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

Naomi Graham   Hao Hu   Jiyoung Im   Xinxin Li*   Henry Wolkowicz

7:39pm, December 26, 2020

Department of Combinatorics and Optimization
Faculty of Mathematics, University of Waterloo, Canada.
Research supported by NSERC.

Contents

1 Introduction **2**
- 1.1 Background . . . 4
  - 1.1.1 Further Notation . . . 6
- 1.2 Contributions and Outline . . . 7

2 The DNN Relaxation and Optimality **7**
- 2.1 Novel Derivation of **DNN** Relaxation . . . 7
  - 2.1.1 Gangster Constraints . . . 7
  - 2.1.2 Facially Reduced **DNN** Relaxations . . . 9
  - 2.1.3 Redundant Constraints . . . 10
- 2.2 Main Model and Optimality Conditions . . . 11
  - 2.2.1 Dual Multiplier . . . 13

3 The rPRSM Algorithm **14**
- 3.1 Outline and Convergence for **rPRSM** . . . 14
- 3.2 Implementation details . . . 17
  - 3.2.1 $\boldsymbol{R}$-subproblem . . . 17
  - 3.2.2 $\boldsymbol{Y}$-subproblem . . . 18
- 3.3 Bounding from Approximate Solutions . . . 19
  - 3.3.1 Lower Bound from Relaxation . . . 19
  - 3.3.2 Upper Bound from Nearest Permutation Matrix . . . 21

*School of Mathematics, Jilin University, Changchun, China. E-mail: `xinxinli@jlu.edu.cn`. This work was supported by the National Natural Science Foundation of China (No.11601183) and Natural Science Foundation for Young Scientist of Jilin Province (No. 20180520212JH).

4 Numerical Experiments with rPRSM 21
4.1 Parameter Settings and Stopping Criteria . . . 22
4.2 Empirical Results . . . 22
4.2.1 Small Size . . . 23
4.2.2 Medium Size . . . 23
4.2.3 Large Size . . . 23
4.3 Comparisons to Other Methods . . . 24

5 Conclusion 26

Index 26

Bibliography 29

List of Figures

1 Relative Gap for **rPRSM** and C-SDP . . . 25
2 Numerical Comparison for **rPRSM** and F2-RLT2-DA . . . 26

List of Tables

4.1 QAPLIB Instances of Small Size . . . 24
4.2 QAPLIB Instances of Medium Size . . . 25
4.3 QAPLIB Instances of Large Size . . . 25

Abstract

We revisit and strengthen splitting methods for solving doubly nonnegative, **DNN**, relaxations of the quadratic assignment problem, **QAP**. We use a modified restricted contractive splitting method, **rPRSM**, approach. Our strengthened bounds exploit new subproblems and new dual multiplier estimates to improve on the bounds and convergence results in the literature.

Key Words: quadratic assignment problem, semidefinite relaxation, doubly nonnegative relaxation, facial reduction, Peaceman-Rachford splitting method.

AMS Subject Classification: 90C22, 90C25, 90C27, 90C59.

1 Introduction

We revisit and strengthen splitting methods for solving doubly nonnegative, **DNN**, relaxations of the quadratic assignment problem, **QAP**. Here **DNN** refers to the semidefinite programming, **SDP**, relaxation with additional nonnegativity constraints on *all* the elements of the matrix variable Y. We use a modified *restricted contractive Peaceman-Rachford splitting method, **rPRSM*** approach. 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 as well as our new bounding techniques. In addition, we get improved rates of convergence from strengthened subproblems and dual multiplier estimates. Our results significantly improve on the recent results in [30].

We include a novel derivation of facial reduction, **FR**, and the gangster constraints, in order to show the strong connections between them, and to illustrate the many redundant constraints that are created. We then take advantage of these redundant constraints in the subproblems in our algorithm and in deriving explicit values for some of the dual variables.

The *quadratic assignment problem, **QAP***, is one of the fundamental combinatorial optimization problems in the field of operations research, and includes many important applications. It is arguably one of the hardest of the *NP*-hard problems with problems of size $n = 30$ still being a challenge, and where proving optimality is particularly difficult; for discussions see e.g., [3, 22]. 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, e.g., 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: scheduling, production, computer manufacture (VLSI design), chemistry (molecular conformation), communication, and other fields, see e.g., [16, 19, 24, 26, 35]. Moreover, many classical combinatorial optimization problems, including the travelling salesman problem, maximum clique problem, and graph partitioning problem, can all be expressed as a **QAP**. For more information see e.g., [5, 9, 11, 31, 32].

That the **QAP** (1.1) is *NP*-hard is given in [18]. The cardinality of the feasible set of permutation matrices Π is $n!$ and it is known that problems typically have *many* local minima. Up to now, there are three main classes of methods for solving **QAP**. The first type is heuristic algorithms, such as genetic algorithms, e.g., [13], ant systems [17] and meta-heuristic algorithms, e.g., [4]. 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 semidefinite programming, **SDP**. Semidefinite programming is proven to have successful implementations and provides tight relaxations, see [2, 39]. There are many well-developed **SDP** solvers based on e.g., interior point methods, e.g., [1, 29, 38]. 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 at el., [30] use 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 [27] that concentrates on the min-cut problem. In addition, we note the work in [25] that also uses **FR** on **QAP** problems, but concentrates on exploiting group symmetry structure.)

1.1 Background

It is known e.g., [15], that many of the **QAP** models, such as the facility location problem, can be formulated using the *trace formulation*:

$$p^*_{\mathbf{QAP}} := \min_{X \in \Pi} \langle AXB - 2C, X \rangle, \tag{1.1}$$

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, i.e., $\langle Y, X \rangle = \operatorname{tr}(YX^T)$, and Π denotes the set of $n \times n$ permutation matrices.

Remark 1.1. *We note that the location problem is symmetric in facilities and locations, i.e., 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 is troublesome if both matrices A, B have zeros in off-diagonal positions, as is the case for many of the instances in QAPLIB [8], the data source that we use.*

We use the following notation from [30]. We denote the *matrix lifting*

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

where $\operatorname{vec}(X)$ 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, $\operatorname{rank}(Y) = 1$. Indexing the rows and columns of Y from 0 to n^2, we can express Y in (1.2) using a block representation as follows:

$$Y = \begin{bmatrix} Y_{00} & \bar{y}^T \\ \bar{y} & \overline{Y} \end{bmatrix}, \quad \bar{y} = \begin{bmatrix} Y_{(10)} \\ Y_{(20)} \\ \vdots \\ Y_{(n0)} \end{bmatrix}, \quad \text{and} \quad \overline{Y} = xx^T = \begin{bmatrix} \overline{Y}_{(11)} & \overline{Y}_{(12)} & \cdots & \overline{Y}_{(1n)} \\ \overline{Y}_{(21)} & \overline{Y}_{(22)} & \cdots & \overline{Y}_{(2n)} \\ \vdots & \ddots & \ddots & \vdots \\ \overline{Y}_{(n1)} & \ddots & \ddots & \overline{Y}_{(nn)} \end{bmatrix}, \tag{1.3}$$

where

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

Let

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

where $\otimes$ denotes the Kronecker product. We further scale L_Q below in (2.8) and (2.9), page 12. With the above notation and matrix lifting, we can reformulate the **QAP** (1.1) equivalently as

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

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

In [39], Zhao et al. derive an **SDP** relaxation as the dual of the Lagrangian relaxation of a quadratically constrained version of (1.4), i.e., the constraint that $X \in \Pi$ is replaced by quadratic constraints, e.g.,

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

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 $Y = \widehat{V} R \widehat{V}^T$, for some full column rank matrix $\widehat{V} \in \mathbb{R}^{(n^2+1)\times((n-1)^2+1)}$ defined below 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:

$$(\mathbf{SDP}) \qquad \begin{array}{ll} \min\limits_{R} & \langle \widehat{V}^T L_Q \widehat{V}, R\rangle \\ \text{s.t.} & \mathcal{G}_{\bar{J}}(\widehat{V} R \widehat{V}^T) = u_0 \\ & R \in \mathbb{S}_+^{(n-1)^2+1}. \end{array} \tag{1.5}$$

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, see [39, Lemma 5.1, Lemma 5.2]. We refer to [39] for details on using the dual of the Lagrangian dual for the derivation of this *facially reduced* ***SDP***.

