SDU: A Semi-Definite programming-based Underestimation method for stochastic global optimization in protein docking*

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

Abstract— **The paper introduces a new stochastic global optimization method targeting protein-protein docking problems, an important class of problems in computational structural biology. The 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 underestimator is used to bias sampling in the search region. It is established that under appropriate conditions SDU locates the global energy minimum with probability approaching one as the sample size grows. A detailed comparison of SDU with a related method of Convex Global Underestimator (CGU), and computational results for protein-protein docking problems are provided.**

Index Terms— **Semi-definite programming, Linear matrix inequalities, optimization, structural biology, protein-protein docking.**

I. Introduction

THIS work is motivated by a fundamental and challenging problem in computational *structural biology*. Proteins interact with each other and with other chemical entities in the cell to form complexes consisting of two (or more) macromolecules. Protein-protein interactions play a central role in metabolic control, signal transduction, and gene regulation. Proteins also interact with a large number of small molecules that serve as substrates, inhibitors, or cofactors in metabolic reactions, as well as with external compounds acting as drugs in modulating biological behavior. Determining the 3-dimensional (3D) structure of a complex from the atomic coordinates of two or more interacting molecules is known as the *molecular docking problem*. Experimental techniques –x-ray crystallography or nuclear magnetic resonance (NMR)– can provide such 3D structures but are time-consuming, expensive, and not universally applicable as many molecular complexes are short-lived and exist only under well defined cellular conditions. As a result, solving these problems computationally is critical and has attracted a lot of attention.

Docking two proteins can be seen as a control problem as one has to control one of them as it approaches and docks with the other; this has parallels with problems arising in robot arm manipulation (Murray et al. [1], Gwak et al. [2]). From a static point of view, thermodynamic principles imply that proteins bind in a way that minimizes the Gibbs free energy of the complex. These energy functions have multiple deep funnels and a huge number of local minima of less depth that are spread over the domain of the function (see Fig. 1). This "complexity" of the energy landscape is perhaps best articulated in the Levinthal paradox [3]: the probability that nature could find the native 3D structures (or "conformations") by random search seems uncomfortably close to zero. More recently though, it was recognized that the funnel-like shape of the energy function may explain why proteins bind with other molecules very fast and in so doing they may follow multiple pathways on the energy surface (see Tsai et al. [4]). In this paper we introduce a new optimization method for solving docking problems. It is exactly this funnel-like shape of the energy function we set out to exploit.

In general, the energy functions include forces that act in different space-scales, resulting in multi-frequency behavior, and leading to a huge number of local minima caused by high-frequency terms. Even though some global optimization methods (see Floudas et al. [5]) have been useful in the related problem of protein folding, it appears that only simulated annealing has been consistently used in docking. Specifically, Baker and co-workers [6] employ simulated annealing on a progressively improving approximation of the energy function. This has been fairly effective but it is computationally expensive and thus applicable to relatively small problem instances. A number of other 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 (e.g., the Semi-Global Simplex (SGS) by Dennis and Vajda [7], and SmoothDock by Camacho and Vajda [8]). Of most relevance to this paper is the Convex Global Underestimator (CGU) method of Phillips et al. [9] where convex canonical quadratic underestimators are used to approximate the envelope spanned by the local minima of the energy function. 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 ([7]). We contend that a critical reason for this poor performance is the restricted class

* Research partially supported by the NSF under grants DMI-0330171, ECS-0426453, CNS-0435312,DMI-0300359, EEC 0088073, by the NIH under grant R01-GM61867, and by the ARO under the ODDR&E MURI2001 Program Grant DAAD19-01-1-0465.

† Corresponding author. Center for Information and Systems Engineering, and Department of Manufacturing Engineering, Boston University, Boston, MA 02215, e-mail: yannisp@bu.edu, url: http://ionia.bu.edu/.

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

§ Center for Information and Systems Engineering, and Department of Manufacturing Engineering, Boston University, e-mail: vakili@bu.edu

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

of underestimators used in CGU. This restriction amounts to a lack of flexibility in capturing the overall shape of the funnels and hence an inability to locate promising regions.

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). The dual to this problem involves *Linear Matrix Inequalities (LMIs)* which have found ubiquitous use in control theory lately. Another key contribution of the new method we introduce is the use of a biased exploration strategy that is guided by the underestimator. We show, theoretically as well as experimentally, that SDU outperforms CGU, often significantly. Moreover, we show that SDU will indeed (probabilistically) converge to the global minimum when applied to a general class of funnel-shaped functions as the number of local minima used for underestimating the energy function grows. This suggests that SDU has the potential to be useful in solving molecular docking problems. A number of numerical results, some involving real instances of the protein-protein docking problem, reinforce this conclusion.

The rest of the paper is organized as follows. Sec. II presents background material on docking and surveys various approaches. Sec. III presents the SDU method. Comparisons with CGU are in Sec. IV. SDU's convergence properties are discussed in Sec. V. Numerical results for a set of test functions are in Sec. VI. Some results on docking proteins are discussed in Sec. VII. We end with conclusions in Sec. VIII.

Notational Conventions: Throughout the paper 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 matrix (or vector) of all zeroes, $\mathbf{e}$ the vector of all ones, $\mathbf{I}$ the identity matrix, and $\mathbf{e}_i$ the ith unit vector. For any vector $\mathbf{x}$ we write $\|\mathbf{x}\|_1$ for the L1 norm, i.e., $\|\mathbf{x}\|_1 = \sum_{i=1}^n |x_i|$, and $\|\mathbf{x}\|$ for the Euclidean norm. We use upper case boldface letters to denote matrices and write $\mathbf{A} = (A_{i,j})_{i,j=1}^n$ for the matrix $\mathbf{A}$ with $(i,j)th$ element equal to $A_{i,j}$. We denote by diag$(\mathbf{x})$ the diagonal matrix with elements $x_1, \ldots, x_n$ in the main diagonal and zeroes elsewhere. We also denote by 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

$$\mathbf{F} \bullet \mathbf{Y} \triangleq \sum_{i=1}^{n} \sum_{j=1}^{n} F_{i,j} Y_{i,j}, \qquad (1)$$

which is the Frobenius inner product of matrices. Finally, for any event $\mathscr{S}$, $1\{\mathscr{S}\}$ denotes its indicator function.

II. Preliminaries and Background

In this section we provide some biological background on the docking problem in order to establish the appropriate context for the optimization methodology we introduce.

State-of-the-art protein-protein docking approaches start by identifying conformations with good surface/chemical complementarity where the two interacting proteins are considered *rigid* bodies. Accordingly, the problem of forming a complex by docking one protein – the ligand – to the other – the receptor – has parallels with problems arising in rigid body motion and robot arm manipulation (see, e.g., Murray et al. [1], Gwak et al. [2]). In particular, one has to control the ligand as it approaches and docks with the receptor. The ligand's relative position lies in the Euclidean group of rigid-body motion, $SE(3)$, which is the semidirect product of $\mathbb{R}^3$ (translations) and the rotation group $SO(3)$. The dynamics of the ligand's approach to the receptor play an important role in nature but are hard to model (and computationally prohibitive to simulate) in the time-scale they evolve. The problem is further complicated by the fact that proteins are not rigid bodies and experience conformational changes as they bind (mostly the side-chains on the interface). The protein-protein docking literature (e.g., [8] and references therein) is predominantly treating the docking problem as a static problem and this is the view we will initially adopt for this problem.

Protein-protein docking approaches initially screen conformations by various measures of surface complementarity which can be efficiently computed using fast Fourier correlation techniques (FFT). However, when docking unbound conformations these methods yield many "false positives" (i.e., conformations that have good score but are far from the native one). An effective approach used by Comeau et al. [10] is to cluster the conformations obtained by the FFT filtering based on how close they are. (A common metric of distance between conformations is the root mean square deviation (RMSD) of their atomic coordinates). Clustering identifies low free energy regions (funnels) one needs to further explore. We will be referring to the problem of exploring a funnel as the *medium-range problem*. The proposed SDU method aims at solving this problem by exploring the funnel shape of the energy function. Next we review key properties of the free energy functions.

A. Free energy and its mathematical properties

Free energy 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. In particular, $\Delta G = G^{RL} - G^R - G^L$, where G^R and G^L are the free energies of the receptor and ligand, respectively, and both G^R and G^L are independent of the conformation of the complex; thus, minimizing G^{RL} reduces to minimizing ΔG.

We use free energy evaluation models that combine molecular mechanics with continuum electrostatics and empirical solvation terms (Halperin et al. [11]). For protein docking applications the binding free energy can be approximated as

$$\Delta G = \Delta E_{elec} + \Delta G^*_{des} - T\Delta S_{sc} + \Delta E_{vdw}, \qquad (2)$$

where ΔE_{elec} is the change in electrostatic energy upon binding, ΔE_{vdw} is the change in van der Waals energy, and ΔG^{*}_{des} – the desolvation free energy – is associated with the removal of the solvent from the interacting surfaces of the protein and ligand, where the * denotes that it includes the change in the solute-solvent van der Waals energy. The entropy term, $-T\Delta S_{sc}$, accounts for the decrease in entropy experienced by the interface side chains upon binding.

