Several observations about using the ALM + semismooth Newton method for solving large scale SDP and beyond

Defeng Sun

Department of Applied Mathematics

THE HONG KONG POLYTECHNIC UNIVERSITY
香港理工大學

One World Optimization Seminar
November 30, 2020

The nearest correlation matrix model

Let us start with a simple Nearest Correlation Matrix (NCM) problem (a terminology coined by Nick Higham in 2002)

$$\begin{aligned} \min \quad & \frac{1}{2}\|X - G\|_F^2 \\ \text{s.t.} \quad & X_{ii} = 1\,,\ i = 1, \ldots, n\,, \\ & X \succeq 0\,, \end{aligned}$$

which is a special case of the best approximation problem

$$\begin{aligned} \min \quad & \frac{1}{2}\|x - c\|^2 \\ \text{s.t.} \quad & \mathcal{A}x \in b + Q\,, \\ & x \in K\,, \end{aligned}$$

where $\mathcal{X}$ is a finite-dimensional real Hilbert space equipped with a scalar product $\langle \cdot, \cdot \rangle$ and its induced norm $\| \cdot \|$

$\mathcal{A} : \mathcal{X} \to \Re^m$ is a linear operator

$Q = \{0\}^p \times \Re^q_+$ is a polyhedral convex cone, $1 \leq p \leq m$, $q = m - p$, and K is a closed convex cone in $\mathcal{X}$.

The dual formulation

The dual of the best approximation problem is in the form of

$$\begin{array}{ll} \max & -\theta(y) := -\left[\frac{1}{2}\|\Pi_K(c+\mathcal{A}^*y)\|^2 - \langle b, y\rangle - \frac{1}{2}\|c\|^2\right] \\ \text{s.t.} & y \in Q^*\,, \end{array}$$

where $Q^* = \Re^p \times \Re^q_+$.

One can solve the above dual by solving the following nonsmooth equation

$$F(y) := y - \Pi_{Q^*}[y - (\mathcal{A}\Pi_K(c+\mathcal{A}^*y) - b)] = 0, \quad y \in \Re^m\,.$$

For the NCM problem,

- $\mathcal{A}(X) = \mathrm{diag}(X)$, a vector consisting of all diagonal entries of X.
- $\mathcal{A}^*(y) = \mathrm{diag}(y)$, the diagonal matrix.
- $b = e$, the vector of all ones in $\Re^n$ and $K = \mathcal{S}^n_+$

$$F(y) = \mathcal{A}\Pi_{\mathcal{S}^n_+}(G + \mathcal{A}^*y) - b.$$

Solving the NCM with a semismooth Newton method

Let $X \in \mathcal{S}^n$ have the following spectral decomposition

$$X = P\Lambda P^T,$$

where Λ is the diagonal matrix of eigenvalues of X arranged in the non-increasing order and P is a corresponding orthogonal matrix of orthonormal eigenvectors.

Then

$$X_+ := \Pi_{\mathcal{S}^n_+}(X) = P\Lambda_+ P^T.$$

When X is nonsingular, $\Pi_{\mathcal{S}^n_+}(\cdot)$ is continuously differentiable near X with

$$\Pi'_{\mathcal{S}^n_+}(X)(\Delta X) = P[\Omega \circ P^T(\Delta X)P]P^T \quad \forall\, \Delta X \in \mathcal{S}^n,$$

where

$$\Omega_{ij} = \frac{\max(0,\lambda_i) - \max(0,\lambda_j)}{\lambda_i - \lambda_j} \text{ if } \lambda_i \neq \lambda_j \;\&\; \Omega_{ij} = (\max(0,\lambda_i))' \text{ otherwise.}$$

We have

- $\|X_+\|^2$ is continuously differentiable with

$$\nabla\left(\frac{1}{2}\|X_+\|^2\right) = X_+,$$

but is not twice continuously differentiable (good news!).

- X_+ is not piecewise smooth unless $n = 1$, but strongly semismooth[1]: for any $X \in \mathcal{S}^n$, $\Pi_{\mathcal{S}^n_+}$ is directionally differentiable at X and it holds for X that

$$\Pi_{\mathcal{S}^n_+}(Y) - \Pi_{\mathcal{S}^n_+}(X) - \Pi'_{\mathcal{S}^n_+}(Y)(Y - X) = O(\|Y - X\|^2)$$

for any $Y \in \mathcal{S}^n$ such that $\Pi_{\mathcal{S}^n_+}(\cdot)$ is differentiable at Y and $Y \to X$.

[1] S. and J. Sun, Semismooth matrix valued functions. *Math of OR* 27 (2002) 150–169.

Numerical efficiency

We test the efficiency of a quadratically convergent Newton's method designed by Houduo Qi and S. [2] The written code is called CorrelationMatrix.m on randomly generated pseudo correlation matrix. The code (in other languages too) is publically available in my webpage. All the experiments are done on the ThinkStation Desktop with Intel (R) Core(TM) i7-8700 processor.

n	relgap		Iterations		Time (s)	
1000	1.70E-08	8.05E-14	5	7	0.9	1.2
3000	3.83E-08	6.41E-14	5	7	13.7	18.5
5000	8.68E-07	1.91E-12	5	7	61.79	83.8
10000	1.81E-07	1.95E-12	5	7	342.42	520.25

Observations: 1. In addition to the strong convexity of the objective function, the key point for the semismooth Newton method to work is the primal constraint non-degeneracy (transversality, generalized LICQ). 2. Ω can be very sparse to reduce computational costs substantially.

[2] H.D. Qi and S., A quadratically convergent Newton method for computing the nearest correlation matrix. *SIAM J. Matrix Analysis and Applications* 28 (2006) 360–385.

Inequality constraints

If we have lower and upper bounds on X, F takes the form

$$F(y) = y - \Pi_{Q^*}\left[y - (\mathcal{A}\Pi_{\mathcal{S}^n_+}(G + \mathcal{A}^* y) - b)\right],$$

which involves double layered projections over convex cones.

A quadratically convergent smoothing Newton method is designed by Yan Gao and S.[3].

Again, highly efficient if the constraint non-degeneracy holds!

[3] Y. Gao and S., Calibrating least squares covariance matrix problems with equality and inequality constraints, SIAM J on Matrix Analysis and Applications 31 (2009), 1432–1457.

Second order cone + PSD cone reformulation

One may write the NCM as a symmetric cone programming with both SDP cone and SOC cone constraints:

$$\begin{aligned} \min \quad & t \\ \text{s.t.} \quad & X_{ii} = 1,\, i = 1, \ldots, n, \\ & y + \operatorname{svec}(X) = \operatorname{svec}(G)\,, \\ & X \in \mathcal{S}_+^n\,,\, t \geq \|y\|_2. \end{aligned}$$

Then, we can solve the problem with interior point methods (IPMs).

