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

Abdo Alfakih† Woosuk L. Jung† Walaa M. Moursi† Henry Wolkowicz†

August 2, 2024, 8:31pm

Abstract

The *simplified* Wasserstein barycenter problem, also known as the cheapest hub problem, consists in selecting one point from each of k given sets, each set consisting of n points, with the aim of minimizing the sum of distances to the barycenter of the k chosen points. This problem is also known as the cheapest hub problem. 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 strong upper and lower bounds. In addition, we extend the problem to allow varying n_j points for the j-th set.

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; ADMM; cheapest hub problem

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

List of Algorithms

List of Tables

List of Figures

1 Introduction

We consider the simplified Wasserstein barycenter problem, **WBP**, 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 a simplification of more general problems of *optimal mass transportation* and the problems of summarizing and combining probability measures that occurs in e.g., statistics and machine learning. In [2, Def. 1.4] this problem is called the *cheapest-hub problem*, and further in [2], a reduction to **WBP** is derived from the *k-clique problem* thus proving NP-hardness.[1] Algorithms for **WBP** with exponential dependence in d are discussed in [2, Sect. 1.3.1].[2] There are many important applications in molecular conformation e.g., [5], clustering [16], supervised and unsupervised learning, etc... For additional details on the theory and applications of optimal transport theory see e.g., [1, 8, 12], lecture link, and the many references therein.

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 when applied to our specific **WBP**.

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. The **sADMM** algorithm allows for efficient upper and provable lower bounding techniques for the original hard **WBP** problem, that in addition helps 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 **WBP** for *surprisingly many* instances, essentially all our randomly generated 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 on a laptop with Intel(R) Core(TM) i5-10210U CPU, 16GB RAM took approximately $2, 570, 46692$ seconds for $n = k = 5, 7, 8$, respectively, to solve the original hard discrete optimization problem to optimality. (Detailed numerics for our algorithm are provided below.)

The **DNN** relaxation can fail to find the exact solution for problems with special structure, i.e., there is a duality gap in the optimal value of the original hard problem and the lower bound found from the **DNN** relaxation. 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. A specific instance is included. Note that we consider that we have an optimal solution to **WBP** if the upper and lower bounds are equal to machine precision as any other feasible solution cannot have a smaller objective value within machine precision.

[1] Recall that the *k-clique problem* is the problem of finding k vertices in a graph such that each pair is close in the sense of being *adjacent*.

[2] We discuss this further below as the complexity of our algorithm does *not* depend on d.

1.1 Outline

We continue in Section 1.2 with preliminary notation. The main NP-hard Problem 2.1 and connections to Euclidean distance matrices, **EDM**, are given in Section 2.

A regularized, facially reduced, doubly nonnegative, **DNN**, relaxation is derived in Section 3. The **FR** in the relaxation fits *naturally* with applying a splitting approach. This is presented in Section 4. We include special bounding techniques and heuristics on the dual multipliers for accelerating the splitting algorithm. The algorithm provides *provable* lower and upper bounds for the original NP-hard Problem 2.1. Thus a *zero* gap (called a duality gap) proves optimality. Our empirics are given in Section 4.4.

In Section 5 we prove that multiple optimal solutions can lead to duality gaps. We include specific instances. Our concluding remarks are in Section 6.

1.2 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 $[k] = 1, 2, \ldots, k$.

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 $\mathbf{DNN} = \mathbb{S}^n_+ \cap \mathcal{N}^n$.

For a set of points $p_i \in \mathbb{R}^d$, we let $P = \begin{bmatrix} p_1^T \\ p_2^T \\ \ldots \\ p_t^T \end{bmatrix} \in \mathbb{R}^{t \times d}$ denote the *configuration matrix*. 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.,

$$P^T e = 0, \; e \text{ vector of ones.}^3$$

We denote the corresponding *Gram matrix*, $G = PP^T$. Then the classical result of Schoenberg [15] 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*, $\mathcal{S}^n_C$ and *hollow subspace*, $\mathcal{S}^n_H$

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

We ignore the dimension n when the meaning is clear. Note that the centered assumption $P^T e = 0 \implies G = PP^T \in \mathcal{S}^n_C$. Further notation is introduced as needed.

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.*

[3] 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.

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**. We follow the notation in [2, Sect. 1.2] and refer to our main problem as the simplified Wasserstein barycenter problem, or Wasserstein barycenter for short.

2.1 Main problem and EDM connection

Our main optimization problem is to find a point in each of k sets to obtain an optimal barycenter. We can think of this as finding a *hub of hubs*. That is, suppose that there are k areas with n airports in each area.[4] We want to choose exactly one airport to act as a minor hub in each of the k areas so that the barycenter for these k minor hubs would serve as a major (best) hub for the k minor hubs. In the literature this is called the *simplified barycenter problem* or *cheapest-hub problem*.

Problem 2.1 (*simplified Wasserstein barycenter,* ***WBP***)**.** *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:*

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

with

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

denoting the corresponding (configuration) matrix of points, ***EDM****, and Gram matrix, respectively. In this paper we allow the set sizes to vary* $n_j, j \in [k]$ *and let* $N = \sum j \in [k] n_j$.

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.[5] 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* $q_i \in \mathbb{R}^d, i = 1, \ldots k$. *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 $\sum_{i=1}^k (q_i - \bar{y}) = 0$. □

[4] We extend the prolem to allow for different sizes for the sets.