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

1. Let $\widehat{Y}$ 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 $\widehat{V} \in \mathbb{R}^{(n^2+1)\times((n-1)^2+1)}$ have orthonormal columns that span the range of $\widehat{Y}$.[1] Every feasible Y of the **SDP** relaxation is contained in the *minimal face,* $\mathcal{F}$ of $\mathbb{S}_+^{n^2+1}$:

$$\mathcal{F} = \widehat{V} \mathbb{S}_+^{(n-1)^2+1} \widehat{V}^T \trianglelefteq \mathbb{S}_+^{n^2+1};$$

$$Y \in \mathcal{F} \implies \text{range}(Y) \subseteq \text{range}(\widehat{V}), \quad Y \in \text{relint}(\mathcal{F}) \implies \text{range}(Y) = \text{range}(\widehat{V}).$$

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

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

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:*

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

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.*

[1] There are several ways of constructing such a matrix $\widehat{V}$. One way is presented in Proposition 2.6, below.

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 $\overline{Y} \in \mathbb{S}^{n^2}$ corresponding to:

$$\begin{array}{l}\text{(a) the off-diagonal elements in the } n \text{ diagonal blocks in } \overline{Y} \text{ in (1.3) ;} \\ \text{(b) the diagonal elements in the off-diagonal blocks in } \overline{Y} \text{ in (1.3) .}\end{array} \tag{1.8}$$

   Many of the constraints that arise from the index set J are redundant. We could remove the indices in the submatrix $\overline{Y} \in \mathbb{S}^{n^2}$ corresponding to all the diagonal 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 pre-specifying 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 1 only in the first coordinate, i.e., the 0-th unit vector. Therefore (1.5) forces all the values of $\widehat{V}R\widehat{V}^T$ corresponding to the indices in $\bar{J}$ to be zero. It also implies that the first entry of $\mathcal{G}_{\bar{J}}(\widehat{V}R\widehat{V}^T)$ is equal to 1, 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 1 only in the (00)-position. Therefore (1.5) forces all the values of $\widehat{V}R\widehat{V}^T$ corresponding to the indices in $\bar{J}$ to be zero, except for the 00 element of $\widehat{V}R\widehat{V}^T$.

Since 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. [30] propose using an **ADMM** approach. They introduce interval cuts (constraints) and obtain a *doubly nonnegative,* ***DNN***, model. The **ADMM** approach is further motivated by the natural splitting of variables that arises with facial reduction:

$$(\mathbf{DNN}) \qquad \begin{array}{rl} \min\limits_{R,Y} & \langle L_Q, Y\rangle \\ \text{s.t.} & \mathcal{G}_{\bar{J}}(Y) = u_0 \\ & Y = \widehat{V}R\widehat{V}^T \\ & R \succeq 0 \\ & 0 \le Y \le 1. \end{array} \tag{1.9}$$

The output of **ADMM** is used to compute lower and upper bounds to the original **QAP** (1.1). For most instances in QAPLIB[2], [30] 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.

1.1.1 Further Notation

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 $\mathbb{S}^n_+$ ($\mathbb{S}^n_{++}$, resp.) to denote the cone of n-by-n positive semidefinite (definite) matrices. We write $X \succeq 0$ if $X \in \mathbb{S}^n_+$, and $X \succ 0$ 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 $A \in R^{m\times n}$, we use $\text{range}(A)$ and $\text{null}(A)$ to denote the range of A and the null space of A, respectively.

For $n \ge 1$, 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. And, recall that u_0 is the first unit vector.

[2] http://coral.ise.lehigh.edu/data-sets/qaplib/qaplib-problem-instances-and-solutions/

1.2 Contributions and Outline

We begin in Section 2 with the modelling 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 *restricted contractive Peaceman-Rachford splitting method,* ***rPRSM*** for solving the strengthened model. We use redundant constraints to strengthened the subproblems and to strengthen 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 as well as with new termination conditions.

For our numerical results in Section 4 we use data from QAPLIB [8]. We show significant improvements over the previous results in [30]. Our concluding remarks are in Section 5.

2 The DNN Relaxation and Optimality

In this section we present details of our *doubly nonnegative,* ***DNN***, relaxation of the **QAP**. This is related to the **SDP** relaxation derived in [39] and the **DNN** relaxation in [30]. 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 [39] 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, i.e.,

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

We let $\mathcal{D} = \mathcal{D}_e \cap \{X \geq 0\}$ denote the *doubly stochastic matrices*. The classical Birkhoff-von Neumann Theorem [6,37] 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, Π, is equal to the intersection:

$$\Pi = \mathcal{D}_e \cap \mathcal{Z}. \tag{2.1}$$

It is of interest that the representation in (2.1) leads to <u>*both*</u> the gangster constraints and facial reduction for the **SDP** relaxation on the lifted variable Y in (1.3), and in particular on $\overline{Y}$. 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 $\overline{Y}$ are all zero; while 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 $\overline{Y}$ 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*

$$\langle A, B\rangle = 0 \iff A \circ B = 0. \qquad \square$$

The following Lemma 2.2 and Corollary 2.3 together show how the representation of Π in (2.1) gives rise to the gangster constraint on the lifted matrix Y in (1.2). We first find (Hadamard product) *exposing vector*s in Lemma 2.2 for lifted zero-one vectors.

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

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

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

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

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

$$\begin{array}{rrcl} & (e_n^T \otimes I_n)x & = & e_n \\ \implies & (e_n \otimes I_n)(e_n^T \otimes I_n)x & = & (e_n \otimes I_n)e_n = e_{n^2} \\ \implies & [(e_n \otimes I_n)(e_n^T \otimes I_n) - I_{n^2}]x & = & e_{n^2} - x \\ \implies & [(e_n e_n^T \otimes I_n) - I_{n^2}]xx^T & = & e_{n^2}x^T - xx^T \\ \implies & \mathrm{tr}\left([(e_n e_n^T \otimes I_n) - I_{n^2}]\, xx^T\right) & = & \mathrm{tr}(e_{n^2}x^T - xx^T). \end{array}$$

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

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

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 2.3. *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. $0 \le Y \le 1$, $Y \succeq 0$, $\mathrm{rank}(Y) = 1$.

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;
- 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 $\mathcal{G}_{\bar{J}}(Y) = u_0$, i.e., Item 1 holds. Item 2 follows from (2.1) and the structure of Y in (1.2). ☐

The following Proposition 2.4 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 2.4. *Suppose that for all $X \in \Pi$, and Y formed from (1.2), If there exists an index (s,t) such that $Y_{st} = Y_{ts} = 0$, but $\{(s,t) \cup (t,s)\} \notin \bar{J}$, i.e., (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.

1. Suppose that $s = (ij) = t, i, j \geq 1$, and so we have $Y_{(ij)(ij)} = 0$. But $\overline{Y} = xx^T$, by (1.2), implies that $X_{ij} = 0$; and this does not hold for all $X \in \Pi$, a contradiction, i.e., we cannot add a diagonal element of Y to the gangster set.
2. If $s \neq t$, we have $Y_{st} = 0$. Since $X \in \Pi$, we infer that Y_{ss} or Y_{tt} must be zero. Note that 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, i.e., as above we cannot add this element of Y to the gangster set. ☐

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 $Y = \widehat{V} R \widehat{V}^T$ in (1.9), arises from consideration of an exposing vector. We define

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

and

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

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 $\operatorname{rank}(H) = 2n - 1$ and the rows sum up to $2e_{n^2}^T$. Then $\operatorname{rank}(K) = 2n - 1$ as well. Moreover, the following Lemma 2.5 is clear.

Lemma 2.5. *Let H be given in (2.2); and let*

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

Then

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

$\square$

From Lemma 2.5, K is an *exposing vector* for all feasible Y_x, see e.g., [14]. Then we can choose a full column rank $\widehat{V}$ with the range equal to the nullspace of K and obtain *facial reduction*, i.e., all feasible Y for the **SDP** relaxation satisfy

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

There are clearly many choices for $\widehat{V}$. We present one in Proposition 2.6 from [39]. But in our implementations we follow [30] and use one with orthonormal columns.

Proposition 2.6 ([39])**.** *Let*

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

and let K be given as in (2.3)*. Then we have* $\operatorname{range}(\widehat{V}) = \operatorname{range}(K)$. $\square$

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*:

$$\begin{array}{rcll} p^*_{\mathbf{DNN}} & = & \min & \langle L_Q, Y\rangle \\ & & \text{s.t.} & \mathcal{G}_{\bar{J}}(Y) = u_0 \\ & & & Y = \widehat{V}R\widehat{V}^T \\ & & & R \succeq 0 \\ & & & 0 \le Y \le 1. \end{array} \tag{2.4}$$

A strictly feasible $\hat{R} \succ 0$ for the facially reduced **SDP** relaxation is given in [39], 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 the model (2.4) that are useful in the subproblems and in pre-specifying 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 [39].

