52nd IEEE Conference on Decision and Control
December 10-13, 2013. Florence, Italy

A New Distributed Algorithm for Side-Chain Positioning in the Process of Protein Docking*

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

***Abstract*— *Side-chain positioning (SCP)* is an important component of computational protein docking methods. Existing SCP methods and available software have been designed for protein folding applications where side-chain positioning is also important. As a result they do not take into account significant special structure that SCP for docking exhibits. We propose a new algorithm which poses SCP as a Maximum Weighted Independent Set (MWIS) problem on an appropriately constructed graph. We develop an approximate algorithm which solves a relaxation of the MWIS and then rounds the solution to obtain a high-quality feasible solution to the problem. The algorithm is fully distributed and can be executed on a large network of processing nodes requiring only local information and message-passing between neighboring nodes. Motivated by the special structure in docking, we establish optimality guarantees for a certain class of graphs. Our results on a benchmark set of enzyme-inhibitor protein complexes show that our predictions are close to the native structure and are comparable to the ones obtained by a state-of-the-art method. The results are substantially improved if rotamers from unbound protein structures are included in the search. We also establish that the use of our SCP algorithm substantially improves docking results.**

I. Introduction

Predicting the 3-dimensional structure of proteins is an important problem in computational structural biology. A protein is a *polypeptide* composed of polymers of *amino acid residues*. The atoms of a residue can be classified into two atom-groups: the *backbone* part and the *side-chain*.

In this work, we focus on side chain prediction in the context of protein-protein association. One of the proteins is called the *receptor* to which the other protein – called the *ligand* – binds. Binding can change the unbound conformation of both proteins and the *protein docking* problem aims at computationally predicting the 3-dimensional structure of the complex given the unbound structures of the two protein partners (see [1], [2] and references therein). Side chains are more flexible than the backbone, hence, side-chain positioning is a key component of protein structure prediction [3]. The *Side-Chain Positioning (SCP)* problem can be defined as follows: given a fixed position and orientation of the ligand with respect to the receptor, and assuming that the backbones remain rigid, predict the interface side-chain conformations that minimize the overall energy of the complex.

The side-chains tend to assume only relatively few more-probable low-energy conformations called *rotamers* [4]. These rotamers depend on the backbone and are identified by applying statistical techniques over a large sample of crystal structures. In this study, we use the "2010 Smooth Backbone-Dependent Rotamer Library" [5].

Side chain positioning has been considered in the context of protein structure prediction (protein folding). In such an application one determines the position of all side chains of a protein. The problem is NP-hard [6] and inapproximable [7] and the same is true for SCP in general. Yet, SCP exhibits significant special structure. First, as we already mentioned, we are only interested in positioning the interface side-chains. Side-chains buried within the proteins are typically well-packed and non-interface surface side-chains do not greatly affect the complex. Second, the inclusion of the unbound conformations in the set of rotamers over which the SCP optimization takes place, has the potential to significantly improve prediction accuracy [8]. As we will see, this is firmly established in our study. Third, interface side-chain interactions tend to be local, involving relatively few side chains. All these call for a special purpose algorithm and that is exactly what motivates the work in this paper.

In this paper we formulate SCP as a *Maximum Weighted Independent Set (MWIS)* problem on an appropriately defined graph, which is also NP-hard. We develop a fully distributed algorithm that can obtain near-optimal solutions. In contrast to related work in the context of folding, we elect to develop an approximate algorithm and forgo optimality because state-of-the-art interaction energy models are approximate as well. What we gain is the distributed nature and efficiency of the algorithm which makes it amenable to high-throughput applications in computer clusters. Moreover, our algorithm has several parameters which we can tune to trade-off solution quality vs. speed. This can be useful in packing large interfaces. We establish that our algorithm obtains an optimal solution to SCP for a special class of problems motivated by the structure of SCP arising in docking. In particular, we can guarantee optimality when the graph over which the MWIS is solved is a *perfect graph*.

We test our algorithm on a benchmark set of protein complexes and establish: (i) it produces high quality predictions, (ii) the inclusion of unbound side-chain conformations significantly improves prediction quality, and (iii) incorporating our algorithm in rigid docking procedures significantly improves the accuracy of the predicted conformations.

Research partially supported by the NIH/NIGMS under grants GM093147 and GM061867, by the NSF under grants EFRI-0735974, CNS-1239021 and IIS-1237022, by the ARO under grants W911NF-11-1-0227 and W911NF-12-1-0390, and by the ONR under grant N00014-10-1-0952.

† Division of Systems Eng., Boston Univ., mohamad@bu.edu.
‡ Dept. of Biomedical Eng., Boston Univ., {midas,vajda}@bu.edu.
¶ Dept. of Mechanical Eng., Boston University, vakili@bu.edu.
§ Corresponding author. Dept. of Electrical & Computer Eng., Boston University, 8 Mary's St., Boston, MA 02215, yannisp@bu.edu.

