

On the Efficiency of Random Permutation for ADMM and Coordinate Descent

Ruoyu Sun

Zhi-Quan Luo

Yinyu Ye ^{*†‡§}

Dec 31, 2018

Abstract

Random permutation is observed to be powerful for optimization algorithms: for multi-block ADMM (alternating direction method of multipliers), while the classical cyclic version diverges, the randomly permuted version converges in practice; for BCD (block coordinate descent), the randomly permuted version is typically faster than other versions. In this paper, we provide strong theoretical evidence that random permutation has positive effects on ADMM and BCD, by analyzing randomly permuted ADMM (RP-ADMM) for solving linear systems of equations, and randomly permuted BCD (RP-BCD) for solving unconstrained quadratic problems. First, we prove that RP-ADMM converges in expectation for solving systems of linear equations. The key technical result is that the spectrum of the expected update matrix of RP-BCD lies in $(-1/3, 1)$, instead of the typical range $(-1, 1)$. Second, we establish expected convergence rates of RP-ADMM for solving linear systems and RP-BCD for solving unconstrained quadratic problems. This expected rate of RP-BCD is $O(n)$ times better than the worst-case rate of cyclic BCD, thus establishing a gap of at least $O(n)$ between RP-BCD and cyclic BCD. To analyze RP-BCD, we propose a conjecture of a new matrix AM-GM (algebraic mean-geometric mean) inequality, and prove a weaker version of it.

^{*}This paper is a strengthened version of a previous technical report “On the expected convergence of randomly permuted ADMM” appeared on arxiv on April 2015, with several new results, mainly the ones on the expected convergence rate of RP-CD and RP-ADMM.

[†]Ruoyu Sun is with University of Illinois at Urbana-Champaign, USA. Part of the work was done when this author was a student at University of Minnesota and a postdoc at Stanford University. Email: ruoyus@illinois.edu.

[‡]Zhi-Quan Luo is with the Chinese University of Hong Kong, Shenzhen, China. He is also affiliated with University of Minnesota, Minneapolis, MN 55455, USA. Email: luoqz@cuhk.edu.cn.

[§]Yinyu Ye is with the Department of Management Science and Engineering, School of Engineering, Stanford University, USA; and International Center of Management Science and Engineering, School of Management and Engineering, Nanjing University, China. Email: yyye@stanford.edu.

Contents

1	Introduction	4
1.1	Summary of Contributions	6
1.2	Related Works	6
1.3	Notation and Organization	7
2	Algorithms	8
2.1	Randomly Permuted ADMM	8
2.1.1	Optimization Formulation of Solving a Linear System of Equations	9
2.1.2	Example of 3-block ADMM	9
2.1.3	General Update Equation of RP-ADMM	10
2.2	Randomly Permuted BCD	11
2.3	Residual Trick for Efficient Implementation of ADMM and BCD	12
2.4	Two Versions of Independently Randomized ADMM	13
2.5	Bernoulli-Randomized ADMM	14
3	Main Results	16
3.1	Expected Convergence of RP-ADMM	16
3.2	Expected Convergence Rate of RP-ADMM and RP-BCD	17
3.3	Matrix AM-GM Inequality	19
4	Proof of Main Results	20
4.1	Proof of Theorem 1	20
4.2	Proof of Theorem 2	20
4.3	Proof of Theorem 3	22
4.3.1	Proof of Claim 4.1	24
4.4	Proof of Proposition 2	25
4.5	Proof of Theorem 4	26
5	Proof of Lemma 1	27
6	Proof of Lemma 2	29

6.1	Proof Overview	29
6.2	Proof of Lemma 2	30
6.3	Proof of Proposition 3 (the induction formula)	32
6.4	Proof of Proposition 1	33
7	Proof of Technical Results for Expected Convergence Rates	33
7.1	Proof of Claim 4.2	33
7.2	Proof of Lemma 4	34
7.2.1	Step 1: Mathematical Induction and Induction Formula	35
7.2.2	Step 2: Relation Between $\lambda_{\max}(A_n Q A_n^T)$ and its analog	37
7.2.3	Step 3: More Precise Bound of λ	40
7.2.4	Proof of Claim 7.1	41
7.3	Proof of Lemma 3	42
8	Numerical Experiments	47
9	Concluding Remarks	50
10	Acknowledgment	50

1 Introduction

A simple yet powerful idea for solving large-scale computational problems is to iteratively solve smaller subproblems. The applications of this idea include coordinate descent (CD), POCS (Projection onto Convex Sets), SGD (Stochastic Gradient Descent). They are well suited for large-scale *unconstrained* optimization problem (see, e.g. Wright [1], for a recent survey of CD) since it decomposes a large problem into small subproblems. The decomposition idea is crucial for huge problems due to both the cheap per-iteration cost and small memory requirement. Moreover, this idea is “orthogonal” to other large-scale optimization ideas such as first-order methods (using only gradient information) and random projection, and thus can be easily combined with other ideas.

This paper is motivated by a natural question: how should we extend the decomposition idea to solve problems *with constraints*? We consider a constrained minimization problem with a convex objective function and linear constraints (this is for motivation; our analysis is for a much simpler version):

$$\begin{aligned} \min_{x_1, \dots, x_n} \quad & f(x_1, x_2, \dots, x_n), \\ \text{s.t.} \quad & A_1 x_1 + \dots + A_n x_n = b, \\ & x_i \in \mathcal{X}_i, \quad i = 1, \dots, n, \end{aligned} \tag{1}$$

where $A_i \in \mathbb{R}^{N \times d_i}$, $b \in \mathbb{R}^{N \times 1}$, $\mathcal{X}_i \subseteq \mathbb{R}^{d_i}$ is a closed convex set, $i = 1, \dots, n$, and $f : \mathbb{R}^{d_1 + d_2 + \dots + d_n} \rightarrow \mathbb{R}$ is a closed convex function. Many machine learning and engineering problems can be cast into linearly-constrained optimization problems with two blocks (see Boyd et al. [2] for many examples) or more than two blocks (e.g. linear programming, robust principal component analysis, composite regularizers for structured sparsity; see Chen et al. [3] and Wang et al. [4] for more examples).

To apply the decomposition idea to a constrained problem, one possible way is to form the augmented Lagrangian function and perform coordinate descent for the primal problem and a gradient step for the dual problem, i.e. combining BCD with augmented Lagrangian method, to obtain the so-called alternating direction method of multipliers (ADMM). ADMM was originally proposed in Glowinski and Marroco [5] (see also Chan and Glowinski [6], Gabay and Mercier [7]) to solve problem (1) when there are only two blocks (i.e. $n = 2$) and the objective function is separable. It is natural and computationally beneficial to extend the original ADMM directly to solve the general n -block problem (1) via the following procedure:

$$\begin{cases} x_1^{k+1} = \arg \min_{x_1 \in \mathcal{X}_1} \mathcal{L}(x_1, x_2^k, \dots, x_n^k; \mu^k), \\ \vdots \\ x_n^{k+1} = \arg \min_{x_n \in \mathcal{X}_n} \mathcal{L}(x_1^{k+1}, \dots, x_{n-1}^{k+1}, x_n; \mu^k), \\ \mu^{k+1} = \mu^k - \beta(A_1 x_1^{k+1} + \dots + A_n x_n^{k+1} - b), \end{cases} \tag{2}$$

where the augmented Lagrangian function

$$\mathcal{L}(x_1, \dots, x_n; \mu) = f(x_1, \dots, x_n) - \mu^T \left(\sum_i A_i x_i - b \right) + \frac{\beta}{2} \left\| \sum_i A_i x_i - b \right\|^2. \tag{3}$$

The convergence of the direct extension of ADMM to multi-block case had been an open question, until a counter-example was recently given in Chen et al. [3]. More specifically, Chen et al. [3] showed

that even for the simplest scenario where the objective function is 0 and the number of blocks is 3, ADMM can be divergent for a certain choice of $A = [A_1, A_2, A_3]$. There are several proposals to overcome the drawback (see, e.g., [8–25]), but they either need to restrict the range of original problems being solved, add additional cost in each step of computation, or limit the stepsize in updating the Lagrange multipliers. These solutions typically slow down the performance of ADMM for solving most practical problems. Moreover, it is not clear how to compare the convergence speed of these algorithms as they typically contain different parameters. One may ask whether a “minimal” modification of cyclic multi-block ADMM (2) can lead to convergence, and whether we can provide some convergence speed analysis that is easy to interpret.

One of the simplest modifications of (2) is to add randomness to the update order. Randomness has been very useful in the analysis of block coordinate descent (BCD) methods and stochastic gradient descent (SGD) methods. In particular, a recent work Sun and Ye [26] showed that randomized CD (R-CD) can be up to $O(n^2)$ times faster than cyclic CD (C-CD) for quadratic minimization in the worst case, where n is the number of variables¹. Another example is the comparison of IAG (Incremental Aggregated Gradient) in Blatt et al. [27] and its randomized version SAG (Stochastic Average Gradient) [28]: it turns out that the introduction of randomness leads to better iteration complexity bounds. There is also some study on randomly permuted version of pure SGD [29]. These examples show that randomization may improve the algorithm in theory and in practice.

It is important to note that the iteration complexity bounds for randomized algorithms are usually established for independent randomization (sampling with replacement), while in practice, random permutation (sampling without replacement) has been reported to exhibit faster convergence (e.g. Shalev et al. [30], Recht and Re [31], Sun [32]). Interestingly, our simulation shows that for solving linear system of equations, randomly permuted ADMM (RP-ADMM) always converges, but independently randomized versions of ADMM can be divergent even for Gaussian data. Therefore, we focus on the analysis of RP-ADMM in this paper.

Random permutation is known to be notoriously difficult to analyze. Even for unconstrained quadratic minimization, the convergence rate of RP-BCD is poorly understood. Many existing works treated cyclic BCD and RP-BCD together [33–35], and thus the best known convergence rate of RP-BCD for general convex problems are in fact the same as that of C-BCD [35]. However, in light of a recent study which established an up to $O(n^2)$ gap between cyclic CD and R-CD [26], it is unlikely that RP-CD has the same convergence rate as C-CD since that would imply RP-CD could be $O(n^2)$ -times slower than R-CD. For the special example that demonstrates the gap between C-CD and R-CD, it was shown recently that RP-CD is faster than R-CD² [36]. However, the general quadratic case seems to be quite difficult, probably due to its close connection to a matrix AM-GM (algebraic mean-geometric mean) inequality [37], the difficulty of which is essentially to prove an inequality in non-commutative algebra.

¹Rigorously speaking, these two bounds are not directly comparable since the result for the randomized version only holds with high probability, while the result for the cyclic version always holds; anyhow, this $O(n^2)$ gap is still meaningful if ignoring this difference between deterministic and randomized algorithm.

²This paper appeared after the first version of the current paper.

1.1 Summary of Contributions

We consider two extremes of a general RP-ADMM: i) the objective is zero, i.e., RP-ADMM for solving a linear system; ii) the constraint is zero and the objective is a quadratic function, i.e., RP-BCD for solving quadratic minimization. Due to the lack of understanding of random permutation for quadratic minimization as discussed previously, we restrict to the two cases in this paper.

The first result of this paper is the expected convergence of RP-ADMM for solving linear systems. More specifically, when the objective function is zero and the constraint is a non-singular square linear system of equations, the expected output of randomly permuted ADMM converges to the unique primal-dual optimal solution. A major technical result in this proof is that the eigenvalues of the expected iteration matrix of RP-BCD for quadratic problems lie in $(-1/3, 1)$, instead of the typical range $(-1, 1)$.

The second result is about the expected convergence rate of RP-ADMM for solving linear systems and RP-BCD for solving quadratic problems. We show that RP-BCD for a convex quadratic minimization problem with equal diagonal entries has expected iteration complexity $O(n \frac{\lambda_{\text{avg}}}{\lambda_{\text{min}}} \log(1/\epsilon))$, where λ_{avg} and λ_{min} are the average eigenvalue and the minimum eigenvalue of the coefficient matrix, and one “iteration” here means a cycle of updating all blocks. This improves an existing bound of $O(n^2 \frac{\lambda_{\text{avg}}}{\lambda_{\text{min}}} \log(1/\epsilon))$ for RP-BCD by a factor of n . Built on this result, we further show that RP-ADMM for solving linear systems achieves the same expected iteration complexity bound $O(n \frac{\lambda_{\text{avg}}}{\lambda_{\text{min}}} \log(1/\epsilon))$.

Technically, we provide a simple and clean proof of the expected convergence, by applying a classical result on the eigenvalues of Jordan product. For proving the expected convergence rate, we propose a new variant of the matrix AM-GM inequality conjecture, and prove a weaker version of this conjecture.

Our result shows that random permutation may be a good answer to the question “how to apply the decomposition idea to solve constrained problems”. As multi-block BCD is widely used for large-scale unconstrained problems, we expect multi-block RP-ADMM to be a good candidate for large-scale linearly constrained problems. Our result provides one of the few direct analyzes of random permutation in optimization algorithms, and offers an explanation of the mysterious gap between RP-ADMM and cyclic ADMM. As reflected by the proof, the intuition is that random permutation provides “3-level symmetrization” that adjusts the spectrum of the update matrix. Based on the analysis for RP-ADMM, we are able to improve the best known complexity of RP-BCD for equally-diagonal quadratic problems by a factor of n , when expressing the complexity only in terms of the quantity $\frac{\lambda_{\text{avg}}}{\lambda_{\text{min}}}$.

1.2 Related Works

This paper is a stronger version of a previous technical report Sun et al. [38] which was not published. Another related work is the paper Chen et al. [39], which modifies the proof of [38] to make it work with a quadratic objective function.

We highlight a few novel contributions of the current paper (neither in the original technical report [38] nor in the paper [39]).

- (i) The current paper provides a much simpler proof for the result of expected convergence.
- (ii) The current paper provides the first convergence rate analysis of RP-ADMM. See Theorem 4

and the proof in Section 4.5, Section 7.2 and Section 7.1.

(iii) The current paper provides an improved convergence rate analysis of RP-BCD, See Theorem 3 and the proof in Section 4.3 and Section 7.3.

(iv) The current paper introduces a theory-motivated algorithm Bernoulli-ADMM, which reduces the sampling time yet still achieves the expected convergence. This update order has not appeared before even in other algorithm setups to our knowledge. See Section 2.5 and Proposition 1.

