Verification Methods for Ill-Conditioned Relaxation Frameworks of Quadratic Assignment Problems.

Marko Lange

Hamburg University of Technology

October 16, 2012

Outline

Outline

Outline

Outline

What is a QAP?

The quadratic assignment problem (**QAP**) was first introduced by Koopmans and Beckmann [9, 1957] to solve a facility location problem.

The corresponding weight/flows and distances can be represented via **square real-valued matrices**. A convenient form for the QAP is:

$$\min_{X \in \Pi} \operatorname{Tr}(AXBX^T) , \tag{1}$$

where Π denotes the set of permutation matrices.

Some Applications

- Hospital Layout as QAP [4, Elshafei]
- Computer-aided layout design [10, Krarup and Pruzan]
- Campus building arrangement [3, Dickey and Hopkins]
- computer manufacturing [5, Eschermann and Wunderlich]
- backboard wiring [11, Steinberg]
- scheduling [1, Bierwirth, Mattfeld and Kopfer]
- process communications, turbine balancing ...

Combinatorial Problems modeled as QAPs

- Traveling Salesman Problem
- Linear Ordering Problem
- Bin-Packing Problem
- Max Clique Problem
- Isomorphism of Graphs

Complexity

The quadratic assignment problem is not only of practical and theoretical importance, it is known to be among the most challenging discrete optimization problems.

In general, instances of size $n \geq 30$ cannot be solved in reasonable time.

Sahni and Gonzales (1976) had shown that the quadratic assignment problem is **NP-hard**.

Approaches for exact Solutions

The QAP is a non-convex quadratic optimization problem.
Searching for the global solution of QAP usually involves:

- Branch-And-Bound,
- Cutting Planes,
- Branch-And-Cut.

Crucial issue in the Branch-And-Bound approach is to find efficient ways to compute strong lower bounds.

Relaxation Techniques

In optimization relaxation techniques are approaches to construct convex/conic optimization problems which envelope the original non-convex problem.

Relaxation techniques that have been applied to QAP:

- Gilmore-Lower Bound,
- RLT level 1-3,
- Lifting-and-Projection,
- Eigenvalue based techniques,
- Matrix Splitting,
- Exploiting of (Group) Symmetry
- ... and many more ...

SDP Relaxation of QAPs based on Matrix Splitting

Recently, J. Peng, H. Mittelmann and X. Li [7, 2010] proposed a new class of semidefinite relaxation models for QAPs. This model is based on matrix splitting.

Relaxation frameworks based on matrix splitting allow us to compute strong lower bounds without the need of lifting the problem to greater dimension.

SDP Relaxation of QAPs based on Matrix Splitting

Recently, J. Peng, H. Mittelmann and X. Li [7, 2010] proposed a new class of semidefinite relaxation models for QAPs. This model is based on matrix splitting.

Relaxation frameworks based on matrix splitting allow us to compute strong lower bounds without the need of lifting the problem to greater dimension.

Some Definitions

- $n :=$ dimension of QAP
- $e := (1, 1, \dots, 1)^T$
- $B_{off} :=$ regard only off-diagonal elements of B
- $Z \succeq Y \Leftrightarrow Z - Y \succeq 0 \Leftrightarrow Z - Y \in \mathcal{S}_+$
- $\mathcal{L}_2(B) := \sqrt{\text{diag}(BB^T)}$
- $\min(B)_i := \min(B_{i,:})$

Some Definitions

- $n :=$ dimension of QAP
- $e := (1, 1, \ldots, 1)^T$
- $B_{off} :=$ regard only off-diagonal elements of B
- $Z \succeq Y \iff Z - Y \succeq 0 \iff Z - Y \in \mathcal{S}_+$
- $\mathcal{L}_2(B) := \sqrt{\mathrm{diag}(BB^T)}$
- $\min(B)_i := \min(B_{i,:})$