The remainder of the paper is organized as follows. SCP is formulated in Sec. II and the algorithm is presented in Sec. III. Optimality results are in Sec. IV. Specifics concerning the SCP application are in Sec. V. Computational results are in Sec. VI and conclusions in Sec. VII.

II. Side-Chain Positioning Formulation

Since we are interested in side-chain prediction in the context of protein docking, we fix the position and orientation of the ligand with respect to the receptor and consider only the side chains that are close enough to the interface between the two proteins. Specifically, we define the *interface residues* $\mathcal{I}$ as the set of all receptor and ligand residues whose C_α atom is within a small distance (10 Å) from a C_α atom located on the other partner. Let $\mathcal{U}_i$ denote the set of rotamers for each residue $i \in \mathcal{I}$ and denote by $|\mathcal{I}|$ the cardinality of $\mathcal{I}$.

The goal of SCP is to choose one rotamer per residue to minimize the free energy of complex. Let i_r denote the rotamer selected for each residue $i \in \mathcal{I}$. Then, the overall energy takes the form:

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

where E_0 is the 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 for $i \neq j$.

Construct now a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ as follows. $\mathcal{V}$ consists of two types of nodes: *single-rotamer nodes* and *pair-rotamer nodes*. For each rotamer $i_r \in \mathcal{U}_i$ of each residue $i \in \mathcal{I}$, let v_{i_r,i_r} denote the corresponding single-rotamer node. Similarly, for each pair of rotamers $i_r \in \mathcal{U}_i$ and $j_s \in \mathcal{U}_j$ of distinct residues $i, j \in \mathcal{I}$ $(i \neq j)$, let v_{i_r,j_s} denote the corresponding pair-rotamer node. Define now a parameter M which is greater than all node energy values, namely, $M > E(i_r)$ for all $i_r \in \mathcal{U}_i$ of all $i \in \mathcal{I}$ and $M > E(i_r, j_s)$ for all $i_r \in \mathcal{U}_i$, $j_s \in \mathcal{U}_j$ of $i, j \in \mathcal{I}$ with $i \neq j$. To each single-rotamer node v_{i_r,i_r} we assign a *weight* $w_{i_r,i_r} = M - E(i_r) > 0$ and to each pair-rotamer node v_{i_r,j_s} we assign a weight $w_{i_r,j_s} = M - E(i_r, j_s) > 0$.

Next, let us define $\mathcal{E}$, the edge-set of $\mathcal{G}$. Each edge $e \in \mathcal{E}$ represents a *conflict* between rotamers indicating that the rotamers corresponding to the nodes incident to e violate the rule that exactly one rotamer is selected per residue. In other words, an edge $(v_{i_r,j_s}, v_{k_t,l_w}) \in \mathcal{E}$ if the the rotamers i_r, j_s, k_t, l_w include two different rotamers for the same residue (e.g., if $i_r \neq k_t$ and $i = k$).

From the construction of graph $\mathcal{G}$ it easily follows that selecting non-conflicting rotamers for the interface $\mathcal{I}$ amounts to finding an *independent set* of $\mathcal{G}$. (An independent set is a set of nodes so that no two nodes in the set are adjacent.) Moreover, minimizing the energy function in (1) amounts to finding an independent set of $\mathcal{G}$ with maximal total node weight. Such a set is known as a *Maximum Weighted Independent Set (MWIS)* and the problem of finding it – the MWIS problem – is NP-hard. The construction of $\mathcal{G}$ readily leads to the following theorem.

Theorem II.1 *Consider an MWIS of $\mathcal{G}$ with total weight W, and let $n = (|\mathcal{I}| + |\mathcal{I}|(|\mathcal{I}| - 1)/2)$ denote the number of nodes in the MWIS. Then the rotamers associated with the nodes in the MWIS form an optimal solution to the SCP problem with associated minimal energy equal to $nM - W$.*

For any graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ with non-negative weights w_i assigned to nodes $i \in \mathcal{V}$, the MWIS problem can be formulated as an *Integer Linear Programming (ILP)* problem:

$$\begin{array}{rll} \max & \sum_{i=1}^N w_i x_i & \\ \text{s.t.} & x_i + x_j \leq 1, & \forall (i,j) \in \mathcal{E}, \\ & x_i \in \{0,1\}, & i = 1, \ldots, N, \end{array} \quad (2)$$

where $N = |\mathcal{V}|$, and x_i is the indicator variable of selecting node i. We call (2) the *edge formulation* of MWIS. In our earlier work [9], an algorithm was introduced for solving (2) that relied on solving its *Linear Programming (LP)* relaxation. In particular, the LP relaxation of (2) is formed by relaxing the integer constraints $x_i \in \{0,1\}$ as $0 \leq x_i \leq 1$. We call this LP the *edge relaxation* of the MWIS.