Besides the technical contributions, we want to emphasize that the current paper is not just adding new result to our previous technical report [38], but actually completes a missing step of the story. From a mathematical point of view, the most striking consequence of our original proof is that the spectral radius of RP-BCD lies in a smaller region $(-1/3, 1)$. It is natural to think that this fundamental fact should have an impact on the analysis of original RP-BCD. Our current paper fills this gap by showing that this result can help build an $O(n)$ gap between the (expected) onvergence rate of RP-BCD and cyclic BCD. A general message is that on one hand, to understand constrained optimization we have to understand unconstrained optimization (analyzing ADMM reduces to analyzing BCD); on the other hand, analyzing constrained optimization helps improve the understanding of unconstrained optimization (the analysis of ADMM leads to progress in BCD). We find this interaction between unconstrained optimization (BCD) and constrained optimization (ADMM) fascinating. The whole story is only revealed in the current paper, but not in the previous technical report [38] or Chen et al. [39].

Besides the above unique aspects, the current paper inherits some interesting numerical findings from the technical report Sun et al. [38] which do not appear in Chen et al. [39]. We find that cyclic ADMM diverges with probability 1 for many random distributions of data, thus showing that the seemingly surprising divergence behavior reported in [3] is quite common. However, it is easy to miss this finding if one uses the Gaussian distribution to generate data. Another interesting finding is that the independently randomized version of ADMM diverges with probability 1 for Gaussian data but not for the counter-example in [3], preventing us from analyzing the independently randomized version. Without these findings, the motivation of studying RP-ADMM would be less clear. See Section 2.4 and Section 8.

1.3 Notation and Organization

Notation. For a matrix X , we denote $X(i, j)$ as the (i, j) -th entry of X , $\text{eig}(X)$ as the set of eigenvalues of X , $\rho(X)$ as the spectral radius of X (i.e. the maximum modulus of the eigenvalues of X), $\|X\|$ as the spectral norm of X , and X^T as the transpose of X . When X is block partitioned, we use $X[i, j]$ to denote the (i, j) -th block of X . When X is a real symmetric matrix, let $\lambda_{\max}(X)$ and $\lambda_{\min}(X)$ denote the maximum and minimum eigenvalue of X respectively. For two real symmetric matrices X_1 and X_2 , $X_1 \succ X_2$ (resp. $X_1 \succeq X_2$) means $X_1 - X_2$ is positive definite (resp. positive semi-definite). We use I_m to denote the identity matrix with dimension m , and we will simply use I when it is clear from the context what the dimension is. For square matrices $U_i \in \mathbb{R}^{u_i \times u_i}$, $i = 1, \dots, k$, we denote $\text{Diag}(U_1, U_2, \dots, U_k)$ as the block-diagonal matrix with U_i being the i -th diagonal block.

Organization. In Section 2, we present three versions of randomized ADMM, with an emphasis on

RP-ADMM. In Section 3, we present our main results Theorem 1, Theorem 2 and their proofs. The subsequent sections are devoted to the proofs of the two technical results Lemma 1 and Lemma 2, which are used in the proof of Theorem 2. In particular, the proof of Lemma 1 is given in Section 5, and the proof of Lemma 2 is given in Section 6.

2 Algorithms

In this section, we will present both randomly permuted and independently randomized versions of ADMM for solving (1), and specialize RP-ADMM for solving a square system of equations. We also present a rather novel algorithm Bernoulli-randomized ADMM (motivated by our proof).

2.1 Randomly Permuted ADMM

In this subsection, we first propose RP-ADMM for solving the general optimization problem (1), then we present the update equation of RP-ADMM for solving a linear system of equations.

Define Γ as

$$\Gamma \triangleq \{\sigma \mid \sigma \text{ is a permutation of } \{1, \dots, n\}\}. \quad (4)$$

At each round, we draw a permutation σ of $\{1, \dots, n\}$ uniformly at random from Γ , and update the primal variables in the order of the permutation, followed by updating the dual variables in a usual way. Obviously, all primal and dual variables are updated exactly once at each round. See Algorithm 1 for the details of RP-ADMM. Note that with a little abuse of notation, the function $\mathcal{L}(x_{\sigma(1)}, x_{\sigma(2)}, \dots, x_{\sigma(n)}; \mu)$ in this algorithm should be understood as $\mathcal{L}(x_1, x_2, \dots, x_n; \mu)$. For example, when $n = 3$ and $\sigma = (231)$, $\mathcal{L}(x_{\sigma(1)}, x_{\sigma(2)}, x_{\sigma(3)}; \mu) = \mathcal{L}(x_2, x_3, x_1; \mu)$ should be understood as $\mathcal{L}(x_1, x_2, x_3; \mu)$.

Algorithm 1 n -block Randomly Permuted ADMM (RP-ADMM)

Initialization: $x_i^0 \in \mathbb{R}^{d_i \times 1}, i = 1, \dots, n; \mu^0 \in \mathbb{R}^{N \times 1}$.

Round k ($k = 0, 1, 2, \dots$):

1) Primal update.

Pick a permutation σ of $\{1, \dots, n\}$ uniformly at random.

For $i = 1, \dots, n$, compute $x_{\sigma(i)}^{k+1}$ by

$$x_{\sigma(i)}^{k+1} = \arg \min_{x_{\sigma(i)} \in \mathcal{X}_{\sigma(i)}} \mathcal{L}(x_{\sigma(1)}^{k+1}, \dots, x_{\sigma(i-1)}^{k+1}, x_{\sigma(i)}, x_{\sigma(i+1)}^k, \dots, x_{\sigma(n)}^k; \mu^k) \quad (5)$$

2) Dual update. Update the dual variable by

$$\mu^{k+1} = \mu^k - \beta \left(\sum_{i=1}^n A_i x_i^{k+1} - b \right). \quad (6)$$

2.1.1 Optimization Formulation of Solving a Linear System of Equations

Consider a special case of (1) where $f_i = 0$, $\mathcal{X}_i = \mathbb{R}^{d_i}$, $\forall i$ and $N = \sum_i d_i$ (i.e. the constraint is a square system of equations). Then problem (1) becomes

$$\begin{aligned} \min_{x \in \mathbb{R}^N} \quad & 0, \\ \text{s.t.} \quad & A_1 x_1 + \cdots + A_n x_n = b, \end{aligned} \quad (7)$$

where $A_i \in \mathbb{R}^{N \times d_i}$, $x_i \in \mathbb{R}^{d_i \times 1}$, $b \in \mathbb{R}^{N \times 1}$. Solving this feasibility problem (with 0 being the objective function) is equivalent to solving a linear system of equations

$$Ax = b, \quad (8)$$

where $A = [A_1, \dots, A_n] \in \mathbb{R}^{N \times N}$, $x = [x_1^T, \dots, x_n^T]^T \in \mathbb{R}^{N \times 1}$, $b \in \mathbb{R}^{N \times 1}$.

Throughout this paper, we assume A is non-singular. Then the unique solution to (8) is $x = A^{-1}b$, and problem (7) has a unique primal-dual optimal solution $(x, \mu) = (A^{-1}b, 0)$. The augmented Lagrangian function (3) for the optimization problem (7) becomes

$$\mathcal{L}(x, \mu) = -\mu^T(Ax - b) + \frac{\beta}{2}\|Ax - b\|^2. \quad (9)$$

Throughout this paper, we assume $\beta = 1$; note that our algorithms and results can be extended to any $\beta > 0$ by simply scaling μ .

2.1.2 Example of 3-block ADMM

Before presenting the update equation of general RP-ADMM for solving (7), we consider a simple case $N = n = 3$, $d_i = 1$, $\forall i$ and $\sigma = (123)$, and let $a_i = A_i \in \mathbb{R}^{3 \times 1}$. The update equations (5) and (6) can be rewritten as

$$\begin{aligned} -a_1^T \mu^k + a_1^T(a_1 x_1^{k+1} + a_2 x_2^k + a_3 x_3^k - b) &= 0, \\ -a_2^T \mu^k + a_2^T(a_1 x_1^{k+1} + a_2 x_2^{k+1} + a_3 x_3^k - b) &= 0, \\ -a_3^T \mu^k + a_3^T(a_1 x_1^{k+1} + a_2 x_2^{k+1} + a_3 x_3^{k+1} - b) &= 0, \\ (a_1 x_1^{k+1} + a_2 x_2^{k+1} + a_3 x_3^{k+1} - b) + \mu^{k+1} - \mu^k &= 0. \end{aligned}$$

Denote $y^k = [x_1^k; x_2^k; x_3^k; (\mu^k)^T] \in \mathbb{R}^{6 \times 1}$, then the above update equation becomes

$$\begin{bmatrix} a_1^T a_1 & 0 & 0 & 0 \\ a_2^T a_1 & a_2^T a_2 & 0 & 0 \\ a_3^T a_1 & a_3^T a_2 & a_3^T a_3 & 0 \\ a_1 & a_2 & a_3 & I_{3 \times 3} \end{bmatrix} y^{k+1} = \begin{bmatrix} 0 & -a_1^T a_2 & -a_1^T a_3 & a_1^T \\ 0 & 0 & -a_2^T a_3 & a_2^T \\ 0 & 0 & 0 & a_3^T \\ 0 & 0 & 0 & I_{3 \times 3} \end{bmatrix} y^k + \begin{bmatrix} A^T b \\ b \end{bmatrix}. \quad (10)$$

Define

$$L \triangleq \begin{bmatrix} a_1^T a_1 & 0 & 0 \\ a_2^T a_1 & a_2^T a_2 & 0 \\ a_3^T a_1 & a_3^T a_2 & a_3^T a_3 \end{bmatrix}, \quad R \triangleq \begin{bmatrix} 0 & -a_1^T a_2 & -a_1^T a_3 \\ 0 & 0 & -a_2^T a_3 \\ 0 & 0 & 0 \end{bmatrix}. \quad (11)$$

The relation between L and R is

$$L - R = A^T A.$$

Define

$$\bar{L} \triangleq \begin{bmatrix} L & 0 \\ A & I_{3 \times 3} \end{bmatrix}, \quad \bar{R} \triangleq \begin{bmatrix} R & A^T \\ 0 & I_{3 \times 3} \end{bmatrix}, \quad \bar{b} = \begin{bmatrix} A^T b \\ b \end{bmatrix} \quad (12)$$

then the update equation (10) becomes $\bar{L}y^{k+1} = \bar{R}y^k + \bar{b}$, i.e.

$$y^{k+1} = (\bar{L})^{-1} \bar{R}y^k + \bar{L}^{-1} \bar{b}. \quad (13)$$

As a side remark, reference Chen et al. [3] provides a specific example of $A \in \mathbb{R}^{3 \times 3}$ so that $\rho((\bar{L})^{-1} \bar{R}) > 1$, which implies the divergence of the above iteration if the update order $\sigma = (123)$ is used all the time. This counterexample disproves the convergence of cyclic 3-block ADMM.

2.1.3 General Update Equation of RP-ADMM

In general, for the optimization problem (7), the primal update (5) becomes

$$-A_{\sigma(i)}^T \mu^k + A_{\sigma(i)}^T \left(\sum_{j=1}^i A_{\sigma(j)} x_{\sigma(j)}^{k+1} + \sum_{l=i+1}^n A_{\sigma(l)} x_{\sigma(l)}^k - b \right) = 0, \quad i = 1, \dots, n. \quad (14)$$

Replacing $\sigma(i), \sigma(j), \sigma(l)$ by i, j, l , we can rewrite the above equation as

$$-A_i^T \mu^k + A_i^T \left(\sum_{\sigma^{-1}(j) \leq \sigma^{-1}(i)} A_j x_j^{k+1} + \sum_{\sigma^{-1}(l) > \sigma^{-1}(i)} A_l x_l^k - b \right) = 0, \quad i = 1, \dots, n, \quad (15)$$

where σ^{-1} denotes the inverse mapping of a permutation σ , i.e. $\sigma(i) = t \Leftrightarrow i = \sigma^{-1}(t)$. Denote the output of Algorithm 1 after round $(k-1)$ as

$$y^k \triangleq [x^k; \mu^k] = [x_1^k; \dots; x_n^k; \mu^k] \in \mathbb{R}^{2N \times 1}. \quad (16)$$

The update equations of Algorithm 1 for solving (7), i.e. (15) and (6), can be written in the matrix form as (when the permutation is σ and $\beta = 1$)

$$y^{k+1} = \bar{L}_\sigma^{-1} \bar{R}_\sigma y^k + \bar{L}_\sigma^{-1} \bar{b}, \quad (17)$$

where $\bar{L}_\sigma, \bar{R}_\sigma, L_\sigma, R_\sigma, \bar{b}$ are defined by

$$\bar{L}_\sigma \triangleq \begin{bmatrix} L_\sigma & 0 \\ A & I_{N \times N} \end{bmatrix}, \quad \bar{R}_\sigma \triangleq \begin{bmatrix} R_\sigma & A^T \\ 0 & I_{N \times N} \end{bmatrix}, \quad \bar{b} = \begin{bmatrix} A^T b \\ b \end{bmatrix}, \quad (18)$$

in which $L_\sigma \in \mathbb{R}^{N \times N}$ has $n \times n$ blocks and the (i, j) -th block is defined as

$$L_\sigma[i, j] \triangleq \begin{cases} A_i^T A_j & \sigma^{-1}(j) \leq \sigma^{-1}(i), \\ 0 & \text{otherwise.} \end{cases} \quad (19)$$

and R_σ is defined as

$$R_\sigma \triangleq L_\sigma - A^T A. \quad (20)$$

Another expression of L_σ , equivalent to (19), is the following:

$$L_\sigma[\sigma(i), \sigma(j)] \triangleq \begin{cases} A_{\sigma(i)}^T A_{\sigma(j)} & j \leq i, \\ 0 & j > i, \end{cases} \quad (21)$$

To illustrate the above expression of L_σ , we consider the n -coordinate case that $d_i = 1, \forall i$. In this case, each block x_i is a single coordinate, and each A_i is a vector. Denote $a_i \triangleq A_i \in \mathbb{R}^{N \times 1}$. Let $L_\sigma(k, l)$ denote the (k, l) -th entry of the matrix L_σ , then the definition (21) becomes

$$L_\sigma(\sigma(i), \sigma(j)) \triangleq \begin{cases} a_{\sigma(i)}^T a_{\sigma(j)} & j \leq i, \\ 0 & j > i, \end{cases} \quad (22)$$

A user-friendly rule for writing L_σ is described as follows (use $\sigma = (231)$ as an example). Start from a zero matrix. First, find all reverse pairs of σ ; here, we say (i, j) is a reverse pair if i appears after j in σ . For the permutation (231), all the reverse pairs are $(1, 3)$, $(3, 2)$ and $(1, 2)$. Second, in the positions corresponding to the reverse pairs, write down the corresponding entries of $A^T A$, i.e. $a_1^T a_3, a_3^T a_2$ and $a_1^T a_2$, respectively. At last, write $a_i^T a_i$ in the diagonal positions. Using this rule, we can write down the expression of $L_{(231)}$ as

$$L_{(231)} = \begin{bmatrix} a_1^T a_1 & a_1^T a_2 & a_1^T a_3 \\ 0 & a_2^T a_2 & 0 \\ 0 & a_3^T a_2 & a_3^T a_3 \end{bmatrix}.$$

A user-friendly rule to quickly check the correctness of an expression of L_σ is the following (still take $\sigma = (231)$ as an example). According to the order of the permutation (231), the 2nd row, the 3rd row and the 1st row should have a strictly decreasing number of zeros (2 zeros, 1 zero and no zero). In contrast, the 2nd column, the 3rd column and the 1st column should have a strictly increasing number of zeros.

