers, 2001, ch. Convex Global Underestimation for Molecular Structure Prediction, pp. 1–18.

[6] S. Vajda, M. Sippl, and J. Novotny, "Empirical potentials and functions for protein folding and binding," *Curr. Op. Struct. Biol.*, vol. 7, pp. 222–228, 1997.

[7] A. Ben-Tal and A. Nemirovski, *Lectures on Modern Convex Optimization*. SIAM, 2001.

[8] J. Rosen and R. Marcia, "Convex quadratic approximation," *Computational Optimization and Applications*, vol. 28, pp. 173–184, 2004.

[9] R. Hettich and K. O. Kortanek, "Semi-infi nite programming: theory, methods and applications," *SIAM Review*, vol. 35, no. 3, pp. 380–429, 1993.