Multi-frequency behavior: The free energy function can be regarded as the sum of three components with different frequencies. First, the following "smooth" component

$$\Delta G_s = \Delta E_{elec} + \Delta G_{des} - T\Delta S_{sc}, \qquad (3)$$

changes relatively slowly along any reaction path, where the desolvation free energy ΔG_{des} does not include the solute-solvent van der Waals term. The full contribution of the desolvation forces is estimated by the atomic contact potential (ACP) [12], where the ACP potential includes the self-energy change on desolvating charge or polar atom groups and the side chain entropy loss, i.e., $\Delta G_{ACP} = \Delta G_{des} - T\Delta S_{sc}$. ΔG_{ACP} is much less sensitive to structural perturbations than ΔE_{vdw} and ΔE_{elec}. The electrostatic energy ΔE_{elec} changes with an intermediate frequency, and the frequency of change is very high for ΔG_{vdw}. Thus, the van der Waals component is essentially a high frequency noise, because on its own it carries very little information about the distance from the native state.

Funnel-like shape: As discussed in Sec. I the concept of the funnel-shaped energy landscapes (see Fig. 1) has had a significant impact on understanding protein folding and docking. This behavior has a physical explanation. Due to the term ΔE_{vdw}, the free energy function defined by (2) exhibits high sensitivity to structural perturbations. In addition, in any conformation without steric overlaps the van der Waals energy term is favorable, resulting in a number of local minima. Since most random perturbations yield overlaps, the local minima are separated by high energy barriers. In the conformations without overlaps, the solute-solute and solute-solvent interfaces are equally well packed, and thus the intermolecular van der Waals interactions in the bound state are largely balanced by solute-solvent interactions in the free state. Therefore, restricting consideration to such conformations, both ΔE_{vdw} and the van der Waals contributions to the desolvation free energy can be removed, and the binding free energy is reduced to the smooth components described by (3). Thus, 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.

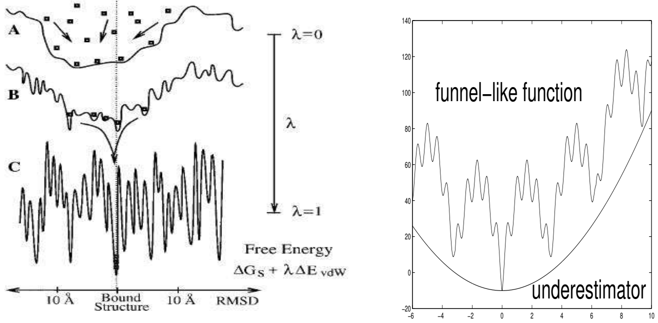


Fig. 1. Left: Protein docking funnels and the operation of the docking method SmoothDock [8]. Right: A funnel-like shaped function and a quadratic function underestimating the surface spanned by the local minima.

B. Implications for global optimization and our method

Summarizing the discussion thus far it is evident that the free energy function in (2) has a huge number of local minima caused by high-frequency terms; primarily the van der Waals term. Even if one ignores changes in the interface side-chains, local minimization in $SE(3)$ [2] enhanced with multi-start would have to be quite lucky to approach a global minimum in some reasonable time. Generic global optimization approaches (e.g., those supported by GAMS [13]) typically require some regularity conditions (e.g., twice continuously differentiable functions) and are not designed for minimizing complex free energy functions having a huge number of local minima. [1] A deterministic branch-and-bound global optimization method, the αBB method (see Floudas et al. [5]), has been a part of a successful protein-folding approach [15] but it can be costly since it systematically explores all of the search region. The same will be true for other deterministic approaches. In fact, this is part of the motivation of the hybrid approach by Klepeis et al. [16] that combine αBB with a randomized search approach using ideas from genetic algorithms. To the best of our knowledge, the only systematic optimization method consistently applied in protein-protein docking is simulated annealing [6]. The CGU method [9], which motivated our work, has also been used in protein folding.

Fortunately, the funnel-like shape of the energy function can provide some guidance. Suppose we have identified (by FFT filtering and clustering) a number of conformations in the native energy funnel and we seek to determine the native conformation. The low-frequency energy terms (cf. (3)) form a smooth function which can often be well approximated by a convex quadratic. To exploit this structure we work on the envelope surface spanned by the set of local minima. This surface inherits the smooth behavior of the low-frequency energy terms. More specifically, we generate a large number of local minima and construct a convex quadratic function that forms the tightest underestimator of all of them. This quadratic function suggests the location of the deep energy valley containing the native structure. We use this information to iteratively refine our search. We note that this approach does not need access to individual energy terms; it only requires access to a program evaluating ΔG.

III. SDU: Semi-Definite Underestimator Method

SDU consists of two key components: (*i*) underestimating the surface spanned by the local minima (see Fig. 1 (Right)), and (*ii*) using the underestimator to guide further exploration.

[1] For instance, BARON [14], does not support sinusoid functions used in our test functions of Sec. VI.

A. Constructing an Underestimator

Let us denote by $f : \mathbb{R}^n \to \mathbb{R}$ the free energy function we seek to minimize and assume we have obtained a set of K local minima $\boldsymbol{\phi}^1, \dots, \boldsymbol{\phi}^K$ of $f(\cdot)$. We are interested in constructing a convex quadratic function $U(\boldsymbol{\phi})$ that underestimates $f(\cdot)$ at all local minima $\boldsymbol{\phi}^i$, $i = 1, \dots, K$. Let

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

where $\mathbf{Q} \in \mathbb{R}^{n\times n}$ is positive semi-definite (hence, $U(\cdot)$ is convex), $\mathbf{b} \in \mathbb{R}^n$, and c is a scalar.

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

$$\begin{array}{ll} \min & \sum_{j=1}^{K} \left(f(\boldsymbol{\phi}^j) - c - {\boldsymbol{\phi}^j}'\mathbf{Q}\boldsymbol{\phi}^j - \mathbf{b}'\boldsymbol{\phi}^j\right) \\ \text{s.t.} & f(\boldsymbol{\phi}^j) \geq c + {\boldsymbol{\phi}^j}'\mathbf{Q}\boldsymbol{\phi}^j + \mathbf{b}'\boldsymbol{\phi}^j,\ j = 1, \dots, K, \\ & \mathbf{Q} \succeq 0, \end{array} \tag{4}$$

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

Let now introduce vectors $\mathbf{b}^+, \mathbf{b}^- \geq \mathbf{0}$, scalars $c^+, c^- \geq 0$ satisfying $\mathbf{b} = \mathbf{b}^+ - \mathbf{b}^-$ and $c = c^+ - c^-$, and slack variables $\mathbf{s} = (s_1, \dots, s_K)$. Define the block diagonal matrix

$$\mathbf{Y} \overset{\triangle}{=} \text{diag}(\mathbf{Q}, \text{diag}(\mathbf{b}^+), \text{diag}(\mathbf{b}^-), c^+, c^-, \text{diag}(\mathbf{s})). \tag{5}$$

Define $\mathbf{F}_0 \overset{\triangle}{=} \text{diag}(\text{diag}(\mathbf{0}), -\text{diag}(\mathbf{e}))$, $\mathbf{F}_j \overset{\triangle}{=} \text{diag}(\boldsymbol{\phi}^j\boldsymbol{\phi}^{j'}, \text{diag}(\boldsymbol{\phi}^j), -\text{diag}(\boldsymbol{\phi}^j), 1, -1, \text{diag}(\mathbf{e}_j))$, for all $j = 1, \dots, K$, where $\mathbf{0}$ is the $(3n+2)$-dimensional zero vector, $\mathbf{e}$ is the K-dimensional vector of ones, and $\mathbf{e}_j$ is the jth unit vector. Let also $\mathbf{E}_{i,j}$ denote the $(3n + K + 2) \times (3n + K + 2)$ matrix with all elements equal to zero except the (i, j)th element which is equal to 1. Then, (4) can be written:

$$\begin{array}{lll} (\textbf{SDP-P}) & \max \mathbf{F}_0 \bullet \mathbf{Y} & \\ & \text{s.t.} & \mathbf{F}_j \bullet \mathbf{Y} = f(\boldsymbol{\phi}^j),\ j = 1, \dots, K, \\ & & \mathbf{E}_{i,j} \bullet \mathbf{Y} = 0, i = n + 1, \dots, 3n + K + 2, \\ & & \qquad j = 1, \dots, i - 1, \\ & & \mathbf{Y} \succeq 0, \end{array} \tag{6}$$

where the decision variable is the matrix $\mathbf{Y}$. Problem (**SDP-P**) in (6) is a *Semi-Definite Programming (SDP)* problem (see Ben-Tal and Nemirovski [17] or Boyd and Vandenberghe [18]). SDP problems can be solved efficiently using interior-point methods [17]. General purpose solvers for SDP problems are readily available (e.g., [19]).

The dual to (**SDP-P**) in (6) is the problem

$$\begin{array}{ll} (\textbf{LMI-D}) & \min \sum_{j=1}^{K} x_j f(\boldsymbol{\phi}^j) \\ & \text{s.t. } \mathbf{Z} = \sum_{j=1}^{K} x_j \mathbf{F}_j + \sum_{i=n+1}^{3n+K+2} \sum_{j=1}^{i-1} w_{i,j}\mathbf{E}_{i,j} - \mathbf{F}_0, \\ & \quad \mathbf{Z} \succeq 0, \end{array} \tag{7}$$

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

Theorem III.1 *Consider a function $f : \mathbb{R}^n \to \mathbb{R}$ and a set of local minima $\boldsymbol{\phi}^1, \dots, \boldsymbol{\phi}^K$ 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 (6), where $\mathbf{Y}$ is defined as in (5). Let $U(\boldsymbol{\phi}) \overset{\triangle}{=} \boldsymbol{\phi}'\mathbf{Q}\boldsymbol{\phi} + (\mathbf{b}^+ - \mathbf{b}^-)'\boldsymbol{\phi} + (c^+ - c^-)$. Then $U(\cdot)$ satisfies $f(\boldsymbol{\phi}^j) \geq U(\boldsymbol{\phi}^j)$ for all $j = 1, \dots, K$ while minimizing $\|(f(\boldsymbol{\phi}^1), \dots, f(\boldsymbol{\phi}^K)) - (U(\boldsymbol{\phi}^1), \dots, U(\boldsymbol{\phi}^K))\|_1$. Moreover, the dual to Problem (**SDP-P**) is the LMI problem (**LMI-D**) in (7).*

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

B. Sampling

Suppose we are seeking the native conformation in some region $\mathscr{B}$. Using a set of local minima $\boldsymbol{\phi}^1, \dots, \boldsymbol{\phi}^K \in \mathscr{B}$ of $f(\cdot)$ we construct an underestimator $U(\cdot)$ as described in Sec. III-A. Depending on the samples we used, and assuming that the constructed underestimator reflects the general structure of $f(\cdot)$, the global minimum of $U(\cdot)$, say $\boldsymbol{\phi}^P$, is in the vicinity of the global minimum of $f(\cdot)$. We will be referring to $\boldsymbol{\phi}^P$ as the *predictive conformation*.

