Exact Solutions for the NP-hard Wasserstein Barycenter Problem using a Doubly Nonnegative Relaxation and a Splitting Method [*]

Abdo Alfakih[†] Jeffrey Cheng[‡] Woosuk L. Jung[‡] Walaa M. Moursi[‡]

Henry Wolkowicz[‡]

November 5, 2023, 9:12pm

Abstract

The simplified Wasserstein barycenter problem consists in selecting one point from k given sets, each set consisting of n points, with the aim of minimizing the sum of distances to the barycenter of the k points chosen. This problem is known to be NP-hard. We compute the Wasserstein barycenter by exploiting the Euclidean distance matrix structure to obtain a facially reduced doubly nonnegative, **DNN**, relaxation. The facial reduction provides a natural splitting for applying the symmetric alternating directions method of multipliers (**sADMM**) to the **DNN** relaxation. The **sADMM** method exploits structure in the subproblems to find stong upper and lower bound.

The purpose of this paper is twofold. First we want to illustrate the strength of this **DNN** relaxation with a splitting approach. Our numerical tests then illustrate the surprising success on random problems, as we generally, efficiently, find the provable exact solution of this NP-hard problem. Comparisons with current commercial software illustrate this surprising efficiency. However, we demonstrate and prove that there is a duality gap for problems with *enough* multiple optimal solutions, and that this arises from problems with highly symmetrized structure.

Keywords: Wasserstein barycenters, semidefinite programming, facial reduction

AMS subject classifications: 90C26, 65K10, 90C27, 90C22

Contents

[*] Department of Combinatorics and Optimization, Faculty of Mathematics, University of Waterloo, Waterloo, Ontario, Canada N2L 3G1; Research supported by The Natural Sciences and Engineering Research Council of Canada.

[†] Dept. Math. & Stat., Univ. of Windsor, Canada

[‡] Dept. Comb. & Opt., Univ. of Waterloo, Canada

List of Algorithms

List of Tables

List of Figures

1 Introduction

We consider the simplified Wasserstein barycenter problem of finding the optimal barycenter of k points, where exactly one point is chosen from k sets of points, each set consisting of n points. This is related to the problem of *optimal mass transportation*. Though it is a polynomial time problem in any fixed dimension, it suffers from the *curse of dimensionality*; it has exponential running time with respect to the dimension. For additional details on the theory and applications see e.g., [1,8].

The purpose of this paper is twofold. First, we provide a successful framework for handling quadratic hard discrete optimization problems; and second, we illustrate the surprising success for our specific problem.

We model our problem as a quadratic objective, quadratic constrained $\{0,1\}$ discrete optimization problem, i.e., we obtain a *binary quadratic* model. We then lift, relax, this hard problem to the doubly nonnegative, **DNN**, cone, the cone of nonnegative, positive semidefinite symmetric matrices and obtain a convex relaxation. Strict feasibility fails for the relaxation, so we apply *facial reduction, **FR***. This results in many constraints becoming redundant and also gives rise to a *natural splitting* that can be exploited by the symmetric alternating directions method of multipliers (**sADMM**). We exploit the structure, and include redundant constraints on the subproblems of the splitting and on the dual variables. Efficient upper and lower bounding techniques are used to help the algorithm stop early.

Extensive tests on random problems are surprisingly efficient and successful, i.e., the relaxation with the upper and lower bounding techniques provide a provable optimal solution to the original hard problem for *surprisingly many* instances. The time for our algorithm for a random problem with $k = n = 25$ in dimension $d = 25$ was of the order of 10 seconds. In contrast, CVX MATLAB with solver being the well known commercial package GUROBI took approximately $2, 348, 18000$ seconds for $n = k = 5, 7, 8$, respectively.

The **DNN** relaxation can fail to find the exact solution for problems with special structure. We include a proof that a sufficient number of linearly independent optimal solutions results in a duality gap between the original hard problem and the **DNN** relaxation.

The paper is organized as follows. Following some notation preliminaries, we present the main problem and a reformulation using Euclidean distance matrices in Section 2. The **DNN** relaxation and optimality conditions are given in Section 3. The details for the **sADMM** algorithm are then presented in Section 4. This includes the bounding techinques, scaling, and numerical tests. The theory for finding problems with duality gaps appears in Section 5, see e.g., Corollary 5.3. Our concluding remarks are in Section 6.

1.1 Notation

We let $S \in \mathbb{S}^n$ denote a matrix in the space of $n \times n$ symmetric matrices equipped with the *trace inner product* $\langle S, T \rangle = \operatorname{tr} ST$; we use $\operatorname{diag}(S) \in \mathbb{R}^n$ to denote the linear mapping to the diagonal of S; the adjoint mapping is $\operatorname{diag}^*(v) = \operatorname{Diag}(v) \in \mathbb{S}^n$. We let $[k] = 1, 2, \dots, k$.

The convex cone of positive semidefinite matrices is denoted $\mathbb{S}^n_+ \subset \mathbb{S}^n$, and we use $X \succeq 0$ for $X \in \mathbb{S}^n_+$. Similarly, for positive definite matrices we use $\mathbb{S}^n_{++}, X \succ 0$. We let $\mathcal{N}^n$ denote $n \times n$ nonnegative symmetric matrices. The cone of doubly nonnegative matrices is **DNN** $= \mathbb{S}^n_+ \cap \mathcal{N}^n$.

For a set of points $p_i \in \mathbb{R}^d$, we let $P = \begin{bmatrix} p_1^T \\ p_2^T \\ \dots \\ p_t^T \end{bmatrix} \in \mathbb{R}^{t\times d}$. Here d is the *embedding dimension.* Without loss of generality, we can assume the points span $\mathbb{R}^d$, and we can translate the points and assume they are centered, i.e.,

$$P^T e = 0,\ e \text{ vector of ones.}^{1}$$

we denote the corresponding *Gram matrix,* $G = PP^T$. Then the classical result of Schoenberg [11] relates a *Euclidean distance matrix,* ***EDM***, with a Gram matrix by applying the *Lindenstrauss operator,* $\mathcal{K}(G)$

$$D = \mathcal{K}(G) = \operatorname{diag}(G)e^T + e\operatorname{diag}(G)^T - 2G.$$

Moreover, this mapping is one-one and onto between the *centered subspace,* $\mathcal{S}_C^n$ and *hollow subspace,* $\mathcal{S}_H^n$

$$\mathcal{S}_C^n = \{X \in \mathbb{S}^n \,:\, Xe = 0\}, \quad \mathcal{S}_H^n = \{X \in \mathbb{S}^n \,:\, \operatorname{diag} X = 0\}.$$

We ignore the dimension n when the meaning clear. Note that the centered assumption $P^T e = 0 \implies G = PP^T \in \mathcal{S}_C^n$.

Remark 1.1 (spherical **EDM**)**.** *For centered points that are on a sphere, without loss of generality with radius* 1*, we then know that* $\operatorname{diag}(G) = e$*, the vector of all ones of appropriate dimension. Therefore, we know that* $\operatorname{tr} G = n$*. In the case of points on a sphere that are also centered the* ***EDM*** *is called regular, i.e., if*

$$Ge = 0,\ \operatorname{diag}(G) = e.$$

2 Simplified Wasserstein Barycenters

We now present the main problem and the connections to Euclidean distance matrices, **EDM**.

2.1 Main problem and EDM connection

Our main optimization problem is to find k points for an optimal barycenter.

Problem 2.1 (*Wasserstein Barycenter*)**.** *Suppose that we are given a finite number of sets* $S_1, ..., S_k$*, each consisting of* n *points in* $\mathbb{R}^d$*. Find the optimal barycenter point* y *after choosing exactly one point from each set:*

$$p_W^* := \min_{\substack{y\in\mathbb{R}^d \\ p_i\in S_i}} \sum_{i\in[k]} \|p_i - y\|^2 = \min_{p_i\in S_i}\min_{y\in\mathbb{R}^d} \sum_{i\in[k]} \|p_i - y\|^2 =: \min_{p_{j_i}\in S_i} F(p_{j_1}, p_{j_2}, \dots, p_{j_k}), \tag{2.1}$$

with

$$P^T = \begin{bmatrix} p_1 & \dots & p_n & p_{n+1} & \dots & p_{nk} \end{bmatrix} \in \mathbb{R}^{d\times nk}, D, G, \tag{2.2}$$

denoting the corresponding matrix of points, ***EDM*** *and Gram matrices, respectively.*

[1] The translation is given by

$$P^T \mapsto P^T - ve^T,$$

where $v := \frac{1}{n}P^T e$ is the barycenter of the points.

By Lemma 2.2 below, the optimal Wasserstein barycenter is the standard barycenter of the k optimal points. It is known [2, Sect. 1.2] that the problem can be phrased using inter-point squared distances. We include a proof to emphasize the connection between Gram and Euclidean distance matrices.[2] We start by recording the following minimal property of the standard barycenter with respect to sum of squared distances.

