Published in final edited form as:
Proc IEEE Conf Decis Control. 2012 ; : 2310–2315.

A Message Passing Approach to Side Chain Positioning with Applications in Protein Docking Refinement *

Mohammad Moghadasi†, **Dima Kozakov**‡, **Artem B. Mamonov**‡, **Pirooz Vakili**¶, **Sandor Vajda**‡, and **Ioannis Ch. Paschalidis**§

†Division of Systems Eng., Boston University, mohamad@bu.edu

‡Dept. of Biomedical Eng., Boston University, {midas,amamonov,vajda}@bu.edu

¶Dept. of Mechanical Eng. and Division of Systems Eng., Boston University, vakili@bu.edu

Abstract

We introduce a message-passing algorithm to solve the *Side Chain Positioning (SCP)* problem. SCP is a crucial component of *protein docking refinement*, which is a key step of an important class of problems in computational structural biology called *protein docking*. We model SCP as a combinatorial optimization problem and formulate it as a *Maximum Weighted Independent Set (MWIS)* problem. We then employ a modified and convergent belief-propagation algorithm to solve a relaxation of MWIS and develop randomized estimation heuristics that use the relaxed solution to obtain an effective MWIS feasible solution. Using a benchmark set of protein complexes we demonstrate that our approach leads to more accurate docking predictions compared to a baseline algorithm that does not solve the SCP.

I. INTRODUCTION

Proteins interact with each other or with other biochemical compounds to carry out some cellular functions such as cell signaling, ligand binding, metabolic control and gene regulation. At each interaction, at least two chemical entities are involved: a *receptor* molecule and a *ligand* molecule that binds to the receptor. Based on thermodynamics principles, proteins bind to each other in a way that minimizes the Gibbs free energy of the complex. Thus, when a ligand binds to a receptor, the atomic coordinates of the whole complex change such that the overall free energy of the complex attains its minimum value. The prediction of the 3-dimensional (3-D) structure of a stable receptor-ligand complex is known as the *protein docking problem*. Experimental techniques such as X-ray crystallography and Nuclear Magnetic Resonance (NMR) can be used to observe such structures, but they are expensive, time-consuming and not universally applicable. Hence, using computational methods to solve such problems has drawn a lot of attention.

Our protein docking procedure, first, samples billions of docked receptor-ligand conformations using either *ClusPro 1.0* or *ClusPro 2.0* servers [1], [2]. Next, the conformations retained are clustered [3], and for each cluster, we find a conformation, called *cluster center*. For each cluster, we pick the conformations whose Root Mean Squared Deviation (RMSD) values of their atomic coordinates to the cluster center are less than 12 Å

*Research partially supported by the NIH/NIGMS under grant 1-R01-GM093147-01, by the NSF under grant EFRI-0735974, by the DOE under grant DE-FG52-06NA27490, by the ARO under grant W911NF-11-1-0227, and by the ONR under grant N00014-10-1-0952.

§Corresponding author. Dept. of Electrical & Computer Eng., and Division of Systems Eng., Boston University, 8 Mary's St., Boston, MA 02215, yannisp@bu.edu.

for further processing. The retained top conformations of each cluster are then refined using a refinement technique such as Semi-Definite programming based Underestimation (SDU) [4], [5] or Monte Carlo Minimization (MCM) [6], [7]. The main goal of this refinement is to locate the global energy minimum within the regions of conformational space defined by the different clusters. The output of the refinement protocol is a set of conformations which are aimed to be the "good" predictions of the native structure.

Side Chain Positioning (SCP) is one of the key components of these refinement algorithms. SCP enables the refinement algorithms to take advantage of the protein side chains' flexibility to predict the lowest energy conformation. In this paper, we model SCP as a combinatorial optimization problem and propose a distributed message-passing algorithm to solve it. Our results show that the proposed algorithm improves the output of the refinement algorithms noticeably, yielding conformations with lower energy and lower RMSD from the true native conformation.

The remainder of the paper is organized as follows. Sec. II provides a basic introduction to the SCP problem and formulates it as a quadratic integer programming problem. Sec. III models SCP as a combinatorial optimization problem and presents our distributed message-passing algorithm. In Sec. IV, we test the algorithm against a protein docking benchmark set. Concluding remarks are in Section V.

II. SIDE CHAIN POSITIONING PROBLEM

A. Preliminaries and Background

Each protein molecule is composed of one or more peptides. Each peptide is a sequence of unbranched chains of recurring building blocks called *amino acid residues*. There are 20 different types of amino acids and the number of residues in a peptide varies from tens to hundreds. Each residue is a molecule composed of two parts, a backbone part and a side chain.

When a ligand binds to a receptor, some conformational changes of the form of slight displacements of the protein atoms are often observed at the interfacial residues that decrease the energy of the complex. Ideally, one would like to predict the lowest energy conformation of the receptor and ligand backbones and side chains. However, due to the high complexity of modeling the backbone movement and its typically rigid structure, most of the classical models keep the backbone fixed, while allowing the side chains to freely move in space [8], [9]. Thus, SCP can be defined as a problem which takes fixed receptor and ligand backbones and predicts the side chain conformations that minimize the overall energy of the complex.