Clearly, the distance of the predictive conformation from the native one depends on whether the underestimator is constructed using a sufficiently rich set of local minima. However, even if that is the case, locating the global minimum is difficult since van der Waals interactions create a large enough number of local minima around the native conformation. Our strategy is to sample in the area around $\boldsymbol{\phi}^P$ such that conformations close to $\boldsymbol{\phi}^P$ are more likely to be selected. In addition, conformations with high enough energy can safely be ignored. This can be achieved by using the following probability density function (pdf) in $\mathscr{B}$:

$$g(\boldsymbol{\phi}) = \frac{U(\boldsymbol{\phi}) - U_{max}}{\int_{\mathscr{B}}(U(\boldsymbol{\phi}) - U_{max})\, d\boldsymbol{\phi}} \overset{\triangle}{=} \frac{U(\boldsymbol{\phi}) - U_{max}}{A},\ \boldsymbol{\phi} \in \mathscr{B},$$

where $U_{max} = \max_{\mathscr{B}} U(\boldsymbol{\phi})$ and A denotes the integral in the denominator.

To generate random samples in $\mathscr{B}$ using the above pdf we will use the so called *acceptance/rejection method.* Let $h(\phi) = 1/V$ be the uniform pdf in $\mathscr{B}$ where $V = \int_{\mathscr{B}} d\boldsymbol{\phi}$. 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, i.e.,

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

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 rule of Step 2 accepts uniformly distributed samples 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 uniformly distributed random variables $\mathbf{x}_1 \in \mathscr{B}$
   and $x_2 \in [\zeta, 1]$.
2) If $x_2 \leq R(\mathbf{x}_1)$, 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; one special case of practical interest is when $\mathscr{B}$ is a box set of the type $\{\mathbf{x} \mid l_i \leq x_i \leq v_i, i = 1, \ldots, n\}$. Then, since $U(\cdot)$ is convex it achieves its maximum at an extreme point of $\mathscr{B}$. 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 $f(\cdot)$ in some region $\mathscr{B}$ of the conformational space; it is presented in Fig. 3. Throughout the algorithm we maintain a set $\mathscr{L}$ of interesting local minima obtained so far including the best such local minimum denoted by $\boldsymbol{\phi}^G$. The evolution SDU depends on the parameters $K, \zeta \in [0, 1], m$ and ϵ, and on 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 intended 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 SDU converges 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 (probabilistically) guarantee convergence for funnel-like functions under a suitable set of conditions.

IV. The importance of the underestimator's structure

Next, we review some earlier work, study its limitations, and assess the relationships with SDU. Let us consider the SDU algorithm under the following key modifications:

1) **Underestimation**. In deriving $U(\cdot)$ impose the constraint that $\mathbf{Q}$ 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 with random (uniform) sampling in the neighborhood $\mathscr{N}(\boldsymbol{\phi}^P)$ of $\boldsymbol{\phi}^P$.

```
1) Initialization: Starting from $\bar{K}$ random points in $\mathscr{B}$
   perform local minimization to obtain $K$ ($K \geq 2n + 1$)
   distinct local minima $\boldsymbol{\phi}^1, \ldots, \boldsymbol{\phi}^K$ of $f(\cdot)$. Set $\mathscr{L} =
   \{\boldsymbol{\phi}^1, \ldots, \boldsymbol{\phi}^K\}$ and $\boldsymbol{\phi}^G = \arg\min_{\boldsymbol{\phi} \in \{\boldsymbol{\phi}^1, \ldots, \boldsymbol{\phi}^K\}} f(\boldsymbol{\phi}^i)$.
2) Underestimation: Solve Problem (SDP-P) in (6) to ob-
   tain the underestimator $U(\boldsymbol{\phi})$. Set the predictive confor-
   mation 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 $\mathscr{L} := \mathscr{L} \setminus \{\boldsymbol{\phi} \in \mathscr{L} \mid R(\boldsymbol{\phi}) <
   \zeta \text{ and } \boldsymbol{\phi} \neq \boldsymbol{\phi}^G\}$.
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 $\hat{\boldsymbol{\phi}}^P \in \mathscr{B}$ and
      $\hat{\boldsymbol{\phi}}^P \notin \mathscr{L}$, set $\mathscr{L} := \mathscr{L} \cup \{\hat{\boldsymbol{\phi}}^P\}$ and $\boldsymbol{\phi}^G :=
      \arg\min_{\boldsymbol{\phi} \in \{\boldsymbol{\phi}^G, \hat{\boldsymbol{\phi}}^P\}} f(\boldsymbol{\phi})$.
   b) Obtain $\bar{m}$ samples from the sampling algorithm of
      Fig. 2. Using these samples as starting points perform
      local minimization to obtain $m$ distinct local minima
      $\mathbf{x}^1, \ldots, \mathbf{x}^m$ of $f(\cdot)$. Set $\mathscr{L} := \mathscr{L} \cup \{\mathbf{x} \mid \mathbf{x} =
      \mathbf{x}^1, \ldots, \mathbf{x}^m, \mathbf{x} \in \mathscr{B}, \mathbf{x} \notin \mathscr{L}\}$ 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 $\|\boldsymbol{\phi}^G - \boldsymbol{\phi}^P\| < \epsilon$ or if there is no
   progress in reducing $f(\boldsymbol{\phi}^G)$ over several iterations then
   stop; otherwise go to Step 2.
```

Fig. 3. The SDU algorithm.

The resulting method is the so called *Convex Global Underestimator (CGU)* method by Phillips et al. [9] for minimizing funnel-like free energy functions in molecular modeling problems. We will argue that these two differences with our proposed SDU drastically affect the performance of the corresponding optimization method. In particular, limiting the underestimator search to the class of canonical parabolas 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. Dennis and Vajda [7] report many such cases where CGU performs poorly. Some attempts in addressing this limitation have been made by Rosen and Marcia [20] but again handling special cases.

A. Comparing the CGU and SDU underestimating approaches

As we discussed in Sec. III a quadratic underestimator will not be informative if either (*i*) $f(\cdot)$ is not funnel-shaped and the envelope of local minima can not be well approximated by a convex quadratic, or (*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 $\bar{\mathbf{Q}} \succeq 0$. We assume that (*i*) 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, and (*ii*) $f(\cdot)$ is continuous on $\mathscr{B}$. The construction of the underestimator based on utilizing all sample points in $\mathscr{B}$ can be formulated as the following (infinite dimensional) optimization problem:

$$\begin{array}{ll} \min & \int_{\mathbf{t}\in\mathscr{B}}(f(\mathbf{t}) - U(\mathbf{t}))\, d\mathbf{t} \\ \text{s.t.} & f(\mathbf{t}) \geq U(\mathbf{t}), \qquad \mathbf{t} \in \mathscr{B}, \end{array} \tag{8}$$

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 (8) 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 (8).*

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 $\mathbf{D} = \text{diag}(d_1, \ldots, d_n)$ and $d_i \geq 0$ for $i = 1, \ldots, n$. Consider the problem in (8) for such a $U(\mathbf{t})$. For simplicity of exposition let us assume that the search region $\mathscr{B}$ is the hypercube defined as $\mathscr{B} = \mathscr{B}_1 \times \cdots \times \mathscr{B}_n$ 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 $\mathbf{z} = (d_1, \ldots, d_n, b_1, \ldots, b_n, c)$. Then, after some algebra, (8) becomes equivalent to:

$$\begin{array}{lll} (\textbf{LSIP-P}) & \max & \mathbf{h}'\mathbf{z} \\ & \text{s.t.} & \mathbf{a}'(\mathbf{t})\mathbf{z} \leq f(\mathbf{t}), \qquad \mathbf{t} \in \mathscr{B}, \end{array} \tag{9}$$

where $\mathbf{z}$ is the decision vector. Note that (**LSIP-P**) involves an infinite number of constraints. A problem with such a structure is known as the *Linear Semi-Infinite Programming (LSIP)* problem (see Hettich and Kortanek [21]). Its dual can be formulated in measure space as follows:

$$\begin{array}{lll} (\textbf{LSIP-D}) & \min & \int_{\mathscr{B}} f(\mathbf{t}) d\mu(\mathbf{t}) \\ & \text{s.t.} & \int_{\mathscr{B}} \mathbf{a}(\mathbf{t}) d\mu(\mathbf{t}) = \mathbf{h},\ \mu \in M^+(\mathscr{B}), \end{array} \tag{10}$$

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

1) Approximate problem converges to LSIP: Next we explore the relationship of the underestimator obtained by solving (**LSIP-P**) in (9) to the underestimator obtained as in Sec. III, namely, by using the function values $f(\mathbf{t}^1), \ldots, f(\mathbf{t}^K)$ at a set of samples $\mathbf{t}^1, \ldots, \mathbf{t}^K$ from $\mathscr{B}$.