For the general case that $d_i \geq 1, \forall i$, we can write down the block partitioned L_σ in a similar way. For example, when $n = 3$ and $\sigma = (231)$, we have

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

2.2 Randomly Permuted BCD

RP-ADMM is a generalization of RP-BCD. In fact, when the constraint does not exist, RP-ADMM reduces to RP-BCD. In this subsection, we present RP-BCD for solving convex quadratic problems. Note that RP-ADMM for solving linear systems and RP-BCD for solving quadratic problems are two extremes of general RP-ADMM: in the former case the objective function is zero, and in the latter case the constraint is zero. Interestingly, the two extreme cases are related as the expected iteration matrix of RP-BCD appears as a component of the expected iteration matrix of RP-ADMM. We will show later that their eigenvalues are closely related.

Consider a special case of (1) where $f(x) = \frac{1}{2} \|Ax - b\|^2$, $\mathcal{X}_i = \mathbb{R}^{d_i}, \forall i$ and there is no constraint. With abuse of notation, we use A to denote the coefficient matrix, while in the original formulation A denotes the constraint matrix. We “recycle” the notation A so that we can build a connection with RP-ADMM for solving linear systems later. Assume $N = \sum_i d_i$. Then problem (1) becomes a least-squares problem

$$\min_{x \in \mathbb{R}^N} \frac{1}{2} \|Ax - b\|^2 = \frac{1}{2} \|A_1 x_1 + \cdots + A_n x_n - b\|^2 \quad (23)$$

where $A_i \in \mathbb{R}^{N \times d_i}$, $x_i \in \mathbb{R}^{d_i \times 1}$, $b \in \mathbb{R}^{N \times 1}$. Similar to Section 2.1.1, we assume A is non-singular. Then the unique solution to (8) is $x = A^{-1}b$.

In the augmented Lagrangian function given in (9), if we delete the first term which depends on the dual variable μ , we obtain the quadratic function $\frac{1}{2}\|Ax - b\|^2$. Thus if we eliminate the dual variable μ in the update equations of RP-ADMM, we will obtain the update equations for RP-BCD. Suppose x^k is the iterate after the k -th epoch (i.e. go through all coordinates once), and σ is the order used in the k -th iteration, then, as a simpler version of (17), we have

$$x^{k+1} = L_\sigma^{-1} R_\sigma x^k + L_\sigma^{-1} b, \quad (24)$$

where L_σ and R_σ are defined as in (21) and (20), and σ is a random permutation.

2.3 Residual Trick for Efficient Implementation of ADMM and BCD

We note here that when $d_i = 1, \forall i$ (in this case BCD becomes CD), per-epoch computation time of ADMM and CD (no matter what order) is $O(n^2)$; or in other words, per-coordinate-update time is $O(n)$. For instance, updating x^{k+1} by (24) in RP-BCD or updating y^{k+1} by (17) in RP-ADMM only takes time $O(n^2)$. As mentioned in Section 3.1 of [40], the trick is to keep track of the residual. For both efficient practical implementation and calculation of computation complexity, one should use this residual trick, but for the ease of theoretical analysis we use the matrix update forms (17) and (24) in this paper; there is no contradiction as our theory only depends on the value of x^k but not the specific procedure to compute x^k .

For completeness, we briefly explain how this trick works in our settings. Suppose $d_i = 1, \forall i$, and we use CD methods to solve (23) with a certain update order (could be any order, such as cyclic, randomized or randomly permuted). Suppose the coordinate i is picked, then x_i is updated by

$$x_i^+ = \frac{1}{A_i^T A_i} [A_i^T (b - A_{-i} x_{-i})], \quad (25)$$

where A_{-i} contains all columns of A except A_i , x_{-i} contains all elements of x except x_i and represents the current values, and x_i^+ represents the new value. A straightforward implementation of (25) requires multiplying x_{-i} by A_{-i} which takes $O(n^2)$ operations. With the residual trick (e.g. [40]), we introduce the residual $r = Ax - b$, and replace (25) by

$$\begin{aligned} x_i^+ &= x_i - \frac{1}{A_i^T A_i} A_i^T r, \\ r^+ &= r + A_i (x_i^+ - x_i). \end{aligned}$$

Now the calculation of x_i^+ and r^+ takes time $O(n)$, and thus one epoch of BCD takes time $O(n^2)$. The same trick can be applied to the primal update of ADMM; with this trick, the dual update (6) can be rewritten as $\mu^+ = \mu - \beta r$ which takes time $O(n)$, and thus one epoch of ADMM takes time $O(n^2)$.

Finally, when $d_i > 1$, similar update equations can still be used except a minor difference that $\frac{1}{A_i^T A_i}$ should be replaced by $(A_i^T A_i)^{-1}$. In a special case that $d_i = d, \forall i$ and $N = dn$, each iteration of BCD takes time $O(Nd + d^3)$ and each epoch takes time $O(N^2 + Nd^2)$. This cost can be reduced if we use BCGD (i.e. not solving the subproblem exactly but updating each block of variables by a gradient step). In order not to make the paper more complicated, we will not discuss the inexact versions of BCD and ADMM in this paper.

2.4 Two Versions of Independently Randomized ADMM

In this subsection, we present two other versions of randomized ADMM which can be divergent according to simulations. The failure of these versions makes us focus on analyzing RP-ADMM in this paper. These versions can be viewed as natural extensions of R-BCD (randomized BCD) [41] and [40].

In the first algorithm, called primal-dual randomized ADMM (PD-RADMM), the whole dual variable is viewed as the $(n + 1)$ -th block. In particular, at each iteration, the algorithm draws one index i from $\{1, \dots, n, n + 1\}$, then performs the following update: if $i \leq n$, update the i -th block of the primal variable; if $i = n + 1$, update the whole dual variable. The details are given in Algorithm 2. We have tested PD-RADMM for the counter-example given in Chen et al. [3], and found that PD-RADMM always diverges (for random initial points).

A variant of PD-RADMM has been proposed in Hong et al. [17] with two differences: first, instead of minimizing the augmented Lagrangian \mathcal{L} , that algorithm minimizes a strongly convex upper bound of \mathcal{L} ; second, that algorithm uses a diminishing dual stepsize. With these two modifications, [17] shows that each limit point of the sequence generated by their algorithm is a primal-dual optimum with probability 1. Note that [17] also proves the same convergence result for the cyclic version of multi-block ADMM with these two modifications, thus it does not show the benefit of randomization.

Algorithm 2 Primal-Dual Randomized ADMM (PD-RADMM)

Iteration t ($t = 0, 1, 2, \dots$):

Pick $i \in \{1, \dots, n, n + 1\}$ uniformly at random;

If $1 \leq i \leq n$:

$$\begin{aligned} x_i^{t+1} &= \arg \min_{x_i \in \mathcal{X}_i} \mathcal{L}(x_1^t, \dots, x_{i-1}^t, x_i, x_{i+1}^t, \dots, x_n^t; \mu^t), \\ x_j^{t+1} &= x_j^t, \quad \forall j \in \{1, \dots, n\} \setminus \{i\}, \\ \mu^{t+1} &= \mu^t. \end{aligned}$$

Else If $i = n + 1$:

$$\begin{aligned} \mu^{t+1} &= \mu^t - \beta(\sum_{i=1}^n A_i x_i^{t+1} - b), \\ x_j^{t+1} &= x_j^t, \quad \forall j \in \{1, \dots, n\}. \end{aligned}$$

End

In the second algorithm, called primal randomized ADMM (P-RADMM), we only perform randomization for the primal variables. In particular, at each round, we first draw n independent random variables j_1, \dots, j_n from the uniform distribution of $\{1, \dots, n\}$ and update x_{j_1}, \dots, x_{j_n} sequentially, then update the dual variable in the usual way. The details are given in Algorithm 3. This algorithm looks quite similar to RP-ADMM as they both update n primal blocks at each round; the difference is that RP-ADMM samples *without replacement* while this algorithm P-RADMM samples *with replacement*. In other words, RP-ADMM updates each block exactly once at each round, while P-RADMM may update one block more than one times or does not update one block at each round.

We have tested P-RADMM in various settings. For the counter-example given in Chen et al. [3], we found that P-RADMM does converge. However, if $n \geq 30$ and A is a Gaussian random matrix (each entry is drawn i.i.d. from $\mathcal{N}(0, 1)$), then P-RADMM diverges in almost all cases we have tested. This phenomenon is rather strange since for random Gaussian matrices A the cyclic ADMM actually converges (according to simulations). An implication is that randomized versions do not always outperform

their deterministic counterparts in terms of convergence.

Since both Algorithm 2 and Algorithm 3 can diverge in certain cases, we will not further study them in this paper. In the rest of the paper, we will focus on RP-ADMM (i.e. Algorithm 1).

Algorithm 3 Primal Randomized ADMM (P-RADMM)

Round k ($k = 0, 1, 2, \dots$):

1) Primal update.

Pick l_1, \dots, l_n independently from the uniform distribution of $\{1, \dots, n\}$.

For $i = 1, \dots, n$:

$$t = kn + i - 1,$$

$$x_{l_i}^{t+1} = \arg \min_{x_{l_i} \in \mathcal{X}_{l_i}} \mathcal{L}(x_1^t, \dots, x_{l_i-1}^t, x_{l_i}, x_{l_i+1}^t, \dots, x_n^t; \mu^t),$$

$$x_j^{t+1} = x_j^t, \quad \forall j \in \{1, \dots, n\} \setminus \{l_i\},$$

$$\mu^{t+1} = \mu^t.$$

End.

2) Dual update.

$$\mu^{(k+1)n} = \mu^{kn} - \beta(\sum_{i=1}^n A_i x_i^{(k+1)n} - b).$$

2.5 Bernoulli-Randomized ADMM

To implement randomly permuted ADMM, one needs to sample from all blocks without replacement. To save the sampling time, we propose another algorithm which we call Bernoulli-randomized ADMM. This algorithm is motivated by the proof of Theorem 1. This updating scheme can be applied to other algorithms such as SGD and coordinate descent methods.

The new update order combines the well-known double-sweep order and Bernoulli-randomization. The original double-sweep order is $(1, 2, \dots, n-1, n, n-1, n-2, \dots, 1)$, meaning that $x_1, x_2, \dots, x_{n-1}, x_n, x_{n-1}, x_{n-2}, \dots, x_1$ are updated sequentially in each “cycle”. It combines the normal cyclic order $(1, 2, \dots, n)$ and a reverse order $(n, n-1, \dots, 1)$. We propose the following updating scheme: add a check box to each block, and in each cycle we perform the following operations.

1. Phase I: go through the blocks x_1, x_2, \dots, x_n one by one sequentially as follows: for each block x_i , flip a fair coin and:
 - (a) if the outcome is “head”, update the block x_i and check the check box;
 - (b) if the outcome is “tail”, do nothing about x_i and uncheck the check box.
2. Phase II: go through the blocks x_n, x_{n-1}, \dots, x_1 in the reverse order, and update x_i if the box is unchecked.

Note that in each cycle we go through each block twice but update each block exactly once so that the number of totally updated blocks remains n . For example, when $n = 5$, (35421) is a possible update order, as shown in the following diagram. Similarly, (13542) is also a possible update order. But (13524) and (35412) are not possible. The set of all possible update orders is given by

$$\Gamma_{\text{BR}} \triangleq \{\sigma \in \Gamma \mid \exists i \in \{1, \dots, n-1\} \text{ such that } \sigma(1) < \sigma(2) < \dots < \sigma(i) \text{ and } \sigma(i+1) > \dots > \sigma(n)\},$$

		1	2	3	4	5
Phase I	begin	skip \rightarrow	skip \rightarrow	3 \rightarrow	skip \rightarrow	5
						\downarrow
Phase II	end	1	\leftarrow 2	\leftarrow skip	\leftarrow 4	\leftarrow skip

where Γ is the set of permutations of $\{1, 2, \dots, n\}$ as defined in (4). In other words, a sequence from Γ_{BR} is a concatenation of an increasing sequence and a decreasing sequence. Note that the permutation $(1, 2, \dots, n)$ is in Γ_{BR} since it can be viewed as the concatenation of an increasing sequence $(1, 2, \dots, n-1)$ and a “decreasing sequence” (n) , and we can let $i = n-1$ in the above definition to cover this case. Similarly, the permutation $(n, n-1, \dots, 1)$ is also in Γ_{BR} as $i = 1$ will cover this case.

The algorithm Bernoulli-randomized ADMM (BR-ADMM) is formally described below. We skip the epoch index k since otherwise the notation would be cumbersome.

Algorithm 4 n -block Bernoulli-Randomized ADMM (BR-ADMM)

Initialization: $x_i^0 \in \mathbb{R}^{d_i \times 1}, i = 1, \dots, n; \mu^0 \in \mathbb{R}^{N \times 1}$.

Round k ($k = 0, 1, 2, \dots$):

1) Primal update.

Set $c_i = 0, i = 1, \dots, n$.

Phase I.

For $i = 1, 2, \dots, n$:

Draw a random variable $\xi \sim \text{Bernnolli}(1/2)$, i.e. $Pr(\xi = 1) = Pr(\xi = 0) = 1/2$.

If $\xi = 1$: set $c_i = 1$ and update x_i by

$$x_i \leftarrow \arg \min_{x_i \in \mathcal{X}_i} \mathcal{L}(x_1, \dots, x_{i-1}, x_i, x_{i+1}, \dots, x_n; \mu). \quad (26)$$

Phase II.