Numerical comparison with SDPT3

We compare our algorithm with the state-of-the-art IPM based software SDPT3[4].

n	relgap		Iterations		Time(s)	
	SDPT3	Newton	SDPT3	Newton	SDPT3	Newton
100	1.79E-09	7.80E-10	13	5	20.95	**0.02**
150	1.74E-06	1.74E-09	15	5	126.3	**0.03**
200	4.41E-09	2.54E-09	14	5	425.3	**0.04**
500	X	8.81E-09	X	5	X	**0.19**

x means SDPT3 is out of memory (117.8 GB memory is required).

[4] R.H Tutuncu, K.C. Toh, and M.J. Todd, Solving semidefinite-quadratic-linear programs using SDPT3, Mathematical Programming Ser. B, 95 (2003), pp. 189-217.

First order method

The dual problem of the NCM is an unconstrained optimization problem:

$$\max \quad -\theta(y) := -\big[\tfrac{1}{2}\|\Pi_{\mathcal{S}^n_+}(G + \mathcal{A}^* y)\|_F^2 - \langle b, y\rangle - \tfrac{1}{2}\|G\|_F^2\big],$$

with

$$\nabla\theta(y) = \mathcal{A}\Pi_{\mathcal{S}^n_+}(G + \mathcal{A}^* y) - b.$$

Then, it is natural to apply the accelerated proximal gradient (APG) method to solve this dual problem:

$$\begin{cases} z^{k+1} = x^k - \nabla\theta(y^k), \\ x^{k+1} = (1 - \frac{2}{k+2})x^k + \frac{2}{k+2} z^k, \\ y^{k+1} = (1 - \frac{2}{(k+1)+2})x^{k+1} + \frac{2}{(k+1)+2} z^{k+1}, \end{cases}$$

where $x^0 = z^0 = y^0$ and y^0 is the given initial point.

Numerical comparison with APG

We compare the APG with the semismooth Newton method on a randomly generated data, for the given integer n, set

$$G := \mathrm{rand}(n, n), G = 0.5 * (G + G') - \mathrm{diag}(\mathrm{diag}(0.5 * (G + G'))) + \mathrm{eye}(n).$$

We set the maximum iterations for APG to be $1,000$ and the tolerance to be $1e-6$.

n	Residual		Iterations		Time(s)	
	APG	Newton	APG	Newton	APG	Newton
100	9.62E-07	1.82E-07	53	4	0.08	**0.02**
500	9.95E-07	1.05E-07	104	5	1.5	**0.26**
1000	9.79E-07	4.22E-07	142	5	8.64	**0.8**
2000	9.48E-07	4.60E-07	191	5	93.01	**4.4**

Numerical comparison with APG (2)

To see the robustness of APG, we also test

$$G := \mathrm{rand}(1000, 1000), G = G + G' - \mathrm{diag}(\mathrm{diag}(G + G')) + \mathrm{eye}(1000).$$

We set the maximum iterations for APG to be $1,000$ and the tolerance to be $1e-6$.

Algorithms	Time (s)	Iterations	Residual
APG	61.1	1000	3.98E-04
Newton	1.4	9	3.34E-07

This motivates us to use semismooth Newton methods to solve semidefinite programming (SDP) under the framework of proximal point algorithms (PPAs).

Linear conic programming

Now we can turn to the following linear conic programming

$$\begin{aligned} \min \quad & \langle c, x \rangle \\ \text{s.t.} \quad & \mathcal{A}x = b, \\ & x \in \mathcal{K} \end{aligned}$$

where $\mathcal{A} : \mathcal{X} \to \Re^m$, $b \in \Re^m$, $c \in \mathcal{X}$, $\mathcal{X}$ is a finite-dimensional real Hilbert space and $\mathcal{K} \subseteq \mathcal{X}$ is a closed convex cone, e.g., the second-order-cone or the PSD (positive and semidefinte) cone. Define the Lagrange function as follows:

$$L(x; y, s) := \langle c, x \rangle + \langle y, b - \mathcal{A}x \rangle - \langle s, x \rangle.$$

Let $\mathcal{K}^*$ be the dual cone of $\mathcal{K}$. Then the (Lagrange) dual of linear conic programming is defined as

$$\max_{y \in \Re^m, s \in \mathcal{K}^*} \left\{ \inf_{x \in \mathcal{X}} L(x; y, s) \right\}.$$

Lagrange dual of linear conic programming

By noting that for any $(x, y, s) \in \mathcal{X} \times \Re^m \times \mathcal{X}^n$,

$$L(x; y, s) := \langle x, c - \mathcal{A}^* y - s \rangle + \langle y, b \rangle$$

we get an explicit formula for the dual problem as in the following

$$\begin{array}{ll} \max & \langle b, y \rangle \\ \text{s.t.} & \mathcal{A}^* y + s = c, \\ & s \in \mathcal{K}^* \end{array}$$

or equivalently

$$\begin{array}{ll} \max & \langle b, y \rangle \\ \text{s.t.} & c - \mathcal{A}^* y \in K^* \end{array}$$

No one will question the above (Lagrange) dual!

Convex composite quadratic programming

Now consider the convex composite quadratic programming (CCQP)

$$\min_{x\in\mathcal{X}} \left\{ \frac{1}{2}\langle x, \mathcal{Q}x\rangle + \langle c, x\rangle + \psi(x) \;\middle|\; \mathcal{A}x = b \right\}$$

- $\mathcal{X}$ and $\mathcal{Y}$ are two finite-dimensional real Hilbert spaces
- $\psi : \mathcal{X} \to (-\infty, +\infty]$ is a closed proper convex function, e.g., $\psi(\cdot) = \delta_P(\cdot)$, the indicator function over a closed convex set P
- $\mathcal{Q} : \mathcal{X} \to \mathcal{X}$ satisfying $\mathcal{Q} = \mathcal{Q}^*$, $\mathcal{Q} \succeq 0$
- $\mathcal{A} : \mathcal{X} \to \mathcal{Y}$ is a given linear mapping
- $b \in \mathcal{Y}$ and $c \in \mathcal{X}$ are given vectors

Convex quadratic programming (continued)

Equivalently,

$$\min_{u,x\in\mathcal{X}} \left\{ \psi(u) + \langle c, x\rangle + \frac{1}{2}\langle x, \mathcal{Q}x\rangle \;\middle|\; u - x = 0, \quad \mathcal{A}x = b \right\}$$