We introduce some additional terminology and notation. Let M_{2n+1} be the cone generated by $\mathbf{a}(\mathbf{t})$ for $\mathbf{t} \in \mathscr{B}$, i.e., $M_{2n+1} = \{\mathbf{w} \mid \mathbf{w} = \int_{\mathscr{B}} \mathbf{a}(\mathbf{t}) d\mu(\mathbf{t}), \mu \in M^+(\mathscr{B})\}$. We denote by $\text{int}(M_{2n+1})$ the interior of M_{2n+1}. We also denote by $L^P_{\geq}(\mathbf{h}, \theta)$ the level sets of the (primal) problem (**LSIP-P**), in particular, $L^P_{\geq}(\mathbf{h}, \theta) = \{\mathbf{z} \mid \mathbf{a}'(\mathbf{t})\mathbf{z} \leq f(\mathbf{t}), \forall \mathbf{t} \in \mathscr{B}; \mathbf{h}'\mathbf{z} \geq \theta\}$. We will say that the primal (**LSIP-P**) is *consistent* if it has a feasible solution. We will say that (**LSIP-P**) is *superconsistent* if the interior of the feasible set is non-empty, that is, if there exist $\mathbf{z}^*$ such that $\mathbf{a}'(\mathbf{t})\mathbf{z}^* < f(\mathbf{t})$ for all $\mathbf{t} \in \mathscr{B}$ (this is Slater's condition). Similarly, we will say that the dual (**LSIP-D**) is *consistent* if $\mathbf{h} \in M_{2n+1}$; the dual will be termed *superconsistent* if $\mathbf{h} \in \text{int}(M_{2n+1})$.

Lemma IV.2 *Assume that (*LSIP-P*) is consistent. Then (*LSIP-D*) is superconsistent and all primal level sets $L^P_{\geq}(\mathbf{h}, \theta)$ are bounded.*

Proof: Recall that we have assumed $\mathscr{B}$ to be compact and $f(\cdot)$ continuous on $\mathscr{B}$. We will first show that (**LSIP-D**) is superconsistent. To that end, note that the uniform measure in $\mathscr{B}$ is a feasible solution of (**LSIP-D**), which implies that $\mathbf{h} \in M_{2n+1}$. Furthermore, the boundary of M_{2n+1} consists of vectors $\mathbf{a}(\mathbf{t}^*)$ for some $\mathbf{t}^* \in \mathscr{B}$. Observe that $\mathbf{h}$ is not on the boundary, i.e., $\mathbf{h} \in \text{int}(M_{2n+1})$ which establishes the superconsistency of (**LSIP-D**). Using a result from [21, Theorem 6.11] the boundedness of all primal level sets follows. ■

Next consider an approximate finite problem constructed from (**LSIP-P**) by only enforcing a finite number of constraints, say for $\mathbf{t}^1, \ldots, \mathbf{t}^K \in \mathscr{B}$. More generally, let $\overline{\mathscr{B}} \subset \mathscr{B}$ be a finite subset of $\mathscr{B}$ and form the LP

$$\begin{array}{lll} (\textbf{LP-P}) & \max & \mathbf{h}'\mathbf{z} \\ & \text{s.t.} & \mathbf{a}'(\mathbf{t})\mathbf{z} \leq f(\mathbf{t}), \qquad \mathbf{t} \in \overline{\mathscr{B}}, \end{array} \tag{11}$$

which can be seen as a finite approximation of (**LSIP-P**). Define $d(\overline{\mathscr{B}}) = \max_{\mathbf{t}\in\mathscr{B}} \min_{\bar{\mathbf{t}}\in\overline{\mathscr{B}}} \|\mathbf{t} - \bar{\mathbf{t}}\|$ as a metric of density of $\overline{\mathscr{B}}$ in $\mathscr{B}$. Then, the boundedness of all (**LSIP-P**) level sets suffices to guarantee the convergence of the (**LP-P**) solutions to the (**LSIP-P**) solutions. The result, which is from [21], is provided in the following theorem.

Theorem IV.3 *For every $\epsilon > 0$ there exists $\delta_\epsilon > 0$ such that for all $\overline{\mathscr{B}} \subset \mathscr{B}$ with $d(\overline{\mathscr{B}}) < \delta_\epsilon$ it is the case that (*LP-P*) is solvable, and for every solution $\bar{\mathbf{z}}$ of (*LP-P*) there exists a solution $\mathbf{z}^*$ of (*LSIP-P*) such that $\|\bar{\mathbf{z}} - \mathbf{z}^*\| < \epsilon$.*

The result 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.

2) Limitations of the CGU underestimator: We now have the necessary machinery to analyze the CGU underestimator. In particular, we consider a class of functions to be underestimated and demonstrate how the CGU underestimator fails.

Suppose we wish to underestimate $f(\mathbf{t}) = \mathbf{t}'\bar{\mathbf{Q}}\mathbf{t} + \bar{\mathbf{b}}'\mathbf{t}$, where

$$\bar{\mathbf{Q}} = \text{diag}\left(\begin{bmatrix} \alpha_1 & \beta \\ \beta & \alpha_2 \end{bmatrix}, \alpha_3, \ldots, \alpha_n\right), \tag{12}$$

$n \geq 2$, $\alpha_i > 0$ for all i, $\beta < 0$, and $\alpha_1\alpha_2 - \beta^2 \geq 0$ to guarantee $\bar{\mathbf{Q}} \succeq 0$. This choice of $\bar{\mathbf{Q}}$ can be seen as the simplest $\bar{\mathbf{Q}}$ with off-diagonal elements; the argument we present next can be generalized to more general cases as well. Let the underestimator be given by $U(\mathbf{t}) = \mathbf{t}'\mathbf{D}\mathbf{t} + \mathbf{b}'\mathbf{t} + c$, where $\mathbf{D} = \text{diag}(d_1, \ldots, d_n) \succeq 0$, $\mathbf{b} = (b_1, \ldots, b_n)$ and $c \in \mathbb{R}$. The CGU underestimator is informative if it manages to locate the minimum of $f(\mathbf{t})$. In particular, we would like the minimizer of $U(\mathbf{t})$ to coincide with the minimizer of $f(\mathbf{t})$. We will show by contradiction that this is not possible in general.

Consider an instance of the underestimation problem at hand where $\mathscr{B}$ is the hypercube defined as $\mathscr{B} = \mathscr{B}_1 \times \cdots \times \mathscr{B}_n$ where all $\mathscr{B}_i$'s are identical and given by $\mathscr{B}_i \overset{\triangle}{=} [-2T/3, T/3]$ for all $i = 1, \ldots, n$ and $T > 1$. (**LSIP-D**) takes the form

$$\begin{aligned} (\textbf{LSIP-D}) \quad \min \quad & \sum_{i=1}^{n} \alpha_i I_2 + \sum_{i=1}^{n} I_1 b_i + 2\beta \mathbf{E}_\mu[t_1 t_2] \\ \text{s.t.} \quad & \int_{\mathscr{B}} \mathbf{a}(\mathbf{t}) d\mu(\mathbf{t}) = \mathbf{h}, \ \mu \in M^+(\mathscr{B}), \end{aligned} \tag{13}$$

where $\mathbf{E}_\mu[\cdot]$ denotes expectation with respect to μ, $I_2 = \frac{1}{T}\int_{\mathscr{B}_1} t^2 dt = T^2/9$, and $I_1 = \frac{1}{T}\int_{\mathscr{B}_1} t dt = -T/6$. Since $\beta < 0$ the dual problem is equivalent to the problem of selecting a measure μ in order to maximize $\mathbf{E}_\mu[t_1 t_2]$ subject to the constraint that the first and second moments of the marginal distributions under μ match those of the uniform distribution (cf. definition of $\mathbf{h}$ in Subsection IV-A). However, for any measure $\mu \in M^+(\mathscr{B})$ the expectation $\mathbf{E}_\mu[(t_1 - t_2)^2]$ is nonnegative, hence,

$$2\mathbf{E}_\mu[t_1 t_2] \le \mathbf{E}_\mu[t_1^2] + \mathbf{E}_\mu[t_2^2] = 2I_2.$$

It follows that at optimality the above inequality holds with equality, that is, t_1 and t_2 are co-linear.

Next, let $V_{(\textbf{LSIP-P})}$ and $V_{(\textbf{LSIP-D})}$ denote the optimal values of (**LSIP-P**) and (**LSIP-D**), respectively. We have already shown (cf. Lemma IV.2) that (**LSIP-D**) is superconsistent. Moreover, the argument in the previous paragraph suggests that $V_{(\textbf{LSIP-D})}$ is finite. Consequently, (**LSIP-P**) has an optimal solution and $V_{(\textbf{LSIP-P})} = V_{(\textbf{LSIP-D})}$ (cf. [21, Theorem 6.9]). In the terminology of [21] we have *perfect duality*. The necessary and sufficient conditions for optimality of a primal-dual pair are primal feasibility, dual feasibility, and perfect duality.