Some Definitions

- $n :=$ dimension of QAP
- $e := (1, 1, \dots, 1)^T$
- $B_{off} :=$ regard only off-diagonal elements of B
- $Z \succeq Y \;\Leftrightarrow\; Z - Y \succeq 0 \;\Leftrightarrow\; Z - Y \in \mathcal{S}_+$
- $\mathcal{L}_2(B) := \sqrt{\text{diag}(BB^T)}$
- $\min(B)_i := \min(B_{i,:})$

Some Definitions

- $n :=$ dimension of QAP
- $e := (1, 1, \dots, 1)^T$
- $B_{off} :=$ regard only off-diagonal elements of B
- $Z \succeq Y \;\Leftrightarrow\; Z - Y \succeq 0 \;\Leftrightarrow\; Z - Y \in \mathcal{S}_+$
- $\mathcal{L}_2(B) := \sqrt{\mathrm{diag}(BB^T)}$
- $\min(B)_i := \min(B_{i,:})$

Some Definitions

- $n :=$ dimension of QAP
- $e := (1, 1, \dots, 1)^T$
- $B_{off} :=$ regard only off-diagonal elements of B
- $Z \succeq Y \;\Leftrightarrow\; Z - Y \succeq 0 \;\Leftrightarrow\; Z - Y \in \mathcal{S}_+$
- $\mathcal{L}_2(B) := \sqrt{\operatorname{diag}(BB^T)}$
- $\min(B)_i := \min(B_{i,:})$

Some Definitions

- $n :=$ dimension of QAP
- $e := (1, 1, \dots, 1)^T$
- $B_{off} :=$ regard only off-diagonal elements of B
- $Z \succeq Y \;\Leftrightarrow\; Z - Y \succeq 0 \;\Leftrightarrow\; Z - Y \in \mathcal{S}_+$
- $\mathcal{L}_2(B) := \sqrt{\text{diag}(BB^T)}$
- $\min(B)_i := \min(B_{i,:})$

Low-dimensional Semidefinite Relaxation

Every permutation matrix is also a unitary matrix:

$\Rightarrow$ $Y = XBX^T$ and B have same set of eigenvalues.

$\Rightarrow$ $Y \succeq 0 \;\Leftrightarrow\; B \succeq 0$.

If $B \succeq 0$, then for the QAP we obtain the SDP-relaxation

$$\min_{Y \succeq 0} \; \operatorname{Tr}(AY). \tag{2}$$

Low-dimensional Semidefinite Relaxation

Every permutation matrix is also a unitary matrix:

$\Rightarrow$ $Y = XBX^T$ and B have same set of eigenvalues.

$\Rightarrow$ $Y \succeq 0 \Leftrightarrow B \succeq 0$.

If $B \succeq 0$, then for the QAP we obtain the SDP-relaxation

$$\min_{Y \succeq 0} \operatorname{Tr}(AY). \tag{2}$$

SDP relaxation of QAP (Peng, Mittelmann and Li, 2010)

$$
\begin{aligned}
\min \quad & \mathrm{Tr}(AY^+) - \mathrm{Tr}(AY^-) \\
s.t. \quad & Xe = X^T e = e, \quad X \geq 0, \\
& Y^+ \succeq XB^+X^T, \quad Y^- \succeq XB^-X^T, \\
& Y^+ e = XB^+ e, \quad Y^- e = XB^- e; \\
& \mathrm{diag}(Y^+) = X\,\mathrm{diag}(B^+), \quad \mathrm{diag}(Y^-) = X\,\mathrm{diag}(B^-), \\
& \min(Y^+_{off}) \geq X \min(B^+_{off}), \quad \max(Y^+_{off}) \leq X \max(B^+_{off}), \\
& \min(Y^-_{off}) \geq X \min(B^-_{off}), \quad \max(Y^-_{off}) \leq X \max(B^-_{off}), \\
& \mathcal{L}_2(Y^+) \leq X\mathcal{L}_2(B^+), \quad \mathcal{L}_2(Y^-) \leq X\mathcal{L}_2(B^-), \\
& \qquad\qquad \vdots \\
& \text{constraints for } (Y^+ + Y^-) \text{ and } (Y^+ - Y^-).
\end{aligned}
$$