For $i = n, n-1, \dots, 1$: if $c_i = 0$, update x_i by (26).

2) Dual update. Update the dual variable by

$$\mu \leftarrow \mu - \beta \left(\sum_{i=1}^n A_i x_i - b \right). \quad (27)$$

For solving linear systems of equations, the update formula is the same as (17), the update formula of RP-ADMM. The difference is that for RP-ADMM σ can be an arbitrary permutation, while for BR-ADMM there is some restriction on σ : it has to be a permutation in Γ_{BR} .

3 Main Results

3.1 Expected Convergence of RP-ADMM

Let σ_i denote the permutation used in round i of Algorithm 1, which is a uniform random variable drawn from the set of permutations Γ . After round k , Algorithm 1 generates a random output y^{k+1} , which depends on the observed draw of the random variable

$$\xi_k = (\sigma_0, \sigma_1, \dots, \sigma_k). \quad (28)$$

We will show that the expected iterate (the iterate y^k is defined in (16))

$$\phi^k = E_{\xi_{k-1}}(y^k) \quad (29)$$

converges to the primal-dual solution of the problem (7). Although the expected convergence does not necessarily imply the convergence in a particular realization, it serves as an evidence of convergence. Our proof seems much different from and more difficult than previous proofs for other randomized methods, since random permutation, as well as spectral radius of non-symmetric matrices, are difficult objects to deal with – not many existing mathematical tools are available to help ³. Note that the extension of this result to the non-square full column-rank case is simple ⁴.

Theorem 1 *Assume the coefficient matrix $A = [A_1, \dots, A_n]$ of the constraint in (7) is a non-singular square matrix. Suppose Algorithm 1 is used to solve problem (7), then the expected output converges to the unique primal-dual optimal solution to (7), i.e.*

$$\{\phi^k\}_{k \rightarrow \infty} \rightarrow \begin{bmatrix} A^{-1}b \\ 0 \end{bmatrix}. \quad (30)$$

Since the update matrix does not depend on previous iterates, we claim (and prove in Section 4.1) that Theorem 1 holds if the expected update matrix has a spectral radius less than 1, i.e. if the following Theorem 2 holds.

Theorem 2 *Suppose $A = [A_1, \dots, A_n] \in \mathbb{R}^{N \times N}$ is non-singular, and $\bar{L}_\sigma^{-1}, \bar{R}_\sigma$ are defined by (18) for any permutation σ . Define*

$$M \triangleq E_\sigma(\bar{L}_\sigma^{-1} \bar{R}_\sigma) = \frac{1}{n!} \sum_{\sigma \in \Gamma} (\bar{L}_\sigma^{-1} \bar{R}_\sigma), \quad (31)$$

where the expectation is taken over the uniform random distribution over Γ , the set of permutations of $\{1, 2, \dots, n\}$. Then the spectral radius of M is smaller than 1, i.e.

$$\rho(M) < 1. \quad (32)$$

³There has been some effort in using random matrix theory to tackle this problem but no progress has been reported to our knowledge. This is partially due to the fact that the desired result seems to be rather tight such that even a small relaxation can lead to failure.

⁴Suppose A is an $m \times n$ full column-rank matrix, where $m \geq n$, and the system $Ax = b$ is feasible. The update formula is $y^{k+1} = (I - L_\sigma^{-1} A^T A)y^k$, which is same as the update formula for solving a square system of equations $\bar{A}x = b$, where $\bar{A} \in \mathbb{R}^{n \times n}$ is the square root matrix of the matrix $A^T A \in \mathbb{R}^{n \times n}$. Now the matrix \bar{A} is a square invertible matrix, thus by applying the result for square system of equations, we can obtain the convergence of the sequence $\phi^k = E(y^k)$.

Remark 3.1 For the counterexample in Chen et al. [3] where $A = [1, 1, 1; 1, 1, 2; 1, 2, 2]$, it is easy to verify that $\rho(M_\sigma) > 1.02$ for any permutation σ of $(1, 2, 3)$. Interestingly, Theorem 2 shows that even if each M_σ is “bad” (with spectral radius larger than 1), the average of them is always “good” (with spectral radius smaller than 1).

Theorem 2 is just a linear algebra result, and can be understood even without knowing the details of the algorithm. However, the proof of Theorem 2 is rather non-trivial. This proof will be provided in Section 4.2, and the technical results used in this proof will be proved in Section 5 and Section 6.

The convergence rate of RP-ADMM for solving linear systems of equations is closely related to the convergence rate of RP-BCD (randomly permuted BCD) for solving quadratic problems. We will discuss their relation and how our results in this paper improve our understanding for RP-BCD.

A similar convergence result holds for BR-ADMM proposed in Section 2.5, as presented below. The proof is a simple modification of the proof of Theorem 1, and can be found in Section 6.4.

Proposition 1 Assume the coefficient matrix $A = [A_1, \dots, A_n]$ of the constraint in (7) is a non-singular square matrix. Suppose Algorithm 4 is used to solve problem (7), then the expected output converges to the unique primal-dual optimal solution to (7).

3.2 Expected Convergence Rate of RP-ADMM and RP-BCD

There is a close relation between RP-ADMM for solving linear systems and RP-CD for solving quadratic problems (see Lemma 2). Thus it is not surprising that we need to understand RP-BCD before understanding RP-ADMM. We will first present an expected convergence rate of RP-BCD (in terms of the expected iterates) for solving quadratic problems, which improves the best existing convergence rate (one type of rates, to be precise) by a factor of n^5 . The result is proved via establishing a weak version of matrix AM-GM inequality. This result also establishes a large gap of $O(n)$ between RP-BCD and C-BCD (cyclic BCD). Second, built upon the result for RP-BCD, we establish a convergence rate of RP-ADMM which is similar to RP-BCD and also n times better than that of C-BCD.

The first result is about the expected convergence rate of RP-BCD for the case $A_i^T A_i = I$. This assumption is made so that the expression is simple, and the case for general A_i is given in the next result.

Theorem 3 (rate of RP-BCD for quadratic functions with identity diagonal blocks) Assume the coefficient matrix $A = [A_1, \dots, A_n]$ is a non-singular square matrix, and $A_i^T A_i = I, \forall i$. Suppose RP-BCD is used to solve problem (23), where x^k denotes the variable after k epochs (each epoch represents one cycle of updating all coordinates). Denote the unique optimal solution as $x^* = A^{-1}b$. Then

$$\|E(x^k) - x^*\| \leq \max \left\{ 1 - \frac{1}{n} \lambda_{\min}(AA^T), \frac{1}{3} \right\}^k \|x^0 - x^*\|. \quad (33)$$

To put this convergence rate result in the context, we consider the simple case that each $d_i = 1$, i.e., each block consists of a single coordinate. In this case, every diagonal entry of $A^T A$ is 1, thus the

⁵Rigorously speaking, this is not a fair comparison as the complexity of C-CD is deterministic complexity.

Table 1: Worst-case computation complexity comparison, using only κ_{CD} as parameter, for equal-diagonal quadratic case (ignore $O(\log \frac{1}{\epsilon})$ factor), and consider the error in the expected iterates for RP-CD

	GD	C-CD	R-CD	RP-CD (Theorem 3)	RP-CD (conjectured)
Computation Complexity	$n^3 \kappa_{\text{CD}}$	$n^4 \kappa_{\text{CD}}$	$n^2 \kappa_{\text{CD}}$	$n^3 \kappa_{\text{CD}}$	$n^2 \kappa_{\text{CD}}$

average eigenvalue of $A^T A$ is 1. Throughout the paper, we consider the total computation complexity⁶; note that we assume the residual trick as described in 2.3 is always used for all methods.

Our Theorem 3 provides an expected computational complexity upper bound $O(n^3 \kappa_{\text{CD}} \log \frac{1}{\epsilon})$ for RP-CD, since each epoch takes $O(n^2)$ time and it requires $O(n^2 \log \frac{1}{\epsilon})$ epochs to achieve error ϵ according to (33). It is known that the computational complexity of R-CD (randomized coordinate descent) to achieve relative accuracy ϵ ⁷ is $O(n^2 \kappa_{\text{CD}} \log \frac{1}{\epsilon})$, where $\kappa_{\text{CD}} = \lambda_{\text{avg}}(A^T A) / \lambda_{\min}(A^T A) = 1 / \lambda_{\min}(A^T A)$ is the ratio of the average eigenvalue over the minimum eigenvalue. It was recently shown that in terms of κ_{CD} and n only, the worst-case complexity of C-CD (cyclic CD) is $O(n^4 \kappa_{\text{CD}} \log \frac{1}{\epsilon})$, which is n^2 times worse than R-CD and n times worse than GD. This shows a large gap between C-CD and R-CD in the worst case.

It was widely conjectured that RP-CD is at least as fast as R-CD, but this conjecture is considered to be rather difficult to prove. For a special class of matrices, recent works [36, 42] validated the conjecture. However, to our knowledge, even for a general quadratic function with equal diagonal entries 1, the previously best known convergence rate of RP-CD is almost the same as C-CD (see [35] [26]), which can be n^2 times worse than that of R-CD. Our Theorem 3 provides an expected computational complexity upper bound $O(n^3 \kappa_{\text{CD}} \log \frac{1}{\epsilon})$ for RP-CD, which is n times faster than C-CD and n times slower than R-CD. This improves the best existing rate by a factor of n ⁸. We summarize the comparison of the complexity for C-CD, R-CD and RP-CD in Table 1.

The following proposition generalizes Theorem 3 to the non-identity-diagonal case, i.e., $A_i^T A_i$ does not need to be an identity matrix.

Proposition 2 (*rate of RP-BCD for quadratic functions, with non-identity blocks*) Assume the coefficient matrix $A = [A_1, \dots, A_n]$ is a non-singular square matrix. Suppose RP-BCD is used to solve problem (23). Denote $D = \text{diag}(A_1^T A_1, \dots, A_n^T A_n)$ as a block-diagonal matrix, and the norm $\|z\|_D = \sqrt{z^T D z}$. Then

$$\|E(x^k) - x^*\|_D \leq \max \left\{ 1 - \frac{1}{n} \lambda_{\min}(D^{1/2} A^T A D^{-1/2}), \frac{1}{3} \right\}^k \|x^0 - x^*\|_D. \quad (34)$$

The proof of Proposition 2 is given in Section 4.4. One can easily transform the quantity $\lambda_{\min}(D^{1/2} A^T A D^{-1/2})$

⁶The computation complexity equals the iteration complexity times the per-iteration cost. We do not present iteration complexity since there may be confusion about whether “one iteration” means n coordinate updates or 1 coordinate update. Presenting iteration complexity is better if one considers a general convex problem, but then one needs to discuss the per-iteration cost. We are considering quadratic problems throughout the paper, so we feel it is more clear to stick to computation complexity.

⁷Here, the relative accuracy ϵ means $\|E(x^k) - x^*\| / \|x^0 - x^*\|$ or $\|E(f(x^k)) - f^*\| / \|f(x^0) - f^*\|$.

⁸Note that this “improvement” is valid when the convergence rate is characterized by only κ_{CD} and n . It is common to use other parameters such as the maximum eigenvalue to characterize the convergence rate (see [26] for a detailed discussion), and our result here does not provide improvement for other kinds of convergence rate.

to certain quantity that only depends on the eigenvalues of $A_i^T A_i$ and $A^T A$. However, as noted in [26], it is far from clear how tight the transformation is, thus we skip the transformation here. In fact, it is related to some open question on the so-called Jacobi-preconditioning. We refer the interested readers to [26] for a detailed discussion of the subtle issues in the non-identity-diagonal case.

At last, we present a result on the expected convergence rate of RP-ADMM for solving linear systems, under the assumption that $A_i^T A_i = I$, $\forall i$. Very similar to Proposition 2, we can also generalize this result to non-identity-diagonal case, i.e., $A_i^T A_i \neq I$, but to save space we skip the generalization here. The proof of Theorem 4 is given in Section 4.5.

Theorem 4 (*Expected convergence rate of RP-ADMM for linear systems*) Assume the coefficient matrix $A = [A_1, \dots, A_n]$ of the constraint in (7) is a non-singular square matrix and $A_i^T A_i = I_{d_i}$. Suppose Algorithm 1 is used to solve problem (7). Denote $y^* = \begin{bmatrix} A^{-1}b \\ 0 \end{bmatrix}$ as the unique primal-dual optimal solution to the problem (7), then

$$\|E(y^k) - y^*\| \leq \left(1 - \frac{1}{2n} \lambda_{\min}(AA^T)\right)^k \|y^0 - y^*\|. \quad (35)$$

This result implies that similar to RP-CD for solving quadratic problems, the complexity of RP-ADMM in terms of the expected iterates for solving linear systems is also at most

$$T_{\text{RP-ADMM}} = O(n^3 \kappa_{\text{CD}} \log(1/\epsilon)).$$

In light of the fact that C-CD has been shown to only achieve a rate $O(n^4 \kappa_{\text{CD}} \log(1/\epsilon))$ [26], the rate of RP-ADMM we obtain is already quite good. Nevertheless, we conjecture that this complexity upper bound can be improved to $O(n^2 \kappa_{\text{CD}} \log(1/\epsilon))$, the same as the conjectured complexity for RP-CD. But an improved rate of RP-ADMM leads to an improved rate of RP-BCD (this should be clear via the comparison of (50) and (57)), thus proving this conjecture is an even more difficult problem than the long-standing open question on RP-CD.

3.3 Matrix AM-GM Inequality

To analyze the convergence rate of randomly permuted algorithms, one major technical challenge is matrix AM-GM (algebraic mean-geometric mean) inequality. The following conjecture of matrix AM-GM inequality was proposed in [37]: for any positive semi-definite matrix $A_1, \dots, A_n \in \mathbb{R}^{n \times n}$,

$$\left\| \frac{1}{n!} \sum_{\sigma=(\sigma_1, \dots, \sigma_n) \in \Gamma} A_{\sigma_n} A_{\sigma_{n-1}} \dots A_{\sigma_1} \right\| \leq \left\| \left(\frac{1}{n} \sum_i A_i \right)^n \right\|. \quad (36)$$

The original version is more general: the number of matrices does not need to be the same as the dimension of the matrix. For simplicity, we just present a simpler version here.