Lemma 2.2. *Suppose that we are given k points $q_i \in \mathbb{R}^d, i = 1, \dots k$. Let $\bar{y} = \frac{1}{k}\sum_{i=1}^k q_i$ denote the barycenter. Then*

$$\bar{y} = \operatorname{argmin}_y \sum_{i=1}^k \frac{1}{2}\|q_i - y\|^2.$$

Proof. The result follows from the stationary point equation $\sum_{i=1}^k (q_i - \bar{y}) = 0$. □

We now have the following useful lemma.

Lemma 2.3. *Let $Q^T = [q_1 \dots q_k] \in \mathbb{R}^{d\times k}$ and let G_Q and D_Q be, respectively, the Gram and the EDM matrices corresponding to the columns in Q^T. Further, let $y = \frac{1}{k}Q^T e$ be the barycenter. Then*

$$e^T D_Q e = 2k\,\mathrm{tr}(G_Q) - 2e^T G_Q e, \tag{2.3}$$

and

$$\sum_{i=1}^k ||q_i - y||^2 = \frac{1}{2k} e^T D_Q e. \tag{2.4}$$

Proof. Let $J = I - ee^T/k$ be the orthogonal projection onto $e^\perp$. Hence, $J^2 = J^T = J$. Moreover, the i-th row $(JQ)_i = (Q - \frac{1}{k}ee^TQ)_i = (q_i - y)^T$. Now

$$\sum_{i=1}^k ||q_i - y||^2 = \mathrm{tr}(JQQ^TJ) = \mathrm{tr}(JG_Q) = \mathrm{tr}(G_Q) - \frac{1}{k}e^T G_Q e.$$

But $D_Q = \mathcal{K}(G_Q) = e\,\mathrm{diag}(G_Q)^T + \mathrm{diag}(G_Q)e^T - 2G_Q$. Therefore, $e^T D_Q e = 2k\,\mathrm{tr}(G_Q) - 2e^T G_Q e$. □

Corollary 2.4. *Consider the main problem (2.1) with optimal Wasserstein barycenter y. This problem is equivalent to finding exactly one point in each set that minimizes the sum of squared distances:*

$$(WIQP) \qquad 2kp_W^* = p^* := \min_{p_1\in S_1,\dots,p_k\in S_k} \sum_{i,j\in[k]} \|p_i - p_j\|^2. \tag{2.5}$$

Proof. Suppose that $Q = \{p_i \mid i \in [k]\}$ is a set of optimal solutions to (2.1) and let y be the barycenter. Without loss of generality, since distances do not change after a translation, we translation all the points p_j by y and obtain $y = 0$. This implies that $G_Q e = P_Q P_Q^T e = 0$. This combined with (2.1) and (2.3) yield

$$\begin{array}{rcl} \sum_{i,j\in[k]} \|p_i - p_j\|^2 & = & e^T D_Q e \\ & = & 2k\,\mathrm{tr}\, G_Q \\ & = & 2k\sum_{i\in[k]} \|p_i\|^2 \\ & = & 2kp_W^*, \end{array} \tag{2.6}$$

where the last equality follows from Lemma 2.2. □

[2] This is called the *cheapest-hub* problem in [2, Sect. 1.2].

2.2 A reformulation using a Euclidean distance matrix

In this section, we reformulate (2.1) using our Euclidean distance matrix (**EDM**) D. Define

$$x := [v_1^T, ..., v_k^T]^T \in \mathbb{R}^{nk}, \quad A := \text{blkdiag}[e^T, ..., e^T] = I \otimes e^T \in \mathbb{R}^{k \times nk},$$

where $\otimes$ denotes the *Kronecker product*. Note that we get $A^T e = e$. Then, the constraints of picking exactly one point from each set can be recast as

$$Ax = e, \, x \text{ binary}. \tag{2.7}$$

Recalling Corollary 2.4 and (2.6) in the proof, we see that (2.1) can be formulated as a binary-constrained quadratic program (**BCQP**) using our Euclidean distance matrix:

$$(\textbf{BCQP}) \qquad \begin{array}{rll} p^* = & \min & x^T D x = \langle D, xx^T \rangle \\ & \text{s.t.} & Ax = e \\ & & x \in \{0,1\}^{kn}. \end{array} \tag{2.8}$$

Remark 2.5 (difficulty of the Wasserstein barycenter problem)**.** *We first note that A is totally unimodular, i.e., every square submatrix has $\det(A_I) \in \{0, \pm 1\}$. Therefore, the basic feasible solutions (vertices of the feasible set) of $Ax = e$, $x \geq 0$, are $\{0, 1\}$ variables. Therefore, these discrete optimization problems with a linear objective yield vertices as optimal solutions and can be solved with simplex type methods. This is what happens for the quadratic assignment problem where the unknown variables are permutation matrices and the problem is relaxed to doubly stochastic matrices (using the Birkhoff-Von Neumann Theorem). Thus, if the objective function is linear we get* $0, 1$ *solutions as the extreme points (basic feasible solutions) are* $0, 1$*.*

However, our quadratic objective function is concave on the span of the feasible set by the properties of distance matrices. Therefore, if we have uniqueness in the solutions we expect $0, 1$ *solutions if we solve the hard concave minimization problem, i.e., the* $0, 1$ *constraints are redundant. However, in our relaxations we linearize the objective as it is not possible to minimize a constrained concave function efficiently in general.*

In summary, the problem appears to be NP-hard due to the minimization of a quadratic function, [9], and the binary $0, 1$ *constraints. However, the unimodularity of the linear constraint matrix suggests that these two constraints both promote binary valued points.*

3 Relaxation of the problem

We now introduce a convex relaxation to the binary quadratic constrained problem in (2.8).

3.1 Semidefinite programming (SDP) relaxation

We start with a **SDP** relaxation of our formulation in (2.8). The idea is to append an extra 1 in front of a feasible vector x, i.e., $\begin{pmatrix} x_0 \\ x \end{pmatrix} = \begin{pmatrix} 1 \\ x \end{pmatrix}$; and then lift it into a rank-1 matrix $Y_x := \begin{pmatrix} 1 \\ x \end{pmatrix} \begin{pmatrix} 1 \\ x \end{pmatrix}^T$. We then relax the nonconvex rank-1 constraint. After the lifting, we impose the constraints that we have from x onto Y, e.g., the $0, 1$ constraints become the $\text{arrow}(Y_x) = e_0$ constraint

$$\text{arrow} : \mathbb{S}^{n+1} \to \mathbb{R}^{n+1} : \begin{bmatrix} s_0 & s^T \\ s & \bar{S} \end{bmatrix} \mapsto \begin{pmatrix} s_0 \\ \text{diag}(\bar{S}) - s \end{pmatrix}.$$

Here we denote e_0, *0-th unit vector*. This implies that the binary constraint on vector x is equivalent to the arrow constraint on the lifted matrix Y_x as long as the rank-one condition holds. The linear constraints $AX = e$ is handled next using **FR**.

3.1.1 SDP reformulation via facial reduction

Recalling (2.2), with matrix variable Y_x, define

$$\hat{D} := \begin{bmatrix} 0 & 0 \\ 0 & D \end{bmatrix} \in \mathbb{S}^{kn+1}, \tag{3.1}$$

and denote the positive semidefinite matrix

$$K := \begin{bmatrix} -e^T \\ A^T \end{bmatrix} \begin{bmatrix} -e^T \\ A^T \end{bmatrix}^T \in \mathbb{S}^{kn+1}_+. \tag{3.2}$$

The objective function of (2.8) now becomes $\langle D, xx^T \rangle = \langle \hat{D}, Y_x \rangle$. For the "only-one-element-from-each-set" linear equality constraint (see (2.7)), we observe that

$$\begin{aligned} Ax = e \iff & \begin{pmatrix} 1 \\ x \end{pmatrix}^T \begin{bmatrix} -e^T \\ A^T \end{bmatrix} = 0 \\ \iff & Y_x K := \begin{pmatrix} 1 \\ x \end{pmatrix} \begin{pmatrix} 1 \\ x \end{pmatrix}^T \begin{bmatrix} -e^T \\ A^T \end{bmatrix} \begin{bmatrix} -e^T \\ A^T \end{bmatrix}^T = 0 \\ \iff & \langle Y_x, K \rangle = 0 \\ \iff & KY_x = 0, \text{ i.e., } \operatorname{range}(Y_x) \subseteq \operatorname{null}(K) = \operatorname{null}\left(\begin{bmatrix} -e & A \end{bmatrix} \right). \end{aligned} \tag{3.3}$$

The last step follows since both $K, Y_x \succeq 0$.

If we choose V full column rnak so that $\operatorname{range}(V) = \operatorname{null}(K)$, then we can *facially reduce* the problem using the substitution

$$Y \leftarrow VRV^T \in V\mathbb{S}^{nk+1-k}_+ V^T \trianglelefteq \mathbb{S}^{kn+1}_+, \tag{3.4}$$

where $\trianglelefteq$ denotes *face of*. This makes the constraint $KY = 0$ redundant.