Let $(\hat{\mathbf{z}}, \hat{\mu})$ be an optimal primal-dual pair, where $\hat{\mathbf{z}} = (\hat{d}_1, \ldots, \hat{d}_n, \hat{b}_1, \ldots, \hat{b}_n, \hat{c})$, and $\hat{\mu}$ is a measure in $M^+(\mathscr{B})$ with the properties identified above. The optimality conditions are:

$$\sum_{i=1}^{n} t_i^2 \hat{d}_i + \sum_{i=1}^{n} t_i \hat{b}_i + \hat{c} \le \sum_{i=1}^{n} t_i^2 \alpha_i + 2\beta t_1 t_2 + \sum_{i=1}^{n} t_i \bar{b}_i, \forall \mathbf{t} \in \mathscr{B}, \tag{14}$$

$$\int_{\mathscr{B}} \mathbf{a}(\mathbf{t}) d\hat{\mu}(\mathbf{t}) = \mathbf{h}, \qquad \hat{\mu} \in M^+(\mathscr{B}), \tag{15}$$

$$\sum_{i=1}^{n} I_2 \hat{d}_i + \sum_{i=1}^{n} I_1 \hat{b}_i + \hat{c} = \sum_{i=1}^{n} \alpha_i I_2 + \sum_{i=1}^{n} I_1 \bar{b}_i + 2\beta I_2. \tag{16}$$

Consider now the primal feasibility condition (14) at $\mathbf{t} = \mathbf{0}$ and at $\mathbf{t} = -2T\mathbf{e}/3$. These two inequalities along with (16) imply $\hat{c} = 0$. Furthermore, (14) at $\mathbf{t} = -T\mathbf{e}/3$ and at $\mathbf{t} = T\mathbf{e}/3$ along with (16) yield

$$\textstyle\sum_{i=1}^{n} (\hat{b}_i - \bar{b}_i) = 0. \tag{17}$$

Next, fix some $j \in \{3, \ldots, n\}$ and some $k > 3/T$ and apply (14) at $\mathbf{t}$ such that $t_j = -1/k$, $t_l = 1/k$ for all $l \neq j$ and at $\mathbf{t}$ such that $t_j = 1/k$, $t_l = -1/k$ for all $l \neq j$. These, along with (17), $\hat{c} = 0$, and (16), imply

$$\hat{b}_1 + \hat{b}_2 = \bar{b}_1 + \bar{b}_2, \qquad \hat{b}_j = \bar{b}_j, \ j = 3, \ldots, n. \tag{18}$$

Now, for any $j \in \{3, \ldots, n\}$ and some $k > 3/T$ apply (14) at $\mathbf{t}$ such that $t_j = \sqrt{2}/k$, $t_l = 1/k$ for all $l \neq j$ and at $\mathbf{t}$ such that $t_j = 0$, $t_l = 1/k$ for all $l \neq j$. These inequalities, along with (17), $\hat{c} = 0$, and (16), lead to

$$\hat{d}_1 + \hat{d}_2 = \alpha_1 + \alpha_2 + 2\beta, \qquad \hat{d}_j = \alpha_j, \ j = 3, \ldots, n. \tag{19}$$

To proceed consider $\hat{\mathbf{z}}$ with $\hat{c} = 0$ satisfying (18) and (19). Let $x = \hat{d}_1 - \alpha_1 - \beta$ and $y = \hat{b}_1 - \bar{b}_1$. We will show that $(x, y) = (0, 0)$. Assume otherwise. Using (18) and (19) the primal feasibility condition (14) reduces to

$$(t_1 - t_2)[\beta(t_1 - t_2) + x(t_1 + t_2) + y] \le 0, \ \forall t_1, t_2 \in \mathscr{B}_1. \tag{20}$$

Let $0 < \delta < T/3$ and apply the above at $(t_1, t_2) = (\delta, -\delta)$ and $(t_1, t_2) = (-\delta, \delta)$. It follows

$$2\beta\delta \le y \le -2\beta\delta.$$

If $y \neq 0$, by selecting δ close enough to zero we can violate primal feasibility. Therefore, $y = 0$. Further, let $1 < k < T$ and apply (20) at $(t_1, t_2) = (k/3, 1/3)$ and $(t_1, t_2) = (1/3, k/3)$. It follows

$$\beta\tfrac{k-1}{k+1} \le x \le \beta\tfrac{1-k}{k+1}.$$

If $x \neq 0$, by selecting k close enough to one we can violate primal feasibility. Therefore, $x = 0$. In summary, we have

$$\begin{aligned} \hat{d}_i = \alpha_i + \beta, \ i = 1, 2, \quad \hat{d}_j = \alpha_j, \ j = 3, \ldots, n, \\ \hat{b}_j = \bar{b}_j, \ j = 1, \ldots, n, \qquad \hat{c} = 0. \end{aligned} \tag{21}$$

The minimum of the CGU underestimator, say $\hat{\mathbf{t}}$, is given by $\hat{\mathbf{t}} = -\frac{1}{2}\hat{\mathbf{D}}^{-1}\hat{\mathbf{b}}$ which due to (IV-A.2) yields

$$\hat{t}_i = -\tfrac{\bar{b}_i}{2(\alpha_i + \beta)}, \ i = 1, 2, \qquad \hat{t}_j = -\tfrac{\bar{b}_j}{2\alpha_j}, \ j = 3, \ldots, n.$$

However, the minimum of $f(\cdot)$, say $\mathbf{t}^*$, is given by

$$\begin{aligned} t_1^* &= -\tfrac{1}{2(\alpha_1\alpha_2 - \beta^2)}(\alpha_2 \bar{b}_1 - \beta \bar{b}_2), \\ t_2^* &= -\tfrac{1}{2(\alpha_1\alpha_2 - \beta^2)}(\alpha_1 \bar{b}_2 - \beta \bar{b}_1), \ t_j^* = -\tfrac{\bar{b}_j}{2\alpha_j}, \ j = 3, \ldots, n. \end{aligned}$$

Clearly, $\hat{\mathbf{t}}$ is different from $\mathbf{t}^*$ in general and the CGU underestimator can not locate the minimum of $f(\cdot)$. We summarize the result of this subsection in the following theorem.

Theorem IV.4 *Let $f(\mathbf{t}) = \mathbf{t}'\bar{\mathbf{Q}}\mathbf{t} + \bar{\mathbf{b}}'\mathbf{t}$, where $\bar{\mathbf{Q}} \succeq 0$ is given in (12). Further, let $\hat{U}(\mathbf{t}) = \mathbf{t}'\hat{\mathbf{D}}\mathbf{t} + \hat{\mathbf{b}}'\mathbf{t} + \hat{c}$ be the function formed from the optimal solution to (**LSIP-P**), i.e., the optimal CGU underestimator to $f(\mathbf{t})$. Then, in general, the minimizer of the underestimator is different from the minimizer of $f(\mathbf{t})$.*

The result of Thm. IV.4 is significant since it establishes that the CGU underestimator cannot locate the minimum of $f(\cdot)$ even if it uses an infinite number of samples. In contrast, according to Prop. IV.1 and Thm. IV.3, SDU will find the minimum of $f(\cdot)$ as the number of samples grows to infinity.

V. On SDU's Convergence

In this section we provide a rough analysis of the SDU algorithm and show that under appropriate conditions it converges to the global minimum of $f(\cdot)$. Since the free energy functions we seek to minimize are funnel-like, for the remainder of this section we will impose a set of structural assumptions on $f(\cdot)$ and the search region $\mathscr{B}$ that reflect this structure. We denote by epi(f) the epigraph of $f(\cdot)$ which is defined as epi$(f) = \{(\boldsymbol{\phi}, w) \mid \boldsymbol{\phi} \in \mathscr{B}, f(\boldsymbol{\phi}) \leq w\}$. We also denote by 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 (Right)). The next proposition provides a sufficient condition for exactly determining the quadratic function $\tilde{U}(\cdot)$ which tightly underestimates all local minima of $f(\cdot)$.

Proposition V.1 *Let Assumption A prevail. Consider the SDU algorithm of Fig. 3 and suppose that in Step 1 we obtain at least $K = \frac{(n+1)(n+2)}{2}$ distinct local minima $\boldsymbol{\phi}^1, \ldots, \phi^K$ of $f(\cdot)$ which are extreme points of conv(epi(f)). Then, the underestimator $U(\cdot)$ obtained in Step 2 is identical to $\tilde{U}(\cdot)$.*

Proof: Constructing the underestimator $U(\cdot)$ is equivalent to solving problem (4). Clearly, $(\mathbf{Q}, \mathbf{b}, c) = (\tilde{\mathbf{Q}}, \tilde{\mathbf{b}}, \tilde{c})$ is an optimal solution because it is feasible and achieves a zero cost. At local minima $\boldsymbol{\phi}^j$ we have $U(\boldsymbol{\phi}^j) = f(\boldsymbol{\phi}^j)$ for all $j = 1, \ldots, K$ which form a linear system of equations. This linear system has $(n^2 + n)/2$ unknown elements of $\mathbf{Q}$ plus n unknown elements of $\mathbf{b}$ plus the unknown value of c. It follows that for $K \geq \frac{(n+1)(n+2)}{2}$ we obtain the exact quadratic function which passes from all extreme points of conv(epi(f)). ■

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 $\boldsymbol{\phi}^*$.