Definition 2.7 ([39, Page 80])**.** *Let $Y \in \mathbb{S}^{n^2+1}$ be blocked as in* (1.3)*. We define the linear transformation* $\mathrm{b}^0\mathrm{diag}\,(Y) : \mathbb{S}^{n^2+1} \to \mathbb{S}^n$ *by the sum of the n-by-n diagonal blocks of Y, i.e.,*

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

We define the linear transformation $\mathrm{o}^0\mathrm{diag}\,(Y) : \mathbb{S}^{n^2+1} \to \mathbb{S}^n$ *by the trace of the block $\overline{Y}_{(ij)}$, i.e.,*

$$\mathrm{o}^0\mathrm{diag}\,(Y) := \left(\operatorname{tr}\left(\overline{Y}_{(ij)}\right)\right)_{ij} \in \mathbb{S}^n.$$

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

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

With Definition 2.7, the following lemma can be derived from [39, Lemma 3.1]. Lemma 2.8 indeed shows three redundant constraints of (2.4).

Lemma 2.8 ([39, Lemma 3.1])**.** *Let V be any full column rank matrix such that* $\mathrm{range}(V) = \mathrm{range}(\widehat{V})$*, where $\widehat{V}$ is given in Proposition 2.6. Suppose $Y = VRV^T$ and $\mathcal{G}_{\bar{J}}(Y) = u_0$ hold. Then the following hold:*

1. $\mathrm{arrow}\,(Y) = 0$.
2. $\mathrm{b}^0\mathrm{diag}\,(Y) = I_n$ *and* $\mathrm{o}^0\mathrm{diag}\,(Y) = I_n$. □

The following Proposition 2.9 shows that the constraint $\mathrm{tr}(R) = n + 1$ is also redundant for model (2.4).

Proposition 2.9. *With orthonormal $\widehat{V}$ whose range is equal to* $\mathrm{range}(K)$*, the constraints $Y = \widehat{V}R\widehat{V}^T$, $R \succeq 0$ and $Y \in \mathcal{Y}$ yield that* $\mathrm{tr}(R) = n + 1$.

Proof. By Lemma 2.8, $\mathrm{b}^0\mathrm{diag}\,(Y) = I_n$ hold. Then with $Y_{00} = 1$, we see that $\mathrm{tr}(Y) = n + 1$. By cyclicity of the trace operator and $\widehat{V}^T\widehat{V} = I$, we see that

$$\mathrm{tr}(R) = \mathrm{tr}(R\widehat{V}^T\widehat{V}) = \mathrm{tr}\left(\widehat{V}R\widehat{V}^T\right) = \mathrm{tr}(Y) = n + 1.$$ □

Remark 2.10. *We take advantage of this in the corresponding R-subproblem and the computation of the lower bound of **QAP**. Note that 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

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

and

$$\mathcal{Y} := \{Y \in \mathbb{S}^{n^2+1} : \mathcal{G}_{\bar{J}}(Y) = u_0, 0 \le Y \le 1, \mathrm{b}^0\mathrm{diag}\,(Y) = I,\ \mathrm{o}^0\mathrm{diag}\,(Y) = I,\ \mathrm{arrow}\,(Y) = 0\}. \tag{2.6}$$

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

$$(\textbf{DNN}) \qquad \begin{array}{rcl} p^*_{\textbf{DNN}} := & \min\limits_{R,Y} & \langle L_Q, Y\rangle \\ & \text{s.t.} & Y = \widehat{V}R\widehat{V}^T \\ & & R \in \mathcal{R} \\ & & Y \in \mathcal{Y}. \end{array} \tag{2.7}$$

The following property of feasible points $Y \in \mathcal{Y}$ in Proposition 2.11 is used in the computation of the Y-subproblem of our algorithm.

Proposition 2.11. *For any $Y \in \mathcal{Y}$, let $\bar{X} = \mathrm{Mat}(\mathrm{diag}(\overline{Y})) \in \mathbb{R}^{n\times n}$ be the matrix formed from the diagonal of $\overline{Y}$ after ignoring the* 00 *element. Then $\bar{X} \in \mathcal{D}$. Moreover, this holds for the first row (and column) of Y.*

Proof. From the $\mathcal{Y}$ constraints $\mathrm{b}^0\mathrm{diag}\,(Y) = I, \mathrm{o}^0\mathrm{diag}\,(Y) = I$, respectively, we get $\sum_{k=1}^n \mathrm{diag}(Y_{(kk)}) = e$ and $\mathrm{tr}(Y_{(kk)}) = 1,\ \forall i \in \{1,\ldots,k\}$, respectively. Then by the definition of $\bar{X}$, we immediately have $\bar{X}e = e$ and $\bar{X}^T e = e$. Note that 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 follow from the arrow constraint. □

Remark 2.12 ((doubly) stochastic optimal Y)**.** *Proposition 2.11 shows that for any feasible $Y \in \mathcal{Y}$, when ignoring the* (00) *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 $X = \mathrm{Mat}\, w$ 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 $P_V = \widehat{V}\widehat{V}^T$; and let $\alpha, \delta > 0$ be the shift and scale parameters. Note that $Y = \widehat{V}R\widehat{V}^T$ implies

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

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

$$L_Q \leftarrow \delta(P_V L_Q P_V + \alpha I). \tag{2.9}$$

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

The Lagrangian function of model (2.7) is:

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

Since a strictly feasible $\hat{R}$, with $\hat{Y} = \widehat{V}\hat{R}\widehat{V}$, exists, we conclude that the following first order optimality conditions for the model (2.7) hold:

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

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

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

where the set $\mathcal{N}_{\mathcal{R}}(R)$ (resp. $\mathcal{N}_{\mathcal{Y}}(Y)$) is the normal cone to the set $\mathcal{R}$ (resp. $\mathcal{Y}$) at R (resp. Y). By the definition of the normal cone, we can easily obtain the following Proposition 2.13.

Proposition 2.13 (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,*

$$R = \mathcal{P}_{\mathcal{R}}(R + \widehat{V}^T Z \widehat{V}) \tag{2.12a}$$

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

$$Y = \widehat{V} R \widehat{V}^T. \tag{2.12c}$$

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 $Y = \widehat{V} R \widehat{V}^T$ 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 2.14 shows that there exists a dual multiplier $Z \in \mathbb{S}^{n^2+1}$ of the 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 2.14. *Let $E_A = \left[\begin{array}{c|c} 1 & 0 \\ \hline 0 & E_{n^2} - I_{n^2} - I_{J_R} \end{array}\right]$, where I_{J_R} is the zero matrix except for 1 in the positions of the redundant gangster elements J_R, Item 3 page 6. Let*

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

and let

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

Consider the following problem:

$$\min_{R,Y} \{ \langle L_Q, Y \rangle : Y = \widehat{V} R \widehat{V}^T,\ R \in \mathcal{R},\ Y \in \mathcal{Y}_A \}. \tag{2.13}$$

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), i.e., they are an optimal primal-dual solution for (2.7).*

Proof. Note that $\mathcal{Y} \subset \mathcal{Y}_A$, where we remove the $\mathrm{b}^0\mathrm{diag}$, $\mathrm{o}^0\mathrm{diag}$ 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) since $\mathcal{Y} \subset \mathcal{Y}_A$. Consider a feasible pair (R, Y) to (2.13). By Item 2 of Lemma 2.8 and the positive semidefiniteness of

$Y = \widehat{V}R\widehat{V}^T$, we have that $\mathrm{b^0diag}\,(Y) = I_n$ and the elements of the diagonal of Y are in the interval $[0,1]$. In addition, since $\mathrm{arrow}\,(Y) = 0$, 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 so 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

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

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

$$Y^* = \widehat{V}R^*\widehat{V}^T, \quad R^* \in \mathcal{R},\, Y^* \in \mathcal{Y}_A. \tag{2.14c}$$

By the definition of the normal cone, we have

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

Since 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.

In order 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, since $Y^* \in \mathcal{Y} \subseteq \mathcal{Y}_A$, 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). □

Remark 2.15. *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 The rPRSM Algorithm

We now present the details of a modification of the so-called restricted contractive Peaceman-Rachford splitting method, **PRSM**, or symmetric **ADMM**, e.g., [23,28]. Our modification involves redundant constraints on subproblems as well as 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:

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

where β is a positive penalty parameter.