The matrix AM-GM inequality is a generalization of the well-known AM-GM inequality: for non-negative numbers a_1, \dots, a_n , the geometric mean $(a_1 a_2 \dots a_n)^{1/n}$ is no more than the algebraic mean $\frac{1}{n} \sum_{i=1}^n a_i$. When extending this inequality to matrix domain, the non-commutative nature of matrix multiplication makes the problem rather difficult to prove.

We observe that we only need to prove a matrix AM-GM inequality for projection matrices. We conjecture that the following matrix AM-GM inequality holds.

Conjecture 3.1 (*matrix AM-GM inequality for projection matrices*) Suppose $P_i \in \mathbb{R}^{N \times N}$, $i = 1, \dots, n$ are projection matrices, then

$$\frac{1}{n!} \sum_{\sigma=(\sigma_1, \dots, \sigma_n) \in \Gamma} P_{\sigma_n} P_{\sigma_{n-1}} \dots P_{\sigma_1} \preceq \left(\frac{1}{n} \sum_i P_i \right)^n. \quad (37)$$

Compared with (36), our conjecture makes a stronger claim on the relation, but it only applies to projection matrices. We have found examples to show that (37) does not hold for general positive semi-definite matrices, but it holds for projection matrices in all of our experiments.

We are not able to prove the new conjecture – that would solve the open question of the best convergence rate of RP-CD for quadratic problem. Nevertheless, inspired by the new conjecture, we prove a weaker version (see Lemma 3), which can lead to an improved convergence rate estimate for RP-CD.

4 Proof of Main Results

4.1 Proof of Theorem 1

Denote σ_k as the permutation used in round k , and define ξ_k as in (28). Rewrite the update equation (17) below (replacing σ by σ_k):

$$y^{k+1} = \bar{L}_{\sigma_k}^{-1} \bar{R}_{\sigma_k} y^k + \bar{L}_{\sigma_k}^{-1} \bar{b}. \quad (38)$$

We first prove (30) for the case $b = 0$. By (18) we have $\bar{b} = 0$, then (38) is simplified to $y^{k+1} = \bar{L}_{\sigma_k}^{-1} \bar{R}_{\sigma_k} y^k$. Taking the expectation of both sides of this equation in ξ_k (see its definition in (28)), and note that y^k is independent of σ_k , we get

$$\phi^{k+1} = E_{\xi_k}(\bar{L}_{\sigma_k}^{-1} \bar{R}_{\sigma_k} y^k) = E_{\sigma_k}(E_{\xi_{k-1}}(\bar{L}_{\sigma_k}^{-1} \bar{R}_{\sigma_k} y^k)) = E_{\sigma_k}(\bar{L}_{\sigma_k}^{-1} \bar{R}_{\sigma_k} \phi^k) = M \phi^k.$$

Since the spectral radius of M is less than 1 by Theorem 2, we have that $\{\phi^k\} \rightarrow 0$, i.e. (30).

We then prove (30) for general b . Let $y^* = [A^{-1}b; 0]$ denote the optimal solution. Then it is easy to verify that

$$y^* = \bar{L}_{\sigma_k}^{-1} \bar{R}_{\sigma_k} y^* + \bar{L}_{\sigma_k}^{-1} \bar{b}$$

for all $\sigma_k \in \Gamma$ (i.e. the optimal solution is the fixed point of the update equation for any order). Compute the difference between this equation and (38) and letting $\hat{y}^k = y^k - y^*$, we get $\hat{y}^{k+1} = \bar{L}_{\sigma_k}^{-1} \bar{R}_{\sigma_k} \hat{y}^k$. According to the proof for the case $b = 0$, we have $E(\hat{y}^k) \rightarrow 0$, which implies $E(y^k) \rightarrow y^*$.

4.2 Proof of Theorem 2

The difficulty of proving Theorem 2 (bounding the spectral radius of M defined in (31)) is two-fold. First, M is a non-symmetric matrix, and there are very few tools to bound the spectral radius of a

non-symmetric matrix. In fact, spectral radius is neither subadditive nor submultiplicative (see, e.g. Kittaneh [43]). Note that the spectral norm of M can be much larger than 1 (there are examples that $\|M\| > 2$), thus we cannot bound the spectral radius simply by the spectral norm. Second, although it is possible to explicitly write each entry of M as a function of the entries of $A^T A$, these functions are very complicated (n -th order polynomials) and it is not clear how to utilize this explicit expression.

The proof outline of Theorem 2 and the main techniques are described below. In Step 0, we provide an expression of the expected update matrix M . In Step 1, we establish the relationship between the eigenvalues of M and the eigenvalues of a simple symmetric matrix AQA^T , where Q is defined in (39). As a consequence, the spectral radius of M is smaller than one iff the eigenvalues of AQA^T lie in the region $(0, 4/3)$. This step partially resolves the first difficulty, i.e. how to deal with the spectral radius of a non-symmetric matrix. In Step 2, we show that the eigenvalues of AQA^T do lie in $(0, 4/3)$ using mathematical induction. The induction analysis circumvents the second difficulty, i.e. how to utilize the relation between M and A .

Step 0: compute the expression of the expected update matrix M . Define

$$Q \triangleq E_\sigma(L_\sigma^{-1}) = \frac{1}{n!} \sum_{\sigma \in \Gamma} L_\sigma^{-1}. \quad (39)$$

It is easy to prove that Q defined by (39) is symmetric. In fact, note that $L_\sigma^T = L_{\bar{\sigma}}$, $\forall \sigma \in \Gamma$, where $\bar{\sigma}$ is a reverse permutation of σ satisfying $\bar{\sigma}(i) = \sigma(n+1-i)$, $\forall i$, thus $Q = \frac{1}{n!} \sum_{\sigma} Q_\sigma = (\frac{1}{n!} \sum_{\sigma} Q_{\bar{\sigma}})^T = Q^T$, where the last step is because the sum of all $Q_{\bar{\sigma}}$ is the same as the sum of all Q_σ .

Denote

$$M_\sigma \triangleq \bar{L}_\sigma^{-1} \bar{R}_\sigma = \bar{L}_\sigma^{-1} \begin{bmatrix} R_\sigma & A^T \\ 0 & I \end{bmatrix}. \quad (40)$$

Substituting the expression of \bar{L}_σ^{-1} into the above relation, and replacing R_σ by $L_\sigma - A^T A$, we obtain

$$M_\sigma = \begin{bmatrix} L_\sigma^{-1} & 0 \\ -AL_\sigma^{-1} & I \end{bmatrix} \begin{bmatrix} L_\sigma - A^T A & A^T \\ 0 & I \end{bmatrix} = \begin{bmatrix} I - L_\sigma^{-1} A^T A & L_\sigma^{-1} A^T \\ -A + AL_\sigma^{-1} A^T A & I - AL_\sigma^{-1} A^T \end{bmatrix}. \quad (41)$$

Since M_σ is linear in L_σ^{-1} , we have

$$\begin{aligned} M &= E_\sigma(M_\sigma) = \begin{bmatrix} I - E_\sigma(L_\sigma^{-1}) A^T A & E_\sigma(L_\sigma^{-1}) A^T \\ -A + AE_\sigma(L_\sigma^{-1}) A^T A & I - AE_\sigma(L_\sigma^{-1}) A^T \end{bmatrix} \\ &= \begin{bmatrix} I - QA^T A & QA^T \\ -A + AQA^T A & I - AQA^T \end{bmatrix}. \end{aligned} \quad (42)$$

Step 1: relate M to a simple symmetric matrix. The main result of Step 1 is given below, and the proof of this result is relegated to Section 5.

Lemma 1 Suppose $A \in \mathbb{R}^{N \times N}$ is non-singular and $Q \in \mathbb{R}^{N \times N}$ is an arbitrary matrix. Define $M \in \mathbb{R}^{2N \times 2N}$ as

$$M = \begin{bmatrix} I - QA^T A & QA^T \\ -A + AQA^T A & I - AQA^T \end{bmatrix}. \quad (43)$$

Then

$$\lambda \in \text{eig}(M) \iff \frac{(1-\lambda)^2}{1-2\lambda} \in \text{eig}(QA^T A). \quad (44)$$

Furthermore, when Q is symmetric, we have

$$\rho(M) < 1 \iff \text{eig}(QA^T A) \subseteq (0, \frac{4}{3}). \quad (45)$$

Remark: For our problem, the matrix Q as defined by (39) is symmetric (see the argument after equation (39)), thus the relation (45) indeed holds according to Lemma 1. For a general non-symmetric Q , (45) does not need to hold, but the first conclusion (44) still holds.

Step 2: Bound the eigenvalues of $QA^T A$. The main result of Step 2 is summarized in the following Lemma 2. The proof of Lemma 2 is given in Section 6.

Lemma 2 Suppose $A = [A_1, \dots, A_n] \in \mathbb{R}^{N \times N}$ is non-singular. Define Q as

$$Q \triangleq E_\sigma(L_\sigma^{-1}) = \frac{1}{n!} \sum_{\sigma \in \Gamma} L_\sigma^{-1}, \quad (46)$$

in which L_σ is defined by (21) and Γ is defined by (4). Then all eigenvalues of $QA^T A$ lie in $(0, 4/3)$, i.e.

$$\text{eig}(QA^T A) \subseteq (0, \frac{4}{3}). \quad (47)$$

Remark: The upper bound $\frac{4}{3}$ in (47) is probably tight, since we have found numerical examples with $\text{eig}(QA^T A) > 1.3333$. Now the expected convergence of RP-ADMM seems to be a pleasant coincidence: Lemma 1 shows that to prove the expected convergence we need to prove $\sup_A \text{eig}(QA^T A)$, a quantity that can be defined without knowing ADMM, is bounded by $4/3$; Lemma 2 and numerical experiments show that this quantity happens to be exactly $4/3$ so that RP-ADMM can converge (in expectation).

Theorem 2 follows immediately from Lemma 1 and Lemma 2.

4.3 Proof of Theorem 3

We first describe the outline of the proof. The expected update matrix of RP-BCD is $I - QA^T A$, and the eigenvalues of this matrix lie in $(-1, 1)$. The expected convergence speed of RP-BCD depends on the distance between the eigenvalues and the two extremes -1 and 1 . Lemma 2 shows that the distance to -1 is at least $1/3$, which is a constant. We will show that the distance to 1 is at least $\lambda_{\min}(A^T A)/n$, by proving a weaker version of matrix AM-GM inequality. Combining the two results, we obtain the expected convergence speed of RP-BCD.

The formal proof is presented below.

According to (24), we have $x^{k+1} - x^* = (I - L_\sigma^{-1} A^T A)(x^k - x^*)$, where σ is the randomly picked permutation at the k -th epoch. Therefore, the expected update formula of RP-BCD for solving the least squares problem is

$$E(x^{k+1}) - x^* = (I - QA^T A)(E(x^k) - x^*). \quad (48)$$

It implies

$$\|E(x^{k+1}) - x^*\| \leq \rho(I - QA^T A) \|E(x^k) - x^*\|. \quad (49)$$

Suppose the eigenvalues of QA^TA are $\eta_1 \geq \eta_2 \geq \dots \geq \eta_n$, then according to Lemma 2,

$$4/3 > \eta_1 > \dots > \eta_n > 0.$$

The eigenvalues of $I - QA^TA$ are

$$-\frac{1}{3} < 1 - \eta_1 \leq \dots \leq 1 - \eta_n < 1,$$

thus the spectral radius of $I - QA^TA$ is

$$\rho(I - QA^TA) = \max\{1 - \eta_n, |1 - \eta_1|\} \leq \max\{1 - \eta_n, \frac{1}{3}\} = \max\{\lambda_{\max}(I - QA^TA), \frac{1}{3}\}. \quad (50)$$

An interesting phenomenon occurs here. The spectral radius is either $1 - \eta_n$ or $|1 - \eta_1|$. In the latter case, $\rho(I - QA^TA) = |1 - \eta_1| \leq 1/3$, implying that $\|E(x^k) - x^*\| \leq \frac{1}{3^k} \|E(x^0) - x^*\|$, or equivalently, the relative error $\|E(x^k) - x^*\|/\|E(x^0) - x^*\|$ achieves ϵ in $\log 3 \log(1/\epsilon)$ epochs. We do not even need to compute η_1 since it will only affect the convergence speed when the speed is already very fast. From a theoretical perspective, the improvement from $\log 3$ to $\log(1/(1 - |1 - \eta_1|))$ is just an improvement in the constant. Therefore, it is reasonable to ignore η_1 and focus on the estimate of $1 - \eta_n$.

To estimate the maximum eigenvalue of $I - QA^TA$ (or equivalently, that of $I - AQA^T$), we first provide a useful identity that connects $I - AQA^T$ and projection matrices $P_i = I - A_i A_i^T$.

Claim 4.1 Suppose $A = [A_1, \dots, A_n]$ is a non-singular square matrix, and $A_i^T A_i = I, \forall i$. For a permutation $\sigma = (\sigma_1, \dots, \sigma_n) \in \Gamma$, L_σ is defined as in (21), and $Q_\sigma = L_\sigma^{-1}$. Denote $P_i = I - A_i A_i^T$, $i = 1, \dots, n$. Then we have

$$I - A Q_\sigma A^T = P_{\sigma_n} P_{\sigma_{n-1}} \dots P_{\sigma_1}, \quad (51a)$$

$$I - A Q A^T = \frac{1}{n!} \sum_{\sigma=(\sigma_1, \dots, \sigma_n) \in \Gamma} P_{\sigma_n} P_{\sigma_{n-1}} \dots P_{\sigma_1}. \quad (51b)$$

The proof of Claim 4.1 is given at the end of this subsection. Claim 4.1 states that $I - A Q A^T$ is exactly equal to $\frac{1}{n!} \sum_{\sigma=(\sigma_1, \dots, \sigma_n) \in \Gamma} P_{\sigma_n} P_{\sigma_{n-1}} \dots P_{\sigma_1}$, thus we only need to estimate the maximal eigenvalue of the latter expression. This is achieved by the following lemma (the proof is given in Section 7.3).

Lemma 3 (weak matrix AM-GM inequality) Suppose $P_i \in \mathbb{R}^{N \times N}, i = 1, \dots, n$ are projection matrices, then

$$\frac{1}{n!} \sum_{\sigma=(\sigma_1, \dots, \sigma_n) \in \Gamma} P_{\sigma_n} P_{\sigma_{n-1}} \dots P_{\sigma_1} \preceq \frac{1}{n} \sum_i P_i. \quad (52)$$

The above Lemma 3 and Claim 4.1 immediately lead to the following corollary.

