

## INFORMS Journal on Computing

Publication details, including instructions for authors and subscription information:  
<http://pubsonline.informs.org>

### A Restricted Dual Peaceman-Rachford Splitting Method for a Strengthened DNN Relaxation for QAP

Naomi Graham, Hao Hu, Jiyong Im, Xinxin Li, Henry Wolkowicz

To cite this article:

Naomi Graham, Hao Hu, Jiyong Im, Xinxin Li, Henry Wolkowicz (2022) A Restricted Dual Peaceman-Rachford Splitting Method for a Strengthened DNN Relaxation for QAP. INFORMS Journal on Computing 34(4):2125-2143. <https://doi.org/10.1287/ijoc.2022.1161>

Full terms and conditions of use: <https://pubsonline.informs.org/Publications/Librarians-Portal/PubsOnLine-Terms-and-Conditions>

This article may be used only for the purposes of research, teaching, and/or private study. Commercial use or systematic downloading (by robots or other automatic processes) is prohibited without explicit Publisher approval, unless otherwise noted. For more information, contact [permissions@informs.org](mailto:permissions@informs.org).

The Publisher does not warrant or guarantee the article's accuracy, completeness, merchantability, fitness for a particular purpose, or non-infringement. Descriptions of, or references to, products or publications, or inclusion of an advertisement in this article, neither constitutes nor implies a guarantee, endorsement, or support of claims made of that product, publication, or service.

Copyright © 2022, INFORMS

Please scroll down for article—it is on subsequent pages



With 12,500 members from nearly 90 countries, INFORMS is the largest international association of operations research (O.R.) and analytics professionals and students. INFORMS provides unique networking and learning opportunities for individual professionals, and organizations of all types and sizes, to better understand and use O.R. and analytics tools and methods to transform strategic visions and achieve better outcomes.

For more information on INFORMS, its publications, membership, or meetings visit <http://www.informs.org>

# A Restricted Dual Peaceman-Rachford Splitting Method for a Strengthened DNN Relaxation for QAP

Naomi Graham,<sup>a</sup> Hao Hu,<sup>b</sup> Jiyoung Im,<sup>c</sup> Xinxin Li,<sup>d,\*</sup> Henry Wolkowicz<sup>e</sup>

<sup>a</sup>Department of Computer Science, The University of British Columbia, Vancouver, British Columbia V6T 1Z4, Canada; <sup>b</sup>School of Mathematical and Statistical Sciences, Clemson University, Clemson, South Carolina 29634; <sup>c</sup>Department of Combinatorics and Optimization, University of Waterloo, Waterloo, N2L 3G1 Ontario, Canada; <sup>d</sup>School of Mathematics, Jilin University, Changchun, Jilin 130012, China;

<sup>e</sup>Department of Combinatorics and Optimization, University of Waterloo, Waterloo, N2L 3G1 Ontario, Canada

\*Corresponding author

Contact: naomigra@cs.ubc.ca (NG); hhu2@clemson.edu (HH); haesol42@gmail.com (JI); xinxinli@jlu.edu.cn,

 <https://orcid.org/0000-0002-6392-701X> (XL); hwolkowicz@uwaterloo.ca (HW)

Received: December 27, 2020

Revised: June 27, 2021; September 28, 2021; November 25, 2021; December 3, 2021

Accepted: December 9, 2021

Published Online in Articles in Advance: March 23, 2022

<https://doi.org/10.1287/ijoc.2022.1161>

Copyright: © 2022 INFORMS

**Abstract.** Splitting methods in optimization arise when one can divide an optimization problem into two or more simpler subproblems. They have proven particularly successful for relaxations of problems involving discrete variables. We revisit and strengthen splitting methods for solving doubly nonnegative relaxations of the particularly difficult, NP-hard quadratic assignment problem. We use a modified restricted contractive splitting method approach. In particular, we show how to exploit redundant constraints in the subproblems. Our strengthened bounds exploit these new subproblems and new dual multiplier estimates to improve on the bounds and convergence results in the literature.

**Summary of Contribution:** In our paper, we consider the quadratic assignment problem (QAP). It is one of the fundamental combinatorial optimization problems in the fields of optimization and operations research and includes many fundamental applications. We revisit and strengthen splitting methods for solving doubly nonnegative (DNN) relaxation of the QAP. We use a modified restricted contractive splitting method. We obtain strengthened bounds from improved lower and upper bounding techniques, and in fact, we solve many of these NP-hard problems to (provable) optimality, thus illustrating both the strength of the DNN relaxation and our new bounding techniques.

**History:** Accepted by Antonio Frangioni, Area Editor for Design & Analysis of Algorithms–Continuous.

**Funding:** N. Graham, H. Hu, J. Im, and H. Wolkowicz were funded by National Research Council Canada [Grant 50503-10827]. X. Li was funded by the National Natural Science Foundation of China [Grants 11601183 and 61872162] and the China Scholarship Council [Grant 201806175127].

**Supplemental Material:** The software that supports the findings of this study is available within the paper and its Supplementary Information (<https://pubsonline.informs.org/doi/suppl/10.1287/ijoc.2022.1161>) or is available from the IJOC GitHub software repository (<https://github.com/INFORMSjoc>) at <http://dx.doi.org/10.5281/zenodo.5709968>.

**Keywords:** quadratic assignment problem • semidefinite relaxation • doubly nonnegative relaxation • facial reduction • Peaceman-Rachford splitting method

## 1. Introduction

Splitting methods allow for numerically hard problems to be divided in simpler subproblems. These methods have proven to be particularly successful for relaxations of hard nonlinear discrete optimization problems. We revisit and provide a strengthened splitting method for solving the doubly nonnegative (DNN) relaxation of the quadratic assignment problem (QAP), arguably one of the hardest of the NP-hard problems. The problem involves finding an optimal permutation matrix, and problems with size  $n = 30$  are still considered difficult. Here the DNN relaxation refers to the semidefinite programming, SDP, relaxation with the addition of nonnegativity constraints on *all*, order  $n^4$ , elements of the relaxed variable. We use a modified *restricted contractive*

*Peaceman-Rachford splitting method* (rPRSM) approach. We obtain strengthened bounds from improved lower and upper bounding techniques applied during the algorithm. As a result, we solve many of these NP-hard problems to (provable) optimality, thus illustrating both the strength of the DNN relaxation and our new bounding techniques. In addition, we get improved rates of convergence from the strengthened subproblems and dual multiplier estimates. Our results significantly improve on the recent results in Oliveira et al. (2018). We also compare with other recent relaxations.

The QAP is one of the fundamental combinatorial optimization problems in the field of operations research and includes many important applications. Proving optimality is particularly difficult, most

probably because of the many local minima in many of the instances in the literature (Barvinok and Stephen 2003, Hahn et al. 2010). The QAP models real-life problems such as facility location. Suppose that we are given a set of  $n$  facilities and a set of  $n$  locations. For each pair of locations  $(s, t)$ , a distance  $B_{st}$  is specified, and for each pair of facilities  $(i, j)$ , a weight or flow  $A_{i,j}$  is specified, for example, the amount of supplies transported between the two facilities. In addition, there is a location (building) cost  $C_{is}$  for assigning a facility  $i$  to a specific location  $s$ . The problem is to assign each facility to a distinct location with the goal of minimizing the sum over all facility-location pairs of the distances between locations multiplied by the corresponding flows between facilities, along with the sum of the location costs. Other applications include the following: scheduling, production, computer manufacture (very large-scale integration design), chemistry (molecular conformation), communication, and other fields (Geoffrion and Graves 1976, Elshafei 1977, Heffley 1977, Krarup and Pruzan 1978, Ugi et al. 1979).

Moreover, many classical combinatorial optimization problems, including the traveling salesman problem, maximum clique problem, and graph partitioning problem, can all be expressed as a QAP (Pardalos and Wolkowicz 1994, Pardalos et al. 1994, Çela 1998, Commander 2003, Bhati and Rasool 2014).

The NP-hardness of the QAP (1.1) is proven in Garey and Johnson (1979). The cardinality of the feasible set of permutation matrices  $\Pi$  is  $n!$ , and it is known that problems typically have *many* local minima. Until now, there were three main classes of methods for solving QAP. The first type is heuristic algorithms, such as genetic algorithms (Drezner 2003), ant systems (Gambardella et al. 1999), and meta-heuristic algorithms (Bashiri and Karimi 2012). These methods usually have short running times and often give optimal or near-optimal solutions. However, the solutions from heuristic algorithms are not reliable, and the performance can vary depending on the type of problem. The second type is branch-and-bound algorithms. Although this approach gives exact solutions, it can be very time consuming and in addition requires strong bounding techniques. For example, obtaining an exact solution using the branch-and-bound method for  $n = 30$  is still considered to be computationally challenging. The third type is based on SDP. SDP is proven to have successful implementations and provides tight relaxations (Zhao et al. 1998, Anstreicher and Brixius 2000). There are many well-developed SDP solvers based on, for example, interior point methods (Mitchell et al. 1998, Wolkowicz et al. 2000, Anjos and Lasserre 2011). However, the running time of the interior point methods do not scale well, and the SDP relaxations become very large for the QAP. In addition, adding additional polyhedral constraints such as interval  $[0, 1]$  constraints can result in having  $O(2n^2)$  constraints, a prohibitive number for interior point methods.

Recently, Oliveira et al. (2018) used an *alternating direction method of multipliers (ADMM)* to solve a facially reduced (FR) SDP relaxation. The FR allows for a natural splitting of variables between the SDP cone and polyhedral constraints. The algorithm provides competitive lower and upper bounds for QAP. In this paper, we modify and improve on this approach. (Our work also follows and relates to that in Li et al. (2021) that concentrates on the min-cut problem. In addition, we note the work in Hu et al. (2019) that also uses FR on QAP problems but concentrates on exploiting group symmetry structure.)

### 1.1. Background

We let  $\mathbb{R}^n$  denote the usual Euclidean space of dimension  $n$ , and let  $\mathbb{S}^n$  denote the space of real symmetric matrices of order  $n$ . We use  $\mathbb{S}_+^n$  ( $\mathbb{S}_{++}^n$ , respectively) to denote the cone of  $n$ -by- $n$  positive semidefinite (definite) matrices. We write  $X \succeq 0$  if  $X \in \mathbb{S}_+^n$ , and  $X > 0$  if  $X \in \mathbb{S}_{++}^n$ . Given  $X \in \mathbb{R}^{n \times n}$ , we use  $\text{tr}(X)$  to denote the trace of  $X$ . We use  $\circ$  to denote the Hadamard (elementwise) product. Given a matrix  $A \in \mathbb{R}^{m \times n}$ , we use  $\text{range}(A)$  and  $\text{null}(A)$  to denote the range of  $A$  and the null space of  $A$ , respectively. For  $n \geq 1$ ,  $e_n$  denotes the vector of all ones of dimension  $n$ ;  $E_n$  denotes the  $n \times n$  matrix of all ones. We omit the subscripts of  $e_n$  and  $E_n$  when the dimension is clear. Also, we define  $u_0$  to be the first unit vector.

It is known (Edwards 1977) that many of the QAP models, such as the facility location problem, can be formulated using the *trace formulation*:

$$p_{\text{QAP}}^* := \min_{X \in \Pi} \langle AXB - 2C, X \rangle, \quad (1.1)$$

where  $A, B \in \mathbb{S}^n$  are real symmetric  $n \times n$  matrices,  $C$  is a real  $n \times n$  matrix,  $\langle \cdot, \cdot \rangle$  denotes the trace inner product, that is,  $\langle Y, X \rangle = \text{tr}(YX^T)$ , and  $\Pi$  denotes the set of  $n \times n$  permutation matrices.

**Remark 1.1.** We note that the location problem is symmetric in facilities and locations; that is, the optimal value is independent of which of  $A, B$  is chosen for distance data and which for flow data. However, the facility location interpretation does not make sense if there are zero distances. In particular, the data are troublesome if both matrices  $A, B$  have zeros in off-diagonal positions, as is the case for many of the instances in the quadratic assignment problem library Burkard et al. (1991), the data source that we use.

We use the following notation from Oliveira et al. (2018). We denote the *matrix lifting*

$$Y := \begin{pmatrix} 1 & \\ & x \end{pmatrix} (1 \ x^T) \in \mathbb{S}^{n^2+1}, \quad x = \text{vec}(X) \in \mathbb{R}^{n^2}, \quad (1.2)$$

where  $\text{vec}(X)$  is the vectorization of the matrix  $X \in \mathbb{R}^{n \times n}$ , columnwise. Then  $Y \in \mathbb{S}_+^{n^2+1}$ , the (convex) cone of real symmetric positive semidefinite matrices of order

$n^2 + 1$ , and the rank,  $\text{rank}(Y) = 1$ . Indexing the rows and columns of  $Y$  from zero to  $n^2$ , we can express  $Y$  in (1.2) using a block representation as follows:

$$Y = \begin{bmatrix} Y_{00} & \bar{y}^T \\ \bar{y} & \bar{Y} \end{bmatrix}, \quad \bar{y} = \begin{bmatrix} Y_{(10)} \\ Y_{(20)} \\ \vdots \\ Y_{(n0)} \end{bmatrix}$$

$$\text{and } \bar{Y} = x x^T = \begin{bmatrix} \bar{Y}_{(11)} & \bar{Y}_{(12)} & \cdots & \bar{Y}_{(1n)} \\ \bar{Y}_{(21)} & \bar{Y}_{(22)} & \cdots & \bar{Y}_{(2n)} \\ \vdots & \ddots & \ddots & \vdots \\ \bar{Y}_{(n1)} & \ddots & \ddots & \bar{Y}_{(nn)} \end{bmatrix}, \quad (1.3)$$

where

$$\bar{Y}_{(ij)} = X_{:i} X_{:j}^T \in \mathbb{R}^{n \times n}, \quad \forall i, j = 1, \dots, n, Y_{(j0)} \in \mathbb{R}^n, \\ \forall j = 1, \dots, n, \text{ and } x \in \mathbb{R}^{n^2}.$$

Let

$$L_Q = \begin{bmatrix} 0 & -(\text{vec}(C)^T) \\ -\text{vec}(C) & B \otimes A \end{bmatrix},$$

where  $\otimes$  denotes the Kronecker product. We further scale  $L_Q$  in (2.8) and (2.9). With the previous notation and matrix lifting, we can reformulate the QAP (1.1) equivalently as

$$p_{\text{QAP}}^* = \min \langle AXB - 2C, X \rangle = \langle L_Q, Y \rangle \\ \text{s.t. } Y := \begin{pmatrix} 1 \\ x \end{pmatrix} \begin{pmatrix} 1 \\ x \end{pmatrix}^T \in \mathbb{S}_+^{n^2+1} \\ X = \text{Mat}(x) \in \Pi, \quad (1.4)$$

where  $\text{Mat} = \text{vec}^*$ , the adjoint transformation.

Zhao et al. (1998) derive an SDP relaxation as the dual of the Lagrangian relaxation of a quadratically constrained version of (1.4), that is, the constraint that  $X \in \Pi$  is replaced by quadratic constraints, for example,

$$\|Xe - e\|^2 = \|X^T e - e\|^2 = 0, X \circ X = X, X^T X = X X^T = I,$$

where  $\circ$  is the Hadamard product and  $e$  is the vector of all ones. After applying the so-called *facial reduction* technique to the SDP relaxation, the variable  $Y$  is expressed as  $Y = \hat{V} R \hat{V}^T$ , for some full column rank matrix  $\hat{V} \in \mathbb{R}^{(n^2+1) \times ((n-1)^2+1)}$  defined in Section 2.1.2. The SDP relaxation then takes on the smaller, greatly simplified form after many of the constraints are shown to be redundant:

$$\min_R \langle \hat{V}^T L_Q \hat{V}, R \rangle \\ \text{(SDP) s.t. } \mathcal{G}_{\bar{J}}(\hat{V} R \hat{V}^T) = u_0 \\ R \in \mathbb{S}_+^{(n-1)^2+1}. \quad (1.5)$$

The linear transformation  $\mathcal{G}_{\bar{J}}(\cdot)$  is called the *gangster operator* as it fixes certain elements of the matrix, and  $u_0$  is the first unit vector. The Slater constraint qualification, strict feasibility, holds for both (1.5) and its dual (Zhao et al. 1998, lemmas 5.1 and 5.2). We refer to Zhao et al. (1998) for details on using the dual of the Lagrangian dual for the derivation of this *facially reduced SDP*.

