Efficient use of semidefinite programming for selection of rotamers in protein conformations *

Forbes Burkowski Yuen-Lam Cheung Henry Wolkowicz†

January 2, 2013

University of Waterloo
Department of Combinatorics and Optimization
Waterloo, Ontario N2L 3G1, Canada

Key Words: Protein Structures, Side Chain Positioning, Euclidean Distance Matrix Completions, Semidefinite Programming.
AMS Subject Classification:

Abstract

Determination of a protein's structure can facilitate an understanding of how the structure changes when that protein combines with other proteins or smaller molecules. In this paper we study a semidefinite programming (SDP) relaxation of the (NP-hard) side chain positioning problem (SCP) presented in Chazelle et al. [4]. We show that the Slater constraint qualification (SCQ) fails for the SDP relaxation. We then show the advantages of using facial reduction to regularize the problem. In fact, after applying facial reduction, we have a *smaller* problem that is *more stable* both in theory and in practice.

Contents

*Research supported by Natural Sciences Engineering Research Council Canada and a grant from AFOSR.
†Research supported by The Natural Sciences and Engineering Research Council of Canada.

List of Tables

List of Figures

List of Algorithms

1 Introduction

Determination of a protein's structure can facilitate an understanding of how the structure changes when that protein combines with other proteins or smaller molecules. These studies are necessary first steps in the understanding of protein functionality. Applications include protein design [11, 12], protein-protein interactions [15, 19, 21], and protein docking (structure based drug design).

From the model in [4] we study the formulation of *side chain positioning* (SCP), i.e., we start with a fixed backbone and a protein sequence and we look for the lowest energy conformation of the protein's side chains on the given backbone. This is an important component of the protein-structure-prediction problem.

We consider a semidefinite programming, SDP, relaxation of SCP and study the proper pre-processing for efficient implementation. In particular, we find that the strict feasibility constraint qualification, the Slater constraint qualification (SCQ), fails. This results in numerical problems. We then apply *facial reduction* to the SDP relaxation. Our numerical tests indicate that facial reduction provides a regularization of the SDP relaxation. The new problem is both smaller and more stable.

1.1 Problem definition

For our purposes, a protein macromolecule is a chain of amino acids that fold into a stable structure with a minimal potential energy. An amino acid has three components: an amino group (H_2N-) and a carboxylic acid group (-COOH) that are bonded to an "alpha" carbon (-C_α-) that is attached to a third atom group called a side-chain. Amino acids are linked to form a protein chain by a condensation reaction that links the amino group of one amino acid to the carboxylic acid group of the next acid. In this reaction, the amino acid gives up a hydrogen atom which combines with the freed (-OH) atoms of the carboxyl group to produce water. Consequently, atoms in the backbone of the protein form a repetitive sequence of triplets: $\cdots NC_\alpha C \quad NC_\alpha C \quad NC_\alpha C \quad NC_\alpha C \cdots$ with each CN bonding being the result of a condensation reaction. We can now visualize the protein chain as this repetitive sequence of atoms with side chain groups sprouting from the alpha carbon atoms. Solving the famous protein folding problem requires an accurate prediction of all atomic positions for the folded minimal energy conformation of such a chain. The problem is known to be NP-hard [1] and so heuristic strategies are typically employed. To make the prediction of protein structure more tractable, the determination of atomic positions is typically subdivided into two sub-problems: a) calculate the positions of atoms in the backbone, b) given the positions of backbone atoms, calculate the conformations of all side-chains. Various techniques such as homology modeling and fold recognition can be used to provide a reasonable assessment of the backbone conformation, see [16].

If we are given the fixed positions of the back-bone atoms, the problem reduces to the simpler (although still NP-hard [1]) side-chain packing problem SCP. As expected, a solution of SCP involves predicting the side-chain conformations that will minimize the potential energy of the protein. The assumption of a fixed backbone gives us reasonable initial conditions for a heuristic solution, although in reality, the conformation of the backbone is dependent on the primary sequence of amino acids and any changes in the side chain positions will produce small perturbations of the positions of the back-bone atoms.

1.1.1 Rotamericity

Even though the problem has been reduced in scope, we will nonetheless utilize a further approximation in an effort to reduce the computational load. This approximation is a discretization of side-chain conformations that capitalizes on the phenomenon of rotamericity. Small side-chains such as glycine and alanine have very simple conformations that are not complicated by the presence of dihedral angles. However, more complicated side-chains contain dihedral angles that can change leading to an infinite set of possibilities for the 3D conformation of the side-chain. Fortunately, it has been observed that a side-chain will typically have a tendency to adopt a conformation that is close to a member of a small set of possible conformations. Each conformation (called a rotamer) is characterized by a particular discrete setting of each dihedral angle in the side-chain.

1.1.2 Discretization via rotamers (rotamer libraries)

Various studies [14, 20] have produced libraries that store the rotamer sets for each of the 20 amino acids. The number of rotamers in a set will depend on the number of dihedral angles in the side-chain. In this paper the more complicated side-chains have rotamer sets with as many as 81 members. There is a computational trade-off involved in this discretization. If the statistical binning employs a larger number of rotamers we will get a higher level of accuracy for the final calculated conformation but there will be a much higher computational cost that grows in a combinatorial fashion with the sizes of the rotamer sets.

1.1.3 Pre-processing to eliminate rotamers

As a preparatory step to reduce the size of the input passed to our algorithm each set of rotamers was filtered using a dead end elimination strategy [6, 14]. Briefly stated, a rotamer can be eliminated from the set associated with some amino acid position if it can be shown that there is another member of the rotamer set that always gives a lower energy value independent of the rotamer choices for the neighbouring amino acids. There are various strategies to do this. We have adopted the approach described in [1, 10].

1.2 Outline

We continue below and present some preliminary notation. Then the model formulation as a $\{0,1\}$-programming problem follows in Section 2. We derive the semidefinite programming (SDP) relaxation in Section 3. This is done both directly and using the dual of the Lagrangian relaxation. In addition, we show the relationships with the relaxation in [4]. The facial reduction is presented in Section 4; finding feasible rounded solutions appears in Section 5.2. The numerical tests are presented in Section 6. Concluding remarks are given in Section 7.

1.3 Preliminaries

Let $\mathcal{G} = (\mathcal{V}, \mathcal{E}, E)$ be a weighted, undirected graph with node set $\mathcal{V} = \bigcup_{i=1}^{p} \mathcal{V}_i$, where each subset $\mathcal{V}_i$ is a set consisting of *rotamers*, and the edge set $\mathcal{E}$ has weights E_{uv}, with edge $uv \cong (u,v) \in \mathcal{E}$.

We work in the vector space of $t \times t$ symmetric matrices, $\mathcal{S}^t$, equipped with the trace inner-product $\langle S, T \rangle = \operatorname{trace} ST$ and with the *Löwner partial order* induced by the cone of positive semidefinite matrices $\mathcal{S}^t_+$, i.e., $S \succeq T$ if $S - T \succeq 0$, is positive semidefinite.

We work with several linear transformations between $\mathcal{S}^t$ and $\mathbb{R}^s$. For $v \in \mathbb{R}^s$, $\mathrm{Diag}(v) \in \mathcal{S}^s$ is the diagonal matrix with diagonal elements taken from v. The adjoint linear transformation $\mathrm{Diag}^*(S) = \mathrm{diag}(S) \in \mathbb{R}^s$ is the vector formed from the diagonal of the matrix S. For the appropriate inner products, the adjoint satisfies

$$\langle \mathrm{diag}(S), v \rangle = \langle S, \mathrm{Diag}(v) \rangle, \quad \forall v \in \mathbb{R}^s, \forall S \in \mathcal{S}^s.$$

We let $\bar{e} = \bar{e}_p, \bar{e}_{m_i}$ denote the vectors of ones of appropriate size. We use the bar to distinguish from the unit vectors. We ignore the subscript when the dimension of $\bar{e}$ is clear. $\bar{E}_k$ is the $k \times k$ matrix of ones. We use $x \circ y$ to denote the Hadamard (elementwise) product of $x, y \in \mathbb{R}^{s \times t}$.

2 Model Formulation

We need to choose exactly one rotamer from each set $\mathcal{V}_i$ based on minimizing the sum of the weights on the edges given in E. The off-diagonal edge weights $E_{uv}, u \neq v$, represent the pairwise interaction energy between the particular pair of rotamers u, v chosen. The diagonal weight E_{uu} is the energy from the interaction between the backbone and the chosen rotamer u. Without loss of generality, we let the size of the subsets $m_k := |\mathcal{V}_k| \geq 1, \forall k$, and we let $m := \begin{pmatrix} m_1 & \dots & m_p \end{pmatrix}^T$ denote the vector of cardinalities of the subsets $\mathcal{V}_i$. We then set $n_0 = |\mathcal{V}| \, (= \sum_k m_k)$, to be the total number of rotamers (i.e., nodes in $\mathcal{G}$); then $n := n_0 + 1$ is the size of matrices in the SDP relaxation below.

2.1 Quadratic integer programming model

Computing the *global minimum-energy conformation* (GMEC) is equivalent to solving the following quadratic integer programming problem

$$\begin{array}{rlll} val_{IQP} = & \min & \displaystyle\sum_{(u,v)\in\mathcal{E}} E_{uv} x_u x_v & \\ \text{(IQP)} & \text{s.t.} & \displaystyle\sum_{u\in\mathcal{V}_k} x_u = 1, & \forall k = 1, \dots, p \\ & & x_u \in \{0,1\}, \forall u \in \mathcal{V}. & \end{array} \tag{2.1}$$

To rewrite IQP using matrix notation, we first label the n_0 nodes in $\mathcal{V}$ as:

$$\mathcal{V}_1 \cong \{1, \dots, m_1\}, \mathcal{V}_2 \cong \{m_1 + 1, \dots, m_1 + m_2\}, \dots, \mathcal{V}_p \cong \left\{ \left(\sum_{k=1}^{p-1} m_k \right) + 1, \dots, n_0 \right\}. \tag{2.2}$$

By abuse of notation, we complete the definition of the weight matrix $E = [E_{uv}]_{u,v\in\mathcal{V}} \in \mathcal{S}^{n_0}$ by setting $E_{uv} = 0$ if $(u, v) \notin \mathcal{E}$. We define the matrix

$$A := \begin{bmatrix} \bar{e}_{m_1}^T & 0 & 0 & \cdots & 0 \\ 0 & \bar{e}_{m_2}^T & 0 & \cdots & 0 \\ 0 & 0 & \bar{e}_{m_3}^T & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & \bar{e}_{m_p}^T \end{bmatrix} \in \{0,1\}^{p \times n_0}.$$

The matrix A satisfies

$$A^TA = \begin{bmatrix} \bar{E}_{m_1} & 0 & 0 & \cdots & 0 \\ 0 & \bar{E}_{m_2} & 0 & \cdots & 0 \\ 0 & 0 & \bar{E}_{m_3} & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & \bar{E}_{m_p} \end{bmatrix}, \qquad AA^T = \begin{bmatrix} m_1 & 0 & 0 & \cdots & 0 \\ 0 & m_2 & 0 & \cdots & 0 \\ 0 & 0 & m_3 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & m_p \end{bmatrix}.$$

We can now rewrite IQP as

$$\text{(IQP)} \qquad \begin{array}{lll} val_{IQP} = & \min & x^TEx \\ & \text{s.t.} & Ax - \bar{e}_p = 0 \in \mathbb{R}^p \\ & & x = \begin{bmatrix} v_1^T & v_2^T & \cdots & v_p^T \end{bmatrix}^T \in \{0,1\}^{(m^T\bar{e})} \\ & & v_k \in \{0,1\}^{m_k}, k = 1, \ldots, p. \end{array} \tag{2.3}$$