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

Algorithm 3.1 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**
$R^{k+1} = \operatorname{argmin}_{R\in\mathcal{R}} \mathcal{L}_A(R, Y^k, Z^k)$
$Z^{k+\frac{1}{2}} = Z^k + \gamma\beta \cdot \mathcal{P}_{\mathcal{Z}_0}\left(Y^k - \widehat{V}R^{k+1}\widehat{V}^T\right)$
$Y^{k+1} = \operatorname{argmin}_{Y\in\mathcal{Y}} \mathcal{L}_A(R^{k+1}, Y, Z^{k+\frac{1}{2}})$
$Z^{k+1} = Z^{k+\frac{1}{2}} + \gamma\beta \cdot \mathcal{P}_{\mathcal{Z}_0}\left(Y^{k+1} - \widehat{V}R^{k+1}\widehat{V}^T\right)$
end while

Remark 3.1. *Algorithm 3.1 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 3.1, we point out the following. The R-update and the Y-update in Algorithm 3.1 are well-defined, i.e., 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 $\mathcal{Y}$. We note that many of the constraints are redundant in the* ***SDP*** *part of the problem, e.g., the trace on R, and the* $\mathrm{b^0diag}$, $\mathrm{o^0diag}$, arrow *on Y. However, these constraints are not redundant within the subproblems themselves and are inexpensive to include. They improve the rate of convergence and the quality of the Y when stopping the* ***rPRSM*** *algorithm early.*

We also note that, in Algorithm 3.1, we update the dual variable Z both after the R-update and the Y-update. This pattern of update in our Algorithm 3.1 is closely related to the strictly contractive Peaceman-Rachford splitting method, ***PRSM****; see e.g., [23, 28]. Indeed, we show in Theorem 3.2 below, that our algorithm can be viewed as a version of semi-proximal strictly contractive* ***PRSM****, see e.g., [21, 28], applied to (3.2). Hence, the convergence of our algorithm can be deduced from the general convergence theory of semi-proximal strictly contractive* ***PRSM****.*

Theorem 3.2. *Let $\{R^k\}, \{Y^k\}, \{Z^k\}$ be the sequences generated by Algorithm 3.1. 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 semi-proximal restricted contractive **PRSM** in [21, 28] applied to the following problem (3.2), where $\mathcal{P}_{\mathcal{Z}_0^c}$ is the projection onto the orthogonal complement of $\mathcal{Z}_0$, i.e., $\mathcal{P}_{\mathcal{Z}_0^c} = I - \mathcal{P}_{\mathcal{Z}_0}$:

$$\begin{array}{ll} \min\limits_{R,Y} & \langle L_Q, \mathcal{P}_{\mathcal{Z}_0}(Y) + \mathcal{P}_{\mathcal{Z}_0^c}(VRV^T)\rangle \\ \text{s.t.} & \mathcal{P}_{\mathcal{Z}_0}(Y) = \mathcal{P}_{\mathcal{Z}_0}(\widehat{V}R\widehat{V}^T) \\ & R \in \mathcal{R} \\ & Y \in \mathcal{Y}. \end{array} \tag{3.2}$$

We show that the sequence generated by the semi-proximal restricted contractive **PRSM** in [21,28] converges to a *Karush-Kuhn-Tucker,* ***KKT*** point of (2.7). In the second step, we show that the sequence generated by Algorithm 3.1 is identical with the sequence generated by the semi-proximal restricted contractive **PRSM** applied to (3.2).

Step 1: We apply the semi-proximal strictly contractive **PRSM** given in [21, 28] 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:

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

where $\gamma \in (0,1)$ is an under-relaxation parameter. Note that 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

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

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

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

while $\mathcal{P}_{\mathcal{Z}_0^c}(\tilde{Y}^{k+1})$ can be chosen to be

$$\mathcal{P}_{\mathcal{Z}_0^c}(\tilde{Y}^{k+1}) = \mathcal{P}_{\mathcal{Z}_0^c}(\widehat{V}\tilde{R}^{k+1}\widehat{V}^T)\ , \quad \forall\, k \ge 0. \tag{3.4}$$

Finally, one can also deduce by induction that $\tilde{Z}^k \in \mathcal{Z}_A$, for all k, since $Z^0 \in \mathcal{Z}_A$. From the general convergence theory of semi-proximal strictly contractive **PRSM** given in [21, 28], we have

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

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

$$0 \in \widehat{V}^T\mathcal{P}_{\mathcal{Z}_0^c}(L_Q)\widehat{V} - \widehat{V}^T\mathcal{P}_{\mathcal{Z}_0}(Z^*)\widehat{V} + \mathcal{N}_{\mathcal{R}}(R^*) \tag{3.5a}$$

$$0 \in \mathcal{P}_{\mathcal{Z}_0}(L_Q) + \mathcal{P}_{\mathcal{Z}_0}(Z^*) + \mathcal{N}_{\mathcal{Y}}(Y^*) \tag{3.5b}$$

$$\mathcal{P}_{\mathcal{Z}_0}(Y^*) = \mathcal{P}_{\mathcal{Z}_0}(\widehat{V}R^*\widehat{V}^T). \tag{3.5c}$$

Since we update $\mathcal{P}_{\mathcal{Z}_0^c}(\tilde{Y}^k)$ by (3.4), we also have that

$$\mathcal{P}_{\mathcal{Z}_0^c}(Y^*) = \mathcal{P}_{\mathcal{Z}_0^c}(\widehat{V}R^*\widehat{V}^T). \tag{3.6}$$

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

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

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

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

Hence, we have shown that the sequence generated by by (3.3) and (3.4), converges to a **KKT** point of the 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 3.1. 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 $k \geq 0$. Since $\tilde{Z}^k \in \mathcal{Z}_A$ and (3.4) holds, we can rewrite the R-subproblem in (3.3) as follows:

$$
\begin{aligned}
& \underset{R\in\mathcal{R}}{\operatorname{argmin}} \langle L_Q, \mathcal{P}_{\mathcal{Z}_0^c}(\widehat{V}R\widehat{V}^T)\rangle - \langle \tilde{Z}^k, \mathcal{P}_{\mathcal{Z}_0}(\widehat{V}R\widehat{V}^T)\rangle + \frac{\beta}{2}\left\|\mathcal{P}_{\mathcal{Z}_0}(\tilde{Y}^k - \widehat{V}R\widehat{V}^T)\right\|_F^2 + \frac{\beta}{2}\left\|\mathcal{P}_{\mathcal{Z}_0^c}(\widehat{V}\tilde{R}^k\widehat{V}^T - \widehat{V}R\widehat{V}^T)\right\|_F^2 \\
= \; & \underset{R\in\mathcal{R}}{\operatorname{argmin}} \langle \mathcal{P}_{\mathcal{Z}_0^c}(L_Q) - \mathcal{P}_{\mathcal{Z}_0}(\tilde{Z}^k), \widehat{V}R\widehat{V}^T\rangle + \frac{\beta}{2}\left\|\mathcal{P}_{\mathcal{Z}_0}(\tilde{Y}^k - \widehat{V}R\widehat{V}^T)\right\|_F^2 + \frac{\beta}{2}\left\|\mathcal{P}_{\mathcal{Z}_0^c}(\widehat{V}\tilde{R}^k\widehat{V}^T - \widehat{V}R\widehat{V}^T)\right\|_F^2 \\
= \; & \underset{R\in\mathcal{R}}{\operatorname{argmin}} \langle -\mathcal{P}_{\mathcal{Z}_0^c}(\tilde{Z}^k) - \mathcal{P}_{\mathcal{Z}_0}(\tilde{Z}^k), \widehat{V}R\widehat{V}^T\rangle + \frac{\beta}{2}\left\|\tilde{Y}^k - \widehat{V}R\widehat{V}^T\right\|_F^2 \\
= \; & \underset{R\in\mathcal{R}}{\operatorname{argmin}} -\langle \tilde{Z}^k, \widehat{V}R\widehat{V}^T\rangle + \frac{\beta}{2}\left\|\tilde{Y}^k - \widehat{V}R\widehat{V}^T\right\|_F^2,
\end{aligned}
$$

where the second "=" is due to $\tilde{Z}^k \in \mathcal{Z}_A$ and (3.4). The above is equivalent to the R-subproblem in Algorithm 3.1, since $\tilde{Z}^k = Z^k$ and $\tilde{Y}^k = Y^k$ by the induction hypothesis. This shows that $\tilde{R}^{k+1} = R^{k+1}$ and it follows that $\tilde{Z}^{k+\frac{1}{2}} = Z^{k+\frac{1}{2}}$. Since $Z^{k+\frac{1}{2}} \in \mathcal{Z}_A$, we can rewrite the Y-subproblem in Algorithm 3.1 as