We now provide the details for  $\hat{V}$ , the gangster operator  $\mathcal{G}_{\bar{J}}$ , and the *gangster index set*,  $\bar{J}$ .

1. Let  $\hat{Y}$  be the barycenter of the set of feasible lifted  $Y$  (1.3) of rank one for the SDP relaxation of (1.4). Let the matrix  $\hat{V} \in \mathbb{R}^{(n^2+1) \times ((n-1)^2+1)}$  have orthonormal columns that span the range of  $\hat{Y}$ .<sup>1</sup> Every feasible  $Y$  of the SDP relaxation is contained in the *minimal face*,  $\mathcal{F}$  of  $\mathbb{S}_+^{n^2+1}$ :

$$\mathcal{F} = \hat{V} \mathbb{S}_+^{(n-1)^2+1} \hat{V}^T \trianglelefteq \mathbb{S}_+^{n^2+1}, \\ Y \in \mathcal{F} \Rightarrow \text{range}(Y) \subseteq \text{range}(\hat{V}), \\ Y \in \text{relint}(\mathcal{F}) \Rightarrow \text{range}(Y) = \text{range}(\hat{V}).$$

2. The gangster operator (transformation) is the linear map  $\mathcal{G}_{\bar{J}} : \mathbb{S}^{n^2+1} \rightarrow \mathbb{R}^{|\bar{J}|}$  defined by

$$\mathcal{G}_{\bar{J}}(Y) = Y_{\bar{J}} \in \mathbb{R}^{|\bar{J}|}, \quad (1.6)$$

where  $\bar{J}$  is a subset of (upper triangular) matrix indices of  $Y$ .

**Remark 1.2.** By abuse of notation, we also consider the gangster operator from  $\mathbb{S}^{n^2+1}$  to  $\mathbb{S}^{n^2+1}$ , depending on the context:

$$\mathcal{G}_{\bar{J}} : \mathbb{S}^{n^2+1} \rightarrow \mathbb{S}^{n^2+1}, \quad [\mathcal{G}_{\bar{J}}(Y)]_{ij} = \begin{cases} Y_{ij} & \text{if } (i,j) \in \bar{J} \text{ or } (j,i) \in \bar{J}, \\ 0 & \text{otherwise.} \end{cases} \quad (1.7)$$

Both formulations of  $\mathcal{G}_{\bar{J}}$  are used for defining a constraint which “shoots holes” in the matrix  $Y$  with entries indexed using  $\bar{J}$ . Although the latter formulation is more explicit, it is not surjective and is not used in the implementations.

3. The gangster index set  $\bar{J}$  is defined to be the union of the top left index (00) with the set of indices  $J$  with  $i < j$  in the submatrix  $\tilde{Y} \in \mathbb{S}^{n^2}$  corresponding to:

- (a) the off-diagonal elements in the  $n$  diagonal blocks in  $\tilde{Y}$  in (1.3);
  - (b) the diagonal elements in the off-diagonal blocks in  $\tilde{Y}$  in (1.3).
- (1.8)

Many of the constraints that arise from the index set  $J$  are redundant. We could remove the indices in the submatrix  $\tilde{Y} \in \mathbb{S}^{n^2}$  corresponding to all the diagonal

Downloaded from informas.org by [129.97.202.75] on 25 August 2023, at 11:39 . For personal use only, all rights reserved.

positions of the last column of off-diagonal blocks and the additional  $(n-2, n-1)$  block. In our implementations, we take advantage of redundant constraints when used as constraints in the subproblems and in prespecifying dual variables. We denote the *redundant gangster constraints*,  $J_R$ .

4. The notation  $u_0$  in (1.5) denotes a vector in  $\{0, 1\}^{|\bar{J}|}$  with one only in the first coordinate, that is, the zeroth unit vector. Therefore, (1.5) forces all the values of  $\hat{V}R\hat{V}^T$  corresponding to the indices in  $\bar{J}$  to be zero. It also implies that the first entry of  $\mathcal{G}_{\bar{j}}(\hat{V}R\hat{V}^T)$  is equal to one, which reflects the fact that  $Y_{00} = 1$  from (1.3). Using the alternative definition of  $\mathcal{G}_{\bar{j}}$  in (1.7), the equivalent constraint is  $\mathcal{G}_{\bar{j}}(Y) = E_{00}$ , where  $E_{00} \in \mathbb{S}^{n^2+1}$  is the  $(0, 1)$ -matrix with one only in the  $(00)$  position. Therefore, (1.5) forces all the values of  $\hat{V}R\hat{V}^T$  corresponding to the indices in  $\bar{J}$  to be zero, except for the  $(0, 0)$ -th element of  $\hat{V}R\hat{V}^T$ .

Because interior point solvers do not scale well, especially when nonnegative or interval cuts are added to the SDP relaxation in (1.5), Oliveira et al. (2018) propose using an ADMM approach. They introduce interval cuts (constraints) and obtain a DNN model. The ADMM approach is further motivated by the natural splitting of variables that arises with facial reduction:

$$\begin{aligned} \min_{R, Y} \quad & \langle L_Q, Y \rangle \\ \text{s.t.} \quad & \mathcal{G}_{\bar{j}}(Y) = u_0 \\ \text{(DNN)} \quad & Y = \hat{V}R\hat{V}^T \\ & R \succeq 0 \\ & 0 \leq Y \leq 1. \end{aligned} \quad (1.9)$$

The output of ADMM is used to compute lower and upper bounds to the original QAP (1.1). For most instances in QAPLIB,<sup>2</sup> Oliveira et al. (2018) obtain competitive lower and upper bounds for the QAP using ADMM. And in several instances, the relaxation and bounds provably find an optimal permutation matrix.

## 1.2. Contributions and Outline

We begin in Section 2 with the modeling and theory. We first give a new joint derivation of the so-called gangster constraints and the facial reduction procedure. Our proposed model for solving (1.9) uses redundant constraints on the variables  $R, Y$ . We include optimality conditions and find explicit values for some of the dual variables by exploiting the redundant constraints.

In Section 3, we derive the modified *rPRSM* for solving the strengthened model. We use redundant constraints to strengthen the subproblems and the lower bounds. We add a randomized perturbation

approach to improve upper bounds. The solution run times are improved by the new dual variable updates and with new termination conditions.

For our numerical results in Section 4, we use data from QAPLIB (Burkard et al. 1991). We show significant improvements over the previous results in Oliveira et al. (2018). Our concluding remarks are in Section 5.

## 2. DNN Relaxation and Optimality

In this section, we present details of our DNN relaxation of the QAP. This is related to the SDP relaxation derived in Zhao et al. (1998) and the DNN relaxation in Oliveira et al. (2018). Our approach is novel in that we see the gangster constraints and facial reduction arise naturally from the relaxation of the row and column sum constraints for  $X \in \Pi$ . The discussion allows us to see the many redundant constraints that can then be used to strengthen our subproblems within our rPRSM algorithm.

### 2.1. Novel Derivation of DNN Relaxation

The derivation of the SDP relaxation in Zhao et al. (1998) starts with the Lagrangian relaxation (dual) and forms the dual of this dual. Then redundant constraints are deleted. We now look at a direct approach for finding this SDP relaxation.

**2.1.1. Gangster Constraints.** Let  $\mathcal{D}_e, \mathcal{Z}$  be the matrix sets of row and column sums equal one, and binary, respectively, that is,

$$\begin{aligned} \mathcal{D}_e &:= \{X \in \mathbb{R}^{n \times n} : Xe = e, X^T e = e\}, \\ \mathcal{Z} &:= \{X \in \mathbb{R}^{n \times n} : X_{ij} \in \{0, 1\}, \forall i, j \in \{1, \dots, n\}\}. \end{aligned}$$

We let  $\mathcal{D} = \mathcal{D}_e \cap \{X \geq 0\}$  denote the *doubly stochastic matrices*. The classical Birkhoff-von Neumann theorem (Birkhoff 1946, von Neumann 1953) states that the permutation matrices are the extreme points of  $\mathcal{D}$ . This leads to the well-known conclusion that the set of  $n$ -by- $n$  permutation matrices,  $\Pi$ , is equal to the intersection:

$$\Pi = \mathcal{D}_e \cap \mathcal{Z}. \quad (2.1)$$

It is of interest that the representation in (2.1) leads to *both* the gangster constraints and facial reduction for the SDP relaxation on the lifted variable  $Y$  in (1.3) and in particular on  $\bar{Y}$ . Not only that, but the row-sum constraints  $Xe = e$ , along with the 0-1 constraint, expressed as  $X \circ X = X$ , give rise to the constraint that the diagonal elements of the off-diagonal blocks of  $\bar{Y}$  are all zero, whereas the column-sum constraint  $X^T e = e$  along with the 0-1 constraints give rise to the constraint that the off-diagonal elements of the diagonal blocks of  $\bar{Y}$  are all zero. The following well-known Lemma 2.1 about complementary slackness (Hadamard orthogonality) is useful.

**Lemma 2.1.** Let  $A, B \in \mathbb{S}^n$ . If  $A$  and  $B$  have nonnegative entries, then

$$\langle A, B \rangle = 0 \Leftrightarrow A \circ B = 0. \quad \square$$

Lemma 2.2 and Corollary 2.1 together show how the representation of  $\Pi$  in (2.1) gives rise to the gangster constraint on the lifted matrix  $Y$  in (1.2). We first find (Hadamard product) exposing vectors in Lemma 2.2 for lifted zero-one vectors.

**Lemma 2.2** (Exposing Vectors). Let  $X \in \mathcal{Z}$  and let  $x := \text{vec}(X)$ . Then the following hold:

1. Row sums:  $Xe_n = e_n \Rightarrow [(e_n e_n^T \otimes I_n) - I_{n^2}] \circ xx^T = 0$ ;
2. Column sums:  $X^T e_n = e_n \Rightarrow [(I_n \otimes e_n e_n^T) - I_{n^2}] \circ xx^T = 0$ .

**Proof.** The structure of the Kronecker allows for:

1. Let  $X \in \mathcal{Z}$  and  $Xe_n = e_n$ . We note that  $X \in \mathcal{Z} \Leftrightarrow x \circ x - x = 0$  and

$$Xe_n = e_n \Leftrightarrow I_n Xe_n = e_n \Leftrightarrow (e_n^T \otimes I_n)x = e_n.$$

We begin by multiplying both sides by  $(e_n^T \otimes I_n)^T = e_n \otimes I_n$ :

$$\begin{aligned} & (e_n^T \otimes I_n)x = e_n \\ \Rightarrow & (e_n \otimes I_n)(e_n^T \otimes I_n)x = (e_n \otimes I_n)e_n = e_{n^2} \\ \Rightarrow & [(e_n \otimes I_n)(e_n^T \otimes I_n) - I_{n^2}]x = e_{n^2} - x \\ \Rightarrow & [(e_n e_n^T \otimes I_n) - I_{n^2}]xx^T = e_{n^2}x^T - xx^T \\ \Rightarrow & \text{tr}([(e_n e_n^T \otimes I_n) - I_{n^2}]xx^T) = \text{tr}(e_{n^2}x^T - xx^T). \end{aligned}$$

Because  $x \circ x = x$ , we have  $\text{tr}(e_{n^2}x^T - xx^T) = 0$ . Therefore, it holds that

$$\text{tr}([(e_n e_n^T \otimes I_n) - I_{n^2}]xx^T) = 0.$$

We note that  $[(e_n e_n^T \otimes I_n) - I_{n^2}]$  and  $xx^T$  are both symmetric and nonnegative. Hence, by Lemma 2.1, we get

$$[(e_n e_n^T \otimes I_n) - I_{n^2}] \circ xx^T = 0.$$

2. The proof for Item 2 is similar.  $\square$

**Corollary 2.1.** Let  $X \in \Pi$ , and let  $Y$  satisfy (1.2). Let  $\mathcal{G}_{\bar{J}}, \bar{J}$  be defined in (1.6) and (1.8). Then the following hold:

1.  $\mathcal{G}_{\bar{J}}(Y) = u_0$ ;
2.  $0 \leq Y \leq 1, Y \succeq 0, \text{rank}(Y) = 1$ .

**Proof.** Note that

- The matrix  $(e_n e_n^T \otimes I_n) - I_{n^2}$  has nonzero entries on the diagonal elements of the off-diagonal blocks; and
- The matrix  $(I_n \otimes e_n e_n^T) - I_{n^2}$  has nonzero entries on the off-diagonal elements of the diagonal blocks.

Therefore, Lemma 2.2, the definition of the gangster indices  $\bar{J}$  in (1.8), and the structure of  $Y$  in (1.2) jointly

give  $\mathcal{G}_{\bar{J}}(Y) = u_0$ ; that is, Item 1 holds. Item 2 follows from (2.1) and the structure of  $Y$  in (1.2).  $\square$

Proposition 2.1 shows that the current gangster index set is the largest possible, in the sense that adding an index implies that at least one element of  $X$  is determined.

**Proposition 2.1.** Suppose that for all  $X \in \Pi$ , and  $Y$  formed from (1.2), there exists an index  $(s, t)$  such that  $Y_{st} = Y_{ts} = 0$ , but  $\{(s, t) \cup (t, s)\} \notin \bar{J}$ ; that is,  $(s, t)$  is added to the gangster set. Then at least one element of  $X$  can be determined. Therefore, the gangster set cannot be increased.

**Proof.** There are two cases to consider.

1. Suppose that  $s = (ij) = t, i, j \geq 1$ , and so we have  $Y_{(ij)(ij)} = 0$ . However,  $\bar{Y} = xx^T$ , by (1.2), implies that  $X_{ij} = 0$ ; this does not hold for all  $X \in \Pi$ , a contradiction; that is, we cannot add a diagonal element of  $Y$  to the gangster set.

2. If  $s \neq t$ , we have  $Y_{st} = 0$ . Because  $X \in \Pi$ , we infer that  $Y_{ss}$  or  $Y_{tt}$  must be zero. The condition  $s \neq t$  and  $\{(s, t) \cup (t, s)\} \notin \bar{J}$  imply that there are two elements in  $X$ , which are not in the same row and column, and the product of them is zero. This clearly does not hold for all  $X \in \Pi$ , a contradiction; that is, as previously noted, we cannot add this element of  $Y$  to the gangster set.  $\square$

**2.1.2. Facially Reduced DNN Relaxations.** We have shown that the representation  $\Pi = \mathcal{D}_e \cap \mathcal{Z}$  gives rise to the gangster constraint and the polyhedral constraint on the variable  $Y$  given in (1.9). As for the derivation of the gangster constraint, we now see that the facial reduction constraint  $Y = \hat{V}R\hat{V}^T$  in (1.9) arises from consideration of an exposing vector. We define

$$H := \begin{bmatrix} e_n^T \otimes I_n \\ I_n \otimes e_n^T \end{bmatrix} \in \mathbb{R}^{2n \times n^2}, \quad (2.2)$$

and

$$K := \begin{bmatrix} -e_{n^2}^T \\ H^T \end{bmatrix} \begin{bmatrix} -e_{n^2} & H \end{bmatrix} = \begin{bmatrix} 2n & -2e_{n^2}^T \\ -2e_{n^2} & H^T H \end{bmatrix} \in \mathbb{S}^{n^2+1}. \quad (2.3)$$

We note that  $H$  arises from the linear equality constraints  $Xe = e, X^T e = e$ . The matrix  $H$  in (2.2) is the well-known matrix in the linear assignment problem with  $\text{rank}(H) = 2n - 1$  and the rows sum up to  $2e_{n^2}^T$ . Then  $\text{rank}(K) = 2n - 1$  as well. Moreover, Lemma 2.3 is clear.