Corollary 4.1 Suppose $A = [A_1, \dots, A_n]$ is a non-singular square matrix, and $A_i^T A_i = I, \forall i$. Suppose $P_i = I - A_i A_i^T, \forall i$. L_σ is defined as in (21), and $Q = E_\sigma(L_\sigma^{-1})$. Then

$$I - A Q A^T \preceq \frac{1}{n} \sum_i P_i. \quad (53)$$

Note that $\frac{1}{n} \sum_i P_i = \frac{1}{n}(nI - \sum_i A_i A_i^T) = I - \frac{1}{n} AA^T$, thus (53) implies

$$I - AQA^T \preceq I - \frac{1}{n} AA^T,$$

which implies

$$\lambda_{\max}(I - AQA^T) \leq 1 - \frac{1}{n} \lambda_{\min}(AA^T). \quad (54)$$

Substituting into (50), we get

$$\rho(I - QA^T A) \leq \max\{\lambda_{\max}(I - QA^T A), \frac{1}{3}\} \leq \max\{1 - \frac{1}{n} \lambda_{\min}(AA^T), 1/3\}.$$

Substituting this relation into (49), we obtain

$$\|E(x^{k+1}) - x^*\| \leq \max\{1 - \frac{1}{n} \lambda_{\min}(AA^T), \frac{1}{3}\} \|E(x^k) - x^*\|.$$

Q.E.D.

Remark: There is a coefficient $1/n$ in front of $\lambda_{\min}(AA^T)$ in (54), and this is why the complexity of RP-CD we establish is n times worse than the conjectured one in Table 1. If Conjecture 3.1 holds, then this factor of $1/n$ would be removed and the conjectured (expected) complexity of RP-CD in Table 1 would hold.

4.3.1 Proof of Claim 4.1

We prove (51a) by induction on n . Without loss of generality, we can assume $\sigma = (1, 2, \dots, n)$, then

$$L_\sigma = \begin{bmatrix} A_1^T A_1 & 0 & \dots & 0 \\ A_2^T A_1 & A_2^T A_2 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ A_n^T A_1 & A_n^T A_2 & \dots & A_n^T A_n \end{bmatrix}. \text{ In this case, (51a) becomes}$$

$$I - AL_\sigma^{-1} A^T = P_n P_{n-1} \dots P_1.$$

The expression obviously holds for $n = 1$. Suppose the expression holds for $n - 1$, i.e., for $\hat{A} = [A_1, \dots, A_{n-1}]$, we have

$$\hat{Z} \triangleq I - \hat{A} \hat{L}_{\hat{\sigma}}^{-1} \hat{A}^T = P_{n-1} \dots P_2 P_1, \quad (55)$$

where $\hat{\sigma} = (1, 2, \dots, n - 1)$ is a permutation of $n - 1$ elements and $\hat{L}_{\hat{\sigma}}$ is the counterpart of L_σ for $n - 1$ blocks defined as

$$\hat{L}_{\hat{\sigma}} = \begin{bmatrix} A_1^T A_1 & 0 & \dots & 0 \\ A_2^T A_1 & A_2^T A_2 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ A_{n-1}^T A_1 & A_{n-1}^T A_2 & \dots & A_{n-1}^T A_{n-1} \end{bmatrix}.$$

The two matrices L_σ and $\hat{L}_{\hat{\sigma}}$ are related by

$$L_\sigma = \begin{bmatrix} \hat{L}_{\hat{\sigma}} & 0 \\ A_n^T \hat{A} & I \end{bmatrix},$$

which implies

$$L_\sigma^{-1} = \begin{bmatrix} \hat{L}_\sigma^{-1} & 0 \\ -A_n^T \hat{A} \hat{L}_\sigma^{-1} & I \end{bmatrix}.$$

Therefore we have

$$\begin{aligned} AL_\sigma^{-1}A^T &= [\hat{A}, A_n] \begin{bmatrix} \hat{L}_\sigma^{-1} & 0 \\ -A_n^T \hat{A} \hat{L}_\sigma^{-1} & I \end{bmatrix} [\hat{A}, A_n]^T = \hat{A} \hat{L}_\sigma^{-1} \hat{A}^T - A_n A_n^T \hat{A} \hat{L}_\sigma^{-1} \hat{A}^T + A_n A_n^T \\ &= \hat{Z} - A_n A_n^T \hat{Z} + A_n A_n^T \\ &= I - (I - A_n A_n^T)(I - \hat{Z}) \\ &= I - P_n P_{n-1} \dots P_1, \end{aligned}$$

where in the last step we use the induction hypothesis (55). Thus we have proved (51a). Summing up (51a) for all possible permutations σ and divide by $n!$, we obtain (51b). \square

4.4 Proof of Proposition 2

According to (48), the (expected) update equation of RP-BCD is given by $E(x^{k+1}) - x^* = (I - QA^T A)(E(x^k) - x^*) = Z(E(x^k) - x^*)$, where $Z = I - QA^T A = I - E(L_\sigma^{-1} A^T A)$.

Consider a new coefficient matrix $\tilde{A} = [\tilde{A}_1, \dots, \tilde{A}_n]$ where $\tilde{A}_i = A_i(A_i^T A_i)^{-\frac{1}{2}}$. Clearly $\tilde{A}_i^T \tilde{A}_i = I_{d_i}$. Denote the corresponding matrices as $\tilde{L}_\sigma, \tilde{Z}$. Define $\Lambda \triangleq \text{Diag}((A_1^T A_1)^{\frac{1}{2}}, \dots, (A_n^T A_n)^{\frac{1}{2}}) = D^{1/2}$. When $\sigma = (1, 2, \dots, n)$, we have

$$L_\sigma = \begin{bmatrix} A_1^T A_1 & 0 & \dots & 0 \\ A_2^T A_1 & A_2^T A_2 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ A_n^T A_1 & A_n^T A_2 & \dots & A_n^T A_n \end{bmatrix}, \quad \tilde{L}_\sigma = \begin{bmatrix} \tilde{A}_1^T \tilde{A}_1 & 0 & \dots & 0 \\ \tilde{A}_2^T \tilde{A}_1 & \tilde{A}_2^T \tilde{A}_2 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ \tilde{A}_n^T \tilde{A}_1 & \tilde{A}_n^T \tilde{A}_2 & \dots & \tilde{A}_n^T \tilde{A}_n \end{bmatrix} = \Lambda^{-1} L_\sigma \Lambda^{-1}.$$

It is not hard to verify that the above relation $\tilde{L}_\sigma = \Lambda^{-1} L_\sigma \Lambda^{-1}$ is true for any σ . Similarly, we have $\tilde{A}^T \tilde{A} = \Lambda^{-1} A^T A \Lambda^{-1}$, thus

$$\tilde{L}_\sigma^{-1} \tilde{A}^T \tilde{A} = \Lambda L_\sigma^{-1} \Lambda \Lambda^{-1} A^T A \Lambda^{-1} = \Lambda L_\sigma^{-1} A^T A \Lambda^{-1}.$$

This implies

$$\tilde{Z} = E(I - \tilde{L}_\sigma^{-1} \tilde{A}^T \tilde{A}) = \Lambda(I - E(L_\sigma^{-1} A^T A)) \Lambda^{-1} = \Lambda Z \Lambda^{-1}.$$

Consider a sequence $\tilde{x}^k = \Lambda x^k$ and define $\tilde{x}^* = \Lambda x^*$. Then from the original update equation we have $\Lambda^{-1}(E(\tilde{x}^{k+1}) - \tilde{x}^*) = Z \Lambda^{-1}(E(\tilde{x}^k) - \tilde{x}^*)$, i.e.,

$$E(\tilde{x}^{k+1}) - \tilde{x}^* = \Lambda Z \Lambda^{-1}(E(\tilde{x}^k) - \tilde{x}^*) = \tilde{Z}(E(\tilde{x}^k) - \tilde{x}^*).$$

According to Theorem 3, we have

$$\|E(\tilde{x}^k) - \tilde{x}^*\| \leq \left\{ 1 - \frac{1}{n} \lambda_{\min}(\tilde{A}^T \tilde{A}), \frac{1}{3} \right\}^k \|\tilde{x}^0 - \tilde{x}^*\|. \quad (56)$$

Note that $\|E(\tilde{x}^k) - \tilde{x}^*\| = \|\Lambda(E(x^k) - x^*)\| = \sqrt{(E(x^k) - x^*)^T \Lambda^2 E(x^k) - x^*} = \|E(x^k) - x^*\|_D$, and $\tilde{A}^T \tilde{A} = \Lambda^{-1} A^T A \Lambda^{-1} = D^{-1/2} A^T \tilde{A} D^{-1/2}$. Substituting into (56), we obtain the desired inequality.

4.5 Proof of Theorem 4

Now we consider the expected convergence rate of RP-ADMM. The difference with the analysis for RP-BCD is that here we need to consider the distance between the eigenvalues of $I - AQA^T$ with $-1/3$ while for RP-BCD what matters is the distance between the eigenvalues of $I - AQA^T$ and -1 which is at least $2/3$ and thus can be ignored.

Claim 4.2 *Suppose the minimum and maximum eigenvalues of QA^TA are $0 < \tau_{\min} \leq \tau_{\max} < 4/3$. Then*

$$\rho(M) = \max \left\{ \sqrt{(1 - \tau_{\min})_+}, (\tau_{\max} - 1)_+ + \sqrt{\tau_{\max}(\tau_{\max} - 1)_+} \right\},$$

where $z_+ = \max\{z, 0\}$. Furthermore, we have

$$\rho(M) \leq \max \left\{ 1 - \frac{3}{4}(4 - 3\tau_{\max}), 1 - \frac{1}{2}\tau_{\min} \right\}. \quad (57)$$

The proof of Claim 4.2 is given in Section 7.1. The next lemma provides a universal estimate of the maximum eigenvalues of QA^TA .

Lemma 4 *The maximum eigenvalues of QA^TA is at most $\frac{4}{3} - \frac{4}{9} \frac{1}{n+1}$, i.e.,*

$$\tau_{\max} = \lambda_{\max}(QA^TA) \leq \frac{4}{3} - \frac{4}{9} \frac{1}{n+1}. \quad (58)$$

The proof of Lemma 4 is given in Section 7.2

According to (54), which is established in the proof of the expected convergence rate of RP-BCD, we have

$$\tau_{\min} = \lambda_{\min}(QA^TA) \geq \frac{1}{n} \lambda_{\min}(A^TA). \quad (59)$$

Substituting the bounds (58) and (59) into (57), we obtain

$$\rho(M) \leq \max \left\{ 1 - \frac{3}{4}(4 - 3\tau_{\max}), 1 - \frac{1}{2}\tau_{\min} \right\} = \max \left\{ 1 - \frac{1}{n+1}, 1 - \frac{1}{2n} \lambda_{\min}(A^TA) \right\}. \quad (60)$$

Since $\lambda_{\min}(A^TA) \leq 1$, $\frac{1}{2n} \leq \frac{1}{n+1}$, this bound can be simplified to

$$\rho(M) \leq 1 - \frac{1}{2n} \lambda_{\min}(A^TA). \quad \mathbf{Q.E.D.}$$

Remark: The eigenvalues of QA^TA lie in the region $(0, 4/3)$, which guarantees the expected convergence of RP-ADMM. To obtain the expected convergence rate, we need to know the distance of the spectrum to the two extremes 0 and $4/3$. We conjecture that the bound can be improved to $\rho(M) \leq 1 - \frac{1}{2} \lambda_{\min}(A^TA)$. This requires more effort than the conjecture of RP-CD: besides showing $\tau_{\min} \geq O(\lambda_{\min}(A^TA))$, we also need to show $\tau_{\max} \leq \frac{4}{3} - O(\lambda_{\min}(A^TA))$. This is left as future work.

5 Proof of Lemma 1

The proof of Lemma 1 relies on two simple techniques. The first technique, as elaborated in the Step 1 below, is to factorize M and rearrange the factors. The second technique, as elaborated in the Step 2 below, is to reduce the dimension by eliminating a variable from the eigenvalue equation.

Step 1: Factorizing M and rearranging the order of multiplication. The following observation is crucial: the matrix M defined by (43) can be factorized as

$$M = \begin{bmatrix} I & 0 \\ -A & I \end{bmatrix} \begin{bmatrix} QA^T & I \\ I & A \end{bmatrix} \begin{bmatrix} -A & I \\ I & 0 \end{bmatrix}.$$

Switching the order of the products by moving the first component to the last, we get a new matrix

$$M' \triangleq \begin{bmatrix} QA^T & I \\ I & A \end{bmatrix} \begin{bmatrix} -A & I \\ I & 0 \end{bmatrix} \begin{bmatrix} I & 0 \\ -A & I \end{bmatrix} = \begin{bmatrix} QA^T & I \\ I & A \end{bmatrix} \begin{bmatrix} -2A & I \\ I & 0 \end{bmatrix} = \begin{bmatrix} I - 2QA^T A & QA^T \\ -A & I \end{bmatrix}. \quad (61)$$

Note that $\text{eig}(XY) = \text{eig}(YX)$ for any two square matrices, thus

$$\text{eig}(M) = \text{eig}(M').$$

To prove (44), we only need to prove

$$\lambda \in \text{eig}(M') \iff \frac{(1-\lambda)^2}{1-2\lambda} \in \text{eig}(QA^T A). \quad (62)$$

Step 2: Relate the eigenvalues of M' to the eigenvalues of $QA^T A$, i.e. prove (62). This step is simple as we only use the definition of eigenvalues. However, note that, without Step 1, just applying the definition of eigenvalues of the original matrix M may not lead to a simple relationship as (62).

We first prove one direction of (62):

$$\lambda \in \text{eig}(M') \implies \frac{(1-\lambda)^2}{1-2\lambda} \in \text{eig}(QA^T A). \quad (63)$$

Suppose $v \in \mathbb{C}^{2N \times 1} \setminus \{0\}$ is an eigenvector of M' corresponding to the eigenvalue λ , i.e.

$$M'v = \lambda v.$$

Partition v as $v = \begin{bmatrix} v_1 \\ v_0 \end{bmatrix}$, where $v_1, v_0 \in \mathbb{C}^{N \times 1}$. Using the expression of M' in (61), we can write the above equation as

$$\begin{bmatrix} I - 2QA^T A & QA^T \\ -A & I \end{bmatrix} \begin{bmatrix} v_1 \\ v_0 \end{bmatrix} = \lambda \begin{bmatrix} v_1 \\ v_0 \end{bmatrix},$$

which implies

$$(I - 2QA^T A)v_1 + QA^T v_0 = \lambda v_1, \quad (64a)$$

$$-Av_1 + v_0 = \lambda v_0. \quad (64b)$$

We claim that (63) holds when $v_1 = 0$. In fact, in this case we must have $v_0 \neq 0$ (otherwise $v = 0$ cannot be an eigenvector). By (64b) we have $\lambda v_0 = v_0$, thus $\lambda = 1$. By (64a) we have $0 = QA^T v_0 = QA^T A(A^{-1}v_0)$, which implies $\frac{(1-\lambda)^2}{1-2\lambda} = 0 \in \text{eig}(QA^T A)$, therefore (63) holds in this case.

