ADMM for the SDP relaxation of the QAP

Danilo Elias Oliveira[1] Henry Wolkowicz[2] YangYang Xu[3]

[1]Department of Combinatorics and Optimization
University of Waterloo (visiting from UFU, Brazil)

[2]Department of Combinatorics and Optimization
University of Waterloo

[3]Institute for Mathematics and its Applications (IMA)
University of Minnesota

Tues. May 30: 香港浸會大學 HONG KONG BAPTIST UNIVERSITY

Outline

- Semidefinite programming, SDP, relaxations have proven to be extremely strong for many hard discrete optimization problems. This is particularly true for the quadratic assignment problem, QAP, one of the hardest NP-hard discrete optimization problems. (Instances size $n = 30$ are still unsolved.)

Outline

- Semidefinite programming, SDP, relaxations have proven to be extremely strong for many hard discrete optimization problems. This is particularly true for the quadratic assignment problem, QAP, one of the hardest NP-hard discrete optimization problems. (Instances size $n = 30$ are still unsolved.)
- difficulties: large dimensional relaxations; inefficiency of the current primal-dual interior point solvers for time and accuracy; high expense in adding cutting plane constraints.

Outline

- Semidefinite programming, SDP, relaxations have proven to be extremely strong for many hard discrete optimization problems. This is particularly true for the quadratic assignment problem, QAP, one of the hardest NP-hard discrete optimization problems. (Instances size $n = 30$ are still unsolved.)
- difficulties: large dimensional relaxations; inefficiency of the current primal-dual interior point solvers for time and accuracy; high expense in adding cutting plane constraints.
- we propose: alternating direction method of multipliers, ADMM , to solve the SDP relaxation along with nonnegativity constraints, i.e., solve the DNN relaxation.

Outline

- Semidefinite programming, SDP, relaxations have proven to be extremely strong for many hard discrete optimization problems. This is particularly true for the quadratic assignment problem, QAP, one of the hardest NP-hard discrete optimization problems. (Instances size $n = 30$ are still unsolved.)
- difficulties: large dimensional relaxations; inefficiency of the current primal-dual interior point solvers for time and accuracy; high expense in adding cutting plane constraints.
- we propose: alternating direction method of multipliers, ADMM , to solve the SDP relaxation along with nonnegativity constraints, i.e., solve the DNN relaxation.
- We exploit facial reduction, FR; it fits well with ADMM .

What is the QAP?

University planning: assign buildings to sites

The quadratic assignment problem, QAP, in the trace formulation

$$\text{(QAP)} \qquad p^* := \min_{X \in \Pi_n} \langle AXB - 2C, X \rangle,$$

$A, B \in \mathbb{S}^n$ real symmetric $n \times n$ matrices, C real $n \times n$,
$\langle \cdot\,, \cdot \rangle$ denotes trace inner product, $\langle Y, X \rangle = \operatorname{trace} YX^\top$,
and Π_n set of $n \times n$ permutation matrices

What is the QAP?

University planning: assign buildings to sites

The quadratic assignment problem, QAP, in the trace formulation

$$\text{(QAP)} \qquad p^* := \min_{X \in \Pi_n} \langle AXB - 2C, X \rangle,$$

$A, B \in \mathbb{S}^n$ real symmetric $n \times n$ matrices, C real $n \times n$,
$\langle \cdot , \cdot \rangle$ denotes trace inner product, $\langle Y, X \rangle = \operatorname{trace} YX^\top$,
and Π_n set of $n \times n$ permutation matrices

assign n facilities to n locations; minimize total cost

use flows in A between a pair of facilities
multiplied by distances in B between their assigned locations
and add location costs in $-\frac{1}{2}C$

Applications Include: (e.g., Nyberg et al '2012)

Koopmans-Beckmann 1957

- facility location planning: Universities, hospital layout, airport gate assignment, wiring problems/circuit boards, typewriters (though max?)
- Bandwith minimization of a graph
- Image processing
- Scheduling
- Supply Chains
- Economics
- Molecular conformations in chemistry
- Manufacturing lines
- Includes as special case: Traveling salesman problem and Maximum cut problem

New Derivation of FR , SDP Relax. in ZKRW , '98

Start new derivation with QQP

$$\begin{array}{lll}\min_X & \langle AXB - 2C, X\rangle & \\ \text{s.t.} & X_{ij}X_{ik} = 0,\ X_{ji}X_{ki} = 0, \forall i, \forall j \neq k, & \text{(gangster)} \\ & X_{ij}^2 - X_{ij} = 0, \forall i, j, & (0-1) \\ & \sum_{i=1}^n X_{ij}^2 - 1 = 0, \forall j,\ \sum_{j=1}^n X_{ij}^2 - 1 = 0, \forall i. & \text{(r-c sums)}\end{array}$$

Start new derivation with QQP

$$
\begin{array}{lll}
\min_X & \langle AXB - 2C, X \rangle & \\
\text{s.t.} & X_{ij}X_{ik} = 0, \ X_{ji}X_{ki} = 0, \ \forall i, \ \forall j \neq k, & \text{(gangster)} \\
& X_{ij}^2 - X_{ij} = 0, \ \forall i, j, & (0-1) \\
& \sum_{i=1}^n X_{ij}^2 - 1 = 0, \ \forall j, \ \sum_{j=1}^n X_{ij}^2 - 1 = 0, \ \forall i. & \text{(r-c sums)}
\end{array}
$$

Gangster constraints

The first set of constraints, the elementwise orthogonality of the row and columns of X, are the gangster constraints. They are particularly strong constraints and enable many of the other constraints (such as orthogonality $XX^T = I, X^TX = I$, row and columns sums are 1) to be redundant. In fact, after the FR, many of these constraints also become redundant.

