ALM, ADMM, and Randomization – Managing Randomness in Optimization Algorithms

Yinyu Ye

Department of Management Science and Engineering

Stanford University

Stanford, CA 94305, U.S.A.

http://www.stanford.edu/~yyye

Recall the Lagrangian Functions

We consider

$$f^* := \min \quad f(\mathbf{x}) \quad \text{s.t.} \quad \mathbf{h}(\mathbf{x}) = \mathbf{0},\ \mathbf{x} \in X. \tag{1}$$

Recall that the Lagrangian function:

$$L(\mathbf{x}, \mathbf{y}) = f(\mathbf{x}) - \mathbf{y}^T \mathbf{h}(\mathbf{x}).$$

and the dual function:

$$\phi(\mathbf{y}) = \min_{\mathbf{x} \in X} L(\mathbf{x}, \mathbf{y}); \tag{2}$$

and the dual problem

$$(f^* \geq)\phi^* := \max \quad \phi(\mathbf{y}). \tag{3}$$

In many cases, one can find $\mathbf{y}^*$ of dual problem (3), a unconstrained optimization problem; then go ahead to find $\mathbf{x}^*$ using (2).

The Gradient and Hessian of ϕ

Let $\mathbf{x}(\mathbf{y})$ be a minimizer of (2). Then

$$\phi(\mathbf{y}) = f(\mathbf{x}(\mathbf{y})) - \mathbf{y}^T \mathbf{h}(\mathbf{x}(\mathbf{y}))$$

Thus,

$$\begin{aligned}
\nabla \phi(\mathbf{y}) &= \nabla f(\mathbf{x}(\mathbf{y}))^T \nabla \mathbf{x}(\mathbf{y}) - \mathbf{y}^T \nabla \mathbf{h}(\mathbf{x}(\mathbf{y})) \nabla \mathbf{x}(\mathbf{y}) - \mathbf{h}(\mathbf{x}(\mathbf{y})) \\
&= (\nabla f(\mathbf{x}(\mathbf{y}))^T - \mathbf{y}^T \nabla \mathbf{h}(\mathbf{x}(\mathbf{y}))) \nabla \mathbf{x}(\mathbf{y}) - \mathbf{h}(\mathbf{x}(\mathbf{y})) \\
&= -\mathbf{h}(\mathbf{x}(\mathbf{y})).
\end{aligned}$$

Similarly, we can derive

$$\nabla^2 \phi(\mathbf{y}) = -\nabla \mathbf{h}(\mathbf{x}(\mathbf{y})) \left(\nabla^2_{\mathbf{x}} L(\mathbf{x}(\mathbf{y}), \mathbf{y}) \right)^{-1} \nabla \mathbf{h}(\mathbf{x}(\mathbf{y}))^T,$$

where $\nabla^2_{\mathbf{x}} L(\mathbf{x}(\mathbf{y}), \mathbf{y})$ is the Hessian of the Lagrangian function that is assumed to be positive definite at any (local) minimizer.

The Toy Example

$$\text{minimize} \quad (x_1 - 1)^2 + (x_2 - 1)^2$$

$$\text{subject to} \quad x_1 + 2x_2 - 1 = 0, \quad 2x_1 + x_2 - 1 = 0.$$

$$L(\mathbf{x}, \mathbf{y}) = (x_1 - 1)^2 + (x_2 - 1)^2 - y_1(x_1 + 2x_2 - 1) - y_2(2x_1 + x_2 - 1).$$

$$x_1 = 0.5y_1 + y_2 + 1, \quad x_2 = y_1 + 0.5y_2 + 1.$$

$$\phi(\mathbf{y}) = -1.25y_1^2 - 1.25y_2^2 - 2y_1y_2 - 2y_1 - 2y_2.$$

$$\nabla\phi(\mathbf{y}) = \begin{pmatrix} 2.5y_1 + 2y_2 + 2 \\ 2y_1 + 2.5y_2 1 + 2 \end{pmatrix},$$

$$\nabla^2\phi(\mathbf{y}) = -\begin{pmatrix} 1 & 2 \\ 2 & 1 \end{pmatrix} \begin{pmatrix} 2 & 0 \\ 0 & 2 \end{pmatrix}^{-1} \begin{pmatrix} 1 & 2 \\ 2 & 1 \end{pmatrix}^T = -\begin{pmatrix} 2.5 & 2 \\ 2 & 2.5 \end{pmatrix}$$

The Augmented Lagrangian Function

In both theory and practice, we actually consider an Augmented Lagrangian function (ALF)

$$L_{\mathcal{A}}(\mathbf{x},\mathbf{y}) = f(\mathbf{x}) - \mathbf{y}^T\mathbf{h}(\mathbf{x}) + \frac{\beta}{2}\|\mathbf{h}(\mathbf{x})\|^2,$$

which corresponds to an equivalent problem of (1):

$$f^* := \min \quad f(\mathbf{x}) + \tfrac{\beta}{2}\|\mathbf{h}(\mathbf{x})\|^2 \quad \text{s.t.} \quad \mathbf{h}(\mathbf{x}) = \mathbf{0},\ \mathbf{x} \in X.$$

Note that, although at feasibility the additional square term in objective is redundant, it helps to improve strict convexity of the Lagrangian function.

The Augmented Lagrangian Dual

Now the dual function:

$$\phi_{\mathcal{A}}(\mathbf{y}) = \min_{\mathbf{x} \in X} L_{\mathcal{A}}(\mathbf{x}, \mathbf{y}); \tag{4}$$

and the dual problem

$$(f^* \geq)\phi^*_{\mathcal{A}} := \max \quad \phi_{\mathcal{A}}(\mathbf{y}). \tag{5}$$

Note that the dual function satisfies $\frac{1}{\beta}$-Lipschitz condition (see Chapter 14 of L&Y).

For the convex optimization case,

$$\mathbf{h}(\mathbf{x}) = A\mathbf{x} - \mathbf{b}$$

we have

$$\nabla^2 L_{\mathcal{A}}(\mathbf{x}, \mathbf{y}) = \nabla^2 f(\mathbf{x}) + \beta(A^T A).$$

The Augmented Lagrangian Method

Augmented Lagrangian Method (ALM):

Start from any $(\mathbf{x}^0 \in X, \mathbf{y}^0)$, we compute a new iterate pair

$$\mathbf{x}^{k+1} = \arg\min_{\mathbf{x} \in X} L_{\mathcal{A}}(\mathbf{x}, \mathbf{y}^k), \text{ then } \mathbf{y}^{k+1} = \mathbf{y}^k - \beta \mathbf{h}(\mathbf{x}^{k+1}).$$

The calculation of $\mathbf{x}$ is used to compute the gradient vector of $\phi_{\mathcal{A}}(\mathbf{y})$, which is a steepest ascent direction.

The method converges just like the Steepest Descent Method (SDM), because the dual function satisfies $\frac{1}{\beta}$-Lipschitz condition.

Other SDM strategies may be adapted to update $\mathbf{y}$ (the Accelerated SDM, Conjugate, Quasi-Newton ...).

Analysis of the Augmented Lagrangian Method

Consider the convex optimization case $\mathbf{h}(\mathbf{x}) = A\mathbf{x} - \mathbf{b}$. Since $\mathbf{x}^{k+1}$ makes KKT condition:

$$\begin{aligned} \mathbf{0} &= \nabla f(\mathbf{x}^{k+1}) - A^T\mathbf{y}^k + \beta A^T(A\mathbf{x}^{k+1} - \mathbf{b}) \\ &= \nabla f(\mathbf{x}^{k+1}) - A^T(\mathbf{y}^k - \beta(A\mathbf{x}^{k+1} - \mathbf{b})) \\ &= \nabla f(\mathbf{x}^{k+1}) - A^T\mathbf{y}^{k+1}, \end{aligned}$$

we only need to be concerned about whether or not $\|A\mathbf{x}^k - \mathbf{b}\|$ converges to zero and how fast it converges. First, from the convexity of $f(\mathbf{x})$, we have

$$\begin{aligned} \mathbf{0} &\le (\nabla f(\mathbf{x}^{k+1}) - \nabla f(\mathbf{x}^k))^T(\mathbf{x}^{k+1} - \mathbf{x}^k) \\ &= (-A^T\mathbf{y}^{k+1} + A^T\mathbf{y}^k)^T(\mathbf{x}^{k+1} - \mathbf{x}^k) \\ &= (\mathbf{y}^{k+1} - \mathbf{y}^k)^T(A\mathbf{x}^{k+1} - A\mathbf{x}^k) \\ &= -\beta(A\mathbf{x}^{k+1} - \mathbf{b})(A\mathbf{x}^{k+1} - \mathbf{b} - (A\mathbf{x}^k - \mathbf{b})), \end{aligned}$$

which implies that $\|A\mathbf{x}^{k+1} - \mathbf{b}\| \le \|A\mathbf{x}^k - \mathbf{b}\|$, that is, the error is non-increasing.

Again, from the convexity, we have

$$\begin{aligned}\mathbf{0} &\le (\nabla f(\mathbf{x}^{k+1}) - \nabla f(\mathbf{x}^*))^T(\mathbf{x}^{k+1} - \mathbf{x}^*)\\ &= (A^T\mathbf{y}^{k+1} - A^T\mathbf{y}^*)^T(\mathbf{x}^{k+1} - \mathbf{x}^*)\\ &= (\mathbf{y}^{k+1} - \mathbf{y}^*)^T(A\mathbf{x}^{k+1} - A\mathbf{x}^*) = (\mathbf{y}^{k+1} - \mathbf{y}^*)^T(A\mathbf{x}^{k+1} - \mathbf{b})\\ &= \tfrac{1}{\beta}(\mathbf{y}^{k+1} - \mathbf{y}^*)^T(\mathbf{y}^k - \mathbf{y}^{k+1}).\end{aligned}$$

Thus, from the positivity of the cross product, we have

$$\begin{aligned}\|\mathbf{y}^k - \mathbf{y}^*\|^2 &= \|\mathbf{y}^k - \mathbf{y}^{k+1} + \mathbf{y}^{k+1} - \mathbf{y}^*\|^2\\ &\ge \|\mathbf{y}^k - \mathbf{y}^{k+1}\|^2 + \|\mathbf{y}^{k+1} - \mathbf{y}^*\|^2\\ &= \beta\|A\mathbf{x}^{k+1} - \mathbf{b}\|^2 + \|\mathbf{y}^{k+1} - \mathbf{y}^*\|^2.\end{aligned}$$

Sum up from 0 to k of the inequality we have

$$\begin{aligned}\|\mathbf{y}^0 - \mathbf{y}^*\|^2 &\ge \|\mathbf{y}^{k+1} - \mathbf{y}^*\|^2 + \beta\sum_{l=0}^{k}\|A\mathbf{x}^{l+1} - \mathbf{b}\|^2\\ &\ge \beta\sum_{l=0}^{k}\|A\mathbf{x}^{l+1} - \mathbf{b}\|^2\\ &\ge (k+1)\beta\|A\mathbf{x}^{k+1} - \mathbf{b}\|^2.\end{aligned}$$

The Alternating Direction Method with Multipliers

For the ADMM method, we consider structured problem

$$\min \quad f_1(\mathbf{x}_1) + f_2(\mathbf{x}_2) \quad \text{s.t.} \quad A_1\mathbf{x}_1 + A_2\mathbf{x}_2 = \mathbf{b},\ \mathbf{x}_1 \in X_1, \mathbf{x}_2 \in X_2 \tag{6}$$

where $f_1(\mathbf{x}_1)$ and $f_2(\mathbf{x}_2)$ are convex closed proper functions, and $\mathcal{X}_1$ and $\mathcal{X}_2$ are convex sets.