In this paper, we consider a tighter relaxation of MWIS. Let us recall some graph-theoretic terminology. Given a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, a *clique* is a subset of nodes such that every two of them are adjacent. The maximum size of a clique in $\mathcal{G}$ is called the *clique number* of $\mathcal{G}$. A *maximal clique* $\mathcal{C}$ is a clique of $\mathcal{G}$ which cannot be extended by adding one more node. Let $\mathcal{S} = \{\mathcal{C}_1, \mathcal{C}_2, \ldots\}$ denote the set of all maximal cliques of $\mathcal{G}$. MWIS can be formulated as the ILP:

$$\begin{array}{rll} \max & \sum_{i=1}^N w_i x_i & \\ \text{s.t.} & \sum_{i\in\mathcal{C}_j} x_i \leq 1, & \forall j : \mathcal{C}_j \in \mathcal{S}, \\ & x_i \in \{0,1\}, & i = 1, \ldots, N. \end{array} \quad (3)$$

We call the inequalities in (3) *clique inequalities*. Although the ILPs (2) and (3) describe the same feasible set, the LP-relaxation of (3) is tighter than that of (2). We call the LP-relaxation of (3) the *clique relaxation* of the MWIS. Solving a tighter relaxation allows us to approach more closely an optimal MWIS solution. However, one should note that the number of maximal cliques in an undirected graph can be large. We will address the scalability of solving (3) in the context of our docking application.

III. A Distributed Algorithm for the Clique-Constrained MWIS

We develop a two-phase algorithm: the first phase solves the clique relaxation and the second phase leverages the relaxed solution to construct an effective MWIS solution.

Consider the clique relaxation:

$$\begin{array}{rll} \max & \sum_{i=1}^N w_i x_i & \\ \text{s.t.} & \sum_{i\in\mathcal{C}_j} x_i \leq 1, & \forall j : \mathcal{C}_j \in \mathcal{S}, \\ & x_i \in [0,1], & i = 1, \ldots, N. \end{array} \quad (4)$$

Even though we define $\mathcal{S}$ as the set of all maximal cliques, the algorithm is also applicable to the generalized case when $\mathcal{S}$ contains any set of cliques in $\mathcal{G}$. In particular, in case we restrict $\mathcal{S}$ to contain only the 2-cliques of $\mathcal{G}$ (i.e., $\mathcal{S}$ would be the edge set of $\mathcal{G}$), the edge relaxation becomes a special case of (4) and the same algorithm can be applied.

A. Phase I: Gradient Projection

The first phase of our algorithm employs the *Gradient Projection (GP)* method to solve the dual of (4). Let us first add a logarithmic barrier to the objective function:

$$\begin{array}{ll} \max & \sum_{i=1}^{N} w_i x_i + \epsilon \sum_{i=1}^{N} (\log x_i + \log (1 - x_i)) \\ \text{s.t.} & \sum_{i \in \mathcal{C}_j} x_i \leq 1, \qquad \forall j : \mathcal{C}_j \in \mathcal{S}, \\ & x_i \in [0, 1], \qquad i = 1, \ldots, N, \end{array} \tag{5}$$

where ϵ is a positive constant, and as $\epsilon \to 0$, the objective function values of (4) and (5) become identical. Considering (5) as the primal problem, each primal variable is associated with a node in $\mathcal{V}$. Let $\boldsymbol{\theta} = (\theta_j; j : \mathcal{C}_j \in \mathcal{S})$ be the dual variables corresponding to the clique constraints in (5).

The dual problem of (5) takes the form:

$$\begin{array}{ll} \min & q(\boldsymbol{\theta}) \\ s.t. & \theta_j \geq 0, \qquad \forall j : \mathcal{C}_j \in \mathcal{S}. \end{array} \tag{6}$$

After some algebra we have:

$$q(\boldsymbol{\theta}) = \sum_{i=1}^{N} \max_{0<x<1} g_i(x) + \sum_{j:\mathcal{C}_j \in \mathcal{S}} \theta_j, \tag{7}$$

where $g_i(x) \triangleq (w_i - \sum_{j:\mathcal{C}_j \in \mathcal{S}, i \in \mathcal{C}_j} \theta_j) x + \epsilon(\log x + \log(1 - x))$, and its unique maximizer $x_i(\boldsymbol{\theta})$ is given by the following lemma; we omit the proof.

Lemma III.1 *For all $i \in \mathcal{V}$, the unique maximizer $x_i(\boldsymbol{\theta}) \in (0, 1)$ of $g_i(x)$ is given by:*

$$x_i(\boldsymbol{\theta}) = \begin{cases} \frac{1 - \frac{2\epsilon}{a_i(\theta)} + \sqrt{\frac{4\epsilon^2}{(a_i(\theta))^2} + 1}}{2}, & \text{if } a_i(\theta) > 0, \\ \frac{1 - \frac{2\epsilon}{a_i(\theta)} - \sqrt{\frac{4\epsilon^2}{(a_i(\theta))^2} + 1}}{2}, & \text{if } a_i(\theta) < 0, \\ 1/2, & \text{if } a_i(\theta) = 0, \end{cases} \tag{8}$$

where $a_i(\boldsymbol{\theta}) = w_i - \sum_{j:\mathcal{C}_j \in \mathcal{S}, i \in \mathcal{C}_j} \theta_j$.