The Lagrangian Dual

Lagrangian

$$
\begin{aligned}
\mathcal{L}_0(X, U, V, W, u, v) \quad = \quad & \langle AXB - 2C, X \rangle + \\
& \sum_{i=1}^{n} \sum_{j \neq k} U_{jk}^{(i)} X_{ij} X_{ik} + \\
& \sum_{i=1}^{n} \sum_{j \neq k} V_{jk}^{(i)} X_{ji} X_{ki} + \\
& \sum_{i,j} W_{ij} (X_{ij}^2 - X_{ij}) + \\
& \sum_{j=1}^{n} u_j \left(\sum_{i=1}^{n} X_{ij}^2 - 1 \right) + \\
& \sum_{i=1}^{n} v_i \left(\sum_{j=1}^{n} X_{ij}^2 - 1 \right) .
\end{aligned}
$$

The Lagrangian Dual

Lagrangian

$$\begin{aligned}\mathcal{L}_0(X,U,V,W,u,v) \;=\; & \langle AXB - 2C, X\rangle + \\ & \textstyle\sum_{i=1}^n \sum_{j\neq k} U_{jk}^{(i)} X_{ij} X_{ik} + \\ & \textstyle\sum_{i=1}^n \sum_{j\neq k} V_{jk}^{(i)} X_{ji} X_{ki} + \\ & \textstyle\sum_{i,j} W_{ij}(X_{ij}^2 - X_{ij}) + \\ & \textstyle\sum_{j=1}^n u_j \left(\sum_{i=1}^n X_{ij}^2 - 1\right) + \\ & \textstyle\sum_{i=1}^n v_i \left(\sum_{j=1}^n X_{ij}^2 - 1\right).\end{aligned}$$

Dual problem is maximization of dual functional d_0

$$\max\; d_0(U,V,W,u,v) := \min_X \mathcal{L}_0(X,U,V,W,u,v).$$

Simplify the Dual by Homogenization with x_0

add single constraint $x_0^2 = 1$, add dual variable w_0

$$\begin{aligned}\mathcal{L}_1(X, x_0, U, V, W, w_0, u, v) \quad = \quad & y^\top [L_Q + \mathcal{B}_1(U) + \mathcal{B}_2(V) + \\ & \mathrm{Arrow}(w, w_0) + \mathcal{K}_1(u) + \\ & \mathcal{K}_2(v)]\, y - e^\top (u + v) - w_0,\end{aligned}$$

where

$$\mathcal{K}_1(u) = \mathrm{blkdiag}(0, u \otimes I), \quad \mathcal{K}_2(v) = \mathrm{blkdiag}(0, I \otimes v)$$

$$\mathrm{Arrow}(w, w_0) = \begin{bmatrix} w_0 & -\frac{1}{2} w^\top \\ -\frac{1}{2} w & \mathrm{Diag}(w) \end{bmatrix}$$

$$\mathcal{B}_1(U) = \mathrm{blkdiag}(0, \tilde{U}), \quad \mathcal{B}_2(V) = \mathrm{blkdiag}(0, \tilde{V}).$$

And, $\tilde{U}$ and $\tilde{V}$ are $n \times n$ block matrices.

SDP Version of Lagrangian Dual

We let $L_Q = \begin{bmatrix} 0 & -\operatorname{vec}(C)^\top \\ -\operatorname{vec}(C) & B \otimes A \end{bmatrix}$.

$$\begin{aligned} \max \quad & -e^\top(u+v) - w_0 \\ \text{s.t.} \quad & L_Q + \mathcal{B}_1(U) + \mathcal{B}_2(V) + \operatorname{Arrow}(w, w_0) + \mathcal{K}_1(u) + \mathcal{K}_2(v) \succeq 0. \end{aligned}$$

SDP Version of Lagrangian Dual

We let $L_Q = \begin{bmatrix} 0 & -\operatorname{vec}(C)^\top \\ -\operatorname{vec}(C) & B \otimes A \end{bmatrix}$.

$$\begin{array}{ll} \max & -e^\top(u+v) - w_0 \\ \text{s.t.} & L_Q + \mathcal{B}_1(U) + \mathcal{B}_2(V) + \operatorname{Arrow}(w, w_0) + \mathcal{K}_1(u) + \mathcal{K}_2(v) \succeq 0. \end{array}$$

Now take the dual of the dual to get SDP relaxation.