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

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*

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

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 $(JQ)_i = (Q - \frac{1}{k}ee^T Q)_i = (q_i - y)^T$. Now

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

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

The following Corollary 2.4 illustrates the connection between the simple Wasserstein barycenter problem[6] of finding the optimal barycenter and the k-clique problem of finding k pairwise adjacent vertices.

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:*

$$(WIQP) \qquad 2Np_W^* = p^* := \min_{\substack{p_i \in S_i \\ i\in[k]}} \sum_{i,j\in[k]} \|p_i - p_j\|^2, \quad N = \sum_{j\in[k]} n_j. \tag{2.5}$$

Proof. Suppose that $P^T = \begin{bmatrix} p_1 & \dots & p_k \end{bmatrix}$ is a matrix of optimal solution vectors 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 $p_i, i \in [k]$, by y and obtain $y = 0$. This implies that the corresponding Gram matrix $Ge = PP^T e = 0$. This combined with (2.1) and (2.3) and the corresponding distance matrix D yield

$$\begin{array}{rcl} \sum_{i,j\in[k]} \|p_i - p_j\|^2 & = & e^T De \\ & = & e^T(\operatorname{diag}(G)e^T + e\,diag(g)^T - 2G)e \\ & = & 2N \operatorname{diag}(G)e \\ & = & 2N \operatorname{tr} G \\ & = & 2N \sum_{i\in[k]} \|p_i\|^2 \\ & = & 2Np_W^*, \end{array} \tag{2.6}$$

where the last equality follows from Lemma 2.2. □

[6] We refer to this as the Wasserstein barycenter problem.

2.2 A reformulation using a Euclidean distance matrix

In this paper we work with p^* and and now provide a reformulation of (2.5) using an **EDM**. Define

$$x := \begin{pmatrix} v_1 \\ \dots \\ v_k \end{pmatrix} \in \mathbb{R}^N,\ v_i \in \mathbb{R}^{n_i}, \quad A := \text{blkdiag}[e_{n_1}^T, ..., e_{n_k}^T].$$

And $A = I \otimes e^T \in \mathbb{R}^{k \times kn}$, if $n_i = n, \forall i$, 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 the Euclidean distance matrix D formed from all N points $P^T = \begin{bmatrix} p_1 & \dots & p_{n_1} & p_{n_1+1} & \dots & p_{n_1+n_2} & \dots & \dots & p_N \end{bmatrix}$:

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

For simplicity in the sequel we often assume that the cardinality of all sets are equal.

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, [13], and the binary $0, 1$ *constraints. This in contrast to the linear programming approaches for the generalized transportation problems solved in e.g., [8] and the references therein. However, the unimodularity of the linear constraint matrix suggests that these two constraints both promote binary valued optimal points.*

3 Facially reduced DNN relaxation

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

3.1 Semidefinite programming (SDP) facially reduced relaxation

We begin with deriving a **SDP** relaxation of our formulation in (2.8). We start with a feasible vector $x \in \mathbb{R}^{kn}$ and set $\begin{pmatrix} x_0 \\ x \end{pmatrix} = \begin{pmatrix} 1 \\ x \end{pmatrix}$. We then lift the vector to a rank-1 matrix $Y_x := \begin{pmatrix} 1 \\ x \end{pmatrix} \begin{pmatrix} 1 \\ x \end{pmatrix}^T$. The convex hull of the lifted vertices of the feasible set of (2.8) yields an equivalent polyhedral set in $\mathbb{S}^{n+1}$. To obtain a convex relaxation, we relax the implicit nonconvex rank-1 constraint on Y_x and linearize the objective function. After the lifting, we impose the constraints that we have from x onto Y, e.g., the $0,1$ constraints $x_i^2 - x_i = 0$ become the $\operatorname{arrow}(Y_x) = e_0$ constraint

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

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. For convenience, we define

$$\operatorname{arrow}_0 : \mathbb{S}^{n+1} \to \mathbb{R}^{n+1} : \begin{bmatrix} s_0 & s^T \\ s & \bar{S} \end{bmatrix} \mapsto \begin{pmatrix} 0 \\ \operatorname{diag}(\bar{S}) - s \end{pmatrix}.$$

The linear constraints $AX = e$ is handled next using **FR**.

3.1.1 SDP reformulation via facial reduction

Denote the positive semidefinite matrix

$$K := \begin{bmatrix} -e^T \\ A^T \end{bmatrix} \begin{bmatrix} -e^T \\ A^T \end{bmatrix}^T \in \mathbb{S}_+^{kn+1}. \tag{3.1}$$

For the "only-one-element-from-each-set" linear equality constraint (see (2.7)), we observe that

$$\begin{aligned} Ax = e \iff & \begin{pmatrix} 1 \\ x \end{pmatrix}^T \begin{bmatrix} -e^T \\ A^T \end{bmatrix} = 0 \\ \iff & Y_x K := \begin{pmatrix} 1 \\ x \end{pmatrix} \begin{pmatrix} 1 \\ x \end{pmatrix}^T \begin{bmatrix} -e^T \\ A^T \end{bmatrix} \begin{bmatrix} -e^T \\ A^T \end{bmatrix}^T = 0 \\ \iff & \langle Y_x, K \rangle = 0 \\ \iff & KY_x = 0, \text{ i.e., } \operatorname{range}(Y_x) \subseteq \operatorname{null}(K) = \operatorname{null}\left(\begin{bmatrix} -e & A \end{bmatrix}\right). \end{aligned} \tag{3.2}$$

The last step follows since both $K, Y_x \succeq 0$. Moreover, this emphasizes that strict feasibility fails for feasible Y even if we ignore the rank one constraint.