Theorem V.2 *Let Assumption A prevail. Consider the SDU algorithm of Fig. 3 and assume that $\mathscr{B}$ contains at least $\frac{(n+1)(n+2)}{2}$ distinct local minima of $f(\cdot)$ which are extreme points of conv(epi(f)). Suppose that in Step 1 of the algorithm we obtain K local minima $\boldsymbol{\phi}^1, \ldots, \phi^K$ of $f(\cdot)$ starting from uniformly distributed starting points in $\mathscr{B}$. 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 ϕ^* of $f(\cdot)$ as $K \to \infty$, namely, $\lim_{K\to\infty} \mathbf{P}[\boldsymbol{\phi}^P = \phi^*] = 1$.*

Proof: Due to Assumption A function $f(\cdot)$ satisfies all the conditions of the capture theorem [22, Proposition 1.2.5]. In particular, for every local minimum $\boldsymbol{\phi}^j$ of $f(\cdot)$ which is an extreme point of conv(epi(f)) there exists an open set $\mathscr{S}_j = \{\boldsymbol{\phi} \mid \|\boldsymbol{\phi} - \boldsymbol{\phi}^j\| < \epsilon^j\}$ such that any convergent local minimization algorithm initialized with any point in $\mathscr{S}_j$ converges to $\boldsymbol{\phi}^j$. [2] We assumed that $\mathscr{B}$ contains at least $M = \frac{(n+1)(n+2)}{2}$ distinct local minima of $f(\cdot)$ which are extreme points of conv(epi(f)). Without loss of generality let $\boldsymbol{\phi}^1, \ldots, \boldsymbol{\phi}^M \in \mathscr{B}$ and set $\rho_j = (\int_{\boldsymbol{\phi}\in\mathscr{S}_j\cap\mathscr{B}} d\boldsymbol{\phi})/(\int_{\boldsymbol{\phi}\in\mathscr{B}} d\boldsymbol{\phi})$, for $j = 1, \ldots, M$. Note that there exists an $\epsilon > 0$ such that $\rho_j > \epsilon$ for all j. By virtue of Proposition V.1 we need at least $M = \frac{(n+1)(n+2)}{2}$ distinct local minima of $f(\cdot)$ which are extreme points of conv(epi(f)) in order to exactly construct $\tilde{U}(\boldsymbol{\phi})$; this suffices to yield $\boldsymbol{\phi}^*$. Let $\mathbf{P}$[success] the probability that we do indeed obtain M distinct local minima of $f(\cdot)$ which are extreme points of conv(epi(f)) by randomly sampling K points in $\mathscr{B}$ and performing local minimization starting from every sample. Let also $\mathscr{A}_j$ the event that such random sampling yields $\boldsymbol{\phi}^j$, $j = 1, \ldots, M$, at least once. For any event $\mathscr{A}$ we denote by $\overline{\mathscr{A}}$ its complement. We have

$$\mathbf{P}[\text{success}] \geq \mathbf{P}[\mathscr{A}_1 \cap \cdots \cap \mathscr{A}_M] = 1 - \mathbf{P}[\overline{\mathscr{A}_1} \cup \cdots \cup \overline{\mathscr{A}_M}] \geq 1 - \sum_{j=1}^{M} \mathbf{P}[\overline{\mathscr{A}_j}] = 1 - \sum_{j=1}^{M} (1 - \rho_j)^K. \quad (22)$$

The first inequality above is due to the fact that in the right hand side we consider only local minima $\boldsymbol{\phi}^1, \ldots, \boldsymbol{\phi}^M$; $\mathscr{B}$ may contain additional local minima which are extreme points of conv(epi(f)). The second inequality above uses the union bound. Next observe that since the ρ_j are bounded away from zero the right hand side of (22) converges to one as $K \to \infty$. ■

We can think of Theorem V.2 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. Actual free energy functions are expected to deviate from the structure put forth in Assumption A. In particular, the local minima of $f(\cdot)$ may not be tightly underestimated by a quadratic function and/or the global minimum of a quadratic underestimator may not coincide with $\boldsymbol{\phi}^*$. These factors have motivated the form of the SDU given in Fig. 3.

VI. Numerical Results for a set of test functions

In this section we apply SDU to test functions that were selected to resemble free energy functions. Throughout we compare SDU with CGU. [3] Here, and for all numerical results reported in this paper, we have used the SDPA [19] semi-definite programming solver.

The test functions we use are from [7] and are given by

$$f(\boldsymbol{\phi}) = f_1(\boldsymbol{\phi}) + f_2(\boldsymbol{\phi}) + f_3(\boldsymbol{\phi}), \quad (23)$$

$$f_1(\boldsymbol{\phi}) = \sum_{i=1}^{n/2} (a\phi_{2i-1}^2 + b\phi_{2i}^2), \;\; f_3(\boldsymbol{\phi}) = \sum_{i=1}^{n} A|\sin(\gamma\phi_{2i})|,$$

[2] For simplicity of the exposition we assume that we can determine $\boldsymbol{\phi}^j$ exactly. Local minimization algorithms can approach $\boldsymbol{\phi}^j$ arbitrarily closely. Using continuity arguments a similar proof can establish that $\boldsymbol{\phi}^P$ converges in probability to an arbitrarily small neighborhood of $\boldsymbol{\phi}^*$ as $K \to \infty$.

[3] We followed the original CGU algorithm as described in [9] and implemented in code made available by the authors of [9].

$$f_2(\boldsymbol{\phi}) = \sum_{i=1}^{n/2} [-c\cos(\alpha\phi_{2i-1}) - d\cos(\beta\phi_{2i})],$$

where $\boldsymbol{\phi} = (\phi_1, \ldots, \phi_n)$, $a = 1$, $b = 2$, $c = 3$, $d = 4$, $\alpha = 3\pi$, $\beta = 4\pi$, and $\gamma = 3.3\pi$. Note that the origin is the global minimum of $f(\cdot)$. The parameter A determines the amplitude of the high frequency term $f_3(\cdot)$, thus, the larger A the more "noisy" and harder to optimize we expect $f(\cdot)$ to be.

First, and to illustrate once again the flexibility that a general quadratic underestimator adds, we consider a version of the test function in (23) for $n = 2$ and rotate the coordinate system with respect to the two axis by 45°.[4] Fig. 4 (top) depicts the underestimators (along a certain vertical hyperplane) obtained by CGU and SDU, denoted by $U_{CGU}(\cdot)$ and $U_{SDU}(\cdot)$ respectively, in their first iteration based on 20 ($= 8n + 4$ for $n = 2$ as required by CGU) uniformly sampled local minima. $U_{CGU}(\boldsymbol{\phi})$ has the global minimum at $(0.5225, -0.7238)$ and $U_{SDU}(\boldsymbol{\phi})$ at $(0.2669, 0.4581)$, which is closer to the true global minimum $(0, 0)$. The bottom figure in Fig. 4 depicts the contours of the two underestimators illustrating that they predict rather different global minima.

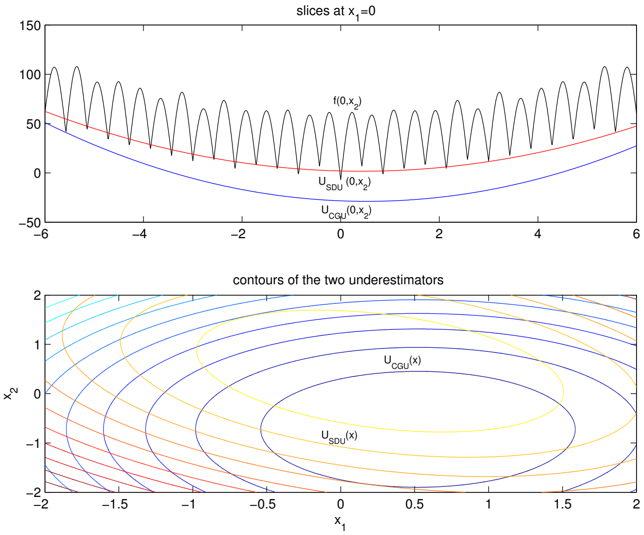


Fig. 4. SDU derives tighter underestimators.

Our next example, concerns the 6-dimensional version of the test function in (23). This is in fact the appropriate space for rigid docking problems (translation and rotation). We applied both CGU and SDU and report various indicators of their performance. For both cases the search region $\mathscr{B}$ was selected as the box $\{\boldsymbol{\phi} \mid -100 \leq \phi_i \leq 10, i = 1 \ldots, n\}$ so that the global minimum at the origin is not at the center of $\mathscr{B}$. We used $8n + 4$ initial samples in $\mathscr{B}$ as required by CGU (which is more than what Prop. V.1 suggests for SDU when $n = 6$). We use the following notation: (*i*) $\boldsymbol{\phi}_0^P$ denotes the initial predictive conformation, that is, the minimum of the initial underestimator we obtain (Step 2 of SDU); (*ii*) $\hat{\boldsymbol{\phi}}_0^P$ denotes the local minimum of $f(\cdot)$ obtained by starting the local minimization at $\boldsymbol{\phi}_0^P$; (*iii*) $\boldsymbol{\phi}^G$ denotes the best local minimum of $f(\cdot)$ found throughout the evolution of the algorithm; (*iv*) $D(\boldsymbol{\phi})$ denotes the (Euclidean) distance of a conformation $\boldsymbol{\phi}$ from the known optimal conformation which is at the origin, i.e., $D(\boldsymbol{\phi}) = (\sum_{i=1}^{n} \phi_i^2)^{1/2}$; (*v*) η denotes a metric of success of the corresponding algorithm adjusted to the difficulty of the optimization problem, more specifically, $\eta = 1\{D(\boldsymbol{\phi}^G) < 1\}$ for the case $A = 10$, $\eta = 1\{D(\boldsymbol{\phi}^G) < 1.2\}$ for the case $A = 20$, and $\eta = 1\{D(\boldsymbol{\phi}^G) < 1.5\}$ for the case $A = 30$; and, finally, (*vi*) n_f denotes the number of times the corresponding algorithm evaluates the function $f(\cdot)$ during its course.