Simplified SDP Relaxation

Dual of Dual

$$(SDP_R) \qquad \begin{array}{rl} p_R^* := \min_R & \langle L_Q, \hat{V} R \hat{V}^\top \rangle \\ \text{s.t.} & \mathcal{G}_J(\hat{V} R \hat{V}^\top) = E_{00} \\ & R \succeq 0, \end{array}$$

- Gangster operator $\mathcal{G}$ *shoots* holes in the matrix $\hat{V} R \hat{V}^\top$.
- J is the index set that guarantees the diagonal blocks are diagonal and the off-diagonal blocks have zero diagonal.
- (Some of these block constraints are redundant as are the previous block constraints.)
- with V (now) providing an orthonormal basis for $e^\perp$ and

$$\hat{V} = \begin{bmatrix} 1 & 0 \\ \frac{1}{n} e & V \otimes V \end{bmatrix} \quad \text{(does FR)}$$

Explicit Primal-Dual Strictly Feasible Points

Lemma (ZKRW , Explicit Primal Strictly Feasible Point)

The matrix $\hat{R}$ defined by

$$\hat{R} := \left[\begin{array}{c|c} 1 & 0 \\ \hline 0 & \frac{1}{n^2(n-1)}\left(nI_{n-1} - E_{n-1}\right) \otimes \left(nI_{n-1} - E_{n-1}\right) \end{array}\right] \in \mathcal{S}_{++}^{(n-1)^2+1}$$

is (strictly) feasible for (SDP_R). □

Dual

Gangster Operator is Self-Adjoint, $\mathcal{G}_J^* = \mathcal{G}_J$

Dual

Gangster Operator is Self-Adjoint, $\mathcal{G}_J^* = \mathcal{G}_J$

Dual Program

$$\begin{array}{rll} d_Y^* := \max\limits_{Y} & \langle E_{00}, Y \rangle & (= Y_{00}) \\ \text{s.t.} & \hat{V}^\top \mathcal{G}_J(Y) \hat{V} \preceq \hat{V}^\top L_Q \hat{V} & \end{array}$$

Dual

Gangster Operator is Self-Adjoint, $\mathcal{G}_J^* = \mathcal{G}_J$

Dual Program

$$d_Y^* := \max_Y \quad \langle E_{00}, Y \rangle \qquad (= Y_{00})$$
$$\text{s.t.} \quad \hat{V}^\top \mathcal{G}_J(Y) \hat{V} \preceq \hat{V}^\top L_Q \hat{V}$$

Lemma (ZKRW , Explicit Dual Strictly Feasible Point)

The matrices $\hat{Y}, \hat{Z}$, with $M > 0$ sufficiently large,

$$\hat{Y} := M \left[\begin{array}{c|c} n & 0 \\ \hline 0 & I_n \otimes (I_n - E_n) \end{array} \right] \in \mathcal{S}_{++}^{(n-1)^2+1},$$

$$\hat{Z} := \hat{V}^\top L_Q \hat{V} - \hat{V}^\top \mathcal{G}_J(\hat{Y}) \hat{V} \in \mathcal{S}_{++}^{(n-1)^2+1},$$

are (strictly) feasible and slack variables for the dual, respectively.

New ADMM Algorithm for the SDP Relaxation

KEEP! both Y and R

rewrite SDP_R equivalently as

$$\min_{R,Y} \left\{ \langle L_Q, Y \rangle \text{ s.t. } \mathcal{G}_J(Y) = E_{00},\ Y = \hat{V} R \hat{V}^\top,\ R \succeq 0 \right\}$$

Therefore we can work with the augmented Lagrange

$$\mathcal{L}_A(R, Y, Z) = \langle L_Q, Y \rangle + \langle Z, Y - \hat{V} R \hat{V}^\top \rangle + \frac{\beta}{2} \| Y - \hat{V} R \hat{V}^\top \|_F^2.$$

New ADMM Algorithm for the SDP Relaxation

KEEP! both Y and R

rewrite SDP_R equivalently as

$$\min_{R,Y} \left\{ \langle L_Q, Y \rangle \text{ s.t. } \mathcal{G}_J(Y) = E_{00},\ Y = \hat{V} R \hat{V}^\top,\ R \succeq 0 \right\}$$

Therefore we can work with the augmented Lagrange

$$\mathcal{L}_A(R, Y, Z) = \langle L_Q, Y \rangle + \langle Z, Y - \hat{V} R \hat{V}^\top \rangle + \frac{\beta}{2} \| Y - \hat{V} R \hat{V}^\top \|_F^2.$$

(R, Y, Z) are the primal reduced, primal, and dual variables, and this denotes the current iterate

$\mathbb{S}^n_{r+}$ denotes the matrices in $\mathbb{S}^n_+$ with rank at most r.

ADMM with Augmented Lagrangian

Updates for (R_+, Y_+, Z_+):

1. $R_+ = \text{argmin}_{R \in \mathbb{S}^n_{r+}} \mathcal{L}_A(R, Y, Z)$
2. $Y_+ = \text{argmin}_{Y \in \mathcal{P}_i} \mathcal{L}_A(R_+, Y, Z)$
3. $Z_+ = Z + \gamma \cdot \beta(Y_+ - \hat{V} R_+ \hat{V}^\top)$

Polyhedral Sets

$\mathcal{P}_1 = \{Y \in \mathbb{S}^{n^2+1} : \mathcal{G}_J(Y) = E_{00}\}$ gangster constraints.
$\mathcal{P}_2 = \mathcal{P}_1 \cap \{0 \leq Y \leq 1\}$ (polytope with nonnegativity)

1. Explicit solution for R

Let $\hat{V}$ be normalized such that $\hat{V}^\top \hat{V} = I$. Then:

$$\begin{aligned} R_+ &= \operatorname{argmin}_{R \succeq 0} \langle Z, Y - \hat{V} R \hat{V}^\top \rangle + \frac{\beta}{2} \| Y - \hat{V} R \hat{V}^\top \|_F^2 \\ &= \operatorname{argmin}_{R \succeq 0} \left\| Y - \hat{V} R \hat{V}^\top + \frac{1}{\beta} Z \right\|_F^2 \\ &= \operatorname{argmin}_{R \succeq 0} \left\| R - \hat{V}^\top \left(Y + \frac{1}{\beta} Z \right) \hat{V} \right\|_F^2 \\ &= \mathcal{P}_{\mathbb{S}_+^{(n-1)^2+1}} \left(\hat{V}^\top \left(Y + \frac{1}{\beta} Z \right) \hat{V} \right), \end{aligned}$$

where we apply the Eckart-Young-Mirsky Theorem and project onto the face of the SDP cone of desired rank ($\leq$ number of positive eigenvalues of the argument).