The corresponding Lagrange function is

$$\begin{aligned} L(u,x;y,s) &:= \psi(u) + \langle c, x\rangle + \tfrac{1}{2}\langle x, \mathcal{Q}x\rangle + \langle s, u - x\rangle + \langle y, b - \mathcal{A}x\rangle \\ &= \langle y, b\rangle + \psi(u) + \langle s, u\rangle + \tfrac{1}{2}\langle x, \mathcal{Q}x\rangle + \langle x, c - \mathcal{A}^* y - s\rangle \end{aligned}$$

and the Lagrange dual of CCQP takes the form of

$$\max_{y\in\mathcal{Y},s\in\mathcal{X}} \left\{ \inf_{u\in\mathcal{X},x\in\mathcal{X}} L(u,x;y,s) \right\}$$

or

$$\max_{y\in\mathcal{Y},s\in\mathcal{X}} \left\{ \langle y, b\rangle + \inf_{u\in\mathcal{X}} \left\{ \psi(u) + \langle s, u\rangle \right\} + \inf_{x\in\mathcal{X}} \left\{ \frac{1}{2}\langle x, \mathcal{Q}x\rangle + \langle x, c - \mathcal{A}^* y - s\rangle \right\} \right\}$$

Wolfe dual

By simplifications, we get the following Lagrange dual

$$\max_{y\in\mathcal{Y},s\in\mathcal{X}} \Big\{ -\psi^*(-s) + \langle y, b\rangle + \theta(y,s)\Big\},$$

where

$$\theta(y,s) := \inf_{x\in\mathcal{X}} \Big\{ \frac{1}{2}\langle x, \mathcal{Q}x\rangle + \langle x, c - \mathcal{A}^* y - s\rangle \Big\}$$

and $\psi^*(\cdot)$ is the Fenchel conjugate of ψ defined by

$$\psi^*(z) := \sup_{u\in\mathcal{X}} \{\langle z, u\rangle - \psi(u)\}.$$

Since the computation of $\theta(y,s)$ is complicated, instead one normally considers the following Wolfe dual [Wolfe, Quart. Appl. Math 1961]

$$\max_{s\in\mathcal{X},x\in\mathcal{X},y\in\mathcal{Y}} \Big\{ -\psi^*(-s) - \frac{1}{2}\langle x, \mathcal{Q}x\rangle + \langle y, b\rangle \;\Big|\; s - \mathcal{Q}x + \mathcal{A}^* y = c\Big\}.$$

Limitations of the Wolfe dual

Note that in the Wolfe dual (in the minimization format)

$$\min_{s\in\mathcal{X},x\in\mathcal{X},y\in\mathcal{Y}} \left\{ \psi^*(-s) + \frac{1}{2}\langle x, \mathcal{Q}x\rangle - \langle y, b\rangle \;\middle|\; s - \mathcal{Q}x + \mathcal{A}^*y = c \right\},$$

the primal variable x is also involved. But more seriously, its solution set, if nonempty, is always unbounded as long as $\mathcal{Q} \not\succ 0$ (the null space of $\mathcal{Q}$ is uncontrollable). It differs substantially from linear conic programming

Primal and dual properties of SDP

- For Linear Semidefinite Programming:
  - The primal (dual, respectively) constraint nondegeneracy is equivalent to the dual (primal, respectively) strong second order sufficient condition (SSOSC) [Chan and S., SIOPT (2008)]
  - The generalized Clarke Jacobian (or the B-subdifferential) of the nonsmooth Karush- Kuhn-Tucker (KKT) solution mapping is nonsingular iff the primal and dual constraint non-degeneracies hold [Chan and S., SIOPT (2008)]
  - The primal (dual, respectively) strict Robinson constraint qualification is equivalent to the dual (primal, respectively) second order sufficient condition (SOSC) [Chao Ding, S., Liwei Zhang, SIOPT (2017)]
- The software SDPNAL for SDP [Xinyuan Zhao, S., Toh, SIOPT 2010] is applied to the dual (low rank or high rank property used)
- Convex quadratic SDP with the Wolfe dual: the above key theoretical connections hold only if $\mathcal{Q} \succ 0$ [Houduo Qi, MOR (2009)]
- Hard, if possible at all, to design an analogue of the software SDPNAL for the Wolfe dual if $\mathcal{Q} \not\succ 0$

The restricted Wolfe dual

Our remedy is to consider the following restricted Wolfe dual (in the minimization format)

$$\min_{s\in\mathcal{X},x'\in\mathcal{X}',y\in\mathcal{Y}} \left\{\psi^*(-s)+\frac{1}{2}\langle x',\mathcal{Q}x'\rangle-\langle y,b\rangle \;\middle|\; s-\mathcal{Q}x'+\mathcal{A}^*y=c\right\},$$

where $\mathcal{X}'$ is the range space of $\mathcal{Q}$, i.e.,

$$\mathcal{X}' := \mathrm{Range}\,(\mathcal{Q}).$$

One can easily check that $\mathcal{Q}:\mathcal{X}'\to\mathcal{X}'$ is self-adjoint and positive definite even if $\mathcal{Q}:\mathcal{X}\to\mathcal{X}$ is not positive definite. Note that if $\mathcal{Q}=0$, then $\mathcal{X}'=\{0\}$ (in this case $\mathcal{Q}$ is still positive definite on $\mathcal{X}'$ – using definition to verify it!).

Also note that x' in the dual is differen from x in the primal, which does not need to stay in $\mathrm{Range}\,(\mathcal{Q})$.

Nice properties of the restricted Wolfe dual

Different from the Wolfe dual, the restricted Wolfe dual possesses the following nice properties:

- It unifies with linear conic programming
- The wonderful equivalent connections between the primal and dual for SDP are now kept [Ying Cui, Chao Ding, 2019 (under revision)]
- The software QSDPNAL[5] for the restricted Wolfe dual of the convex quadratic SDP is developed [Xudong Li, S., Toh, MPC 2018]
- In a word, both theory and computation favor the restricted Wolfe dual

Too good to be true? — the key is to keep x' in $\mathrm{Range}\,(\mathrm{Q})$. But, how? Implementable?