To model the flexibility of the side chains upon binding, let us first introduce the concept of *rotamers*. Although the side chains may be able to move freely in space, they tend to occupy only a finite number of more probable conformations in actual protein structures called rotamers. The detailed information of all the rotamers of all different types of residues is collected into massive data sets called rotamer libraries. For this study, we used the *"2010 Smooth Backbone-Dependent Rotamer Library"* [10].

It follows that SCP can now be rewritten as the following combinatorial optimization problem: given a receptor-ligand complex with fixed backbones and flexible side chains, the goal is to choose one rotamer for each side chain such that the overall energy of the complex is minimized.

B. SCP as a Quadratic Integer Programming

In this section, we present our SCP model and formulation. We adapt a framework similar to [8] derived for protein folding applications.

Geometrically, the space of rigid-body motions in 3-D space is described in terms of the members of the Special Euclidean group $SE(3)$ expressing rigid body orientation and position. An element of $SE(3)$ is of the form $\xi = (\rho, \mathbf{R})$ where $\rho \in \mathbb{R}^3$ describes the coordinates of the origin of a body with respect to an inertial frame reference and **R** is a 3 × 3 real matrix denoting the orientation of the body with respect to the inertial frame reference.

Let us denote by $\xi \in SE(3)$ the position and orientation of the ligand with respect to the receptor. Our SCP algorithm will select rotamers for all residues in the interface between the receptor and the ligand. To focus on a single receptor-ligand conformation we fix ξ, and for ease of notation we will suppress the dependence on ξ of the various quantities we define in the sequel. Define $\mathscr{I}$ as the set of all receptor and ligand residues in the interface. The interface of a receptor-ligand complex is the set of all residues in each molecule of the complex whose C_α atom is within a small distance (10 Å in our work) from a C_α atom located on the partner molecule. Let $\mathscr{U}_i$ denote the set of rotamers for each residue $i \in \mathscr{I}$ and denote by $|\mathscr{I}|$ the cardinality of $\mathscr{I}$.

Consider a feasible solution to the SCP problem, which is a set of rotamers that includes exactly one rotamer i_r from each interface residue $i \in \mathscr{I}$. The overall energy E associated with this set of rotamers can be decomposed as follows:

$$E = E_0 + \sum_{i \in \mathscr{I}} E(i_r) + \sum_{i,j \in \mathscr{I}: i<j} E(i_r, j_s), \quad (1)$$

where E_0 is self-energy of the two backbones, $E(i_r)$ is the energy of the interaction between rotamer i_r from residue i and the two backbones including the self-energy of the rotamer i_r, and $E(i_r, j_s)$ is the pairwise interaction energy between the selected rotamers i_r and j_s, which respectively correspond to the two different residues i and j.

Let us construct an undirected $|\mathscr{I}|$-partite graph $\tilde{\mathscr{G}} = (\tilde{\mathscr{V}}, \tilde{\varepsilon})$ with node set $\tilde{\mathscr{V}} = \tilde{\mathscr{V}}_1 \cup \cdots \cup \tilde{\mathscr{V}}_{|\mathscr{I}|}$, in which each $\tilde{\mathscr{V}}_i, i = 1, \cdots, |\mathscr{I}|$, corresponds to the residue $i \in \mathscr{I}$, and includes a node u for each rotamer $i_r \in \mathscr{U}_i$ with a weight equal to $E_{uu} = E(i_r)$. For every pair of nodes $u \in \tilde{\mathscr{V}}_i$ and $v \in \tilde{\mathscr{V}}_j, i, j = 1, \cdots, |\mathscr{I}|$, we draw an edge with a weight equal to $E_{uv} = E(i_r, j_s)$, where i_r and j_s are the rotamers corresponding to u and v, respectively. The SCP problem is equivalent to selecting one node per $\tilde{\mathscr{V}}_i$ in order to minimize the total weight of the resulting subgraph and can be formulated as the following *Quadratic Integer Programming (QIP)* problem:

$$\begin{array}{lll} \min & \Sigma_{u,v \in \tilde{\mathscr{V}}} E_{uv} y_u y_v & \\ s.t. & \Sigma_{u \in \tilde{\mathscr{V}}_i} y_u = 1, & i = 1, \cdots, |\mathscr{I}|, \\ & y_u \in \{0, 1\}, & u \in \tilde{\mathscr{V}}, \end{array} \quad (2)$$

where the decision variables y_u are the indicator variables selecting the rotamer corresponding to node u.

QIP problems are in the class of NP-hard problems. By merging the variables y_u and y_v into y_{uv}, and at the expense of a larger number of decision variables, one can turn (2) into an *Integer Linear Programming (ILP)* problem. ILPs are also NP-hard, yet there exist good solvers that can solve relatively small instances [9]. A *Semi-Definite Programming (SDP)* relaxation of (2) was developed in [8] and applied to protein folding. In general, SDP relaxations of combinatorial optimization problems have been shown effective in a class of hard problems. Another approach based on a *Second Order Cone Programming (SOCP)* relaxation of (2) was developed in [11]. The main drawback of these (SDP or SOCP) relaxations is that one ends up with a centralized algorithm, and can not take advantage of task parallelism. In this paper, we propose a fully distributed algorithm which is computationally more efficient than centralized algorithms.