Original ADMM (Glowinski $\&$ Marrocco '75, Gabay $\&$ Mercier '76):

$$\begin{cases} \mathbf{x}_1^{k+1} = \arg\min\{L_{\mathcal{A}}(\mathbf{x}_1, \mathbf{x}_2^k, \mathbf{y}^k) \,|\, \mathbf{x}_1 \in X_1\}, \\ \mathbf{x}_2^{k+1} = \arg\min\{L_{\mathcal{A}}(\mathbf{x}_1^{k+1}, \mathbf{x}_2, \mathbf{y}^k) \,|\, \mathbf{x}_2 \in X_2\}, \\ \mathbf{y}^{k+1} = \mathbf{y}^k - \beta(A_1\mathbf{x}_1^{k+1} + A_2\mathbf{x}_2^{k+1} - \mathbf{b}). \end{cases}$$

where theAugmented Lagrangian function $L_{\mathcal{A}}$ again is

$$L_{\mathcal{A}}(\mathbf{x}_1, \mathbf{x}_2, \mathbf{y}) = \sum_{i=1}^{2} f_i(\mathbf{x}_i) - \mathbf{y}^T\Big(\sum_{i=1}^{2} A_i\mathbf{x}_i - \mathbf{b}\Big) + \frac{\beta}{2}\Big\|\sum_{i=1}^{2} A_i\mathbf{x}_i - \mathbf{b}\Big\|^2.$$

Again, one can prove that the iterates converge with the same speed.

Direct Application of ADMM to Dual Linear Programming I

Consider the dual LP

$$\begin{array}{ll} \text{maximize}_{(\mathbf{y},\mathbf{s})} & \mathbf{b}^T\mathbf{y} \\ \text{s.t.} & A^T\mathbf{y}+\mathbf{s}=\mathbf{c},\ \mathbf{s}\geq\mathbf{0}. \end{array}$$

The augmented Lagrangian function would be

$$L_{\mathcal{A}}(\mathbf{y},\mathbf{s},\mathbf{x}) = -\mathbf{b}^T\mathbf{y} - \mathbf{x}^T(A^T\mathbf{y}+\mathbf{s}-\mathbf{c}) + \frac{\beta}{2}\|A^T\mathbf{y}+\mathbf{s}-\mathbf{c}\|^2,$$

where β is a positive parameter, and $\mathbf{x}$ is the multiplier vector.

Direct Application of ADMM to Dual Linear Programming II

The ADMM for the dual is straightforward: starting from any $\mathbf{y}^0$, $\mathbf{s}^0 \geq \mathbf{0}$, and multiplier $\mathbf{x}^0$,

- Update variable $\mathbf{y}$:
$$\mathbf{y}^{k+1} = \arg\min_{\mathbf{y}} L(\mathbf{y}, \mathbf{s}^k, \mathbf{x}^k);$$
- Update slack variable $\mathbf{s}$:
$$\mathbf{s}^{k+1} = \arg\min_{\mathbf{s}\geq\mathbf{0}} L(\mathbf{y}^{k+1}, \mathbf{s}, \mathbf{x}^k);$$
- Update multipliers $\mathbf{x}$:
$$\mathbf{x}^{k+1} = \mathbf{x}^k - \beta(A^T\mathbf{y}^{k+1} + \mathbf{s}^{k+1} - \mathbf{c}).$$

Note that the updates of $\mathbf{y}$ is a least-squares problem with constant matrix, and the update of $\mathbf{s}$ has a simple close form. (Also note that $\mathbf{x}$ would be non-positive at the end, since we changed maximization to minimization of the dual.)

To split $\mathbf{y}$ into multi blocks and update cyclically in random order?

The ADMM with Three Blocks?

The ADMM method resembles the Block Coordinate Descent (BCD) Method – What about ADMM for

$$\min \quad f_1(\mathbf{x}_1) + f_2(\mathbf{x}_2) + f_3(\mathbf{x}_3) \quad \text{s.t.} \quad A_1\mathbf{x}_1 + A_2\mathbf{x}_2 + A_3\mathbf{x}_3 = \mathbf{b},$$

where the augmented Lagrangian function

$$\begin{aligned} L_{\mathcal{A}}(\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3, \mathbf{y}) = \quad & f_1(\mathbf{x}_1) + f_2(\mathbf{x}_2) + f_3(\mathbf{x}_3) - \mathbf{y}^T(A_1\mathbf{x}_1 + A_2\mathbf{x}_2 + A_3\mathbf{x}_3 - \mathbf{b}) \\ & + \frac{\beta}{2}\|A_1\mathbf{x}_1 + A_2\mathbf{x}_2 + A_3\mathbf{x}_3 - \mathbf{b}\|^2. \end{aligned}$$

Then, for any given $(\mathbf{x}_1^k, \mathbf{x}_2^k, \mathbf{x}_3^k, \mathbf{y}^k)$, the direct extension of ADMM would do

$$\begin{aligned} \mathbf{x}_1^{k+1} &= \arg\min_{\mathbf{x}_1} L_{\mathcal{A}}(\mathbf{x}_1, \mathbf{x}_2^k, \mathbf{x}_3^k, \mathbf{y}^k), \\ \mathbf{x}_2^{k+1} &= \arg\min_{\mathbf{x}_2} L_{\mathcal{A}}(\mathbf{x}_1^{k+1}, \mathbf{x}_2, \mathbf{x}_3^k, \mathbf{y}^k), \\ \mathbf{x}_3^{k+1} &= \arg\min_{\mathbf{x}_3} L_{\mathcal{A}}(\mathbf{x}_1^{k+1}, \mathbf{x}_2^{k+1}, \mathbf{x}_3, \mathbf{y}^k), \\ \mathbf{y}^{k+1} &= \mathbf{y}^k - \beta(A_1\mathbf{x}^{k+1} + A_2\mathbf{x}_2^{k+1} + A_3\mathbf{x}_3^{k+1} - \mathbf{b}). \end{aligned}$$

Does it Converge?

Not easy to analyze the convergence: the operator theory for the ADMM cannot be directly extended to the ADMM with three blocks, since the proof for two blocks breaks down for three blocks.

Existing results for convergence:

- Strong convexity; plus carefully select β in a specific range.
- Other restricted conditions on the problem, and take a sufficiently smaller step-size factor $1 > \gamma > 0$ in dual update

$$\mathbf{y}^{k+1} = \mathbf{y}^k - \gamma\beta(A_1\mathbf{x}_1^{k+1} + A_2\mathbf{x}_2^{k+1} + A_3\mathbf{x}_3^{k+1} - \mathbf{b}).$$

- Various post correction steps, which are costly.

But, these did not answer the open question whether or not the direct extension of multi-block ADMM converges under the original simple convexity assumption.

Divergent Example of the Extended ADMM I

We have recently resolved this long-standing question:

Theorem 1 *There existing an example where the direct extension of ADMM of three blocks is not necessarily convergent for any choice of* β*. Moreover, for any randomly generated initial point, ADMM diverges with probability one.*

Consider the system of homogeneous linear equations with three block where each block has a single variable with unique solution $\mathbf{x}^* = \mathbf{0}$:

$$A_1x_1 + A_2x_2 + A_3x_3 = \mathbf{0}, \text{ where } A = (A_1, A_2, A_3) = \begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 2 \\ 1 & 2 & 2 \end{pmatrix}.$$

Divergent Example of the Extended ADMM II

The ADMM with $\beta = 1$ is a linear matrix mapping

$$\begin{pmatrix} 3 & 0 & 0 & 0 & 0 & 0 \\ 4 & 6 & 0 & 0 & 0 & 0 \\ 5 & 7 & 9 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 2 & 0 & 1 & 0 \\ 1 & 2 & 2 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \mathbf{x}^{k+1} \\ \mathbf{y}^{k+1} \end{pmatrix} = \begin{pmatrix} 0 & -4 & -5 & 1 & 1 & 1 \\ 0 & 0 & -7 & 1 & 1 & 2 \\ 0 & 0 & 0 & 1 & 2 & 2 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \mathbf{x}^{k} \\ \mathbf{y}^{k} \end{pmatrix}.$$

which can be reduced to

$$\begin{pmatrix} x_2^{k+1} \\ x_3^{k+1} \\ \mathbf{y}^{k+1} \end{pmatrix} = M \begin{pmatrix} x_2^{k} \\ x_3^{k} \\ \mathbf{y}^{k} \end{pmatrix},$$

where

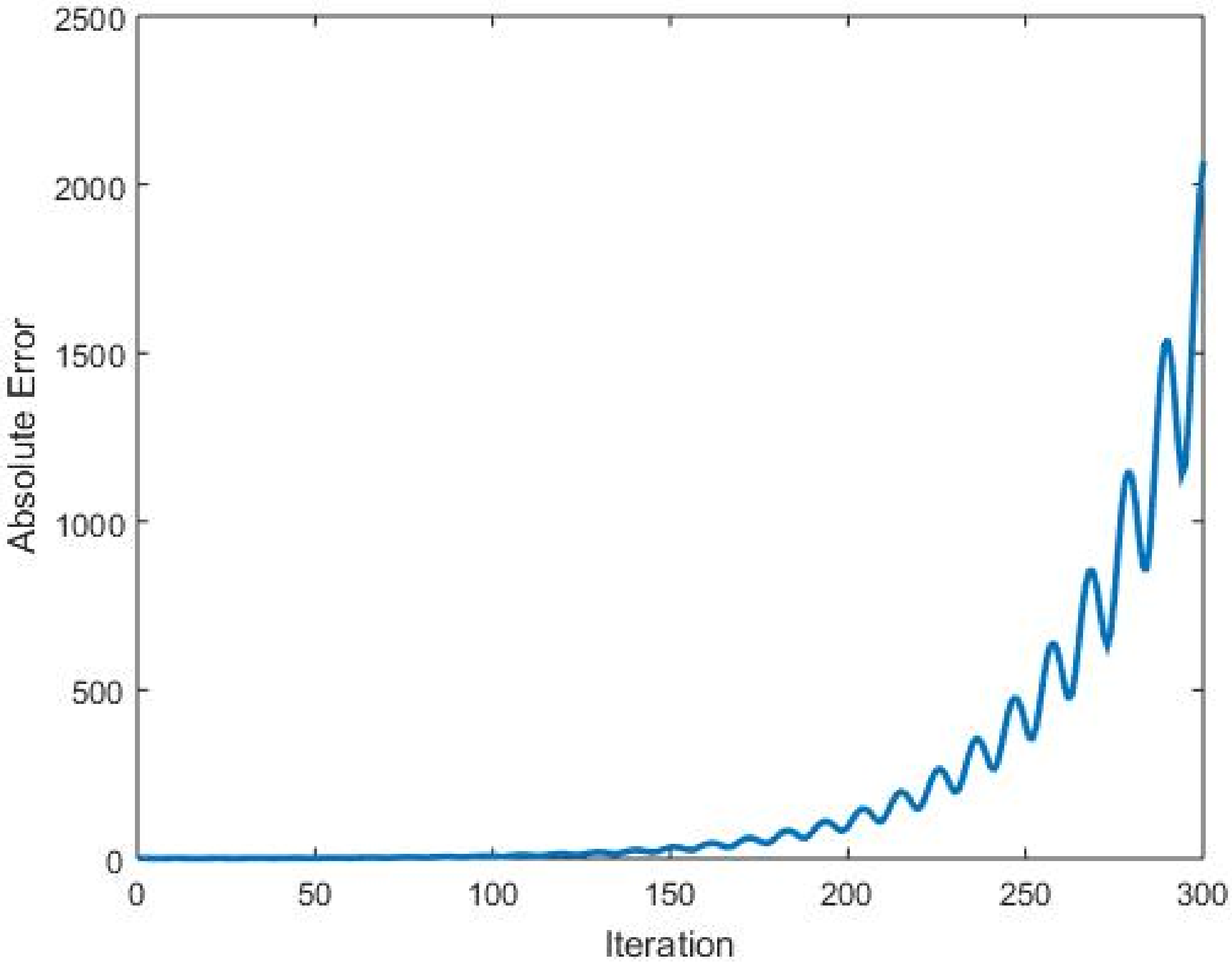
$$M = \frac{1}{162}\begin{pmatrix} 144 & -9 & -9 & -9 & 18 \\ 8 & 157 & -5 & 13 & -8 \\ 64 & 122 & 122 & -58 & -64 \\ 56 & -35 & -35 & 91 & -56 \\ -88 & -26 & -26 & -62 & 88 \end{pmatrix}.$$

The matrix $M = V\mathrm{Diag(d)V^{-1}}$ has $d = \begin{pmatrix} 0.9836 + 0.2984i \\ 0.9836 - 0.2984i \\ 0.8744 + 0.2310i \\ 0.8744 - 0.2310i \\ 0 \end{pmatrix}$. Note that $\rho(M) = |d_1| = |d_2| > 1.$

which implies that the mapping is not a contraction.