[5] Publicly available at `https://blog.nus.edu.sg/mattohkc/softwares/qsdpnal/` and `https://github.com/MatOpt/QSDPNAL`

Augmented Lagrange function of restricted Wolfe dual

For given $\sigma > 0$, the augmented Lagrange function of the restriced Wolfe dual of the CCQP can be written as

$$\begin{aligned} L_\sigma(s, x', y; x) \quad := \quad & \psi^*(-s) + \tfrac{1}{2}\langle x', \mathcal{Q}x' \rangle - \langle y, b \rangle \\ & + \langle x, s - \mathcal{Q}x' + \mathcal{A}^* y - c \rangle + \tfrac{\sigma}{2} \| s - \mathcal{Q}x' + \mathcal{A}^* y - c \|^2, \end{aligned}$$

which, fixing the dual variable x, is a proper closed convex (non-separable) function in the first block variable s plus a convex quadratic function in terms of (s, x', y).

Note that y can be further split into many pieces as you please: $y = [y^1; y^2; \ldots; y^p]$. No need to decompose s while x' must be kept as one block if the range space is used: a total of $1 + (1 + p)$ blocks

Augmented Lagrange function of the primal CCQP

For given $\sigma > 0$, the augmented Lagrange function of the CCQP (primal)

$$\min_{u,x\in\mathcal{X}} \left\{ \psi(u) + \langle c, x\rangle + \frac{1}{2}\langle x, \mathcal{Q}x\rangle \;\middle|\; u - x = 0, \quad \mathcal{A}x = b \right\}$$

takes the form of

$$\begin{aligned} L_\sigma(u, x; y, s) \;\; := \;\; & \psi(u) + \tfrac{1}{2}\langle x, \mathcal{Q}x\rangle + \langle c, x\rangle + \langle y, b - \mathcal{A}x\rangle + \langle s, u - x\rangle \\ & + \tfrac{\sigma}{2}\|b - \mathcal{A}x\|^2 + \tfrac{\sigma}{2}\|u - x\|^2, \end{aligned}$$

which, fixing the dual variables y and s, is a proper closed convex function in the first block variable u plus a convex quadratic (non-separable) function in terms of (u, x).

- Note that x can be further split into as many pieces as you like: $x = [x^1; x^2; \ldots; x^q]$. There are a total of $1 + q$ blocks.
- The introduction of u is not only necessary for the restricted Wolfe dual but also crucial for computations: not a computationally good idea to split x into multiple parts without u even if $\psi(\cdot)$ is separable.

Two observations

- Note that the augmented Lagrange function for the CCQP in the primal form does not contain a nonsmooth term for x even if it is non-separable for x.
- The (mysterious?) forms of the two augmented Lagrange functions lead to the discovery of the symmetric Gauss-Seidel decomposition theorem!

Convex composite quadratic programming (II)

Actually, we can consider more general convex composite quadratic programming (CCQP)

$$\min_{x\in\mathcal{X}} \left\{ \psi(x) + \frac{1}{2}\langle x, \mathcal{Q}x\rangle - \langle c, x\rangle \;\middle|\; \mathcal{A}_E x = b_E,\ \mathcal{A}_I x - b_I \in \mathcal{K} \right\}$$

- $\psi : \mathcal{X} \to (-\infty, +\infty]$ is a closed proper convex function [simple]
- $\mathcal{Q} : \mathcal{X} \to \mathcal{X}$ satisfying $\mathcal{Q} = \mathcal{Q}^*,\ \mathcal{Q} \succeq 0$
- $\mathcal{A}_E : \mathcal{X} \to \mathcal{Z}_1$ and $\mathcal{A}_I : \mathcal{X} \to \mathcal{Z}_2$, given linear mappings
- $b = (b_E; b_I) \in \mathcal{Z} := \mathcal{Z}_1 \times \mathcal{Z}_2$, given vector
- $c \in \mathcal{X}$ is given.
- $\mathcal{K} \subseteq \mathcal{Z}_2$ is a closed convex set (cone) [simple]
- $\mathcal{X}$, $\mathcal{Z}_1$, and $\mathcal{Z}_2$ are finite-dimensional real Hilbert spaces

CCQP and its restricted Wolfe dual (II)

Equivalently,

$$\min_{x\in\mathcal{X},x'\in\mathcal{Z}_2}\left\{\psi(x)+\delta_{\mathcal{K}}(x')+\frac{1}{2}\langle x,\mathcal{Q}x\rangle-\langle c,x\rangle \;\middle|\; \begin{pmatrix}\mathcal{A}_E & 0\\ \mathcal{A}_I & -\mathcal{I}\end{pmatrix}\begin{pmatrix}x\\ x'\end{pmatrix}=b\right\},$$

whose restricted Wolfe dual (in the minimization format) is

$$\min_{\substack{s\in\mathcal{Y},z\in\mathcal{Z}\\ y'\in\mathrm{Range}(\mathcal{Q})}}\left\{p(s)+\frac{1}{2}\langle y',\mathcal{Q}y'\rangle-\langle b,z\rangle \;\middle|\; s+\begin{pmatrix}\mathcal{Q}\\ 0\end{pmatrix}y'-\begin{pmatrix}\mathcal{A}_E^* & \mathcal{A}_I^*\\ 0 & -\mathcal{I}\end{pmatrix}z=\begin{pmatrix}c\\ 0\end{pmatrix}\right\}$$

- $s:=(u,v)\in\mathcal{Y}:=\mathcal{X}\times\mathcal{Z}_2$
- $p(s):=p(u,v)=\psi^*(u)+\delta^*_{\mathcal{K}}(v)$
- $\delta_{\mathcal{K}}(\cdot)$ is the indicator function over $\mathcal{K}$

A general form of symmetric Gauss-Seidel iteration

Consider the **block** vector $\mathbf{x} = (\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_s) \in \mathcal{X} := \mathcal{X}_1 \times \mathcal{X}_2 \times \cdots \times \mathcal{X}_s$. Given a positive semidefinite linear operator $\mathcal{Q}$ such that

$$\mathcal{Q}\mathbf{x} \equiv \begin{pmatrix} \mathcal{Q}_{11} & \mathcal{Q}_{12} & \cdots & \mathcal{Q}_{1s} \\ \mathcal{Q}_{12}^* & \mathcal{Q}_{22} & \cdots & \mathcal{Q}_{2s} \\ \vdots & \vdots & \ddots & \vdots \\ \mathcal{Q}_{1s}^* & \mathcal{Q}_{2s}^* & \cdots & \mathcal{Q}_{ss} \end{pmatrix} \begin{pmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \\ \vdots \\ \mathbf{x}_s \end{pmatrix}, \quad \mathcal{Q}_{ii} \succ 0.$$

Let $p : \mathcal{X}_1 \to (-\infty, +\infty]$ be a given closed proper convex function. Let the quadratic function

$$q(\mathbf{x}) := \tfrac{1}{2}\langle \mathbf{x},\, \mathcal{Q}\mathbf{x}\rangle - \langle \mathbf{r},\, \mathbf{x}\rangle.$$

An inexact block sGS iteration

Consider the following block decomposition:

$$\mathcal{U}\mathbf{x} \equiv \begin{pmatrix} 0 & \mathcal{Q}_{12} & \cdots & \mathcal{Q}_{1s} \\ & \ddots & & \vdots \\ & & \ddots & \mathcal{Q}_{(s-1)s} \\ & & & 0 \end{pmatrix} \begin{pmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \\ \vdots \\ \mathbf{x}_s \end{pmatrix}.$$

Then $\mathcal{Q} = \mathcal{U}^* + \mathcal{D} + \mathcal{U}$, where $\mathcal{D}\mathbf{x} = (\mathcal{Q}_{11}\mathbf{x}_1, \ldots, \mathcal{Q}_{ss}\mathbf{x}_s)$.