Remark 3.1. *Note that we need V to satisfy $V^TV = I$ for our application. We can rewrite the matrix $\begin{bmatrix} -e & A \end{bmatrix}$ by permuting columns as follows*

$$\begin{bmatrix} -e & A \end{bmatrix} P = \begin{bmatrix} I_k & I_k \otimes e_{n-1}^T & -e \end{bmatrix} = \begin{bmatrix} I_k & \bar{E} \end{bmatrix},$$

thus defining $\bar{E}$. Therefore, we get a basis of the nullspace up to a permutation of rows of

$$\begin{bmatrix} -\bar{E} \\ I_{kn-k+1} \end{bmatrix} = \begin{bmatrix} \begin{bmatrix} -I_k \otimes e_{n-1}^T & e \end{bmatrix} \\ I_{kn-k+1} \end{bmatrix}$$

We now immediately get k orthogonal columns. For a typical matrix V see Figure 3.1. We now explicitly find the V in Lemma 3.2.

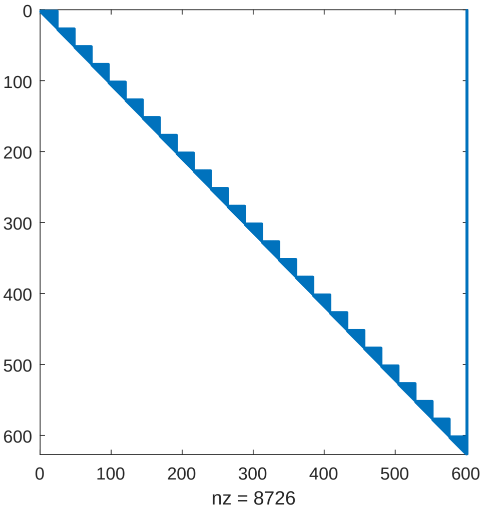

Figure 3.1: V matrix for k=20, n=20

Lemma 3.2. *Let k, n be given positive integers and from above let*

$$A = \left[I_k \otimes e_n^T\right], \; B = \left[-e_k \quad A\right].$$

Let $\mathcal{O} \in \mathbb{R}^{n-1\times n-1}$ be the strictly upper triangular matrix of ones of order $n-1$. Set

$$v = \left(\frac{1}{j+j^2}\right)_j \in \mathbb{R}^{n-1}, \; \beta = -1/\sqrt{n^2+nk}, \; \alpha = n\beta.$$

Let $\bar{\mathcal{O}} = -\mathcal{O}\,\mathrm{Diag}(v)$ with diagonal changed to $1:j.*v$. Then we have

$$V = \begin{bmatrix} 0 & \alpha \\ I_k \otimes \bar{\mathcal{O}} & \beta e \end{bmatrix} \in \mathbb{R}^{nk+1\times(n-1)k+1}, \quad V^T V = I, \; BV = 0.$$

Proof. The first $k(n-1)$ columns are clearly orthonormal and sum to zero by construction. they are constructed in exactly that way with the off diagonal elements above the diagonal all equal. Therefore they are also orthogonal to the last column. The α, β are found satisfying orthogonality as well as being in the nullspace of B. □

We leave open the question on how to exploit the structure of V to obtain efficient matrix-matrix multiplications of the form VRV^T needed in our algorithm.

Lemma 3.3. *Let k, n be given positive integers and from above let*

$$A = \left[I_k \otimes e_n^T\right], \; B = \left[-e_k \quad A\right].$$

Let $\mathcal{O} \in \mathbb{R}^{n-1\times n-1}$ be the strictly upper triangular matrix of ones of order $n-1$. Set

$$v = \left(\frac{1}{\sqrt{j+j^2}}\right)_j \in \mathbb{R}^{n-1}, \; \bar{v} = \left(\frac{j}{\sqrt{j+j^2}}\right)_j \in \mathbb{R}^{n-1}, \; \beta = -1/\sqrt{n^2+nk}, \textit{ and } \alpha = n\beta.$$

Let $\tilde{\mathcal{O}} = -\mathcal{O}\,\mathrm{Diag}(v) + \mathrm{Diag}(\bar{v})$ and set

$$\bar{\mathcal{O}} = \begin{bmatrix} -v^T \\ \tilde{\mathcal{O}} \end{bmatrix} = \begin{bmatrix} -v_1 & -v_2 & -v_3 & \cdots & -v_{n-1} \\ \bar{v}_1 & -v_2 & -v_3 & \cdots & -v_{n-1} \\ 0 & \bar{v}_2 & -v_3 & \cdots & -v_{n-1} \\ 0 & 0 & \bar{v}_3 & \cdots & -v_{n-1} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & \bar{v}_{n-1} \end{bmatrix}.$$

Then we have

$$V = \begin{bmatrix} 0 & \alpha \\ I_k \otimes \bar{\mathcal{O}} & \beta e \end{bmatrix} \in \mathbb{R}^{nk+1 \times (n-1)k+1}, \quad V^T V = I,\, BV = 0.$$

Proof. Denote the j-th column of V by V_j and define $J_s := \{j_1^s, j_2^s, \dots, j_{n-1}^s\}$, where $j_r^s = (n-1)(s-1)+r$. Notice that J_s is the index set of columns of V in s-th block. $j \in J_{k+1}$ means V_j is the last column of V.

We first prove that $V^T V = I$, i.e., column vectors of V is orthonormal. Let $i, j \in \{1, \dots, (n-1)k+1\}$. We consider the following cases:

If $j \leq (n-1)k$, then

$$V_j^T V_j = j v_j^2 + \bar{v}_j^2 = \frac{j}{j+j^2} + \frac{j^2}{j+j^2} = 1.$$

If $j = (n-1)k+1$, then

$$V_j^T V_j = \alpha^2 + nk\beta^2 = (n^2+nk)\beta^2 = 1.$$

Now let $i < j$. If $i, j \in J_s$ for some $s \leq k$. Then,

$$\begin{aligned} V_i^T V_j &= i v_i v_j - \bar{v}_i v_j \\ &= i \cdot \frac{1}{\sqrt{i+i^2}} \frac{1}{\sqrt{j+j^2}} - \frac{i}{\sqrt{i+i^2}} \frac{1}{\sqrt{j+j^2}} = 0. \end{aligned}$$

If $j = (n-1)k+1$. Then,

$$V_i^T V_j = -i v_i \beta + \bar{v}_i \beta = (-i v_i + i v_i)\beta = 0.$$

If $i \in J_s$, $j \in J_t$ with $s < t \leq k$. For each row, at least one of the vectors has 0 entry, so trivially $V_i^T V_j = 0$. This proves that $V^T V = I$.

Secondly, we observe $BV = 0$, i.e., $V \in \mathrm{null}(B)$. To this end, we will see that $BV_j = 0$ for each $j = 1, \dots, (n-1)k+1$. Fix $s \in \{1, \dots, k\}$. If $j = (n-1)k+1$,

$$\left(BV_j\right)_s = -\alpha + n\beta = -n\beta + n\beta = 0,$$

Now assume that $j \leq (n-1)k$. If $j \in J_s$, then

$$\left(BV_j\right)_s = -j v_j + \bar{v}_j = -j v_j + j v_j = 0, \text{ for each } i = 1, \dots, k.$$

Otherwise, trivially $\left(BV_j\right)_s = 0$. This justifies $BV = 0$. □

We continue and clarify the specific role of the arrow constraint.

Proposition 3.4. *The following holds:*

$$\left\{Y \in \mathbb{S}_+^{nk+1} : \operatorname{rank}(Y) = 1, \operatorname{arrow}(Y) = e_0\right\} = \left\{Y = \begin{pmatrix}1\\x\end{pmatrix}\begin{pmatrix}1\\x\end{pmatrix}^T : x \in \{0,1\}^{nk}\right\}.$$

Proof. ($\supseteq$): This is clear from the definitions.
($\subseteq$): Since Y is symmetric, positive semidefinite and has rank 1, there exists $x_0 \in \mathbb{R}$ and $x \in \mathbb{R}^{nk}$ such that $Y = \begin{pmatrix}x_0\\x\end{pmatrix}\begin{pmatrix}x_0\\x\end{pmatrix}^T$. Since $\operatorname{arrow}(Y) = e_0$, $x_0^2 = 1$ and $x \circ x = x_0 x$. If $x_0 = 1, x \in \{0,1\}^{nk}$; otherwise $x_0 = -1$ and $x \in \{0,-1\}^n$ and it is easy to verify that

$$\left\{\begin{pmatrix}1\\x\end{pmatrix}\begin{pmatrix}1\\x\end{pmatrix}^T : x \in \{0,1\}^{nk}\right\} = \left\{\begin{pmatrix}-1\\x\end{pmatrix}\begin{pmatrix}-1\\x\end{pmatrix}^T : x \in \{0,-1\}^{n}\right\}.$$

□

Therefore, the **SDP** reformulation is

$$(\textbf{SDP}) \qquad \begin{array}{rll} p^* = & \min_{Y\in\mathbb{S}^{nk+1}} & \langle \hat{D}, Y\rangle \\ & & \operatorname{arrow}(Y) = e_0 \\ & & \operatorname{rank}(Y) = 1 \\ & & KY = 0 \\ & & Y \succeq 0. \end{array}$$

And if we substitute using the *facial vector* $Y \leftarrow VRV^T$, then we can discard the $KY = 0$ constraint.