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

$$Y \leftarrow VRV^T \in V\mathbb{S}_+^{nk+1-k}V^T \trianglelefteq \mathbb{S}_+^{kn+1}, \tag{3.3}$$

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. Alternatively,

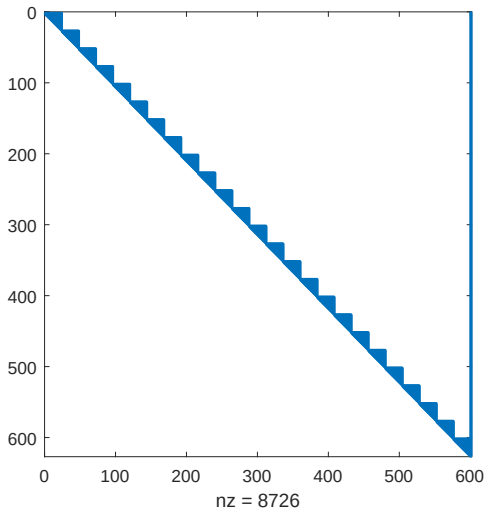

Figure 3.1: V matrix for k=20, n=20

we can use the MATLAB *QR algorithm $[q,\tilde{}] = qr(-e \quad A)$ and use the last part of q for the nullspace. This results in a relatively sparse orthonormal basis for the nullspace.*

We now explicitly find an efficient form for V in Lemma 3.2.

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

$$A = \begin{bmatrix} I_k \otimes e_n^T \end{bmatrix}, \; B = \begin{bmatrix} -e_k & A \end{bmatrix}.$$

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

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

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 $J_s := \{j_1^s, j_2^s, \ldots, j_{n-1}^s\}$, 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 $i, j \in \{1, \ldots, (n-1)k+1\}$. 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., $V \in \text{null}(B)$. To this end, we will see that $BV_j = 0$ for each $j = 1, \ldots, (n-1)k+1$. Fix $s \in \{1, \ldots, k\}$. 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

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

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

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.

We continue and clarify the usefulness of the arrow constraint.

Proposition 3.3. *The following holds:*

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

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 $x \in \mathbb{R}^{kn}$ such that $Y = \begin{pmatrix} x_0 \\ x \end{pmatrix} \begin{pmatrix} x_0 \\ x \end{pmatrix}^T$. Since $\text{arrow}(Y) = e_0$, $x_0^2 = 1$ and $x \circ x = x_0 x$. If $x_0 = 1, x \in \{0,1\}^{kn}$; otherwise $x_0 = -1$ and $x \in \{0,-1\}^n$ and it is easy to verify that

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

□

Recall the objective function in (2.8), and the lifted matrix variable Y_x. Define

$$\hat{D} := \begin{bmatrix} 0 & 0 \\ 0 & D \end{bmatrix} \in \mathbb{S}^{kn+1}. \tag{3.4}$$

The objective function of (2.8) now becomes $\langle D, xx^T \rangle = \langle \hat{D}, Y_x \rangle$. Then the **SDP** reformulation of (2.8) becomes

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

And if we substitute using the *facial vector*, V, $Y \leftarrow VRV^T$, then we can discard the $KY = 0$ constraint. We assume that the columns of the facial vector V form an orthonormal basis for the nullspace of K.

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

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

However, the improved processing efficiency of this convex relaxation trades off with the accuracy of solving the original NP-hard problem. The rank of an optimal Y now can be greater than 1. The idea now is to impose a "correct" amount of redundant constraints in the **SDP** model that reduces the rank of an optimal solution as much as possible, but does not hurt 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 3.4. *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., [14], and see that

$$A = I_k \otimes e^T,\ A^T = I_k \otimes e, \quad A^TA = I_k \otimes ee^T,$$

i.e., $A^TA = \text{BlkDiag}(ee^T \dots ee^T)$, a block diagonal structure. Therefore the columns of A are unit vectors and $\text{Diag}(\text{diag}(A^TA)) = I_{kn}$. The nonnegativity results follow from the definition, as does $Y_{00} = 1$.

Then

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

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

From Proposition 3.4, we see that the *gangster indices,* $\mathcal{J}$ 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 $\hat{\mathcal{J}} := \{(0,0)\} \cup \mathcal{J}$. We define the *gangster constraint mapping,* $\mathcal{G}_{\mathcal{J}}$:

$$\mathcal{G}_{\mathcal{J}}(Y) = Y(\mathcal{J}) \in \mathbb{R}^{|\mathcal{J}|},$$

i.e., the elements of Y indexed by the index set $\mathcal{J}$.

Now the **SDP** relaxation model becomes

$$\begin{array}{rll}
p^* = & \min_{Y \in \mathbb{S}^{kn+1}} & \langle \hat{D}, Y \rangle \\
& & Y_{00} = 1 \\
& & \text{arrow}_0(Y) = 0 \\
& & \mathcal{G}_{\mathcal{J}}(Y) = 0 \\
& & KY = 0 \\
& & Y \succeq 0.
\end{array} \tag{3.6}$$

The following follows from [5, Thm 2.1].

Proposition 3.5. *Consider the* ***SDP*** *relaxation* (3.6) *but without the* arrow_0 *constraint. Suppose that an optimal solution satisfies* $\text{rank}(Y) = 1$. *Then the constraint*

$$\text{arrow}_0(Y) = 0$$

holds, i.e., it was a redundant constraint.