Chen, He, Y, and Yuan [*Math Programming* 2016]

Residuals vs Iteration Counts



Does Strong Convexity Help?

Consider the following example

$$\begin{array}{ll} \min & 0.05x_1^2 + 0.05x_2^2 + 0.05x_3^2 \\ \text{s.t.} & \begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 2 \\ 1 & 2 & 2 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = 0. \end{array}$$

- Then, the linear mapping matrix M in the extended ADMM ($\beta = 1$) has $\rho(M) = 1.0087 > 1$
- Therefore the directly extended ADMM still diverges
- Even for strongly convex programming, the extended ADMM is not necessarily convergent for $\beta > 0$ in a certain range.

Does the Small-Stepsize Help?

Recall that, In the small stepsized ADMM, the Lagrangian multiplier is updated by

$$\mathbf{y}^{k+1} := \mathbf{y}^k - \gamma\beta(A_1\mathbf{x}_1^{k+1} + A_2\mathbf{x}_2^{k+1} + \ldots + A_3\mathbf{x}_3^{k+1}).$$

Convergence is proved:

- One block (Augmented Lagrangian Method): $\gamma \in (0, 2)$, (Hestenes '69, Powell '69).
- Two blocks (Alternating Direction Method of Multipliers: $\gamma \in (0, \frac{1+\sqrt{5}}{2})$, (Glowinski, '84).
- Three blocks: for γ sufficiently small provided additional conditions on the problem, (Hong $\&$ Luo '12).

Question: Is there a problem-data-independent γ such that the method converges?

A Numerical Study

For any given $\gamma > 0$, consider the linear system

$$\begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 1+\gamma \\ 1 & 1+\gamma & 1+\gamma \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \mathbf{0}.$$

γ	1	0.1	1e-2	1e-3	1e-4	1e-5	1e-6	1e-7
$\rho(M)$	1.0278	1.0026	1.0001	> 1	> 1	> 1	> 1	> 1

Table 1: The radius of M

Thus, there is no practical problem-data-independent γ such that the small-stepsize ADMM variant works.

How to Make it Converge?

- There are many complicated correction method in the ADMM-type method, but ...

- Question: Is there a ”simple correction” of the ADMM for the multi-block convex minimization problems?

  Random-Permuted ADMM (RP-ADMM) for 3 blocks: in each round, draw a random permutation $\sigma = (\sigma(1), \sigma(2), \sigma(3))$ of $\{1, 2, 3\}$, and

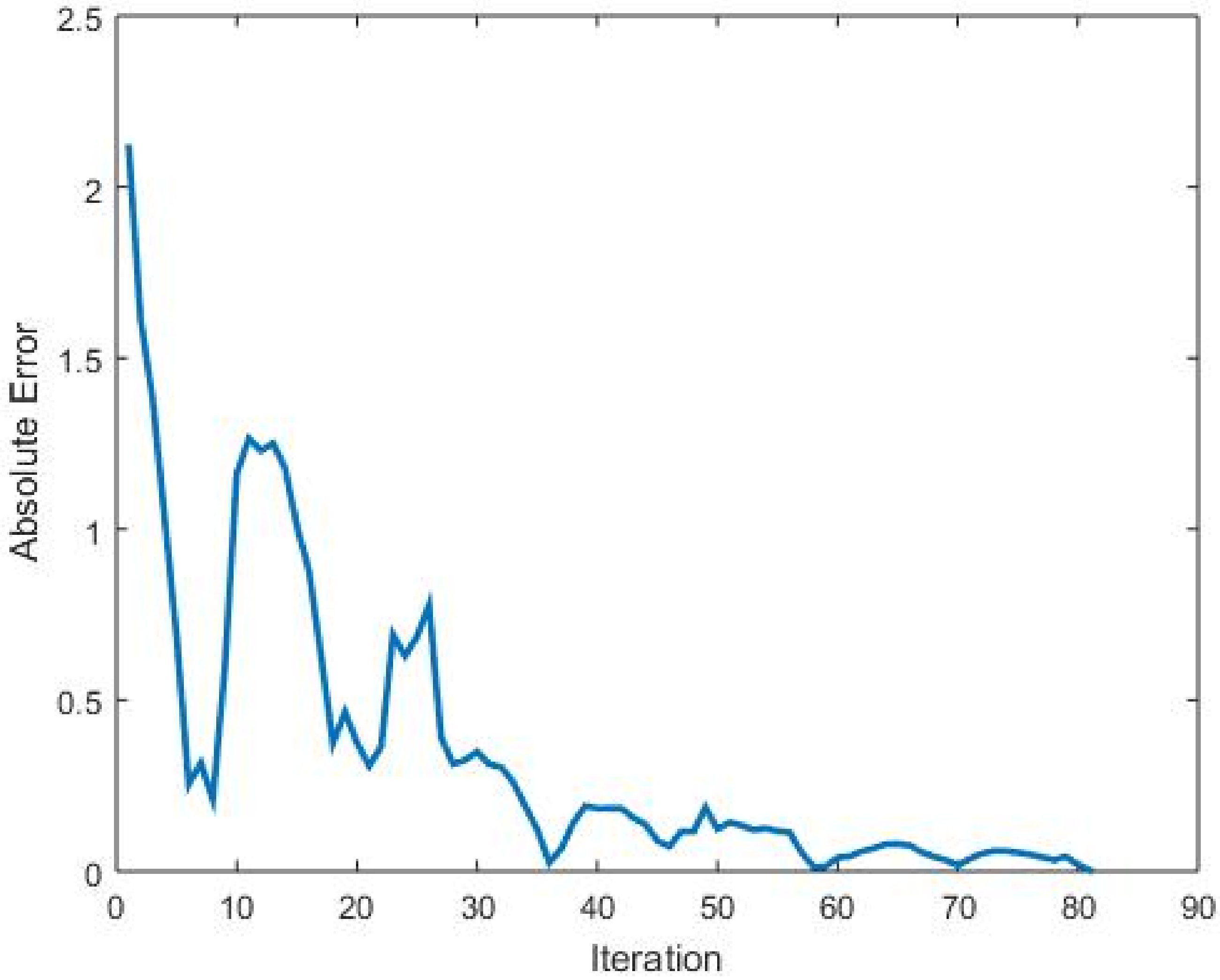
  $$\text{Update } \mathbf{x}_{\sigma(1)} \rightarrow \mathbf{x}_{\sigma(2)} \rightarrow \mathbf{x}_{\sigma(n)} \rightarrow \mathbf{y}.$$

  (This is the block sample without replacement, and there are total 6 different orderings.)

- Interpretation: Force “absolute fairness” among blocks, and make the mapping matrix more symmetric.

- Computations indicate it always works!

The Diverging Example with Random Permutation



Any Theory Behind the Success?

$$\begin{array}{rl} \min_{\mathbf{x}\in R^N} & f_1(\mathbf{x}_1)+\ldots+f_n(\mathbf{x}_n), \\ \text{s.t.} & A\mathbf{x}:=A_1\mathbf{x}_1+\cdots+A_n\mathbf{x}_n=\mathbf{b}, \\ & \mathbf{x}_i\in X_i\subset R^{d_i},\; i=1,\ldots,n. \end{array} \qquad (7)$$

$$L_{\mathcal{A}}(\mathbf{x}_1,\ldots,\mathbf{x}_n;\mathbf{y})=\sum_i f_i(x_i)-\mathbf{y}^T(\sum_i A_i\mathbf{x}_i-\mathbf{b})+\frac{\beta}{2}\|\sum_i A_i\mathbf{x}_i-\mathbf{b}\|^2$$

The Randomly Permuted Cyclic Extension Multi-block ADMM update in each round with a randomly permuted order $\sigma=(\sigma(1),\ldots,\sigma(n))$ of $\{1,\ldots,n\}$

$$\begin{array}{c} \mathbf{x}_{\sigma(1)} \longleftarrow \arg\min_{\mathbf{x}_1\in\mathcal{X}_1} L_{\mathcal{A}}(\mathbf{x}_1,\ldots,\mathbf{x}_n;\mathbf{y}), \\ \cdots \\ \mathbf{x}_{\sigma(n)} \longleftarrow \arg\min_{\mathbf{x}_n\in\mathcal{X}_n} L_{\mathcal{A}}(\mathbf{x}_1,\ldots,\mathbf{x}_n;\mathbf{y}), \\ \mathbf{y} \longleftarrow \mathbf{y}-\beta(A\mathbf{x}-\mathbf{b}), \end{array}$$

Random Permuted ADMM for Linear Systems

Consider solving a nonsingular square system of linear equations ($f_i = 0, \ \forall i$).

$$\begin{array}{ll} \min_{\mathbf{x} \in R^N} & 0, \\ \text{s.t.} & A_1 \mathbf{x}_1 + \cdots + A_n \mathbf{x}_n = \mathbf{b}, \end{array}$$

RP-ADMM generates $\mathbf{z}^k = (\mathbf{x}^k; \mathbf{y}^k)$, an r.v., depending on

$$\boldsymbol{\xi}_k = (\sigma_1, \ldots, \sigma_k), \quad \mathbf{z}^i = M_{\sigma_i} \mathbf{z}^{i-1}, \ i = 1, ..., k,$$

where σ_i is the random permutation at the i-th round.

Denote the expected iterate $\phi^k := \mathbf{E}_{\boldsymbol{\xi}_k} [\mathbf{z}^k]$

Theorem 2 *(Sun et al. [2016]) The expected output converges to the unique solution of the linear system equations any number of variables $N \geq 1$ and any block number $N \geq n \geq 1$.*

Remark: Expected convergence $\neq$ convergence, but is a strong evidence for convergence for solving most problems, e.g., when iterates are bounded.

The Average Mapping is a Contraction

- The update equation of RP-ADMM is
$$\mathbf{z}^{k+1} = M_\sigma \mathbf{z}^k,$$
where $M_\sigma \in R^{2N\times 2N}$ depend on σ.

- Define the expected update matrix as
$$M = \mathbf{E}_\sigma[M_\sigma] = \frac{1}{n!}\sum_\sigma M_\sigma.$$

Theorem 3 *(Sun et al. [2016]) The spectral radius of M, $\rho(M)$, is strictly less than 1 for any integer $N \geq 1$ and any block number $N \geq n \geq 1$.*

Remark: For A in the divergence example, $\rho(M_\sigma) > 1$ for any σ

– Averaging Helps, a Lot.

Sketch of the Proof of Theorem 2

Theorem 3 implies Theorem 2 is relatively easy to show.