$$
\begin{aligned}
& \underset{Y\in\mathcal{Y}}{\operatorname{argmin}} \langle L_Q + Z^{k+\frac{1}{2}}, Y\rangle + \frac{\beta}{2}\|Y - \widehat{V}R^{k+1}\widehat{V}^T\|_F^2 \\
= \; & \underset{Y\in\mathcal{Y}}{\operatorname{argmin}} \langle \mathcal{P}_{\mathcal{Z}_0}(L_Q + Z^{k+\frac{1}{2}}), Y\rangle + \frac{\beta}{2}\|\mathcal{P}_{\mathcal{Z}_0}(Y - \widehat{V}R^{k+1}\widehat{V}^T)\|_F^2 + \frac{\beta}{2}\|\mathcal{P}_{\mathcal{Z}_0^c}(Y - \widehat{V}R^{k+1}\widehat{V}^T)\|_F^2 \\
= \; & \underset{Y\in\mathcal{Y}}{\operatorname{argmin}} \langle L_Q, \mathcal{P}_{\mathcal{Z}_0}(Y)\rangle + \langle Z^{k+\frac{1}{2}}, \mathcal{P}_{\mathcal{Z}_0}(Y)\rangle + \frac{\beta}{2}\left\|\mathcal{P}_{\mathcal{Z}_0}(Y - \widehat{V}R^{k+1}\widehat{V}^T)\right\|_F^2 + \frac{\beta}{2}\|\mathcal{P}_{\mathcal{Z}_0^c}(Y - \widehat{V}R^{k+1}\widehat{V}^T)\|_F^2,
\end{aligned}
$$

where the first "=" is due to $Z^{k+\frac{1}{2}} \in \mathcal{Z}_A$. 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 above 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 $\{(\tilde{R}^k, \tilde{Y}^k, \tilde{Z}^k)\}$. □

3.2 Implementation details

Note that the explicit Z-updates in Algorithm 3.1 is 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 3.1. We define $\mathcal{P}_{\mathcal{R}}(W)$ to be the projection of W onto the compact set $\mathcal{R}$, where $\mathcal{R} := \left\{R \in \mathbb{S}_+^{(n-1)^2+1} : \operatorname{tr}(R) = n+1\right\}$. 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:

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

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

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; see, for example [10, 36]. Therefore the R-updates reduce to the projection of the vector of the positive eigenvalues of $\widehat{V}^T\left(Y^k + \frac{1}{\beta}Z^k\right)\widehat{V}$ onto the simplex Δ.

3.2.2 Y-subproblem

In this section we present the formula for solving the Y-subproblem in Algorithm 3.1. By completing the square at the current iterates $R^{k+1}, Z^{k+\frac{1}{2}}$, we get

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

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

$$
\mathcal{Y} := \{Y \in \mathbb{S}^{n^2+1} : \mathcal{G}_{\bar{J}}(Y) = u_0,\ 0 \le Y \le 1,\ \mathrm{b}^0\mathrm{diag}\,(Y) = I,\ \mathrm{o}^0\mathrm{diag}\,(Y) = I,\ \mathrm{arrow}\,(Y) = 0\}.
$$

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

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

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

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* $\operatorname{Mat} t$*, $i = 1, \ldots, n$. We denote the i-th subvector in the diagonal (except for the 00 element), first column and first row of T by the column vectors t_i^d, t_i^c and t_i^r, respectively. Then*

$$\begin{array}{rll} s = & \operatorname{argmin}_s & \left(\|s - t^d\|^2 + \|s - t^c\|^2 + \|s - t^r\|^2\right) \\ & \text{s.t.} & \operatorname{Mat}(s) \in \mathcal{D}. \end{array} \tag{3.8}$$

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,

$$\begin{array}{ll} \min\limits_{s} & \|s - \frac{1}{3}(t^d + t^c + t^r)\|^2 \\ \text{s.t.} & \operatorname{Mat}(s) \in \mathcal{D}. \end{array} \tag{3.9}$$

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

$$\begin{array}{ll} \min\limits_{S \in \mathbb{R}^{n \times n}} & \|S - \widetilde{T}_a\|^2 \\ \text{s.t.} & S \in \mathcal{D}. \end{array} \tag{3.10}$$

Denote the optimal solution of (3.10) by S^*, then $s = \operatorname{vec}(S^*)$. This relates with Proposition 2.11, 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 $p^*_{\mathbf{QAP}}$. This can often help in stopping the algorithm early and also prove optimality for our current permutation X for the original **QAP**. This follows on the approach in [27].

3.3.1 Lower Bound from Relaxation

Exact solutions of the 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

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

where $\lambda_{\max}(\widehat{V}^T Z \widehat{V})$ denotes the largest eigenvalue of $\widehat{V}^T Z \widehat{V}$. In Theorem 3.4 below, we show that $\max\limits_{Z} g(Z)$ is indeed the Lagrange dual problem of our main **DNN** relaxation (2.7).

Theorem 3.4. *Let g be the functional defined in (3.11). Then the problem*

$$d_Z^* := \max_Z g(Z) \tag{3.12}$$

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

$$p^*_{\mathbf{DNN}} = d_Z^*, \text{ and } d_Z^* \text{ is attained.}$$

Proof. Note that the function $\widehat{V}^T Z \widehat{V}$ is linear in Z. Therefore the largest eigenvalue function $\lambda_{\max}(\widehat{V}^T Z \widehat{V})$ is a convex function of Z. Thus the argument of the minimum in (3.12)

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

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

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

$$\begin{aligned}
p^*_{\mathbf{DNN}} &= \min_{R\in\mathcal{R},Y\in\mathcal{Y}} \max_{Z} \left\{\langle L_Q, Y\rangle + \langle Z, Y - \widehat{V}R\widehat{V}^T\rangle\right\} \\
&= \max_{Z} \min_{R\in\mathcal{R},Y\in\mathcal{Y}} \left\{\langle L_Q, Y\rangle + \langle Z, Y - \widehat{V}R\widehat{V}^T\rangle\right\} && \text{(3.13a)} \\
&= \max_{Z} \left\{\min_{Y\in\mathcal{Y}} \{\langle L_Q, Y\rangle + \langle Z, Y\rangle\} + \min_{R\in\mathcal{R}} \langle Z, -\widehat{V}R\widehat{V}^T\rangle\right\} \\
&= \max_{Z} \left\{\min_{Y\in\mathcal{Y}} \{\langle L_Q, Y\rangle + \langle Z, Y\rangle\} + \min_{R\in\mathcal{R}} \langle \widehat{V}^T Z\widehat{V}, -R\rangle\right\} \\
&= \max_{Z} \left\{\min_{Y\in\mathcal{Y}} \langle L_Q + Z, Y\rangle - (n+1)\lambda_{\max}(\widehat{V}^T Z\widehat{V})\right\} && \text{(3.13b)} \\
&= d^*_Z,
\end{aligned}$$

where:

1. (3.13a) follows from [34, Corollary 28.2.2, Theorem 28.4] and the fact that (2.7) has generalized Slater points, see [39];[3]

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

We see from [34, Corollary 28.2.2, Corollary 28.4.1] that the dual optimal value d^*_Z is attained. □

Remark 3.5. *Since the Lagrange dual problem in Theorem 3.4 is an unconstrained maximization problem, evaluating g defined in (3.11) at the k-th iterate Z^k yields a valid lower bound for $p^*_{\boldsymbol{DNN}}$, i.e., $g(Z^k) \le p^*_{\boldsymbol{DNN}} \le p^*_{\boldsymbol{QAP}}$. The functional g also strengthens the bound given in [30, 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 2.14 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 $P_V := \widehat{V}\widehat{V}^T$. Then for any $r, s \in \mathbb{R}$, the objective in (2.7) can be replaced by*

$$\langle r(P_V L_Q P_V + sI), Y\rangle. \tag{3.14}$$

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 $p^*_{\boldsymbol{QAP}}$ is an even integer. Therefore applying the ceiling operator to $g(Z^k)$ still gives a valid lower bound to $p^*_{\boldsymbol{QAP}}$. 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]Note that 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.

3.3.2 Upper Bound from Nearest Permutation Matrix

In [30], the authors 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

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

Any solution to the 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 above, the permutation matrices are the extreme points of the set of doubly stochastic matrices $\mathcal{D}$. Hence we reformulate the problem (3.15) as the linear program

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

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

For an approximate optimum Y^{out}, The first approach in [30] sets $\operatorname{vec}(\bar{X})$ to be the first column of Y^{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 $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_r > 0$. And 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 [30] is to use $\operatorname{vec}(\bar{X}) = \lambda_1 v_1$ in solving (3.16), where (λ_1, v_1) is the most dominant eigenpair of Y^{out}.