Proof. Suppose that the hypothesis holds but the arrow_0 does not. Then, without loss of generality, we can assume that it fails for the $0,1$ element of Y. But then the top left 2×2 principal minor is rank 2, a contradiction. □

Our empirical tests on random problems without the *arrowz* constraint confirmed this result. However, the exctra redundant constraint is useful for the subproblems in the splitting approach.

3.2 Doubly nonnegative (DNN) relaxation

We now split the problem by using two variables $\{Y, R\}$ and apply a doubly nonnegative relaxation to (3.6). This *natural splitting* uses the facial reduction obtained in (3.3) 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 $x \in \{0,1\}^{kn}$. Hence, we can impose the redundant element-wise $[0,1]$-bound constraint on Y, i.e: $0 \leq Y \leq 1$. Moreover, the constraint $KY = 0$ is redundant once we apply **FR**, i.e., we get

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

We now add a redundant trace constraint on Y and transform it onto R.

Lemma 3.6. *Let* $Y \in \mathbb{S}^{kn+1}, Y = VRV^T, R \in \mathbb{S}^{nk+1-k}$. *Then*

$$KY = 0, \text{arrow}(Y) = e_0 \implies \text{tr}(Y) = \text{tr}(R) = k + 1.$$

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

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

By expanding the first column of the product, we get $\sum_{i=1}^n Y_{jn+i,0} = 1, \forall j \in \{0, \dots, k-1\}$. Since $\text{arrow}(Y) = e_0$, this implies that $\text{tr}(Y) = Y_{0,0} + \sum_{j=1}^k \sum_{i=1}^n Y_{jn+i,0} = 1 + k$. Since we choose the facial vector V to have orthonormal columns, the facial constraint yields

$$1 + k = \text{tr}(Y) = \text{tr}(VRV^T) = \text{tr}(RV^TV) = \text{tr}(R),$$

□

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

$$\mathcal{Y} := \{Y \in \mathbb{S}^{nk+1} : Y_{00} = 1, \mathcal{G}_{\mathcal{J}}(Y) = 0, \text{arrow}_0(Y) = 0, 0 \leq Y \leq 1\}, \ \mathcal{R} := \{R \in \mathbb{S}_+^{nk+1-k} : \text{tr}(R) = k{+}1\}. \tag{3.7}$$

Our ***DNN*** *relaxation* model is:

$$(\textbf{DNN}) \qquad \begin{array}{lll} p^*_{\textbf{DNN}} := \min_{R,Y} & \langle \hat{D}, Y \rangle \\ \text{s.t.} & Y = VRV^T \\ & Y \in \mathcal{Y} \\ & R \in \mathcal{R}. \end{array} \tag{3.8}$$

Observe that every feasible Y is both nonnegative element-wise and ***PSD*** , i.e., 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. Below, we do solve this using the MOSEK solver. However, one cannot get high accuracy and therefore the approximate optimal value we get is not a provable lower bound for Problem 2.1. Moreover, the expense does not scale well with N, and we cannot apply weak duality and use the dual solution as both primal and dual feasibility are not highly accurate. We overcome this problem for the splitting method in Section 4.1.1 below.

3.2.1 Characterization of optimality for DNN relaxation

Note that the *linear* mapping $\mathcal{M}(Y, R) := Y - VRV^T$ is surjective. Therefore the **KKT** optimality conditions hold for (3.8) with the normal cone $N_{\mathcal{Y}\times\mathcal{R}}(Y, R)$. In addition, the interior of the closed convex feasible set $\text{int}\,(\mathcal{Y} \times \mathcal{R}) \neq \emptyset$ implies that

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

We can now use the corresponding Lagrangian with dual variable Z:

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

where $\iota_S(\cdot)$ is the *indicator function* for the set S. Therefore the first-order optimality conditions to the problem in (3.8) are: a primal-dual pair (Y, R, Z) is optimal if, and only if,

$$Y = VRV^T, \quad R \in \mathcal{R},\, Y \in \mathcal{Y} \qquad \text{(primal feasibility)} \tag{3.9a}$$

$$0 \in -V^T Z V + \mathcal{N}_{\mathcal{R}}(R) \qquad \text{(dual } R \text{ feasibility)} \tag{3.9b}$$

$$0 \in \hat{D} + Z + \mathcal{N}_{\mathcal{Y}}(Y) \qquad \text{(dual } Y \text{ feasibility)} \tag{3.9c}$$

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 (3.8))**.** *The primal-dual pair* (R, Y, Z) *is optimal for* (3.8) *if, and only if,* (3.9) *holds if, and only if,*

$$R = \mathcal{P}_{\mathcal{R}}(R + V^T Z V) \tag{3.10a}$$

$$Y = \mathcal{P}_{\mathcal{Y}}(Y - \hat{D} - Z) \tag{3.10b}$$

$$Y = VRV^T. \tag{3.10c}$$

4 sADMM algorithm

The augmented Lagrangian corresponding to the **DNN** relaxation (3.8) with parameter $\beta > 0$ is

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

To solve the problem in (3.8), we will use the symmetric alternating directions method of multipliers **sADMM** that has intermediate updates of dual multipliers Z_t. It updates the dual variable twice: once after the R-update and then again after the Y-update. This approach has been used successfully in [5, 11]. (We include the details here for the reader.) 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 $Z_0 \in \mathbb{S}^{nk+1}$. The updates ($\forall k \in \mathbb{N}$):

$$\begin{array}{rcl} R_{k+1} & = & \operatorname{argmin}_{R \in \mathbb{S}^{nk+1-k}} \mathcal{L}_\beta(Y_k, R, Z_k) \\ Z_{k+\frac{1}{2}} & = & Z_k + \beta(Y_k - VR_{k+1}V^T) \\ Y_{k+1} & = & \operatorname{argmin}_{Y \in \mathbb{S}^{nk+1}} \mathcal{L}_\beta(Y, R_{k+1}, Z_{k+\frac{1}{2}}) \\ Z_{k+1} & = & Z_{k+\frac{1}{2}} + \beta(Y_{k+1} - VR_{k+1}V^T). \end{array} \tag{4.2}$$

In our **DNN** model (3.8), 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

The success of our splitting method is dependent on efficiently solving the subproblems. We start with using a spectral decomposition of M, implicitly defined below, to get the:

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

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., [6].