Method		CGU	SDU	CGU	SDU	CGU	SDU
A		10		20		30	
$\boldsymbol{\phi}_0^P$	$D(\boldsymbol{\phi}_0^P)$	0.701	0.448	1.343	0.772	1.468	0.983
	$\sigma_{\boldsymbol{\phi}_0^P}$	0.291	0.164	1.185	0.295	0.659	0.429
$\hat{\boldsymbol{\phi}}_0^P$	$D(\hat{\boldsymbol{\phi}}_0^P)$	0.864	0.654	1.558	1.155	1.713	1.466
	$\sigma_{\hat{\boldsymbol{\phi}}_0^P}$	0.527	0.558	1.281	0.802	0.921	0.930
	$f(\hat{\boldsymbol{\phi}}_0^P)$	18.260	13.151	56.159	48.955	92.472	76.981
	$\sigma_{f(\hat{\boldsymbol{\phi}}_0^P)}$	13.532	12.435	35.353	21.347	28.816	29.100
$\boldsymbol{\phi}^G$	$D(\boldsymbol{\phi}^G)$	0.752	0.435	1.400	0.897	1.723	1.067
	$\sigma_{\boldsymbol{\phi}^G}$	0.578	0.304	0.695	0.631	0.729	0.566
	$f(\boldsymbol{\phi}^G)$	-3.424	4.659	15.247	28.911	26.321	49.757
	$\sigma_{f(\boldsymbol{\phi}^G)}$	6.425	9.435	7.304	13.810	9.298	20.167
$\eta(\%)$		**69**	**95**	**46**	**88**	**41**	**83**
n_f		157.30	91.67	153.66	91.48	169.52	92.38

TABLE I
CGU AND SDU APPLIED TO THE TEST FUNCTION IN (23) WHEN $n = 6$.

Table I reports results for the function in (23) and Table II (left) reports results for the function in (23) when we rotate the coordinate system by 60° with respect to all six axes. In each of these tables we considered three problem instances for A equal to 10, 20, and 30, respectively. For each case we performed 100 independent runs of both CGU and SDU. [5] The various values we report are the means of the corresponding quantities over the 100 independent runs. We also report the corresponding standard deviations; σ_X denotes the standard deviation of X.

We should note that the most important performance metric is the closeness of the conformation found to the best one (measured by $D(\boldsymbol{\phi}^G)$ and η). There are a number of reasons for this preference, including that (*i*) the 3D structure of the bound complex determines its function, (*ii*) molecular simulation may be used to discover the native conformation starting from a conformation close to it, and (*iii*) the energy evaluation models used by molecular docking algorithms are approximations of the actual free energy function. As a result, a conformation closer to the native one (like the ones SDU discovers) is more useful than some deep local minima further away (like the ones CGU may discover).

For the cases where the test function is not rotated, CGU, not surprisingly, performs reasonably well. SDU produces

[4] That is, we minimize $f(\mathbf{R}^{-1}\boldsymbol{\phi})$ where $\mathbf{R}$ is the product of axis rotation matrices, one for each axis.

[5] Note that for each run a number of random samples in $\mathscr{B}$ are been generated.

Method		ASA	CGU	SDU	ASA	CGU	SDU	ASA	CGU	SDU
A		10			20			30		
ϕ_0^P	$D(\phi_0^P)$		3.506	0.456		11.528	0.731		11.727	1.035
	$\sigma_{\phi_0^P}$		2.548	0.184		25.15	0.302		20.629	0.400
$\hat{\phi}_0^P$	$D(\hat{\phi}_0^P)$		3.306	0.659		4.722	0.829		8.614	1.259
	$\sigma_{\hat{\phi}_0^P}$		1.736	0.639		2.991	0.440		13.424	0.661
	$f(\hat{\phi}_0^P)$		50.769	25.649		109.156	66.150		415.843	95.0605
	$\sigma_{f(\hat{\phi}_0^P)}$		27.153	12.260		82.185	23.013		1669.64	29.169
ϕ^G	$D(\phi^G)$	2.66	1.837	0.490	3.53	2.539	0.799	4.16	3.013	1.128
	σ_{ϕ^G}	0.92	0.672	0.322	1.04	0.952	0.509	1.13	1.117	0.698
	$f(\phi^G)$	-0.87	8.108	17.006	9.01	26.319	49.985	19.53	42.614	75.172
	$\sigma_{f(\phi^G)}$	7.48	5.19	10.46	10.5	9.45	17.01	12.28	12.18	22.28
$\eta(\%)$		**0**	**7**	**93**	**0**	**3**	**86**	**0**	**4**	**84**
n_f		86106	230.36	91.89	81500	229.84	85.95	82273	246.74	89.77

i	ϕ_1^G	ϕ_2^G	ϕ_3^G	$f(\phi^G)$	$D(\phi^G)$
1	-0.195	-2.716	0.702	-65.58	2.82
2	-0.403	0.282	-0.197	-102.25	0.53

i	ϕ_1^G	ϕ_2^G	ϕ_3^G	$f(\phi^G)$	$D(\phi^G)$
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 II

LEFT: CGU AND SDU APPLIED TO THE TEST FUNCTION IN (23) WHEN $n = 6$ AND THE COORDINATE SYSTEM IS ROTATED BY 60° WITH RESPECT TO ALL SIX AXES. RIGHT: SDU APPLIED TO THE 1BRS (TOP) AND 1MLC (BOTTOM).

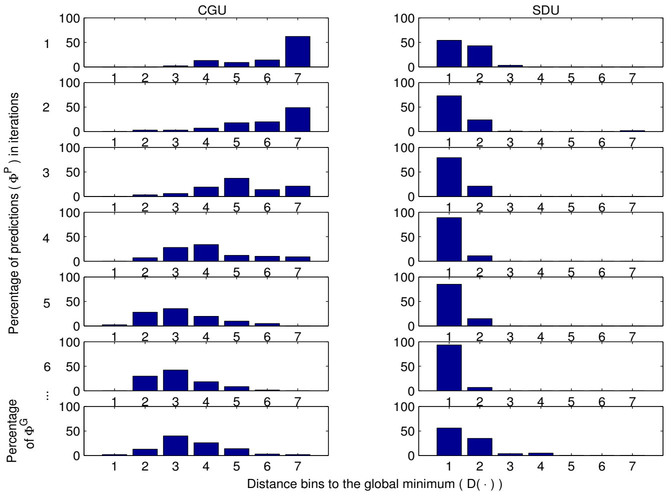


Fig. 5. Histograms of predictions and final solutions reported by CGU and SDU for the case corresponding to $A = 30$ in Table II (left).

conformations with somewhat higher energy but closer to the native one. The performance difference between SDU and CGU is more dramatic in the cases where the test function has been rotated. In the three cases considered CGU fails to discover a conformation close enough to the native one while SDU largely succeeds (observe the differences in η). The key reason, we believe, is that SDU can better capture the structure of the native funnel, which is used to bias sampling in that area. This latter point is perhaps more clearly illustrated in Fig. 5. We draw histograms of the distance of the predictive conformation during the evolution of both algorithms. "Prediction i" corresponds to ϕ_k^P obtained at the kth iteration. The horizontal axis plots the values of $D(\phi_k^P)$ ($D(\phi^G)$ for the bottom graph) and the vertical axis plots the percentage of problem instances (out of the 100 runs). It can be seen that SDU starts with a much better initial predictive conformation than CGU and maintains its predictive conformation within 2 units of the native one at all subsequent iterations. The best local minimum found by SDU is not always within 1 unit, but in 84% of the cases it was within 1.5 unit (vs. 4% with CGU).

We also compared SDU and CGU to a commonly used simulated annealing algorithm, the ASA algorithm in [23] (see Tab. II). Even though ASA reaches some deep local minima it ends up converging far away from the global minimum.

In terms of computational efficiency and running time, CGU and ASA take on the order of a few seconds in a top-of-the-line PC and SDU takes on the order of 5 minutes without particular effort at optimizing the code. The key reason is that SDU solves a number of (relatively large) SDP problems. However, SDU performs much fewer function evaluations (by a factor of 3 compared to CGU and by a factor of 1000 compared to ASA as it is evident by the n_f values in Tab. II). For the test functions we considered in this section the function evaluation time is negligible using (23). Yet, in docking actual proteins function evaluations (e.g., using CHARMM [24] or other complex potentials) dominate all other tasks; for instance, in the results we report in the next section function evaluations take more than 90% of the running time.

VII. 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 bodies. This is certainly true for their backbones and, as we will see, we will allow some flexibility in determining the conformation of the side chains in the interface between the two proteins. We consider 3D problems where one protein is held fixed, both proteins are 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 constrain these three translational variables to be in a 10Å cube centered around the origin of the coordinate system, which corresponds to the native position of the ligand. That is, we have already identified the dominant funnel of the energy function and seek to solve the medium-range energy minimization problem within this funnel.

i	ϕ_1^G	ϕ_2^G	ϕ_3^G	$f(\boldsymbol{\phi}^G)$	$D(\boldsymbol{\phi}^G)$
1	1.001	-2.555	1.395	-25.98	3.08
2	1.307	-2.369	1.363	-28.77	3.03
3	1.088	-1.610	1.310	-33.47	2.34
4	1.083	-2.612	-0.120	-34.46	2.82
5	1.617	-1.750	0.919	-36.42	2.56
6	0.888	-1.297	0.083	-46.37	1.58
7	-0.182	-0.587	0.529	-49.98	0.81
8	0.170	-0.669	-0.387	-53.93	0.82

TABLE III

SDU APPLIED TO 2PTC.