**Lemma 2.3.** Let  $H$  be given in (2.2), and let

$$X \in \mathbb{R}^{n \times n}, x = \text{vec}(X), Y_x = \begin{pmatrix} 1 \\ x \end{pmatrix} \begin{pmatrix} 1 \\ x \end{pmatrix}^T.$$

Then we have

$$Xe = e, X^T e = e \Rightarrow Y_x K = 0,$$

where  $K$  is defined in (2.3).

**Proof.** From the definition of  $H$  in (2.2), we have

$$\begin{aligned} Xe = e, X^T e = e &\Leftrightarrow Hx = e \\ &\Leftrightarrow \begin{pmatrix} 1 \\ x \end{pmatrix}^T \begin{pmatrix} -e^T \\ H^T \end{pmatrix} = 0 \\ &\Rightarrow \begin{pmatrix} 1 \\ x \end{pmatrix} \begin{pmatrix} 1 \\ x \end{pmatrix}^T \begin{pmatrix} -e^T \\ H^T \end{pmatrix} \begin{pmatrix} -e^T \\ H^T \end{pmatrix}^T = 0 \\ &\Leftrightarrow Y_x K = 0. \quad \square \end{aligned}$$

From Lemma 2.3,  $K$  is an *exposing vector* for all feasible  $Y_x$  (Drusvyatskiy and Wolkowicz 2017). Then we can choose a full column rank  $\hat{V}$  with the range equal to the nullspace of  $K$  and obtain *facial reduction*; that is, all feasible  $Y$  for the SDP relaxation satisfy

$$Y \in \hat{V} \mathbb{S}_+^{(n-1)^2+1} \hat{V}^T \trianglelefteq \mathbb{S}_+^{n^2+1}.$$

There are clearly many choices for  $\hat{V}$ . We present one in Proposition 2.2 from Zhao et al. (1998). However, in our implementations, we follow Oliveira et al. (2018) and use one with orthonormal columns.

**Proposition 2.2** (Zhao et al. 1998). Let

$$\hat{V} = \left[ \begin{array}{c|c} 1 & 0 \\ \hline \frac{1}{n} e_{n^2} & V_e \otimes V_e \end{array} \right] \in \mathbb{R}^{(n^2+1) \times ((n-1)^2+1)},$$

$$V_e = \begin{bmatrix} I_{n-1} \\ -e_{n-1}^T \end{bmatrix} \in \mathbb{R}^{n \times (n-1)},$$

and let  $K$  be given as in (2.3). Then we have  $\text{range}(\hat{V}) = \text{null}(K)$ .  $\square$

Our DNN relaxation has the lifted  $Y$  from (1.2) and (1.4) and the FR variable  $R$  from (1.5). The relation between  $R, Y$  provides the natural *splitting*:

$$\begin{aligned} p_{\text{DNN}}^* &= \min \langle L_Q, Y \rangle \\ \text{s.t.} \quad &\mathcal{G}_j(Y) = u_0 \\ &Y = \hat{V} R \hat{V}^T \\ &R \succeq 0 \\ &0 \leq Y \leq 1. \end{aligned} \tag{2.4}$$

A strictly feasible  $\hat{R} > 0$  for the facially reduced SDP relaxation is given in Zhao et al. (1998), based on the barycenter  $\hat{Y}$  of the lifted matrices  $Y$  in (1.2). Therefore,  $0 < \hat{Y}_{\bar{j}^c} < 1$ , and this pair  $(\hat{R}, \hat{Y})$  is strictly feasible in (2.4).

**2.1.3. Redundant Constraints.** We continue in this section with some redundant constraints for Model (2.4) that are useful in the subproblems and in prespecifying values of some dual variables. Although the constraints are redundant for Model (2.4), they are not redundant when the subproblems of rPRSM are considered as independent optimization problems. To derive those constraints, we first recall three linear transformations defined in Zhao et al. (1998).

**Definition 2.1** (Zhao et al. 1998, p. 80). Let  $Y \in \mathbb{S}^{n^2+1}$  be blocked as in (1.3). We define the linear transformation  $\text{b}^0 \text{diag}(Y) : \mathbb{S}^{n^2+1} \rightarrow \mathbb{S}^n$  by the sum of the  $n$ -by- $n$  diagonal blocks of  $Y$ , that is,

$$\text{b}^0 \text{diag}(Y) := \sum_{k=1}^n Y_{(kk)} \in \mathbb{S}^n.$$

We define the linear transformation  $\text{o}^0 \text{diag}(Y) : \mathbb{S}^{n^2+1} \rightarrow \mathbb{S}^n$  by the trace of the block  $\tilde{Y}_{(ij)}$ , that is,

$$\text{o}^0 \text{diag}(Y) := \left( \text{tr}(\tilde{Y}_{(ij)}) \right)_{ij} \in \mathbb{S}^n.$$

We define the linear transformation  $\text{arrow}(Y) : \mathbb{S}^{n^2+1} \rightarrow \mathbb{R}^{n^2+1}$  by the difference of the first column and diagonal of  $Y$ :

$$\text{arrow}(Y) := (Y_{(:,1)} - \text{diag}(Y)) \in \mathbb{R}^{n^2+1}.$$

With Definition 2.1, the following lemma can be derived from Zhao et al. (1998, lemma 3.1). Lemma 2.4 indeed shows three redundant constraints of (2.4).

**Lemma 2.4** (Zhao et al. 1998, Lemma 3.1). Let  $V$  be any full column rank matrix such that  $\text{range}(V) = \text{range}(\hat{V})$ , where  $\hat{V}$  is given in Proposition 2.2. Suppose  $Y = VRV^T$  and  $\mathcal{G}_j(Y) = u_0$  hold. Then we have  $\text{arrow}(Y) = 0$ ,  $\text{b}^0 \text{diag}(Y) = I_n$  and  $\text{o}^0 \text{diag}(Y) = I_n$ .

Proposition 2.3 shows that the constraint  $\text{tr}(R) = n + 1$  is also redundant for Model (2.4).

**Proposition 2.3.** With orthonormal  $\hat{V}$  whose range is equal to  $\text{range}(K)$ , the constraints  $Y = \hat{V} R \hat{V}^T$ ,  $R \succeq 0$  and  $Y \in \mathcal{Y}$  yield that  $\text{tr}(R) = n + 1$ .

**Proof.** By Lemma 2.4,  $\text{b}^0 \text{diag}(Y) = I_n$  hold. Then with  $Y_{00} = 1$ , we see that  $\text{tr}(Y) = n + 1$ . By cyclicity of the trace operator and  $\hat{V}^T \hat{V} = I$ , we see that

$$\text{tr}(R) = \text{tr}(R \hat{V}^T \hat{V}) = \text{tr}(\hat{V} R \hat{V}^T) = \text{tr}(Y) = n + 1. \quad \square$$

**Remark 2.1.** We take advantage of this in the corresponding  $R$ -subproblem and the computation of the lower bound of QAP. We could add more redundant constraints to (DNN). For example, we could strengthen the relaxation by restricting each row/column (ignoring the first row/column) to be a multiple of a vectorized doubly stochastic matrix.

## 2.2. Main Model and Optimality Conditions

We now derive the main splitting model. We define the cone and polyhedral constraints, respectively, as

$$\mathcal{R} := \left\{ R \in \mathbb{S}^{(n-1)^2+1} : R \succeq 0, \text{tr}(R) = n + 1 \right\}, \quad (2.5)$$

and

$$\begin{aligned} \mathcal{Y} := \{ Y \in \mathbb{S}^{n^2+1} : \mathcal{G}_j(Y) = u_0, 0 \leq Y \leq 1, \\ \text{b}^0 \text{diag}(Y) = I, \text{o}^0 \text{diag}(Y) = I, \\ \text{arrow}(Y) = 0 \}. \end{aligned} \quad (2.6)$$

Replacing the constraints in (2.4) with (2.5) and (2.6), we obtain the following DNN relaxation that we solve using rPRSM:

$$\begin{aligned} (DNN) \quad p_{DNN}^* := \min_{R, Y} \quad & \langle L_Q, Y \rangle \\ \text{s.t.} \quad & Y = \hat{V}R\hat{V}^T \\ & R \in \mathcal{R} \\ & Y \in \mathcal{Y}. \end{aligned} \quad (2.7)$$

The following property of feasible points  $Y \in \mathcal{Y}$  in Proposition 2.4 is used in the computation of the  $Y$ -subproblem of our algorithm.

**Proposition 2.4.** *For any  $Y \in \mathcal{Y}$ , let  $\bar{X} = \text{Mat}(\text{diag}(\bar{Y})) \in \mathbb{R}^{n \times n}$  be the matrix formed from the diagonal of  $\bar{Y}$  after ignoring the  $(0,0)$ -th element.  $\bar{X} \in \mathcal{D}$ . Moreover, this holds for the first row (and column) of  $Y$ .*

**Proof.** From the  $\mathcal{Y}$  constraints  $\text{b}^0 \text{diag}(Y) = I, \text{o}^0 \text{diag}(Y) = I$ , respectively, we get  $\sum_{k=1}^n \text{diag}(Y_{(kk)}) = e$  and  $\text{tr}(Y_{(kk)}) = 1, \forall i \in \{1, \dots, k\}$ , respectively. Then by the definition of  $\bar{X}$ , we immediately have  $\bar{X}e = e$  and  $\bar{X}^T e = e$ . The nonnegativity constraint in  $\mathcal{Y}$  implies  $\bar{X} \geq 0$ . Therefore,  $\bar{X} \in \mathcal{D}$ .

The equivalent result for the first row and column follows from the arrow constraint.  $\square$

**Remark 2.2** ((Doubly) Stochastic Optimal  $Y$ ). Proposition 2.4 shows that for any feasible  $Y \in \mathcal{Y}$ , when ignoring the  $(0,0)$ -th element, then the diagonal, the first row, and the first column of  $Y$ , can all be reshaped into doubly stochastic matrices. In fact, in addition to this, if  $Y \in \mathcal{Y}, v \in \mathbb{R}^{n^2+1}$  is a nonnegative random vector, and we set  $w = Yv$  with  $w \leftarrow w/w_1$ , then  $X = \text{Mat}w$  satisfies the row and column sum constraints. Therefore, for an optimal  $Y$  and choosing  $v \geq 0$ , this  $X$  is doubly stochastic, and if  $v$  is a unit vector, then we see that every column of  $Y$  is doubly stochastic.

Define the orthogonal projection  $P_V = \hat{V}\hat{V}^T$ , and let  $\alpha, \delta > 0$  be the shift and scale parameters. Note that

$Y = \hat{V}R\hat{V}^T$  implies

$$\begin{aligned} \delta \langle L_Q, Y \rangle &= \delta \langle L_Q + \alpha I, Y \rangle - (n+1)\delta\alpha \\ &= \delta \langle L_Q + \alpha I, P_V Y P_V \rangle - (n+1)\delta\alpha \\ &= \langle \delta(P_V L_Q P_V + \alpha I), Y \rangle - (n+1)\delta\alpha. \end{aligned} \quad (2.8)$$

Therefore, the original objective value is

$$\langle L_Q, Y \rangle = \frac{1}{\delta} \langle \delta(P_V L_Q P_V + \alpha I), Y \rangle - (n+1)\alpha.$$

By abuse of notation, we use

$$L_Q \leftarrow \delta(P_V L_Q P_V + \alpha I). \quad (2.9)$$

We use these values for our lower and upper bounds, because the data are integer valued, and we can improve the bounds by rounding.

The Lagrangian function of Model (2.7) is

$$\mathcal{L}(R, Y, Z) = \langle L_Q, Y \rangle + \langle Z, Y - \hat{V}R\hat{V}^T \rangle. \quad (2.10)$$

Because a strictly feasible  $\hat{R}$ , with  $\hat{Y} = \hat{V}\hat{R}\hat{V}^T$ , exists, we conclude that the following first-order optimality conditions for Model (2.7) hold:

$$0 \in -\hat{V}^T Z \hat{V} + \mathcal{N}_{\mathcal{R}}(R), \quad (\text{dual } R \text{ feasibility}), \quad (2.11a)$$

$$0 \in L_Q + Z + \mathcal{N}_{\mathcal{Y}}(Y), \quad (\text{dual } Y \text{ feasibility}), \quad (2.11b)$$

$$Y = \hat{V}R\hat{V}^T, \quad R \in \mathcal{R}, Y \in \mathcal{Y}, \quad (\text{primal feasibility}), \quad (2.11c)$$

where the set  $\mathcal{N}_{\mathcal{R}}(R)$  (respectively,  $\mathcal{N}_{\mathcal{Y}}(Y)$ ) is the normal cone to the set  $\mathcal{R}$  (respectively,  $\mathcal{Y}$ ) at  $R$  (respectively,  $Y$ ). By the definition of the normal cone, we can easily obtain the following Proposition 2.5.

**Proposition 2.5** (Characterization of Optimality for (2.7)). *The primal-dual  $R, Y, Z$  are optimal for (2.7) if, and only if, (2.11) holds if, and only if,*

$$R = \mathcal{P}_{\mathcal{R}}(R + \hat{V}^T Z \hat{V}), \quad (2.12a)$$

$$Y = \mathcal{P}_{\mathcal{Y}}(Y - L_Q - Z), \quad (2.12b)$$

$$Y = \hat{V}R\hat{V}^T. \quad (2.12c)$$

We use (2.12) as one of the stopping criteria of the rPRSM in our numerical experiments.

**2.2.1. Dual Multiplier.** As in all constrained optimization, the Lagrange (dual) multiplier, here denoted  $Z$ , is essential in finding an optimal solution and critical in obtaining strong lower bounds. Moreover, a compact set of dual multipliers is an indication of stability for the primal problem. If the optimal  $Z$  would be completely known for the Lagrangian function in (2.10), then the primal feasibility equation  $Y = \hat{V}R\hat{V}^T$  can be ignored in the optimality conditions in (2.11). We now present properties on  $Z$  that are exploited in



our algorithm in Section 3. Theorem 2.1 shows that there exists a dual multiplier  $Z \in \mathbb{S}^{n^2+1}$  of Model (2.7) that, except for the  $(0,0)$ -th entry, has a known diagonal, first column, and first row and known elements in the redundant gangster positions. This allows for faster convergence for our algorithm of Section 3.

**Theorem 2.1.** Let  $E_A = \begin{bmatrix} 1 & 0 \\ 0 & E_{n^2} - I_{n^2} - I_{J_R} \end{bmatrix}$ , where  $I_{J_R}$  is the zero matrix except for one in the positions of the redundant gangster elements  $J_R$ , Item 3, on page 3. Let

$$\mathcal{Y}_A := \left\{ Y \in \mathbb{S}^{n^2+1} : \mathcal{G}_{\setminus J_R}(Y) = E_{00}, 0 \leq E_A \circ Y \leq 1, \right. \\ \left. \text{arrow}(Y) = 0 \right\},$$

and let

$$\mathcal{Z}_A := \left\{ Z \in \mathbb{S}^{n^2+1} : (Z + L_Q)_{ij} = 0, \forall i, j \text{ in arrow positions,} \right. \\ \left. \text{and } \forall ij \in J_R \right\}.$$

Consider the following problem:

$$\min_{R, Y} \left\{ \langle L_Q, Y \rangle : Y = \hat{V}R\hat{V}^T, R \in \mathcal{R}, Y \in \mathcal{Y}_A \right\}. \quad (2.13)$$

Then the following holds:

1. The feasible sets of (2.7) and (2.13) are the same.
2. Let  $(R^*, Y^*, Z^*)$  be an optimal primal-dual solution for (2.13). Then  $Z^* \in \mathcal{Z}_A$ .
3. Let  $(R^*, Y^*)$  be an optimal pair for (2.7). Then there exists  $Z^* \in \mathcal{Z}_A$  such that  $(R^*, Y^*, Z^*)$  solves (2.11); that is, they are an optimal primal-dual solution for (2.7).