2. Explicit solution for Y

$i = 1$, first linear constraint, Y-subproblem, closed-form solution

$$\begin{aligned} Y_+ &= \operatorname{argmin}_{\mathcal{G}_J(Y)=E_{00}} \langle L_Q, Y\rangle + \langle Z, Y - \hat{V}R_+\hat{V}^\top\rangle + \\ &\qquad \tfrac{\beta}{2}\|Y - \hat{V}R_+\hat{V}^\top\|_F^2 \\ &= \operatorname{argmin}_{\mathcal{G}_J(Y)=E_{00}} \left\| Y - \hat{V}R_+\hat{V}^\top + \tfrac{L_Q+Z}{\beta} \right\|_F^2 \\ &= E_{00} + \mathcal{G}_{J^c}\left(\hat{V}R_+\hat{V}^\top - \tfrac{L_Q+Z}{\beta}\right). \end{aligned}$$

2. Explicit solution for Y

$i=1$, first linear constraint, Y-subproblem, closed-form solution

$$\begin{aligned} Y_+ &= \operatorname{argmin}_{\mathcal{G}_J(Y)=E_{00}} \langle L_Q, Y\rangle + \langle Z, Y - \hat{V}R_+\hat{V}^\top\rangle + \\ &\qquad \tfrac{\beta}{2}\|Y - \hat{V}R_+\hat{V}^\top\|_F^2 \\ &= \operatorname{argmin}_{\mathcal{G}_J(Y)=E_{00}} \left\| Y - \hat{V}R_+\hat{V}^\top + \frac{L_Q+Z}{\beta}\right\|_F^2 \\ &= E_{00} + \mathcal{G}_{J^c}\left(\hat{V}R_+\hat{V}^\top - \frac{L_Q+Z}{\beta}\right). \end{aligned}$$

major advantage of using ADMM : we can easily add $0 \leq \hat{V}R\hat{V}^\top \leq 1$ to solve the DNN!:

$$p^*_{RY} := \min_{R,Y}\{\langle L_Q, Y\rangle : \mathcal{G}_J(Y) = E_{00},\ 0 \leq Y \leq 1,\ Y = \hat{V}R\hat{V}^\top,\ R \succeq 0\}$$

Update Y_+ Becomes:

$$Y_+ = E_{00} + \min\left(1, \max\left(0, \mathcal{G}_{J^c}\left(\hat{V}R_+\hat{V}^\top - \frac{L_Q+Z}{\beta}\right)\right)\right)$$

Lower bound from Inaccurate Solutions

$(R^{out}, Y^{out}, Z^{out})$ output

Lemma

Let $\mathcal{R} := \{R \succeq 0\}, \quad \mathcal{Y} := \{Y : \mathcal{G}_J(Y) = E_{00},\ 0 \le Y \le 1\}$
$\mathcal{Z} := \{Z : \hat{V}^\top Z \hat{V} \preceq 0\}$ *and*
$g(Z) := \min_{Y \in \mathcal{Y}} \{\langle L_Q + Z, Y \rangle\}$, *be the* ADMM *dual function.*
Then the dual of ADMM *satisfies weak duality and is:*

$$\begin{aligned} d_Z^* &:= \max_{Z \in \mathcal{Z}} g(Z) \\ &\le p_R^*. \end{aligned}$$

Proof

Proof.

The dual problem can be derived as

$$
\begin{aligned}
d_Z^* := & \max_{Z} \min_{R\in\mathcal{R}, Y\in\mathcal{Y}} \langle L_Q, Y\rangle + \langle Z, Y - \hat{V}R\hat{V}^\top\rangle \\
= & \max_{Z} \min_{Y\in\mathcal{Y}} \langle L_Q, Y\rangle + \langle Z, Y\rangle + \min_{R\in\mathcal{R}} \langle Z, -\hat{V}R\hat{V}^\top\rangle \\
= & \max_{Z} \min_{Y\in\mathcal{Y}} \langle L_Q, Y\rangle + \langle Z, Y\rangle + \min_{R\in\mathcal{R}} \langle \hat{V}^\top Z\hat{V}, -R\rangle \\
= & \max_{Z\in\mathcal{Z}} \min_{Y\in\mathcal{Y}} \langle L_Q + Z, Y\rangle, \\
= & \max_{Z\in\mathcal{Z}} g(Z)
\end{aligned}
$$

Weak duality follows by exchanging the max and min. □

Implementation: Lower bound from Inaccurate Solutions

- $Z \in \mathcal{Z} \implies g(Z)$ is a lower bound; therefore use projection $g\big(\mathcal{P}_{\mathcal{Z}}(Z^{out})\big)$ as lower bound,