For simplicity consider $n = 2$. Each iteration is

$$\text{either} \quad \mathbf{z}^{k+1} = M_1 \mathbf{z}^k \;\text{ or }\; \mathbf{z}^{k+1} = M_2 \mathbf{z}^k.$$

Therefore

$$\begin{aligned} E(\mathbf{z}^1) &= \tfrac{M_1+M_2}{2}\mathbf{z}^0 = M\mathbf{z}^0; \\ E(\mathbf{z}^2) &= \tfrac{1}{4}(M_1^2 + M_1M_2 + M_2M_1 + M_2^2)\mathbf{z}^0 = M^2\mathbf{z}^0, \\ &\ldots \\ E(\mathbf{z}^k) &= M^k\mathbf{z}^0. \end{aligned}$$

Thus, $\rho(M) < 1$ implies convergence in expectation.

Math Problem of Theorem 3

- Define

$$Q := E(L_\sigma^{-1}) = \frac{1}{n!} \sum_\sigma L_\sigma^{-1}. \tag{8}$$

- Example:

$$L_{(231)} = \begin{pmatrix} 1 & A_1^T A_2 & A_1^T A_3 \\ 0 & 1 & 0 \\ 0 & A_3^T A_2 & 1 \end{pmatrix}.$$

- Need to prove that, for all A, $\rho(M) < 1$ where

$$M = \begin{pmatrix} I - QA^T A & QA^T \\ -A + AQA^T A & I - AQA^T \end{pmatrix}.$$

Difficulties of Proving Theorem 3

- **Difficulty 1**: Few tools deal with spectral radius of non-symmetric matrices.
  - E.g. $\rho(X+Y) \le \rho(X) + \rho(Y)$ and $\rho(XY) \le \rho(X)\rho(Y)$ don't hold.
  - Though $\rho(M) < \|M\|$, it turns out $\|M\| > 2.3$ for the counterexample.
- **Difficulty 2**: M is a complicated function of A.
  - $n = 3$, let $(A^T A)_{k,l} = b_{kl}$, then $Q_{12} = -\frac{1}{2} b_{12} + \frac{1}{6} b_{13} b_{23}$.
  - $n = 4$, $Q_{12} = -\frac{1}{2!} b_{12} + \frac{1}{3!}(b_{13}b_{32} + b_{14}b_{42}) - \frac{1}{4!}(b_{13}b_{34}b_{42} + b_{14}b_{43}b_{32})$.
- **Solution**: Symmetrization and Mathematical Induction.

Two Main Lemmas to Prove Theorem 3

- **Step 1**: Relate M to a symmetric matrix AQA^T.

  Lemma 1

  $$\mathbf{y} \in \textit{eig}(M) \Longleftrightarrow \frac{(1-\mathbf{y})^2}{1-2\mathbf{y}} \in \textit{eig}(AQA^T).$$

  Since Q defined by Q def is symmetric, we have

  $$\rho(M) < 1 \Longleftrightarrow \textit{eig}(AQA^T) \subseteq (0, \frac{4}{3}).$$

- **Step 2**: Bound eigenvalues of AQA^T - prove by induction.

  Lemma 2

  $$\textit{eig}(AQA^T) \subseteq (0, \frac{4}{3}).$$

- Remark: $4/3$ is "almost" tight; for $n = 3,$ maximum ≈ 1.18. Increase to $4/3$ as n increases.

RP-ADMM for Linear Constrained Convex QP

In general, consider a convex quadratic optimization problem

$$\begin{aligned} \min_{\mathbf{x}\in R^N} \quad & \mathbf{c}_1^T\mathbf{x}_1 + \ldots + \mathbf{c}_n^T\mathbf{x}_n + \tfrac{1}{2}\mathbf{x}^TQ\mathbf{x}, \\ \text{s.t.} \quad & A\mathbf{x} := A_1\mathbf{x}_1 + \cdots + A_n\mathbf{x}_n = \mathbf{b}. \end{aligned} \tag{9}$$

Theorem 4 *Under some technical assumptions, the expected output of randomly permuted ADMM converges to the solution of the original problem for any integer* $N \geq 1$ *and any block number* $N \geq n \geq 1$..

Key Observation: The objective function of the problem is not separable so that the traditional proof of two-block ADMM does not work.

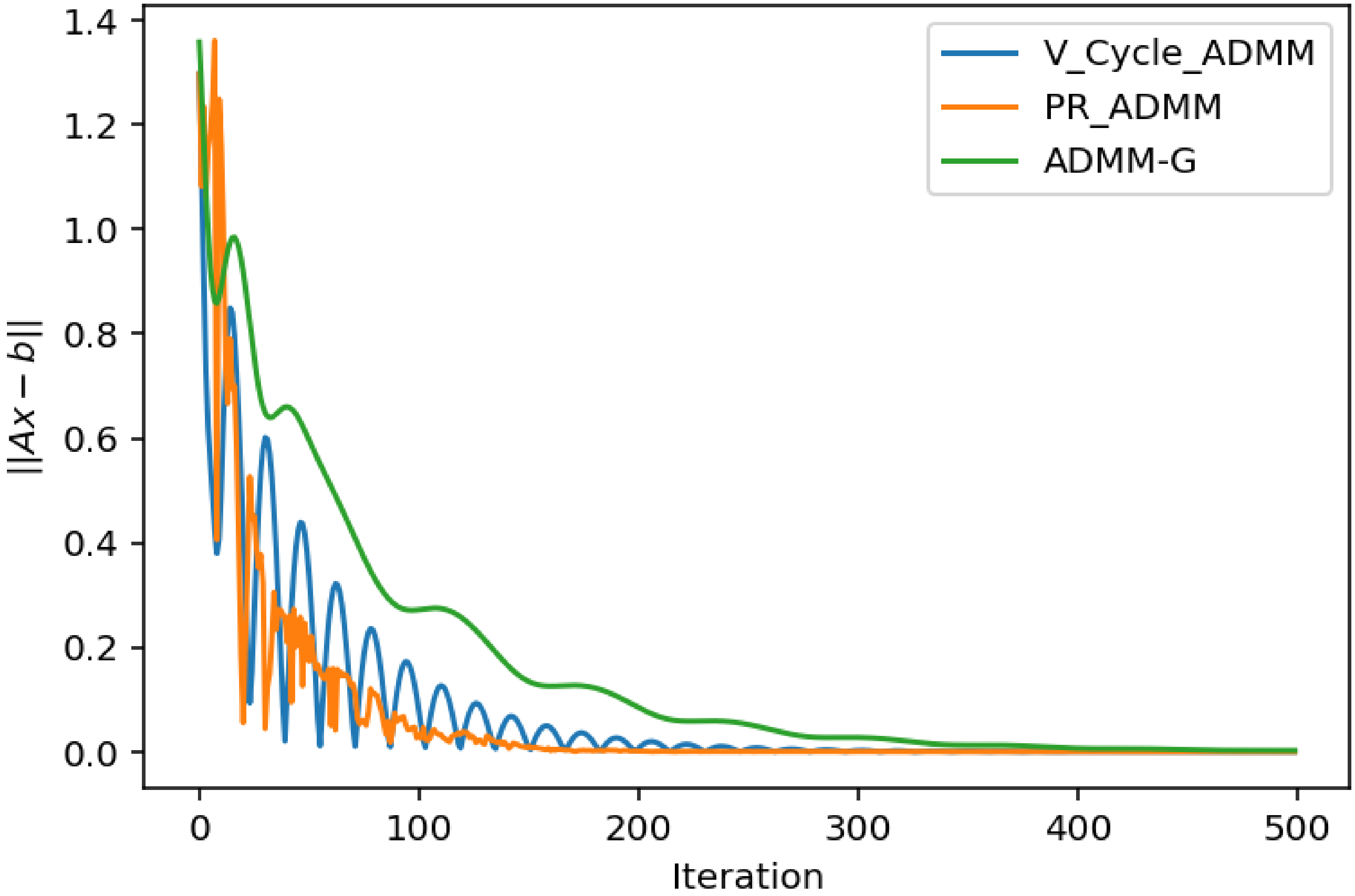
[Chen et al. 2018]

V-Cycle or Double Sweep ADMM

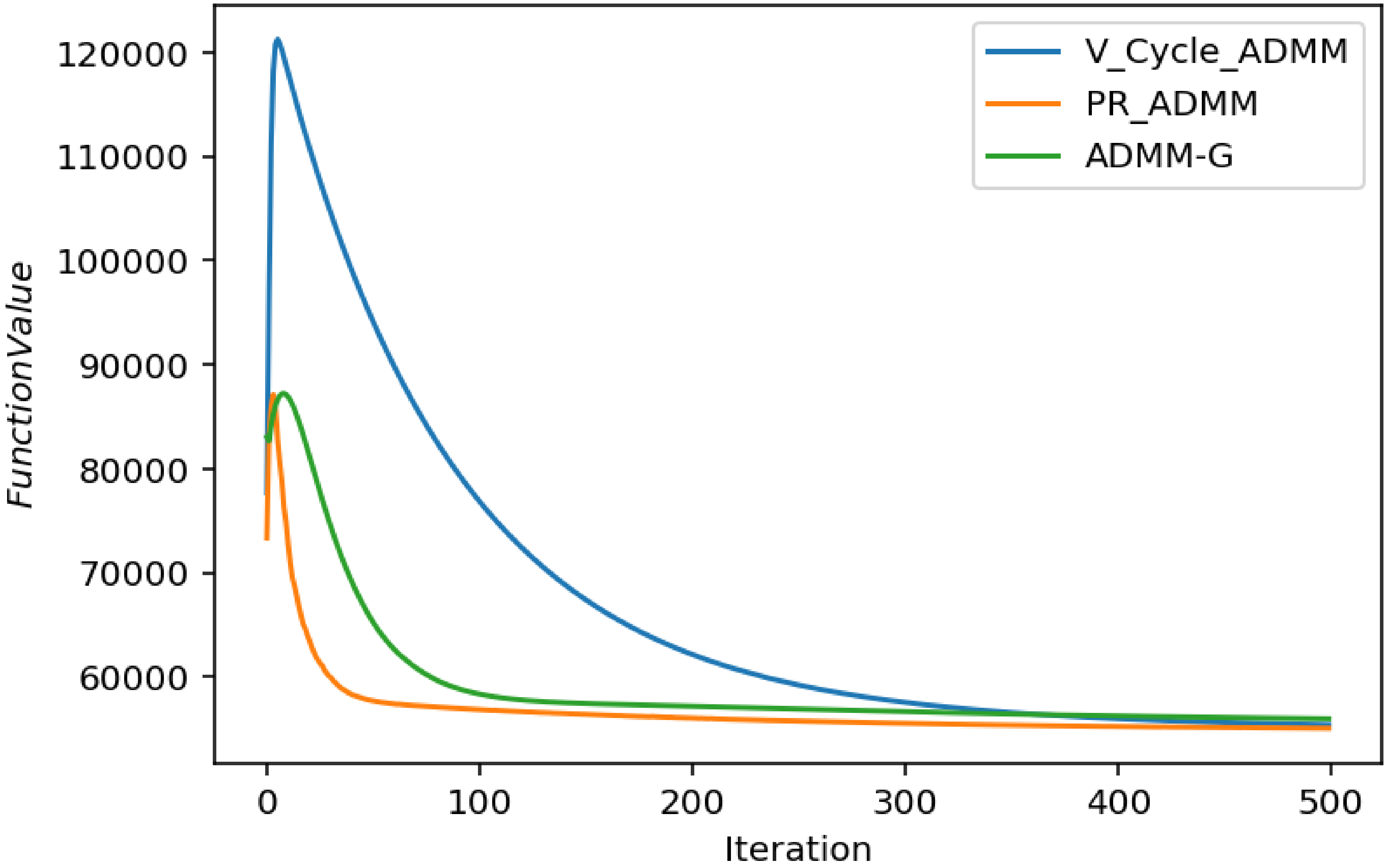
It was proved that it converges for solving system of linear equations:

$$
\begin{aligned}
\mathbf{x}_1 &\longleftarrow \arg\min_{\mathbf{x}_1 \in \mathcal{X}_1} L_{\mathcal{A}}(\mathbf{x}_1, \ldots, \mathbf{x}_n; \mathbf{y}), \\
&\vdots \\
\mathbf{x}_n &\longleftarrow \arg\min_{\mathbf{x}_n \in \mathcal{X}_n} L_{\mathcal{A}}(\mathbf{x}_1, \ldots, \mathbf{x}_n; \mathbf{y}), \\
\mathbf{x}_{n-1} &\longleftarrow \arg\min_{\mathbf{x}_1 \in \mathcal{X}_1} L_{\mathcal{A}}(\mathbf{x}_1, \ldots, \mathbf{x}_n; \mathbf{y}), \\
&\vdots \\
\mathbf{x}_1 &\longleftarrow \arg\min_{\mathbf{x}_n \in \mathcal{X}_n} L_{\mathcal{A}}(\mathbf{x}_1, \ldots, \mathbf{x}_n; \mathbf{y}), \\
\mathbf{y} &\longleftarrow \mathbf{y} - \beta(A\mathbf{x} - \mathbf{b}),
\end{aligned}
$$

Divergent Counter Example



Solve Some Optimization Problems



More Randomness: Randomly Assembled Cyclic ADMM (RAC-ADMM)

Add More Randomness: Randomly select variables in each block + Randomly permuting block order.