We would like to obtain an SDP relaxation of IQP. We accomplish this by using QQP below, an equivalent quadratically constrained quadratic problem. We replace the linear equality constraint $Ax - \bar{e}_p = 0$ by the quadratic $\|Ax - \bar{e}_p\|^2 = 0$. The $0,1$ constraints are modelled using $x \circ x - x = 0$. We add two *redundant quadratic constraints* in QQP within the brackets $\{\}$.

$$\text{(QQP)} \qquad \begin{array}{lll} val_{IQP} = val_{QQP} = & \min_x & x^TEx \\ & \text{s.t.} & \|\bar{e}_p - Ax\|^2 = 0 \\ & & x \circ x - x = 0 \\ & & \left\{ \begin{array}{l} \left(A^TA - I\right) \circ \left(xx^T\right) = 0 \\ (xx^T)_{ij} \geq 0, \ \forall\, (i,j) \in \mathcal{I}, \end{array} \right\} \end{array}$$

where $\mathcal{I} \subseteq \{(i,j) : 1 \leq i < j \leq n_0\}$ is fixed. (Note that the elementwise inequality $xx^T \geq 0$ is a *valid inequality* for IQP.) The two *redundant quadratic constraints* in QQP within the brackets $\{\}$ are useful for restricting the feasible matrices Y in the SDP relaxation. We see below: the redundant Hadamard multiplicative constraint helps to guarantee that the diagonal blocks in the feasible matrices Y are in fact diagonal matrices; the nonnegativity constraints provide a useful cutting plane approach. It is interesting to note that the paper [4] specifically tries to use a minimum number of constraints to model their quadratic problem for IQP; in direct contrast, we add many redundant constraints. We emphasize that adding redundant constraints can significantly improve the Lagrangian relaxation and so the SDP relaxation, as we see below. After we obtain the SDP relaxation, we then remove redundant linear constraints by enforcing linear independence.

In addition, note that we could use p separate norm constraints for each row of $\bar{e}_p - Ax = 0$, rather than the single norm constraint. But, as we see below in Section 4.3, the relaxations end up being equivalent.

3 Lagrangian and SDP Relaxations

We now derive our SDP relaxation using QQP. Our derivation differs from that in [4]. Rather than squaring each side of the linear equality constraint in (2.1), we move the 1 in each constraint to the left and use the norm squared of the complete equation. Furthermore, instead of using the usual *lifting* approach, we use Lagrangian duality to obtain the SDP relaxation. In addition, we

show that the Slater constraint qualification (SCQ), i.e., the existence of a strictly feasible point, fails for this SDP relaxation, and so also for the SDP relaxation in [4]. This generally results in severe numerical difficulties for SDP solvers, and can also result in a *duality gap*. To remove these difficulties we project the problem onto the *minimal face* in order to obtain an SDP relaxation that satisfies SCQ. Thus, we obtain both a smaller model, as well as a more stable program.

3.1 Forming the SDP relaxation

Starting with the quadratic-quadratic model in QQP, we now follow the four steps in [22, 23], see also the recipe in [18]:

1. form the Lagrangian relaxation;
2. apply homogenization;
3. simplify to obtain the dual and an equivalent SDP;
4. take the dual to obtain the SDP relaxation of the original IQP.

Let $\lambda \in \mathbb{R}, w \in \mathbb{R}^{n_0}, \Lambda \in \mathcal{S}^{n_0}, \eta \in \mathbb{R}^{|\mathcal{I}|}$. Define the projection

$$\overline{\mathcal{P}}_{\mathcal{I}} : \mathcal{S}^{n_0} \to \mathbb{R}^{|\mathcal{I}|} : X \mapsto [X_{ij}]_{(i,j)\in\mathcal{I}}.$$

The adjoint $\overline{\mathcal{P}}^*_{\mathcal{I}} : \mathbb{R}^{|\mathcal{I}|} \to \mathcal{S}^{n_0}$ is essentially half the inverse mapping, i.e., with X symmetric and $X_{ij} = 0, \forall (i,j) \notin \mathcal{I}$. The Lagrangian for QQP is

$$\begin{array}{rcl} L(x,\lambda,w,\Lambda,\eta) & = & x^T E x + \lambda \left\| \bar{e}_p - Ax \right\|^2 + w^T (x \circ x - x) \\ & & \quad + \langle \Lambda, (A^T A - I) \circ (xx^T) \rangle - \sum_{(i,j)\in\mathcal{I}} \eta_{ij} (xx^T)_{ij} \\ & = & \left\{ \begin{array}{ll} x^T E x + \lambda x^T A^T A x + w^T (x \circ x) & \\ \qquad + \langle \Lambda, (A^T A - I) \circ (xx^T) \rangle - \langle \eta, \overline{\mathcal{P}}_{\mathcal{I}}(xx^T) \rangle & \underline{\text{quadratic}} \\ -2\lambda \bar{e}_p^T A x - w^T x & \underline{\text{linear}} \\ +\lambda p. & \underline{\text{constant}} \end{array} \right. \end{array}$$

3.1.1 Lagrangian relaxation and homogenization

We begin with the Lagrangian relaxation. We use a homogenizing variable $x_0 \in \mathbb{R}$ and an additional Lagrange multiplier $t \in \mathbb{R}$. The Lagrangian relaxation with homogenization becomes

$$\begin{array}{rcl} d^* & := & \max\limits_{\lambda,w,\Lambda,\eta\geq 0} \left\{ \min\limits_{x} L(x,\lambda,w,\Lambda) \right\} \\ & = & \max\limits_{\lambda,w,\Lambda,\eta\geq 0} \left\{ \min\limits_{x,x_0^2=1} x^T E x + \lambda x^T A^T A x + w^T (x \circ x) + \langle \Lambda, (A^T A - I) \circ (xx^T) \rangle \right. \\ & & \qquad \left. - \langle \overline{\mathcal{P}}^*_{\mathcal{I}}(\eta), xx^T \rangle - 2\lambda x_0 \bar{e}_p^T A x - x_0 w^T x + \lambda p \right\} \qquad (3.1) \\ & = & \max\limits_{\lambda,w,\Lambda,\eta\geq 0,t} \left\{ \min\limits_{x,x_0} x^T E x + \lambda x^T A^T A x + w^T (x \circ x) + \langle \Lambda, (A^T A - I) \circ (xx^T) \rangle \right. \\ & & \qquad \left. - \langle \overline{\mathcal{P}}^*_{\mathcal{I}}(\eta), xx^T \rangle - 2\lambda x_0 \bar{e}_{n_0}^T x - x_0 w^T x + \lambda p + t(1 - x_0^2) \right\}. \end{array}$$

If $x_0 = -1$ in the second equality, then it can be replaced by $x_0 = +1$ if we change the sign $x \leftarrow -x$ as well. The last equality holds by the strong duality of the trust region subproblem, e.g., [9]. Note also that $\bar{e}_p^T A = \bar{e}_{n_0}^T$.

The homogenization allows us to write the inner minimization problem in the last expression in (3.1) as

$$\lambda p + t + \min_{x,x_0} \begin{bmatrix} x_0 & x^T \end{bmatrix} \begin{bmatrix} -t & \left(-\lambda \bar{e}_{n_0}^T - \frac{1}{2} w^T\right) \\ \left(-\lambda \bar{e}_{n_0} - \frac{1}{2} w\right) & \left(E + \lambda A^T A + \mathrm{Diag}(w) + \Lambda \circ (A^T A - I) - \overline{\mathcal{P}}_{\mathcal{I}}^*(\eta)\right) \end{bmatrix} \begin{bmatrix} x_0 \\ x \end{bmatrix},$$

i.e., an unconstrained homogeneous quadratic. Therefore, the minimum is attained at $x_0 = 0, x = 0$, both zero, under the hidden/implicit constraint that the Hessian with respect to x_0, x is positive semidefinite.

Therefore, the Lagrangian relaxation of QQP in (3.1) is equivalent to our first SDP:

$$\begin{array}{lll} d^* = & \max & t + p\lambda \\ \text{(SDP-1)} & \text{s.t.} & \begin{bmatrix} t & \left(\lambda \bar{e}_{n_0}^T + \frac{1}{2} w^T\right) \\ \left(\lambda \bar{e}_{n_0} + \frac{1}{2} w\right) & \left(-\lambda A^T A - \mathrm{Diag}(w) - \Lambda \circ (A^T A - I) + \overline{\mathcal{P}}_{\mathcal{I}}^*(\eta)\right) \end{bmatrix} \preceq \begin{bmatrix} 0 & 0 \\ 0 & E \end{bmatrix} \\ & & \eta \geq 0 \\ & & \lambda, t \in \mathbb{R}, w \in \mathbb{R}^{n_0}, \Lambda \in \mathcal{S}^{n_0}, \eta \in \mathbb{R}^{|\mathcal{I}|}. \end{array} \tag{3.2}$$

Recall that the additional Lagrange multiplier Λ arises from the redundant constraint $(A^T A - I) \circ (xx^T) = 0$, i.e., we could fix $\Lambda = 0$ in SDP-1. Similarly we could set $\eta = 0$. However, having extra dual variables means that the lower bound d^* might be larger (i.e., better). More precisely, it is not clear that these redundant constraints will remain redundant in the SDP relaxation of QQP. Once we have the SDP relaxation, the constraints are linear and one can then determine accurately which constraints are redundant by ensuring linear independence.

3.1.2 Linear transformations and their adjoints

We now rewrite the constraints in SDP-1 in a clearer form using the following linear transformations acting on $v \in \mathbb{R}^{n_0}, S \in \mathcal{S}^{n_0}$. By abuse of notation, the definitions change depending on whether they act on a scalar, vector, or matrix. We include the adjoints as well. The left-hand side of the constraint for SDP-1 can be written using the following linear transformations, thus defining them implicitly.

$$\begin{array}{rcl} LHS & = & \begin{bmatrix} t & \lambda \bar{e}_{n_0}^T + \frac{1}{2} w^T \\ \lambda \bar{e}_{n_0} + \frac{1}{2} w & -\lambda A^T A - \mathrm{Diag}(w) - \Lambda \circ (A^T A - I) + \overline{\mathcal{P}}_{\mathcal{I}}^*(\eta) \end{bmatrix} \\ & = & \begin{bmatrix} t & 0 \\ 0 & 0 \end{bmatrix} + \begin{bmatrix} 0 & \lambda \bar{e}_{n_0}^T \\ \lambda \bar{e}_{n_0} & -\lambda A^T A \end{bmatrix} - \begin{bmatrix} 0 & -\frac{1}{2} w^T \\ -\frac{1}{2} w & \mathrm{Diag}(w) \end{bmatrix} - \begin{bmatrix} 0 & 0 \\ 0 & \Lambda \circ (A^T A - I) \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & \overline{\mathcal{P}}_{\mathcal{I}}^*(\eta) \end{bmatrix} \\ & =: & {}^1\mathcal{O}(t) + {}^{\mathrm{e}}\mathrm{BDiag}(\lambda) - \mathrm{Arrow}(w) - {}^{\mathrm{d}}\mathrm{BDiag}(\Lambda) + \mathcal{P}_{\mathcal{I}}^*(\eta). \end{array} \tag{3.3}$$

We partition $Y = \begin{bmatrix} Y_{00} & y^T \\ y & \bar{Y} \end{bmatrix} \in \mathcal{S}^n, \bar{Y} \in \mathcal{S}^{n_0}$. And we define $\mathcal{P}_{\mathcal{I}}(Y)$ by extending $\overline{\mathcal{P}}_{\mathcal{I}}(\bar{Y})$, i.e., $\mathcal{P}_{\mathcal{I}}(Y) = \overline{\mathcal{P}}_{\mathcal{I}}(\bar{Y})$. We can derive the adjoints of the linear transformations in (3.3) as follows:

1. ${}^1\mathcal{O}^*(Y) = Y_{00}$
2. ${}^{\mathrm{e}}\mathrm{bdiag}(Y) := {}^{\mathrm{e}}\mathrm{BDiag}^*(Y) = -\langle \bar{Y}, A^T A\rangle + 2\bar{e}_{n_0}^T y$
3. $\mathrm{arrow}(Y) := \mathrm{Arrow}^*(Y) = \mathrm{diag}(\bar{Y}) - y$
4. ${}^{\mathrm{d}}\mathrm{bdiag}(Y) := {}^{\mathrm{d}}\mathrm{BDiag}^*(Y) = \bar{Y} \circ (A^T A - I)$

The operator in Item 4 is a so-called *gangster operator* as it *shoots holes/zeros* in the matrix $\bar{Y}$ when it guarantees that the diagonal blocks are themselves diagonal matrices, see [22, 23].