III. METHODOLOGY

In this section, we first model SCP as a *Maximum Weighted Independent Set (MWIS)* problem and then develop our distributed algorithm. The algorithm is similar to one from our earlier work in [12], [13] which was developed in a different application context (wireless networks). We introduce some modifications to suit the problem at hand.

A. Maximum Weighted Independent Set Formulation

1) Maximum Weighted Independent Set Problem—MWIS is a well-studied NP-hard combinatorial optimization problem. The goal of the problem is to find the heaviest *independent set* of nodes in a given undirected graph $\mathcal{G}=(\mathcal{V},\varepsilon)$ with non-negative weights on the nodes. A set is called independent if no two nodes in it are adjacent. We can formulate MWIS as the following *Integer Programming (IP)* problem:

$$\begin{array}{lll} max & \Sigma_{i=1}^{N} w_i x_i & \\ s.t. & x_i + x_j \le 1, & \forall (i,j) \in \varepsilon, \\ & x_i \in \{0,1\}, & i=1,\cdots,N, \end{array} \qquad (3)$$

where $N=|\mathcal{V}|, w_i \ge 0$ is the weight of node *i*, and x_i is the indicator variable of selecting node *i*.

2) SCP as a MWIS Problem—To re-formulate SCP as a MWIS problem, we construct a new undirected graph $\mathcal{G}=(\mathcal{V},\varepsilon)$ from $\tilde{\mathcal{G}}=\left(\tilde{\mathcal{V}},\tilde{\varepsilon}\right)$ as we next describe. $\mathcal{V}$ consists of nodes v_{rs} for pairs of rotamers $r \in \mathcal{U}_i$ and $s \in \mathcal{U}_j$, $i \neq j$, and nodes v_{rr} for each rotamer $r \in \mathcal{U}_i$. We associate with v_{rs} nodes an energy equal to $E(r, s)$; this is the E_{rs} edge weight in $\tilde{\mathcal{G}}$ or, equivalently, the pairwise energy between the two residues with selected rotamers *r* and *s*, respectively. We also associate with v_{rr} nodes an energy equal to $E(r)$, i.e., the residue-backbone and residue self-energy corresponding to rotamer *r*. We then assign node weights so that nodes with lower associated energy values have larger weights. To that end, we define a parameter *M* which is greater than all the energy values, and we set the node weights as follows: for nodes v_{rs}, $w_{rs} = M - E_{rs} > 0$, and for v_{rr} nodes, $w_{rr} = M - E_{rr} > 0$.

Next we introduce the edges of $\mathcal{G}$ that are designed to identify conflicts between rotamers. Specifically, an edge $(v_{rs}, v_{tw}) \in \varepsilon$ if the choice of rotamers *r, s, t, w* has a conflict. This conflict arises if there exist two different rotamers from the same residue in the set of $\{r, s, t, w\}$. (It is not possible to have more than two conflicting rotamers in this set due to the definition of the nodes). The construction of $\mathcal{G}$ guarantees that for each residue one and only one rotamer can be selected.

From the construction of $\mathcal{G}$ and the discussion in this section the following result is obtained.

Theorem III.1 *Consider a MWIS of the graph* $\mathscr{G}$ with total weight W. Then the rotamers associated with the nodes in the MWIS form an optimal solution to the SCP problem with associated minimal energy equal to M − W.

B. A Distributed Algorithm for MWIS

Following [12] we will first solve the *Linear Programming (LP)* relaxation of (3) and then use the optimal solution of the relaxation to estimate feasible ILP solutions. The LP relaxation of (3) is derived from (3) by replacing the last (integrality) constraint with $0 \leq x_i \leq 1$ for all $i = 1, \dots, N$. Such a problem can be solved efficiently by LP solvers, but in a centralized fashion. Here however, we employ a fully distributed approach from [12] that uses only local information at the graph nodes. The first phase consists of a *coloring* and a *gradient projection* procedure, which can be performed in parallel. The second phase takes the outputs of the first phase as its input, and estimates a feasible solution to the MWIS problem.

The distributed algorithm proposed in [12] uses a coloring method from [14] which is developed for a general unweighted graph. This coloring algorithm works well with the whole protocol if the GP phase detects enough number of nodes from the solution set and equivalently outputs enough number of x_i's with assigned values close to 1. In SCP application, due to the specific graph structure that SCP modeling imposes, the GP phase is not as successful as in the sensor network application reported in [12]. Thus, the coloring phase plays a more significant role in finding the optimal solution in the SCP problem. We modified the coloring algorithm of [14] with the some randomized heuristics in order to enable the coloring phase to prioritize the nodes of graph G based on their weights.