We can rewrite the dual function as $q(\boldsymbol{\theta}) = \sum_{i=1}^{N} g_i(x_i(\boldsymbol{\theta})) + \sum_{j:\mathcal{C}_j \in \mathcal{S}} \theta_j$. Since $g_i(x_i(\boldsymbol{\theta}))$ is continuously differentiable with respect to $\boldsymbol{\theta}$, $q(\boldsymbol{\theta})$ is also continuously differentiable. We can verify that for any $j : \mathcal{C}_j \in \mathcal{S}$

$$\frac{\partial q(\boldsymbol{\theta})}{\partial \theta_j} = 1 - \sum_{i:\mathcal{C}_j \in \mathcal{S}, i \in \mathcal{C}_j} x_i(\boldsymbol{\theta}). \tag{9}$$

We can now employ the GP method for solving the dual problem (6); see Fig. 1. The algorithm only involves message-passing among adjacent nodes and uses local information. At each iteration n of the algorithm, $\mathbf{x}^{(n)}$ and $\boldsymbol{\theta}^{(n)}$ denote the values of the vectors $\mathbf{x}$ and $\boldsymbol{\theta}$, and γ is a prespecified step-size. We also use $\mathcal{N}_i$ to denote the set of all nodes adjacent to node i.

For sufficiently small step-size γ, Thm. III.2 establishes the convergence of the algorithm; we omit the proof. D, in the statement of the theorem, denotes the degree of graph $\mathcal{G}$.

Theorem III.2 *For any γ such that $0 < \gamma < \frac{\epsilon}{2^{\frac{3D+5}{2}}(D+1)^{\frac{3}{2}}}$, the GP algorithm in Fig. 1 converges to the optimal primal-dual pair $(\mathbf{x}^*, \boldsymbol{\theta}^*)$ that solves problems (5) and (6). Moreover, and to reach a dual solution $\boldsymbol{\theta}^{(n)}$ satisfying $|q(\boldsymbol{\theta}^{(n)}) - q(\boldsymbol{\theta}^*)| \leq \sigma$, the algorithm requires $O(1/\sigma)$ iterations.*

1) Initialization: set $\theta_j^{(0)} := \max_{i:i \in \mathcal{C}_j} \{w_i\}$ for all $j : \mathcal{C}_j \in \mathcal{S}$. Calculate $x_i^{(0)}$ according to Eq. (8) 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 $\theta_j^{(n)} = [\theta_j^{(n-1)} - \gamma(1 - \sum_{k:\mathcal{C}_j \in \mathcal{S}, k \in \mathcal{C}_j} x_k^{(n-1)})]_+$, $\forall j : \mathcal{C}_j \in \mathcal{S}, i \in \mathcal{C}_j$;
   c) node i calculates $x_i^{(n)}$ according to Eq. (8) using $\theta_j^{(n)}$, $\forall j : \mathcal{C}_j \in \mathcal{S}, i \in \mathcal{C}_j$.
3) Set $n := n + 1$ and go to Step 2.

Fig. 1. Gradient projection algorithm for solving (6).

The algorithm in Fig. 1 is an infinite loop, and requires a stopping criterion. One option is to stop whenever $|\theta_j^{(n)} - \theta_j^{(n-1)}| \leq \delta$ for all $j : \mathcal{C}_j \in \mathcal{S}, i \in \mathcal{C}_j$. We note that this can be done in a distributed manner using an algorithm that computes a maximum over a graph. Another practical issue is to devise a procedure to select an effective ϵ. We use a systematic method called the *barrier method* [10], to ensure that the output $\mathbf{x}^*$ of GP converges to the optimal solution of (4): we start with an initial value ϵ_0 and run the algorithm until its convergence, and then reduce ϵ and repeat the process until two consecutive runs yield $\boldsymbol{\theta}$'s that are close enough. This can also be done in a distributed manner by pre-storing a fixed decreasing sequence of the ϵ's at the nodes.

In this method, $\epsilon \to 0$ geometrically, and for any given accuracy ϵ_d, we need a polynomial number of iterations in $\log(1/\epsilon_d)$ to converge to an approximate optimal solution of (5). Thus, at each iteration of the barrier method, a pseudo-polynomial number of iterations is needed to achieve a desired accuracy δ. This implies that the gradient projection approach shown in Fig. 1 converges pseudo-polynomially to an approximate optimal solution of the clique relaxation (4).