Let $\hat{\delta} \equiv (\hat{\delta}_1, \ldots, \hat{\delta}_s)$ and $\delta^+ \equiv (\delta_1^+, \ldots, \delta_s^+)$ with $\hat{\delta}_1 = \delta_1^+$ being given error tolerance vectors. Define

$$\Delta(\hat{\delta}, \delta^+) := \delta^+ + \mathcal{U}\mathcal{D}^{-1}(\delta^+ - \hat{\delta}),\ \mathcal{T} := \mathcal{U}\mathcal{D}^{-1}\mathcal{U}^*\ (\text{sGS decomp. op.}).$$

Note that $\mathcal{T} \succeq 0$ is NOT positive definite. Let $\mathbf{y} \in \mathcal{X}$ be given. Define

$$\mathbf{x}^+ := \arg\min_{\mathbf{x}\in\mathcal{X}} \left\{ p(\mathbf{x}_1) + q(\mathbf{x}) + \frac{1}{2}\|\mathbf{x} - \mathbf{y}\|_{\mathcal{T}}^2 - \langle \Delta(\hat{\delta}, \delta^+),\ \mathbf{x} \rangle \right\}. \tag{1}$$

(1) looks complicated, but is much easier to solve!

An inexact block sGS decomposition theorem

Theorem (Xudong Li, S., Toh, MP 2019)

Given $\mathbf{y}$. *For* $i = s, \ldots, 2$, *define*

$$\begin{aligned} \hat{\mathbf{x}}_i &:= \arg\min_{\mathbf{x}_i} \{ p(\mathbf{y}_1) + q(\mathbf{y}_{\le i-1}, \mathbf{x}_i, \hat{\mathbf{x}}_{\ge i+1}) - \langle \hat{\delta}_i, \mathbf{x}_i \rangle \} \\ &= \mathcal{Q}_{ii}^{-1} \big(\mathrm{r}_i + \hat{\delta}_i - \textstyle\sum_{j=1}^{i-1} \mathcal{Q}_{ji}^* \mathbf{y}_j - \sum_{j=i+1}^{s} \mathcal{Q}_{ij} \hat{\mathbf{x}}_j \big) \end{aligned}$$

computed in the backward GS cycle. **The optimal solution** $\mathbf{x}^+$ **in** (1) **can be obtained exactly via**

$$\begin{aligned} \mathbf{x}_1^+ &= \arg\min_{\mathbf{x}_1} \{ p(\mathbf{x}_1) + q(\mathbf{x}_1, \hat{\mathbf{x}}_{\ge 2}) - \langle \delta_1^+, \mathbf{x}_1 \rangle \}, \\ \mathbf{x}_i^+ &= \arg\min_{\mathbf{x}_i} \{ p(\mathbf{x}_1^+) + q(\mathbf{x}_{\le i-1}^+, \mathbf{x}_i, \hat{\mathbf{x}}_{\ge i+1}) - \langle \delta_i^+, \mathbf{x}_i \rangle \} \\ &= \mathcal{Q}_{ii}^{-1} (\mathrm{r}_i + \delta_i^+ - \textstyle\sum_{j=1}^{i-1} \mathcal{Q}_{ji}^* \mathbf{x}_j^+ - \sum_{j=i+1}^{s} \mathcal{Q}_{ij} \hat{\mathbf{x}}_j), \quad i \ge 2, \end{aligned}$$

where $\mathbf{x}_i^+$, $i = 1, 2, \ldots, s$, *is computed in the forward GS cycle.*

Reduces to the classical block sGS if both $p(\cdot) \equiv 0$ and $\delta = 0$.

Caution: this theorem (symmetric property) does not hold for GS even $s = 2$.

Consider the convex optimization model:

$$\begin{array}{ll} \min & \theta(y_1) + f(y_1, y_2, \ldots, y_s) \\ \text{s.t.} & \mathcal{A}_1^* y_1 + \mathcal{A}_2^* y_2 + \cdots + \mathcal{A}_s^* y_s = c\,. \end{array} \tag{2}$$

Linear mappings: $\mathcal{A}_i,\ i = 1, \ldots, s,\ \mathcal{A}^* y = \sum_{i=1}^s \mathcal{A}_i^* y_i,\ y := (y_1, \ldots, y_s)$.
Closed proper convex function $\theta : \mathcal{Y}_1 \to (-\infty, +\infty]$ and convex quadratic function $f(y) = \frac{1}{2}\langle y,\ \mathcal{Q}y\rangle - \langle b,\ y\rangle$. Then, (3) can be written compactly as

$$\min\{\theta(y_1) + f(y) \mid \mathcal{A}^* y = c\},$$

which is a very general CCQP.
Given $\sigma > 0$, the augmented Lagrangian function of the CCQP is

$$\mathcal{L}_\sigma(y; x) = \theta(y_1) + \underbrace{f(y) + \langle x,\ \mathcal{A}^* y - c\rangle + \frac{\sigma}{2}\|\mathcal{A}^* y - c\|^2}_{\text{quadratic}}\,.$$

The proximal augmented Lagrangian method (pALM) for the CCQP:

Given (y^0, x^0) in the domain and $\tau \in (0, 2)$. For $k = 0, 1, \ldots$

Step 1. $y^{k+1} \approx \arg\min \mathcal{L}_\sigma(y; x^k) + \frac{1}{2}\|y - y^k\|^2_{\mathcal{T}}$

$$= \arg\min_y \left\{\theta(y_1) + f(y) + \langle x^k, \mathcal{A}^* y - c\rangle + \frac{\sigma}{2}\|\mathcal{A}^* y - c\|^2 + \frac{1}{2}\|y - y^k\|^2_{\mathcal{T}}\right\}.$$

Step 2. $x^{k+1} = x^k + \tau\sigma(\mathcal{A}^* y^{k+1} - c)$.