1) Phase I.1: Coloring—

The objective of the *Coloring* procedure is to color all nodes of $\mathscr{G}$ using the minimum possible number of colors such that no two adjacent nodes share the same color. In this work, we use the self-stabilizing algorithm proposed in [14] which can be implemented in a distributed fashion. This algorithm needs to take one node as the special node, i.e., the *root*, and to inform each node whether it is the root or not. The root is the first node that the algorithm colors. Graph $\mathscr{G}$ can be colored with at most $2D$ colors [14], where D is the degree of $\mathscr{G}$. This procedure can be done in a number of steps which is polynomial in size of $\mathscr{G}$ [14]. The color assigned to node i is represented by an integer $c_i \in \{1, \dots, 2D\}$. Thus, the output of the coloring procedure is of the form of a vector $\mathbf{c} = \{c_1, \dots, c_N\}$.

If node i is colored before node j, then $c_i \leq c_j$ and the priority of node i is more than node j. These relational priorities are consequential in the estimation phase, and a good choice of a coloring policy can improve the overall performance of the protocol. In this work we select the node with the highest weight as the root, and we find this node in a distributed fashion as described in [12].

The algorithm in [14] is designed for a general unweighted graph and does not use the weights of the nodes in $\mathscr{G}$. We modify this algorithm with the following randomized heuristic in order to improve the quality of the MWIS feasible solution our two-phase algorithm obtains. Let $\mathscr{U}$ be the set of uncolored nodes of $\mathscr{G}$. Select the nodes in $\mathscr{U}$ which account for the top 50% of the overall weight of $\mathscr{U}$ and let them form set $\widehat{\mathscr{U}} \subset \mathscr{U}$. For each node $i \in \widehat{\mathscr{U}}$, compute $S_i = \left[w_i - \Sigma_{j\in\mathscr{N}_i} w_j\right]$. Then, shifting those values to ensure they are non-negative, and normalizing by some normalization factor C we obtain $\widehat{S}_i = \left(S_i - \min_{j\in\widehat{\mathscr{U}}} S_j\right)/C$. Now, instead of the general approach applied in [14] to choose the next node to color, we select one of the nodes of $\widehat{\mathscr{U}}$ with probability $p_i = \widehat{S}_i$. This heuristic essentially filters out the low weight nodes at each run and increases the priority of heavier nodes by coloring them earlier.

2) Phase I.2: Gradient Projection—The *Gradient Projection (GP)* procedure solves the LP relaxation of (3) and its dual concurrently. The algorithm starts by adding a logarithmic barrier function to the objective:

$$\begin{array}{lll} max & \Sigma_{i=1}^{N} w_i x_i + \epsilon & \Sigma_{i=1}^{N} \log \quad x_i \\ s.t. & x_i + x_j \le 1, & \forall (i,j) \in \varepsilon, \\ & 0 \le x_i \le 1, & i = 1, \cdots, N, \end{array} \quad (4)$$

where ε is a positive constant. Viewing (4) as the primal problem, each primal variable is associated with a node in $\mathscr{V}$. Let $\boldsymbol{\theta} = \{\theta_{ij}; (i, j) \in \varepsilon\}$ denote the dual variables of the first set of constraints in (4). Note that θ_{ij} and θ_{ji} are identical due to the undirected structure of $\mathscr{G}$. Therefore, we can rewrite $\boldsymbol{\theta} = \{\theta_{ij}; (i, j) \in \varepsilon, i < j\}$ so that each edge of $\mathscr{G}$ is associated with one and only one dual variable. As shown in [12], the dual problem of (4) has the following form:

$$\begin{array}{ll} \min & q(\theta) \\ s.t. & \theta_{ij} \ge 0, \quad \forall (i,j) \in \varepsilon, \end{array} \quad (5)$$

with

$$q(\theta) = \sum_{i=0}^{N} \max_{0 \le x \le 1} g_i(x) + \sum_{(i,j) \in \varepsilon} \theta_{i,j}, \quad (6)$$

where $g_i(x) \equiv \left(w_i - \Sigma_{j \in \mathscr{N}_i} \theta_{ij}\right) x + \epsilon \quad \log \quad x$, and its unique maximizer $x_i(\boldsymbol{\theta}) \in [0, 1]$ is given by:

$$x_i(\theta) = \begin{cases} \frac{\varepsilon}{\Sigma_{j \in \mathscr{N}_i} \theta_{ij} - w_i}, & \text{if} \quad \Sigma_{j \in \mathscr{N}_i} \theta_{ij} \ge w_i + \varepsilon, \\ 1, & \text{otherwise.} \end{cases} \quad (7)$$

It is not hard to verify that for any $(i, j) \in \varepsilon$:

$$\frac{\partial q(\theta)}{\partial \theta_{ij}} = 1 - x_i(\theta) - x_j(\theta), \quad (8)$$

and $q(\boldsymbol{\theta})$ is continuously differentiable. Employing a gradient projection method to solve the dual we obtain the algorithm shown in Fig. 1, where $[\cdot]_+ = \max\{\cdot, 0\}$. At each iteration n of this algorithm, $x^{(n)}$ and $\boldsymbol{\theta}^{(n)}$ denote the values of the vectors x and $\boldsymbol{\theta}$, γ, and is a pre-specified step-size.

Theorem III.2 guarantees the convergence of the GP algorithm; the proof is in [12].