Inspired by the approximation algorithm in [20], now let ξ be a random vector in $\mathbb{R}^r$ with elements in $(0,1)$, and in decreasing order. We use ξ to perturb the eigenvalues $\lambda_1,\ldots,\lambda_r$ 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 $3\lceil\log(n)\rceil$ and choose the best (smallest) as the upper bound.

4 Numerical Experiments with rPRSM

We now present numerical results for Algorithm 3.1, **rPRSM**, with the bounding strategies discussed in Section 3.3. The parameter settings and stopping criteria are in Section 4.1, below. We use symmetric[4] data from QAPLIP[5]. In Section 4.2 we examine the comparative performance between **rPRSM** and [30, **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 two recently proposed relaxation methods [7, C-SDP] and [12, F2-RLT2-DA].

[4]We exclude instances that have asymmetric data matrices.

[5]http://coral.ise.lehigh.edu/data-sets/qaplib/qaplib-problem-instances-and-solutions/

4.1 Parameter Settings and Stopping Criteria

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

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

   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[6]

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

   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, maxiter = 40000, is reached.

   (b) For given tolerance ϵ, the following bound on the primal and dual residuals holds for m_t sequential times:

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

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

   (c) Let $\{l_1, \ldots, l_k\}$ and $\{u_1, \ldots, u_k\}$ be sequences of lower and upper bounds from Section 3.3.1 and Section 3.3.2, respectively. The lower (resp. upper) bounds do not change for m_l (resp. 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

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

   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 [30, **ADMM**] on instances from QAPLIB. We divide the instances into three groups based on sizes:

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

For **ADMM** we use the parameters from [30], i.e., $\beta = n/3, \gamma = 1.618$; and we adopt the same stopping criteria for both **ADMM** and **rPRSM**. All instances in Tables 4.1 to 4.3 use MATLAB version 2020a on with two Intel Xeon Gold 6244 8-core 3.6 GHz (Cascade Lake) and 192 Gigabyte memory.

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

[6] The formula for Y^0 is introduced in [39, Theorem 3.1].

1. **opt**: global optimal value; marked with * when unknown.
2. **lbd**: lower bound from **rPRSM**;
3. **ubd**: upper bound from **rPRSM**;
4. **rel.gap**: relative gap from **rPRSM**:

$$\text{relative gap} := 2\,\frac{\text{best feasible upper bound} - \text{best lower bound}}{\text{best feasible upper bound} + \text{best lower bound} + 1}; \tag{4.1}$$

5. **rel-opt-gap**: relative optimality gap from **rPRSM** using the known true optimal value and the lower bound;
6. **rel.gap$^{\mathbf{A}}$**: relative gap from [30, **ADMM**] with tolerance $\epsilon = 10^{-5}$;
7. **iter**: number of iterations by **rPRSM** with tolerance $\epsilon = 10^{-5}$;
8. **iter$^{\mathbf{A}}$**: number of iterations from [30, **ADMM**] with tolerance $\epsilon = 10^{-5}$;
9. **time (sec)**: solver **rPRSM** time.

4.2.1 Small Size

Comparing columns **iter** and **iter$^{\mathbf{A}}$** in Table 4.1, we see that 37 instances were treated with fewer iterations using **rPRSM**, i.e., **rPRSM** converges faster in general than **ADMM** for the small-size QAPLIB instances. In particular, 45 out of 46 instances are solved with relative gaps just as good as the ones obtained by **ADMM** and these instances are marked with boldface in Table 4.1. We have found provably optimal solutions for instances

chr12b *chr12c* *chr15a* *chr15b* *chr15c* *chr18a* *chr20a* *chr20b* *esc16e* *esc16f* *esc16j*
had12 *had14* *had16* *had18* *had20* *rou12* *scr12* *scr15* *tai10a* *tai12a*.

We also observe from columns **iter** and **iter$^{\mathbf{A}}$** in Table 4.1 that **rPRSM** gives reduction in number of iterations in many instances; 37 out of 46 instances use fewer number of iterations using **rPRSM** compared to **ADMM**. For **rPRSM** alone we observe that most of the instances show good bounds with reasonable amount of time. Most of the instances are solved within two minutes using the machine described above.

4.2.2 Medium Size

Table 4.2 contains results on 29 QAPLIB instances with sizes $n \in \{22, \ldots, 40\}$. Columns **rel.gap** and **rel.gap$^{\mathbf{A}}$** in Table 4.2 show that **rPRSM** produces competitive relative gaps compared to **ADMM**. In particular, all the instances are solved with relative gaps just as good as the ones obtained by **ADMM** and these instances are marked with boldface in Table 4.2. 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 4.3 contains results on 9 QAPLIB instances with sizes $n \in \{41, \ldots, 64\}$. We observe that **rPRSM** outputs better relative gaps than **ADMM** on 8 instances and this is due to 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 converges faster than **ADMM**.

Table 4.1: QAPLIB Instances of Small Size

Problem Data			Numerical Results					Timing		
#	**name**	**true-opt**	**lbd**	**ubd**	**rel.gap**	**rel.opt.gap**	**rel.gap^A**	**iter**	**iter^A**	**time(sec)**
1	chr12a	9552	9548	9552	0.04	0	0.02	11500	24800	130.04
2	chr12b	9742	9742	9742	**0**	0	0.08	10300	26700	113.96
3	chr12c	11156	11156	11156	**0**	0	0	1600	19400	17.41
4	chr15a	9896	9896	9896	**0**	0	0.28	6700	30900	126.20
5	chr15b	7990	7990	7990	**0**	0	0.03	3500	20300	70.67
6	chr15c	9504	9504	9504	**0**	0	0.08	1800	20000	28.53
7	chr18a	11098	11098	11098	**0**	0	0	2000	20600	61.64
8	chr18b	1534	1534	1794	**15.62**	15.62	75.22	5558	12600	172.94
9	chr20a	2192	2192	2192	**0**	0	0.18	3700	33700	156.45
10	chr20b	2298	2298	2298	**0**	0	0	1200	26200	58.09
11	chr20c	14142	14136	14142	**0.04**	0	0.15	30900	33700	1325.01
12	els19	17212548	17208748	17212548	**0.02**	0	0.35	30800	40000	1106.23
13	esc16a	68	64	74	**14.39**	8.39	41.72	399	597	10.22
14	esc16b	292	290	292	**0.69**	0	6.01	302	386	6.89
15	esc16c	160	154	166	**7.48**	3.67	34.32	399	896	8.58
16	esc16d	16	14	16	**12.90**	0	118.18	299	659	4.96
17	esc16e	28	28	28	**0**	0	69.05	100	556	3.03
18	esc16f	0	0	0	**0**	0	0	1	1	0.02
19	esc16g	26	26	28	**7.27**	7.27	69.23	300	695	6.88
20	esc16h	996	978	1100	**11.74**	9.92	31.90	1362	609	28.75
21	esc16i	14	12	14	**14.81**	0	101.96	1016	2044	25.15
22	esc16j	8	8	8	**0**	0	82.76	100	799	2.11
23	had12	1652	1652	1652	**0**	0	0	300	11600	3.92
24	had14	2724	2724	2724	**0**	0	0	400	20300	5.52
25	had16	3720	3720	3720	**0**	0	0	600	18100	12.28
26	had18	5358	5358	5358	**0**	0	0.02	1300	34700	40.66
27	had20	6922	6922	6922	**0**	0	0.13	2300	40000	106.96
28	nug12	578	568	728	**24.67**	22.95	27.86	1416	2884	15.70
29	nug14	1014	1012	1022	**0.98**	0.79	1.08	2832	19600	44.65
30	nug15	1150	1142	1280	**11.39**	10.70	16.33	2161	5812	40.45
31	nug16a	1610	1600	1610	**0.62**	0	0.62	6217	19300	138.71
32	nug16b	1240	1220	1258	**3.07**	1.44	25.41	3454	2347	80.00
33	nug17	1732	1708	1756	**2.77**	1.38	2.77	6194	6401	159.42
34	nug18	1930	1894	2022	**6.54**	4.65	12.84	9555	3988	285.40
35	nug20	2570	2508	2702	**7.45**	5.01	18.43	7065	2386	266.59
36	rou12	235528	235528	235528	**0**	0	0	3700	34200	35.98
37	rou15	354210	350216	360702	**2.95**	1.82	4.89	2531	3946	39.94
38	rou20	725522	695180	781532	**11.70**	7.43	14.93	7099	1538	281.71
39	scr12	31410	31410	31410	**0**	0	19.38	400	4268	3.93
40	scr15	51140	51140	51140	**0**	0	21.96	700	5489	12.48
41	scr20	110030	106804	132826	**21.72**	18.77	43.71	11599	9705	425.22
42	tai10a	135028	135028	135028	**0**	0	0.01	1200	21400	5.95
43	tai12a	224416	224416	224416	**0**	0	0	300	4300	2.68
44	tai15a	388214	377100	403890	**6.86**	3.96	9.03	2644	2245	39.96
45	tai17a	491812	476526	534328	**11.44**	8.29	16.25	2940	1399	64.67
46	tai20a	703482	671676	762166	**12.62**	8.01	19.03	3733	999	136.38

4.3 Comparisons to Other Methods

In this section we compare our results with two recent papers on relaxations for **QAP**.[7]

Comparison to C-SDP([7]) Here we compare our numerical result with the results presented by Ferreira et al. [7]. Briefly, Ferreira et al. [7] propose a semidefinite relaxation based algorithm C-SDP. The algorithm applies to relatively sparse data and hence their results are presented for chr

[7] For more comparisons, see e.g., [30, Table 4.1, Table 4.2] to view a complete list of lower bounds using bundle method presented in [33].