Implementation: Lower bound from Inaccurate Solutions

- $Z \in \mathcal{Z} \implies g(Z)$ is a lower bound; therefore use projection $g\big(\mathcal{P}_{\mathcal{Z}}(Z^{out})\big)$ as lower bound,
- to get $\mathcal{P}_{\mathcal{Z}}(\tilde{Z})$: Let $\bar{V} = (\hat{V}, \hat{V}_{\perp})$ be an orthogonal matrix; let $\bar{V}^\top Z \bar{V} = W = \begin{bmatrix} W_{11} & W_{12} \\ W_{21} & W_{22} \end{bmatrix}$; Then:

$$\hat{V}^\top Z \hat{V} \preceq 0 \Leftrightarrow \hat{V}^\top Z \hat{V} = \hat{V}^\top \bar{V} W \bar{V}^\top \hat{V} = W_{11} \preceq 0.$$

Hence,

$$\begin{aligned} \mathcal{P}_{\mathcal{Z}}(\tilde{Z}) &= \text{argmin}_{Z \in \mathcal{Z}} \|Z - \tilde{Z}\|_F^2 \\ &= \text{argmin}_{W_{11} \preceq 0} \|\bar{V} W \bar{V}^\top - \tilde{Z}\|_F^2 \\ &= \text{argmin}_{W_{11} \preceq 0} \|W - \bar{V}^\top \tilde{Z} \bar{V}\|_F^2 \\ &= \begin{bmatrix} \mathcal{P}_{\mathcal{S}_-}(\tilde{W}_{11}) & \tilde{W}_{12} \\ \tilde{W}_{21} & \tilde{W}_{22} \end{bmatrix}, \end{aligned}$$

Upper Bound from Feasible Solution

$(R^{out}, Y^{out}, Z^{out})$ output of ADMM

- obtain best rank-one approximation of Y from largest eigenvalue and corresponding eigenvector: $\lambda vv^\top$.

Upper Bound from Feasible Solution

$(R^{out}, Y^{out}, Z^{out})$ output of ADMM

- obtain best rank-one approximation of Y from largest eigenvalue and corresponding eigenvector: $\lambda v v^\top$.
- Reshape to get square matrix X^{out} as an approximate optimal permutation matrix.

Upper Bound from Feasible Solution

$(R^{out}, Y^{out}, Z^{out})$ output of ADMM

- obtain best rank-one approximation of Y from largest eigenvalue and corresponding eigenvector: $\lambda vv^\top$.
- Reshape to get square matrix X^{out} as an approximate optimal permutation matrix.
- Since X permutation matrix implies $\operatorname{trace} X^T X = n$, a constant, we get

$$\|X^{out} - X\|_F^2 = -2 \operatorname{trace} X^T X^{out} + \text{ constant.}$$

Take advantage of the Birkoff, von Neumann Theorem: permutation matrices are extreme points of the doubly stochastic matrices.

Upper Bound from Feasible Solution

$(R^{out}, Y^{out}, Z^{out})$ output of ADMM

- obtain best rank-one approximation of Y from largest eigenvalue and corresponding eigenvector: $\lambda vv^\top$.
- Reshape to get square matrix X^{out} as an approximate optimal permutation matrix.
- Since X permutation matrix implies $\operatorname{trace} X^T X = n$, a constant, we get

$$\|X^{out} - X\|_F^2 = -2 \operatorname{trace} X^T X^{out} + \text{ constant.}$$

  Take advantage of the Birkoff, von Neumann Theorem: permutation matrices are extreme points of the doubly stochastic matrices.
- Solve the linear program

$$\max_X \left\{ \langle X^{out}, X \rangle : Xe = e,\ X^\top e = e,\ X \geq 0 \right\}$$

Low Rank Solutions

Cheat

Project R onto a rank-one matrix,

$$R_+ = \mathcal{P}_{\mathcal{S}_+^{(n-1)^2+1} \cap \mathcal{R}_1} \left(\hat{V}^\top (Y + \frac{Z}{\beta}) \hat{V} \right),$$

where $\mathcal{R}_1 = \{R : \text{rank}(R) = 1\}$ denotes the set of rank-one matrices. For a symmetric matrix W with largest eigenvalue $\lambda > 0$ and corresponding eigenvector w, we have

$$\mathcal{P}_{\mathcal{S}_+^{(n-1)^2+1} \cap \mathcal{R}_1} = \lambda w w^\top.$$

Often provides better feasible solutions/upper bounds.
Proved optimality in 6 instances.

Different choices for $V, \widehat{V}$

matrix $\widehat{V}$ is essential; sparse $\widehat{V}$ helps in projection using a sparse eigenvalue code. From several, the most successful (from Hao, Wang, Pong, W. '14):

$$V = \left[\left[\begin{bmatrix} I_{\lfloor \frac{n}{2} \rfloor} \otimes \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ -1 \end{bmatrix} \end{bmatrix} \atop 0_{(n-2\lfloor \frac{n}{2} \rfloor), \lfloor \frac{n}{2} \rfloor} \right] \left[\begin{bmatrix} I_{\lfloor \frac{n}{4} \rfloor} \otimes \frac{1}{2} \begin{bmatrix} 1 \\ 1 \\ -1 \\ -1 \end{bmatrix} \end{bmatrix} \atop 0_{(n-4\lfloor \frac{n}{4} \rfloor), \lfloor \frac{n}{4} \rfloor} \right] [\ldots] \left[\widehat{V} \right] \right]_{n \times n-1}$$

i.e., the block matrix consisting of t blocks formed from Kronecker products along with one block $\widehat{V}$ to complete the appropriate size so that $V^\top V = I_{n-1}$, $V^\top e = 0$.