3.1.2 Relaxing the rank-1 constraint

Since the **NP**-hardness of the **SDP** formulation comes from the rank-1 constraint, we now relax the problem by deleting this constraint. The **SDP** relaxation of the above model is

$$(\textbf{SDP}\ \text{relax}) \qquad \begin{array}{rll} p^* = & \min_{Y\in\mathbb{S}^{nk+1}} & \langle \hat{D}, Y\rangle \\ & & \operatorname{arrow}(Y) = e_0 \\ & & KY = 0 \\ & & Y \succeq 0. \end{array} \tag{3.5}$$

However, the improved processing efficiency of the relaxation model trades off with the accuracy of solving the original model. The rank of an optimal Y now can be greater than 1. The idea now is to impose a “right” amount of redundant constraints in the **SDP** model that reduces the rank of an optimal solution as much as possible, without hurting the processing efficiency of the model too much.

3.1.3 The gangster constraint

The *gangster constraint* fixes at 0 (shoots holes at) certain entries in the matrix. The entries are given in the *gangster index, $\mathcal{J}$*. By abuse of notation, we allow one entry to be fixed at 1. The gangster constraint in our case comes from the linear constraint $Ax = e$ combined with the binary constraint on x. We let $S \circ T$ denote the Hadamard (elementwise) product.

Proposition 3.5. *Let x be feasible for BCQP. Then*

$$[A^TA-I]\circ xx^T=0,$$

and $A^TA-I\geq 0, xx^T\geq 0$. Define the gangster indices

$$\mathcal{J}:=\left\{ij:\left(A^TA-I\right)_{ij}>0\right\}.$$

The gangster constraint on Y in (3.5) is $Y_{00}=1$ and

$$\mathcal{J}(Y)=Y_{\mathcal{J}}=0\in\mathbb{R}^{|\mathcal{J}|}.$$

Proof. Recall that $x\in\mathbb{R}^{kn}_+$. We now use basic properties of the Kronecker product, e.g., [10], and see that

$$A=I_k\otimes e^T,\ e\in\mathbb{R}^n,\ A^T=I_k\otimes e,\ e\in\mathbb{R}^{nk},\quad A^TA=I_k\otimes ee^T,$$

i.e., A^TA has the following block diagonal structure, where $\star=1$:

$$\begin{bmatrix}\begin{bmatrix}1&\star&\star\\ \star&\ddots&\star\\ \star&\star&1\end{bmatrix}&&&\\ &\begin{bmatrix}1&\star&\star\\ \star&\ddots&\star\\ \star&\star&1\end{bmatrix}&&\\ &&\ddots&\\ &&&\begin{bmatrix}1&\star&\star\\ \star&\ddots&\star\\ \star&\star&1\end{bmatrix}\end{bmatrix}.$$

Therefore the columns of A are unit vectors and $\operatorname{Diag}(\operatorname{diag}(A^TA))=I_{kn}$. The nonnegativity results follow from the definition, as does $Y_{00}=1$.

Then

$$\begin{array}{rcl}Ax=e&\iff&A^TAx=A^Te=\operatorname{diag}(A^TA)\\ &\iff&A^TAx-Ix=A^Te-Ix=\operatorname{diag}(A^TA)-\operatorname{Diag}[\operatorname{diag}(A^TA)]x\\ &\iff&(A^TA-I)x=\operatorname{diag}(A^TA)\circ(e-x)=e-x\\ &\iff&(A^TA-I)xx^T=(e-x)x^T=ex^T-xx^T\\ &\iff&\operatorname{tr}[(A^TA-I)xx^T]=\operatorname{tr}[ex^T-xx^T]=\sum_{i=1}^{nk}x_i-x_i^2=0\\ &\iff&(A^TA-I)\circ xx^T=0.\end{array}$$

The final conclusion now follows from the nonnegativities in the Hadamard product. □

From Proposition 3.5, we see that the gangster indices J are the nonzeros of the matrix A^TA-I, i.e., the set of off-diagonal indices of the n-by-n diagonal blocks of the bottom right of Y_x. Our complete gangster index is $\hat{\mathcal{J}}:=\{(0,0)\}\cup\mathcal{J}$.

Now, the **SDP** relaxation model becomes

$$\begin{array}{rll}p^*=&\min_{Y\in\mathbb{S}^{nk+1}}&\langle\hat{D},Y\rangle\\ &&\operatorname{arrow}(Y)=e_0\\ &&\mathcal{G}_{\hat{J}}(Y)=e_0\\ &&KY=0\\ &&Y\succeq 0.\end{array}\tag{3.6}$$

3.2 Doubly nonnegative (DNN) relaxation

We now split the primal variable Y into two variables $\{Y, R\}$ and apply a doubly nonnegative relaxation to (3.6). This *natural splitting* uses the facial reduction obtained in (3.4) but with orthonormal columns chosen for the facial vector V.

Recall that the lifting for Y_x has the form $\begin{pmatrix}1\\x\end{pmatrix}\begin{pmatrix}1\\x\end{pmatrix}^T$, where $x \in \{0,1\}^{nk}$. Hence, we can impose the redundant element-wise $[0,1]$-bound constraint on Y, i.e: $0 \leq Y \leq 1$.

Recall that the constraint $KY = 0$ is redundant once we apply the facial reduction technique. The facial reduction naturally brings in a second primal variable $R \in \mathbb{S}_+^{nk+1-k}$, i.e., we use

$$Y \succeq 0, KY = 0 \iff Y = VRV^T, R \in \mathbb{S}_+^{nk+1-k}.$$

Next, we derive a redundant trace constraint on Y and transform it onto R.

Proposition 3.6. *We have*

$$\{Y \in \mathbb{S}^{nk+1} : KY = 0, \operatorname{arrow}(Y) = e_0\} \subseteq \{Y \in \mathbb{S}^{nk+1} : \operatorname{tr}(Y) = k+1\}.$$

Proof. Recall that $K := \begin{bmatrix}-e^T\\A^T\end{bmatrix}\begin{bmatrix}-e^T\\A^T\end{bmatrix}^T$. Since $\operatorname{null}(K) = \operatorname{null}\left(\begin{bmatrix}-e^T\\A^T\end{bmatrix}^T\right)$, we have

$$0 = KY \iff 0 = \begin{bmatrix}-1 & e^T & ... & 0^T\\ ... & ... & ... & ...\\ -1 & 0^T & ... & e^T\end{bmatrix}\begin{bmatrix}Y_{0,0} & ... & Y_{0,nk}\\ ... & ... & ...\\ Y_{nk,0} & ... & Y_{nk,nk}\end{bmatrix}.$$

By expanding the first column of the product, we get $\sum_{i=1}^n Y_{jn+i,0} = 1, \forall j \in \{0, ..., k-1\}$. Since $\operatorname{arrow}(Y) = e_0$, this implies that $\operatorname{tr}(Y) = Y_{0,0} + \sum_{j=1}^k \sum_{i=1}^n Y_{jn+i,0} = 1 + k$. □

Now, the facial constraint says that $1 + k = \operatorname{tr}(Y) = \operatorname{tr}(VRV^T) = \operatorname{tr}(RV^TV) = \operatorname{tr}(R)$, since we choose the facial vector V to have orthonormal columns.

Next, we incorporate all these constraints into the **SDP** relaxation model to form the **DNN** relaxation model. Define the two sets

$$\mathcal{Y} := \{Y \in \mathbb{S}^{nk+1} : \mathcal{G}_{\hat{J}}(Y) = Y_{\hat{J}} = e_0, \operatorname{arrow}(Y) = e_0, 0 \leq Y \leq 1\},\ \mathcal{R} := \{R \in \mathbb{S}_+^{nk+1-k} : \operatorname{tr}(R) = k+1\}.$$

Thus, the **DNN** relaxation model is:

$$(\textbf{DNN}) \qquad \begin{array}{cl} \min_{R,Y} & \langle \hat{D}, Y\rangle \\ \text{s.t.} & Y = VRV^T \\ & Y \in \mathcal{Y} \\ & R \in \mathcal{R} \end{array} \tag{3.7}$$

Observe that every feasible Y is nonnegative element-wise and every feasible R is ***PSD*** . Hence, this is a **DNN** relaxation. The splitting allows for the two cones to be handled separately. Combining them into one and applying e.g., an interior point approach is very costly.

3.2.1 Optimality conditions

Note that the *linear* mapping $\mathcal{M}(Y,R) := Y - VRV^T$ is surjective, the interior of the closed convex feasible set $\operatorname{int}(\mathcal{Y}\times\mathcal{R}) \neq \emptyset$, and the normal cone at a feasible pair (Y,R) :?" satisfies

$$N_{\mathcal{Y}\times\mathcal{R}}(Y,R) = N_{\mathcal{Y}}(Y) \times N_{\mathcal{R}}(R).$$

The corresponding Lagrangian with dual variable Z is

$$\mathcal{L}(Y,R,Z) = \langle \hat{D}, Y\rangle + \langle Z, Y - VRV^T\rangle + \iota_{\mathcal{Y}}(Y) + \iota_{\mathcal{R}}(R),$$

where $\iota_S(\cdot)$ is the *indicator function* for the set S. Therefore the first-order optimality conditions to the problem in (3.7) yields that a primal-dual pair (Y,R,Z) is optimal if, and only if,

$$\begin{aligned}
&Y = VRV^T, \quad R\in\mathcal{R},\ Y\in\mathcal{Y} && \text{(primal feasibility)} && (3.8a)\\
&0 \in -V^T ZV + \mathcal{N}_{\mathcal{R}}(R) && \text{(dual } R \text{ feasibility)} && (3.8b)\\
&0 \in \hat{D} + Z + \mathcal{N}_{\mathcal{Y}}(Y) && \text{(dual } Y \text{ feasibility)} && (3.8c)
\end{aligned}$$

By the definition of the normal cone, we can easily obtain the following Proposition 3.7.