We then prove (63) for the case

$$v_1 \neq 0. \quad (65)$$

The equation (64b) implies $(1 - \lambda)v_0 = Av_1$. Multiplying both sides of (64a) by $(1 - \lambda)$ and invoking this equation, we get

$$(1 - \lambda)(I - 2QA^T A)v_1 + QA^T Av_1 = (1 - \lambda)\lambda v_1.$$

This relation can be simplified to

$$(1 - 2\lambda)QA^T Av_1 = (1 - \lambda)^2 v_1. \quad (66)$$

We must have $\lambda \neq \frac{1}{2}$; otherwise, the above relation implies $v_1 = 0$, which contradicts (65). Then (66) becomes

$$QA^T Av_1 = \frac{(1 - \lambda)^2}{1 - 2\lambda} v_1. \quad (67)$$

Therefore, $\frac{(1 - \lambda)^2}{1 - 2\lambda}$ is an eigenvalue of $QA^T A$, with the corresponding eigenvector $v_1 \neq 0$, which finishes the proof of (63).

The other direction ⁹

$$\lambda \in \text{eig}(M) \iff \frac{(1 - \lambda)^2}{1 - 2\lambda} \in \text{eig}(QA^T A) \quad (68)$$

is easy to prove. Suppose $\frac{(1 - \lambda)^2}{1 - 2\lambda} \in \text{eig}(QA^T A)$. We consider two cases.

Case 1: $\frac{(1 - \lambda)^2}{1 - 2\lambda} = 0$. In this case $\lambda = 1$. Since $0 = \frac{(1 - \lambda)^2}{1 - 2\lambda} \in \text{eig}(QA^T A)$, there exists $v_0 \in \mathbb{C}^N \setminus \{0\}$ such that $QA^T Av_0 = 0$ and Let $v_1 = (0, \dots, 0)^T \in \mathbb{C}^{N \times 1}$, then v_0, v_1 and $\lambda = 1$ satisfy (64). Thus $v = \begin{bmatrix} v_1 \\ v_0 \end{bmatrix} \in \mathbb{C}^{2N} \setminus \{0\}$ satisfies $Mv = \lambda v$, which implies $\lambda = 1 \in \text{eig}(M)$.

Case 2: $\frac{(1 - \lambda)^2}{1 - 2\lambda} \neq 0$, then $\lambda \neq 1$. Let v_1 be the eigenvector corresponding to $\frac{(1 - \lambda)^2}{1 - 2\lambda}$ (i.e. pick v_1 that satisfies (67)), and define $v_0 = v_1/(1 - \lambda)$. It is easy to verify that $v = \begin{bmatrix} v_1 \\ v_0 \end{bmatrix}$ satisfies $Mv = \lambda v$, which implies $\lambda \in \text{eig}(M)$.

Step 3: When Q is symmetric, prove (45) by simple algebraic computation.

Since Q is symmetric, we know that $\text{eig}(QA^T A) = \text{eig}(AQ A^T) \subseteq \mathbb{R}$. Suppose $\tau \in \mathbb{R}$ is an eigenvalue of $QA^T A$, then any λ satisfying $\frac{(1 - \lambda)^2}{1 - 2\lambda} = \tau$ is an eigenvalue of M . This relation can be rewritten as $\lambda^2 + 2(\tau - 1)\lambda + (1 - \tau) = 0$, which, as a real-coefficient quadratic equation in λ , has two roots

$$\lambda_1 = 1 - \tau + \sqrt{\tau(\tau - 1)}, \quad \lambda_2 = 1 - \tau - \sqrt{\tau(\tau - 1)}. \quad (69)$$

Note that when $\tau(\tau - 1) < 0$, the expression $\sqrt{\tau(\tau - 1)}$ denotes a complex number $i\sqrt{\tau(1 - \tau)}$, where i is the imaginary unit. To prove (45), we only need to prove

$$\max\{|\lambda_1|, |\lambda_2|\} < 1 \iff 0 < \tau < \frac{4}{3}. \quad (70)$$

Consider three cases.

Case 1: $\tau < 0$. Then $\tau(\tau - 1) = |\tau|(|\tau| + 1) > 0$. In this case, $\lambda_1 = 1 + |\tau| + \sqrt{|\tau|(|\tau| + 1)} > 1$.

⁹For the purpose of proving Theorem 2, we do not need to prove this direction. Here we present the proof since it is quite straightforward and makes the result more comprehensive.

Case 2: $0 < \tau < 1$. Then $\tau(\tau - 1) < 0$, and (69) can be rewritten as

$$\lambda_{1,2} = 1 - \tau \pm i\sqrt{\tau(1 - \tau)},$$

which implies $|\lambda_1| = |\lambda_2| = \sqrt{(1 - \tau)^2 + \tau(1 - \tau)} = \sqrt{1 - \tau} < 1$.

Case 3: $\tau > 1$. Then $\tau(\tau - 1) > 0$. According to (69), it is easy to verify $\lambda_1 > 0 > \lambda_2$ and

$$|\lambda_2| = \tau - 1 + \sqrt{\tau(\tau - 1)} > 1 - \tau + \sqrt{\tau(\tau - 1)} = |\lambda_1|.$$

Then we have

$$\max\{|\lambda_1|, |\lambda_2|\} < 1 \iff |\lambda_2| = \tau - 1 + \sqrt{\tau(\tau - 1)} < 1 \iff 1 < \tau < \frac{4}{3}.$$

Combining the conclusions of the three cases immediately leads to (70).

6 Proof of Lemma 2

This section is devoted to the proof of Lemma 2. We first give a proof overview in Section 6.1. The formal proof of Lemma 2 is given in Section 6.2. The proofs of the technical results involved in the proof are given in the subsequent subsections.

Without loss of generality, we can assume

$$A_i^T A_i = I_{d_i \times d_i}, \quad i = 1, \dots, n.$$

To show this, let us write M_σ, M as $M_\sigma(A_1, \dots, A_n)$ and $M(A_1, \dots, A_n)$ respectively, i.e. functions of the coefficient matrix (A_1, \dots, A_n) . Define $\tilde{A}_i = A_i(A_i^T A_i)^{-\frac{1}{2}}$ and

$$D \triangleq \text{Diag}((A_1^T A_1)^{-\frac{1}{2}}, \dots, (A_n^T A_n)^{-\frac{1}{2}}, I_{N \times N}).$$

It is easy to verify that $M_\sigma(A_1, \dots, A_n) = D^{-1} M_\sigma(\tilde{A}_1, \dots, \tilde{A}_n) D$, which implies

$$M(A_1, \dots, A_n) = D^{-1} M(\tilde{A}_1, \dots, \tilde{A}_n) D.$$

Thus $\rho(M(A_1, \dots, A_n)) = \rho(M(\tilde{A}_1, \dots, \tilde{A}_n))$. In other words, normalizing A_i to \tilde{A}_i , which satisfies $\tilde{A}_i^T \tilde{A}_i = I_{d_i \times d_i}$, does not change the spectral radius of M .

6.1 Proof Overview

In the proof overview, we discuss a few issues one may encounter when proving the result, and how we resolve these issues.

The simulations show that $\|QA^T A\| < \frac{4}{3} \ll \|Q\| \|A^T A\|$, thus we cannot relax $\|QA^T A\|$ to the product of $\|Q\|$ and $\|A^T A\|$, and have to treat $QA^T A$ as a single subject. However, each entry of $QA^T A$ is a complicated function (in fact, a high order polynomial) of the entries of $A^T A$. In other words, Q is like a black box. To open the “black box”, we use a simple expression of $Z = I - AQA^T$ proved in Claim

4.1, i.e., $Z = E_\sigma(P_{\sigma_1} \dots, P_{\sigma_n})$, where $P_i = I - A_i A_i^T$ is directly related to A_i . The problem becomes how to connect the eigenvalues of $E_\sigma(P_{\sigma_1} \dots, P_{\sigma_n})$ with those of $AA^T = \sum_i A_i A_i^T = n - \sum_i P_i$.

Although this is a clear linear algebra problem, it is not easy to obtain a lower bound of $E_\sigma(P_{\sigma_1} \dots, P_{\sigma_n})$. In fact, even though we know the eigenvalues of $Z = E_\sigma(P_{\sigma_1} \dots, P_{\sigma_n})$ are lower bounded by -1 because RP-CD converges, it is not clear how to prove this lower bound directly from a linear algebra perspective.

In our solution, we apply two tricks. The first trick is to view $E_\sigma(P_{\sigma_1} \dots, P_{\sigma_n})$ as an induction formula that connects it and its lower dimensional analogs. This is based on a simple observation that any permutation $(\sigma_1 \sigma_2 \dots \sigma_n)$ can be written as the concatenation of $(\sigma_1 \sigma_2 \dots \sigma_{n-1})$ and σ_n , thus the expression of $Z = E_\sigma(P_{\sigma_1} \dots, P_{\sigma_n})$ can be decomposed accordingly. We then reduce the problem to bounding the eigenvalues of a Jordan product $P_n \hat{Z} + \hat{Z} P_n$, where P_n is a projection matrix and \hat{Z} is the lower dimensional analog of Z . The second trick is to apply a formula on the eigenvalues of Jordan product developed by Strang in 1962 [44]. Somewhat surprisingly, his formula exactly leads to the desired lower bound of $-1/3$.

6.2 Proof of Lemma 2

The proof can be divided into three steps: first provide an alternative expression of AQA^T , then prove an induction formula, and finally apply Strang's formula to perform mathematical induction. This subsection contains the major part of the proof, and the intermediate technical results will be proved in later subsections.

Step 0: Expression of $I - AQA^T$. As proved in Claim 4.1, we have a simple expression of the update matrix $I - AQA^T$

$$I - AQA^T = \frac{1}{n!} \sum_{\sigma=(\sigma_1, \dots, \sigma_n) \in \Gamma} P_{\sigma_n} P_{\sigma_{n-1}} \dots P_{\sigma_1}.$$

Step 1: Induction formula.

For any $k \in [n]$, define

$$\Gamma_k \triangleq \{\sigma' \mid \sigma' \text{ is a permutation of } [n] \setminus \{k\}\}. \quad (71)$$

For any $\sigma' \in \Gamma_k$, we define $L_{\sigma'} \in \mathbb{R}^{(N-d_k) \times (N-d_k)}$ as a $(n-1) \times (n-1)$ block-partitioned matrix, with the $(\sigma'(i), \sigma'(j))$ -th block being

$$L_{\sigma'}[\sigma'(i), \sigma'(j)] \triangleq \begin{cases} A_{\sigma'(i)}^T A_{\sigma'(j)} & i \geq j, \\ 0 & i < j, \end{cases} \quad (72)$$

We then define $\hat{Q}_k \in \mathbb{R}^{(N-d_k) \times (N-d_k)}$ by

$$\hat{Q}_k \triangleq \frac{1}{|\Gamma_k|} \sum_{\sigma' \in \Gamma_k} L_{\sigma'}^{-1}, \quad k = 1, \dots, n. \quad (73)$$

Define W_k as the k -th block-column of $A^T A$ excluding the block $A_k^T A_k$, i.e.

$$W_k = [A_k^T A_1, \dots, A_k^T A_{k-1}, A_k^T A_{k+1}, \dots, A_k^T A_n]^T, \quad \forall k \in [n]. \quad (74)$$

Based on the expression of $I - AQA^T$ presented before, we build a connection between the update matrix $I - AQA^T$ and its lower dimensional analogs. The proof of Proposition 3 is given in Section 6.3.

Proposition 3 *Define*

$$Z = I - AQA^T, \quad \hat{Z}_k = I - \hat{A}_k \hat{Q}_k \hat{A}_k^T,$$

where Q is defined as in (39), $\hat{A}_k = [A_1, \dots, A_{k-1}, A_{k+1}, \dots, A_n]$, and \hat{Q}_k is defined in (73), and $P_k = I - A_k A_k^T$. Then we have

$$Z = \frac{1}{2n} \sum_{k=1}^n (P_k \hat{Z}_k + \hat{Z}_k P_k). \quad (75)$$

Step 2: Applying Strang's result on Jordan product to perform mathematical induction.

It is obvious that the product of two symmetric matrices is not necessarily symmetric, so it is common to encounter the symmetrized product $XY + YX$, which is called Jordan product of two matrices X and Y . Our induction formula basically states that Z is the average of the Jordan product of the lower dimensional analog and P_k .

The eigenvalues of the Jordan product of two matrices have been studied before. The following result is proved in Strang [44].

Lemma 5 ([44, Theorem 1]; eigenvalues of Jordan product) *Suppose two symmetric positive-semidefinite matrices X and Y satisfy*

$$\alpha_1 I \preceq X \preceq \alpha_n I, \quad \beta_1 I \preceq Y \preceq \beta_n I,$$

then the maximal (resp. minimal) eigenvalue of the Jordan product $XY + YX$ are the largest (resp. smallest) of the set

$$\left\{ 2\alpha_i \beta_j, i, j \in \{1, n\}, \frac{16\alpha_1 \alpha_n \beta_1 \beta_n - (\beta_1 - \beta_n)^2 (\alpha_1 - \alpha_n)^2}{4(\alpha_1 + \alpha_n)(\beta_1 + \beta_n)} \right\}. \quad (76)$$

Let us come back to the proof of Lemma 2. We use mathematical induction to prove Lemma 2. For the basis of the induction ($n = 1$), Lemma 2 holds since $QA^T A = I_{d_1 \times d_1}$. Assume Lemma 2 holds for $n - 1$, we will prove Lemma 2 for n .