B. Phase II: Estimation

The GP algorithm of Fig. 1, yields an optimal solution $\mathbf{x}^* = (x_1^*, \ldots, x_N^*)$ to the clique relaxation (4). If the solution consists of all integer values, then it is clearly an optimal solution to MWIS. However, in a general graph, $\mathbf{x}^*$ will not necessarily be integer. Phase II is designed to leverage $\mathbf{x}^*$ and obtain a feasible MWIS solution. First we state a key property from [11].

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

Next, we introduce a greedy estimation algorithm to construct a feasible solution to the MWIS using $\mathbf{x}^*$. This algorithm is shown in Fig. 2, where $\tilde{\mathbf{x}}$ represents the vector of estimated MWIS decision variables, and χ stands for the

"undetermined" state of a decision variable $\hat{x}_i$. The algorithm first assigns $\tilde{x}_i = x_i^*$ for any node i whose x_i^* is 0 or 1. Then, any remaining node i is set to 0 or 1 in a way that maintains the feasibility of the solution only after all nodes in the immediate neighborhood $\mathcal{N}_i$ with weight greater than w_i are processed.

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 $(\hat{x}_i^{(n-1)}, w_i)$ 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 $w_i > w_j$ or $\hat{x}_j^{(n-1)} = 0$ for all $j \in \mathcal{N}_i$, set $\hat{x}_i^{(n)} := 1$.
3) If $n = N$, 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. Greedy estimation algorithm to obtain a feasible solution to the MWIS problem (3).

In case of a tie in step 2(b) of the algorithm, i.e., node i finds a tie for the largest weight in its neighborhood, we can use a unique ID pre-assigned to each node to break the tie. The correctness of the estimation algorithm in Fig. 2 follows from its construction and is stated in Thm. III.4.

Theorem III.4 *The algorithm in Fig. 2 outputs a feasible solution to the MWIS problem (3).*

IV. Optimality for Perfect Graphs

Thm. IV.1 establishes the optimality of our algorithm for perfect graphs; we omit the proof due to space limitations.

Theorem IV.1 *The optimal solution x^* of (4) obtained by the GP algorithm is optimal for the MWIS problem (3) when $\mathcal{G}$ is perfect and the optimal solution of problem (4) is unique.*

We note that the uniqueness assumption in Thm. IV.1 is not very restrictive. In particular, one can randomly perturb the node weights of the graph $\mathcal{G}$ such that only one of the (potentially multiple) optimal solutions remains optimal.

V. Solving SCP as a Clique-Constrained MWIS

The number of nodes in $\mathcal{G}$ increases quadratically with the number of interface residues $|\mathcal{I}|$. To resolve this issue, we partition $\mathcal{I}$ into non-overlapping non-empty *clusters* as follows. First, we compute the interaction energy values between each pair of residues in $\mathcal{I}$. If the interaction energy value between two residues is greater than a small enough value ϵ_c, then we say that the two residues interact with each other. We call a subset of $\mathcal{I}_k \subseteq \mathcal{I}$ a cluster if: (i) $|\mathcal{I}_k| > 1$, (ii) for each residue $r \in \mathcal{I}_k$ there exists at least one residue $s \in \mathcal{I}_k$ that interacts with r, and (iii) there is no residue in $\mathcal{I} \setminus \mathcal{I}_k$ that interacts with any of the residues in $\mathcal{I}_k$.

Clustering residues as described above leads to partitioning $\mathcal{I}$ as $\mathcal{I} = \mathcal{I}_1 \cup \cdots \cup \mathcal{I}_M$, where M is the number of clusters and singletons identified. Each such subset $\mathcal{I}_i$, $i = 1, \ldots, M$, can be "packed" independently (and in parallel) of the others using our MWIS formulation and algorithm.

Based on our statistical analysis on a benchmark set of protein complexes, we conclude that a significant number of clusters contain only 2 residues. In this light, we first prove that for 2-residue clusters our GP algorithm finds an optimal solution to the MWIS problem (3). Then, we relax most of the clique constraints from (3) and keep few "critical" constraints. This makes the problem much smaller, hence, cheaper to solve. Our numerical results show that for 2-residue clusters, the optimality is still valid in the relaxed formulation. Lastly, we generalize the approximate algorithm for clusters with more than 2 residues.

A. An exact algorithm for 2-residue clusters

Our key result establishes the *perfectness* of $\mathcal{G}(\mathcal{V}, \mathcal{E})$ for 2-residue clusters.

Theorem V.1 *Let $\mathcal{G}(\mathcal{V}, \mathcal{E})$ be the graphical representation of a 2-residue cluster for the MWIS problem (3). $\mathcal{G}(\mathcal{V}, \mathcal{E})$ is perfect.*

Provided that $\mathcal{G}(\mathcal{V}, \mathcal{E})$ is perfect, to satisfy the optimality conditions established in Thm. IV.1, we need to make sure that the optimal solution of problem (5) is unique. To guarantee this uniqueness, as discussed earlier, it suffices to randomly perturb the weights of nodes of $\mathcal{G}(\mathcal{V}, \mathcal{E})$. This leads to the following theorem.