SDP relaxation of QAP (Peng, Mittelmann and Li, 2010)

$$
\begin{aligned}
\min \quad & \mathrm{Tr}(AY^+) - \mathrm{Tr}(AY^-) \\
s.t. \quad & Xe = X^T e = e, \quad X \geq 0, \\
& Y^+ \succeq XB^+X^T, \quad Y^- \succeq XB^-X^T, \\
& Y^+ e = XB^+ e, \quad Y^- e = XB^- e; \\
& \mathrm{diag}(Y^+) = X\,\mathrm{diag}(B^+), \quad \mathrm{diag}(Y^-) = X\,\mathrm{diag}(B^-), \\
& \min(Y^+_{off}) \geq X \min(B^+_{off}), \quad \max(Y^+_{off}) \leq X \max(B^+_{off}), \\
& \min(Y^-_{off}) \geq X \min(B^-_{off}), \quad \max(Y^-_{off}) \leq X \max(B^-_{off}), \\
& \mathcal{L}_2(Y^+) \leq X\mathcal{L}_2(B^+), \quad \mathcal{L}_2(Y^-) \leq X\mathcal{L}_2(B^-), \\
& \qquad \vdots \\
& \text{constraints for } (Y^+ + Y^-) \text{ and } (Y^+ - Y^-).
\end{aligned}
$$

Linear Constraints

$$B \geq \min(\min(B)) \quad \Leftrightarrow \quad Y \geq \min(\min(B))$$

$$\downarrow$$

$$B \geq \min(B)e^T \Leftrightarrow Y \geq X\min(B)e^T \Leftrightarrow \min(Y) \geq X\min(B)$$

$$\downarrow$$

$$B \geq c_l e^T + e r_l \quad \Leftrightarrow \quad Y \geq X c_l e^T + e r_l X^T$$

$$\downarrow$$

$$B \geq v_l e^T + e v_l^T \quad \Leftrightarrow \quad Y \geq X v_l e^T + e v_l^T X^T$$

Semidefinite Constraints

$$B \succeq 0 \quad \Leftrightarrow \quad Y \succeq 0$$

$$\downarrow$$

$$B \succeq B \quad \Leftrightarrow \quad Y \succeq XBX^T$$

$$\downarrow$$

$$B \succeq VV^T \quad \Leftrightarrow \quad Y \succeq XVV^TX^T$$

$$\begin{pmatrix} Y & XV \\ V^TX^T & I \end{pmatrix} \succeq 0$$

Extraction of Linear Term

$$B = \tilde{B} + \mu e e^T \quad \Leftrightarrow \quad Y = \tilde{Y} + \mu e e^T$$

$$\downarrow$$

$$B = \tilde{B} + \operatorname{diag}(v_d) + \mu e e^T \quad \Leftrightarrow \quad Y = \tilde{Y} + \operatorname{diag}(X v_d) + \mu e e^T$$

$$\downarrow$$

$$\tilde{B} + \operatorname{diag}(v_d) + v_s e^T + e v_s^T \quad \Leftrightarrow \quad \tilde{Y} + \operatorname{diag}(X v_d) + X v_s e^T + e v_s^T X^T$$

$$\tilde{B}, \tilde{Y} \succeq 0$$

Matrix Splitting

$$B = B^{+} - B^{-} \quad \Leftrightarrow \quad Y = Y^{+} - Y^{-}$$

$$\downarrow$$

$$B = \tilde{B} + B_{lin} \quad \Leftrightarrow \quad Y = \tilde{Y} + Y_{lin}$$

$$\downarrow$$

$$B = B_{psd} + B_{soc} + B_{lin} \quad \Leftrightarrow \quad Y = Y_{psd} + Y_{soc} + Y_{lin}$$

Nonredundant matrix splitting [8, Peng, Zhu, Luo and Toh, 2011] for tighter relaxations.