An alternative representation of the constraint matrix in (3.3) uses linear combinations of the following matrices.

$$A_t = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}; \quad A_\lambda = \begin{bmatrix} 0 & \bar{e}_{n_0}^T \\ \bar{e}_{n_0} & -A^T A \end{bmatrix}; \quad A_j = \begin{bmatrix} 0 & \frac{1}{2} e_j^T \\ \frac{1}{2} e_j & -\mathrm{Diag}(e_j) \end{bmatrix}. \tag{3.4}$$

3.1.3 SDP relaxation from dual of Lagrangian relaxation

Recall that the Lagrangian relaxation with the linear transformations introduced above is equivalent to the following SDP.

$$\begin{array}{lll} d^* = & \max & t + p\lambda \\ \text{(SDP-1)} & \text{s.t.} & {}^1\mathcal{O}(t) + {}^{\mathrm{e}}\mathrm{BDiag}(\lambda) - \mathrm{Arrow}(w) - {}^{\mathrm{d}}\mathrm{BDiag}(\Lambda) + \mathcal{P}_{\mathcal{I}}^*(\eta) \preceq \begin{bmatrix} 0 & 0 \\ 0 & E \end{bmatrix} \\ & & \eta \geq 0 \\ & & \lambda, t \in \mathbb{R}, w \in \mathbb{R}^{n_0}, \Lambda \in \mathcal{S}^{n_0}, \eta \in \mathbb{R}^{|\mathcal{I}|}. \end{array} \tag{3.5}$$

We now take the Lagrangian dual of SDP-1 to obtain the SDP relaxation of QQP.

$$\begin{array}{lll} d^{**} := & \min\limits_{Y} & \left\langle \begin{bmatrix} 0 & 0 \\ 0 & E \end{bmatrix}, Y \right\rangle = \langle E, \bar{Y} \rangle \\ & \text{s.t.} & Y_{00} = 1 \\ & & {}^{\mathrm{e}}\mathrm{bdiag}(Y) = p \\ \text{(DSDP-1)} & & \mathrm{arrow}(Y) = 0 \\ & & {}^{\mathrm{d}}\mathrm{bdiag}(Y) = 0 \\ & & \mathcal{P}_{\mathcal{I}}(Y) \geq 0 \\ & & Y = \begin{bmatrix} Y_{00} & y^T \\ y & \bar{Y} \end{bmatrix} \succeq 0. \end{array} \tag{3.6}$$

An equivalent form of DSDP-1 in (3.6) that uses the explicit matrices in (3.4) is given in the following.

$$\begin{array}{lll} d^{**} = & \min\limits_{Y} & \left\langle \begin{bmatrix} 0 & 0 \\ 0 & E \end{bmatrix}, Y \right\rangle \\ & \text{s.t.} & \langle A_t, Y \rangle = 1 \\ & & \langle A_\lambda, Y \rangle = p \\ \text{(DSDP-1)} & & \langle A_j, Y \rangle = 0, \ \forall\, j = 1, \ldots, n_0 \\ & & \bar{Y} \circ (A^T A - I) = 0 \\ & & \bar{Y}_{ij} \geq 0, \ \forall\, (i,j) \in \mathcal{I} \\ & & Y = \begin{bmatrix} Y_{00} & y^T \\ y & \bar{Y} \end{bmatrix} \succeq 0. \end{array} \tag{3.7}$$

Proposition 3.1. *Strong duality holds for SDP-1, i.e., $d^* = d^{**}$ and d^{**} is attained.*

Proof. The results follow since SCQ holds for SDP-1 as both w, t are free variables. □

We shall see below and in Section 4 that, on the other hand, SCQ fails for DSDP-1.

Remark 3.2. *We now describe some properties for the constraints in* (3.6) *(equivalently in* (3.7)*).*

- *The first constraint guarantees that $Y_{00} = 1$.*
- *The second constraint guarantees that the elements of the diagonal blocks sum to the elements of the first row (and column).*
- *The third constraint implies that the matrix Y has the arrow property, i.e., the first row is equal to the diagonal. This with the second constraint implies that the*

  off-diagonal elements of the diagonal blocks sum to 0*.*

  This means that if we include a nonnegativity constraint that $Y \geq 0$, then the diagonal blocks of Y are, in fact, diagonal matrices. This means that $\sum_{i=1}^{p}(m_i - 1)m_i/2$ elements are restricted to be 0*. This* gangster constraint *is a strong constraint on the matrix Y. However, we note that we assumed $Y \geq 0$ to guarantee this.*
- *The fourth constraint guarantees that the off-diagonal elements of the diagonal blocks are indeed all* 0*. We see that this holds without any nonnegativity requirements on Y.*
- *The fifth constraint imposes nonnegativity constraints on some of the elements of Y, indexed by $\{(i+1, j+1) : (i,j) \in \mathcal{I}\}$. While we can always pick $\mathcal{I} = \{(i,j) : 1 \leq i < j \leq n_0\}$, so that*

$$\begin{cases} \bar{Y}_{ij} & \geq 0, \ \forall\, (i,j) \in \mathcal{I} \\ \operatorname{arrow}(Y) & = 0 \\ Y & \succeq 0 \end{cases} \qquad \Longrightarrow \qquad Y \geq 0,$$

  such a choice of $\mathcal{I}$ would lead to $n_0(n_0-1)/2$ inequalities, meaning that DSDP-1 has many more inequality constraints.

Note that feasible points for DSDP-1 exhibit extra structure that is not obvious. First, Y feasible implies that Y has constant trace. Also, as we see in Theorem 4.6, the elements of each diagonal block of Y sums to one.

Proposition 3.3. *If Y is feasible for* (3.7)*, then*

$$\operatorname{trace} \bar{Y} = p, \quad \text{where } Y = \begin{bmatrix} Y_{00} & y^T \\ y & \bar{Y} \end{bmatrix}. \tag{3.8}$$

Proof. The arrow constraint $\operatorname{arrow}(Y) = 0$ together with $\bar{Y} \circ (A^T A - I) = 0$ implies that

$$p = \langle A_\lambda, Y \rangle = 2\bar{e}_{n_0}^T y - \left\langle \bar{Y}, A^T A \right\rangle = 2 \operatorname{trace} \bar{Y} - \operatorname{trace} \bar{Y} = \operatorname{trace} \bar{Y}.$$

This proves (3.8). □

Recall that $n = 1 + n_0$ is the dimension of DSDP-1, i.e., $Y \in \mathcal{S}^n_+ = \mathcal{S}^{1+n_0}_+$, the cone of positive semidefinite matrices. And, note the semidefiniteness of the following matrix (see [22])

$$0 \neq A_{p,\lambda} := \begin{bmatrix} p & 0 \\ 0 & 0 \end{bmatrix} - A_\lambda = \begin{bmatrix} p & -\bar{e}_{n_0}^T \\ -\bar{e}_{n_0} & A^T A \end{bmatrix} \succeq 0.^1 \tag{3.9}$$

Then, for $Y \succeq 0$, feasible for DSDP-1, we get

$$\langle A_{p,\lambda}, Y \rangle = p\langle A_t, Y \rangle - \langle A_\lambda, Y \rangle = p - p = 0. \tag{3.10}$$

Therefore $\mathcal{R}(Y) \subseteq \text{Null}(A_{p,\lambda}) \subsetneq \mathcal{S}^n$, i.e., SCQ fails. In fact, on the feasible set of DSDP-1, we have

$$\text{rank}\,(Y) \leq n - \text{rank}\,(A_{p,\lambda}) = n - \text{rank}\,(A) - 1 = n - p - 1 < n. \tag{3.11}$$

It is now clear that SCQ fails for the SDP relaxation DSDP-1 in (3.7), i.e., the semidefiniteness of both $A_{p,\lambda}$ and Y in the constraint $\langle A_{p,\lambda}, Y \rangle = 0$ means that Y is singular for all Y in the feasible set of (3.7). We will take advantage of this rank deficiency below when we do *facial reduction*, see Section 4.

4 Facial Reduction of the SDP Relaxation

We have seen in (3.9) that the matrix $0 \neq A_{p,\lambda} \succeq 0$. Moreover, this matrix is formed from two constraints so that Y feasible implies that $\langle A_{p,\lambda}, Y \rangle = 0$. Since feasibility implies that $Y \succeq 0$, we conclude that Y is singular. Therefore, we see that the SDP relaxation DSDP-1 does not satisfy the standard SCQ, see Proposition 4.3. This can cause serious problems for interior point methods, which is the method of choice for many current SDP solvers. In fact, the range of a feasible Y is restricted, i.e. $\mathcal{R}(Y) \subseteq \text{Null}(A_{p,\lambda})$. We can take advantage of this and *facially reduce* the problem. The main result in this section from the facial reduction is that DSDP-1 is equivalent to the smaller SDP

$$\begin{array}{ll} \min\limits_{X \in \mathcal{S}^{n-p}} & \left\langle \hat{E}, X \right\rangle \\ \text{s.t.} & \text{arrow}(X) = 0, \\ & {}^{\text{d}}\text{bdiag}(X) = 0, \\ & X_{00} = 1, \\ & X \succeq 0, \\ & (WXW^T)_{ij} \geq 0 \quad \forall\, (i,j) \in \mathcal{I}, \end{array} \tag{4.1}$$

[1] To see that $A_{p,\lambda}$ is positive semidefinite, we consider its Schur complement

$$A^T A - \frac{1}{p} \bar{e}_{n_0} \bar{e}_{n_0}^T.$$

For any $x = [x^{(1)}; x^{(2)}; \ldots; x^{(p)}] \in \mathbb{R}^{n_0}$ with $x^{(i)} \in \mathbb{R}^{m_i}$ for $i = 1, \ldots, p$, we have

$$\left(\bar{e}_{n_0}^T x\right)^2 = \left(\sum_{i=1}^p \bar{e}_{m_i}^T x^{(i)}\right)^2 \leq \left(\sum_{i=1}^p \left|\bar{e}_{m_i}^T x^{(i)}\right|\right)^2 \leq p \sum_{i=1}^p \left(\bar{e}_{m_i}^T x^{(i)}\right)^2 = p\|Ax\|^2 = p x^T A^T A x.$$

This shows that $x^T \left(A^T A - \frac{1}{p} \bar{e}_{n_0} \bar{e}_{n_0}^T\right) x \geq 0$ for all $x \in \mathbb{R}^{n_0}$.

whose dual is given by

$$\begin{array}{ll} \max\limits_{t,w,\Lambda,\xi} & t \\ \text{s.t.} & {}^{1}\mathcal{O}(t) + \text{Arrow}(w) + {}^{\text{d}}\text{BDiag}(\Lambda) + \sum_{(i,j)\in\mathcal{I}} W^T(e_ie_j^T + e_je_i^T)W\xi_{ij} \preceq \hat{E} \\ & \xi \geq 0,\ \xi \in \mathbb{R}^{|\mathcal{I}|}. \end{array} \tag{4.2}$$

First, we need to introduce a few concepts on facial reduction.