Table 4.2: QAPLIB Instances of Medium Size

Problem Data			Numerical Results					Timing		
#	name	true-opt	lbd	ubd	rel.gap	rel.opt.gap	rel.gap$^{\mathbf{A}}$	iter	iter$^{\mathbf{A}}$	time(sec)
47	chr22a	6156	6156	6156	**0**	0	0.02	11500	40000	613.03
48	chr22b	6194	6190	6194	**0.06**	0	0.11	13500	39300	673.22
49	chr25a	3796	3796	3796	**0**	0	0	6200	35600	450.22
50	esc32a	130	104	168	**46.89**	25.42	106.90	15100	12400	2553.03
51	esc32b	168	132	220	**49.86**	26.74	92.49	1000	4144	167.59
52	esc32c	642	616	642	**4.13**	0	23.23	2500	2052	418.83
53	esc32d	200	192	220	**13.56**	9.50	41.08	670	1430	117.00
54	esc32e	2	2	18	**152.38**	152.38	152.38	700	3086	112.26
55	esc32g	6	6	12	**63.16**	63.16	121.21	500	999	81.81
56	esc32h	438	426	452	**5.92**	3.14	30.14	6500	17600	1097.87
57	kra30a	88900	86838	96430	**10.47**	8.13	15.91	9898	3799	1319.97
58	kra30b	91420	87858	101640	**14.55**	10.59	28.84	5480	5017	750.38
59	kra32	88700	85776	93050	**8.14**	4.79	30.03	4959	4173	870.14
60	nug21	2438	2382	2644	**10.42**	8.11	12.36	6439	5729	274.09
61	nug22	3596	3530	3678	**4.11**	2.25	12.76	7279	7573	359.10
62	nug24	3488	3402	3770	**10.26**	7.77	16.25	4543	4447	294.82
63	nug25	3744	3626	3966	**8.96**	5.76	15.37	11687	7799	864.25
64	nug27	5234	5130	5496	**6.89**	4.88	17.08	10039	8609	1010.56
65	nug28	5166	5026	5676	**12.15**	9.41	18.55	8387	7533	943.84
66	nug30	6124	5950	6610	**10.51**	7.63	20.21	11321	9036	1581.33
67	ste36a	9526	9260	9980	**7.48**	4.65	42.28	19500	27300	5262.87
68	ste36b	15852	15668	15932	**1.67**	0.50	82.03	29000	40000	7889.04
69	ste36c	8239110	8134808	8394142	**3.14**	1.86	36.15	36499	40000	9819.15
70	tai25a	1167256	1096656	1264590	**14.22**	8.00	20.55	2264	999	164.11
71	tai30a	1818146	1706872	1984536	**15.04**	8.75	15.21	4550	1599	623.39
72	tai35a	2422002	2216646	2625284	**16.88**	8.06	22.34	3161	1599	777.17
73	tai40a	3139370	2843310	3455540	**19.44**	9.59	23.43	5577	2299	5546.57
74	tho30	149936	143576	166336	**14.69**	10.37	24.33	8321	7729	1122.28
75	tho40	240516	226522	257642	**12.86**	6.88	25.19	15535	12460	17832.61

Table 4.3: QAPLIB Instances of Large Size

Problem Data			Numerical Results					Timing		
#	name	true-opt	lbd	ubd	rel.gap	rel.opt.gap	rel.gap$^{\mathbf{A}}$	iter	iter$^{\mathbf{A}}$	time(sec)
76	esc64a	116	98	260	90.25	76.39	80.97	400	1200	1085.52
77	sko42*	15812	15336	16244	**5.75**	2.70	17.24	5511	10700	6245.96
78	sko49*	23386	22654	24406	**7.45**	4.27	16.87	9484	16900	12213.03
79	sko56*	34458	33390	36468	**8.81**	5.67	15.92	5792	15100	11669.07
80	sko64*	48498	47022	50762	**7.65**	4.56	16.15	10021	21100	23033.17
81	tai50a*	4938796	4390980	5517228	**22.73**	11.06	25.79	2331	3300	1238.71
82	tai60a*	7205962	6326344	7895180	**22.06**	9.13	26.03	3799	5100	4939.96
83	tai64c	1855928	1811354	1887500	**4.12**	1.69	38.79	800	2400	1461.00
84	wil50*	48816	48126	50834	**5.47**	4.05	9.37	5384	11000	2971.40

and esc families in QAPLIB. Figure 1 below illustrates the relative gaps arising from **rPRSM** and C-SDP. The numerics used in Figure 1 can be found in [7, Table 3-4]. The horizontal axis indicates the instance name on QAPLIB whereas the vertical axis indicates the relative gap[8]. Figure 1 illustrates that **rPRSM** yields much stronger relative gaps than C-SDP.

[8]We selected the best result given in [7, Table3, Table 4] for different parameters. We point out that [7] used a different formula for the gap computation. In this paper, we recomputed the relative gaps using (4.1) for a proper comparison. [7] used similar approach for upper bounds as in our paper, that is, the projection onto permutation matrices using [6, 37].

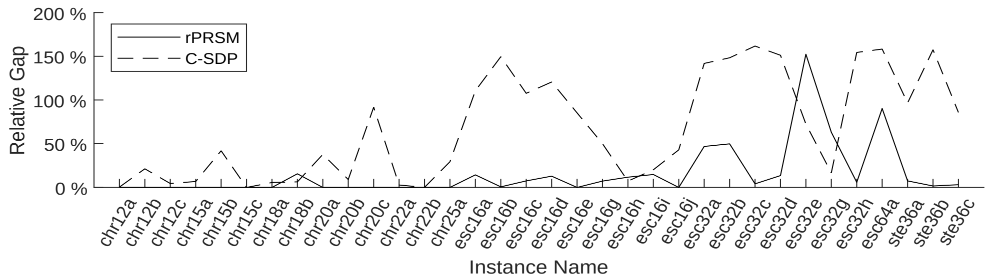
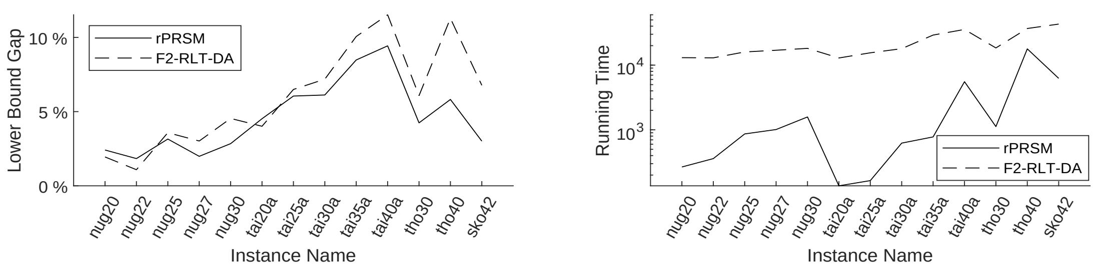


Figure 1: Relative Gap for **rPRSM** and C-SDP

Comparison to F2-RLT2-DA([12]) Date and Nagi [12] 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 [9] 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 [10] 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 even though the two methods give similar lower bounds to **QAP**, **rPRSM** is less time-consuming even considering the differences in the hardware[11].



(a) Lower Bound Gap

(b) Running Time

Figure 2: Numerical Comparison for **rPRSM** and F2-RLT2-DA

5 Conclusion

In this paper we introduce a strengthened splitting method for solving the facially reduced **DNN** relaxation for the **QAP**. That is, given constraints that are difficult to engage simultaneously, we

[9] 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.