Theorem V.2 *The optimal solution $\mathbf{x}^*$ of (4) obtained by the GP algorithm shown in Fig. 1 is optimal for the MWIS problem (3) of a 2-residue cluster with perturbed node weights.*

B. An approximate algorithm for 2-residue clusters

Thm. V.2 requires that the full MWIS (3) formulation is solved and this includes all maximal cliques of $\mathcal{G}$ which can be many. Next we consider a relaxation.

Consider the basic formulation of MWIS provided in (2), which is a special case of (3) including only 2-clique constraints. Select a few additional cliques as follows. Fix a 2-residue cluster $\mathcal{I}_t$ composed of two residues $\{i, j\}$. Suppose residue i (resp., j) has n (resp., m) rotamers $i_1, \ldots, i_n$ (resp., $j_1, \ldots, j_m$). The graph we constructed in Sec. II has single rotamer nodes $\mathcal{V}_i = \{v_{i_1,i_1}, \ldots, v_{i_n,i_n}\}$ and $\mathcal{V}_j = \{v_{j_1,j_1}, \ldots, v_{j_m,j_m}\}$, as well as pair-rotamer nodes $\mathcal{V}_{i,j} = \{v_{i_r,j_s}; r = 1, \ldots, n, s = 1, \ldots, m\}$. From the construction of $\mathcal{G}$ it follows that each of these three sets of nodes is a clique of $\mathcal{G}$. Set $\mathcal{S}_{relaxed} = \{\mathcal{V}_i, \mathcal{V}_j, \mathcal{V}_{i,j}, \mathcal{E}\}$ and consider the following formulation:

$$\begin{array}{ll} \max & \sum_{i=1}^{N} w_i x_i \\ \text{s.t.} & \sum_{i \in \mathcal{C}_j} x_i \leq 1, \qquad \forall j : \mathcal{C}_j \in \mathcal{S}_{relaxed}, \\ & x_i \in \{0, 1\}, \qquad i = 1, \ldots, N. \end{array} \tag{10}$$

We can then apply the algorithm of Sec. III to the LP relaxation of (10). Although we can not guarantee an optimal solution since not all cliques of $\mathcal{G}$ are included in the formulation, our numerical results (cf. Sec. VI) show that in all problem instances resulting in 2-residue clusters we obtain an optimal MWIS solution.

C. An approximate algorithm for larger clusters

Consider a K-residue cluster $\mathcal{I}_t \subseteq \mathcal{I}$ including residues $\{i^{(1)}, \ldots, i^{(K)}\}$. For each pair of residues $(i^{(k)}, i^{(l)})$, we consider the cliques $\mathcal{V}_{i^{(k)}}$ and $\mathcal{V}_{i^{(l)}}$ containing all single-rotamer nodes of $\mathcal{G}$ corresponding to $i^{(k)}$ and $i^{(l)}$, respectively, and the clique $\mathcal{V}_{i^{(k)},i^{(l)}}$ containing all pair-rotamers nodes from $i^{(k)}$ and $i^{(l)}$. With K residues, the total number of such cliques is $D = K + K(K-1)/2$. We set $\mathcal{S}_{relaxed} = \{\mathcal{E}, \mathcal{V}_{i^{(k)}};\ k = 1, \ldots, K, \mathcal{V}_{i^{(k)},i^{(l)}};\ k, l = 1, \ldots, K, k \neq l\}$ and solve the relaxed problem of (10) with this particular $\mathcal{S}_{relaxed}$. Now, of course, $\mathcal{G}$ is not necessarily perfect and our GP algorithm can yield fractional solutions; hence, the use of the greedy estimation algorithm is needed to obtain a feasible MWIS solution.

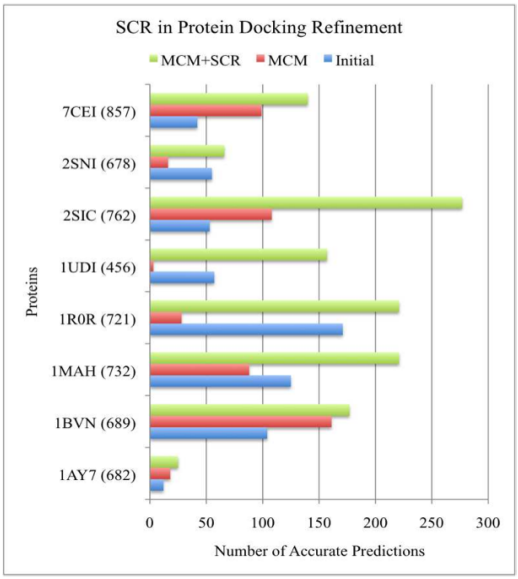

Fig. 3. SCP in docking refinement.