Definition 4.1. *A set $\mathcal{K}$ in a vector space is a* convex cone *if $\mathcal{K} \subseteq \mathcal{K} + \mathcal{K}$ and $\lambda\mathcal{K} \subseteq \mathcal{K}, \forall \lambda \geq 0$. A convex cone $\mathcal{T} \subseteq \mathcal{K}$ is a* face *of the convex cone $\mathcal{K}$, denoted $\mathcal{T} \trianglelefteq \mathcal{K}$, if*

$$x, y \in \mathcal{K}, x + y \in \mathcal{T} \implies x, y \in \mathcal{T}.$$

If in addition $\mathcal{T} \neq \mathcal{K}$, then we denote this by $\mathcal{T} \triangleleft \mathcal{K}$.

Definition 4.2. *The minimal face of DSDP-1 in* (3.7) *is the smallest face containing the feasible set $\mathcal{F}_D$ of DSDP-1 and is denoted*

$$f_D = \bigcap \{\mathcal{T} \trianglelefteq \mathcal{S}_+^n : \mathcal{F}_D \subseteq \mathcal{T}\}.$$

Proposition 4.3. *The minimal face f_D of* (3.7) *is a proper face*

$$f_D \triangleleft \mathcal{S}_+^n. \tag{4.3}$$

Proof. Let $Y \in \mathcal{F}_D$, the feasible set of DSDP-1. Since $0 \neq A_{p,\lambda} \succeq 0, Y \succeq 0$ and $\langle A_{p,\lambda}, Y\rangle = 0$ in DSDP-1 (see (3.10)), we conclude that $\mathcal{R}(Y) \subseteq \text{Null}(A_{p,\lambda})$, i.e., the feasible set is a subset of the proper face $\{A_{p,\lambda}\}^{\perp} \cap \mathcal{S}_+^n \triangleleft \mathcal{S}_+^n$. □

The failure of the SCQ is equivalent to (4.3), since DSDP-1 has a feasible solution $Y \succ 0$ if and only if the minimal face f_D containing the feasible region of DSDP-1 is $\mathcal{S}_+^n$ itself. To avoid the failure of the SCQ and the difficulties that this causes for interior point solvers, we find the minimal face containing the feasible set and facially reduce our problem [23]. We reformulate our problem and eliminate the equality constraints in advance.

4.1 Minimal face of SDP relaxation when $\mathcal{I} = \emptyset$

In this section, we show that when we choose $\mathcal{I} = \emptyset$, then DSDP-1 is equivalent to

$$\begin{array}{ll} \min\limits_{X\in\mathcal{S}^{n-p}} & \left\langle \hat{E}, X \right\rangle \\ \text{s.t.} & \text{arrow}(X) = 0, \\ & {}^{\text{d}}\text{bdiag}(X) = 0, \\ & X_{00} = 1, \\ & X \succeq 0, \end{array} \tag{4.4}$$

where $\hat{E} := W^T \begin{bmatrix} 0 & 0 \\ 0 & E \end{bmatrix} W$ with W defined below in (4.8). In addition, the dual of (4.4) is given by

$$\begin{array}{ll} \max\limits_{t,w,\Lambda} & t \\ \text{s.t.} & {}^{1}\mathcal{O}(t) + \text{Arrow}(w) + {}^{\text{d}}\text{BDiag}(\Lambda) \preceq \hat{E}. \end{array} \tag{4.5}$$

We will show that (4.4) and (4.5) both satisfy the SCQ.

Let

$$B_k := \begin{bmatrix} I_{k-1} \\ -\bar{e}_{k-1}^T \end{bmatrix} \in \mathbb{R}^{k\times(k-1)}. \tag{4.6}$$

We take the convention that any matrix $B \in \mathbb{R}^{s\times t}$ is a vacuous matrix whenever s or $t = 0$. In particular, B_1 is a vacuous matrix. Note that for any integer $k \geq 2$,

$$\{\bar{e}_k\}^\perp = \mathcal{R}\left(B_k\right). \tag{4.7}$$

Define the $n \times (n-p)$ block arrow matrix W by

$$W = \begin{array}{c} \\ 1 \\ m_1 \\ m_2 \\ \\ m_p \end{array} \begin{array}{c} \begin{array}{ccccc} 1 & m_1{-}1 & m_2{-}1 & & m_p{-}1 \end{array} \\ \begin{bmatrix} 1 & 0 & 0 & \cdots & 0 \\ e_{m_1} & B_{m_1} & 0 & \cdots & 0 \\ e_{m_2} & 0 & B_{m_2} & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ e_{m_p} & 0 & 0 & \cdots & B_{m_p} \end{bmatrix} \end{array}, \tag{4.8}$$

and for convenience let $w \in \mathbb{R}^n$ be the first (0-) column of W. Indeed, the range of W is the nullspace of $A_{p,\lambda}$.

Lemma 4.4.

$$\mathrm{Null}\left(A_{p,\lambda}\right) = \mathrm{span}\left(\{w\} \cup \left\{ \begin{pmatrix} 0 \\ q_0 \end{pmatrix} : q_0 \in \bigotimes_{i=1}^{p} \{\bar{e}_{m_i}\}^\perp \right\}\right) = \mathcal{R}\left(W\right), \tag{4.9}$$

where $\mathcal{L}_1 \bigotimes \mathcal{L}_2 := \{(x_1, x_2) : x_i \in \mathcal{L}_i,\ i = 1, 2\}$ *for any linear subspaces* $\mathcal{L}_1$, $\mathcal{L}_2$.

Proof. By (4.7),

$$\begin{aligned} \bigotimes_{i=1}^{p} \{\bar{e}_{m_i}\}^\perp &= \mathrm{span}\left\{ e_i \otimes z_i : z_i \in \{\bar{e}_{m_i}\}^\perp,\ e_i \in \mathbb{R}^p,\ i = 1, \ldots, p \right\} \\ &= \mathrm{span}\left\{ e_i \otimes B_{m_i} : m_i > 1,\ i = 1, \ldots, p \right\}, \end{aligned}$$

where $C \otimes D$ denotes the Kronecker product of matrices C and D. This proves the second equality. As a side note, observe that

$$\dim\left(\bigotimes_{i=1}^{p} \{\bar{e}_{m_i}\}^\perp \right) = \sum_{i=1}^{p} \dim\{\bar{e}_{m_i}\}^\perp = \sum_{i=1}^{p} (m_i - 1).$$

Now we prove the first equality. It is immediate that each column of W is in the nullspace of $A_{p,\lambda}$. We show that the $n-p$ columns of W form a basis of $\mathrm{Null}\left(A_{p,\lambda}\right)$. First note that W is full column rank. Observe also that the range of $A_{p,\lambda}$ has dimension at least p. Now for each $i = 1, \ldots, p$, let

$$u_i := \begin{pmatrix} -1 \\ 0_{m_1} \\ \vdots \\ \bar{e}_{m_i} \\ \vdots \\ 0_{m_p} \end{pmatrix}.$$

The set $\{u_1, \ldots, u_p\} \in \mathcal{R}(A_{p,\lambda})$ is linearly independent. This implies that rank $(A_{p,\lambda}) \geq p$. Hence $\dim(\mathrm{Null}(A_{p,\lambda})) = n - p$, and the columns of W form a basis of $\mathrm{Null}(A_{p,\lambda})$. This proves the first equality. $\square$

Next, we show that the feasible set of DSDP-1 is contained in $W\mathcal{S}_+^{n-p}W^T$. Define

$$\begin{aligned} \mathcal{B} &:= \left\{(i,j) : i < j,\ (A^TA)_{ij} = 1\right\} && (4.10)\\ &= \left\{(i,j) : i < j,\ i-1, j-1 \in \mathcal{V}_k \text{ for some } k\right\}, && (4.11)\end{aligned}$$

and, for $k = 1, \ldots, p$,

$$\bar{m}_k := \begin{cases} 0 & \text{if } k = 1, \\ \sum_{l=1}^{k-1} m_l & \text{if } k > 1. \end{cases} \tag{4.12}$$

Lemma 4.5. *For $j = 1, \ldots, n_0$,*

$$W^TA_jW = \begin{cases} \frac{1}{2}(e_1e_{j-k+2}^T + e_{j-k+2}e_1^T) - e_{j-k+2}e_{j-k+2}^T & \text{if } j \neq \bar{m}_k + m_k \text{ for any } k, \\ \displaystyle\sum_{i\in\mathcal{V}_k} \frac{1}{2}(e_1e_{i-k+2}^T + e_{i-k+2}e_1^T) - e_{i-k+2}e_{i-k+2}^T & \\ \quad - \displaystyle\sum_{\substack{(i-1,l-1)\in\mathcal{V}_k,\\ i<l}} e_{i-k+1}e_{l-k+1}^T + e_{l-k+1}e_{i-k+1}^T & \text{if } j = \bar{m}_k + m_k. \end{cases} \tag{4.13}$$

For $(i,j) \in \mathcal{B}$, i.e., for $i-1, j-1 \in \mathcal{V}_k$ (where $k \in \{1, \ldots, p\}$) with $i < j$,

$$\begin{aligned} &W^T(e_ie_j^T + e_je_i^T)W \\ &\quad = \begin{cases} e_{i-k+1}e_{j-k+1}^T + e_{j-k+1}e_{i-k+1}^T & \text{if } j < \bar{m}_k + m_k + 1, \\ e_1e_{i-k+2}^T + e_{i-k+2}e_1^T - 2e_{i-k+2}e_{i-k+2}^T & \\ \quad - \displaystyle\sum_{\substack{l-1\in\mathcal{V}_k\setminus\{i-1\},\\ l\neq\bar{m}_k+m_k+1}} e_{i-k+1}e_{l-k+1}^T + e_{l-k+1}e_{i-k+1}^T & \text{if } j = \bar{m}_k + m_k + 1. \end{cases} \end{aligned} \tag{4.14}$$

Proof. We compute W^TA_jW (for $j = 1, \ldots, n_0$). From (2.2),

$$\mathcal{V}_k = \{\bar{m}_k + 1, \bar{m}_k + 2, \ldots, \bar{m}_k + m_k\}, \quad \forall\, k = 1, \ldots, p.$$

By definition of W in (4.8), we have

$$W^Te_1 = e_1 \in \mathbb{R}^{n-p}.$$

For any $i \in \{2, \ldots, n\}$, we have $i - 1 \in \mathcal{V}_k$ for some $k \in \{1, \ldots, p\}$.