Proposition 3.7 (characterization of optimality for **DNN** in (3.7))**.** *The primal-dual pair* (R,Y,Z) *is optimal for* (3.7) *if, and only if,* (3.8) *holds if, and only if,*

$$\begin{aligned}
R &= \mathcal{P}_{\mathcal{R}}(R + V^T ZV) && (3.9a)\\
Y &= \mathcal{P}_{\mathcal{Y}}(Y - \hat{D} - Z) && (3.9b)\\
Y &= VRV^T && (3.9c)
\end{aligned}$$

4 sADMM algorithm

The augmented Lagrangian corresponding to (3.7) with parameter $\beta > 0$ is

$$\mathcal{L}_\beta(Y,R,Z) := \langle \hat{D}, Y\rangle + \langle Z, Y - VRV^T\rangle + \frac{\beta}{2}\|Y - VRV^T\|_F^2 + \iota_{\mathcal{Y}}(Y) + \iota_{\mathcal{R}}(R). \qquad (4.1)$$

To solve the problem in (3.7), we will use the symmetric alternating directions method of multipliers **sADMM** that has intermediate updates of multipliers. It updates the dual variable twice: once after the R-update and then again after the Y-update. Hence, both the R-update and the Y-update take into account newly updated dual variable information. Let $Y_0 \in \mathbb{S}^{nk+1}$ and let $Z_0 \in \mathbb{S}^{nk+1}$. Update $(\forall k \in \mathbb{N})$:

$$\begin{array}{rcl}
R_{k+1} &=& \operatorname{argmin}_{R\in\mathbb{S}^{nk+1-k}} \mathcal{L}_\beta(R, Y_k, Z_k)\\
Z_{k+\frac12} &=& Z_k + \beta(Y_k - VR_{k+1}V^T)\\
Y_{k+1} &=& \operatorname{argmin}_{Y\in\mathbb{S}^{nk+1}} \mathcal{L}_\beta(R_{k+1}, Y, Z_{k+\frac12})\\
Z_{k+1} &=& Z_{k+\frac12} + \beta(Y_{k+1} - VR_{k+1}V^T).
\end{array} \qquad (4.2)$$

In our **DNN** model (3.7), the objective function is continuous and the feasible set is compact. By the extreme value theorem, an optimal primal pair (Y^*, R^*) always exists. As seen above, the constraint is linear and surjective and strong duality holds. (See the optimality conditions in Section 3.2.1). In fact, in our application we modify the dual multiplier update using a projection, see Lemma 4.1 and Algorithm 4.1.

Explicit Primal updates for R, Y

We start with using a spectral decomposition of M below to get the:

$$\begin{array}{rcll} R-\text{update} & = & \text{argmin}_{R\in\mathbb{S}^{nk+1-k}}\, \mathcal{L}_\beta(R, Y^k, Z^k) & \\ & = & \text{argmin}_{R\in\mathcal{R}}\, \|Y_k - VRV^T + \frac{1}{\beta}Z_k\|_F^2 & \text{by completing the square} \\ & = & \text{argmin}_{R\in\mathcal{R}}\, \|V^TY_kV - R + \frac{1}{\beta}V^TZ_kV\|_F^2 & \text{since } V^TV = I \\ & = & \text{argmin}_{R\in\mathcal{R}}\, \|R - V^T(Y_k + \frac{1}{\beta}Z_k)V\|_F^2 & \\ & = & \mathcal{P}_\mathcal{R}[V^T(Y_k + \frac{1}{\beta}Z_k)V] & =: \mathcal{P}_\mathcal{R}(M);\ M = U\,\text{Diag}(d)U^T \\ & = & U\,\text{Diag}[\mathcal{P}_{\Delta_{k+1}}(d)]U^T & \end{array}$$

where $\mathcal{P}_{\Delta_{k+1}}$ denotes the projection onto the *simplex* $\Delta_{k+1} := \{x \in \mathbb{R}^n_+ : \langle e, x\rangle = 1 + k\}$, see e.g., [5].

Next for the

$$\begin{array}{rcll} \text{Y}-\text{update} & = & \text{argmin}_{Y\in\mathbb{S}^{nk+1}}\, \mathcal{L}_\beta(R_{k+1}, Y, Z_{k+\frac{1}{2}}) & \\ & = & \text{argmin}_{Y\in\mathcal{Y}}\, \|Y - [VR_{k+1}V^T - \frac{1}{\beta}(\hat{D} + Z_{k+\frac{1}{2}})]\|_F^2 & \text{by completing the square} \\ & = & \mathcal{P}_\mathcal{Y}\left(VR_{k+1}V^T - \frac{1}{\beta}(\hat{D} + Z_{k+\frac{1}{2}})\right) & \\ & = & \mathcal{P}_{arrowbox}\left(\mathcal{G}_{\hat{\mathcal{J}}}[VR_{k+1}V^T - \frac{1}{\beta}(\hat{D} + Z_{k+\frac{1}{2}})]\right) & \end{array}$$

where $\mathcal{G}_{\hat{\mathcal{J}}}$ is the gangster constraint and $\mathcal{P}_{arrowbox}$ projects onto the polyhedral set $\{Y \in \mathbb{S}^{nk+1} : Y_{ij} \in [0, 1], \text{arrow}(Y) = e_0\}$.

Dual updates

The correct choice of the Lagrange dual multiplier Z is important in the progress of the algorithm and in obtaining strong lower bounds. In addition, if the set of dual multipliers for all iterations is compact, then it indicates the stability of the primal problem. If an optimal Z^* for (3.7) is known in advance, then there is no need to impose the primal feasibility constraint $Y = VRV^T$. Hence, following the idea of exploiting redundant constraints, we aim to identify certain properties of an optimal dual multiplier and impose that property at each iteration of our algorithm.

Lemma 4.1. *Let*

$$\mathcal{Z}_A := \left\{ Z \in \mathbb{S}^{nk+1} : (Z + \hat{D})_{i,i} = 0, (Z + \hat{D})_{0,i} = 0, (Z + \hat{D})_{i,0} = 0, i = 1, ..., nk \right\}.$$

Let (Y^*, R^*, Z^*) be an optimal primal-dual pair for the ***DNN*** *in (3.7). Then, $Z^* \in \mathcal{Z}_A$.*

Proof. The proof of this fact uses the dual Y feasibility condition (3.8c) and a reformulation of the Y-feasible set. The details are in [6, Thm 2.14] and [4]. □

In view of Lemma 4.1 we propose the following modification of the symmetric **ADMM** algorithm, e.g., [7]. Our modification is in the way we update the multiplier. At every intial or intermediate update of the multiplier we project the dual variable onto $\mathcal{Z}_A$, i.e:

- $Z_{j+\frac{1}{2}} := Z_j + \beta\mathcal{P}_{\mathcal{Z}_A}(Y_j - VR_{j+1}V^T)$;
- $Z_{j+1} := Z_{j+\frac{1}{2}} + \beta\mathcal{P}_{\mathcal{Z}_A}(Y_{j+1} - VR_{j+1}V^T)$.

Algorithm 4.1 sADMM, modified symmetric **ADMM**

```
Initialization: $j = 0, Y_j = 0 \in S^{nk+1}, Z_j = P_{Z_A}(0), \beta = \max(\lfloor \frac{nk+1}{k} \rfloor, 1), \gamma = 0.9$
while termination criteria are not met do
    $R_{j+1} = U\,\mathrm{Diag}[P_{\Delta_{j+1}}(d)]U^T$ where $U\,\mathrm{Diag}(d)U^T = \mathrm{eig}(V^T(Y_j + \frac{1}{\beta}Z_j)V)$
    $Z_{j+\frac{1}{2}} = Z_j + \gamma\beta P_{Z_A}(Y_j - VR_{j+1}V^T)$
    $Y_{j+1} = P_{box}[\mathcal{G}_{\hat{J}}(VR_{j+1}V^T - \frac{1}{\beta}(\hat{D} + Z_{j+\frac{1}{2}}))]$
    $Z_{j+1} = Z_{j+\frac{1}{2}} + \gamma\beta P_{Z_A}(Y_{j+1} - VR_{j+1}V^T)$
    $j = j + 1$
end while
```

Note that a convergence proof using the modified updates is given in [6, Thm 3.2]. Therefore, in view of the **ADMM** updates (4.2) we propose the following Algorithm 4.1 with modified Z updates.

Remark 4.2. *In passing, we point out that we could choose any $\gamma \in (0,1)$ and $\beta > 0$. Theoretically this is all what we need. In our numerical experiments for Algorithm 4.1 we used an adaptive β based on the discussion in Section 4.3.1.*

4.1 Bounding and duality gaps

Strong upper and lower bounds allow for early stopping conditions as well as proving optimality.