Next for the

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

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

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 (3.8) 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*

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

*Let (Y^*, R^*, Z^*) be an optimal primal-dual pair for the **DNN** in (3.8). Then, $Z^* \in \mathcal{Z}_A$.*

Proof. The proof of this fact uses the dual Y feasibility condition (3.9c) and a reformulation of the Y-feasible set. The details are in [9, Thm 2.14] and [5]. □

In view of Lemma 4.1 we propose the following modification of the symmetric **ADMM** algorithm, e.g., [10]. 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:

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

Note that a convergence proof using the modified updates is given in [9, Thm 3.2]. Therefore, in view of the **ADMM** updates (4.2) we propose the following Algorithm 4.1 with modified Z updates.

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**
  $R_{j+1} = U \operatorname{Diag}[P_{\Delta_{j+1}}(d)]U^T$ where $U \operatorname{Diag}(d)U^T = \operatorname{eig}(V^T(Y_j + \frac{1}{\beta}Z_j)V)$
  $Z_{j+\frac{1}{2}} = Z_j + \gamma\beta P_{Z_A}(Y_j - VR_{j+1}V^T)$
  $Y_{j+1} = P_{box}[\mathcal{G}_j(VR_{j+1}V^T - \frac{1}{\beta}(\hat{D} + Z_{j+\frac{1}{2}}))]$
  $Z_{j+1} = Z_{j+\frac{1}{2}} + \gamma\beta P_{Z_A}(Y_{j+1} - VR_{j+1}V^T)$
  $j = j + 1$
end while

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. We now provide provable upper and lower bounds to machine precision.

4.1.1 Provable lower bound to NP-hard problem

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 j, and applying weak duality, a lower bound to the optimal value of the **DNN** model (3.8) is

$$\begin{array}{rcl} p^*_{\mathbf{DNN}} & \geq & \max_Z g(Z) \\ & \geq & \min_{Y\in\mathcal{Y}}\langle \hat{D}+Z_j, Y\rangle - (k+1)\lambda_{max}(V^TZ_jV). \end{array} \tag{4.3}$$

Note that from the definition of $\mathcal{Y}$ in (3.7), this bound is found from solving: an LP with a simplex type feasible set; and an eigenvalue problem. Thus both values can be found accurately and efficiently. Moreover, since **DNN** is a relaxation, weak duality implies that this lower bound is a *provable lower bound* for the original NP-hard Problem 2.1.

4.1.2 Upper bounds

As for the upper bound, we consider two strategies for finding feasible solutions to the **BCQP** in (2.8). The 0-column approach is to take the first column $Y(1:end,0)$ and compute its nearest feasible solution to **BCQP**. It is equivalent to the greedy approach of using only the maximum weight index for each consecutive block of length n. The proof is in [5, Section 3.2.2].

Alternatively, we use the eigenvector of Y corresponding to the largest eigenvalue. The Perron-Frobenius Theorem implies this eigenvector is nonnegative, as Y is nonnegative. We then compute the nearest feasible solution to **BCQP**. It is again equivalent to the greedy approach.

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 j is defined to be $\frac{UB_j - LB_j}{|UB_j|+|LB_j|+1}$ where UB_j, LB_j denotes the best upper, lower, respectively, bound found up to 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 j as follows:

- Primal residual $r_j := \|Y_j - VR_jV^T\|$;
- Dual-R residual $s_j^R := \|R_j - \mathcal{P}_{\mathcal{R}}\left(R_j + V^TZ_jV\right)\|$;
- Dual-Y residual $s_j^Y := \left\|Y_j - \mathcal{P}_{\mathcal{Y}}\left(Y_j - \hat{D} - Z_{j+\frac{1}{2}}\right)\right\|$.

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 ϵ, a given tolerance;
- $KKTres := \max\{r_j, s_j^R, s_j^Y\} < \eta$, a given tolerance.
- Both the least upper bound and the greatest lower bound have not changed for boundCounterMax:=200 times (stalling).

4.3 Heuristics for algorithm acceleration

4.3.1 Adaptive step size

We apply the heuristic idea presented in [4], 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 does 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,

$$\begin{aligned} \langle \hat{D}, Y\rangle &:= \langle \hat{D} + \alpha I, Y\rangle - (n+1)\alpha \\ &= \langle \hat{D} + \alpha I, P_V Y P_V\rangle - (n+1)\alpha \\ &= \langle (P_V \hat{D} P_V + \alpha I), Y\rangle - (n+1)\alpha. \end{aligned} \tag{4.4}$$

Hence,

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

This lets us transform $\hat{D}$ into $\delta(P_V \hat{D} P_V + \alpha I)$ without changing the optimal 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 two linux servers: (i) fastlinux: greyling22 Dell R840 4 Intel Xeon Gold 6254, with 3.10 GHz, 72 core and 384 GB; and (ii) biglinux: Dell PowerEdge R6625, two AMD EPYC 9754 128-core 2.25 GHz, 1.5 TB.