The Framework (F-SMSR)

$$
\begin{aligned}
\min \quad & \sum_{i}^{s} \operatorname{Tr}(AY_i) \\
s.t. \quad & Xe = X^T e = e, \; X \geq 0, \\
& Y_1 = \operatorname{diag}(Xv_1) + Xc_1 e^T + er_1 X^T, \\
& \mathcal{L}_2(Y_2) \leq X \mathcal{L}_2(B_2), \\
& \left.\begin{aligned}
& Y_i \succeq X V_i V_i^T X^T, \\
& \operatorname{diag}(Y_i) = X \operatorname{diag}(B_i), \\
& Y_i e = X B_i e, \\
& (Y_i)_{off} \geq X c_{li} e^T + e r_{li} X^T, \\
& (Y_i)_{off} \leq X c_{ui} e^T + e r_{ui} X^T,
\end{aligned}\right\} i = 2, \ldots, s
\end{aligned}
$$

Results

$$R_{gap} = 1 - \frac{\text{lower bound}}{\text{best known feasible objective value}}$$

	MSDR3	S-SVD	$SDRMS_{sum}$	F-SMSR
kra30a	18.51%	16.86%	16.80%	13.78%
kra30b	20.00%	17.80%	17.71%	14.03%
ste36a	44.98%	19.58%	19.19%	17.89%
ste36c	36.93%	18.24%	14.64%	12.27%
tai30b	86.69%	14.75%	12.82%	10.38%
tai35b	82.44%	21.68%	14.52%	13.28%*
tai40b	67.75%	14.57%	11.43%	10.76%*
tai50b	61.78%	16.86%	13.79%	12,58%*

* new best known bounds

Verification

Is the verification approach really necessary?

Ill-Posed Relaxations

We observed in our experiments that sometimes the SDP solver in CVX stops because of slow progress or a large gap. In such a case, we use the procedure described in [2] (coded by Kim-chuan Toh) to find a rigorous lower bound for our relaxation model which further yields a lower bound for the underlying QAP.

[7, Peng, Mittelmann and Li]

Ill-Posed Relaxations 2

A straight forward CVX implementation of the F-SMSR model leads to the following results.

Problem	CVX - Status
kra30a	76643 *
kra30b	78596
ste36a	7822 *
ste36c	failed
tai30b	failed
tai35b	unbounded
tai40b	failed
tai50b	unbounded

* numerical problems

Are modern solvers not reliable enough?

Constructing well-posed Problems

There are different approaches to overcome the ill-possedness of these frameworks:

- add small tolerance gaps to the relaxation constraints
- single-sided (conic) relaxation of equality constraints
- reverse tightening and relaxing of constraints
- **penalty approach**

Constructing well-posed Problems

There are different approaches to overcome the ill-possedness of these frameworks:

- add small tolerance gaps to the relaxation constraints
- single-sided (conic) relaxation of equality constraints
- reverse tightening and relaxing of constraints
- **penalty approach**

Constructing well-posed Problems

There are different approaches to overcome the ill-possedness of these frameworks:

- add small tolerance gaps to the relaxation constraints
- single-sided (conic) relaxation of equality constraints
- reverse tightening and relaxing of constraints
- penalty approach

Constructing well-posed Problems

There are different approaches to overcome the ill-possedness of these frameworks:

- add small tolerance gaps to the relaxation constraints
- single-sided (conic) relaxation of equality constraints
- reverse tightening and relaxing of constraints
- **penalty approach**

Penalty Approach

Instead of inhibiting the violation of constraints we allow these violations but penalize them with high cost factors:

$$Xe = e \quad \xrightarrow[\text{variables}]{\text{add slick}} \quad Xe + s = e, \;\; p \geq \|s\|_2, \tag{3}$$

where the error variable p will be penalized to minimize the distance to the actual solution of the corresponding relaxation.