More precisely, in each ADMM step

- Randomly (without replacement) assemble primal variables into blocks $\mathbf{x}_i, \; i = 1, ..., n$.

- Then

$$\mathbf{x}_1 \longleftarrow \arg\min_{\mathbf{x}_1 \in \mathcal{X}_1} L_{\mathcal{A}}(\mathbf{x}_1, \ldots, \mathbf{x}_n; \mathbf{y}),$$
$$\ldots$$
$$\mathbf{x}_n \longleftarrow \arg\min_{\mathbf{x}_n \in \mathcal{X}_n} L_{\mathcal{A}}(\mathbf{x}_1, \ldots, \mathbf{x}_n; \mathbf{y}),$$
$$\mathbf{y} \longleftarrow \mathbf{y} - \beta(A\mathbf{x} - \mathbf{b}),$$

The idea originates from a randomized block coordinate descent (BCD) method from K. Mihic, K. Ryan, and A. Wood, “Randomized decomposition solver with the quadratic assignment problem as a case study,” INFORMS Journal on Computing, 30 (2018), pp. 295-308.

RAC-ADMM Interpretation: Double Randomness

- RAC-ADMM could be viewed as a double-randomness procedure based on RP-ADMM
- RAC-ADMM is equivalent as

Step 1 : Uniformly random choose a Block Composition Structure (which variables should be assembled into a block for all n blocks)

Step 2 : After selecting a block composition structure, do random permutation across n blocks for updating

- Consider the example: $N = 6$, $n = 3$, and each block has two variables. Then

$$\#\text{Block Composition Structures} = 15 \qquad \#\text{RP} = n! = 6 \qquad \#\text{RAC} = 90.$$

Equivalent as first uniformly random choose one among all 15 different block composition structure, then randomly permute across blocks for variable updates.

Theorem 5 *(Mihic, Zhu and Y [2018]) The expected output from RAS-ADMM converges to the unique solution of the linear system equations any number of variables $N \geq 1$ and any block number $N \geq n \geq 1$.*

But Is the More the Better?

Consider optimization problem $\mathbf{x} \in R^6$ (Mihic, Zhu and Y [2018]):

$$\min_{\mathbf{x}} \quad \mathbf{0}^T\mathbf{x} \quad \text{s.t.} \quad A\mathbf{x} = \mathbf{0}.$$

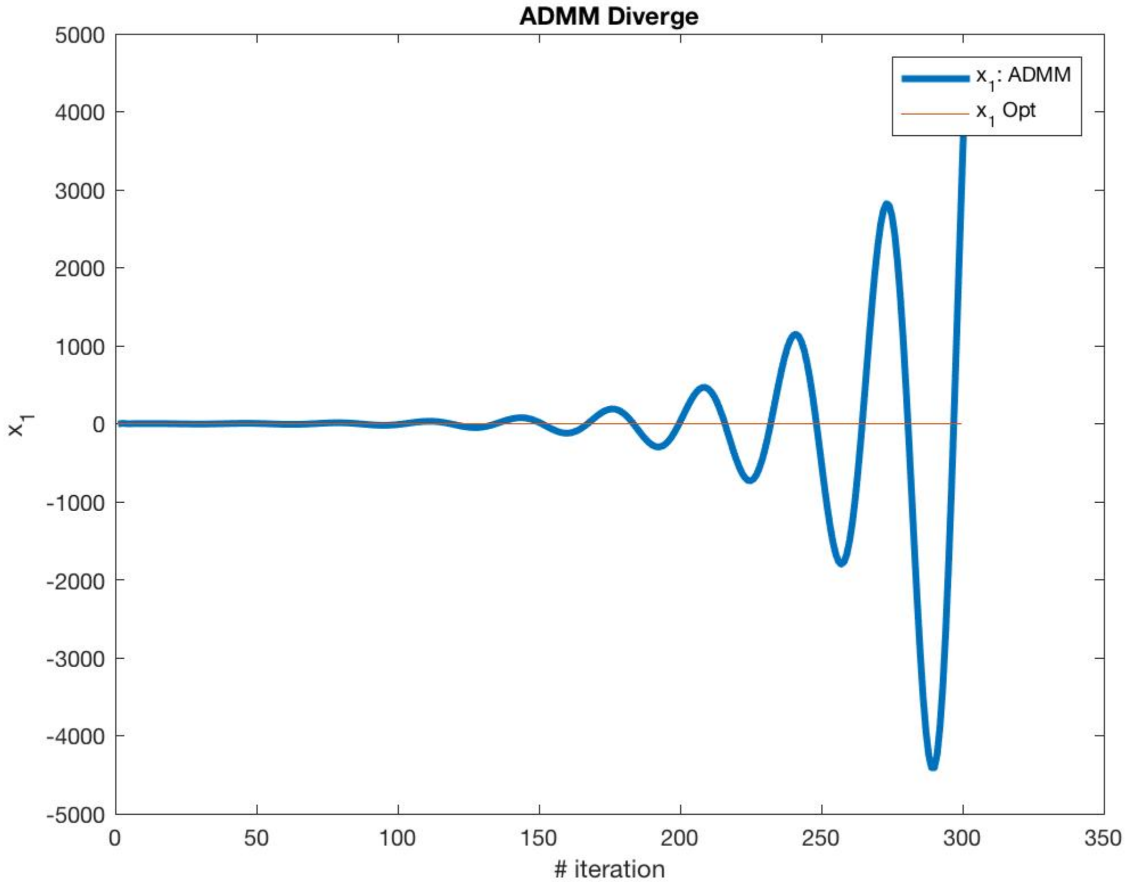
where

$$A = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 2 \\ 1 & 1 & 1 & 1 & 2 & 2 \\ 1 & 1 & 1 & 2 & 2 & 2 \\ 1 & 1 & 2 & 2 & 2 & 2 \\ 1 & 2 & 2 & 2 & 2 & 2 \end{pmatrix}$$

- Partition all variables equally into 3 blocks, compare ADMM, RSC ADMM and RP ADMM.
- Initial solutions and parameters of this specific model are $\mathbf{x}_0, \mathbf{y}_0 \sim N(0, 5I)$ and $\beta = 1$.

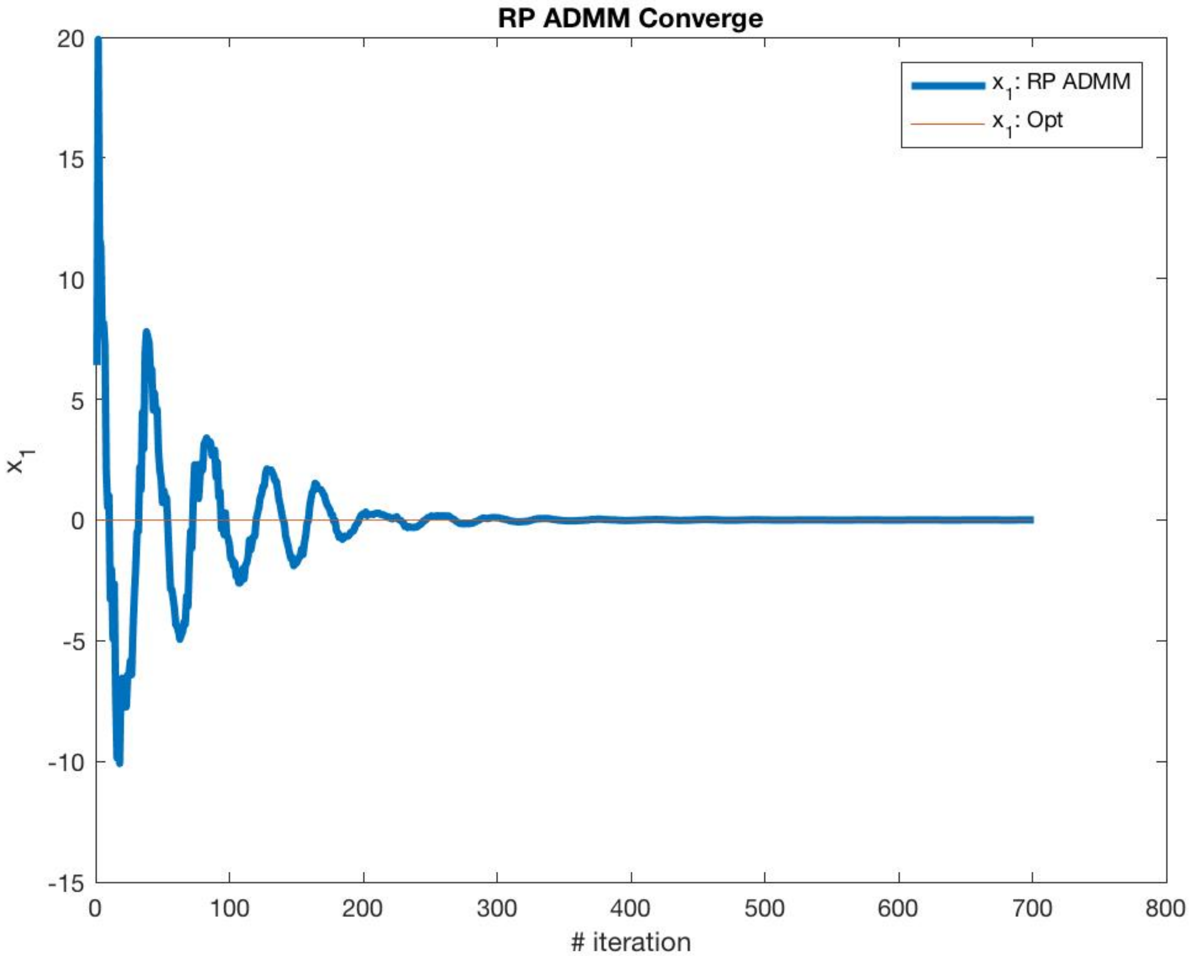
Regular ADMM diverges

Fix the Block Composition Structure $[x_1, x_2],\ [x_3, x_4],\ [x_5, x_6]$, and use the ppdate Order $[x_1, x_2] \rightarrow [x_3, x_4] \rightarrow [x_5, x_6]$