- If $i - 1 < \bar{m}_k + m_k$, then

  $$e_i^TW = e_{i-k+1}^T.$$

  Hence

  $$\begin{aligned} W^TA_{i-1}W &= \frac{1}{2}(W^Te_1e_i^TW + W^Te_ie_1^TW) - W^Te_ie_i^TW \\ &= \frac{1}{2}(e_1e_{i-k+1}^T + e_{i-k+1}e_1^T) - e_{i-k+1}e_{i-k+1}^T \\ &= A_{i-k} \in \mathcal{S}^{n-p}. \end{aligned}$$

- If $i-1 = \bar{m}_k + m_k$, then

$$\begin{aligned} e_i^T W = e_{\bar{m}_k+m_k+1}^T W &= \begin{bmatrix} 1 & 1 & m_1-1 & & m_k-1 & & m_p-1 \\ 1 & 0 & \cdots & -\bar{e}^T & \cdots & 0 \end{bmatrix} \\ &= e_1^T - \sum_{l=\bar{m}_k-k+3}^{\bar{m}_k+m_k-k+1} e_l^T \\ &= e_1^T - \sum_{\substack{l-1\in\mathcal{V}_k,\\ l\neq\bar{m}_k+m_k+1}} e_{l-k+1}^T \end{aligned}$$

Hence

$$\begin{aligned} W^T A_{i-1} W &= \frac{1}{2}(W^T e_1 e_{\bar{m}_k+m_k+1}^T W + W^T e_{\bar{m}_k+m_k+1} e_1^T W) - W^T e_{\bar{m}_k+m_k+1} e_{\bar{m}_k+m_k+1}^T W \\ &= \begin{bmatrix} 1 & \cdots & -\frac{1}{2}\bar{e}^T & \cdots & 0 \\ \vdots & & \vdots & & \vdots \\ -\frac{1}{2}\bar{e} & \cdots & 0 & \cdots & 0 \\ \vdots & & \vdots & & \vdots \\ 0 & \cdots & 0 & \cdots & 0 \end{bmatrix} - \begin{bmatrix} 1 & \cdots & -\bar{e}^T & \cdots & 0 \\ \vdots & & \vdots & & \vdots \\ -\bar{e} & \cdots & \bar{E} & \cdots & 0 \\ \vdots & & \vdots & & \vdots \\ 0 & \cdots & 0 & \cdots & 0 \end{bmatrix} \\ &= \sum_{l\in\mathcal{V}_k} A_{l-k+1} - \sum_{l-1,j-1\in\mathcal{V}_k,\ l<j} e_{l-k+1}e_{j-k+1}^T + e_{j-k+1}e_{l-k+1}^T. \end{aligned}$$

This proves (4.13).

Now we compute $W^T E_{ij} W$ for $(i,j) \in \mathcal{B}$. For all $i-1, j-1 \in \mathcal{V}_k$ with $i < j < \bar{m}_k + m_k + 1$,

$$W^T e_i e_j^T W + W^T e_j e_i^T W = e_{i-k+1}e_{j-k+1}^T + e_{j-k+1}e_{i-k+1}^T.$$

If $i-1, j-1 \in \mathcal{V}_k$ with $i < j = \bar{m}_k + m_k + 1$, then

$$\begin{aligned} & W^T e_i e_j^T W + W^T e_j e_i^T W \\ = \ & e_{i-k+1}e_1^T + e_1 e_{i-k+1}^T - \sum_{\substack{l-1\in\mathcal{V}_k,\\ l\neq\bar{m}_k+m_k+1}} e_{i-k+1}e_{l-k+1}^T + e_{l-k+1}e_{i-k+1}^T \\ = \ & 2A_{i-k+1} - \sum_{\substack{l-1\in\mathcal{V}_k\setminus\{i-1\},\\ l\neq\bar{m}_k+m_k+1}} e_{i-k+1}e_{l-k+1}^T + e_{l-k+1}e_{i-k+1}^T. \end{aligned}$$

□

We now see that the dual program is equivalent to a surprisingly simple program. Note that except for the arrow constraint the other constraints are trivial in the sense that either the variable is free or fixed at 0. (And $X_{00} = 1$.)

Theorem 4.6. *When $\mathcal{I} = \emptyset$, DSDP-1 is equivalent to*

$$\begin{array}{ll} \min\limits_{X\in\mathcal{S}^{n-p}} & \left\langle \hat{E}, X \right\rangle \\ \text{s.t.} & \text{arrow}(X) = 0, \\ & {}^{\text{d}}\text{bdiag}(X) = 0, \\ & X_{00} = 1, \\ & X \succeq 0, \end{array} \tag{4.15}$$

where $\hat{E} := W^T \begin{bmatrix} 0 & 0 \\ 0 & E \end{bmatrix} W$.

Proof. It is immediate that

$$\langle A_{p,\lambda}, Y\rangle = 0,\ Y \succeq 0 \quad \Longleftrightarrow \quad Y = WXW^T \text{ for some } X \in \mathcal{S}_+^{n-p}.$$

By definition of A and $\mathcal{B}$, we then have

$$\bar{Y} \circ (A^T A - I_{n_0}) = 0 \iff Y_{ij} = 0, \forall\, (i,j) \in \mathcal{B}.$$

Moreover,

$$(WXW^T)_{00} = e_1^T WXW^T e_1 = e_1^T X e_1 = X_{00}.$$

Therefore, when $\mathcal{I} = \emptyset$, $Y \in \mathcal{S}^n$ is feasible for DSDP-1 if, and only if,

$$\begin{array}{rcl} Y & = & WXW^T, \\ \left\langle W^T A_j W, X \right\rangle & = & 0,\ \forall\, j = 1, \ldots, n_0 \\ \left\langle W^T (e_i e_j^T + e_j e_i^T) W, X \right\rangle & = & 0,\ \forall\, (i,j) \in \mathcal{B}, \\ X_{00} & = & 1, \\ X & \succeq & 0. \end{array} \tag{4.16}$$

Also, for any $X \in \mathcal{S}^{n-p}$, by (4.13) and (4.14),

$$\begin{array}{rcl} \left\langle W^T A_j W, X \right\rangle & = & 0,\ \forall\, j = 1, \ldots, n_0 \\ \left\langle W^T (e_i e_j^T + e_j e_i^T) W, X \right\rangle & = & 0,\ \forall\, (i,j) \in \mathcal{B}, \end{array} \tag{4.17}$$

if, and only if,

$$\begin{array}{rcl} \left\langle e_1 e_{j+1}^T + e_{j+1} e_1^T - 2 e_{j+1} e_{j+1}, X \right\rangle & = & 0,\ \forall\, j = 1, \ldots, n-p-1 \\ X_{i-k+1, j-k+1} & = & 0,\ \forall\, (i,j) \in \mathcal{B} \text{ with } j \neq \bar{m}_k + m_k + 1 \text{ for any } k, \end{array} \tag{4.18}$$

if, and only if, $\mathrm{arrow}(X) = 0$ and ${}^{\mathrm{d}}\mathrm{bdiag}(X) = 0$. Therefore, when $\mathcal{I} = \emptyset$, Y is feasible for DSDP-1 if, and only if, $Y = WXW^T$ for some X such that

$$\begin{array}{rcl} \mathrm{arrow}(X) & = & 0, \\ {}^{\mathrm{d}}\mathrm{bdiag}(X) & = & 0, \\ X_{00} & = & 1, \\ X & \succeq & 0. \end{array} \tag{4.19}$$

This shows that DSDP-1 is equivalent to (4.15) when $\mathcal{I} = \emptyset$. □

Next we show that (4.15) and its dual both satisfy the SCQ.

Proposition 4.7. *The dual of* (4.15) *is given by*

$$\begin{array}{ll} \max_{t,w,\Lambda} & t \\ \text{s.t.} & {}^{1}\mathcal{O}(t) + \mathrm{Arrow}(w) + {}^{\mathrm{d}}\mathrm{BDiag}(\Lambda) \preceq \hat{E}, \end{array} \tag{4.20}$$

and both (4.15) *and* (4.20) *satisfy SCQ.*

Proof. The fact that (4.20) is the dual of (4.15) simply follows from the definition of the linear operators ${}^1\mathcal{O}$, Arrow and ${}^{\mathrm{d}}$BDiag and their respective adjoints $X \mapsto X_{00}$, arrow and ${}^{\mathrm{d}}$bdiag. To see that (4.15) satisfies the SCQ, note that

$$\hat{X} = {}^1\mathcal{O}(1) + \frac{1}{2(n_0-p)} \operatorname{Arrow}(\bar{e}_{n_0-p}) \tag{4.21}$$

is a Slater point for (4.15), as

$$\begin{aligned} \frac{1}{2(n_0-p)} I_{n_0-p} - \frac{1}{4(n_0-p)^2} \bar{e}_{n_0-p} \bar{e}_{n_0-p}^T &\succeq \frac{1}{2(n_0-p)} I_{n_0-p} - \frac{1}{4(n_0-p)^2} \lambda_{\max}\left(\bar{e}_{n_0-p} \bar{e}_{n_0-p}^T\right) \\ &= \frac{1}{4(n_0-p)} I_{n_0-p}. \end{aligned}$$

To see that (4.20) satisfies the SCQ, take any positive $\alpha > \lambda_{\min}(\hat{E})$, and

$$t = -\frac{\alpha - \lambda_{\min}(\hat{E})}{\lambda_{\min}(\hat{X})}, \quad w = -\frac{\alpha - \lambda_{\min}(\hat{E})}{2(n_0-p)\lambda_{\min}(\hat{X})} \bar{e}_{n_0-p}, \quad \Lambda = 0.$$

Then

$$\begin{aligned} \hat{E} - {}^1\mathcal{O}(t) - \operatorname{Arrow}(w) - {}^{\mathrm{d}}\mathrm{BDiag}(\Lambda) &= \hat{E} + \frac{\alpha - \lambda_{\min}(\hat{E})}{\lambda_{\min}(\hat{X})} \hat{X} \\ &\succeq \hat{E} + (\alpha - \lambda_{\min}(\hat{E}))I \\ &\succeq \alpha I. \end{aligned}$$

Hence (t, w, Λ) is a Slater point for (4.20). □

A consequence of Proposition 4.7 is that each of the diagonal blocks of any feasible point sum to one.

Corollary 4.8. *For $k = 1, \ldots, p$, let $\chi_k \in \mathbb{R}^n$ denote the characteristic vector of $\mathcal{V}_k$, i.e.,*

$$(\chi_k)_0 = 0 \quad \textit{and} \quad (\chi_k)_u = \begin{cases} 1 & \textit{if } u \in \mathcal{V}_k \\ 0 & \textit{if } u \notin \mathcal{V}_k \end{cases} \quad \textit{for } u \in \mathcal{V}.$$

Then if Y is feasible for DSDP-1, we get

$$\left\langle \chi_k \chi_k^T, Y \right\rangle = \left\langle \operatorname{Diag}(\chi_k), Y \right\rangle = 1, \ \textit{for } k = 1, \ldots, p. \tag{4.22}$$

Proof. For $k = 1, \ldots, p$, since each column of B_k sums to zero, we have that $W^T \chi_k = e_1$ and

$$\begin{aligned} \left\langle \chi_k \chi_k^T, Y \right\rangle &= \left\langle W^T \chi_k \chi_k^T W, X \right\rangle \\ &= \left\langle e_1 e_1^T, X \right\rangle = 1; \\ \left\langle \operatorname{Diag}(\chi_k), Y \right\rangle &= \chi_k^T \operatorname{diag}(Y) \\ &= \chi_k^T (WXW^T)_{:1} \\ &= \chi_k^T (WX)(W^T)_{:1} \\ &= (W^T \chi_k)^T X e_1 \\ &= e_1^T X e_1 = 1. \end{aligned}$$

□

Corollary 4.8 implies that when $\mathcal{I} = \{(i,j) : 1 \le i,j \le n\}$, DSDP-1 is indeed equivalent to the SDP program that appeared in [4]:

$$
\begin{array}{rll}
d^{**} \ge \min & \left\langle \begin{bmatrix} 0 & 0 \\ 0 & E \end{bmatrix}, Y \right\rangle & \\
\text{s.t.} & \langle \mathrm{Diag}(\chi_i), Y \rangle = 1, \quad \forall\, i = 1,\dots,p & \\
& \left\langle \chi_i \chi_i^T, Y \right\rangle = 1, \quad \forall\, i = 1,\dots,p & (4.23)\\
& \langle A_j, Y \rangle = 0, \quad \forall\, j = 1,\dots,n_0 & \\
& Y_{00} = 1, & \\
& Y \in \mathcal{S}_+^n,\ Y \ge 0. &
\end{array}
$$