- The penalty approach can also be realized using **LP** instead of **SOCP**.
- The same approach is also applicable to cone constraints.

Penalty Approach

Instead of inhibiting the violation of constraints we allow these violations but penalize them with high cost factors:

$$Xe = e \quad \xrightarrow[\text{variables}]{\text{add slick}} \quad Xe + s = e, \;\; p \geq \|s\|_2, \tag{3}$$

where the error variable p will be penalized to minimize the distance to the actual solution of the corresponding relaxation.

- The penalty approach can also be realized using **LP** instead of **SOCP**.
- The same approach is also applicable to cone constraints.

Penalty Approach

Instead of inhibiting the violation of constraints we allow these violations but penalize them with high cost factors:

$$Xe = e \quad \xrightarrow[\text{variables}]{\text{add slick}} \quad Xe + s = e, \;\; p \geq \|s\|_2, \tag{3}$$

where the error variable p will be penalized to minimize the distance to the actual solution of the corresponding relaxation.

- The penalty approach can also be realized using **LP** instead of **SOCP**.
- The same approach is also applicable to cone constraints.

Rigorous Bounds

Statements

- In general, it is very difficult to compute rigorous bounds for bad conditioned or even ill-posed optimization problems.
- This is particularly true for higher dimensioned problems.
- Most general approaches will fail to compute useful rigorous bounds!

Rigorous Bounds

Statements

- In general, it is very difficult to compute rigorous bounds for bad conditioned or even ill-posed optimization problems.
- This is particularly true for higher dimensioned problems.
- Most general approaches will fail to compute useful rigorous bounds!

Rigorous Bounds

Statements

- In general, it is very difficult to compute rigorous bounds for bad conditioned or even ill-posed optimization problems.
- This is particularly true for higher dimensioned problems.
- Most general approaches will fail to compute useful rigorous bounds!

Rigorous Bounds for Combinatorial Problems

For relaxation of combinatorial problems like the QAP the situation is different, even for the ill-posed ones:

- It is a-priori known that the relaxations have feasible solutions.
- All optimization variables are bounded. The corresponding upper bounds can be computed easily.
- By using the techniques described in [6, Jansson, 2009] it is possible to compute rigorous lower bounds.
- The procedure is implemented in the software package **VSDP**.

Rigorous Bounds for Combinatorial Problems

For relaxation of combinatorial problems like the QAP the situation is different, even for the ill-posed ones:

- It is a-priori known that the relaxations have feasible solutions.
- All optimization variables are bounded. The corresponding upper bounds can be computed easily.
- By using the techniques described in [6, Jansson, 2009] it is possible to compute rigorous lower bounds.
- The procedure is implemented in the software package **VSDP**.

Rigorous Bounds for Combinatorial Problems

For relaxation of combinatorial problems like the QAP the situation is different, even for the ill-posed ones:

- It is a-priori known that the relaxations have feasible solutions.
- All optimization variables are bounded. The corresponding upper bounds can be computed easily.
- By using the techniques described in [6, Jansson, 2009] it is possible to compute rigorous lower bounds.
- The procedure is implemented in the software package **VSDP**.

Rigorous Bounds for Combinatorial Problems

For relaxation of combinatorial problems like the QAP the situation is different, even for the ill-posed ones:

- It is a-priori known that the relaxations have feasible solutions.
- All optimization variables are bounded. The corresponding upper bounds can be computed easily.
- By using the techniques described in [6, Jansson, 2009] it is possible to compute rigorous lower bounds.
- The procedure is implemented in the software package **VSDP**.