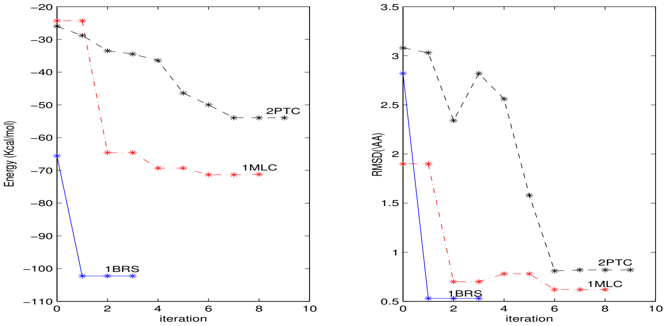


Fig. 6. The evolution of SDU for the 1BRS, 1MLC, and 2PTC complexes.

The (free) energy functional we minimize includes van der Waals interactions (ΔE_{vdW}), the desolvation energy (ΔE_{des}) and the electrostatic energy (ΔE_{elec}). We use the software package CHARMM [24] to evaluate the free energy and to perform local minimization. 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.

In Table II (right) and Table III we report results for the complexes 1BRS (barnase/barstar), 1MLC (a 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 in the tables), 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). As discussed earlier, the key measure of success is the proximity of $D(\boldsymbol{\phi}^G)$ to zero.

The evolution of SDU in all three complexes considered is depicted in Fig. 6. A better illustration is given in Fig. 7 for the 1BRS complex. In both figures we plot the fixed receptor. We generated 28 random positions of the ligand in the initial search region and in the left figure we plot a sphere at the center of the ligand for each one if these 28 ligand positions. Spheres in lighter shades of gray correspond to lower energy conformations. The position of the native ligand is shown as a dotted sphere and pointed by an arrow. After performing the first iteration of SDU we sample a new set of ligand positions using the procedure in SDU's Step 5 (b) (cf. Fig. 3). The centers of these ligands are depicted in the right figure. The corresponding structures are both very close to the native one and have low energies.

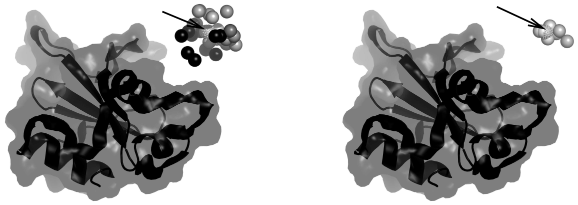


Fig. 7. Illustration of the 1BRS docking procedure.

We conclude by noting that the starting point for our medium-range optimization is the cluster of best conformations produced by the techniques in [10]. As seen in Fig. 6, these produce conformations within 3-5 Å from the native one. Our method is able to reduce this distance to less than 1 Å in all three complexes considered. In most cases this is considered a very successful identification of the native conformation as the (approximate) energy models used do not provide a much greater degree of accuracy.

VIII. CONCLUSIONS

We introduced a new method for minimizing free-energy functions appearing in molecular docking and other important problems in computational structural biology. These energy functions are notoriously hard to minimize as they consist of terms acting in disparate space-scales and have a huge number of local minima. Yet, in certain areas 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 performs better than an alternative (CGU) method. Finally, we applied our algorithm to some 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 state-of-the-art methods produce.

ACKNOWLEDGMENTS

We thank Carlos Camacho for many useful discussions.

REFERENCES

[1] R. M. Murray, Z. Li, and S. S. Sastry, *A Mathematical Introduction to Robotic Manipulation*. CRC Press, 1994.

[2] S. Gwak, J. Kim, and F. C. Park, "Numerical optimization on the euclidean group with applications to camera calibration," *TRA*, vol. 19, no. 1, pp. 65–74, 2003.

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

[4] C.-J. Tsai, S. Kumar, B. Ma, and R. Nussinov, "Folding funnels, binding funnels, and protein function," *Protein Sci.*, vol. 8, pp. 1981–1990, 1999.

[5] C. A. Floudas, J. L. Klepeis, and P. M. Pardalos, “Global optimization approaches in protein folding and peptide docking,” in *DIMACS Series in Discrete Mathematics and Theoretical Computer Science* (F. Roberts, ed.), vol. 47, pp. 141–171, 1999.

[6] J. J. Gray, S. Moughon, C. Wang, O. Schueler-Furman, B. Kuhlman, C. A. Rohl, and D. Baker, “Protein-protein docking with simultaneous optimization of rigid-body displacement and side-chain conformations,” *J. Molecular Biology*, vol. 331, pp. 281–299, 2003.

[7] 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.

[8] C. Camacho and S. Vajda, “Protein docking along smooth association pathways,” *Proc. Natl. Acad. Sci. USA*, vol. 98, pp. 10636–10641, 2001.

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

[10] S. R. Comeau, D. Gatchell, S. Vajda, and C. J. Camacho, “Cluspro: An automated docking and discrimination method for the prediction of protein complexes,” *Bioinformatics*, vol. 20, pp. 45–50, 2004.

[11] I. Halperin, B. Ma, H. Wolfson, and R. Nussinov, “Principles of docking: An overview of search algorithms and a guide to scoring functions,” *Proteins*, vol. 47, pp. 409–443, 2002.

[12] C. Zhang, G. Vasmatzis, and J. Cornette, “Determination of atomic desolvation energies from the structures of crystallized proteins,” *J Mol Biol*, vol. 267, pp. 707–726, 1997.

[13] “General Algebraic Modeling System (GAMS).” http://www.gams.com/.

[14] M. Tawarmalani and N. V. Sahinidis, “Global optimization of mixed-integer nonlinear programs: A theoretical and practical study,” *Mathematical Programming*, vol. 99, no. 3, pp. 563–591, 2004.

[15] J. L. Klepeis and C. A. Floudas, “Ab initio tertiary structure predicition of proteins,” *J. of Global Optimization*, vol. 25, pp. 113–140, 2003.

[16] J. L. Klepeis, M. J. Pieja, and C. A. Floudas, “A new class of hybrid global optimization algorithms for peptide structure predicition: integrated hybrids,” *Computer Physics Communications*, vol. 151, pp. 121–140, 2003.

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

[18] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge University Press, 2004.

[19] K. Fujisawa, M. Kojima, K. Nakata, and M. Yamashita, *SDPA (SemiDefinite Programming Algorithm) User’s Manual Version 6.00*. http://grid.r.dendai.ac.jp/sdpa/, 2002.

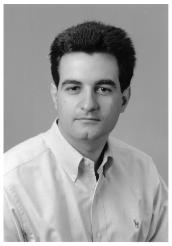
[20] J. Rosen and R. Marcia, “Convex quadratic approximation,” *Computational Optimization and Applications*, vol. 28, pp. 173–184, 2004.

[21] R. Hettich and K. O. Kortanek, “Semi-infinite programming: theory, methods and applications,” *SIAM Review*, vol. 35, no. 3, pp. 380–429, 1993.

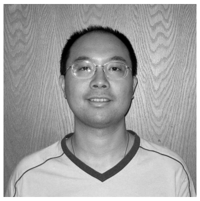
[22] D. Bertsekas, *Nonlinear Programming*. Belmont, MA: Athena Scientific, 2nd ed., 1999.

[23] L. Ingber, “Adaptive Simulated Annealing (ASA).” Global optimization C-code, Caltech Alumni Association, Pasadena, CA, 1993. Available: http://www.ingber.com/.

[24] B. Brooks, R. Bruccoleri, B. Olafson, D. States, S. Swaminathan, and K. M., “CHARMM: A program for macromolecular energy, minimization, and dynamics calculations,” *J Comput Chem*, vol. 4, pp. 187–217, 1983.



Ioannis Ch. Paschalidis (M’96, SM’06) received the Diploma in ECE from the National Technical University of Athens, Greece, in 1991, and the S.M. and Ph.D. degrees in EECS from the Massachusetts Institute of Technology (MIT), Cambridge, Massachusetts, in 1993 and 1996, respectively. He joined Boston University in 1996 where he is Associate Professor of Manufacturing Engineering and the Co-Director of the Center for Information and Systems Engineering. His current research interests lie in the fields of systems and control, optimization, networking, operations research, and computational biology. He is an associate editor of Operations Research Letters, and the IEEE Transactions on Automatic Control.



Yang Shen (S’06) received his B.E. in automatic control from the University of Science and Technology of China (USTC), Hefei, China, in 2002. He is currently a Ph.D. candidate in systems engineering at Boston University, Boston, MA, and affiliated with the Center for Information and Systems Engineering (CISE) and the Structural Bioinformatics Laboratory. His research interests include modeling, simulation and optimization of complex systems, especially biological systems of protein-protein or protein-ligand interactions.

Pirooz Vakili received the Ph.D. degree in Mathematics from Harvard University in 1989. He is currently an Associate Professor in the Department of Manufacturing Engineering at Boston University. His current research interests include efficient Monte Carlo simulation, optimization, computational finance, and bioinformatics.

Sandor Vajda (M’86) received the Diploma in EE from the Gubkin Institute, Russia, in 1972, the Diploma in applied mathematics from the Eotvos Lorand University, Hungary, in 1980, and the Ph.D. degree in chemistry from the Hungarian Academy of Science in 1983. In 1990 he joined Boston University, where he is Professor of Biomedical Engineering and the co-director of the Biomolecular Engineering Research Center. His current research interests are in method development for modeling of biological macromolecules, with emphasis on molecular interactions. In 2004 he and his students founded SolMap Pharmaceuticals, Cambridge, MA, a start-up company focusing on fragment-based drug design.