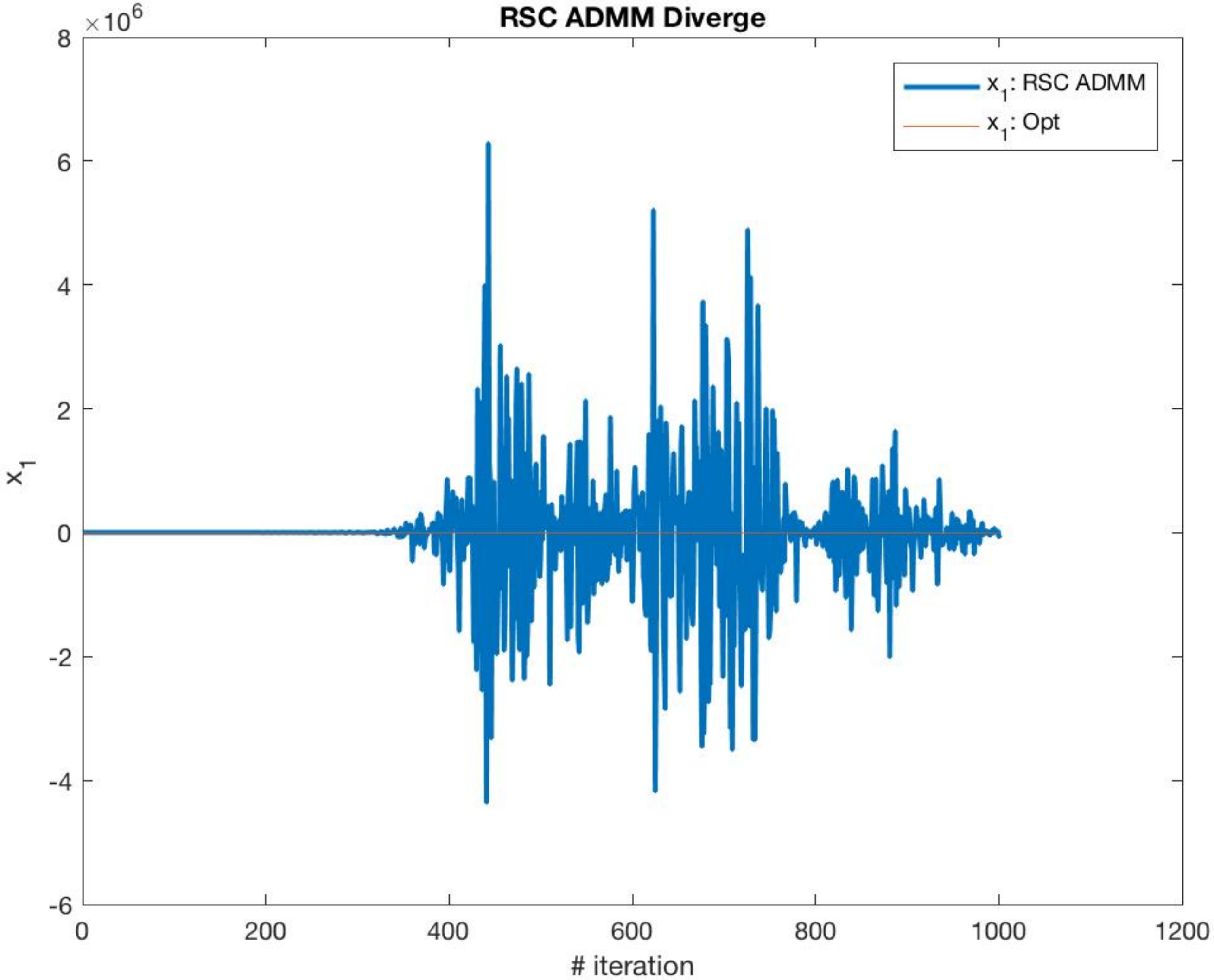


RP ADMM converges

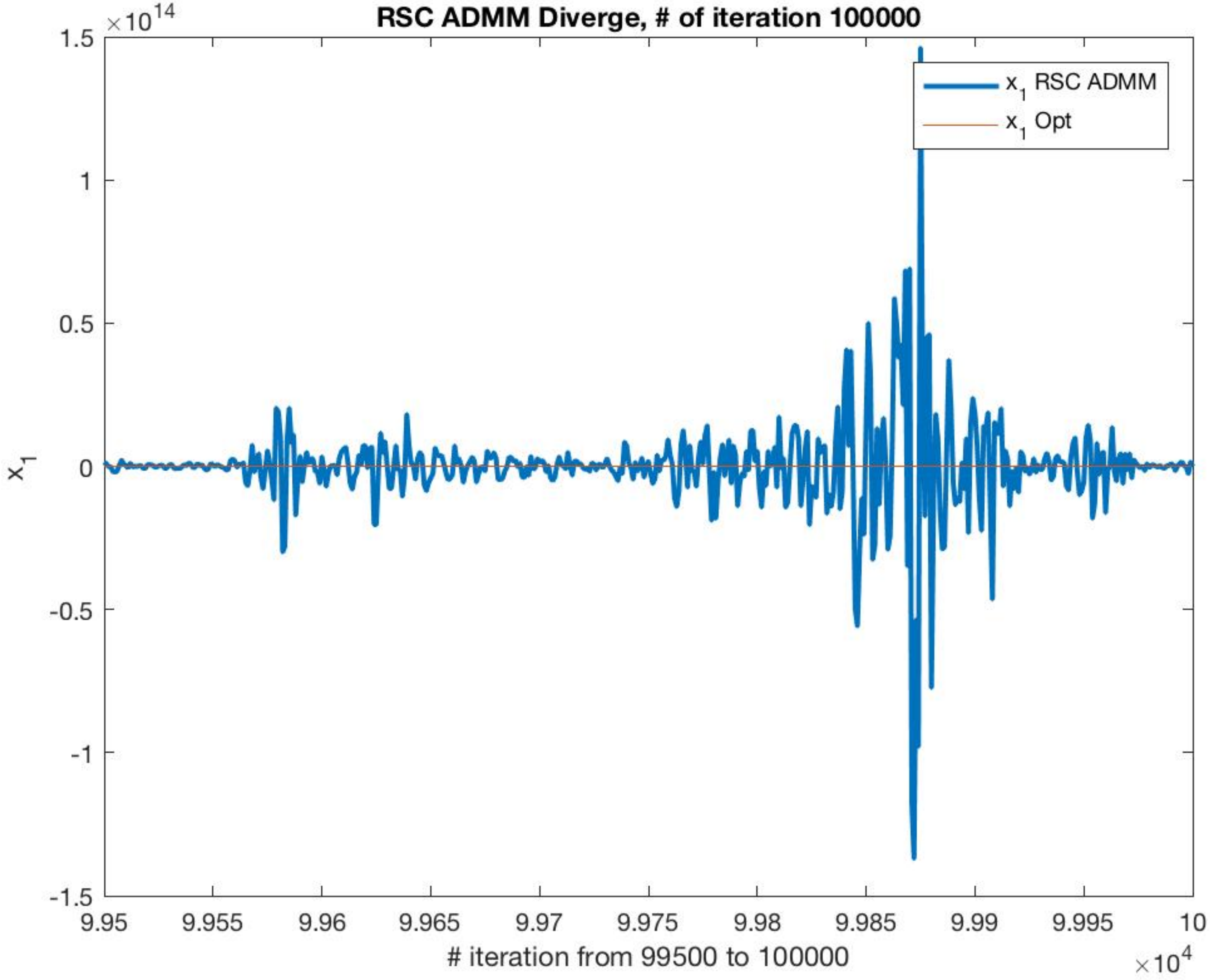
Fix the Block Composition Structure $[x_1, x_2],\ [x_3, x_4],\ [x_5, x_6]$, and use the RP-ADMM



RAC-ADMM Does Not Converge I



RAC-ADMM Does Not Converge II



Recall Convergence in Expectation

- The problem can be reformulated as a linear mapping

$$(\mathbf{x}^{k+1};\mathbf{y}^{k+1}) = M_\sigma(\mathbf{x}^k;\mathbf{y}^k).$$

- For all mapping matrices of RP-ADMM M_σ:

$$\rho(\mathbf{E}[M_\sigma]) = 0.9887 < 1.$$

For RAC-ADMM:

- For all block composition structures and permutation mapping matrices $M_{(RAC,\sigma)}$:

$$\rho(\mathbf{E}[M_{(RAC,\sigma)}]) = 0.8215 < 1.$$

Strong Notion for Convergence: Convergence Almost Surely

Let $\mathbf{z}^k = [\mathbf{x}^k; \mathbf{y}^k]$, for any mutli-block ADMM randomized algorithm, let $\{M_{\sigma_k}\}$ be the set of all possible updating matrices. At each iteration k, we randomly choose a M_{σ_k} from the set and update

$$\mathbf{z}^{k+1} = M_{\sigma_k}\mathbf{z}^k.$$

Now consider the Expected Kronecker Product of Mapping Matrices $M_{\sigma_k} \otimes M_{\sigma_k}$. If one can prove

$$\rho(\mathbf{E}[M_{\sigma_k} \otimes M_{\sigma_k}]) < 1.$$

Then from Borel-Cantelli's theorem, $\mathbf{x}^k$ converges to the solution almost surely, or a.s. in short.

On this Example:

- $$\rho(\mathbf{E}[M_{\sigma_k}^{RP} \otimes M_{\sigma_k}^{RP}]) = 0.9852, \quad (\rho(\mathbf{E}[M_{\sigma_k}^{RP}]) = 0.9887)$$

  that is, RP-ADMM does converge almost surely for this specific linear system

- In fact, RP-ADMM with any fixed one of the all possible 15 block composition structures converges almost surely for this specific linear system

- Unfortunately,

  $$\rho(\mathbf{E}[M_{\sigma_k}^{RAC} \otimes M_{\sigma_k}^{RAC}]) = 1.0948, \quad (\rho(\mathbf{E}[M_{\sigma_k}^{RAC}]) = 0.8215)$$