		RMSD			χ_1			χ_{1+2}		
		scwrl	mwis		scwrl	mwis		scwrl	mwis	
Protein	#Res		-UB	+UB		-UB	+UB		-UB	+UB
1AY7	13	9	7	8	12	9	10	8	7	9
1B6C	15	4	4	4	10	9	9	12	12	13
1BUH	13	1	1	1	6	7	8	6	4	5
1BVK	12	4	6	7	6	6	7	3	1	5
1GPW	11	4	6	6	7	7	7	6	6	6
1I2M	20	8	10	12	12	13	17	10	12	14
1KAC	15	6	3	8	7	4	7	5	3	6
1M10	18	6	5	8	13	12	15	8	11	12
1MLC	8	4	3	4	6	6	7	4	4	5
1R0R	14	6	8	9	10	11	12	9	9	9
1S1Q	4	3	2	2	3	3	2	3	2	1
2HRK	19	3	3	8	12	13	15	11	12	15
2O8V	7	2	1	2	5	4	5	3	2	4
2OOB	8	3	4	5	4	6	6	4	3	5
2PCC	5	2	4	4	4	4	4	3	4	4
2SNI	16	10	7	10	12	12	12	11	7	7
7CEI	10	4	4	4	7	8	7	4	5	5
Total	208	79	78	102	136	134	150	110	104	125
%		37.9	37.5	49.0	65.3	64.4	72.1	52.8	50.0	60.1

TABLE I
COMPARING SCWRL4.0 AND MWIS

VI. COMPUTATIONAL RESULTS

A. Algorithm performance

As discussed earlier, our algorithm is an approximation algorithm which iteratively solves a relaxation of MWIS and uses the relaxed solution to construct a "good" feasible solution. The structure of the algorithm provides us with considerable flexibility; we can for instance trade-off quality of the solution vs. speed by relaxing the stopping criterion of the GP phase. Furthermore, the algorithm is fully distributed and is designed to be run over multiple processors.

To verify the accuracy of our algorithm, we compare its results with the SCWRL4.0 package [12] on a benchmark set of unbound enzyme-inhibitor protein complexes. SCWRL4.0 is considered the state-of-the-art in side chain prediction. It is available in executable form and implements an exact method which is centralized. To make the comparison fair we used a special feature of the input file that instructs SCWRL4.0 to optimize the exact same interface side chains optimized by our algorithm. Moreover, we tuned the parameters of our algorithm (e.g., GP stopping criterion) such that both algorithms take the same amount of running time over the full benchmark set. Of course, the amount of computational resources used is not the same as our algorithm uses multiple processors during that time.

We use three performance metrics to evaluate the quality of the solutions. The first metric compares the *Root Mean Square Deviation (RMSD)* of the interface residues from their native conformation obtained by experimental techniques and reported in the Protein Data Bank (PDB) [13]. A predicted residue is considered *"accurate"* if its RMSD from native is no more than 1 Å. The second metric compares the first dihedral angle χ_1 of the residue to the native. Following [12], we call a prediction accurate if χ_1 deviates no more than 40 degrees from its native value. Finally, the third metric compares both the 1st and 2nd dihedral angle (χ_1 and χ_2, respectively) to the native ones. Again, a predicted residue is accurate if each of χ_1 and χ_2 deviate no more than $40°$ from the native values.

Our results are reported in Table I. Protein complexes are listed by their PDB code. For each complex we run both our algorithm and SCWRL4.0 and report the number of interface residues whose predicted conformation is considered accurate according to the RMSD, χ_1 and χ_{1+2} criteria. For each criterion the column "*-UB*" lists results when both algorithms use the rotamer library from [5] which is the one used in SCWRL4.0. This library does not include the rotamers observed in the unbound proteins. We can see (last row of the Table) that our algorithm underperforms SCWRL4.0 by 0.4%, 0.9%, and 2.8% in the three respective criteria. This

is understandable since we solve the problem approximately.

Motivated by [8], we consider the impact of including the unbound rotamers for each residue in our search. This was only possible in our algorithm since for SCWRL4.0 we only have an executable form. The results are listed in the "*+UB*" columns and we can see that performance improves substantially.

B. Application: MWIS in protein docking

To further assess the effectiveness of our SCP algorithm we use it as part of our protein docking refinement procedure. As mentioned in Section I, SCP can become a component of energy evaluation (embedded in local minimization [14]) in refinement techniques [1], [15]. In particular, we retain low energy sampled conformations from rigid-docking techniques [16] and for each conformation we run a docking refinement procedure based on a *Monte Carlo Minimization (MCM)* approach. We run 50 MCM iterations for each conformation and after each iteration we decide either to accept or reject the move based on the *Metropolis* criterion. Results for 8 enzyme-inhibitor protein complexes are shown in Fig. 3. For each complex, we compare three different sets of conformations: (i) "*Initial,*" the low energy conformations generated from rigid-docking techniques [16], (ii) *"MCM,"* the conformations refined by MCM without the use of SCP, and (iii) *"MCM+SCP,"* the conformations refined by MCM which uses SCP as a component of energy evaluation. For each set, we calculate the RMSD of each conformation from the native structure. A prediction is considered "accurate" when this RMSD is below 5 Å. In Fig. 3, we report the number of accurate predictions in each set of conformations described above. The total number of input conformations for each protein complex is indicated in the parentheses after the protein PDB name. It follows that the use of SCP significantly increases the number of near-native predictions.