**Proof.** Note that  $\mathcal{Y} \subset \mathcal{Y}_A$ , where we remove the  $\text{b}^0 \text{diag}$ ,  $\text{o}^0 \text{diag}$  and the polyhedral constraints on the diagonal, the first row and column, the redundant gangster constraints, but leave the arrow constraint. Clearly, every feasible solution of (2.7) is feasible for (2.13) because  $\mathcal{Y} \subset \mathcal{Y}_A$ . Consider a feasible pair  $(R, Y)$  to (2.13). By Item 2 of Lemma 2.4 and the positive semidefiniteness of  $Y = \hat{V}R\hat{V}^T$ , we have that  $\text{b}^0 \text{diag}(Y) = I_n$  and the elements of the diagonal of  $Y$  are in the interval  $[0, 1]$ . In addition, because  $\text{arrow}(Y) = 0$ , the elements of the first row and column of  $Y$  are also in the interval  $[0, 1]$ . Thus, we conclude that  $Y \in \mathcal{Y}$  and (2.7) and (2.13) have equal feasible sets and therefore are equivalent problems. Thus, the first assertion is proved.

Let  $(R^*, Y^*, Z^*)$  be an optimal primal-dual solution for (2.13). Then according to the first-order optimality condition, we have

$$0 \in -\hat{V}^T Z^* \hat{V} + \mathcal{N}_{\mathcal{R}}(R^*), \quad (2.14a)$$

$$0 \in L_Q + Z^* + \mathcal{N}_{\mathcal{Y}_A}(Y^*), \quad (2.14b)$$

$$Y^* = \hat{V}R^*\hat{V}^T, \quad R^* \in \mathcal{R}, Y^* \in \mathcal{Y}_A. \quad (2.14c)$$

By the definition of the normal cone, we have

$$0 \in L_Q + Z^* + \mathcal{N}_{\mathcal{Y}_A}(Y^*) \\ \Leftrightarrow \langle Y - Y^*, L_Q + Z^* \rangle \geq 0, \quad \forall Y \in \mathcal{Y}_A.$$

Because the diagonal and the first column and row of  $Y \in \mathcal{Y}_A$  except for the first element are unconstrained, as are all the redundant gangster positions, we see that

$$(E_{n^2+1} - E_A) \circ (Z^* + L_Q) = 0.$$

This implies that  $Z^* \in \mathcal{Z}_A$  and proves Item 2.

To prove Item 3, it suffices to show that the triple  $(R^*, Y^*, Z^*)$  also solves (2.11). We note that (2.14a) and (2.14c) imply that (2.11a) and (2.11c) hold with  $(R^*, Y^*, Z^*)$  in the place of  $(R, Y, Z)$ . In addition, because  $Y^* \in \mathcal{Y} \subseteq \mathcal{Y}_A$ , we see that  $\mathcal{N}_{\mathcal{Y}_A}(Y^*) \subseteq \mathcal{N}_{\mathcal{Y}}(Y^*)$ . This together with (2.14b) shows that (2.11b) holds with  $(Y^*, Z^*)$  in the place of  $(Y, Z)$ . Thus, we have shown that  $(R^*, Y^*, Z^*)$  also solves (2.11).  $\square$

**Remark 2.3.** Dual variables are sensitivity coefficients for the optimal value with respect to perturbations in the constraints. Before scaling,  $L$  has zeros in the positions identified in  $\mathcal{Z}_A$ , as it is formed from the Kronecker product of adjacency matrices.

### 3. rPRSM Algorithm

We now present the details of a modification of the so-called strictly contractive PRSM or symmetric ADMM (He et al. 2014, Li and Yuan 2015). Our modification involves redundant constraints on subproblems and on the update of dual variables.

#### 3.1. Outline and Convergence for rPRSM

The augmented Lagrangian function for (2.7) with Lagrange multiplier  $Z$  is

$$\mathcal{L}_A(R, Y, Z) = \langle L_Q, Y \rangle + \left\langle Z, Y - \hat{V}R\hat{V}^T \right\rangle \\ + \frac{\beta}{2} \left\| \left\| Y - \hat{V}R\hat{V}^T \right\|_F \right\|^2, \quad (3.1)$$

where  $\beta$  is a positive penalty parameter.

Define  $\mathcal{Z}_0 := \{Z \in \mathbb{S}^{n^2+1} : Z_{i,i} = 0, Z_{0,i} = Z_{i,0} = 0, i = 1, \dots, n^2\}$  and let  $\mathcal{P}_{\mathcal{Z}_0}$  be the projection onto the set  $\mathcal{Z}_0$ . Our proposed algorithm reads as follows.

**Algorithm 1** (rPRSM for DNN in (2.7))

**Initialize:**  $\mathcal{L}_A$  augmented Lagrangian in (3.1);  $\gamma \in (0, 1)$ , under-relaxation parameter;  $\beta \in (0, \infty)$ , penalty parameter;  $\mathcal{R}, \mathcal{Y}$  subproblem sets from (2.5);  $Y^0$ ; and  $Z^0 \in \mathcal{Z}_A$ ;

**while** tolerances not met **do**

$$\begin{aligned}
 R^{k+1} &= \underset{R \in \mathcal{R}}{\operatorname{arg\,min}} \mathcal{L}_A(R, Y^k, Z^k) \\
 Z^{k+\frac{1}{2}} &= Z^k + \gamma\beta \cdot \mathcal{P}_{\mathcal{Z}_0} \left( Y^k - \hat{V}R^{k+1}\hat{V}^T \right) \\
 Y^{k+1} &= \underset{Y \in \mathcal{Y}}{\operatorname{arg\,min}} \mathcal{L}_A(R^{k+1}, Y, Z^{k+\frac{1}{2}}) \\
 Z^{k+1} &= Z^{k+\frac{1}{2}} + \gamma\beta \cdot \mathcal{P}_{\mathcal{Z}_0} \left( Y^{k+1} - \hat{V}R^{k+1}\hat{V}^T \right)
 \end{aligned}$$

**end while**

**Remark 3.1.** Algorithm 1 can be summarized as follows: alternate minimization of variables  $R$  and  $Y$  interlaced by the dual variable  $Z$  update. Before discussing the convergence of Algorithm 1, we point out the following. The  $R$ -update and the  $Y$ -update in Algorithm 1 are well defined; that is, the subproblems involved have unique solutions. This follows from the strict convexity of  $\mathcal{L}_A$  with respect to  $R$ ,  $Y$  and the convexity and compactness of the sets  $\mathcal{R}$  and  $\mathcal{Y}$ . Many of the constraints are redundant in the SDP part of the problem, for example, the trace on  $R$ , and the  $\mathbf{b}^0 \operatorname{diag}$ ,  $\mathbf{o}^0 \operatorname{diag}$ ,  $\operatorname{arrow}$  on  $Y$ . However, these constraints are not redundant within the subproblems themselves and are inexpensive to include. For example, the trace constraint enforces compactness in the  $R$ -subproblem. They improve the rate of convergence and the quality of the  $Y$  when stopping the rPRSM algorithm early.

We also note that, in Algorithm 1, we update the dual variable  $Z$  both after the  $R$ -update and the  $Y$ -update. This pattern of update in our Algorithm 1 is closely related to the strictly contractive PRSM (He et al. 2014, Li and Yuan 2015). Indeed, we show in Theorem 3.1 that our algorithm can be viewed as a version of *semiproximal strictly contractive PRSM* (Gu et al. 2015, Li and Yuan 2015), applied to (3.2). Hence, the convergence of our algorithm can be deduced from the general convergence theory of semiproximal strictly contractive PRSM.

**Theorem 3.1.** Let  $\{R^k\}, \{Y^k\}, \{Z^k\}$  be the sequences generated by Algorithm 1. Then the sequence  $\{(R^k, Y^k)\}$  converges to a primal optimal pair  $(R^*, Y^*)$  of (2.7), and  $\{Z^k\}$  converges to an optimal dual solution  $Z^* \in \mathcal{Z}_A$ .

**Proof.** The proof is divided into two steps. In the first step, we consider the convergence of the semiproximal restricted contractive PRSM in Gu et al. (2015) and Li and Yuan (2015) applied to the following problem (3.2), where  $\mathcal{P}_{\mathcal{Z}_0^c}$  is the projection onto the orthogonal complement of  $\mathcal{Z}_0$ , that is,

$$\mathcal{P}_{\mathcal{Z}_0^c} = I - \mathcal{P}_{\mathcal{Z}_0}:$$

$$\begin{aligned}
 \min_{R, Y} \quad & \langle L_Q, \mathcal{P}_{\mathcal{Z}_0}(Y) + \mathcal{P}_{\mathcal{Z}_0^c}(\hat{V}R\hat{V}^T) \rangle \\
 \text{s.t.} \quad & \mathcal{P}_{\mathcal{Z}_0}(Y) = \mathcal{P}_{\mathcal{Z}_0}(\hat{V}R\hat{V}^T) \\
 & R \in \mathcal{R} \\
 & Y \in \mathcal{Y}.
 \end{aligned} \tag{3.2}$$

We show that the sequence generated by the semiproximal restricted contractive PRSM in Gu et al. (2015) and Li and Yuan (2015) converges to a *Karush-Kuhn-Tucker (KKT)* point of (2.7). In the second step, we show that the sequence generated by Algorithm 1 is identical with the sequence generated by the semiproximal restricted contractive PRSM applied to (3.2).

Step 1. We apply the semiproximal strictly contractive PRSM given in Gu et al. (2015) and Li and Yuan (2015) to (3.2). Let  $(\tilde{R}^0, \tilde{Y}^0, \tilde{Z}^0) := (R^0, Y^0, Z^0)$ , where  $R^0$  and  $Y^0$  are chosen to satisfy (2.7) and  $Z^0 \in \mathcal{Z}_A$ . Consider the following update:

$$\begin{aligned}
 \tilde{R}^{k+1} &= \underset{R \in \mathcal{R}}{\operatorname{arg\,min}} \left( \langle L_Q, \mathcal{P}_{\mathcal{Z}_0^c}(\hat{V}R\hat{V}^T) \rangle - \langle \tilde{Z}^k, \mathcal{P}_{\mathcal{Z}_0}(\hat{V}R\hat{V}^T) \rangle \right. \\
 &\quad \left. + \frac{\beta}{2} \left\| \mathcal{P}_{\mathcal{Z}_0}(\tilde{Y}^k - \hat{V}R\hat{V}^T) \right\|_F^2 \right. \\
 &\quad \left. + \frac{\beta}{2} \left\| \mathcal{P}_{\mathcal{Z}_0^c}(\hat{V}R\hat{V}^T - \hat{V}\tilde{R}^k\hat{V}^T) \right\|_F^2 \right) \\
 \tilde{Z}^{k+\frac{1}{2}} &= \tilde{Z}^k + \gamma\beta \mathcal{P}_{\mathcal{Z}_0} \left( \tilde{Y}^k - \hat{V}\tilde{R}^{k+1}\hat{V}^T \right), \\
 \tilde{Y}^{k+1} &\in \underset{Y \in \mathcal{Y}}{\operatorname{arg\,min}} \left( \langle L_Q, \mathcal{P}_{\mathcal{Z}_0}(Y) \rangle + \langle \tilde{Z}^{k+\frac{1}{2}}, \mathcal{P}_{\mathcal{Z}_0}(Y) \rangle \right. \\
 &\quad \left. + \frac{\beta}{2} \left\| \mathcal{P}_{\mathcal{Z}_0} \left( Y - \hat{V}\tilde{R}^{k+1}\hat{V}^T \right) \right\|_F^2 \right) \\
 \tilde{Z}^{k+1} &= \tilde{Z}^{k+\frac{1}{2}} + \gamma\beta \mathcal{P}_{\mathcal{Z}_0} \left( \tilde{Y}^{k+1} - \hat{V}\tilde{R}^{k+1}\hat{V}^T \right),
 \end{aligned} \tag{3.3}$$

where  $\gamma \in (0, 1)$  is an under-relaxation parameter. The  $R$ -update in (3.3) is well defined because the subproblem involved is a strongly convex problem. By completing the square in the  $Y$ -subproblem, we have that

$$\tilde{Y}^{k+1} \in \underset{Y \in \mathcal{Y}}{\operatorname{arg\,min}} \left\| \mathcal{P}_{\mathcal{Z}_0}(Y) - \left( \mathcal{P}_{\mathcal{Z}_0}(\hat{V}\tilde{R}^{k+1}\hat{V}^T) - \frac{1}{\beta} (L_Q + \tilde{Z}^{k+\frac{1}{2}}) \right) \right\|_F^2.$$

We note that  $\mathcal{P}_{\mathcal{Z}_0}(\tilde{Y}^{k+1})$  is uniquely determined with

$$\mathcal{P}_{\mathcal{Z}_0}(\tilde{Y}^{k+1}) = \mathcal{P}_{\mathcal{Z}_0} \left( \hat{V}\tilde{R}^{k+1}\hat{V}^T \right) - \frac{1}{\beta} (L_Q + \tilde{Z}^{k+\frac{1}{2}}),$$

whereas  $\mathcal{P}_{Z_0^c}(\tilde{Y}^{k+1})$  can be chosen to be

$$\mathcal{P}_{Z_0^c}(\tilde{Y}^{k+1}) = \mathcal{P}_{Z_0^c}(\hat{V}\tilde{R}^{k+1}\hat{V}^T), \quad \forall k \geq 0. \quad (3.4)$$

Finally, one can also deduce by induction that  $\tilde{Z}^k \in \mathcal{Z}_A$ , for all  $k$ , because  $Z^0 \in \mathcal{Z}_A$ . From the general convergence theory of semiproximal strictly contractive PRSM given in Gu et al. (2015) and Li and Yuan (2015), we have

$$(\tilde{R}^k, \tilde{Y}^k, \tilde{Z}^k) \rightarrow (R^*, Y^*, Z^*) \in \mathcal{R} \times \mathcal{Y} \times \mathcal{Z}_A,$$

where the convergence of  $\{\tilde{R}^k\}$  follows from the injectivity of the map  $R \mapsto \hat{V}R\hat{V}^T$ . Thus, the triple  $(R^*, Y^*, Z^*)$  solves the optimality condition for (3.2), that is,

$$0 \in \hat{V}^T \mathcal{P}_{Z_0^c}(L_Q)\hat{V} - \hat{V}^T \mathcal{P}_{Z_0}(Z^*)\hat{V} + \mathcal{N}_{\mathcal{R}}(R^*), \quad (3.5a)$$

$$0 \in \mathcal{P}_{Z_0}(L_Q) + \mathcal{P}_{Z_0}(Z^*) + \mathcal{N}_{\mathcal{Y}}(Y^*), \quad (3.5b)$$

$$\mathcal{P}_{Z_0}(Y^*) = \mathcal{P}_{Z_0}(\hat{V}R^*\hat{V}^T). \quad (3.5c)$$

Because we update  $\mathcal{P}_{Z_0^c}(\tilde{Y}^k)$  by (3.4), we also have that

$$\mathcal{P}_{Z_0^c}(Y^*) = \mathcal{P}_{Z_0^c}(\hat{V}R^*\hat{V}^T). \quad (3.6)$$

Next we show that the triple  $(R^*, Y^*, Z^*)$  is also a KKT point of Model (2.7). First, it follows from (3.5c) and (3.6) that

$$Y^* = \hat{V}R^*\hat{V}^T.$$

Second, we can deduce from (3.5a), (3.5b), and  $Z^* \in \mathcal{Z}_A$  that

$$0 \in -\hat{V}^T Z^* \hat{V} + \mathcal{N}_{\mathcal{R}}(R^*) \quad \text{and} \quad 0 \in L_Q + Z^* + \mathcal{N}_{\mathcal{Y}}(Y^*).$$

Hence, we have shown that the sequence generated by (3.3) and (3.4) converges to a KKT point of Model (2.7).