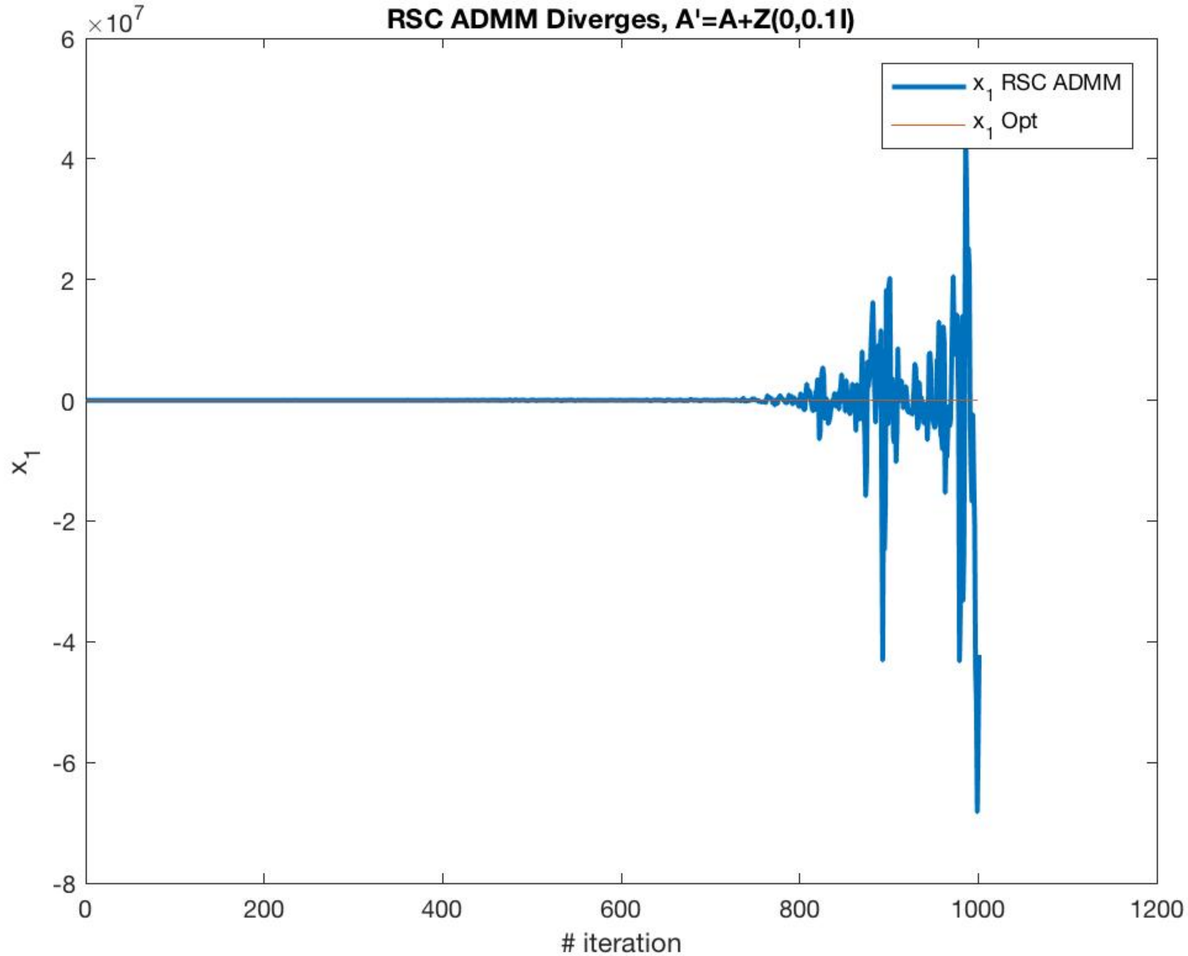
  that is, RAC-ADMM does not converge almost surely for this specific linear system.

- (Random) initial solutions do not change the convergence pattern.

Mihic, Zhu and Y [2018]

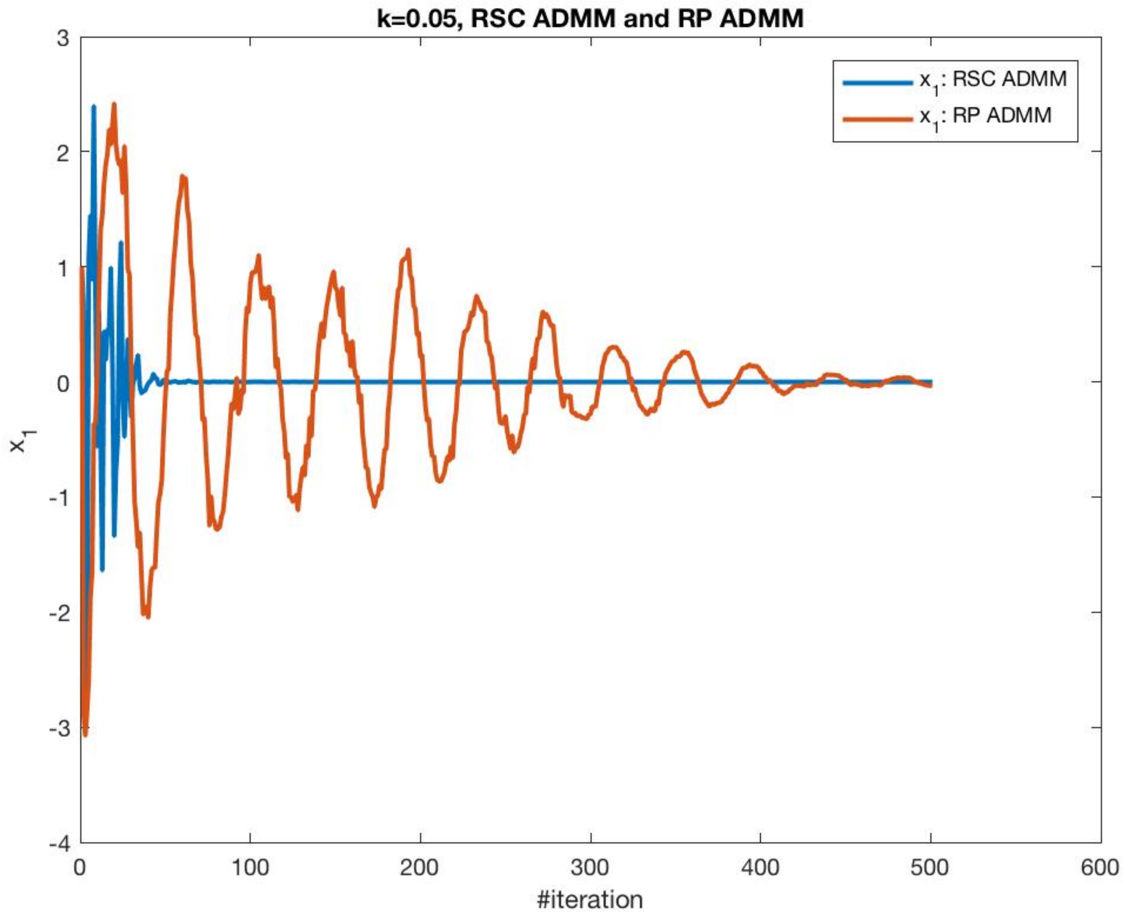
Moderate Noise does Not Change the RAC-ADMM Outcome

Set $A' = A + N(0, 0.1I)$



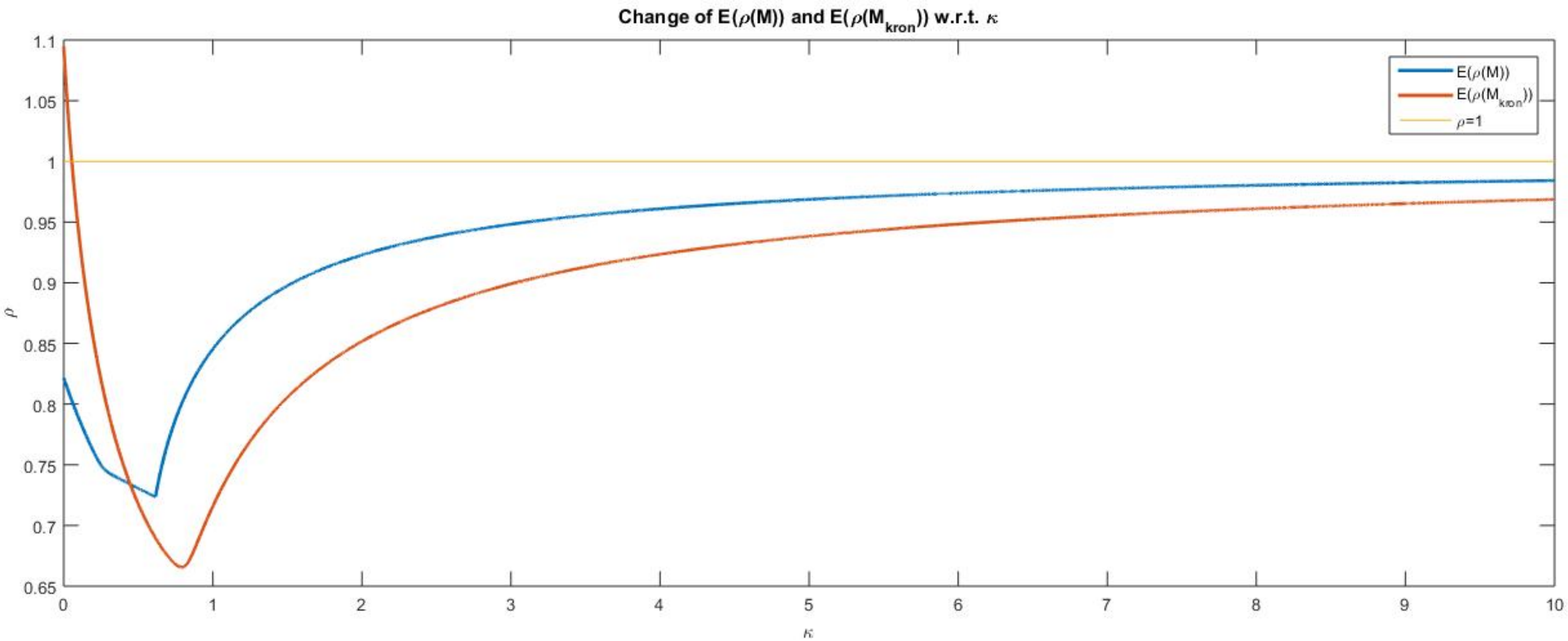
When is Safer for RAC-ADMM?

For this example, instead of setting objective function equals to null, set the objective function $\frac{\kappa}{2}\mathbf{x}^T\mathbf{x} = \frac{\kappa}{2}\|\mathbf{x}\|^2$. Then, with small $\kappa = 0.05$, RAC-ADMM now converges faster than RP-ADMM!



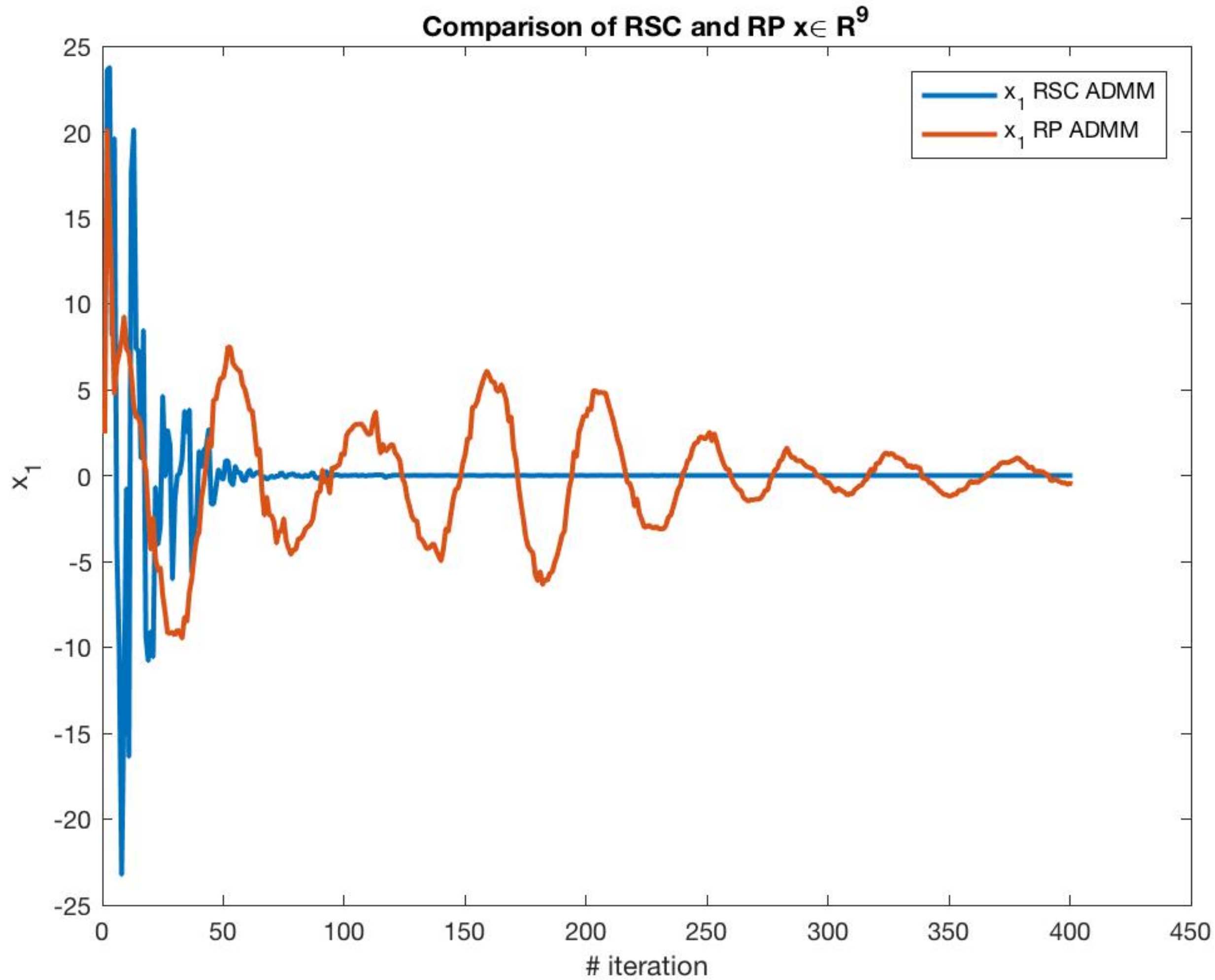
Convex Regularization Kelps in General?