Depending on fixed size or varying sizes for the sets, the size of the problem is $N = kn$, or $N = \sum_i n_i$, vectors respectively, in embedding dimension d. For the varying sizes n_i, we chose them randomly in the interval of width 5 about the given n value. Thus they increase in expected value with n. However, our quadratic model *hides* the dimension d as we only use the distances between the N points. This can be seen in the times in the tables. The size of the **DNN** model is $N + 1$ irrespective of d. The *hardness* of the problem is often hiding in the rank of the optimal solution Y of the **DNN** relaxation, i.e., if the rank is one, then we have solved the original NP-hard problem. However, if the rank is large, then the heuristics for the upper and lower bounds may not be enough to find an optimal solution for Problem 2.1.

The results below illustrate the efficiency and surprising success of our algorithm in finding the exact solution of the original NP-hard problem.

1. In Figure 4.1, page 19, we see the slow (linear) growth, other than outliers, for the computation time versus the size $N = kn$. This was for sets of equal size with $d = (3 : 3 : 27), k = (36 : 2 : 50), n = (47 : 1 : 60)$ with the averages taken of two problems for each data instance.[7]

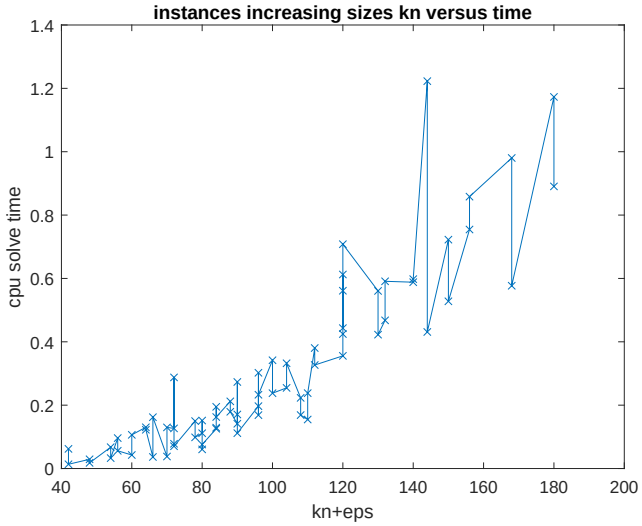

Figure 4.1: size $N = kn$ versus cpu time; illustrating linear time

2. In Table 4.1, page 20, we see a comparison between using the **sADMM** approach and CVX with the Sedumi solver. We are using varying sizes for the sets and posting the total number of points under N. We can see the times for the cvxsolver increase dramatically. The relative duality gap from cvx is *not* a provable gap as the lower bound is obtained using the dual optimal value minus the posted accuracy of the solve. We do not have accurate primal or dual solutions from cvx. Thus the relative gap is essentially the posted accuracy from cvx. We do find a nearest feasible point to find the upper bound. In summary, we see that the dramatic difference in time and the improved accuracy with the guaranteed lower bound that verifies optimality.

[7]Done using MATLAB version 2024a on fastlinux, cpu157.math.private, Dell PowerEdge R660, Two Intel Xeon Gold 6434, 8-core 3.7 GHz (Sapphire Rapids),256 GB

dim/sets/size			Time (s)		rel. duality gap	
d	k	N	ADMM	CVXsolver	ADMM	CVXsolver
2	8	56	0.14	10.47	1.5e-14	1.3e-10
2	8	72	0.28	46.09	2.0e-14	9.1e-10
2	8	88	0.34	75.33	3.3e-15	5.2e-10
2	8	104	0.45	275.38	2.8e-14	6.1e-10
2	9	63	0.26	24.55	2.9e-15	3.2e-10
2	9	81	0.31	75.90	-1.5e-16	3.5e-10
2	9	99	0.30	373.53	1.6e-14	3.8e-10
2	9	117	0.67	809.28	-4.2e-14	5.9e-09
2	10	70	0.22	43.13	3.1e-14	1.2e-10
2	10	90	0.30	250.04	5.9e-16	3.0e-10
2	10	110	0.42	553.62	2.5e-14	4.4e-10
2	10	130	0.62	1555.54	3.4e-15	2.8e-09
3	8	56	0.10	9.55	4.6e-14	1.5e-10
3	8	72	0.38	46.60	2.6e-15	5.8e-10
3	8	88	0.21	77.41	-3.1e-15	6.6e-10
3	8	104	0.38	280.12	1.7e-14	7.6e-10
3	9	63	0.15	19.97	3.4e-15	1.6e-10
3	9	81	0.13	81.98	7.7e-15	3.3e-10
3	9	99	0.24	360.76	-2.8e-15	4.5e-10
3	9	117	0.62	803.72	4.7e-14	5.2e-09
3	10	70	0.21	39.18	5.2e-15	1.7e-10
3	10	90	0.23	236.75	9.1e-17	1.4e-10
3	10	110	0.29	562.02	3.0e-15	3.5e-10
3	10	130	0.76	1473.45	3.3e-14	4.7e-09

Table 4.1: Comparing ADMM with CVX Solver Sedumi

3. We include large problems in Table 4.2, page 21. Other than outliers, the times are very reasonable.

5 Multiple optimal solutions and duality gaps

We now show that *multiple optimal* solutions for the original hard problem can lead to a duality gap between the optimal value of the original NP-hard problem and the lower bound found from the **DNN** relaxation.