Numerical Tests

We used MATLAB version 8.6.0.267246 (R2015b) on a PC Dell Optiplex 9020 64-bit, with 16 Gig, running Windows 7. We heuristically set $\gamma = 1.618$ and $\beta = \frac{n}{3}$ in ADMM. We tested two different stopping tolerances $1e{-}12$ and $1e{-}5$.

We tested all QAP symmetric instances from QAPLIB with size up to $n = 100$ and compared with Rendl,Sotirov 06 and a standard p-d i-p approach.

QAPLIB: I, bounds

0. Instance name	1. opt value	2. Bundle [6] LowBnd	3. HKM-FR LowBnd	4. ADMM LowBnd	5. feas UpBnd	6. ADMM %gap	7. ADMM vs Bundle %Impr LowBnd
Esc16a	68	59	50	64	70	8.82	7.35
Esc16b	292	288	276	290	294	1.37	0.68
Esc16c	160	142	132	154	192	23.75	7.50
Esc16d	16	8	-12	13	18	31.25	31.25
Esc16e	28	23	13	27	32	17.86	14.29
Esc16g	26	20	11	25	38	50.00	19.23
Esc16h	996	970	909	977	1064	8.73	0.70
Esc16i	14	9	-21	12	20	57.14	21.43
Esc16j	8	7	-4	8	12	50.00	12.50
Had12	1652	1643	1641	1652	1652	0.00	0.54
Had14	2724	2715	2709	2724	2724	0.00	0.33
Had16	3720	3699	3678	3720	3720	0.00	0.56
Had18	5358	5317	5287	5358	5358	0.00	0.77
Had20	6922	6885	6848	6922	6930	0.12	0.53
Kra30a	88900	77647	-1111	86838	105650	21.16	10.34
Kra30b	91420	81156	-1111	87858	102370	15.87	7.33
Kra32	88700	79659	-1111	85773	103070	19.50	6.89
Nug12	578	557	530	568	686	20.42	1.90
Nug14	1014	992	960	1011	1022	1.08	1.87
Nug15	1150	1122	1071	1141	1332	16.61	1.65
Nug16a	1610	1570	1528	1600	1610	0.62	1.86
Nug16b	1240	1188	1139	1219	1366	11.85	2.50
Nug17	1732	1669	1622	1708	1756	2.77	2.25
Nug18	1930	1852	1802	1894	2160	13.78	2.18
Nug20	2570	2451	2386	2507	2754	9.61	2.18
Nug21	2438	2323	2386	2382	2748	15.01	2.42
Nug22	3596	3440	3396	3529	3860	9.20	2.47
Nug24	3488	3310	-1111	3402	3842	12.61	2.64
Nug25	3744	3535	-1111	3626	4074	11.97	2.43
Nug27	5234	4965	-1111	5130	5788	12.57	3.15
Nug28	5166	4901	-1111	5026	5492	9.02	2.42
Nug30	6124	5803	-1111	5950	6720	12.57	2.40
Rou12	235528	223680	221161	235528	235528	0.00	5.03
Rou15	354210	333287	323235	350217	367782	4.96	4.78
Rou20	725522	663833	642856	695181	765390	9.68	4.32
Scr12	31410	29321	23973	31410	45414	44.58	6.65
Scr15	51140	48836	42204	51140	55760	9.03	4.51
Scr20	110030	94998	83302	106801	124522	16.11	10.73
Tai12a	224416	222784	215637	224416	224416	0.00	0.73
Tai15a	388214	364761	349586	377101	412760	9.19	3.18
Tai17a	491812	451317	441294	476525	546366	14.20	5.13
Tai20a	703482	637300	619092	671675	750450	11.20	4.89
Tai25a	1167256	1041337	-1111	1096657	1271696	15.00	4.74
∗Tai30a	1818146	1652186	-1111	1706871	1942086	12.94	3.01
Tho30	149936	136059	-1111	143576	162882	12.88	5.01