Theorem III.2 *For any* γ *such that* $0 < \gamma < \frac{\epsilon}{D\sqrt{|\varepsilon|}}$, the GP algorithm converges to the optimal primal-dual pair (x^*, $\boldsymbol{\theta}^*$) *that solves problems (4) and (5).*

The algorithm in Fig. 1 requires a stopping criterion; one possibility is to stop whenever $|\theta_{ij}^{(n)} - \theta_{ij}^{(n-1)}| \le \delta$ for all $(i, j) \in \varepsilon$. Choosing an appropriate ε is another practical issue. In general, it is not easy to guess a good particular ε before running the algorithm. Instead, we start with some ε to run the algorithm until its convergence, and then reduce ε and repeat the process until two consecutive runs yield θ's that are close enough.

3) Phase II: Estimation—This phase constructs a feasible solution to (3). By solving (4) using the GP procedure with the diminishing ε policy, we obtain an optimal solution $x_i^* \in [0, 1]$ for all nodes $i = 1, \dots, N$ which can be fractional. Yet, all integer x_i^* obtained are in fact "correct" as the following Lemma from [12] establishes. Then, it suffices to "round" just the fractional x_i^*.

Lemma III.3 For any $i \in \mathcal{V}$ where $x_i^* \in \{0, 1\}$, there is always an optimal solution $\tilde{x}$ *to problem (3) such that* $\tilde{x}_i = x_i^*$.

This can be done with the algorithm shown in Fig. 2, where $\tilde{x}$ represents the vector of estimated MWIS decision variables, and $\mathcal{X}$ stands for the "undetermined" state of a decision variable $\widehat{x_i}$.

This algorithm first assigns $\tilde{x}_i = x_i^*$ for any node i whose x_i^* is 0 or 1. Then, for any remaining node i, it iterates the following procedure: it first checks the neighbors of node i, if there exist a node $j \in \mathcal{N}_i$ whose assigned value $\widehat{x_j}$ is equal to 1, the algorithm assigns $\widehat{x_i} = 0$ in favor of feasibility of the solution. If there is no such node j, the algorithm compares c_i, the color of node i, with all its neighbors. If node i has the highest priority compared to its neighbors, it sets $\widehat{x_i} = 1$; otherwise it does nothing and continues to the next iteration.

At each iteration of the algorithm depicted in Fig. 2 at least one new node is colored. Thus, the algorithm takes at most $2D$ iterations, since the most number of colors needed to color a graph is $2D$ according to the coloring method we used. We summarize this discussion in the following theorem.

Theorem III.4 The estimation algorithm in Fig. 2 outputs a feasible solution for problem (3).

So far, we formulated the SCP problem (2) as a MWIS problem (3) and employed a variant of the distributed algorithm proposed in [12] to solve it. In particular, we introduced some randomized coloring heuristics that are beneficial in our side chain positioning application. These heuristics help the coloring phase to assign more priority to the nodes whose weights are greater than the summation of all the weights of their neighbors. Thus, if GP fails in deciding which node to select amongst a set of connected nodes, the coloring will help the protocol to pick the nodes with higher priority first. Consider a simple case when GP can not decide which node to choose between two adjacent nodes, i.e. GP assigns $x_i = 0.5$ and $x_j = 0.5$ when $(i, j) \in \varepsilon$, then the estimation phase will assign $x_i = 1$ and $x_j = 0$ if $c_i < c_j$.

IV. COMPUTATIONAL RESULTS

We tested SCP on a benchmark set consisting of 15 enzyme inhibitor protein complexes listed in the first column of Table I. To generate the input data for each complex, as mentioned in the Introduction, our docking procedure first samples billions of docked receptor-ligand conformations, then clusters and filters them. The retained thousands of conformations are used as the input data for the *refinement* stage.

The refinement algorithm we use is based on an MCM approach. For each conformation, our algorithm works by iteratively proposing a sequence of rotational and translational motions of the ligand while fixing the receptor. We denote by $\xi \in SE(3)$ the initial position and orientation of the ligand with respect to the receptor and run 50 MCM iterations. The final position, denoted by $\widehat{\xi}$ is our prediction corresponding to that specific initial conformation.

At each MCM iteration, the following rigid-body motions are applied to the ligand: first the ligand position and orientation with respect to the receptor are randomly perturbed to obtain $\tilde{\xi}$, and then a rigid-body minimization algorithm starts from $\tilde{\xi}$ and locally minimizes the energy of the complex over ligand positions in $SE(39)$ to obtain $\tilde{\xi}^*$. We then compare the total energies of the conformational states at ξ and $\tilde{\xi}^*$ and decide either to accept or reject this move based on Metropolis criterion. In case of acceptance, $\tilde{\xi}^*$ will be used as an input to the next MCM iteration.

Based on this criterion the probability of accepting an attempted conformation is:

$$P_{accept}=\min\left(1, e^{-\Delta E/T}\right), \quad (9)$$

where ΔE is the difference between the total energy of the resulting conformation $\tilde{\xi}^*$ and the energy of the initial conformation ξ, and T is the absolute temperature of the system.

In addition to running the random perturbation and rigid-body minimization steps that exert rigid-body motions to the ligand, we can also use SCP as another important step of MCM iterations to position the side chains of the interfacial residues of both receptor and ligand in order to reduce the total energy of the complex. In our procedure, we run the SCP algorithm right after the completion of the random perturbation and before running the rigid-body minimization.