5.1 Criteria for duality gaps

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

dim/sets/size			Time (s)	rel.duality gap
d	k	N	ADMM	ADMM
8	30	1200	104.81	-1.9e-15
8	30	1230	67.08	-3.2e-14
8	31	1240	94.90	-1.3e-14
8	31	1271	81.93	2.5e-14
8	32	1280	75.29	3.1e-14
8	32	1312	2025.42	1.8e-13
9	30	1200	4586.51	1.2e-13
9	30	1230	63.91	2.6e-14
9	31	1240	93.96	3.3e-14
9	31	1271	71.31	3.2e-14
9	32	1280	92.89	3.6e-14
9	32	1312	86.67	-2.2e-13

Table 4.2: Large problems with **sADMM**on biglinux server

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. That $\hat{X} \in \mathbb{R}^n_{++}$ is clear from the hypothesis. Now 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 $v \neq 0$. □

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

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

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 $\operatorname{tr} YQ = y^T Qy > p^* = \operatorname{tr} \hat{X}Q$. 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 (3.1) used for facial reduction and the facially reduced **DNN** relaxation in (3.8).

Corollary 5.3. *We consider the **BCQP** in (2.8) with optimal value p^*, and the **DNN** relaxation in (3.8). 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 with a duality gap

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 can exist between the optimal value of the original NP-hard problem and the optimal alue of the **DNN** relaxation.

Example 5.4 (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, page 24. We use the corresponding nine points in the configuration matrix ordered* $1 - 9$ *counter-clockwise in the triangles ordered counter-clockwise.*

$$P = \begin{bmatrix} 1.7536 & 0.0137 \\ 0.6195 & 0.6362 \\ 0.6239 & -0.6643 \\ 0.2590 & 0.8609 \\ -0.8839 & 1.5449 \\ -0.8692 & 0.2201 \\ 0.2629 & -0.8740 \\ -0.8937 & -0.2100 \\ -0.8721 & -1.5275 \end{bmatrix}$$

The distances are ordered by choosing the points in lexicographic order: $(1,1,1),(1,1,2),\ldots(3,1,1),\ldots,(3,3,3)$. *The unique minimum distance is* 11.1607 *obtained from the points* $(2,3,2)$ *and with the primal optimal* $x^* = \begin{pmatrix}0 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 0\end{pmatrix}^T$. *The optimal value from cvx to* 9 *decimals precision is* 10.8246, *thus verifying an empirical duality gap of* .3 *to* 9 *decimals precision. The motivation for this counter-example is to have near optimal solutions. One could shrink triangle two to make point* 4 *equidistant to points* $7, 8$ *and move point* 2 *closer to point* 3 *and thus have a tie optimal solution.*[8]

Note that the maximum distance is 56.0227 *obtained from points* $(1,2,3)$.

search. The optimal value of the relaxation if found accurately to ???? add reference to Miguel????

However, when k *is even, only one optimal solution clearly exists and the duality gap becomes trivial. An example with* $k = 6 = n$ *follows in Figure 5.2, page 25.*

6 Conclusion

In this paper we presented a strategy for solving a class of NP-hard binary quadratic problems. This involves formulating a **DNN** relaxation, applying **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. In particular, the structure of both the primal and dual solutions is exploited in the updating steps of the **sADMM**. We applied this to the **NP**-hard computational problem called the Simplified Wasserstein Barycenter problem.

[8] The wheel graph was used successfully to obtain duality gaps for the second lifting of the max-cut problem, see [3].

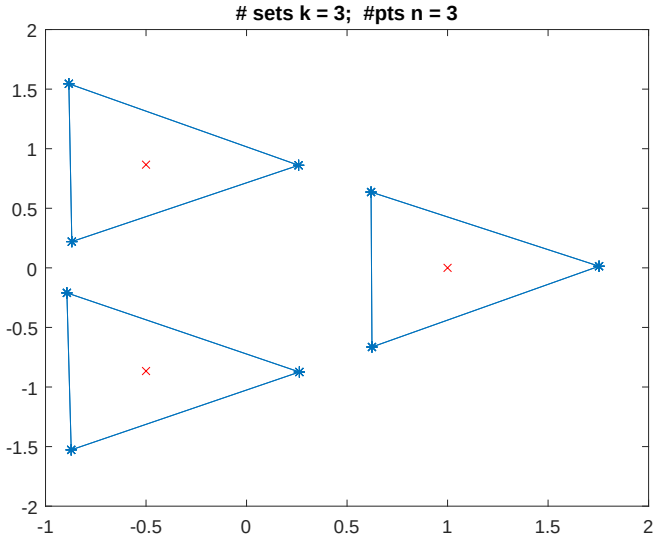

Figure 5.1: Duality gap for wheel of wheels: k=3=n

Surprisingly, for the random problems we generated the gap between bounds was zero to machine precision and we were able to provably solve the original NP-hard optimization problem, i.e., if there were another optimal solution, then it would yield the same optimal value to machine precision. This coincided with $\text{rank}(Y^*) = 1$ for the optimal solution found for the **DNN** relaxation. We observed that the embedding dimension d is hidden in the **DNN** relaxation. However, for specially constructed input data that has near multiple optimal solutions, the algorithm had difficulty breaking ties and the result was gaps between lower and upper bounds conciding with $\text{rank}(Y^*) > 1$, i.e., the original Wasserstein problem was not solved to optimality. To continue a process to optimality, the size of d would be involved in any branch and bound process.

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. Does the lack of gaps correspond to large volumes for the normal cones at points on the boundary of the feasible set? 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.