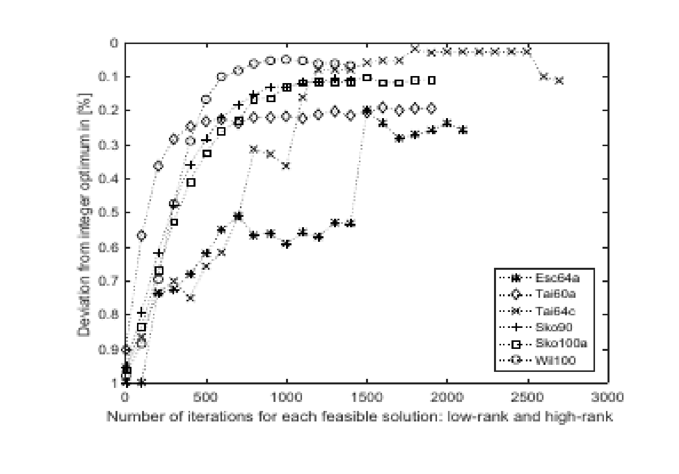
QAPLIB: I, times/iters

0 Instance name	1 Tol5 cpusec HighRk	2 Tol5 cpusec LowRk	3 Tol12/5 cpuratio HighRk	4 HKM cpuratio Tol 9	5 ADMM iterations HighRk	6 ADMM iterations LowRk
Esc16a	2.99e+01	3.49	3.72	10.78	2053	289
Esc16b	5.07e+00	3.54	1.97	10.39	338	311
Esc16c	1.41e+01	4.15	4.09	9.51	961	359
Esc16d	2.77e+01	2.62	4.35	14.37	1889	230
Esc16e	3.92e+01	3.49	4.60	10.81	2719	297
Esc16g	5.57e+01	3.49	2.49	10.56	3839	284
Esc16h	6.42e+00	3.07	2.08	12.18	433	271
Esc16i	1.75e+02	3.27	2.46	11.54	11653	276
Esc16j	1.09e+02	3.57	4.10	10.68	6898	296
Had12	1.12e+01	0.66	1.00	9.57	2682	157
Had14	3.56e+01	1.51	1.17	11.72	3919	169
Had16	1.87e+02	2.40	1.00	16.44	14179	210
Had18	4.73e+02	5.18	2.19	15.38	18071	259
Had20	4.47e+02	11.20	4.28	15.77	9038	309
Kra30a	3.04e+03	154.97	4.72	-1111	8466	603
Kra30b	4.61e+03	159.58	3.12	-1111	12882	627
Kra32	1.74e+04	204.38	0.99	-1111	40000	702
Nug12	2.67e+01	0.73	6.76	8.50	5813	150
Nug14	7.35e+01	1.49	4.96	10.61	7667	167
Nug15	9.40e+01	2.44	5.94	10.57	6547	204
Nug16a	1.82e+02	2.24	3.32	16.73	11591	193
Nug16b	9.66e+01	2.22	6.27	16.96	6433	184
Nug17	2.32e+02	3.46	3.61	16.47	10727	204
Nug18	4.28e+02	4.50	2.52	18.48	15862	226
Nug20	5.10e+02	9.40	4.10	17.65	9786	261
Nug21	1.53e+03	15.28	1.81	15.00	22465	309
Nug22	2.29e+03	20.56	1.44	14.75	27842	339
Nug24	1.37e+03	28.78	3.30	-1111	12150	368
Nug25	3.50e+03	39.06	1.68	-1111	24051	376
Nug27	5.70e+03	74.12	1.59	-1111	25205	444
Nug28	4.51e+03	78.39	2.17	-1111	18417	448
Nug30	8.05e+03	132.01	1.77	-1111	22614	507
Rou12	2.74e+01	0.61	1.04	10.42	6327	127
Rou15	3.21e+01	2.16	8.68	11.80	2219	170
Rou20	1.96e+02	9.63	10.84	17.22	3684	263
Scr12	4.56e+00	0.72	2.48	9.48	1135	150
Scr15	1.46e+01	1.77	1.80	14.66	1061	148
Scr20	2.07e+03	8.90	1.02	19.38	40000	243
Tai12a	1.83e+00	0.61	1.06	9.77	421	127
Tai15a	2.86e+01	1.95	14.74	12.51	1955	157
Tai17a	6.57e+01	3.64	7.44	14.94	2997	216
Tai20a	1.48e+02	9.44	14.52	16.95	2755	252
Tai25a	3.33e+02	36.63	5.66	-1111	2244	350
Tai30a	1.34e+03	135.63	10.83	-1111	3698	527
Tho30	6.39e+03	128.40	2.22	-1111	17858	487