Our tests on 15 protein complexes indicate the effectiveness of SCP for the protein docking refinement procedures. For each protein complex, we compare three different conformations: (*i*) the initial conformations before refinement, (*ii*) the refined conformations excluding SCP from the MCM iterations, and (*iii*) the refined conformations including SCP as a step in the MCM iterations. For each one of these conformations we calculate its RMSD from the native structure and report in Table I the number of conformations whose RMSD to the native structure is less than 5 Å (in columns 3, 4, and 5 for conformations of type (i), (ii), and (iii), respectively). This number corresponds to the number of “good” predictions our protein docking procedure can make for a specific protein complex. A prediction is considered “good” when the RMSD of the predicted structure is below 5 Å from its native structure.

Our results show that in 12 (out of the 15) protein complexes, using our SCP algorithm increases the number of “good” predictions and improves the overall refinement performance. These 12 complexes are marked in the last column of Table I. The remaining 3 complexes whose number of “good” predictions decreases by adding SCP to the protocol are 1BVN, 1NW9 and 1PPE. In 1BVN, even though the number of good predictions decreases by adding SCP, the overall performance of the protocol is still acceptable since the algorithm has predicted 201 “good” predictions. In 1NW9 and 1PPE the inputs to the refinement procedure do not have an adequate number of near-native structures to initiate good starting conformations. 1NW9 has only 39 and 1PPE has only 1 near-native structures, which means that the ligand and the receptor have not bound to each other well and using SCP may not help. However, there are several complexes with similar low initial near-native concentrations for which SCP is pretty effective; these include 1K74, 1N8O, 1NW9, 1TMQ and 7CEI.

Another criterion we use to evaluate the effect of SCP on protein docking refinement is to investigate how an *RMSD-score plot* changes by applying SCP. Fig. 3 depicts such plots for two protein complexes: 1R0R and 7CEI. Each plot consists of numerous points whose exact number can be found in the 2nd column of Table I. Each point represents a single conformation refined by either the MCM or the MCM+SCP method. The *x*-coordinate of each point corresponds to the RMSD between the associated conformation and the native structure. The *y*-coordinate indicates the overall energy of the conformation after refinement. In protein docking refinement, we aim to achieve an RMSD-score plot which exhibits a funnel leading to low energy and low RMSD structures. In presence of such funnels, as we approach to the native structure (as RMSD decreases), the overall energy value decreases. This correlation is inline with the fact that the native structure attains the lowest overall energy value compared to other possible conformations. Our results in Fig. 3 show that if we run MCM refinement without SCP, the RMSD-score plots usually lack such funnels. However, if we add our SCP algorithm to the MCM steps, the resulting RMSD-score plots reveal the presence of such a funnel. Furthermore, compared to the baseline algorithm, MCM+SCP achieves the same trend in covering the whole conformational space and gives rise to the same number of clusters while reveals a funnel-like behavior leading to low energy and low RMSD structures.

V. CONCLUSIONS

We formulated the side chain positioning problem as a maximum weighted independent set problem and devised a message-passing algorithm to solve it. Compared to alternative algorithms, the main advantage of our approach is that it takes advantage of task parallelism when solving the optimization problem. In the context of side chain positioning application, the parallel approach is of great importance due to the large problem instances one has to tackle.

One of the most significant applications of side chain positioning is in protein docking refinement procedures. To verify the effectiveness of our algorithm, we have implemented a refinement protocol based on a Monte Carlo Minimization approach, and compared the overall refinement results in the absence and in the presence of side chain positioning in each Monte Carlo step. Our results show that the latter improves the overall performance of protein docking refinement procedures in two different ways: increasing the number of near native predictions and revealing a funnel-like behavior in RMSD-score plots that leads to low-energy near-native structures.

References

[1]. Comeau SR, Gatchell D, Vajda S, Camacho CJ. Cluspro: An automated docking and discrimination method for the prediction of protein complexes. Bioinformatics. 2004; 20:45–50. [PubMed: 14693807]

[2]. Kozakov D, Brenke R, Comeau SR, Vajda S. PIPER: An FFT-based protein docking program with pairwise potentials. Proteins. 2006; 65:392–406. [PubMed: 16933295]

[3]. Kozakov D, Clodfelter K, Vajda S, Camacho C. Optimal clustering for detecting near-native conformations in protein docking. Biophysical Journal. 2005; 89(2):867–875. [PubMed: 15908573]

[4]. Paschalidis IC, Shen Y, Vakili P, Vajda S. SDU: A semidefinite programming-based underestimation method for stochastic global optimization in protein docking. IEEE Trans. Automat. Contr. 2007; 52(4):664–676. [PubMed: 19759849]

[5]. Shen Y, Paschalidis IC, Vakili P, Vajda S. Protein Docking by the Underestimation of Free Energy Funnels in the Space of Encounter Complexes. PLoS Computational Biology. 2008; 4(10)

[6]. Kozakov D, Schueler-Furman O, Vajda S. Discrimination of near-native structures in protein–protein docking by testing the stability of local minima. Proteins: Structure, Function, and Bioinformatics. 2008; 72(3):993–1004.