4.1.1 Lower bounds

The Lagrangian dual function to the **DNN** model $g : \mathbb{S}^{nk+1} \to \mathbb{R}$ is

$$\begin{array}{rcl} g(Z) & = & \min_{R\in\mathcal{R}, Y\in\mathcal{Y}} \langle \hat{D}, Y\rangle + \langle Z, Y - VRV^T\rangle \\ & = & \min_{Y\in\mathcal{Y}, R\in\mathcal{R}} \langle \hat{D} + Z, Y\rangle - \langle Z, VRV^T\rangle \\ & = & \min_{Y\in\mathcal{Y}} \langle \hat{D} + Z, Y\rangle + \min_{R\in\mathcal{R}}(-\langle V^TZV, R\rangle) \\ & = & \min_{Y\in\mathcal{Y}} \langle \hat{D} + Z, Y\rangle - \max_{R\in\mathcal{R}} \langle V^TZV, R\rangle \\ & = & \min_{Y\in\mathcal{Y}} \langle \hat{D} + Z, Y\rangle - \max_{\|v\|^2=(k+1)} v^TV^TZVv \\ & = & \min_{Y\in\mathcal{Y}} \langle \hat{D} + Z, Y\rangle - (k+1)\lambda_{max}(V^TZV). \end{array}$$

Hence, at iteration k, a lower bound to the optimal value of the **DNN** model is

$$g(Z_k) = \min_{Y\in\mathcal{Y}} \langle \hat{D} + Z_k, Y\rangle - (k+1)\lambda_{max}(V^TZ_kV).$$

4.1.2 Upper bounds

As for the upper bound, we consider two strategies for finding feasible solutions to the **BCQP**. The 0-column approach is to take the first column $Y(1 : \mathit{end}, 0)$ and compute its nearest feasible solution to **BCQP**. It is equivalent to signal only the maximum weight index for each consecutive block of length n. The proof is in [4, section 3.2.2].

Alternatively, we use the dominant eigenvector of Y. and compute its nearest feasible solution to **BCQP**. It is again equivalent to signal only the maximum weight index for each consecutive block of length n.

Then, we compare the objective values for both approaches and select the upper bound with smaller magnitude. The relative duality gap at the current iterate k is defined to be $\frac{UB_k - LB_k}{|UB_k|+|LB_k|+1}$ where UB_k denotes upper bound at the current iterate and LB_k denotes lower bound at the current iterate.

4.2 Stopping criterion

By Proposition 3.7, we can define the primal and dual residuals of the **sADMM** algorithm at iterate k as follows:

- Primal residual $r_k := Y_k - VR_kV^T$;
- Dual-R residual $s_k^R := R_k - \mathcal{P}_{\mathcal{R}}\left(R_k + V^T Z_k V\right)$;
- Dual-Y residual $s_k^Y := Y_k - \mathcal{P}_{\mathcal{Y}}\left(Y_k - \hat{D} - Z_{k+\frac{1}{2}}\right)$.

We terminate the algorithm once one of the following conditions is satisfied:

- The maximum number of iterations $(maxiter) := 10^4 + k(nk+1)$ is reached;
- The relative duality gap is less or equal to $\epsilon := 10^{-5}$;
- $KKTres := \max\{r_k, s_k^R, s_k^Y\} < \eta := 10^{-5}$;
- Both the least upper bound and the greatest lower bound have not changed for boundCounterMax:=200 times.

4.3 Speed-up

4.3.1 Adaptive step size

We apply the heuristic idea presented in [3], namely we bound the gap between the primal and dual residual norms within a factor of $\mu := 2$ as they converge to 0. This guarantees that they converge to 0 at about the same rate and one residual will not overshoot the other residual by too much. Since a large penalty β prioritizes primal feasibility over dual feasibility and a small penalty β prioritizes dual feasibility over primal feasibility, we scale β by a factor of $\tau_{inc} := 2$ if the primal residual overshoots the dual residual by a factor of μ and scale β down by a factor of $\tau_{dec} := 2$ if the dual residual overshoots the primal residual by a factor of μ. Otherwise, we keep β unchanged. Specifically,

$$\beta_{j+1} := \begin{cases} \tau^{incr}\beta_j, & \|r_j\|_2 > \mu\|s_j\|_2; \\ \frac{\beta_j}{\tau^{decr}}, & \|s_j\|_2 > \mu\|r_j\|_2; \\ \beta_j, & \text{otherwise.} \end{cases}$$

4.3.2 Transformation and scaling

In this section, we consider translating and scaling the objective function i.e., $\hat{D}$. Define the orthogonal projection map $P_V := VV^T$. Then,

$$\begin{array}{rcl} \langle \hat{D}, Y\rangle & := & \langle \hat{D} + \alpha I, Y\rangle - (n+1)\alpha \\ & = & \langle \hat{D} + \alpha I, P_V Y P_V\rangle - (n+1)\alpha \\ & = & \langle (P_V \hat{D} P_V + \alpha I), Y\rangle - (n+1)\alpha. \end{array} \tag{4.3}$$

Hence,

$$\begin{aligned} \langle \hat{D}, Y\rangle \text{ is minimized } \quad &\iff \delta\langle \hat{D}, Y\rangle = \langle \delta(P_V\hat{D}P_V + \alpha I), Y\rangle - (n+1)\delta\alpha \text{ is minimized} \\ &\iff \langle \delta(P_V\hat{D}P_V + \alpha I), Y\rangle \text{ is minimized.} \end{aligned}$$

This lets us transform $\hat{D}$ into $\delta(P_V\hat{D}P_V + \alpha I)$ without changing the optimum solutions.

Numerical experiments show that once we scale $\hat{D}$ by some $\delta < 0$, the convergence becomes faster for the aforementioned input data distributions. There seems to be an optimal δ that minimizes the number of iterations for convergence.

4.4 Numerical Tests

We now illustrate the efficiency of our algorithm on medium and large scale randomly generated problems. We used MATLAB version 2022a on a greyling22 Dell R840 4 Intel Xeon Gold 6254, with 3.10 GHz, 72 core and 384 GB.

Both Tables 4.1 and 4.2 illustrate the efficiency and surprising success of our algorithm. Table 4.1, page 17, provides the running time and relative gap comparisons of **sADMM** compared to the MOSEK solver in CVX MATLAB.

Specifications			Time (s)		Relative duality gap	
d	n	k	**sADMM**	Mosek	**sADMM**	Mosek
2	7	5	2.33e-01	3.66e-01	9.80e-08	2.41e-09
2	8	6	3.90e-01	6.94e-01	2.76e-10	5.91e-11
2	9	7	3.53e-01	1.30e+00	6.59e-07	1.55e-11
2	10	8	3.75e-01	3.92e+00	4.82e-08	4.96e-12
2	11	9	4.63e-01	1.30e+01	1.92e-09	2.21e-12
2	12	10	5.41e-01	3.09e+01	9.32e-10	8.41e-10
2	13	11	7.22e-01	7.31e+01	1.83e-08	2.94e-11

Table 4.1: running time and relative gap comparisons

Table 4.2, page 18 shows the scalability of the **sADMM** algorithm for data of large size.

d	n	k	Time(s)	**KKT** residual	Relative duality gap
3	3	3	2.36e-02	2.20e-07	7.52e-15
4	4	4	1.38e-01	3.10e-08	9.95e-17
5	5	5	1.80e-01	7.02e-09	3.42e-16
6	6	6	3.06e-01	1.89e-08	9.09e-15
7	7	7	4.79e-01	1.19e-06	1.65e-14
8	8	8	3.16e-01	1.51e-06	5.83e-15
9	9	9	5.11e-01	1.43e-07	1.42e-14
10	10	10	5.46e-01	1.51e-07	1.46e-14
11	11	11	2.71e-01	7.38e-09	3.01e-14
12	12	12	1.01e+00	2.34e-08	2.02e-14
13	13	13	1.48e+00	4.76e-09	1.64e-14
14	14	14	2.98e+00	1.21e-06	2.75e-14
15	15	15	1.54e+00	9.83e-08	1.10e-14
16	16	16	1.27e+00	6.76e-08	1.70e-14
17	17	17	1.80e+00	1.36e-08	2.46e-14
18	18	18	2.44e+00	2.93e-06	3.17e-15
19	19	19	3.19e+00	9.19e-10	1.15e-14
20	20	20	5.53e+00	1.56e-09	4.15e-15
21	21	21	6.25e+00	1.53e-08	3.86e-14
22	22	22	1.38e+01	2.67e-06	1.32e-14
23	23	23	1.35e+01	4.16e-09	1.42e-14
24	24	24	1.64e+01	8.28e-07	3.56e-14
25	25	25	2.72e+01	1.73e-09	8.10e-16

Table 4.2: scalability **sADMM** algorithm for data of large size

5 Multiple Optimal Solutions and Duality Gaps

We now see that *multiple optimal* solutions for the original hard problem can lead to a duality gap between the original NP-hard problem and the **DNN** relaxation.