- $\mathcal{T}$ is the block sGS decomposition operator of $\mathcal{Q} + \sigma\mathcal{A}\mathcal{A}^*$, which does not need to be formulated explicitly. Note that $\mathcal{T} \succeq 0$ but $\mathcal{T} \not\succ 0$. So it is not a classical pALM, but a "semiproximal" ALM.
- y^{k+1} is obtained via the inexact block sGS procedure [s blocks in total].
- In practice, the dual step-length τ is often chosen in $[1.618, 1.95]$, e.g, $\tau = 1.9$.

Linearly constrained CCQP (LCCQP-2)

Consider the LCCQP with two non-quadratic blocks:

$$\begin{array}{ll} \min & \theta(y_1) + q(y_1, \ldots, y_s) + \phi(z_1) + h(z_1, \ldots, z_t) \\ \text{s.t.} & \mathcal{A}_1^* y_1 + \cdots + \mathcal{A}_s^* y_s + \mathcal{B}_1^* z_1 + \cdots + \mathcal{B}_t^* z_t = c \end{array} \tag{3}$$

linear mappings: $\mathcal{A}_i,\ i = 1, \ldots, s,\ \mathcal{A}^* y = \sum_{i=1}^s \mathcal{A}_i^* y_i,\ y := (y_1, \ldots, y_s)$
linear mappings: $\mathcal{B}_j,\ j = 1, \ldots, t,\ \mathcal{B}^* z = \sum_{i=1}^t \mathcal{B}_i^* z_i,\ z := (z_1, \ldots, z_t)$
closed proper convex functions $\theta : \mathcal{Y}_1 \to (-\infty, \infty]$, $\phi : \mathcal{Z}_1 \to (-\infty, \infty]$
convex quadratics: $q(y) = \frac{1}{2}\langle y,\ \mathcal{Q}y\rangle - \langle b,\ y\rangle$, $h(z) = \frac{1}{2}\langle z,\ \mathcal{H}z\rangle - \langle d,\ z\rangle$
Write (3) compactly as

$$\min\{\theta(y_1) + q(y) + \phi(z_1) + h(z) \mid \mathcal{A}^* y + \mathcal{B}^* z = c\}.$$

Given $\sigma > 0$, the associate augmented Lagrangian function is

$$\mathcal{L}_\sigma(y, z; x) = \theta(y_1) + q(y) + \phi(z_1) + h(z) + \frac{\sigma}{2}\|\mathcal{A}^* y + \mathcal{B}^* z - c + x/\sigma\|^2 - \frac{\|x\|^2}{2\sigma}$$

sGS-pADMM for multi-block LCCQP-2

Given (y^0, z^0, x^0) in the domain. Iterate

Step 1. $y^{k+1} \approx \arg\min_y \left\{ \mathcal{L}_\sigma(y, z^k; x^k) + \frac{1}{2}\|y - y^k\|_{\mathcal{T}}^2 \right\}$

$$= \underset{y=(y_1,y_2,\ldots,y_s)}{\arg\min} \left\{ \theta(y_1) + \frac{1}{2}\langle y, (\mathcal{Q} + \sigma\mathcal{A}\mathcal{A}^*)y\rangle - \langle r^k, y\rangle + \frac{1}{2}\|y - y^k\|_{\mathcal{T}}^2 \right\}.$$

Step 2. $z^{k+1} \approx \arg\min_z \left\{ \mathcal{L}_\sigma(y^{k+1}, z; x^k) + \frac{1}{2}\|z - z^k\|_{\mathcal{S}}^2 \right\}$

$$= \underset{z=(z_1,z_2,\ldots,z_t)}{\arg\min} \left\{ \phi(z_1) + \frac{1}{2}\langle z, (\mathcal{H} + \sigma\mathcal{B}\mathcal{B}^*)z\rangle - \langle s^k, z\rangle + \frac{1}{2}\|z - z^k\|_{\mathcal{S}}^2 \right\}.$$

Step 3. $x^{k+1} = x^k + \tau\sigma(\mathcal{A}^* y^{k+1} + \mathcal{B}^* z^{k+1} - c)$, where step-length $\tau \in (0, \frac{1+\sqrt{5}}{2})$.

- $\mathcal{T}$ = block sGS operator of $\mathcal{Q} + \sigma\mathcal{A}\mathcal{A}^*$,
  $\mathcal{S}$ = block sGS operator of $\mathcal{H} + \sigma\mathcal{B}\mathcal{B}^*$.
- $y^{k+1} = (y_1, y_2, \ldots, y_s)^{k+1}$ is obtained via one cycle of the inexact block sGS iteration. Similarly for $z^{k+1} = (z_1, z_2, \ldots, z_t)^{k+1}$.
- sGS-pADMM has well developed convergence guarantee.

Doubly nonnegative semidefinite programming (DNN-SDP)

Define $\mathcal{N} = \{X \in \mathbb{S}^n \mid X \geq 0\}$ (cone of nonnegative matrices)

$$\min \Big\{ \langle C,\, X\rangle \mid \mathcal{A}(X) = b,\; X \in \mathbb{S}^n_+,\; X \in \mathcal{N} \Big\}$$

$$\text{(dual)} \qquad -\min \Big\{ \delta_{\mathbb{S}^n_+}(S) - \langle b,\, y\rangle + \delta_{\mathcal{N}}(Z) \mid S + \mathcal{A}^* y + Z = C \Big\}$$

Input $(y_0, S_0; X_0)$. For $l = 0, 1, \ldots,$ let $\widehat{C}_l = C - \sigma^{-1} X_l$

[1a] $\widehat{\mathbf{y}}_{l+1} \approx \text{argmin}_{y \in \mathbb{R}^m} \{\mathcal{L}_\sigma(y, S_l, Z_l; X_l)\} \longrightarrow$ solve $\mathcal{A}\mathcal{A}^* y = \text{rhs}$

[1b] $S_{l+1} = \text{argmin}_{S \in \mathbb{S}^n_+} \{\mathcal{L}_\sigma(\widehat{\mathbf{y}}_{l+1}, S, Z_l; X_l)\} = \Pi_{\mathbb{S}^n_+}(\widehat{C}_l - \mathcal{A}^* \widehat{\mathbf{y}}_{l+1} - Z_l)$

[1c] $y_{l+1} \approx \text{argmin}_{y \in \mathbb{R}^m} \{\mathcal{L}_\sigma(y, S_{l+1}, Z_l; X_l)\} \longrightarrow$ solve $\mathcal{A}\mathcal{A}^* y = \text{rhs}$