4.2 Minimal face of DSDP-1 in the presence of nonnegativity constraints

In this section, we remove the previous assumption that $\mathcal{I}$ is empty, and show that DSDP-1 is equivalent to (4.15) with the additional constraints $(WXW^T)_{ij} \ge 0$ for all $(i,j) \in \mathcal{I}$, provided that

$$
\mathcal{I} \subseteq \mathcal{I}_{\ge 0} := \{(i,j) : 1 < i < j \le n,\ (i-1, j-1) \notin \mathcal{B}\},
$$

and $\mathcal{B}$ is defined in (4.10). To prove our claim, consider

$$
\begin{array}{rll}
d^{**}_{\ge 0} = \min\limits_{Y} & \left\langle \begin{bmatrix} 0 & 0 \\ 0 & E \end{bmatrix}, Y \right\rangle & \\
\text{s.t.} & Y_{00} = 1 & \\
& \mathrm{trace}\, \bar{Y} = p & \\
& {}^{\mathrm{e}}\mathrm{bdiag}(Y) = p & (4.24)\\
& \mathrm{arrow}(Y) = 0 & \\
& {}^{\mathrm{d}}\mathrm{bdiag}(Y) = 0 & \\
& Y \succeq 0 & \\
& Y_{ij} \ge 0 \quad \forall\,(i,j) \in \mathcal{I}_{\ge 0}, &
\end{array}
$$

Observe that since $Y \succeq 0$ implies $\mathrm{diag}(Y) \ge 0$ and Y satisfies ${}^{\mathrm{d}}\mathrm{bdiag}(Y) = 0$ as well as $\mathrm{arrow}(Y) = 0$, if Y is feasible for (4.24), then $Y \ge 0$. Also,

$$
val_{IQP} \ge d^{**}_{\ge 0} \ge d^{**}. \tag{4.25}
$$

By Theorem 4.6, the problem (4.24) is equivalent to

$$
\begin{array}{lll}
\min_{X \in \mathcal{S}^{n-p}} & \left\langle \hat{E}, X \right\rangle & \\
\text{s.t.} & \mathrm{arrow}(X) = 0, & \\
& {}^{\mathrm{d}}\mathrm{bdiag}(X) = 0, & \\
& X_{00} = 1, & (4.26)\\
& X \succeq 0, & \\
& (WXW^T)_{ij} \ge 0 \quad \forall\,(i,j) \in \mathcal{I}_{\ge 0}. &
\end{array}
$$

The dual of (4.26) is given by

$$
\begin{array}{ll}
\max_{t,w,\Lambda,y}\ t \quad \text{s.t.} & {}^{1}\mathcal{O}(t) + \mathrm{Arrow}(w) + {}^{\mathrm{d}}\mathrm{BDiag}(\Lambda) + \sum_{(i,j)\in\mathcal{I}_{\ge 0}} W^T(e_i e_j^T + e_j e_i^T) W y_{ij} \preceq \hat{E} \\
& y \ge 0,
\end{array} \tag{4.27}
$$

and satisfies the SCQ by Proposition 4.7. We claim that (4.26) also satisfies the SCQ.

Proposition 4.9. *Define*

$$D = \bar{e}_{n-p}\bar{e}_{n-p}^T - {}^1\mathcal{O}(1) - \mathrm{Arrow}(\bar{e}) - {}^{\mathrm{d}}\mathrm{BDiag}(\bar{e}_{n_0-p}\bar{e}_{n_0-p}^T) \in \mathcal{S}^{n-p}\,.$$

Then there exists $\alpha > 0$ such that $\hat{X}+\alpha D$ is strictly feasible for (4.26)*, where $\hat{X}$ is defined in* (4.21)*. (That is, in addition to being feasible for* (4.26)*, we have the inequalities are satisfied strictly).*

Proof. By definition of D,

$${}^{\mathrm{d}}\mathrm{bdiag}(D) = 0,\ \mathrm{arrow}(D) = 0,\ D_{00} = 0.$$

Since $\hat{X}$ is a positive definite matrix as pointed out in Proposition 4.7, for sufficiently small $\alpha > 0$ $\hat{X}+\alpha D$ is feasible for (4.15). In fact, noting that D is indefinite, we have $\lambda_{\min}(\hat{X})/(-\lambda_{\min}(D)) > 0$ and if $0 < \alpha < \lambda_{\min}(\hat{X})/(-\lambda_{\min}(D))$, then

$$\lambda_{\min}(\hat{X} + \alpha D) \geq \lambda_{\min}(\hat{X}) + \alpha\lambda_{\min}(D) > 0.$$

Now we show that for small $\alpha > 0$, we indeed have $Y = W(\hat{X} + \alpha D)W^T$ satisfies $Y_{ij} \geq 0$ for all $0 < i < j$. Write

$$\hat{Y} := W\hat{X}W^T = \begin{array}{c} \\ 1 \\ m_1 \\ m_2 \\ \vdots \\ m_p \end{array} \begin{array}{c} \begin{array}{ccccc} 1 & m_1 & m_2 & \cdots & m_p \end{array} \\ \begin{bmatrix} 1 & \hat{Y}_{01} & \hat{Y}_{02} & \cdots & \hat{Y}_{0p} \\ \hat{Y}_{10} & \hat{Y}_{11} & \hat{Y}_{12} & \cdots & \hat{Y}_{1p} \\ \hat{Y}_{20} & \hat{Y}_{21} & \hat{Y}_{22} & \cdots & \hat{Y}_{2p} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \hat{Y}_{p0} & \hat{Y}_{p1} & \hat{Y}_{p2} & \cdots & \hat{Y}_{pp} \end{bmatrix} \end{array}.$$

Then for $i = 1, \ldots, p$, $2(n_0 - p) = \sum_{i=1}^p 2(m_i - 1)$ so

$$\begin{aligned} \hat{Y}_{i0} &= \begin{bmatrix} \frac{1}{2(n_0-p)}\bar{e}_{m_i-1} \\ 1 - \frac{m_i-1}{2(n_0-p)} \end{bmatrix} > 0 \\ \hat{Y}_{ii} &= \mathrm{Diag}(Y_{i0}) \end{aligned}$$

$$\text{and} \quad \hat{Y}_{ij} = \begin{array}{c} \\ m_i \\ 1 \end{array} \begin{array}{c} \begin{array}{cc} m_j-1 & 1 \end{array} \\ \begin{bmatrix} 0 & \frac{1}{2(n_0-p)}\bar{e}_{m_i-1} \\ \frac{1}{2(n_0-p)}\bar{e}_{m_j-1}^T & 1 - \frac{m_i+m_j-2}{2(n_0-p)} \end{bmatrix} \end{array} \quad (i \neq j).$$

Next note that

$$\tilde{Y} := WDW^T = \begin{array}{c} \\ 1 \\ m_1 \\ m_2 \\ \vdots \\ m_p \end{array} \begin{array}{c} \begin{array}{ccccc} 1 & m_1 & m_2 & \cdots & m_p \end{array} \\ \begin{bmatrix} 0 & 0 & 0 & \cdots & 0 \\ 0 & 0 & \tilde{Y}_{12} & \cdots & \tilde{Y}_{1p} \\ 0 & \tilde{Y}_{21} & 0 & \cdots & \tilde{Y}_{2p} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & \tilde{Y}_{p1} & \tilde{Y}_{p2} & \cdots & 0 \end{bmatrix} \end{array},$$

where for $i \neq j$,

$$\tilde{Y}_{ij} = \begin{array}{c} \\ m_i \\ 1 \end{array} \begin{array}{c} \begin{array}{cc} m_j-1 & 1 \end{array} \\ \begin{bmatrix} \bar{e}_{m_i-1}\bar{e}_{m_j-1}^T & -(m_j-1)\bar{e}_{m_i-1} \\ -(m_i-1)\bar{e}_{m_j-1}^T & (m_i-1)(m_j-1) \end{bmatrix} \end{array}.$$

Hence when $0 < \alpha < \frac{1}{2(n_0-p)^2}$ we have that for all $0 < i < j$, $(\hat{Y} + \alpha\tilde{Y})_{ij} > 0$. Therefore, for sufficiently small $\alpha > 0$, we indeed have that $Y = W(\hat{X} + \alpha D)W^T$ is strictly feasible for (4.26). □

As a remark, note that (4.26) and its dual remain strictly feasible even when $\mathcal{I}_{\geq 0}$ is replaced by one of its proper subsets. Due to the high cost of adding all the nonnegativity constraints, our implementation chooses a special subset to add.

4.3 Equivalence of two relaxations

The inequality (4.25) suggests that by requiring that $(WXW^T)_{ij} \geq 0$ for all $(i,j) \in \mathcal{I}_{\geq 0}$, we can potentially get a better optimal solution in the SDP. Nonetheless, the number of inequality constraints would then be $O(n_0^2)$, meaning that solving (4.26) would be a challenge for standard SDP solvers even if n_0 is not too big. To ensure a faster runtime, we adopt a cutting plane strategy.

5 Implementation

This section discusses our heuristics used for obtaining near optimal solutions of IQP. First we discuss the cutting plane technique, which adds nonnegativity constraints over a number of iterations to produce tighter SDP relaxations and at the same time attempts to keep the computation costs low. Then we outline the techniques we use for obtaining integral solution feasible for IQP from the SDP solutions.

5.1 Cutting plane technique

While the constraint $(WXW^T)_{ij} \geq 0$ is valid for all $(i,j) \in \mathcal{I}_{\geq 0}$, it is extremely expensive to solve the SDP relaxation with all such constraints (i.e., the SDP in (4.26)). To balance the tradeoff between computation costs and adding in as many useful valid constraints as possible, we employ the following cutting plane technique. We start with a small initial set $\mathcal{I} \subset \mathcal{I}_{\geq 0}$ and solve

$$\begin{array}{ll} \min_{X \in \mathcal{S}^{n-p}} & \left\langle \hat{E}, X \right\rangle \\ \text{s.t.} & \text{arrow}(X) = 0, \\ & {}^{\text{d}}\text{bdiag}(X) = 0, \\ & X_{00} = 1, \\ & X \succeq 0, \\ & (WXW^T)_{ij} \geq 0 \quad \forall\, (i,j) \in \mathcal{I}. \end{array} \tag{5.1}$$

Then we find the *most violated* constraints, that is, $(i,j) \in \mathcal{I}_{\geq 0} \backslash \mathcal{I}$ such that $(WXW^T)_{ij}$ is negative and the value $E_{ij}(WXW^T)_{ij}$ is very negative (which happens when $E_{ij} >> 0$), and update $\mathcal{I}$ to include these new indices resulting in a slightly larger index set, and finally we resolve (5.1).

We fix the number of "cuts" (i.e., nonnegativity constraints $(WXW^T)_{ij} \geq 0$) to be added in each iteration. The number of cuts incremented has to be chosen with care: the number being too small would lead to slow progress, and the number being too large would add unnecessary computation cost to get to the final solution. We observe that the larger the problem size is, the larger increment of cuts is required in each iteration in order to reach a nearly optimal solution efficiently.

5.2 Rounding to integral solution

From the solution of the SDP relaxation, we need to obtain a good approximation for the original problem IQP. We adopt the two techniques introduced in [4]:

- *Perron-Frobenius rounding*: let $u \in \mathbb{R}^n$ be an eigenvector corresponding to the largest eigenvalue of the optimal solution Y^* of DSDP-1. It was proved in [4] that the *normalized* vector $u' := \frac{p}{u_2+\ldots+u_n}\begin{pmatrix} u_2 \\ \vdots \\ u_n \end{pmatrix}$ satisfies the constraint $Au' = \bar{e}_p$, and $u' \geq 0$ if Y^* is nonnegative. Even without the constraint $Y \geq 0$ in DSDP-1, u' is still empirically found to be nonnegative.[2]
- *Projection rounding*: the diagonal $\begin{pmatrix} 1 \\ u'' \end{pmatrix}$ of the optimal solution Y^* is used. Again, u'' satisfies $Au'' = \bar{e}_p$, $u'' \geq 0$.