[7]. Gray JJ, Moughon S, Wang C, Schueler-Furman O, Kuhlman B, Rohl CA, Baker D. Protein-protein docking with simultaneous optimization of rigid-body displacement and side-chain conformations. J. Molecular Biology. 2003; 331:281–299.

[8]. Chazelle B, Kingsford C, Singh M. A semidefinite programming approach to side chain positioning with new rounding strategies. INFORMS Journal on Computing. 2004; 16(4):380–392.

[9]. Kingsford C, Chazelle B, Singh M. Solving and analyzing sidechain positioning problems using linear and integer programming. Bioinformatics. 2005; 21(7):1028–1036. [PubMed: 15546935]

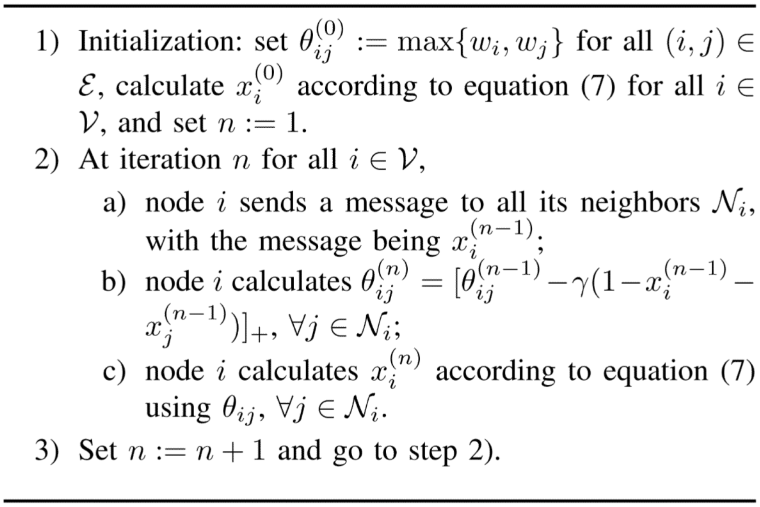
[10]. Shapovalov M, Dunbrack R Jr. A smoothed backbone-dependent rotamer library for proteins derived from adaptive kernel density estimates and regressions. Structure. 2011; 19(6):844–858. [PubMed: 21645855]

[11]. Kumar, M.; Torr, P.; Zisserman, A. Solving markov random fields using second order cone programming relaxations; Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on; IEEE. 2006; p. 1045-1052.

[12]. Paschalidis, IC.; Lai, W.; Huang, F. On distributed multiple access control for wireless sensor networks; Proceedings of the 48th Annual Allerton Conference on Communication, Control, and Computing; Monticello, Illinois. September 29 – October 1 2010;

[13]. Paschalidis, IC. Tech. Rep. 2011-IR-0047. Center for Information and Systems Engineering, Boston University; 2011. A novel message passing algorithm for the maximum weighted independent set problem.

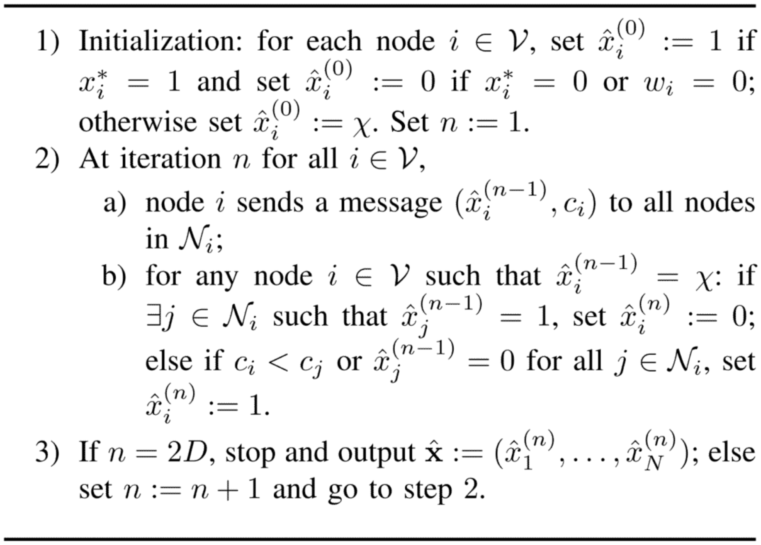
[14]. Kosowski A, Kuszner L. Self-stablizing algorithms for graph coloring with improved performance guarantees. Lecture Notes in Computer Science. 2006; 4029/2006:1150–1159.



```
1) Initialization: set $\theta_{ij}^{(0)} := \max\{w_i, w_j\}$ for all $(i, j) \in \mathcal{E}$, calculate $x_i^{(0)}$ according to equation (7) for all $i \in \mathcal{V}$, and set $n := 1$.
2) At iteration $n$ for all $i \in \mathcal{V}$,
    a) node $i$ sends a message to all its neighbors $\mathcal{N}_i$, with the message being $x_i^{(n-1)}$;
    b) node $i$ calculates $\theta_{ij}^{(n)} = [\theta_{ij}^{(n-1)} - \gamma(1 - x_i^{(n-1)} - x_j^{(n-1)})]_+, \forall j \in \mathcal{N}_i$;
    c) node $i$ calculates $x_i^{(n)}$ according to equation (7) using $\theta_{ij}, \forall j \in \mathcal{N}_i$.
3) Set $n := n + 1$ and go to step 2).
```

Fig. 1.