Step 2. We now claim that the sequence  $\{(\tilde{R}^k, \tilde{Z}^{k-\frac{1}{2}}, \tilde{Y}^k, \tilde{Z}^k)\}$  generated by (3.3) and (3.4), starting from  $(\tilde{R}^0, \tilde{Y}^0, \tilde{Z}^0) := (R^0, Y^0, Z^0)$ , is identical to the sequence  $\{(R^k, Z^{k-\frac{1}{2}}, Y^k, Z^k)\}$  given by Algorithm 1. We prove by induction. First, we clearly have  $(\tilde{R}^0, \tilde{Y}^0, \tilde{Z}^0) = (R^0, Y^0, Z^0)$  by the definition. Suppose that  $(\tilde{R}^k, \tilde{Y}^k, \tilde{Z}^k) = (R^k, Y^k, Z^k)$  for some  $k \geq 0$ . Because  $\tilde{Z}^k \in \mathcal{Z}_A$  and (3.4) holds,

we can rewrite the  $R$ -subproblem in (3.3) as follows:

$$\begin{aligned} & \operatorname{argmin}_{R \in \mathcal{R}} \langle L_Q, \mathcal{P}_{Z_0^c}(\hat{V}R\hat{V}^T) \rangle - \langle \tilde{Z}^k, \mathcal{P}_{Z_0}(\hat{V}R\hat{V}^T) \rangle \\ & + \frac{\beta}{2} \left\| \mathcal{P}_{Z_0}(\tilde{Y}^k - \hat{V}R\hat{V}^T) \right\|_F^2 + \frac{\beta}{2} \left\| \mathcal{P}_{Z_0^c}(\hat{V}\tilde{R}^k\hat{V}^T - \hat{V}R\hat{V}^T) \right\|_F^2 \\ & = \operatorname{argmin}_{R \in \mathcal{R}} \langle \mathcal{P}_{Z_0^c}(L_Q) - \mathcal{P}_{Z_0}(\tilde{Z}^k), \hat{V}R\hat{V}^T \rangle \\ & + \frac{\beta}{2} \left\| \mathcal{P}_{Z_0}(\tilde{Y}^k - \hat{V}R\hat{V}^T) \right\|_F^2 + \frac{\beta}{2} \left\| \mathcal{P}_{Z_0^c}(\hat{V}\tilde{R}^k\hat{V}^T - \hat{V}R\hat{V}^T) \right\|_F^2 \\ & = \operatorname{argmin}_{R \in \mathcal{R}} \langle -\mathcal{P}_{Z_0^c}(\tilde{Z}^k) - \mathcal{P}_{Z_0}(\tilde{Z}^k), \hat{V}R\hat{V}^T \rangle \\ & + \frac{\beta}{2} \left\| \tilde{Y}^k - \hat{V}R\hat{V}^T \right\|_F^2 \\ & = \operatorname{argmin}_{R \in \mathcal{R}} - \langle \tilde{Z}^k, \hat{V}R\hat{V}^T \rangle + \frac{\beta}{2} \left\| \tilde{Y}^k - \hat{V}R\hat{V}^T \right\|_F^2, \end{aligned}$$

where the second “=” is because of  $\tilde{Z}^k \in \mathcal{Z}_A$  and (3.4). This is equivalent to the  $R$ -subproblem in Algorithm 1, because  $\tilde{Z}^k = Z^k$  and  $\tilde{Y}^k = Y^k$  by the induction hypothesis. This shows that  $\tilde{R}^{k+1} = R^{k+1}$ , and it follows that  $\tilde{Z}^{k+\frac{1}{2}} = Z^{k+\frac{1}{2}}$ . Because  $Z^{k+\frac{1}{2}} \in \mathcal{Z}_A$ , we can rewrite the  $Y$ -subproblem in Algorithm 1 as

$$\begin{aligned} & \operatorname{argmin}_{Y \in \mathcal{Y}} \langle L_Q + Z^{k+\frac{1}{2}}, Y \rangle + \frac{\beta}{2} \left\| Y - \hat{V}R^{k+1}\hat{V}^T \right\|_F^2 \\ & = \operatorname{argmin}_{Y \in \mathcal{Y}} \langle \mathcal{P}_{Z_0}(L_Q + Z^{k+\frac{1}{2}}), Y \rangle + \frac{\beta}{2} \left\| \mathcal{P}_{Z_0}(Y - \hat{V}R^{k+1}\hat{V}^T) \right\|_F^2 \\ & + \frac{\beta}{2} \left\| \mathcal{P}_{Z_0^c}(Y - \hat{V}R^{k+1}\hat{V}^T) \right\|_F^2 \\ & = \operatorname{argmin}_{Y \in \mathcal{Y}} \langle L_Q, \mathcal{P}_{Z_0}(Y) \rangle + \langle Z^{k+\frac{1}{2}}, \mathcal{P}_{Z_0}(Y) \rangle \\ & + \frac{\beta}{2} \left\| \mathcal{P}_{Z_0}(Y - \hat{V}R^{k+1}\hat{V}^T) \right\|_F^2 + \frac{\beta}{2} \left\| \mathcal{P}_{Z_0^c}(Y - \hat{V}R^{k+1}\hat{V}^T) \right\|_F^2, \end{aligned}$$

where the first “=” is because of  $Z^{k+\frac{1}{2}} \in \mathcal{Z}_A$ . Hence, with  $\tilde{R}^{k+1} = R^{k+1}$  and  $\tilde{Z}^{k+\frac{1}{2}} = Z^{k+\frac{1}{2}}$ , we have that the previous subproblem generates  $\tilde{Y}^{k+1}$  defined in (3.3) and (3.4). Thus, we have  $\tilde{Y}^{k+1} = Y^{k+1}$ , and it follows that  $\tilde{Z}^{k+1} = Z^{k+1}$  holds. This completes the proof for  $\{(R^k, Y^k, Z^k)\}_{k \in \mathbb{N}} \equiv \{(\tilde{R}^k, \tilde{Y}^k, \tilde{Z}^k)\}_{k \in \mathbb{N}}$ , and the alleged convergence behavior of  $\{(R^k, Y^k, Z^k)\}$  follows from that of  $\{(\tilde{R}^k, \tilde{Y}^k, \tilde{Z}^k)\}$ .  $\square$

**Remark 3.2.** There are numerous studies on the convergence rate of splitting methods, in particular on ADMM and its variants. For example, worst-case  $\mathcal{O}(1/k)$  convergence measured by the iteration complexity has been established for ADMM in both the ergodic and nonergodic senses. Here  $k$  is the iteration count (He and Yuan 2012, 2015; Monteiro and Svaiter 2013). With special structure assumptions, local and global linear convergence results for ADMM appear in Han and Yuan (2013), Yang and Han (2016), Han et al. (2018), Yuan et al. (2020), Liu et al. (2018), Boley (2013), and Deng and Yin (2016). As in Remark 3.1, our proposed rPRSM can be viewed as a version of the semiproximal strictly contractive PRSM in Gu et al. (2015). Thus, it has a worst-case  $\mathcal{O}(1/k)$  convergence rate in both the ergodic and nonergodic sense. Linear convergence rate results on PRSM and its variants but with strongly convex and polyhedral constraint set assumptions appear in Gu et al. (2015), Davis and Yin (2017), and He et al. (2018).

### 3.2. Implementation Details

The explicit  $Z$ -updates in Algorithm 1 are simple and easy. We now show that we have explicit expressions for  $R$ -updates and  $Y$ -updates as well.

**3.2.1. R-Subproblem.** In this section, we present the formula for solving the  $R$ -subproblem in Algorithm 1. We define  $\mathcal{P}_{\mathcal{R}}(W)$  to be the projection of  $W$  onto the compact set  $\mathcal{R}$ , where  $\mathcal{R} := \{R \in \mathbb{S}_+^{(n-1)^2+1} : \text{tr}(R) = n + 1\}$ . By completing the square at the current iterates  $Y^k, Z^k$ , the  $R$ -subproblem can be explicitly solved by the projection operator  $\mathcal{P}_{\mathcal{R}}$  as follows:

$$\begin{aligned} R^{k+1} &= \underset{R \in \mathcal{R}}{\operatorname{argmin}} - \left\langle Z^k, \hat{V}R\hat{V}^T \right\rangle + \frac{\beta}{2} \left\| Y^k - \hat{V}R\hat{V}^T \right\|_F^2 \\ &= \underset{R \in \mathcal{R}}{\operatorname{argmin}} \frac{\beta}{2} \left\| Y^k - \hat{V}R\hat{V}^T + \frac{1}{\beta} Z^k \right\|_F^2 \\ &= \underset{R \in \mathcal{R}}{\operatorname{argmin}} \frac{\beta}{2} \left\| R - \hat{V}^T \left( Y^k + \frac{1}{\beta} Z^k \right) \hat{V} \right\|_F^2 \\ &= \mathcal{P}_{\mathcal{R}} \left( \hat{V}^T \left( Y^k + \frac{1}{\beta} Z^k \right) \hat{V} \right), \end{aligned}$$

where the third equality follows from the assumption  $\hat{V}^T \hat{V} = I$ .

For a given symmetric matrix  $W \in \mathbb{S}^{(n-1)^2+1}$ , we now show how to perform the projection  $\mathcal{P}_{\mathcal{R}}(W)$ . Using the eigenvalue decomposition  $W = U\Lambda U^T$ , we have

$$\mathcal{P}_{\mathcal{R}}(W) = U \operatorname{Diag}(\mathcal{P}_{\Delta}(\operatorname{diag}(\Lambda))) U^T,$$

where  $\mathcal{P}_{\Delta}(\operatorname{diag}(\Lambda))$  denotes the projection of  $\operatorname{diag}(\Lambda)$  onto the simplex

$$\Delta = \left\{ \lambda \in \mathbb{R}_+^{(n-1)^2+1} : \lambda^T e = n + 1 \right\}.$$

Projections onto simplices can be performed efficiently via some standard root-finding strategies (van den Berg and Friedlander 2008/2009, Chen and Ye 2011). Therefore, the  $R$ -updates reduce to the projection of the vector of the positive eigenvalues of  $\hat{V}^T \left( Y^k + \frac{1}{\beta} Z^k \right) \hat{V}$  onto the simplex  $\Delta$ .

**3.2.2. Y-Subproblem.** In this section, we present the formula for solving the  $Y$ -subproblem in Algorithm 1. By completing the square at the current iterates  $R^{k+1}, Z^{k+\frac{1}{2}}$ , we get

$$\begin{aligned} Y^{k+1} &= \underset{Y \in \mathcal{Y}}{\operatorname{argmin}} \langle L_Q, Y \rangle + \left\langle Z^{k+\frac{1}{2}}, Y - \hat{V}R^{k+1}\hat{V}^T \right\rangle \\ &\quad + \frac{\beta}{2} \left\| Y - \hat{V}R^{k+1}\hat{V}^T \right\|_F^2 \\ &= \underset{Y \in \mathcal{Y}}{\operatorname{argmin}} \frac{\beta}{2} \left\| Y - \left( \hat{V}R^{k+1}\hat{V}^T - \frac{1}{\beta} (L_Q + Z^{k+\frac{1}{2}}) \right) \right\|_F^2. \end{aligned}$$

Recall that the  $Y$ -subproblem involves the projection onto the polyhedral set in (2.6):

$$\begin{aligned} \mathcal{Y} &:= \{Y \in \mathbb{S}^{n^2+1} : \mathcal{G}_j(Y) = u_0, 0 \leq Y \leq 1, \mathbf{b}^0 \operatorname{diag}(Y) \\ &= I, \mathbf{o}^0 \operatorname{diag}(Y) = I, \operatorname{arrow}(Y) \\ &= 0\}. \end{aligned}$$

Set  $T := \left( \hat{V}R^{k+1}\hat{V}^T - \frac{1}{\beta} (L_Q + Z^{k+\frac{1}{2}}) \right)$ . Then we update  $Y^{k+1}$  as follows:

$$(Y^{k+1})_{ij} = \begin{cases} 1 & \text{if } i = j = 0, \\ s_{ij} & \text{if } i = j > 0 \text{ or} \\ & (ij = 0 \text{ and } i + j > 0), \\ 0 & \text{if } ij \text{ or } ji \in \bar{j}/(00), \\ \min\{1, \max\{T_{ij}, 0\}\} & \text{otherwise,} \end{cases} \quad (3.7)$$

where  $s \in \mathbb{R}^{n^2}$  is determined as in (3.8).

**Remark 3.3** (Calculating  $s$  in (3.7)). Given any column vector  $t \in \mathbb{R}^{n^2}$ , we let  $t_i^c$  denote the  $i$ th column of  $\operatorname{Mat}t$ ,  $i = 1, \dots, n$ . We denote the  $i$ th subvector in the diagonal (except for the  $(0,0)$ -th element), first column and first row of  $T$  by the column vectors  $t_i^d, t_i^c$  and  $t_i^r$ , respectively. Then

$$\begin{aligned} s &= \underset{s}{\operatorname{argmin}} (\|s - t^d\|^2 + \|s - t^c\|^2 + \|s - t^r\|^2) \\ &\text{s.t. } \operatorname{Mat}(s) \in \mathcal{D}. \end{aligned} \quad (3.8)$$

By completing the squares in the objective of (3.8) and removing the redundant  $s \leq 1$ , we transform (3.8) into the following equivalent optimization problem:

$$\begin{aligned} \min_s \quad & \left\| s - \frac{1}{3}(t^d + t^c + t^r) \right\|^2 \\ \text{s.t.} \quad & \text{Mat}(s) \in \mathcal{D}. \end{aligned} \quad (3.9)$$

We reshape  $\frac{1}{3}(t^d + t^c + t^r)$  into an  $n$ -by- $n$  matrix  $\tilde{T}_a$  column by column. Then we can rewrite (3.9) equivalently as

$$\begin{aligned} \min_{S \in \mathbb{R}^{n \times n}} \quad & \|S - \tilde{T}_a\|^2 \\ \text{s.t.} \quad & S \in \mathcal{D}. \end{aligned} \quad (3.10)$$

Denote the optimal solution of (3.10) by  $S^*$ , then  $s = \text{vec}(S^*)$ . This relates with Proposition 2.4; in each iteration, we project the arrow positions of  $Y$  to the set of doubly stochastic matrices.

### 3.3. Bounding from Approximate Solutions

Primal and dual solutions from our algorithm are approximate. We would like to obtain useful lower and upper bounds for the optimal value  $p_{\text{QAP}}^*$ . This can often help in stopping the algorithm early and prove optimality for our current permutation  $X$  for the original QAP. This follows the approach in Li et al. (2021).

**3.3.1. Lower Bound from Relaxation.** Exact solutions of Relaxation (2.7) provide lower bounds to the original QAP (1.1). However, the size of Problem (2.7) can be extremely large, and it could be very expensive to obtain solutions of high accuracy. In this section, we present an inexpensive way to obtain a valid lower bound using the output with moderate accuracy from our algorithm.