[10] The running time for F2-RLT2-DA is obtained by using the average time per iteration presented in [12] multiplied by 2000 as F2-RLT2-DA runs the algorithm for 2000 iterations. The running time for **rPRSM** is drawn from Tables 4.1 to 4.3.

[11] 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.

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.

References

[1] A. Anjos and J. Lasserre, eds., *Handbook on Semidefinite, Conic and Polynomial Optimization*, International Series in Operations Research & Management Science, Springer-Verlag, 2011. 3

[2] K. Anstreicher and N. Brixius, *Solving quadratic assignment problems using convex quadratic programming relaxations*, tech. report, University of Iowa, Iowa City, IA, 2000. 3

[3] A. Barvinok and T. Stephen, *The distribution of values in the quadratic assignment problem*, Math. Oper. Res., 28 (2003), pp. 64–91. 3

[4] M. Bashiri and H. Karimi, *Effective heuristics and meta-heuristics for the quadratic assignment problem with tuned parameters and analytical comparisons*, Journal of Industrial Engineering International, 8 (2012), p. 6. 3

[5] R. Bhati and A. Rasool, *Quadratic assignment problem and its relevance to the real world: A survey*, International Journal of Computer Applications, 96 (2014), pp. 42–47. 3

[6] G. Birkoff, *Tres observaciones sobre el algebra lineal*, Univ. Nac. Tucuman Rev., Ser. A (1946), pp. 147–151. 7, 25

[7] J. Bravo Ferreira, Y. Khoo, and A. Singer, *Semidefinite programming approach for the quadratic assignment problem with a sparse graph*, Comput. Optim. Appl., 69 (2018), pp. 677–712. 21, 24, 25

[8] R. Burkard, S. Karisch, and F. Rendl, *QAPLIB – a quadratic assignment problem library*, European J. Oper. Res., 55 (1991), pp. 115–119. anjos.mgi.polymtl.ca/qaplib/. 4, 7

[9] E. Çela, *The quadratic assignment problem*, vol. 1 of Combinatorial Optimization, Kluwer Academic Publishers, Dordrecht, 1998. Theory and algorithms. 3

[10] Y. Chen and X. Ye, *Projection onto a simplex*, arXiv preprint arXiv:1101.6081, (2011). 18

[11] C. Commander, *A survey of the quadratic assignment problem, with applications*, PhD thesis, University of Florida, 2003. PhD Thesis. 3

[12] K. Date and R. Nagi, *Level 2 reformulation linearization technique-based parallel algorithms for solving large quadratic assignment problems on graphics processing unit clusters*, INFORMS J. Comput., 31 (2019), pp. 771–789. 21, 26

[13] Z. Drezner, *A new genetic algorithm for the quadratic assignment problem*, INFORMS J. Comput., 15 (2003), pp. 320–330. 3

[14] D. Drusvyatskiy and H. Wolkowicz, *The many faces of degeneracy in conic optimization*, Foundations and Trends® in Optimization, 3 (2017), pp. 77–170. 10

[15] C. S. Edwards, *The derivation of a greedy approximator for the koopmans-beckmann quadratic assignment problem*, in Proceedings CP77 Combinatorial Prog. Conf., Liverpool, 1977, pp. 55–86. 4

[16] A. Elshafei, *Hospital layout as a quadratic assignment problem*, Operations Research Quarterly, 28 (1977), pp. 167–179. 3

[17] L. Gambardella, É. Taillard, and M. Dorigo, *Ant colonies for the quadratic assignment problem*, Journal of the operational research society, 50 (1999), pp. 167–176. 3

[18] M. R. Garey and D. S. Johnson, *Computers and Intractability*, W. H. Freeman and Company, San Francisco, 1979. 3

[19] A. Geoffrion and G. Graves, *Scheduling parallel production lines with changeover costs: Practical applications of a quadratic assignment/LP approach*, Operations Research, 24 (1976), pp. 595–610. 3

[20] M. Goemans and D. Williamson, *Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming*, J. Assoc. Comput. Mach., 42(6) (1995), pp. 1115–1145. 21

[21] Y. Gu, B. Jiang, and D. Han, *A semi-proximal-based strictly contractive Peaceman-Rachford splitting method*, arXiv e-prints, (2015), p. arXiv:1506.02221. 15, 16

[22] P. Hahn, Y.-R. Zhu, M. Guignard, and J. Smith, *Exact solution of emerging quadratic assignment problems*, Int. Trans. Oper. Res., 17 (2010), pp. 525–552. 3

[23] B. He, H. Liu, Z. Wang, and X. Yuan, *A strictly contractive peaceman–rachford splitting method for convex programming*, SIAM Journal on Optimization, 24 (2014), pp. 1011–1040. 14, 15

[24] D. Heffley, *Assigning runners to a relay team*, in Optimal strategies in sports, vol. 5, North Holland Amsterdam, 1977, pp. 169–171. 3

[25] H. Hu, R. Sotirov, and H. Wolkowicz, *Facial reduction for symmetry reduced semidefinite programs*, 2019. 3

[26] J. Krarup and P. Pruzan, *Computer-aided layout design*, in Mathematical programming in use, Springer, 1978, pp. 75–94. 3

[27] X. Li, T. Pong, H. Sun, and H. Wolkowicz, *A strictly contractive Peaceman-Rachford splitting method for the doubly nonnegative relaxation of the minimum cut problem*, Comput. Optim. Appl., (2020), pp. accepted Dec. 23, 2020. 40 pages, research report. 3, 19

[28] X. Li and X. Yuan, *A proximal strictly contractive Peaceman-Rachford splitting method for convex programming with applications to imaging*, SIAM J. Imaging Sci., 8 (2015), pp. 1332–1365. 14, 15, 16

[29] J. Mitchell, P. Pardalos, and M. Resende, *Interior point methods for combinatorial optimization*, in Handbook of combinatorial optimization, Springer, 1998, pp. 189–297. 3

[30] D. Oliveira, H. Wolkowicz, and Y. Xu, *ADMM for the SDP relaxation of the QAP*, Math. Program. Comput., 10 (2018), pp. 631–658. 2, 3, 4, 6, 7, 10, 20, 21, 22, 23, 24

[31] P. Pardalos, F. Rendl, and H. Wolkowicz, *The quadratic assignment problem: a survey and recent developments*, in Quadratic assignment and related problems (New Brunswick, NJ, 1993), P. Pardalos and H. Wolkowicz, eds., Amer. Math. Soc., Providence, RI, 1994, pp. 1–42. 3

[32] P. Pardalos and H. Wolkowicz, eds., *Quadratic assignment and related problems*, American Mathematical Society, Providence, RI, 1994. Papers from the workshop held at Rutgers University, New Brunswick, New Jersey, May 20–21, 1993. 3

[33] F. Rendl and R. Sotirov, *Bounds for the quadratic assignment problem using the bundle method*, Math. Program., 109 (2007), pp. 505–524. 24

[34] R. Rockafellar, *Convex analysis*, Princeton Landmarks in Mathematics, Princeton University Press, Princeton, NJ, 1997. Reprint of the 1970 original, Princeton Paperbacks. 20

[35] I. Ugi, J. Bauer, J. Brandt, J. Friedrich, J. Gasteiger, C. Jochum, and W. Schubert, *Neue anwendungsgebiete für computer in der chemie*, Angewandte Chemie, 91 (1979), pp. 99–111. 3

[36] E. van den Berg and M. Friedlander, *Probing the Pareto frontier for basis pursuit solutions*, SIAM J. Sci. Comput., 31 (2008/09), pp. 890–912. 18

[37] J. von Neumann, *A certain zero-sum two-person game equivalent to the optimal assignment problem*, in Contributions to the theory of games, vol. 2, Annals of Mathematics Studies, no. 28, Princeton University Press, Princeton, N. J., 1953, pp. 5–12. 7, 25

[38] H. Wolkowicz, R. Saigal, and L. Vandenberghe, eds., *Handbook of semidefinite programming*, International Series in Operations Research & Management Science, 27, Kluwer Academic Publishers, Boston, MA, 2000. Theory, algorithms, and applications. 3

[39] Q. Zhao, S. Karisch, F. Rendl, and H. Wolkowicz, *Semidefinite programming relaxations for the quadratic assignment problem*, vol. 2, 1998, pp. 71–109. Semidefinite programming and interior-point approaches for combinatorial optimization problems (Toronto, ON, 1996). 3, 5, 7, 10, 11, 20, 22