QAPLIB II

	1. opt value	2. ADMM LowBnd	3. feas UpBnd	4. ADMM %gap	5 Tol5 cpusec HighRk	6 Tol5 cpusec LowRk	7 ADMM iterations HighRk	8 ADMM iterations LowRk
Chr12a	9552	9552	9552	0.00	8.22e+01	5.44e-01	21061	117
Chr12b	9742	9742	9742	0.00	4.17e+01	4.92e-01	10592	119
Chr12c	11156	11156	11156	0.00	9.40e+01	5.25e-01	23982	115
Chr15a	9896	9896	9896	0.00	4.00e+02	2.04e+00	31937	173
Chr15b	7990	7990	7990	0.00	5.20e+01	1.76e+00	3976	133
Chr15c	9504	9504	9504	0.00	2.98e+01	1.71e+00	2192	147
Chr18a	11098	11098	11098	0.00	1.03e+03	3.95e+00	40000	198
Chr18b	1534	1534	2420	57.76	1.04e+02	5.12e+00	3843	255
Chr20a	2192	2192	2192	0.00	1.92e+03	7.73e+00	40000	217
Chr20b	2298	2298	2298	0.00	3.14e+02	8.91e+00	6355	243
Chr20c	14142	14139	14142	0.02	1.89e+03	8.28e+00	40000	232
Chr22a	6156	6156	6156	0.00	1.09e+03	1.76e+01	14051	310
Chr22b	6194	6194	6194	0.00	9.35e+02	1.78e+01	11418	304
Chr25a	3796	3796	3796	0.00	8.48e+02	3.62e+01	6164	355
Els19	17212548	17209785	17212548	0.02	1.52e+03	8.44e+00	40000	269
Esc16f	0	0	0	0	3.00e+02	2.84e+02	40000	40000
Esc32a	130	104	244	107.69	8.79e+03	2.24e+02	20514	765
Esc32b	168	132	244	66.67	7.57e+03	2.05e+02	17926	728
Esc32c	642	616	730	17.76	1.34e+03	2.25e+02	3177	779
Esc32d	200	191	242	25.50	2.66e+03	2.45e+02	6334	843
Esc32e	2	2	2	0.00	5.48e+03	2.35e+02	13040	849
Esc32f	2	2	2	0.00	5.53e+03	2.35e+02	13040	849
Esc32g	6	6	8	33.33	1.85e+03	2.34e+02	4405	831
Esc32h	438	425	496	16.21	9.17e+03	2.29e+02	21527	790
*Sko42	15812	15335	17022	10.67	3.72e+04	1.13e+03	21017	994
*Sko49	23386	22653	25290	11.28	1.13e+05	2.93e+03	28784	1333
*Sko56	34458	33389	37324	11.42	3.07e+05	6.50e+03	40000	1709
Ste36a	9526	9259	12906	38.28	3.13e+04	4.13e+02	40000	798
Ste36b	15852	15668	32552	106.51	3.12e+04	3.98e+02	40000	770
Ste36c	8239110	8134716	10396856	27.46	3.13e+04	3.96e+02	40000	764
*Tai35a	2422002	2216645	2599924	15.82	2.41e+03	3.48e+02	3225	661
*Tai40a	3139370	2843311	3392692	17.50	6.04e+03	6.93e+02	4665	852
*Tai50a	4938796	4390976	5319048	18.79	2.31e+04	3.15e+03	5393	1343
*Tho40	240516	226522	261744	14.64	2.70e+04	6.63e+02	21138	799
*Wil50	48816	48125	50394	4.65	6.55e+04	3.39e+03	15377	1419
Esc64a	116	98	132	29.31	3.14e+04	1.31e+04	2141	2200
*Sko64	48498	46711	51716	10.32	1.85e+04	1.22e+04	1500	1500
*Sko72	66256	63896	70816	10.44	3.66e+04	2.22e+04	1500	1500
*Sko81	90998	86968	99276	13.53	9.46e+04	4.07e+04	1500	1500
*Sko90	115534	109843	124212	12.44	6.39e+05	7.15e+04	1500	1500
*Sko100a	152002	145775	164050	12.02	1.99e+05	1.31e+05	2000	2000
*Sko100b	153890	147332	166198	12.26	2.00e+05	1.31e+05	2000	2000
*Sko100c	147862	142018	161872	13.43	1.98e+05	1.31e+05	2000	2000
*Sko100d	149576	143205	159984	11.22	3.78e+05	1.37e+05	2000	2000
*Sko100e	149150	142977	162406	13.03	4.89e+05	1.37e+05	2000	2000
*Sko100f	149036	142413	161138	12.56	2.10e+05	1.32e+05	2000	2000
*Tai60a	7205962	6319630	7809204	20.67	2.24e+04	1.03e+04	2000	1964
Tai64c	1855928	1811312	2042710	12.47	4.26e+04	1.71e+04	2842	2856
*Tai80a	13499184	11652111	14395938	20.33	2.01e+05	6.03e+04	4000	3731
*Tai100	21052466	17390716	22641778	24.94	9.89e+04	6.28e+04	1000	1000
*Wil100	273038	265623	287398	7.98	1.50e+05	9.85e+04	1500	1500

Figure: Relative gap versus iterations for large instances



Conclusion

- We presented ADMM framework for QAP that exploits facial reduction. (Keeping both R and Y appears to be advantageous.)
- solve large problems to extremely high accuracy while solving the DNN relaxation. This yielded improved bounds. Several problems were solved to optimality.
- The ADMM approach together with facial reduction appears to be very promising; success for graph partitioning, second lifting of max-cut, and a special case of the min-cut problem.

Thanks for your attention!

ADMM for the SDP relaxation of the QAP

Danilo Elias Oliveira[1] Henry Wolkowicz[2] YangYang Xu[3]

[1]Department of Combinatorics and Optimization
University of Waterloo (visiting from UFU, Brazil)

[2]Department of Combinatorics and Optimization
University of Waterloo

[3]Institute for Mathematics and its Applications (IMA)
University of Minnesota

Tues. May 30: 香港浸會大學 HONG KONG BAPTIST UNIVERSITY

References

- R.E. Burkard, S. Karisch, and F. Rendl, *QAPLIB – a quadratic assignment problem library*, European J. Oper. Res. 55 (1991), 115–119, anjos.mgi.polymtl.ca/qaplib/.
- R.E. Burkard, S.E. Karisch, and F. Rendl, *QAPLIB—a quadratic assignment problem library*, J. Global Optim. 10 (1997), no. 4, 391–403. MR 1457185
- T.C. Koopmans and M.J. Beckmann, *Assignment problems and the location of economic activities*, Econometrica 25 (1957), 53–76.
- Axel Nyberg and Tapio Westerlund, *A new exact discrete linear reformulation of the quadratic assignment problem*, European Journal of Operational Research 220 (2012), no. 2, 314–319.
- T.K. Pong, H. Sun, N. Wang, and H. Wolkowicz, *Eigenvalue, quadratic programming, and semidefinite programming relaxations for a cut minimization problem*, Comput. Optim. Appl. 63 (2016), no. 2, 333–364. MR 3457444
- F. Rendl and R. Sotirov, *Bounds for the quadratic assignment problem using the bundle method*, Math. Program. 109 (2007), no. 2-3, Ser. B, 505–524. MR MR2295153 (2008c:90043)
- Q. Zhao, S.E. Karisch, F. Rendl, and H. Wolkowicz, *Semidefinite programming relaxations for the quadratic assignment problem*, J. Comb. Optim. 2 (1998), no. 1, 71–109, Semidefinite programming and interior-point approaches for combinatorial optimization problems (Fields Institute, Toronto, ON, 1996). MR 99f:90103