Finally, we are gathering data about airports in North America by state and province ourairports.com/continents/NA/. We plan on solving the problem of finding the best hub in each state (or province) in order to find the location for the best hub for the country.

Acknowledgment The authors would like to thank Jiahui (Jeffrey) Cheng for his contributions to the work in this paper. The joint research continued from the work jointly started that resulted in Mr Cheng's Master's thesis [7]. (University of Waterloo, 2023-05-04). The authors have relied on results from the Masters thesis with link at University of Waterloo. The content and codes in this paper have changed.

Data availability statement The codes with descriptions are available at this link: www.math.uwaterloo.ca/~hwolkowi/henry/reports/2024codesWassersteinAdmm.d/

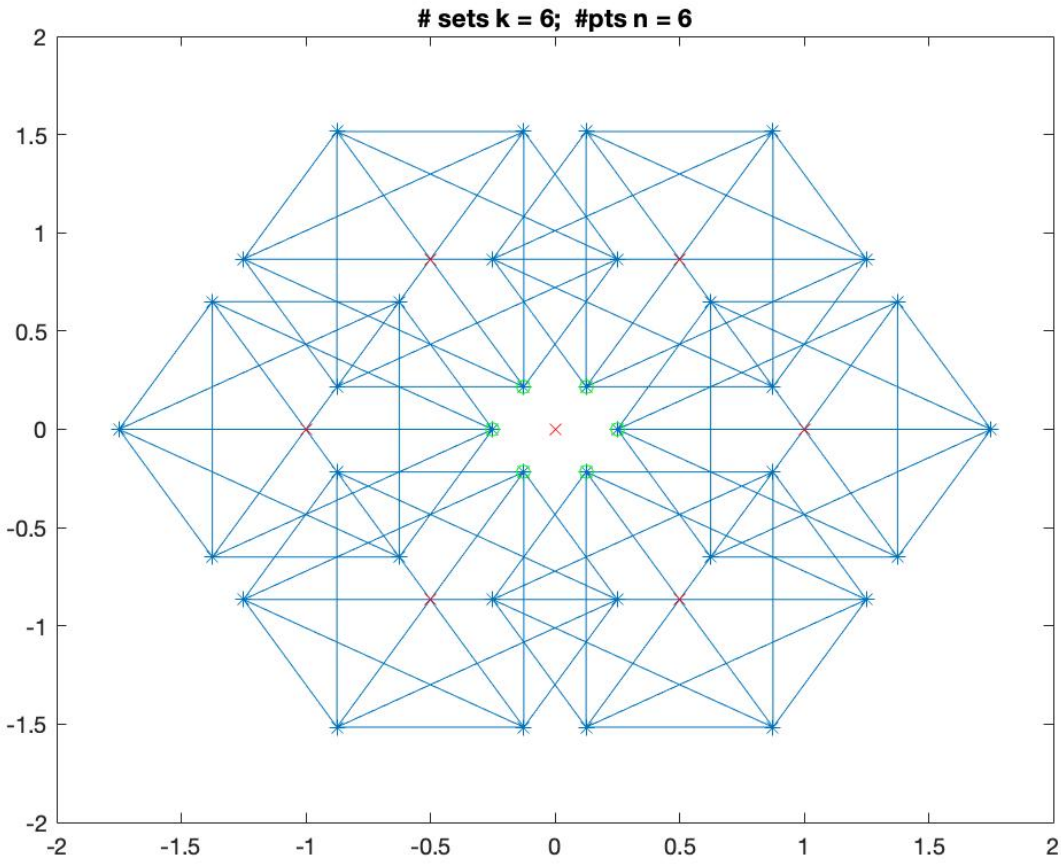

Figure 5.2: No duality gap for wheel of wheels: k=6=n

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. 3, 5

[3] M. Anjos and H. Wolkowicz, *Strengthened semidefinite relaxations via a second lifting for the Max-Cut problem*, Discrete Appl. Math., 119 (2002), pp. 79–106. Foundations of heuristics in combinatorial optimization. 23

[4] 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. 18

[5] 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. 3, 12, 15, 16, 17

[6] Y. Chen and X. Ye, *Projection onto a simplex*, 2011. 15

[7] J. Cheng, *Semidefinite programming relaxations of the simplified wasserstein barycenter problem: An ADMM approach*, Master's thesis, University of Waterloo, 2023. 24

[8] D. Ge, H. Wang, Z. Xiong, and Y. Ye, *Interior-point methods strike back: Solving the Wasserstein barycenter problem*, in 33rd Conference on Neural Information Processing Systems (NeurIPS 2019), Vancouver, Canada, vol. 33, 2019, pp. 1–12. 3, 7

[9] 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. 16

[10] 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. 16

[11] X. Li, T. Pong, H. Sun, and H. Wolkowicz, *A strictly contractive Peaceman-Rachford splitting method for the doubly nonnegative relaxation of the minimum cut problem*, Comput. Optim. Appl., 78 (2021), pp. 853–891. 15

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

[13] P. Pardalos and S. Vavasis, *Quadratic programming with one negative eigenvalue is NP-hard*, J. Global Optim., 1 (1991), pp. 15–22. 7

[14] K. Schaecke, *Essay on: The Kronecker product*, Master's thesis, University of Waterloo, 2004. 12

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

[16] J. Ye, P. Wu, J. Z. Wang, and J. Li, *Fast discrete distribution clustering using Wasserstein barycenter with sparse support*, IEEE Transactions on Signal Processing, 65 (2017), pp. 2317–2332. 3