When given a fractional solution c (obtained from either of the above mentioned techniques), we can compute a nearest integral solution x, i.e., x is the nearest vector to c among all feasible solutions of IQP. The nearest integral solution can be found via a simple linear program.

Proposition 5.1. *For any $c \in \mathbb{R}^{n_0}$, the integer program*

$$\min_x \|x - c\| \ \textit{s.t.}\ Ax = \bar{e}_p,\ x \in \{0,1\}^{n_0} \tag{5.2}$$

is equivalent to a linear program.

Proof. Observe that for any feasible solution x of (5.2), the equality constraint $x^T x = p$ is satisfied, so $\|x - c\|^2 = -2c^T x + (\|c\|^2 + p)$. Hence (5.2) is equivalent to the linear integer program

$$\min_x (-c)^T x \text{ s.t. } Ax = \bar{e}_p,\ x \in \{0,1\}^{n_0}. \tag{5.3}$$

Since the columns of A are drawn from the identity matrix I_p, A is totally unimodular. This implies that (5.3) is equivalent to the linear program

$$\min_x (-c)^T x \text{ s.t. } Ax = \bar{e}_p,\ x \in [0,1]^{n_0}, \tag{5.4}$$

which has an integral optimal solution. □

Observe that (5.4) essentially produces a greedy solution, in the sense that for each partition $k = 1, \ldots, p$, $x_k = e_{i_k}$, where i_k is an index such that the maximum entry of the subvector c_k lies in the i_k-th position.

[2] We note that the Perron-Frobenius rounding is equivalent to the best rank-one approximation as given by the Eckart-Young Theorem, [8].

5.3 Summary of algorithm

This section gives an overview of the algorithm we use for performing the SDP relaxation with a cutting plane technique, followed by a rounding to get an approximate solution to IQP.

Below is a list of parameters:

- `numcuts` : number of cuts added each time (dependent on problem size)
- `tol` : zero tolerance on the entries of Y (currently at -1e-8, same as the tolerance of linear infeasibility in SDPT3)
- `maxiter` : maximum number of cutting plane iterations
- `r` : number of times the same rounded solution is allowed to repeat consecutively (i.e., if the same rounded solution can only appear at most `r` times in a row)
- `ceil_E` : the ceiling on the values of E (currently 1e5)

Below is the list of the inputs needed:

- $E \in \mathcal{S}^{n_0}$: energy matrix
- $m \in \mathbb{R}^p$: vector of sizes of residues
- p : number of residues
- $\mathcal{I} \subset \{(i,j) : 1 < i < j \leq n\}$: the set of indices such that the nonnegativty constraint $(WXW^T)_{ij}$ is to be added to the SDP (4.12) for all $(i,j) \in \mathcal{I}$.

At the end of the algorithm we have a collection of rounded solutions v_1, v_2 which empirically give progressively better objective values.

6 Numerical Tests

We now present some numerical results on proteins from the Protein Data Bank (PDB) [3]. For a sidechain packing problem with p partitions of sizes $m_1, \ldots, m_p$, the size of the matrix variable in the facially reduced SDP relaxation is $(n-p) \times (n-p)$, where $n = \sum_{i=1}^{p} m_i + 1$; and, the number of linear equality constraints is $n - p + \sum_{i=1}^{p} \frac{(m_i-1)(m_i)-2}{2}$. In the particular case where $m_i = \bar{m}$ is constant for $i = 1, \ldots, p$, we have that the number of constraints equals $n - p + \frac{p(\bar{m}-1)(\bar{m}-2)}{2}$.

6.1 Relative differences between objective value in IQP and dual SDP optimal value

We measure the quality of the rounded solution using the *relative difference* between the optimal value from (4.20) and the objective value of a feasible solution for IQP generated by the rounding techniques introduced in Section 5.2. (See also the relative gap introduced in [4].) Observe that for any instance of IQP, if we obtain (approximately) optimal primal-dual solutions $(X^*; t^*, w^*, \Lambda^*)$

for the SDP relaxation (4.15)-(4.20) and use the rounding procedure as in Section 5.2 to obtain a feasible solution x for IQP, then

$$x^T E x \geq val_{IQP} \geq d^{**} = d^* = \tilde{d}^* := t^*, \tag{6.1}$$

where $\tilde{d}^*$ is the objective value of the dual (4.20) attained by (t^*, w^*, Λ^*). We then measure the quality of the integral solution obtained from rounding the SDP solution by using the following relative difference

$$\frac{x^T E x - \tilde{d}^*}{\frac{1}{2}\left|x^T E x + \tilde{d}^*\right|}. \tag{6.2}$$

6.2 Computation results on proteins from PDB

First we present some detailed results on the protein triose phosphate isomerase (1TIM) [2]. We then present a summary of results on 26 proteins. The tests are conducted using SDPT3-4.0 on MATLAB version R2012a, on a Linux machine with Intel(R) Xeon(R) CPU E5620 @ 2.40 GHz and 46.76 GB RAM.

Our experiments used a selection of proteins from the Protein Data Bank [3] to get the coordinates of atoms in the backbone. To illustrate the processing applied to these proteins, we consider one example: the protein triose phosphate isomerase [2] (PDB identifier: 1TIM). Extraction of the protein coordinates was done using a Python script that executed in the UCSF Chimera molecular modeling environment [17]. Chimera scripts allow one to make substitutions of sidechains with rotamers that are available from a library provided by the Dunbrack Laboratory [7]. The script calculated both intrinsic energies and the potential energies between interacting rotamers by using an energy function that was implemented with parameter values taken from the research done by Cornell et al. [5]. There are some implementation details involved in the programming of this function and in an effort to provide the same energy calculations as Chazelle et al., we followed the suggestions contained in Kingsford's thesis (see section 3.2.7) [13].

6.2.1 Protein example (PDB ID = 1TIM)

We report on the computation results for the protein triose phosphate isomerase (1TIM) , using different number of cut increments. Table 1 gives the basic information about 1TIM.

To avoid overflow due to the large values in E, we set 10^5 as a ceiling on the elements of E. And we use $\mathcal{I} = \left\{(i,j) : 1 < i < j \leq n,\ E_{ij} > 10^4\right\}$ as the set of initial nonnegativity constraint indices. Table 2 compares the results from different choices of cut increments.

Figure 1a shows the optimal values of the SDP relaxation DSDP-1 over the iterations (with increasingly many nonnegativitiy constraints). In the three different choices of `numcut`, the optimal value increases and reaches a constant order (~ 1000) towards the end of the algorithm. Figure 1b shows the rank of X over the iterations. (Tolerance for zero is set at 1e-5.) We can see a decreasing trend in the rank of X over the iterations. In all different choices of cut increments, we get $\text{rank}\,(X) = 1$ at termination which implies that we have indeed found the optimal integer solution with no rounding needed.

(a) Optimal value of DSDP-1 (b) Rank of X

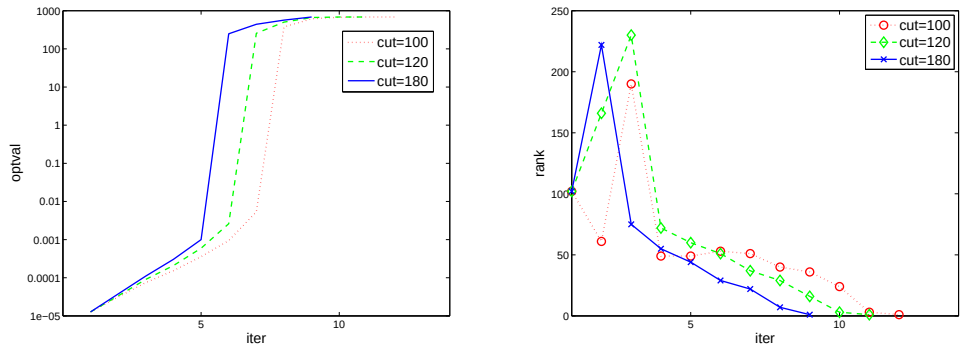

Figure 1: results of DSDP-1

6.2.2 Summary of results on 26 proteins from PDB

Table 3 shows a summary of the numerical results on different proteins, using the cutting plane technique versus the SDP relaxation (4.23) introduced in [4]. A total of 26 proteins of various sizes (n_0) are tested, and the computational results are shown in Tables 3 (for proteins with small n_0), 4 (for medium-sized proteins with n_0) and 5. Hyphens in the tables refer to tests that cannot be completed (because of insufficient memory).

In the numerical tests, `SCPCP` consistently produces better integral solutions in a shorter time than the method proposed in [4]. `SCPCP` consistently uses only a fraction of the computation time. The accuracy of the SDP solution is also higher in the `SCPCP` output, and more importantly, the integer solutions resulting from rounding is essentially optimal because they are usually very close to the dual SDP optimal value, which is a lower bound on the optimal value of IQP.

A few metrics are reported:

- *run time*. In the numerical tests, there is a significant difference in the run time. Even though `SCPCP` requires the solution of multiple SDPs, the fact that in each SDP only a relatively small amount of nonnegativity constraints are imposed reduces the amount of computation work needed.

- *dual SDP optval*, the optimal value of (4.2). Only the final iterate of `SCPCP` is reported. Since (4.2) is the dual of the SDP relaxation (4.1) of IQP, the optimal value of (4.2) serves as a lower bound of the optimal value of IQP.

- *objval in IQP*, the objective value of the integer solution obtained from Perron-Frobenius rounding.

- *relative diff*, relative difference between the objective value in IQP and the optimal value of (4.2). (See (6.2).) As mentioned in Section 6.1, the smaller the relative difference is, the more we can be sure that the integer solution is indeed optimal for IQP.

- *relative gap*, the relative duality gap of the SDP solution. This serves as a measure of how accurate the SDP was solved.