We conjecture that there exists $\overline{\kappa}$ such that for all $\kappa \geq \overline{\kappa}$, the spectral radius of expected Kronecker Product Mapping Matrix is strictly less than one.



Increase of Block Size Kelps

Increase matrix dimension of A to 9 where each block consists of three variables.



Increase of Block Size Kelps (continued)

Here RP refer to RP-ADMM with block structure $[x_1, x_2, x_3],\ [x_4, x_5, x_6],\ [x_7, x_8, x_9]$

$$\rho^{RP} = 0.9926 \qquad \rho^{RP}_{Kron.} = 0.9903$$

$$\rho^{RSC} = 0.8006 \qquad \rho^{RSC}_{Kron.} = 0.9836$$

Mihic, Zhu and Y [2018]

Experiments on the Markowitz Mean-Variance Model

Consider the regularized (2-norm) Markowitz Mean-Variance Model

$$\begin{array}{ll} \min_{\mathbf{x}} & f(\mathbf{x}) = \mathbf{x}^T V \mathbf{x} + \tau \mathbf{c}^T \mathbf{x} + \frac{\kappa}{2} ||\mathbf{x}||_2^2 \\ \text{s.t} & \mathbf{x} \in X \end{array}$$

where typically $X = \{\mathbf{x} : \mathbf{e}^T \mathbf{x} = 1, \mathbf{x} \in \mathbf{R}^n_+\}$.

In the following numerical experiments, we generate positive definite covariance matrix V and return vector $\mathbf{c}$ randomly. For a 6 variable instance with $\kappa = 1.e - 5$:

$$\rho^{RP} = 0.7539 \qquad \rho^{RP}_{Kron.} = 0.5787$$

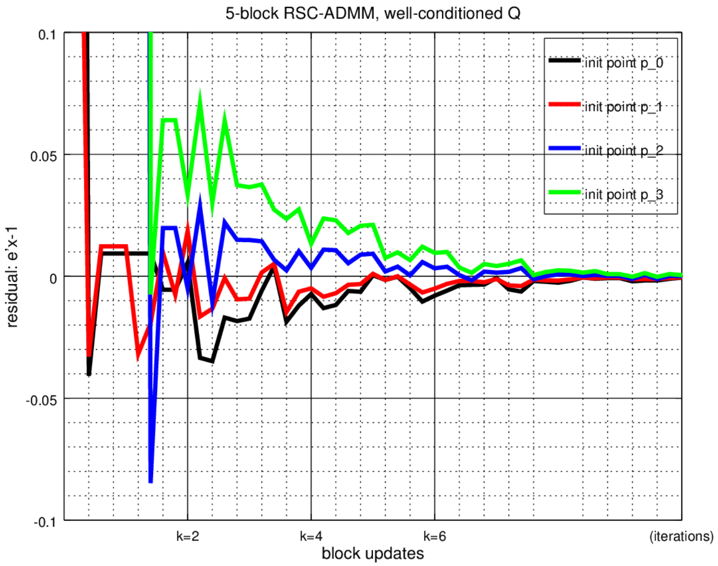
$$\rho^{RSC} = 0.4815 \qquad \rho^{RSC}_{Kron.} = 0.2947$$

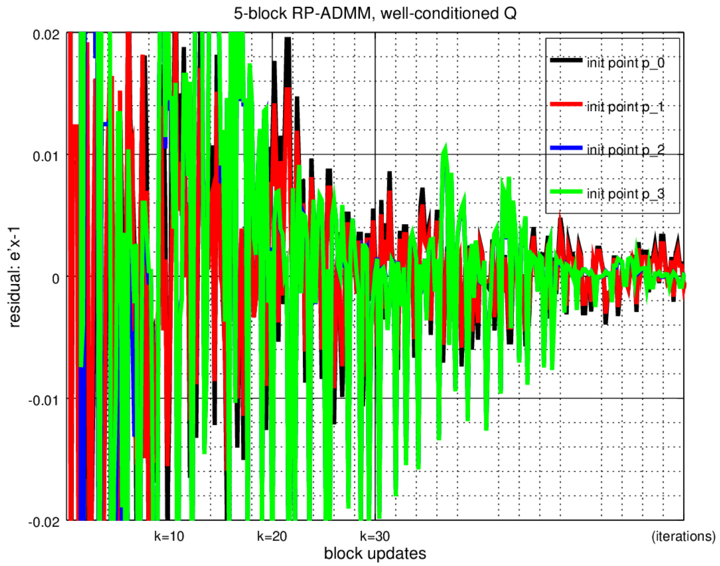
We even consider $X = \{\mathbf{x} : \mathbf{e}^T \mathbf{x} = m, \mathbf{x} \in \{0,1\}^n, m < n\}$ a Binary-Variant of Markowitz-based portfolio selection with the cardinality constraint.

Numerical Results: Markowitz Mean-Variance Model

	μ	σ^2	min	max
RSC-ADMM	18.7	9.8	2	72
ADMM	197.1	175.8	2	$>$1000
RP-ADMM	241.1	231.9	2	$>$1000

Table 2: 4800 variables, 5 Blocks, $\beta = 1$, Number of iterations till convergence





Solution Times: Markowitz Mean-Variance Model

RAC-ADMM

n. blocks	objVal	solver time [s]	model time [s]
2	0.2996511	3254	38
3	0.2996513	973	51
4	0.2996510	365	69
5	0.2996513	166	83
6	0.2996512	97	99

Table 3: 4800 variables, $\beta = 1$, $\|A\mathbf{x} - \mathbf{b}\| \leq 1.e - 6$

model time: total time spent preparing sub problems.

Gurobi Direct Convex QP Run: obj val: 0.299650947, and time[s]: 588.95.

Numerical results: Binary Markowitz Mean-Variance Model I

	Number of blocks							
	k=2		k=3		k=4		k=5	
	μ	σ^2	μ	σ^2	μ	σ^2	μ	σ^2
RSC-ADMM	0.28	0.14	0.12	0.09	0.15	0.08	0.05	0.04
ADMM	0.85	0.48	1.20	0.50	1.06	0.57	1.09	0.59
RP-ADMM	0.66	0.29	1.26	0.68	0.72	0.34	1.24	0.51

Table 4: Gap of the best local solution for different number of blocks (4800 variables, 5 Blocks, $\beta = 1$)

Gap (*gap*$_{GA}$) between the best solution found by the algorithms (*objVal*$_A$)and the solution found by Gurobi (*objVal*$_G$) by solving a problem as whole is defined by:

$$gap_{GA} = \frac{objVal_G - objVal_A}{objVal_G} \times 100$$

Numerical results: Binary Markowitz Mean-Variance Model II

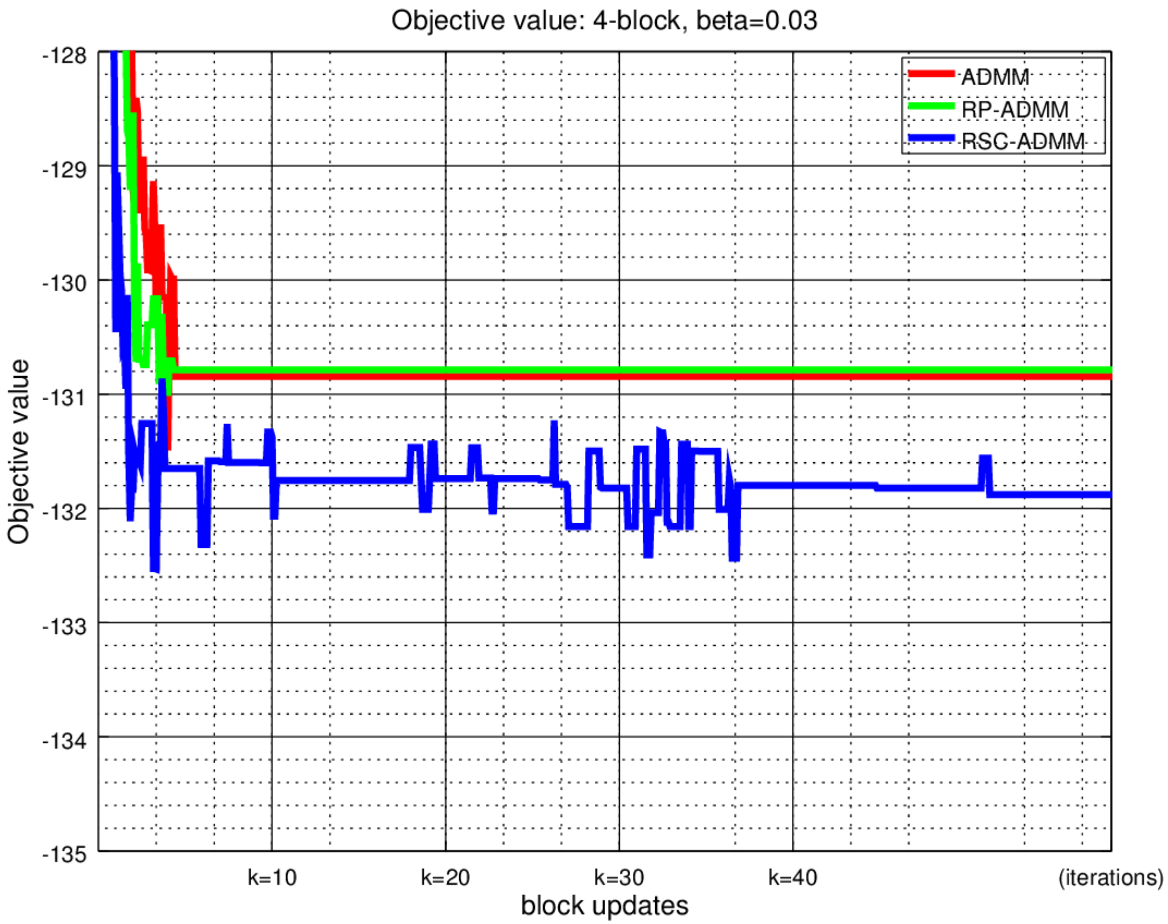


Figure 1: Evaluation of the objective function value for binary Markowitz model

Extensions and Research Directions (Suggested Project #4)

- Convergence almost surely for RP-ADMM on solving the example with general N and n??
- Convergence almost surely for RP-ADMM on solving general linear system of equations??
- Convergence almost surely for RAC-ADMM on solving convex QP programs??
- Generalize to solving linear programquitming problems??
- Generalize to solving general convex optimization at large??
- Generalize to solving non-convex or binary optimization??