5.1 Criteria for Duality Gaps

To find duality gaps for **SDP** relaxations, we want to find points outside of the convex hull of the lifted vertices. The following Lemma 5.1 and Corollary 5.2 provides this between a general hard problem with multiple optimal solutions and its **DNN** relaxation.

Lemma 5.1. *Let $\{x_i\}_{i=1}^n \subset \mathbb{R}^n_+$ be a linearly independent set with $\sum_i x_i > 0$. Define the lifted vertices and barycenter, respectively,*

$$\left\{X_i = x_i x_i^T\right\}_{i=1}^n \subset \mathbb{S}^n, \quad \hat{X} := \frac{1}{n}\sum_{i=1}^n X_i.$$

Then

$$\hat{X} \in \mathbb{S}^n_{++} \cap \mathbb{R}^n_{++} \quad (= \operatorname{int} \boldsymbol{DNN}).$$

Proof. We note that $X_i \succeq 0, \forall i$ and so $\hat{X} \succeq 0$ as well. To obtain a contradiction, suppose that $0 = \hat{X}v$, for some $0 \neq v \in \mathbb{R}^n$. Then

$$0 = v^T \hat{X} v = v^T \sum_i X_i v \implies 0 = v^T X_i v, \forall i \implies (v^T x_i)^2 = 0, \forall i \implies v = 0,$$

by the linear independence assumption; thus contradicting $v \neq 0$. That $\hat{X} \in \mathbb{R}^n_{++}$ is clear from the hypothesis. □

Corollary 5.2. *Suppose that the hypotheses of Lemma 5.1 hold. Moreover, suppose that the points $x_i, i = 1, \ldots, n$, are optimal for a given hard minimization problem*

$$(P) \quad p^* = \min \left\{ x^T Q x \, : \, x \in \{0,1\}^n \right\},$$

*with $p^* = x_i^T Q x_i, \forall i$. Moreover, suppose that there exists a feasible y with $y \neq x_i, \forall i$, and y not optimal, $y \in \{0,1\}^n, y^T Q y > p^*$. Then the **DNN** relaxation has feasible points $Y = yy^T, Z$ such that*

$$\operatorname{tr} YQ > p^* > \operatorname{tr} ZQ,$$

i.e., Z yields a duality gap.

Proof. From Lemma 5.1 we have that the barycenter satisfies both $\hat{X} \succ 0, \hat{X} > 0$. Note that $\operatorname{tr} YQ = y^T Q y > p^* = \operatorname{tr} \hat{X} Q$. Therefore, $\operatorname{tr}(\hat{X} - Y)Q < 0$, and for $\epsilon > 0$,

$$\operatorname{tr}(\hat{X} + \epsilon(\hat{X} - Y)Q = p^* + \epsilon \operatorname{tr}(X - Y)Q < p^*.$$

Moreover, the line segment $[Y, \hat{X} + \epsilon(\hat{X} - Y)]$ is feasible for the **SDP** relaxation for small enough $\epsilon > 0$ by $\hat{X} \in \operatorname{int} \mathbf{DNN}$. Therefore, we set $Z_\epsilon = \hat{X} + \epsilon(\hat{X} - Y), 0 < \epsilon << 1$ and obtain a duality gap. □

We can extend this theory to problems with general linear constraints $Ax = b$ by using **FR**. We now specifically extend it to our **BCQP** in (2.8). We need $nk + 1 - k$ linearly independent optimal points. This can be obtained when we choose $k >> n$. Recall the matrix K in (3.2) used for facial reduction and the facially reduced **DNN** relaxation in (3.7).

Corollary 5.3. *We consider the **BCQP** with optimal value p^*, and the **DNN** relaxation in (3.7). Let*

$$\left\{ y_i = \begin{pmatrix} 1 \\ x_i \end{pmatrix} \right\}_{i=1}^{nk+1-k} \subset \mathbb{R}^{nk+1}_+$$

*be a linearly independent set that are optimal for **BCQP** and with $\sum_i y_i > 0$. Define the lifted vertices and barycenter, respectively,*

$$\left\{ Y_i = y_i y_i^T \right\}_i, \forall i, \quad \hat{Y} := \frac{1}{n} \sum_{i=1}^{n} Y_i.$$

*Moreover, suppose that there exists a feasible $\bar{x}$ for **BCQP** that is not optimal. Then*

$$\hat{Y} = V \hat{R} V^T \succeq 0, \; \hat{Y} > 0, \; \hat{R} \succ 0.$$

And there exists $Z = V R_Z V^T, R_Z \succ 0$ with optimal value $\operatorname{tr} DZ < p^*$, yielding a duality gap.

Proof. First note that incident vectors are feasible for the linear constraints and this guarantees that we have enough feasible points to guarantee that the barycenter satisfies $\hat{Y} > 0$. All lifted feasible points of the relaxation are in the minimal face and have a corresponding matrix R for the facial reduction $Y = VRV^T$. Since $R \succ 0$ after the **FR**, we can apply the same proof as in Corollary 5.2. In addition, note that the linear constraints, the arrow constraint and gangster constraints, remain satisfied in the line formed from two feasible points. □

5.2 Examples

We illustrate the above theory with some specific problems with special structure that have multiple optimal solutions for the original **NP**-hard problem. We see that a duality gap exists between the optimal solution of the original problem and the **DNN** relaxation.

Example 5.4. *first, we consider the simplest case where $n = k = 2$. Define $S_1 := \left\{ \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 10 \\ 0 \end{bmatrix} \right\}$ and $S_2 := \left\{ \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \begin{bmatrix} 0 \\ -1 \end{bmatrix} \right\}$. Clearly, the optimal solution of the simplified Wasserstein barycenter problem with respect to this data distribution is to pick the first point of S_1 and either the first or the second point of S_2. The former selection matches the solution vector $x = \begin{bmatrix} 1 \\ 0 \\ 1 \\ 0 \end{bmatrix}$ corresponding to the lifted matrix $\begin{bmatrix} 1 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$ of the **DNN** formulation. The latter selection matches the solution vector $x = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 1 \end{bmatrix}$ corresponding to the lifted matrix $\begin{bmatrix} 1 & 1 & 0 & 0 & 1 \\ 1 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 \end{bmatrix}$ of the **DNN** formulation. Observe that the convex combination of these two matrices with coefficients $\{0.5, 0.5\}$ is*

$$\tilde{Y} = \begin{bmatrix} 1 & 1 & 0 & 0.5 & 0.5 \\ 1 & 1 & 0 & 0.5 & 0.5 \\ 0 & 0 & 0 & 0 & 0 \\ 0.5 & 0.5 & 0 & 0.5 & 0 \\ 0.5 & 0.5 & 0 & 0 & 0.5 \end{bmatrix}$$

whose facially reduced component $\tilde{R} = \begin{bmatrix} 2 & 1 & 0 \\ 1 & 0.5 & 0 \\ 0 & 0 & 0.5 \end{bmatrix}$ has rank 2.

Recall the Lagrangian dual function that we used in section 4.1.1 for computing the lower bound:

$$g(Z) = \min_{Y \in \mathcal{Y}} \langle \hat{D} + Z, Y \rangle - \max_{R \in \mathcal{R}} \langle V^T Z V, R \rangle.$$

With $\tilde{Z} := \begin{bmatrix} -0.3619 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1.3699 & -1 & -1 \\ 0 & 1.3699 & 0 & -1.5826 & -1.5826 \\ 0 & -1 & -1.5826 & 0 & 0.7873 \\ 0 & -1 & -1.5826 & 0.7873 & 0 \end{bmatrix}$*, the* ***sADMM*** *algorithm terminates with a* ***KKT*** *residual of 8.9157e-11.*

With $\hat{D} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 101 & 101 \\ 0 & 1 & 101 & 0 & 0 \\ 0 & 1 & 101 & 0 & 0 \end{bmatrix}$*, we have* $g(\hat{Z}) = 1.6381 < 2 = \langle \hat{D}, \tilde{Y} \rangle$*, admitting a strictly positive duality gap.*

Example 5.5 (Odd wheels)**.** *We next present another input data distribution for which the duality gap between the optimal value of the* ***BCQP*** *formulation and the Lagrangian dual value is non-trivial. The issue is again the non-uniqueness of the optimal solutions and the* ***sADMM*** *algorithm fails to break ties among them.*

The data distributions compose of a wheel of wheels, i.e., a wheel with an odd number of sets each of which is a wheel. Hence we call it an odd wheel. Given problem size parameters (k, n, d)*, define*

- $\theta_k := \frac{2\pi}{k}$.
- *a set of* k *centroids encoded by a matrix* $C \in \mathbb{R}^{k\times 2}$ *such that*

$$C(i,:) = \begin{bmatrix}\cos(i-1)\theta_k & \sin(i-1)\theta_k\end{bmatrix}, i = 1, ..., k.$$

- *the radius of each cluster* $r_k := \frac{\sqrt{\cos(\theta_k - 1)^2 + \sin\theta_k^2}}{4}$.
- *the set of input points encoded by a matrix* $P := (C \otimes e) + r_k(e \otimes C) \in \mathbb{R}^{k^2,2}$.