[2] $Z_{l+1} = \text{argmin}_{Z \in \mathcal{N}} \{\mathcal{L}_\sigma(y_{l+1}, S_{l+1}, Z; X_l)\} = \Pi_{\mathcal{N}}(\widehat{C}_l - \mathcal{A}^* y_{l+1} - S_{l+1})$

[3] $X_{l+1} = X_l + \tau\sigma(\mathcal{A}^* y_{l+1} + S_{l+1} + Z_{l+1} - C), \quad \tau \in (0, \frac{1+\sqrt{5}}{2})$

sGS-pADMM is a convergent enhancement [S., Toh, Liuqin Yang, SIOPT 2015] of the sequential Gauss-Seidel ADMM whose convergence is not guaranteed. [one can swop the positions of S and Z]

Numerical performance

Number of problems solved to the accuracy of 10^{-6} in relative KKT residual.

problem set (No.) \ solver	sGS-ADMM	SDPAD	2EBD	ADMM3g
θ_+ (58)	58	58	56	54
FAP (7)	7	7	7	7
QAP (95)	39	30	16	28
BIQ (134)	134	134	134	130
RCP (120)	120	114	109	113
Total (414)	358	343	322	332

Comparison of sGS-ADMM, SDPAD, ADMM3g and 2EBD

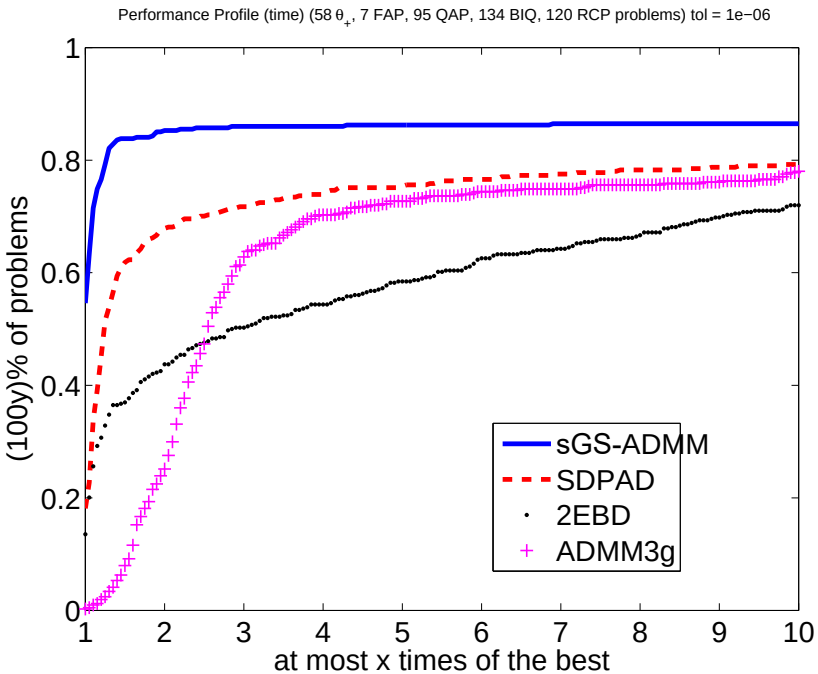

Figure: Performance profiles (time).

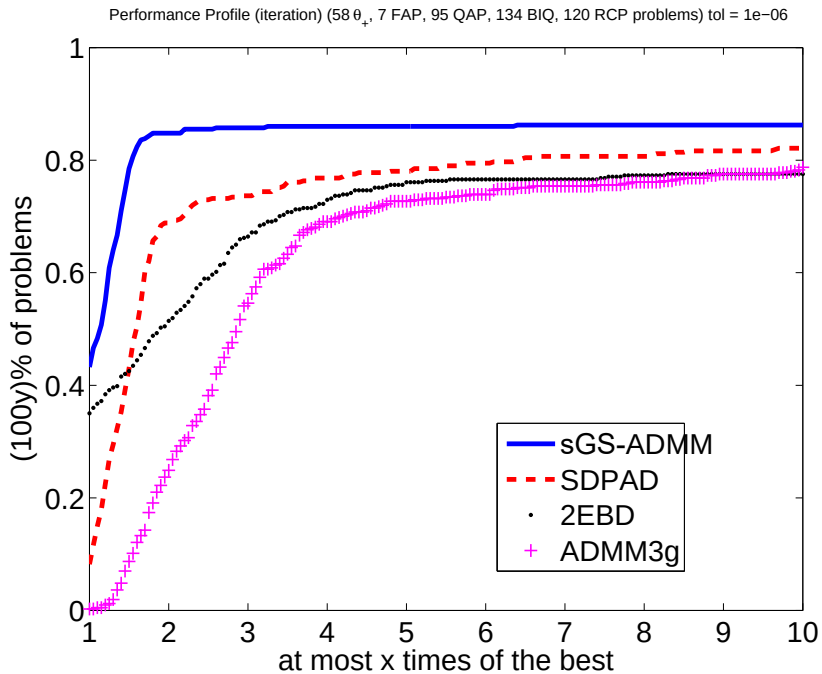

Figure: Performance profiles (iteration).

Observations and Extensions

- In the sGS procedure, the coefficient matrices of the linear systems to be solved only need to be factorized once at the start of the procedure. The additional costs of the repetitions are minimal and can be offset by the larger step length $\tau \in (0, 2)$ or $\tau \in (0, \frac{1+\sqrt{5}}{2})$.
- There are many applications that can be "solved" via block sGS + pALM or block sGS+ pADMM if the solution accuracy is not a big concern.
- More extensions can be done. For example, for convex quadratic semidefinite programming, and other convex composite conic programming problems.
- To make the algorithms even faster, we often introduce indefinite proximal terms with guaranteed convergence [Liang Chen, S., Toh, Ning Zhang, JCM 2019]
- We need something more than the "sGS + pALM" (or "pADMM)" – semismooth Newton methods (SDPNAL+ solves all the above 414 examples)

One very simple example: a convex QP with Birkhoff

Convex QP:

$$(\mathbf{P}) \quad \min \left\{ \frac{1}{2}\langle X,\, \mathcal{Q}X\rangle + \langle G,\, X\rangle \mid X \in \mathfrak{B}_n \right\},$$

self-adjoint linear operator $\mathcal{Q} \succeq 0$ and the Birkhoff polytope:

$$\mathfrak{B}_n := \{X \in \Re^{n\times n} \mid Xe = e,\, X^T e = e, X \geq 0\}$$

$e \in \Re^n$: the vector of all ones.

$$(\mathbf{D}) \min \left\{ \delta^*_{\mathfrak{B}_n}(Z) + \frac{1}{2}\langle W,\, \mathcal{Q}W\rangle \mid Z + \mathcal{Q}W + G = 0,\, W \in \mathrm{Range}(\mathcal{Q}) \right\}$$

$\delta^*_{\mathfrak{B}_n}$: the conjugate of the indicator function $\delta_{\mathfrak{B}_n}$ [Li, S., Toh, MP 2020]

ALM function for (**D**), given $\sigma > 0$

$$\mathcal{L}_\sigma(Z, W; X) = \delta^*_{\mathfrak{B}_n}(Z) + \frac{1}{2}\langle W,\, \mathcal{Q}W\rangle - \langle X,\, Z + \mathcal{Q}W + G\rangle + \frac{\sigma}{2}\|Z + \mathcal{Q}W + G\|^2$$

ALM for (D)

Algorithm ALM: An augmented Lagrangian method for (D).