VII. Conclusions

We developed a new distributed algorithm for solving the SCP problem arising in the context of protein docking. Side-chain positioning has been considered in the context of protein folding, thus, our work fills a void by providing a side-chain prediction procedure to be used in existing docking protocols [1], [15]. The algorithm can be efficiently implemented in a network of processors and involves only local communications between neighboring processors. It solves SCP approximately, obtaining feasible solutions for general problem instances.

Computational results on a protein docking benchmark set suggest that these solutions lead to high-accuracy side-chain predictions and that the inclusion of the unbound rotamers can lead to better predictions. This suggests that the inclusion of these rotamers in SCWRL4.0 is a promising direction for future work. An additional feature of our algorithm is its flexibility in trading off accuracy against running time. Together with its distributed nature, this can lead to fast solutions for large interfaces which can be useful in high-throughput docking applications. Finally, we also demonstrated that adding SCP to docking refinement protocols significantly improves the docking results.

References

[1] Y. Shen, I. C. Paschalidis, P. Vakili, and S. Vajda, "Protein Docking by the Underestimation of Free Energy Funnels in the Space of Encounter Complexes," *PLoS Computational Biology*, vol. 4, no. 10, 2008.

[2] D. Kozakov, D. R. Hall, D. Beglov, R. Brenke, S. R. Comeau, Y. Shen, K. Li, J. Zheng, P. Vakili, I. C. Paschalidis, and S. Vajda, "Achieving reliability and high accuracy in automated protein docking: ClusPro, PIPER, SDU, and stability analysis in CAPRI rounds 13-19," *PROTEINS, Special Issue: Fourth Meeting on the Critical Assessment of PRedicted Interactions*, vol. 78, no. 15, pp. 3124–3130, 2010.

[3] C. Lee and S. Subbiah, "Prediction of protein side-chain conformation by packing optimization," *J. Molecular Biol.*, vol. 217, pp. 373–388, (1991).

[4] J. Ponder and F. Richards, "Tertiary templates for proteins: use of packing criteria in the enumeration of allowed sequences for different structural classes," *J. Molecular Biol.*, vol. 193, pp. 775–791, (1987).

[5] M. Shapovalov and R. Dunbrack, "A smoothed backbone-dependent rotamer library for proteins derived from adaptive kernel density estimates and regressions," *Structure*, vol. 19, pp. 844–858, (2011).

[6] N. Pierce and E. Winfree, "Protein design is NP-hard," *Protein Engrg.*, vol. 15, pp. 779–782, (2002).

[7] B. Chazelle, C. Kingsford, and M. Singh, "A semidefinite programming approach to side chain positioning with new rounding strategies," *INFORMS Journal on Computing*, vol. 16, pp. 380–392, (2004).

[8] D. Beglov, D. Hall, R. Brenke, M. Shapovalov, R. Dunbrack, D. Kozakov, and S. Vajda, "Minimal ensembles of side chain conformers for modeling protein-protein interactions," *Proteins: Structure, Function, and Bioinformatics*, vol. 802, pp. 591–601, (2012).

[9] M. Moghadasi, D. Kozakov, A. Mamonov, P. Vakili, S. Vajda, and I. Paschalidis, "A message passing approach to side chain positioning with applications in protein docking refinement," *in Proceedings of 51th IEEE Conference on Decision and Control, Maui, Hawaii*, (2012).

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

[11] A. Schrijver, *Combinatorial Optimization*. Springer, 2003.

[12] G. Krivov, M. Shapovalov, and R. Dunbrack, "Improved prediction of protein side-chain conformations with SCWRL4," *Proteins*, vol. 77, pp. 778–795, (2009).

[13] H. Berman, J. Westbrook, Z. Feng, G. Gilliland, T. N. Bhat, H. Weissig, I. N. Shindyalov, and P. E. Bourne, "The protein data bank," *Nucleic Acids Research*, vol. 28, pp. 235–242, 2000, http://www.rcsb.org/pdb/home/home.do.

[14] H. Mirzaei, D. Beglov, I. Paschalidis, S. Vajda, P. Vakili, and D. Kozakov, "Rigid body energy minimization on manifolds for molecular docking," *Journal of Chemical Theory and Computation*, p. 591601, (2012).

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

[16] D. Kozakov, R. Brenke, S. Comeau, and S. Vajda, "PIPER: An FFT-based protein docking program with pairwise potentials," *Proteins: Structure, Function, and Bioinformatics*, vol. 65, pp. 392–406, (2006).