Rigorous Bounds for Combinatorial Problems

For relaxation of combinatorial problems like the QAP the situation is different, even for the ill-posed ones:

- It is a-priori known that the relaxations have feasible solutions.
- All optimization variables are bounded. The corresponding upper bounds can be computed easily.
- By using the techniques described in [6, Jansson, 2009] it is possible to compute rigorous lower bounds.
- The procedure is implemented in the software package **VSDP**.

Results

$$v_{gap} = 1 - \frac{\text{verified bound}}{\text{solver approximation}}$$

Problem	VSDP gap
kra30a	4.36e-10
kra30b	1.44e-10
ste36a	4.12e-11
ste36c	8.83e-06
tai30b	1.25e-08
tai35b	8.31e-09
tai40b	7.16e-08
tai50b	3.19e-07

Summary

- SDP relaxations based on matrix splitting have a strong potential for the use in a branch-and-bound approach.
- Compared to other frameworks these relaxations need more effort and anticipation in the implementation.
- The additional effort is rewarded by strong bounds at a relative cheap cost.
- Although the framework tends to be ill-conditioned for many problem instances it is possible to compute rigorous inclusions at insignificant additional cost.

Summary

- SDP relaxations based on matrix splitting have a strong potential for the use in a branch-and-bound approach.
- Compared to other frameworks these relaxations need more effort and anticipation in the implementation.
- The additional effort is rewarded by strong bounds at a relative cheap cost.
- Although the framework tends to be ill-conditioned for many problem instances it is possible to compute rigorous inclusions at insignificant additional cost.

Summary

- SDP relaxations based on matrix splitting have a strong potential for the use in a branch-and-bound approach.
- Compared to other frameworks these relaxations need more effort and anticipation in the implementation.
- The additional effort is rewarded by strong bounds at a relative cheap cost.
- Although the framework tends to be ill-conditioned for many problem instances it is possible to compute rigorous inclusions at insignificant additional cost.

Summary

- SDP relaxations based on matrix splitting have a strong potential for the use in a branch-and-bound approach.
- Compared to other frameworks these relaxations need more effort and anticipation in the implementation.
- The additional effort is rewarded by strong bounds at a relative cheap cost.
- Although the framework tends to be ill-conditioned for many problem instances it is possible to compute rigorous inclusions at insignificant additional cost.

D. C. M. Christian Bierwirth and H. Kopfer.
On permutation representations for scheduling problems.
Lecture Notes in Computer Science, 1141:310–318, 1996.

D. C. Christian Jansson and C. Keil.
Rigorous error bounds for the optimal value in semidefinite programming.
SIAM J. Numerical Analysis, 46:188–200, 2007.

J. W. Dickey and J. W. Hopkins.
Transportation Research, 6:59–68, 1972.

A. N. Elshafei.
Hospital layout as a quadratic assignment problem.
Operations Research Quarterly, 28:167–179, 1977.

B. Eschermann and H.-J. Wunderlich.
Optimized synthesis of self-testable finite state machines, 1990.
In 20th International Symposium on Fault-Tolerant Computing (FFTCS 20), Newcastle upon Tyne, 26-28th June.

C. Jansson.
On verified numerical computations in convex programming.
Japan J. Indust. Appl. Math., 26:337–363, 2009.

H. D. M. Jiming Peng and X. Li.
A new relaxation framework for quadratic assignment problems based on matrix splitting.
Mathematical Programming Computation, 2:59–77, 2010.

H. L. Jiming Peng, Tao Zhu and K.-C. Toh.
Semidefinite relaxation of quadratic assignment problems based on nonredundant matrix splitting, 2011.

T. C. Koopmans and M. Beckmann.
Assignment problems and the location of economic activities.
Econometrica, 25:53–76, 1957.

J. Krarup and P. M. Pruzan.
Computer-aided layout design.
Mathematical Programming Study, 9:75–94, 1978.

L. Steinberg.
The backboard wiring problem: A placement algorithm.
SIAM Review, 3:37–50, 1961.