Gradient projection algorithm for solving (5).



```
1) Initialization: for each node $i \in \mathcal{V}$, set $\hat{x}_i^{(0)} := 1$ if $x_i^* = 1$ and set $\hat{x}_i^{(0)} := 0$ if $x_i^* = 0$ or $w_i = 0$; otherwise set $\hat{x}_i^{(0)} := \chi$. Set $n := 1$.
2) At iteration $n$ for all $i \in \mathcal{V}$,
    a) node $i$ sends a message $(\hat{x}_i^{(n-1)}, c_i)$ to all nodes in $\mathcal{N}_i$;
    b) for any node $i \in \mathcal{V}$ such that $\hat{x}_i^{(n-1)} = \chi$: if $\exists j \in \mathcal{N}_i$ such that $\hat{x}_j^{(n-1)} = 1$, set $\hat{x}_i^{(n)} := 0$; else if $c_i < c_j$ or $\hat{x}_j^{(n-1)} = 0$ for all $j \in \mathcal{N}_i$, set $\hat{x}_i^{(n)} := 1$.
3) If $n = 2D$, stop and output $\hat{\mathbf{x}} := (\hat{x}_1^{(n)}, \ldots, \hat{x}_N^{(n)})$; else set $n := n + 1$ and go to step 2.
```

Fig. 2.

Rounding x* to obtain a feasible solution for (3).

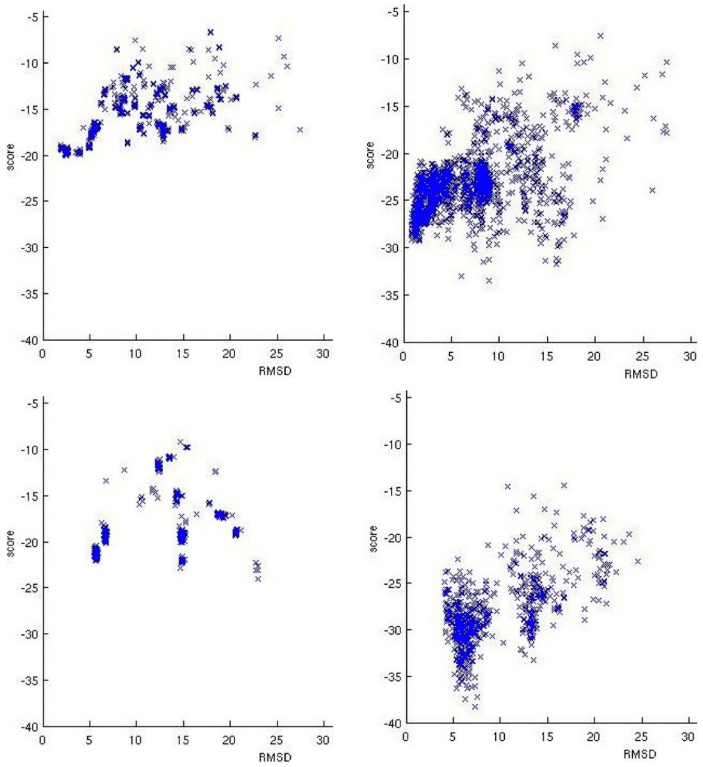


Fig. 3.

RMSD-score plots for complexes 1R0R and 7CEI. Upper left: 1R0R plot after MCM. Upper right: 1R0R plot after MCM+SCP. Lower left: 7CEI plot after MCM. Lower right: 7CEI plot after MCM+SCP. The plots on the right show that applying SCP in the MCM steps improves the RMSD-score plots to reveal a funnel-like behavior.

TABLE I

Number of Near Native Predictions

Protein	Total	Initial	MCM	MCM+SCP	Improvement
1AY7	3655	90	176	239	✓
1BVN	2001	165	295	201	
1E6E	1830	256	416	486	✓
1K74	265	36	43	80	✓
1MAH	1896	188	170	450	✓
1N8O	2974	58	1	106	✓
1NW9	2202	39	122	51	
1PPE	2001	1	34	8	
1R0R	2271	262	83	489	✓
1TMQ	2343	28	56	95	✓
1UDI	2041	143	10	443	✓
2B42	1945	47	0	18	✓
2SIC	2066	79	287	513	✓
2SNI	2280	174	69	502	✓
7CEI	2931	30	1	122	✓

For each protein complex in the test set, “Protein” denotes its PDB identifier. “Total” corresponds to the total number of conformations to refine. “Initial” shows the number of *under* 5 Å conformations among the total initial conformations. MCM denotes the number of *under* 5 Å conformations refined after running McM without side chain positioning, while MCM+SCP specifies the number of *under* 5 Å refined conformations when side chain positioning is included in the MCM steps.