Algorithm 1: `SCPCP`

Parameters(`numcuts`, `tol`, `maxiter`, `r`, `ceil_E`);
Input(E, m, p, $\mathcal{I}$);
Output(v_1, v_2: *feasible solutions for IQP obtained from Perron-Frobenius and projection roundings*);

`Initialization`;
$n \leftarrow \sum_i m_i + 1$;
$E \leftarrow \min\{E, \texttt{ceil\_E}\}$ (elementwise);
$\mathcal{I}_{\text{nonneg}} \leftarrow \{(i,j) : 1 < i < j \leq n,\ (i-1, j-1) \notin \mathcal{B},\ i, j \text{ integral}\}$;

`First iteration`;

`SUBROUTINE`

- solve

$$\begin{aligned} \min_X \langle W^T E W, X \rangle \quad \text{s.t.} \quad & X_{00} = 1,\ \text{arrow}(X) = 0,\ {}^{\text{d}}\text{bdiag}(X) = 0, \\ & (WXW^T)_{ij} \geq 0\ \forall\, (i,j) \in \mathcal{I}, \\ & X \succeq 0, \end{aligned} \tag{5.5}$$

  for optimal solution X^*

- $Y^* \leftarrow WX^*W^T$
- obtain v_1 from Perron-Frobenius rounding, and v_2 from projection rounding

for $k = 1 : \texttt{maxiter}$ **do**
  if $Y^*_{ij} < \texttt{tol}$ *for some* i, j **then**
    $\mathcal{I}_{\text{new}} \leftarrow \mathcal{I}_{\text{nonneg}} \cap \left\{(i,j) : Y^*_{ij} < \texttt{tol}\right\}$;
    if $|\mathcal{I}_{\text{new}}| >$`numcut` **then**
      if $E_{ij}Y_{ij} \geq 0$ *for all* $(i,j) \in \mathcal{I}_{\text{new}}$ **then**
        $\mathcal{I}_{\text{new}} \leftarrow$ the set of indices $(i,j) \in \mathcal{I}_{\text{nonneg}}$ with $i < j$ corresponding to the `numcut`-most negative Y_{ij}
      else
        $\mathcal{I}_{\text{new}} \leftarrow$ the set of indices $(i,j) \in \mathcal{I}_{\text{new}}$ with $i < j$ corresponding to the `numcut`-most negative $E_{ij}Y_{ij}$
      end
      $\mathcal{I} \leftarrow \mathcal{I} \cup \mathcal{I}_{\text{new}}$
    end
    repeat `SUBROUTINE`;
    if v_1, v_2 *are the same as in the previous* `r` *iterates* **then**
      STOP
    end
  end
end

Table 1: Information on input data for 1TIM

Total number of residues / partitions	249
Total number of rotamers / nodes	819
Number of energy values / edges	66520
$\max_{i,j} E_{i,j}$	5.80e+15
$\min_{i,j} E_{i,j}$	-7.7783
Number of valid nonnegativity constraints $\left(= \frac{1}{2}\left(n_0^2 - \sum_{k=1}^{p} m_k^2\right) \right)$	329760

Table 2: Information on output for 1TIM

Increments in cuts	100	120	180
Total time elapsed (hr)	2.51	2.16	1.36
Number of iterations	12	11	9
Final number of nonneg. constr.	2306	2247	2217
Percentage of valid nonneg. constr. used	0.70 %	0.68 %	0.67%
dual SDP optval	685.61	685.61	685.61
objval for IQP	685.61	685.61	685.61
relative diff	5.81e-12	8.68e-12	4.62e-13

Table 3: Results on small proteins

Protein	n_0	p	run time (sec)		dual SDP optval		objval in IQP		relative diff		relative gap	
			SCPCP	[4]	SCPCP	[4]	SCPCP	[4]	SCPCP	[4]	SCPCP	[4]
1AAC	117	85	6.58	296.06	-206.33	-206.33	-206.33	-206.33	5.75E-11	1.72E-05	1.30E-09	4.21E-04
1AHO	108	54	7.97	364.73	33.53	33.53	33.53	33.53	8.44E-11	4.95E-05	2.45E-09	4.68E-04
1BRF	130	45	14.96	977.08	-31.11	-31.11	-31.11	-31.11	3.92E-11	2.27E-05	3.08E-09	1.24E-04
1CC7	160	66	28.60	1059.06	-63.76	-2.30E+07	-63.76	3.73E+04	1.13E-11	2.01	1.27E-09	1.11
1CKU	115	60	5.46	815.18	113.83	113.83	113.83	113.83	7.17E-11	4.79E-05	3.42E-09	1.13E-04
1CRN	65	37	12.76	46.42	-14.87	-14.87	-14.87	-14.87	1.64E-12	3.05E-05	2.20E-10	3.66E-04
1CTJ	153	61	16.15	777.31	-129.53	-6.69E+06	-129.53	174.65	2.98E-11	2.00	2.29E-09	1.07
1D4T	188	89	41.32	2775.34	-173.03	-2.96E+07	-173.03	291.13	3.88E-11	2.00	1.35E-09	1.20
1IGD	82	50	5.51	189.04	-69.25	-69.25	-69.25	-69.25	4.79E-10	2.74E-06	5.76E-09	3.39E-05
1PLC	129	82	14.32	1766.03	-1.50	-1.50	-1.50	-1.50	1.28E-11	7.28E-04	4.60E-10	1.09E-03
1VFY	134	63	23.49	1765.36	-90.09	-90.09	-90.09	-90.09	1.67E-11	-1.11E-05	9.15E-10	3.79E-05
4RXN	98	48	18.44	366.48	-21.65	-21.65	-21.65	-21.65	1.48E-11	2.62E-05	4.19E-10	6.67E-05

Table 4: Results on medium-sized proteins

Protein	n_0	p	run time (min)		dual SDP optval		objval in IQP		relative diff		relative gap	
			SCPCP	[4]	SCPCP	[4]	SCPCP	[4]	SCPCP	[4]	SCPCP	[4]
1B9O	265	112	0.64	254.85	-140.24	-5.63E+07	-140.24	1.91E+06	1.19E-11	2.14	1.45E-09	1.24
1C5E	200	71	2.59	70.63	-131.75	-6.46E+04	-131.75	148.82	4.93E-11	2.01	5.02E-09	1.00
1C9O	207	53	2.15	66.50	-83.55	-1.88E+06	-83.55	1628.10	3.35E-12	2.00	2.77E-10	1.02
1CZP	237	83	1.90	143.95	-37.88	-2.26E+04	-37.88	1254.42	8.30E-11	2.24	1.03E-08	1.00
1MFM	216	118	0.19	102.11	-201.29	-7.36E+07	-201.29	1369.92	2.01E-11	2.00	1.24E-09	1.09
1QQ4	365	143	5.70	-	-102.40	-	-102.40	-	6.49E-11	-	2.27E-08	-
1QTN	302	134	5.04	-	-178.77	-	-178.77	-	2.24E-11	-	4.12E-09	-
1QU9	287	101	7.55	-	-124.96	-	-124.96	-	1.80E-11	-	5.52E-09	-

Table 5: Results on large proteins (SCPCP only)

Protein	n_0	p	run time (hr)	dual SDP optval	Objval in IQP	rel. diff	rel. gap	numcut	# iter	Final # cuts
1CEX	435	146	0.08	140.20	140.20	1.26E-11	5.57E-09	40	9	485
1CZ9	615	111	3.96	497.46	497.46	2.98E-13	6.37E-10	60	25	1997
1QJ4	545	221	0.15	-286.83	-286.83	5.31E-12	1.14E-09	60	14	1027
1RCF	581	142	0.85	-191.54	-191.54	3.71E-12	1.15E-08	60	17	1305
2PTH	930	151	29.65	-159.41	-159.41	8.69E-09	7.63E-06	120	34	7247
5P21	464	144	0.31	-135.75	-135.75	1.39E-12	7.33E-10	40	16	822

7 Conclusion

We applied facial reduction to the SDP relaxation of the side chain positioning problem. We ended up with a smaller, more rigorous relaxation, as seen in our numerical tests, i.e., our optimal solutions were generally better and obtained in a shorter time than those obtained from the approach in [4].

References

[1] T. Akutsu. NP-hardness results for protein side-chain packing. *Genome Informatics*, 8:180–186, 1997. 3, 4

[2] D.W. Banner, A. Bloomer, G.A. Petsko, D.C. Phillips, and I.A. Wilson. Atomic coordinates for triose phosphate isomerase from chicken muscle. *Biochem Biophys Res Commun*, 72(1):146–55, 1976. 23

[3] H.M. 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 Res*, 28:235–242, 2000. 22, 23

[4] B. Chazelle, C. Kingsford, and M. Singh. A semidefinite programming approach to side chain positioning with new rounding strategies. *INFORMS J. Comput.*, 16(4):380–392, 2004. 1, 3, 4, 6, 7, 18, 21, 22, 24, 27, 28

[5] W.D. Cornell, P. Cieplak, C.I. Bayly, I.R. Gould, K.M. Merz, Jr., D.M. Ferguson, D.C. Spellmeyer, T. Fox, J.W. Caldwell, and P.A. Kollman. A second generation force field for the simulation of proteins, nucleic acids, and organic molecules. *J. Am. Chem. Soc.*, 117(19):5179–5197, 1995. 23

[6] J. Desmet, M. de Maeyer, B. Hazes, and I. Lasters. The dead-end elimination theorem and its use in protein side-chain positioning. *Nature*, 356:539–542, 1992. 4

[7] R.L. Dunbrack, Jr. and M. Karplus. Backbone-dependent Rotamer Library for Proteins Application to Side-chain Prediction. *Journal of Molecular Biology*, 230(2):543–574, March 1993. 23

[8] C. Eckart and G. Young. The approximation of one matrix by another of lower rank. *Psychometrika*, 1:211–218, 1936. 21

[9] C. Fortin and H. Wolkowicz. The trust region subproblem and semidefinite programming. *Optim. Methods Softw.*, 19(1):41–67, 2004. 8

[10] R.F. Goldstein. Efficient rotamer elimination applied to protein side-chains and related spin glasses. *Biophysical Journal*, 66(5):1335 – 1340, 1994. 4

[11] D. B. Gordon and S. L. Mayo. Branch-and-terminate: a combinatorial optimization algorithm for protein design. *Structure*, 7(9):1089–1098, September 1999. 3

[12] X. Hu, D. Beratan, and W. Yang. Emergent strategies for inverse molecular design. *Science in China Series B: Chemistry*, 52:1769–1776, 2009. 10.1007/s11426-009-0260-3. 3

[13] C.L. Kingsford. *Computational approaches to problems in protein structure and function*. PhD thesis, Princeton University, Princeton, NJ, USA, 2005. AAI3188669. 23

[14] S.C. Lovell, J.M. Word, J.S. Richardson, and D.C. Richardson. The penultimate rotamer library. *Proteins: Structure Function and Genetics*, 40:389–408, 2000. 4

[15] D. Motiejunas, R. Gabdoulline, T. Wang, A. Feldman-Salit, T. Johann, P. J. Winn, and R.C. Wade. Protein-protein docking by simulating the process of association subject to biochemical constraints. *Proteins*, 71:1955–1969, 2008. 3

[16] K.A. Olszewski, L. Yan, D. Edwards, and T. Yeh. From fold recognition to homology modeling: an analysis of protein modeling challenges at different levels of prediction complexity. *Computers and Chemistry*, 24:499–510, 2000. 3

[17] E.F. Pettersen, T.D. Goddard, C.C. Huang, G.S. Couch, D.M. Greenblatt, E.C. Meng, and T.E. Ferrin. UCSF Chimera–a visualization system for exploratory research and analysis. *Journal of computational chemistry*, 25(13):1605–1612, October 2004. 23

[18] S. Poljak, F. Rendl, and H. Wolkowicz. A recipe for semidefinite relaxation for $(0,1)$-quadratic programming. *J. Global Optim.*, 7(1):51–73, 1995. 7

[19] D. Rajamani, S. Thiel, S. Vajda, , and C.J. Camacho. Anchor residues in protein-protein interactions. *PNAS*, 101(31):11287–11292, 2004. 3

[20] M.V. Shapovalov and R.L. Dunbrack. A smoothed backbone-dependent rotamer library for proteins derived from adaptive kernel density estimates and regressions. *Structure*, 19(6):844 – 858, 2011. 4

[21] C. Wang, O. Schueler-Furman, and D. Baker. Improved side-chain modeling for protein-protein docking. *Protein Sci.*, 14(5):1328–1339, 2005. 3

[22] H. Wolkowicz and Q. Zhao. Semidefinite programming relaxations for the graph partitioning problem. *Discrete Appl. Math.*, 96/97:461–479, 1999. Selected for the special Editors' Choice, Edition 1999. 7, 9, 11

[23] Q. Zhao, S.E. Karisch, F. Rendl, and H. Wolkowicz. Semidefinite programming relaxations for the quadratic assignment problem. *J. Comb. Optim.*, 2(1):71–109, 1998. Semidefinite programming and interior-point approaches for combinatorial optimization problems (Fields Institute, Toronto, ON, 1996). 7, 9, 12

Index