When k *is odd, there exists more than one optimal solution. A simple example with* $k = 3 = n$ *follows in Figure 5.1.*

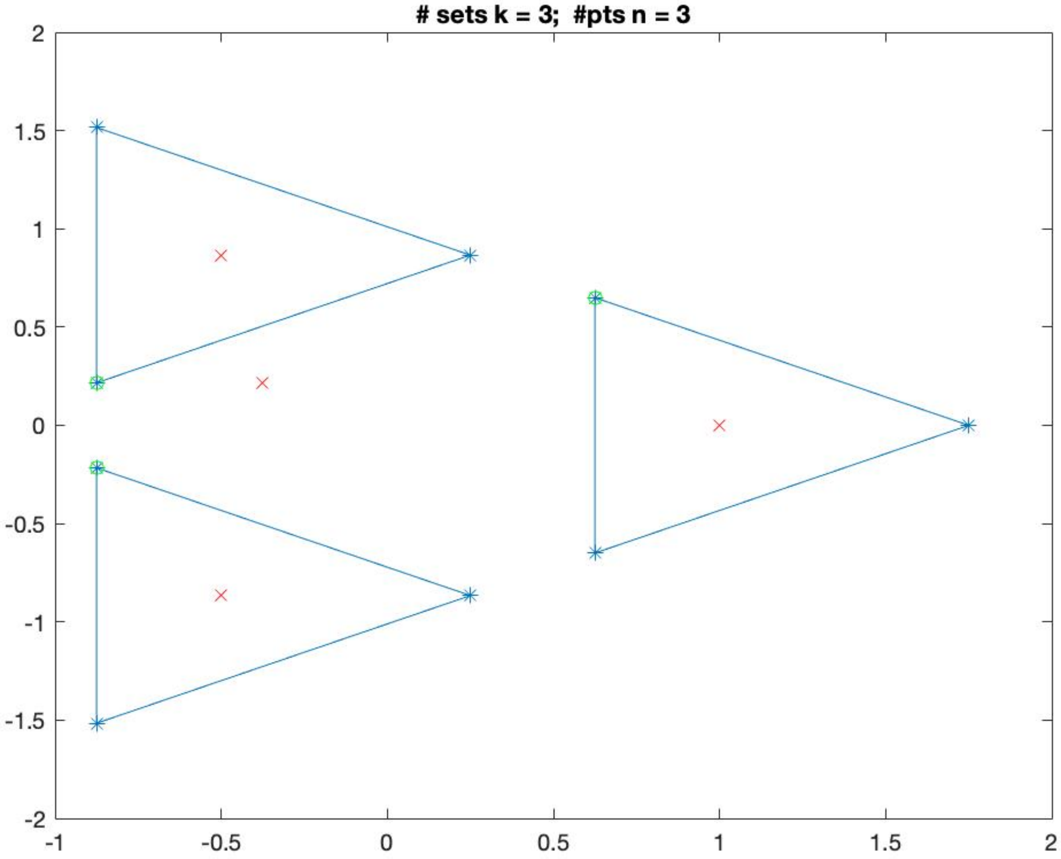

Figure 5.1: k=3=n

A simple inspection of the picture shows that reflecting the selected green points along the x-axis gives another optimal solution. In fact, for this example, six different optimal solutions exist.

However, when k is even, only one optimal solution exists and the duality gap becomes trivial. An example with $k = 6 = n$ follows in Figure 5.2.

6 Conclusion

In this paper we presented a strategy for solving NP-hard binary quadratic problems. This involves formulating a **DNN** relaxation, **FR** that gives rise to a natural splitting for a symmetric alternating directions method of multipliers **sADMM** with intermediate update of multipliers and strong upper and lower bounding techniques. We applied this to the **NP**-hard computational problem called the Simplified Wasserstein Barycenter problem.

Surprisingly, for the random problems we generated the gap between bounds was zero and we were able to provably solve the original NP-hard optimization problem. However, for specially constructed input data that had multiple optimal solutions, the algorithm had difficulty breaking

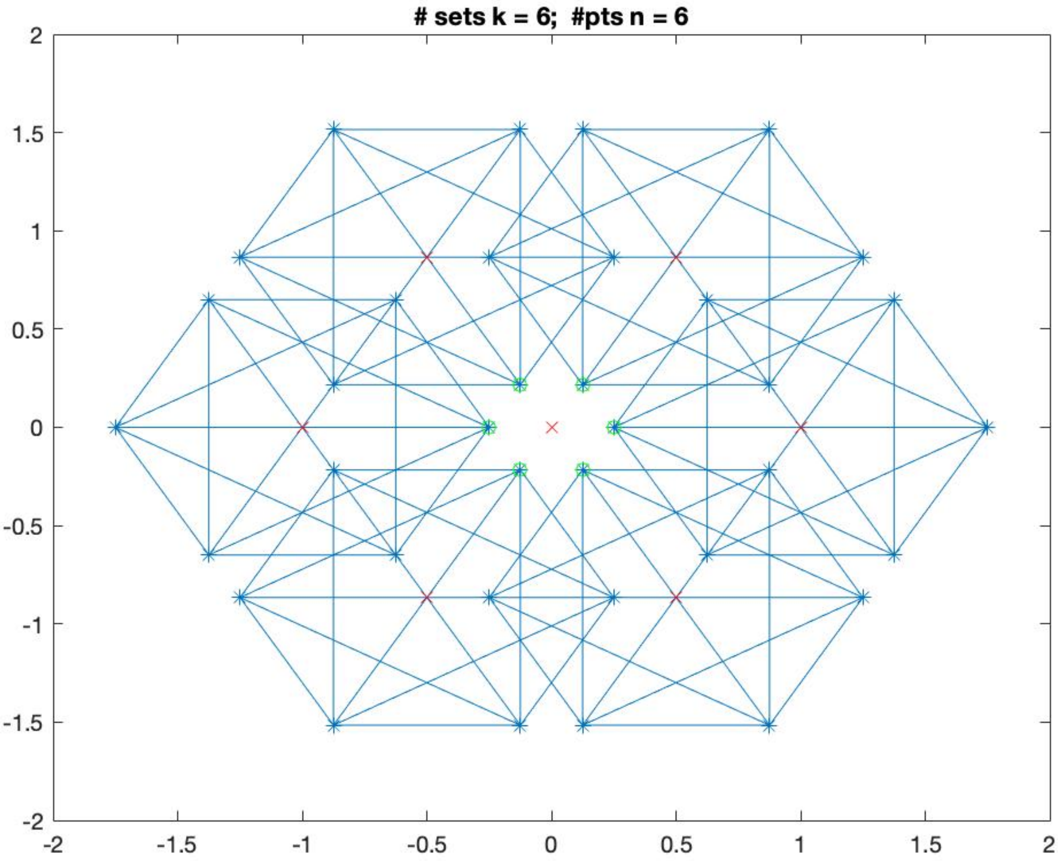

Figure 5.2: k=6=n

ties and the result was gaps between lower and upper bounds, i.e., the original Wasserstein problem was not solved to optimality. We provided theoretical proof that such problems have a duality gap.

As for future research, we want to better understand the theoretical reasons for the positive duality gaps and find more classes of problems where this occurs. In addition, we want to understand what happens under small perturbations to problems with duality gaps, i.e., if the gaps can be closed with perturbations.

Index

References

[1] J. Altschuler and E. Boix-Adsera, *Wasserstein barycenters can be computed in polynomial time in fixed dimension*, tech. rep., 2020. 3

[2] J. Altschuler and E. Boix-Adserà, *Wasserstein barycenters are NP-hard to compute*, SIAM J. Math. Data Sci., 4 (2022), pp. 179–203. 5

[3] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, *Distributed optimization and statistical learning via the alternating direction method of multipliers*, Found. Trends Machine Learning, 3 (2011), pp. 1–122. 16

[4] F. Burkowski, H. Im, and H. Wolkowicz, *A Peaceman-Rachford splitting method for the protein side-chain positioning problem*, tech. rep., University of Waterloo, Waterloo, Ontario, 2022. arxiv.org/abs/2009.01450,21. 14, 15

[5] Y. Chen and X. Ye, *Projection onto a simplex*, 2011. 14

[6] N. Graham, H. Hu, H. Im, X. Li, and H. Wolkowicz, *A restricted dual Peaceman-Rachford splitting method for a strengthened DNN relaxation for QAP*, INFORMS J. Comput., 34 (2022), pp. 2125–2143. 14, 15

[7] B. He, H. Liu, Z. Wang, and X. Yuan, *A strictly contractive Peaceman-Rachford splitting method for convex programming*, SIAM J. Optim., 24 (2014), pp. 1011–1040. 14

[8] V. Panaretos and Y. Zemel, *Statistical aspects of wasserstein distances*, Annual Review of Statistics and Its Application, 6 (2019), pp. 405–431. 3

[9] P. Pardalos and S. Vavasis, *Quadratic programming with one negative eigenvalue is NP-hard*, J. Global Optim., 1 (1991), pp. 15–22. 6

[10] K. Schaecke, *Essay on: The Kronecker product*, Master's thesis, University of Waterloo, 2004. 11

[11] I. Schoenberg, *Metric spaces and positive definite functions*, Trans. Amer. Math. Soc., 44 (1938), pp. 522–536. 4