Our approach is based on the following functional:

$$g(Z) := \min_{Y \in \mathcal{Y}} \langle L_Q + Z, Y \rangle - (n+1)\lambda_{\max}(\hat{V}^T Z \hat{V}), \quad (3.11)$$

where  $\lambda_{\max}(\hat{V}^T Z \hat{V})$  denotes the largest eigenvalue of  $\hat{V}^T Z \hat{V}$ . In Theorem 3.2, we show that  $\max_Z g(Z)$  is indeed the Lagrange dual problem of our main DNN relaxation (2.7).

**Theorem 3.2.** *Let  $g$  be the functional defined in (3.11). Then the problem*

$$d_Z^* := \max_Z g(Z) \quad (3.12)$$

*is a concave maximization problem. Furthermore, strong duality holds for the primal (2.7) with dual (3.12), that is,*

$$p_{\text{DNN}}^* = d_Z^*, \text{ and } d_Z^* \text{ is attained.}$$

**Proof.** The function  $\hat{V}^T Z \hat{V}$  is linear in  $Z$ . Therefore, the largest eigenvalue function  $\lambda_{\max}(\hat{V}^T Z \hat{V})$  is a convex function of  $Z$ . Thus, the argument of the

minimum in (3.12)

$$\langle L_Q + Z, Y \rangle - (n+1)\lambda_{\max}(\hat{V}^T Z \hat{V})$$

is concave in  $Z$ . The concavity of  $g$  is now clear.

We derive (3.12) via the Lagrange dual problem of (2.7):

$$\begin{aligned} p_{\text{DNN}}^* &= \min_{R \in \mathcal{R}, Y \in \mathcal{Y}} \max_Z \left\{ \langle L_Q, Y \rangle + \langle Z, Y - \hat{V}R\hat{V}^T \rangle \right\} \\ &= \max_Z \min_{R \in \mathcal{R}, Y \in \mathcal{Y}} \left\{ \langle L_Q, Y \rangle + \langle Z, Y - \hat{V}R\hat{V}^T \rangle \right\}, \end{aligned} \quad (3.13a)$$

$$\begin{aligned} &= \max_Z \left\{ \min_{Y \in \mathcal{Y}} \{ \langle L_Q, Y \rangle + \langle Z, Y \rangle \} + \min_{R \in \mathcal{R}} \langle Z, -\hat{V}R\hat{V}^T \rangle \right\} \\ &= \max_Z \left\{ \min_{Y \in \mathcal{Y}} \{ \langle L_Q, Y \rangle + \langle Z, Y \rangle \} + \min_{R \in \mathcal{R}} \langle \hat{V}^T Z \hat{V}, -R \rangle \right\} \\ &= \max_Z \left\{ \min_{Y \in \mathcal{Y}} \langle L_Q + Z, Y \rangle - (n+1)\lambda_{\max}(\hat{V}^T Z \hat{V}) \right\} \\ &= d_Z^*, \end{aligned} \quad (3.13b)$$

where

1. (3.13a) follows from Rockafellar (1997, corollary 28.2.2 and theorem 28.4) and the fact that (2.7) has generalized Slater points (Zhao et al. 1998)<sup>3</sup>;

2. (3.13b) follows from the definition of  $\mathcal{R}$  and the Rayleigh Principle.

We see from Rockafellar (1997, corollary 28.2.2 and corollary 28.4.1) that the dual optimal value  $d_Z^*$  is attained.  $\square$

**Remark 3.4.** Because the Lagrange dual problem in Theorem 3.2 is an unconstrained maximization problem, evaluating  $g$  defined in (3.11) at the  $k$ th iterate  $Z^k$  yields a valid lower bound for  $p_{\text{DNN}}^*$ , that is,  $g(Z^k) \leq p_{\text{DNN}}^* \leq p_{\text{QAP}}^*$ . The functional  $g$  also strengthens the bound given in Oliveira et al. (2018, lemma 3.2). We also see in (3.13b) that  $Z \prec 0$  provides a positive contribution to the eigenvalue part of the lower bound. Moreover, Theorem 2.1 implies that the contribution from  $J_R$  position, the diagonal, first row, and column of  $L_Q + Z$  (except for the (0, 0)th element) is zero. This motivates scaling  $L_Q$  to be positive definite.

Let  $P_V := \hat{V} \hat{V}^T$ . Then for any  $r, s \in \mathbb{R}$ , the objective in (2.7) can be replaced by

$$\langle r(P_V L_Q P_V + sI), Y \rangle. \quad (3.14)$$

We obtain the same solution pair  $(R^*, Y^*)$  of (2.7). Another advantage is that it potentially forces the dual multiplier  $Z^*$  to be negative definite, and thus the lower bound is larger. Additional strategies can be used to strengthen the lower bound  $g(Z^k)$ . Suppose that the given data matrices  $A, B$  are symmetric and integral, then from (1.1), we know that  $p_{\text{QAP}}^*$  is an even integer.

Therefore, applying the ceiling operator to  $g(Z^k)$  still gives a valid lower bound to  $p_{\text{QAP}}^*$ . According to this prior information, we can strengthen the lower bound with the even number in the pair  $\{\lceil g(Z^k) \rceil, \lceil g(Z^k) \rceil + 1\}$ .

### 3.3.2. Upper Bound from Nearest Permutation Matrix.

Oliveira et al. (2018) present two methods for obtaining upper bounds using a nearest permutation matrix. In this section, we present a new strategy using a nearest permutation matrix.

Given  $\bar{X} \in \mathbb{R}^{n \times n}$ , the nearest permutation matrix  $X^*$  from  $\bar{X}$  is found by solving

$$X^* = \operatorname{argmin}_{X \in \Pi} \frac{1}{2} \|X - \bar{X}\|_F^2 = \operatorname{argmin}_{X \in \Pi} -\langle \bar{X}, X \rangle. \quad (3.15)$$

Any solution to Problem (3.15) yields a feasible solution to the original QAP, which gives a valid upper bound  $\operatorname{tr}(AX^*B(X^*)^T)$ . As discussed previously, the permutation matrices are the extreme points of the set of doubly stochastic matrices  $\mathcal{D}$ . Hence, we reformulate Problem (3.15) as the linear program

$$\max_{x \in \mathbb{R}^{n^2}} \{ \langle \operatorname{vec}(\bar{X}), x \rangle : (I_n \otimes e^T)x = e, (e^T \otimes I_n)x = e, x \geq 0 \}, \quad (3.16)$$

and we solve (3.16) using a simplex method type algorithm.

For an approximate optimum  $Y^{\text{out}}$ , The first approach in Oliveira et al. (2018) sets  $\operatorname{vec}(\bar{X})$  to be the first column of  $Y^{\text{out}}$  ignoring the first element and then solves (3.16). Now let  $Y^{\text{out}} = \sum_{i=1}^r \lambda_i v_i v_i^T$  be the spectral decomposition, with  $\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_r > 0$ . By abuse of notation, we set  $v_i$  to be the vectors in  $\mathbb{R}^{n^2}$  formed by removing the first element from  $v_i$ . The second approach presented in Oliveira et al. (2018) is to use  $\operatorname{vec}(\bar{X}) = \lambda_1 v_1$  in solving (3.16), where  $(\lambda_1, v_1)$  is the most dominant eigenpair of  $Y^{\text{out}}$ .

Inspired by the approximation algorithm in Goemans and Williamson (1995), now let  $\xi$  be a random vector in  $\mathbb{R}^r$  with elements in  $(0, 1)$ , and in decreasing order. We use  $\xi$  to perturb the eigenvalues  $\lambda_1, \dots, \lambda_r$  and form  $\bar{X}$  for the upper bound problem (3.16) so that

$$\operatorname{vec}(\bar{X}) = \sum_{i=1}^r \xi_i \lambda_i v_i.$$

We repeat this  $\max\{1, \min(3 * \lceil \log(n) \rceil, \text{ubest} - \text{lbest})\}$  number of times, where  $\text{ubest}$  and  $\text{lbest}$  refer to the best upper and lower bounds achieved during the algorithmic routine, respectively. We update the current upper bound  $\text{ubest}$  if a smaller upper bound is obtained.

## 4. Numerical Experiments with rPRSM

We now present numerical results for Algorithm 1 (Graham et al. 2021, rPRSM), with the bounding

strategies discussed in Section 3.3. The parameter settings and stopping criteria are in Section 4.1. We use symmetric<sup>4</sup> data to examine the comparative performance between rPRSM and Oliveira et al. (2018, ADMM). We aim to show that our proposed rPRSM shows improvements on convergence rates and relative gaps. In Section 4.3, we compare rPRSM with the three recently proposed relaxation methods (Bravo Ferreira et al. 2018, C-SDP; Date and Nagi 2019, F2-RLT2-DA; Yang et al. 2015, SDPNAL+).

### 4.1. Parameter Settings and Stopping Criteria

1. We scale the data  $L_Q$  from (3.14) as follows:

$$\begin{aligned} L_1 &\leftarrow P_V L_Q P_V, \\ L_2 &\leftarrow L_1 + \sigma_L I, \quad \text{where } \sigma_L := \max\{0, -\lfloor \lambda_{\min}(L_Q) \rfloor\} + 10n, \\ L_3 &\leftarrow \frac{n^2}{\alpha} L_2, \quad \text{where } \alpha := \lceil \|L_2\|_F \rceil. \end{aligned}$$

We set the penalty parameter  $\beta = \frac{n}{3}$  and the under-relaxation parameter  $\gamma = 0.9$  for the dual variable update. We choose the initial iterates<sup>5</sup>:

$$Y^0 = \frac{1}{n!} \sum_{X \in \Pi} (1; \operatorname{vec}(X))(1; \operatorname{vec}(X))^T \quad \text{and} \quad Z^0 = \mathcal{P}_{\mathcal{Z}_A}(0).$$

We compute the lower and upper bounds every 100 iterations. The tolerance for the projection onto the set of doubly stochastic matrices in Remark 3.3 is set to be  $10^{-4}$ .

2. We terminate rPRSM when one of the following conditions is satisfied.

- (a) The maximum number of iterations,  $\text{maxiter} = 40,000$  is reached.
- (b) For given tolerance  $\epsilon$ , the following bound on the primal and dual residuals holds for  $m_t$  sequential times:

$$\max \left\{ \frac{\|Y^k - \hat{V} R^k \hat{V}^T\|_F}{\|Y^k\|_F}, \beta \|Y^k - Y^{k-1}\|_F \right\} < \epsilon.$$

We set  $\epsilon = 10^{-4}$  and  $m_t = 100$ .

- (c) Let  $\{l_1, \dots, l_k\}$  and  $\{u_1, \dots, u_k\}$  be sequences of lower and upper bounds from Sections 3.3.1 and 3.3.2, respectively. The lower (respectively, upper) bounds do not change for  $m_l$  (respectively,  $m_u$ ) sequential times. We set  $m_l = m_u = 100$ .

- (d) The KKT conditions given in (2.12) are satisfied to a certain precision. More specifically, for a predefined tolerance  $\delta > 0$ , it holds that

$$\max \left\{ \left\| R^k - \mathcal{P}_{\mathcal{R}} \left( R^k + \hat{V}^T Z^k \hat{V} \right) \right\|_F, \|Y^k - \mathcal{P}_{\mathcal{Y}}(Y^k - L_Q - Z^k)\|_F, \|Y^k - \hat{V} R^k \hat{V}^T\|_F \right\} < \delta.$$

We use this stopping criterion for instances with  $n$  larger than 20, and we set the tolerance  $\delta = 10^{-5}$  when it is used.

### 4.2. Empirical Results

We now compare results from rPRSM and Oliveira et al. (2018, ADMM) on instances from QAPLIB. We divide the instances into three groups based on sizes:

$$n \in \{10, \dots, 20\}, \{21, \dots, 40\}, \{41, \dots, 64\}.$$

For ADMM, we use the parameters from Oliveira et al. (2018), that is,  $\beta = n/3, \gamma = 1.618$ , and we adopt the same

stopping criteria for both ADMM and rPRSM. All instances in Tables 1–3 are tested using MATLAB version 2021a on Dell XPS 8940 with 11th Gen Intel(R) Core(TM) i5-11400 @ 2.60GHz 2.59GHz, and 32 GB memory.

The following provides extra details for the headers in the various tables.

1. true-opt: global optimal value; marked with \* when unknown.
2. lbd: lower bound from rPRSM;
3. ubd: upper bound from rPRSM;

**Table 1.** Lower and Upper Bounds, Relative Gaps, Iteration Numbers, Running Time Obtained from rPRSM and Oliveira et al. (2018, ADMM) on the QAPLIB Instances of Small Size

Problem data			Numerical results					Timing			
No.	name	true-opt	lbd	ubd	rel.gap	rel.opt.gap	rel.gap <sup>A</sup>	iter	iter <sup>A</sup>	time	time <sup>A</sup>
1	chr12a	9552	9548	9552	0.04	0.04	0.02	11500	24800	18.47	33.31
2	chr12b	9742	9742	9742	0	0	0.08	10300	26700	18.28	35.02
3	chr12c	11156	11156	11156	0	0	0	1600	19400	2.88	25.95
4	chr15a	9896	9896	9896	0	0	0.28	6700	30900	21.70	92.49
5	chr15b	7990	7990	7990	0	0	0.03	3500	20300	11.37	60.45
6	chr15c	9504	9504	9504	0	0	0.08	1800	20000	6.40	59.06
7	chr18a	11098	11098	11098	0	0	0	2000	20600	15.53	143.21
8	chr18b	1534	1534	1846	18.46	0	59.83	5558	12600	48.56	102.17
9	chr20a	2192	2192	2192	0	0	0.18	3700	33700	50.76	447.00
10	chr20b	2298	2298	2298	0	0	0	1200	26200	18.23	377.53
11	chr20c	14142	14136	14142	0.04	0.04	0.15	30900	33700	416.98	445.15
12	els19	17212548	17208748	17212548	0.02	0.02	0.35	30800	40000	341.05	424.29
13	esc16a	68	64	68	6.02	6.02	47.34	398	597	1.97	2.74
14	esc16b	292	290	294	1.37	0.69	6.66	399	399	2.07	1.78
15	esc16c	160	154	176	13.29	3.81	31.61	386	896	2.01	4.07
16	esc16d	16	14	16	12.90	12.90	87.50	282	659	1.45	3.05
17	esc16e	28	28	32	13.11	0	65.85	299	556	1.54	2.57
18	esc16f	0	0	0	0	0	0	1	1	0.01	0.01
19	esc16g	26	26	40	41.79	0	78.57	300	695	1.51	3.05
20	esc16h	996	978	1054	7.48	1.82	31.76	1362	609	6.73	2.71
21	esc16i	14	12	14	14.81	14.81	88.89	1016	2044	4.75	9.05
22	esc16j	8	8	8	0	0	82.76	200	787	1.01	3.49
23	had12	1652	1652	1652	0	0	0	300	11600	0.58	16.05
24	had14	2724	2724	2724	0	0	0	400	20300	1.27	50.57
25	had16	3720	3720	3720	0	0	0	600	18100	3.16	74.82
26	had18	5358	5358	5358	0	0	0.02	1300	34700	11.04	273.82
27	had20	6922	6922	6922	0	0	0.13	2300	40000	34.57	571.06
28	nug12	578	568	728	24.67	1.74	27.86	1416	2884	2.91	4.21
29	nug14	1014	1012	1022	0.98	0.20	1.08	2832	19600	9.19	50.62
30	nug15	1150	1142	1280	11.39	0.70	16.33	2161	5812	8.83	19.83
31	nug16a	1610	1600	1610	0.62	0.62	0.62	6217	19300	33.68	90.71
32	nug16b	1240	1220	1250	2.43	1.63	25.41	3454	2347	17.86	10.70
33	nug17	1732	1708	1756	2.77	1.39	2.77	6194	6401	43.55	40.80
34	nug18	1930	1894	2022	6.54	1.88	12.84	9555	3988	83.53	32.61
35	nug20	2570	2508	2702	7.45	2.44	16.90	7065	2386	109.85	35.72
36	rou12	235528	235528	235528	0	0	0	3700	34200	7.05	51.43
37	rou15	354210	350216	360702	2.95	1.13	4.89	2531	3946	10.61	14.38
38	rou20	725522	695180	781532	11.70	4.27	14.93	7024	1538	109.60	22.93
39	scr12	31410	31410	31410	0	0	19.38	400	4268	0.81	5.89
40	scr15	51140	51140	51140	0	0	2.67	700	5489	3.00	18.05
41	scr20	110030	106804	132826	21.72	2.98	33.40	11599	9705	173.01	136.32
42	tai10a	135028	135028	135028	0	0	0.01	1200	21400	1.48	15.90
43	tai12a	224416	224416	224416	0	0	0	300	4300	0.48	5.63
44	tai15a	388214	377100	403890	6.86	2.90	9.03	2644	2245	11.24	8.13
45	tai17a	491812	476526	534328	11.44	3.16	16.25	2940	1399	21.47	9.40
46	tai20a	703482	671676	762166	12.62	4.63	19.03	3733	999	58.64	15.10

Downloaded from informs.org by [129.97.202.75] on 25 August 2023, at 11:39 . For personal use only, all rights reserved.