Given $\sigma_0 > 0$, iterates $k = 0, 1, \ldots$

Step 1. Compute

$$(Z^{k+1}, W^{k+1}) \approx \operatorname{argmin} \left\{ \begin{array}{l} \Psi_k(Z, W) := \mathcal{L}_{\sigma_k}(Z, W; X^k) \\ \mid \ (Z, W) \in \Re^{n \times n} \times \operatorname{Range}(\mathcal{Q}) \end{array} \right\}.$$

Step 2. Compute

$$X^{k+1} = X^k - \sigma_k(Z^{k+1} + \mathcal{Q}W^{k+1} + G).$$

Update $\sigma_{k+1} \uparrow \sigma_\infty \leq \infty$.

Convex piecewise linear-quadratic minimization (SDP is more complicated):

error bound holds $\Longrightarrow$ ALM converges asymptotically superlinearly

Semismooth Newton-CG method for inner problem

For any $W \in \mathrm{Range}(\mathcal{Q})$,

$$\psi(W) := \inf_Z \mathcal{L}_\sigma(Z, W; \widehat{X}), \quad Z(W) := \widehat{X} - \sigma(\mathcal{Q}W + G)$$

Subproblem solution $(\overline{Z}, \overline{W})$:

$$\overline{W} = \arg\min \{\psi(W) \mid W \in \mathrm{Range}(\mathcal{Q})\},$$
$$\overline{Z} = \sigma^{-1}\big(Z(\overline{W}) - \Pi_{\mathfrak{B}_n}(Z(\overline{W}))\big)$$

For all $W \in \mathrm{Range}(\mathcal{Q})$,

$$\nabla\psi(W) = \mathcal{Q}(W - \Pi_{\mathfrak{B}_n}(Z(W)))$$

Semismooth Newton CG solves nonsmooth piecewise affine equation

$$\nabla\psi(W) = 0, \quad W \in \mathrm{Range}(\mathcal{Q}).$$

Semismooth Newton-CG method for inner problem

Given $\widehat{W}$, linear operator $\mathcal{M} : \Re^{n\times n} \to \Re^{n\times n}$

$$\mathcal{M}(\Delta W) := \mathcal{Q}(\mathcal{I} + \sigma P_{HS}\mathcal{Q})\Delta W, \quad \forall \Delta W \in \Re^{n\times n}$$

P_{HS}: the HS-Jacobian of $\Pi_{\mathfrak{B}_n}$ at $Z(\widehat{W})$

j-th iter., solve linear system (CG)

$$\mathcal{M}_j dW + \nabla\psi(W^j) = 0, \quad dW \in \mathrm{Range}(\mathcal{Q})$$

Global convergence: Line search (using $\psi(W)$)

Local convergence:
positive definiteness of $\mathcal{M}$ on $\mathrm{Range}(\mathcal{Q})$ $\Longrightarrow$ at least superlinear

Numerical results

Given $A, B \in \mathcal{S}^n$, quadratic assignment problem (QAP):

$$\min\{\langle X,\, AXB\rangle \mid X \in \{0,1\}^{n\times n} \cap \mathfrak{B}_n\}$$

Convex relaxation [Anstreicher et al. MP, 2001]:

$$\min\{\langle X,\, \mathcal{Q}X\rangle \mid X \in \mathfrak{B}_n\}$$

Self-adjoint linear operator $\mathcal{Q}(X) := AXB - SX - XT, \quad \mathcal{Q} \succeq 0$

Matrices $S, T \in \mathcal{S}^n$ obtained from [Anstreicher et al. MP, 2001]

Relative KKT residual:

$$\eta = \frac{\|X - \Pi_{\mathfrak{B}_n}(X - \mathcal{Q}X)\|}{1 + \|X\| + \|\mathcal{Q}X\|}$$

Matrices A, B from QAPLIB

Numerical results for QAP

"a": Gurobi, "b": ALM

		iter	η	time
problem	n	a | b (itersub)	a|b	a|b
lipa80a	80	11 | 25 (68)	1.3-6 | 7.3-8	2:46 | 01
lipa90a	90	11 | 20 (54)	2.7-6 | 8.8-8	5:32 | 01
sko100a	100	14 | 26 (95)	8.5-6 | 8.5-8	2:06 | 11
tai100a	100	11 | 18 (52)	1.3-6 | 9.5-8	10:31 | 02
tai100b	100	11 | 27 (98)	1.3-6 | 9.1-8	10:31 | 13
tai80b	80	11 | 27 (98)	1.2-6 | 8.5-8	2:36 | 07
tai256c	256	* | 2 (4)	* | 2.1-16	* | 00
tai150b	150	19 | 27 (94)	4.3-7 | 9.3-8	2:46:17 | 13
tho150	150	16 | 24 (96)	5.6-6 | 9.9-8	18:52 | 22

"*": Gurobi out of memory (128 G RAM)

"tai150b": Gurobi reports error, "small positive term" needed

Conclusions

- SDPNAL/SDPNAL+ and QSDPNAL: two-phase augmented Lagrangian methods for solving large scale SDP and convex quadratic SDPs, which are publically available.
- Primal constraint non-degeneracy is the key for using the semismooth Newton methods successfully (e.g., the NCM problem); the quadratic growth condition is crucial for the fast convergence of the (dual) ALM; and the primal-dual errors bounds can easily fail[6]
- We are still at the very early stages of solving large scale SDPs and beyond: more theory, more algorithms, and more software packages are needed.

[6] Ying Cui and S., and Toh, "On the R-superlinear convergence of the KKT residuals generated by the augmented Lagrangian method for convex composite conic programming, Mathematical Programming 178 (2019) 381–415

Thank you for your attention!