**Table 2.** Lower and Upper Bounds, Relative Gaps, Iteration Numbers, Running Time Obtained from rPRSM and Oliveira et al. (2018, ADMM) on the QAPLIB Instances of Medium Size

Problem data			Numerical results					Timing			
No.	name	true-opt	lbd	ubd	rel.gap	rel.opt.gap	rel.gap <sup>A</sup>	iter	iter <sup>A</sup>	time	time <sup>A</sup>
47	chr22a	6156	6156	6156	<b>0</b>	0	0.02	11500	40000	257.32	869.69
48	chr22b	6194	6190	6194	<b>0.06</b>	0.06	0.11	13500	39300	333.04	922.56
49	chr25a	3796	3796	3796	<b>0</b>	0	0	6200	35600	251.42	1350.42
50	esc32a	130	104	158	<b>41.06</b>	22.13	106.07	17700	18200	2607.52	2493.35
51	esc32b	168	132	216	<b>48.14</b>	23.92	96.69	1000	4000	150.17	551.33
52	esc32c	642	616	644	<b>4.44</b>	4.13	27.43	2500	1700	377.82	238.25
53	esc32d	200	192	220	<b>13.56</b>	4.07	54.37	670	1400	99.38	194.64
54	esc32e	2	2	24	162.96	0	141.18	700	3000	104.63	420.31
55	esc32g	6	6	22	110.34	0	26.67	500	900	75.42	126.24
56	esc32h	438	426	452	<b>5.92</b>	2.77	33.46	6500	11300	975.33	1568.05
57	kra30a	88900	86838	95760	<b>9.77</b>	2.35	16.50	9898	3700	1041.70	390.89
58	kra30b	91420	87858	101640	<b>14.55</b>	3.97	27.87	5480	4900	575.28	501.67
59	kra32	88700	85776	94350	<b>9.52</b>	3.35	35.29	4959	4100	738.47	576.96
60	nug21	2438	2382	2644	<b>10.42</b>	2.32	12.36	6439	5600	129.38	106.93
61	nug22	3596	3530	3678	<b>4.11</b>	1.85	12.76	7279	7400	182.73	176.07
62	nug24	3488	3402	3770	<b>10.26</b>	2.50	16.25	4543	4300	167.09	149.64
63	nug25	3744	3626	3984	<b>9.41</b>	3.20	15.37	10400	7500	518.30	309.37
64	nug27	5234	5130	5496	<b>6.89</b>	2.01	17.08	10039	8400	699.69	507.98
65	nug28	5166	5026	5644	<b>11.58</b>	2.75	18.55	8387	7200	687.17	521.18
66	nug30	6124	5950	6610	<b>10.51</b>	2.88	19.83	11321	8800	1190.14	903.52
67	ste36a	9526	9260	9980	<b>7.48</b>	2.83	42.28	19500	27300	5473.12	7479.83
68	ste36b	15852	15668	15932	<b>1.67</b>	1.17	82.03	29000	40000	7936.73	10967.49
69	ste36c	8239110	8134808	8394142	<b>3.14</b>	1.27	36.15	36499	40000	9880.20	11183.03
70	tai25a	1167256	1096656	1264590	<b>14.22</b>	6.24	20.56	2264	800	101.26	34.42
71	tai30a	1818146	1706872	1984536	<b>15.04</b>	6.31	15.21	4550	1400	474.63	142.48
72	tai35a*	2422002	2216646	2605986	<b>16.15</b>	8.85	22.34	3161	1500	732.88	358.34
73	tai40a*	3139370	2843310	3455540	<b>19.44</b>	9.90	23.43	5577	2200	2631.93	1053.03
74	tho30	149936	143576	166336	<b>14.69</b>	4.33	24.33	8321	7400	895.77	773.30
75	tho40*	240516	226522	256442	<b>12.39</b>	5.99	26.25	15535	12200	7281.85	5700.18

4. rel.gap: relative gap from rPRSM:

relative gap

$$:= 2 \frac{\text{best feasible upper bound} - \text{best lower bound}}{\text{best feasible upper bound} + \text{best lower bound} + 1}; \quad (4.1)$$

5. rel.opt.gap: relative optimality gap from rPRSM using the known true optimal value and the lower bound;

6. rel.gap<sup>A</sup>: relative gap from (Oliveira et al. 2018, ADMM) with tolerance  $\epsilon = 10^{-5}$ ;

7. iter: number of iterations by rPRSM with tolerance  $\epsilon = 10^{-5}$ ;

8. iter<sup>A</sup>: number of iterations from (Oliveira et al. 2018, ADMM) with tolerance  $\epsilon = 10^{-5}$ ;

9. time: solver rPRSM time;

10. time<sup>A</sup>: solver (Oliveira et al. 2018, ADMM) time.

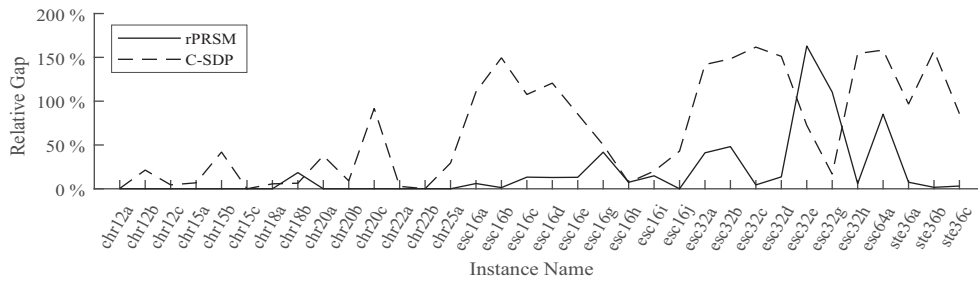
**4.2.1. Small Size.** Comparing columns iter and iter<sup>A</sup> in Table 1, we see that 37 instances were treated with fewer iterations using rPRSM; that is, rPRSM converges

**Table 3.** Lower and Upper Bounds, Relative Gaps, Iteration Numbers, Running Time Obtained from rPRSM and Oliveira et al. (2018, ADMM) on the QAPLIB Instances of Large Size

Problem data			Numerical results					Timing			
No.	name	true-opt	lbd	ubd	rel.gap	rel.opt.gap	rel.gap <sup>A</sup>	iter	iter <sup>A</sup>	time	time <sup>A</sup>
76	esc64a	116	98	244	85.13	16.74	75.71	400	1200	3049.78	9161.95
77	sko42*	15812	15336	16244	<b>5.75</b>	3.06	17.24	5511	10700	3083.39	6086.78
78	sko49*	23386	22654	24162	<b>6.44</b>	3.18	16.59	9484	16900	13453.32	24638.60
79	sko56*	34458	33390	36468	<b>8.81</b>	3.15	16.60	5792	15100	18663.36	48683.65
80	sko64*	48498	47022	50322	<b>6.78</b>	3.09	15.54	10021	21100	73824.91	152841.35
81	tai50a*	4938796	4390980	5517228	<b>22.73</b>	11.74	25.79	2331	3300	3792.80	5345.89
82	tai60a	7205962	6326344	7895180	<b>22.06</b>	13.00	26.03	3799	5100	18807.38	25109.36
83	tai64c*	1855928	1811354	1887500	<b>4.12</b>	2.43	38.79	800	2400	6139.92	18157.72
84	wil50	48816	48126	50834	<b>5.47</b>	1.42	9.38	5384	11000	9165.35	18236.57

Downloaded from informs.org by [129.97.202.75] on 25 August 2023, at 11:39 . For personal use only, all rights reserved.



**Figure 1.** Relative Gap for rPRSM and C-SDP

faster in general than ADMM for the small-size QAPLIB instances. In particular, 45 of 46 instances are solved with relative gaps just as good as the ones obtained by ADMM, and these instances are marked with bold in Table 1. We have found provably optimal solutions for instances

*chr12b* *chr12c* *chr15a* *chr15b* *chr15c* *chr18a*  
*chr20a* *chr20b* *esc16f* *esc16j* *had12* *had14*  
*had16* *had18* *had20* *rou12* *scr12* *scr15*  
*tai10a* *tai12a*.

We also observe from columns *iter* and *iter<sup>A</sup>* in Table 1 that rPRSM gives reduction in number of iterations in many instances; 37 of 46 instances use fewer or equal number of iterations using rPRSM compared with ADMM. For rPRSM alone, we observe that most of the instances show good bounds with a reasonable amount of time. Most of the instances are solved within a minute using the machine described previously.

**4.2.2. Medium Size.** Table 2 contains results on 29 QAPLIB instances with sizes  $n \in \{22, \dots, 40\}$ . Columns *rel.gap* and *rel.gap<sup>A</sup>* in Table 2 show that rPRSM produces competitive relative gaps compared with ADMM. In particular, 27 instances are solved with relative gaps just as good as the ones obtained by ADMM, and these instances are marked with bold in Table 2. We have found provably optimal solutions for instances *chr22a* and *chr25a*. For rPRSM alone, we observe that most of the instances show good bounds with reasonable amount of time.

**4.2.3. Large Size.** Table 3 contains results on nine QAPLIB instances with sizes  $n \in \{41, \dots, 64\}$ . We observe that rPRSM outputs better relative gaps than ADMM on eight instances, and this is because of the random perturbation approach presented in Section 3.3.2. We also obtain reduction on the number of iterations. It indicates that our strategies taken on *R* and *Z* updates in rPRSM help the iterates converge faster than ADMM.

### 4.3. Comparisons with Other Methods

We now compare our results with three recent papers on relaxations for QAP.<sup>6</sup>

**4.3.1. Comparison with C-SDP.** Briefly, C-SDP (Bravo Ferreira et al. 2018) is a semidefinite relaxation-based algorithm that applies to relatively sparse data, and in particular, are presented for the *chr* and *esc* families in QAPLIB. Figure 1 illustrates the relative gaps arising from rPRSM and C-SDP. The numerics used in Figure 1 can be found in Bravo Ferreira et al. (2018, tables 3 and 4).

The horizontal axis indicates the instance name on QAPLIB, whereas the vertical axis indicates the relative gap.<sup>7</sup> Figure 1 illustrates that rPRSM yields much stronger relative gaps than C-SDP.

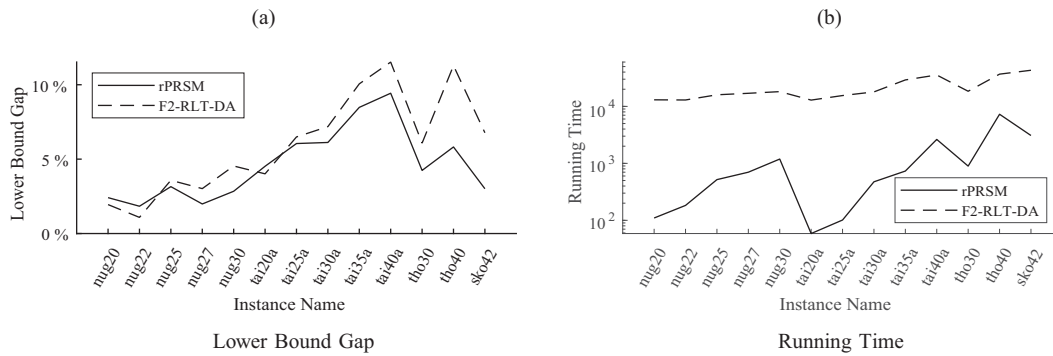
**4.3.2. Comparison with F2-RLT2-DA.** Date and Nagi (2019) propose F2-RLT2-DA, a linearization technique-based parallel algorithm (GPU-based) for obtaining lower bounds via Lagrangian relaxation. Figure 2(a) illustrates the comparisons on lower bound gap<sup>8</sup> using rPRSM and F2-RLT2-DA. It shows that both rPRSM and F2-RLT2-DA output competitive lower bounds to the best-known feasible values for QAP. Figure 2(b) illustrates the comparisons on the running time<sup>9</sup> in seconds using rPRSM and F2-RLT2-DA. We observe that the running time of F2-RLT2-DA is much longer than the running time of rPRSM; F2-RLT2-DA requires at least 10 times longer than rPRSM. Furthermore, from Figure 2 we observe that, although the two methods give similar lower bounds to QAP, rPRSM is less time-consuming even considering the differences in the hardware.<sup>10</sup>

**4.3.3. Comparison with SDPNAL+.** SDPNAL+ (Yang et al. 2015) is one of the state-of-the-art software packages for large-scale SDPs with bound constraints. As suggested by the user guide, an SDP relaxation for QAP can be formulated as

$$\min_Y \{ \langle C, Y \rangle : \mathcal{A}(Y) = b, Y \succeq 0, Y \preceq 0 \}, \quad (4.2)$$

where the affine constraints  $\mathcal{A}(Y) = b$  and the lifted variable *Y* differ from ours and are given in Povh and Rendl (2009).

Figure 2. Numerical Comparison for rPRSM and F2-RLT2-DA



We now develop a process that takes the output from SDPNAL+ to obtain valid lower bounds. Define  $\mathcal{Y} := \{Y \succeq 0 : \text{tr}Y = n\}$ . Let

$$\hat{g}(y, W, Z) := \min_{Y \in \mathcal{Y}} \{ \langle C, Y \rangle + \langle y, b - \mathcal{A}(Y) \rangle - \langle Z, Y \rangle - \langle W, Y \rangle \}$$

be the dual functional for (4.2). The dual of (4.2) is

$$d^* = \max_{y, W \succeq 0, Z \succeq 0} \hat{g}(y, W, Z).$$

Let  $(\bar{y}, \bar{W}, \bar{Z})$  be the dual solution that we obtain at the termination of SDPNAL+. Define  $\bar{S} = C - \mathcal{A}^*(\bar{y}) - \bar{Z} - \bar{W}$ . We compute the valid lower bound as follows:

$$d^* \geq \langle b, \bar{y} \rangle + \min_{Y \in \mathcal{Y}} \{ \langle \bar{S}, Y \rangle \} = \langle b, \bar{y} \rangle + n \lambda_{\min}(\bar{S}).$$

Additionally, as noted in Remark 3.4, we take the lower bound with the even number in the pair  $\{ \lceil \hat{g}(\bar{y}, \bar{W}, \bar{Z}) \rceil, \lceil \hat{g}(\bar{y}, \bar{W}, \bar{Z}) \rceil + 1 \}$ . Table 4 contains the numerical results. The header rel.opt.gap (rel.opt.gap<sup>N</sup>, respectively) refers to the value rel.opt.gap used in Tables 1–3 for rPRSM (SDPNAL+, respectively). The headers time and time<sup>N</sup> refer to the running time for the two algorithms. The header ubdtime refers to the time that rPRSM consumes in computing the upper bounds using the strategy in Section 3.3.2. In Table 4, we present comparative performance that have different relative gaps among the 84 instances from Tables 1–3.

The running time for rPRSM appears different from Tables 1–3 because we measure the running time based on the wall-clock time. We observe that the lower bound and the running time from the two different methods are competitive. SDPNAL+ is a MATLAB software package, but some subroutines are implemented in C language via Mex files to improve its efficiency, whereas rPRSM is implemented using MATLAB only. We emphasize that the running time for rPRSM includes strategies for computing upper and lower bounds from Section 3.3. This means that

we sometimes obtain a zero-duality gap and a proof of optimality for the original QAP.

## 5. Conclusion

In this paper we introduce a strengthened splitting method for solving the facially reduced DNN

Table 4. Relative Optimality Gaps, Running Time of rPRSM vs. SDPNAL+ on 84 Instances from QAPLIB

Problem data		Numerical results		Timing		
No.	name	rel.opt.gap	rel.opt.gap <sup>N</sup>	time	ubdtime	time <sup>N</sup>
1	chr12a	0.042	0	20.78	4.55	4.69
11	chr20c	0.042	0.028	423.35	20.22	84.11
12	els19	0.022	0.028	351.37	21.31	53.89
29	nug14	0.197	0.395	9.33	1.69	19.86
35	nug20	2.441	2.521	105.52	4.89	81.09
37	rou15	1.134	1.145	10.92	1.52	14.24
38	rou20	4.271	4.279	106.77	5.03	31.72
40	scr15	0	0.004	3.35	0.46	4.37
41	scr20	2.976	3.002	171.94	8.23	82.14
44	tai15a	2.904	2.910	11.31	1.58	14.63
45	tai17a	3.157	3.163	21.53	1.89	23.58
46	tai20a	4.626	4.632	56.66	2.63	37.94
48	chr22b	0.065	0.032	301.09	7.22	167.14
57	kra30a	2.347	2.381	978.50	13.17	380.87
58	kra30b	3.974	4.001	542.78	7.27	326.17
59	kra32	3.352	3.373	678.02	7.75	716.33
61	nug22	1.852	1.909	165.43	6.36	134.08
67	ste36a	2.832	2.875	5046.15	43.98	1312.15
68	ste36b	1.167	1.308	7522.91	65.56	2119.24
69	ste36c	1.274	1.325	9594.39	83.49	2025.44
70	tai25a	6.237	6.243	95.80	2.32	71.38
71	tai30a	6.313	6.319	448.89	6.09	277.67
72	tai35a*	8.854	8.858	698.77	6.43	840.71
73	tai40a*	9.897	9.901	2532.16	18.95	1541.78
74	tho30	4.334	4.353	840.09	11.06	371.06
75	tho40*	5.993	6.008	6764.66	52.79	2298.18
77	sko42*	3.056	3.069	3213.49	23.33	2304.08
78	sko49*	3.180	3.189	13921.23	89.83	5881.42
79	sko56*	3.148	3.208	18727.49	129.02	10055.00
80	sko64*	3.090	3.312	74783.02	533.01	10119.32
81	tai50a*	11.743	11.748	3902.22	25.75	6006.06
82	tai60a*	13.000	13.058	19124.50	131.87	10275.26
83	tai64c	2.431	2.462	6152.15	37.03	6826.23
84	wil50*	1.424	1.432	9093.90	57.92	9342.94

relaxation for the QAP. That is, given constraints that are difficult to engage simultaneously, we distribute the constraints into two simpler subproblems to solve them efficiently. In addition, we provide a straightforward derivation of facial reduction and the gangster constraints via a direct lifting. In our strengthened model and algorithm, we also incorporate redundant constraints to the model that are not redundant in the subproblems arising from the splitting; more specifically, the trace constraint in the  $R$ -subproblem and the projection onto the set of doubly stochastic matrices in the  $Y$ -subproblem. We also exploit the set of dual optimal multipliers and provide customized dual updates in the algorithm, which leads a new strategy for strengthening the lower bounds.

## Endnotes

- <sup>1</sup> There are several ways of constructing such a matrix  $\hat{V}$ . One way is presented in Proposition 2.2.
- <sup>2</sup> See <http://coral.ise.lehigh.edu/data-sets/qaplib/qaplib-problem-instances-and-solutions/> (accessed March 5, 2022).
- <sup>3</sup> The Lagrangian is linear in  $R$ ,  $Y$  and linear in  $Z$ . Moreover, both constraint sets  $\mathcal{R}, \mathcal{Y}$  are convex and compact. Therefore, the result also follows from the classical Von Neumann-Fan minmax theorem.
- <sup>4</sup> We exclude instances that have asymmetric data matrices.
- <sup>5</sup> The formula for  $Y^0$  is introduced in Zhao et al. (1998, theorem 3.1).
- <sup>6</sup> For more comparisons, see Oliveira et al. (2018, tables 4.1 and 4.2) that includes a complete list of lower bounds using the bundle method in Rendl and Sotirov (2007).
- <sup>7</sup> We selected the best result given in Bravo Ferreira et al. (2018, tables 3 and 4) for different parameters. We point out that Bravo Ferreira et al. (2018) used a different formula for the gap computation. In this paper, we recomputed the relative gaps using (4.1) for a proper comparison. Bravo Ferreira et al. (2018) used a similar approach for upper bounds as in our paper, that is, the projection onto permutation matrices using von Neumann (1953) and Birkoff (1946).
- <sup>8</sup> We compute the lower bound gap by  $100 * (p^* - l) / p^* \%$ , where  $p^*$  is the best-known feasible value to QAP and  $l$  is the lower bound.
- <sup>9</sup> The running time for F2-RLT2-DA is obtained using the average time per iteration presented in Date and Nagi (2019) multiplied by 2,000 because F2-RLT2-DA runs the algorithm for 2,000 iterations. The running time for rPRSM is drawn from Tables 1–3.
- <sup>10</sup> F2-RLT2-DA was coded in C++ and CUDA C programming languages and deployed on the Blue Waters Supercomputing facility at the University of Illinois at Urbana-Champaign. Each processing element consists of an AMD Interlagos model 6276 CPU with eight cores, 2.3-GHz clock speed, and 32-GB memory connected to an NVIDIA GK110 “Kepler” K20X GPU with 2,688 processor cores and 6-GB memory.

## References

- Anjos A, Lasserre J, eds. (2011) *Handbook on Semidefinite, Conic and Polynomial Optimization*. International Series in Operations Research & Management Science (Springer-Verlag, Berlin).
- Anstreicher K, Brixius N (2000) Solving quadratic assignment problems using convex quadratic programming relaxations. Technical report, University of Iowa, Iowa City.
- Barvinok A, Stephen T (2003) The distribution of values in the quadratic assignment problem. *Math. Oper. Res.* 28(1):64–91.
- Bashiri M, Karimi H (2012) Effective heuristics and meta-heuristics for the quadratic assignment problem with tuned parameters and analytical comparisons. *J. Industrial Engng. Internat.* 8(1):6.
- Bhati R, Rasool A (2014) Quadratic assignment problem and its relevance to the real world: A survey. *Internat. J. Comput. Appl.* 96(9): 42–47.
- Birkoff G (1946) Tres observaciones sobre el algebra lineal. *Univ. Nac. Tucuman Rev. Ser. A* 5:147–151.
- Boley D (2013) Local linear convergence of the alternating direction method of multipliers on quadratic or linear programs. *SIAM J. Optim.* 23(4):2183–2207.
- Bravo Ferreira J, Khoo Y, Singer A (2018) Semidefinite programming approach for the quadratic assignment problem with a sparse graph. *Comput. Optim. Appl.* 69(3):677–712.
- Burkard R, Karisch S, Rendl F (1991) QAPLIB: A quadratic assignment problem library. *Eur. J. Oper. Res.* 55:115–119.
- Çela E (1998) *The Quadratic Assignment Problem*, vol. 1 of Combinatorial Optimization (Kluwer Academic Publishers, Dordrecht, Germany).
- Chen Y, Ye X (2011) Projection onto a simplex. Preprint, submitted February 10, <https://arxiv.org/abs/1101.6081>.
- Commander C (2003) A survey of the quadratic assignment problem, with applications. PhD thesis, University of Florida, Gainesville.
- Date K, Nagi R (2019) Level 2 reformulation linearization technique-based parallel algorithms for solving large quadratic assignment problems on graphics processing unit clusters. *INFORMS J. Comput.* 31(4):771–789.
- Davis D, Yin W (2017) Faster convergence rates of relaxed Peaceman-Rachford and ADMM under regularity assumptions. *Math. Oper. Res.* 42(3):783–805.
- Deng W, Yin W (2016) On the global and linear convergence of the generalized alternating direction method of multipliers. *J. Sci. Comput.* 66(3):889–916.
- Drezner Z (2003) A new genetic algorithm for the quadratic assignment problem. *INFORMS J. Comput.* 15(3):320–330.
- Drusvyatskiy D, Wolkowicz H (2017) The many faces of degeneracy in conic optimization. *Foundations Trends Optim.* 3(2):77–170.
- Edwards C (1977) The derivation of a greedy approximator for the Koopmans-Beckmann quadratic assignment problem. *Proc. CP77 Combinatorial Prog. Conf., Liverpool*, 55–86.
- Elshafei A (1977) Hospital layout as a quadratic assignment problem. *Oper. Res. Quart.* 28:167–179.
- Gambardella L, Taillard É, Dorigo M (1999) Ant colonies for the quadratic assignment problem. *J. Oper. Res. Soc.* 50(2):167–176.
- Garey MR, Johnson DS (1979) *Computers and Intractability* (W. H. Freeman and Company, San Francisco).
- Geoffroy A, Graves G (1976) Scheduling parallel production lines with changeover costs: Practical applications of a quadratic assignment/LP approach. *Oper. Res.* 24:595–610.
- Goemans M, Williamson D (1995) Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming. *J. ACM* 42(6):1115–1145.
- Graham N, Hu H, Im J, Li X, Wolkowicz H (2021) rPRSM Version v1.0 <http://dx.doi.org/10.5281/zenodo.5709968>, <https://github.com/INFORMSJoC/2020.0336>.
- Gu Y, Jiang B, Han D (2015) An indefinite-proximal-based strictly contractive Peaceman-Rachford splitting method. Preprint, submitted January 1, 2021, <https://arxiv.org/abs/1506.02221>.
- Hahn P, Zhu YR, Guignard M, Smith J (2010) Exact solution of emerging quadratic assignment problems. *Internat. Trans. Oper. Res.* 17(5):525–552.
- Han D, Yuan X (2013) Local linear convergence of the alternating direction method of multipliers for quadratic programs. *SIAM J. Numer. Anal.* 51(6):3446–3457.

- Han D, Sun D, Zhang L (2018) Linear rate convergence of the alternating direction method of multipliers for convex composite programming. *Math. Oper. Res.* 43(2):622–637.
- He B, Yuan X (2012) On the  $o(1/n)$  convergence rate of the Douglas-Rachford alternating direction method. *SIAM J. Numerical Anal.* 50(2):700–709.
- He B, Yuan X (2015) On non-ergodic convergence rate of Douglas-Rachford alternating direction method of multipliers. *Numerische Mathematik* 130(3):567–577.
- He Y, Li H, Liu X (2018) Relaxed inertial proximal Peaceman-Rachford splitting method for separable convex programming. *Frontiers Math. China* 13(3):555–578.
- He B, Liu H, Wang Z, Yuan X (2014) A strictly contractive Peaceman-Rachford splitting method for convex programming. *SIAM J. Optim.* 24(3):1011–1040.
- Heffley D (1977) Assigning runners to a relay team. *Optimal Strategies in Sports*, vol. 5 (North Holland, Amsterdam), 169–171.
- Hu H, Sotirov R, Wolkowicz H (2019) Facial reduction for symmetry reduced semidefinite programs. Preprint, submitted February 3, 2022, <https://arxiv.org/abs/1912.10245>.
- Krurup J, Pruzan P (1978) Computer-aided layout design. *Mathematical Programming in Use* (Springer, Berlin), 75–94.
- Li X, Yuan X (2015) A proximal strictly contractive Peaceman-Rachford splitting method for convex programming with applications to imaging. *SIAM J. Imaging Sci.* 8(2):1332–1365.
- Li X, Pong T, Sun H, Wolkowicz H (2021) A strictly contractive Peaceman-Rachford splitting method for the doubly nonnegative relaxation of the minimum cut problem. *Comput. Optim. Appl.* 78(3):853–891.
- Liu Y, Yuan X, Zeng S, Zhang J (2018) Partial error bound conditions and the linear convergence rate of the alternating direction method of multipliers. *SIAM J. Numer. Anal.* 56(4):2095–2123.
- Mitchell J, Pardalos P, Resende M (1998) Interior point methods for combinatorial optimization. *Handbook of Combinatorial Optimization* (Springer, Berlin), 189–297.
- Monteiro R, Svaiter B (2013) Iteration-complexity of block-decomposition algorithms and the alternating direction method of multipliers. *SIAM J. Optim.* 23(1):475–507.
- Oliveira D, Wolkowicz H, Xu Y (2018) ADMM for the SDP relaxation of the QAP. *Math. Programming Comput.* 10(4):631–658.
- Pardalos P, Wolkowicz H, eds. (1994) *Quadratic Assignment and Related Problems* (AMS, Providence, RI).
- Pardalos P, Rendl F, Wolkowicz H (1994) The quadratic assignment problem: A survey and recent developments. Pardalos P, Wolkowicz H, eds. *Quadratic Assignment and Related Problems* (AMS, Providence, RI), 1–42.
- Povh J, Rendl F (2009) Copositive and semidefinite relaxations of the quadratic assignment problem. *Discrete Optim.* 6(3):231–241.
- Rendl F, Sotirov R (2007) Bounds for the quadratic assignment problem using the bundle method. *Math. Programming* 109(2–3, Ser. B): 505–524.
- Rockafellar R (1997) *Convex Analysis*. Princeton Landmarks in Mathematics (Princeton University Press, Princeton, NJ).
- Ugi I, Bauer J, Brandt J, Friedrich J, Gasteiger J, Jochum C, Schubert W (1979) Neue Anwendungsgebiete für computer in der chemie. *Angew. Chem.* 91(2):99–111.
- van den Berg E, Friedlander M (2008/2009) Probing the Pareto frontier for basis pursuit solutions. *SIAM J. Sci. Comput.* 31(2): 890–912.
- von Neumann J (1953) A certain zero-sum two-person game equivalent to the optimal assignment problem. *Contributions to the Theory of Games*, vol. 2, Annals of Mathematics Studies, no. 28 (Princeton University Press, Princeton, NJ), 5–12.
- Wolkowicz H, Saigal R, Vandenberghe L, eds. (2000) *Handbook of Semidefinite Programming*. International Series in Operations Research & Management Science (Kluwer Academic Publishers, Boston, MA).
- Yang L, Sun D, Toh KC (2015) SDPNAL+: A majorized semismooth Newton-CG augmented Lagrangian method for semidefinite programming with nonnegative constraints. *Math. Programming Comput.* 7(3):331–366.
- Yang W, Han D (2016) Linear convergence of the alternating direction method of multipliers for a class of convex optimization problems. *SIAM J. Numer. Anal.* 54(2):625–640.
- Yuan X, Zeng S, Zhang J (2020) Discerning the linear convergence of ADMM for structured convex optimization through the lens of variational analysis. *J. Machine Learn. Res.* 21:1–75.
- Zhao Q, Karisch S, Rendl F, Wolkowicz H (1998) Semidefinite programming relaxations for the quadratic assignment problem. *J. Combinatorial Optim.* 2(1):71–109.