Consider one term of (75) $P_k \hat{Z}_k + \hat{Z}_k P_k$. Note that $P_k = I - A_k A_k^T$ is a projection matrix, since we have assumed $A_k^T A_k = I$. Combining with the induction hypothesis, we have

$$0 \preceq P_k \preceq I, \quad -\frac{1}{3}I \prec \hat{Z}_k \prec I.$$

Let $\alpha_1 = 0, \alpha_n = 1, \beta_1 = -1/3, \beta_n = 1$, then the set (76) becomes (keep the repeated values)

$$\{0, 0, -2/3, 2, -2/3\}.$$

Then by Lemma 5 we have

$$-\frac{1}{3}I \preceq \frac{1}{2}(P_k \hat{Z}_k + \hat{Z}_k P_k) \preceq I.$$

Note that since by the induction hypothesis the eigenvalues of \hat{Z}_k cannot achieve the extreme values of region $(-1/3, 1)$, the eigenvalues of $\frac{1}{2}(P_k \hat{Z}_k + \hat{Z}_k P_k)$ also cannot¹⁰. So we have

$$-\frac{1}{3}I \prec \frac{1}{2}(P_k \hat{Z}_k + \hat{Z}_k P_k) \prec I.$$

Thus according to (75) we have

$$-\frac{1}{3}I \prec Z \prec I.$$

This finishes the induction step. **Q.E.D.**

Remark: Where does the magical number $-1/3$ come from? It is actually the strange and complicated term $\frac{16\alpha_1\alpha_n\beta_1\beta_n-(\beta_1-\beta_n)^2(\alpha_1-\alpha_n^2)}{4(\alpha_1+\alpha_n)(\beta_1+\beta_n)}$ in Strang's result (76), which occurs due to the special structure of the Jordan product.

6.3 Proof of Proposition 3 (the induction formula)

It is easy to build an induction formula from the expression (51b). For example, when $n = 3$, the matrix $\sum_{\sigma} P_{\sigma_1} P_{\sigma_2} P_{\sigma_3}$ can be decomposed as the sum of $P_1(P_2P_3 + P_3P_2) + (P_2P_3 + P_3P_2)P_1$ and two other similar terms (changing the outside part P_1 to P_2, P_3 and the inside part $P_2P_3 + P_3P_2$ correspondingly). The inside part $P_2P_3 + P_3P_2$ only involves two matrices, thus is a lower-dimensional analog. To make this even easier to see, denote $X = P_1, Y = P_2, Z = P_3$, then

$$\begin{aligned} 2 \sum_{\text{permute } X, Y, Z} XYZ &= [X(YZ + ZY) + (YZ + ZY)X] + [Y(XZ + ZX) + (XZ + ZX)Y] \\ &\quad + [Z(XY + YX) + (XY + YX)Z]. \end{aligned}$$

A rigorous argument based on the above intuition is given as follows. Applying the formula (51b) to the matrix $P_1, \dots, P_{k-1}, P_{k+1}, \dots, P_n$, and by the definition $\hat{A}_k = [A_1, \dots, A_{k-1}, A_{k+1}, \dots, A_n]$ and the definition of \hat{Q}_k in (73), we have

$$I - \hat{A}_k \hat{Q}_k \hat{A}_k = \frac{1}{(n-1)!} \sum_{\sigma=(\sigma_1, \dots, \sigma_{n-1}) \in \Gamma_k} P_{\sigma_{n-1}} P_{\sigma_{n-1}} \dots P_{\sigma_1}.$$

We then have

$$\begin{aligned} 2(I - AQA^T) &= \frac{2}{n!} \sum_{\sigma=(\sigma_1, \dots, \sigma_n) \in \Gamma} P_{\sigma_n} P_{\sigma_{n-1}} \dots P_{\sigma_1} \\ &= \frac{1}{n} \frac{1}{(n-1)!} \sum_{k=1}^n \sum_{\sigma=(\sigma_1, \dots, \sigma_{n-1}) \in \Gamma_k} (P_k P_{\sigma_{n-1}} P_{\sigma_{n-1}} \dots P_{\sigma_1} + P_{\sigma_{n-1}} P_{\sigma_{n-1}} \dots P_{\sigma_1} P_k) \\ &= \frac{1}{n} \sum_{k=1}^n (P_k (I - \hat{A}_k \hat{Q}_k \hat{A}_k^T) + (I - \hat{A}_k \hat{Q}_k \hat{A}_k) P_k), \end{aligned}$$

which is the desired formula.

¹⁰A more detailed argument is as follows. Since $-1/3 \preceq \hat{Z}_k$, we can let $\beta_1 = -1/3 + \epsilon$ for a sufficiently small positive number ϵ , while keeping $\alpha_1 = 0, \alpha_n = 1, \beta_n = 1$. The set (76) now becomes $\{0, 0, -2/3 + 2\epsilon, 2, -\frac{(4/3-\epsilon)^2}{4(2/3+\epsilon)}\}$. Both $-2/3 + 2\epsilon$ and $-\frac{(4/3-\epsilon)^2}{4(2/3+\epsilon)}$ are strictly larger than $-1/3$, thus the extreme value $-1/3$ cannot be achieved. By a similar argument the other extreme value 1 also cannot be achieved.

6.4 Proof of Proposition 1

We provide the proof of the expected convergence of BR-ADMM here, as this proof is a slightly smaller subset of the proof of Theorem 1. We will just describe the necessary modifications.

We only need to prove a similar version of Theorem 2, i.e., the spectral radius of the expected update matrix of BR-ADMM is less than 1. Throughout the proof, we need to change the matrix $Q = \frac{1}{|\Gamma|} \sum_{\sigma \in \Gamma} Q_\sigma$ to another one defined as

$$Q^{\text{BR}} \triangleq \frac{1}{|\Gamma^{\text{BR}}|} \sum_{\sigma \in \Gamma^{\text{BR}}} Q_\sigma, \quad (77)$$

where Γ^{BR} denotes the set of all possible permutations according to the Bernoulli randomization rule. It is easy to see that $|\Gamma^{\text{BR}}| = 2^n$. Other matrices such as M should be changed accordingly.

The proof of Theorem 2 mainly consists of Lemma 1 and Lemma 2. Since Lemma 1 has nothing to do with the specific expression of Q , so we only need to prove Lemma 2 for BR-ADMM, i.e., the matrix $AQ^{\text{BR}}A^T$ has all eigenvalues in the region $(0, 4/3)$. Following the proof of Lemma 2, we divide the proof into three steps.

Step 0: Expression of $Z^{\text{BR}} \triangleq I - AQ^{\text{BR}}A^T$. In Claim 4.1, we have prove the expression (51a) that $I - AQ_\sigma A^T = P_{\sigma_n} P_{\sigma_{n-1}} \dots P_{\sigma_1}$ for any permutation σ , which implies

$$Z^{\text{BR}} = I - AQ^{\text{BR}}A^T \stackrel{(77)}{=} \frac{1}{2^n} \sum_{\sigma \in \Gamma^{\text{BR}}} P_{\sigma_n} P_{\sigma_{n-1}} \dots P_{\sigma_1}.$$

Step 1: Induction formula. Notice that a characteristic of the Bernoulli randomization rule is: the first block is either updated first or updated last. For instance, when $n = 4$, $(1, 3, 4, 2)$ is a feasible permutation in Γ^{BR} and $(3, 4, 2, 1)$ is also a feasible permutation, but $(3, 1, 4, 2)$ is not feasible. After removing the first block, the rest $n - 1$ blocks form a permutation in $\hat{\Gamma}_{\text{BR}}$, where $\hat{\Gamma}_{\text{BR}}$ is the set of all permutation of $2, 3, \dots, n$ according to the Bernoulli randomization rule. In other words, we have $\Gamma^{\text{BR}} = \{(1, \hat{\sigma}), (\hat{\sigma}, 1), \text{ where } \hat{\sigma} \in \hat{\Gamma}_{\text{BR}}\}$. Thus we have an induction formula

$$Z^{\text{BR}} = \frac{1}{2^n} \sum_{\sigma=(\sigma_1, \dots, \sigma_{n-1}) \in \hat{\Gamma}_{\text{BR}}} (P_1 P_{\sigma_{n-1}} \dots P_{\sigma_1} + P_{\sigma_{n-1}} \dots P_{\sigma_1} P_1) = \frac{1}{2} (P_1 \hat{Z}^{\text{BR}} + \hat{Z}^{\text{BR}} P_1), \quad (78)$$

where \hat{Z}^{BR} is the lower dimensional analog of Z^{BR} for the rest $n - 1$ blocks (after removing the first block).

Step 2: Applying mathematical induction. This step is almost the same as Step 2 of the proof of Lemma 2. More specifically, combining the induction hypothesis that $\text{eig}(\hat{Z}^{\text{BR}}) \in (-1/3, 1)$, Strang's result Lemma 5 and (78), we obtain the desired result $\text{eig}(Z^{\text{BR}}) \in (-1/3, 1)$. This finishes the proof.

7 Proof of Technical Results for Expected Convergence Rates

7.1 Proof of Claim 4.2

Suppose all the distinct eigenvalues of $I - QA^T A$ are $0 < \tau_{N'} < \dots < \tau_1 < 4/3$, where $1 \leq N' \leq N$. Denote $\tau_{\min} = \tau_{N'}, \tau_{\max} = \tau_1$. According to Lemma 1, the expected update matrix of RP-ADMM M

has $2N'$ distinct eigenvalues $\lambda_{k,1}, \lambda_{k,2}$ given by

$$\lambda_{k,1} = 1 - \tau_k + \sqrt{\tau_k(\tau_k - 1)}, \quad \lambda_{k,2} = 1 - \tau_k - \sqrt{\tau_k(\tau_k - 1)}, \quad k = 1, \dots, N'.$$

Suppose the integer $m \in [1, N' + 1]$ satisfies $\tau_m \leq 1 < \tau_{m-1}$. When $m = 1$, every $\tau_k \leq 1$; when $m = N' + 1$, every $\tau_k > 1$.

For $N' \geq k \geq m$, i.e., $\tau_k \leq 1$, we have $\tau_k(\tau_k - 1) \leq 0$, thus the two corresponding eigenvalues of M are

$$\lambda_{k,1} = 1 - \tau \pm i\sqrt{\tau(1 - \tau)}, \quad \lambda_{k,2} = 1 - \tau \pm i\sqrt{\tau(1 - \tau)},$$

which implies $|\lambda_{k,1}| = |\lambda_{k,2}| = \sqrt{(1 - \tau_k)^2 + \tau_k(1 - \tau_k)} = \sqrt{1 - \tau_k}$. Thus $\rho_1 = \max_{N' \geq k \geq m} \{|\lambda_{k,1}|, |\lambda_{k,2}|\} = \sqrt{1 - \tau_{N'}} = \sqrt{1 - \tau_{\min}}$ if such k exists; when such k does not exist, i.e., $\tau_k > 1 \forall k$ we denote $\rho_1 = 0$ which equals $\sqrt{(1 - \tau_{\min})_+}$. In summary, we have $\rho_1 = \sqrt{(1 - \tau_{\min})_+}$.

For $m - 1 \geq k \geq 1$, i.e., $\tau_k > 1$, we have $\tau_k(\tau_k - 1) > 0$. It is easy to verify $\lambda_{k,1} > 0 > \lambda_{k,2}$ and

$$|\lambda_{k,2}| = \tau_k - 1 + \sqrt{\tau_k(\tau_k - 1)} > 1 - \tau_k + \sqrt{\tau_k(\tau_k - 1)} = |\lambda_{k,1}|.$$

Denote $\rho_2 = \max_{m-1 \geq k \geq 1} \{|\lambda_{k,1}|, |\lambda_{k,2}|\}$, then $\rho_2 = \max_{m-1 \geq k \geq 1} \{|\lambda_{k,2}|\} = \max_{m-1 \geq k \geq 1} \{\tau_k - 1 + \sqrt{\tau_k(\tau_k - 1)}\} = \tau_{\max} - 1 + \sqrt{\tau_{\max}(\tau_{\max} - 1)}$ if such k exists; when such k does not exist, i.e., $\tau_k \leq 1 \forall k$, we denote $\rho_2 = 0$ which equals $(\tau_{\max} - 1)_+ + \sqrt{\tau_{\max}((\tau_{\max} - 1)_+)}$.

Combining the two scenarios, we have $\rho(M) = \max_{N' \geq k \geq 1} \{|\lambda_{k,1}|, |\lambda_{k,2}|\} = \max\{\rho_1, \rho_2\} = \max\{\sqrt{(1 - \tau_{\min})_+}, (\tau_{\max} - 1)_+ + \sqrt{\tau_{\max}((\tau_{\max} - 1)_+)}\}$.

Next, we prove

$$\begin{aligned} (\tau_{\max} - 1)_+ + \sqrt{\tau_{\max}(\tau_{\max} - 1)_+} &\leq \max \left\{ 1 - \frac{3}{4}(4 - 3\tau_{\max}), 0 \right\}, \\ \sqrt{(1 - \tau_{\min})_+} &\leq 1 - \frac{1}{2}\tau_{\min}. \end{aligned} \tag{79}$$

In fact, when $4/3 \geq \tau \geq 1$, we have $1 - (\tau - 1 + \sqrt{\tau(\tau - 1)}) = 2 - \tau - \sqrt{\tau(\tau - 1)} = \frac{(2 - \tau)^2 - \tau(\tau - 1)}{2 - \tau + \sqrt{\tau(\tau - 1)}} = \frac{3 - 4\tau}{2 - \tau + \sqrt{\tau(\tau - 1)}} \geq \frac{3}{4}(3 - 4\tau)$, thus $\tau - 1 + \sqrt{\tau(\tau - 1)} \leq 1 - \frac{3}{4}(3 - 4\tau)$. When $\tau < 1$, clearly $\tau - 1 + \sqrt{\tau(\tau - 1)} = 0$. Thus $(\tau - 1)_+ + \sqrt{\tau(\tau - 1)_+} \leq \max\{0, 1 - \frac{3}{4}(4 - 3\tau)\}$. For the second relation, if $0 \leq \tau < 1$ then $\sqrt{1 - \tau} = 1 - \frac{\tau}{1 + \sqrt{1 - \tau}} \leq 1 - \frac{\tau}{2}$; if $1 \leq \tau \leq 4/3$ then $\sqrt{(1 - \tau)_+} = 0 < 1 - \frac{1}{2}\tau$. Thus $\sqrt{(1 - \tau)_+} \leq 1 - \frac{1}{2}\tau$ holds for any $\tau \in [0, 4/3]$.

Substituting (79) into the expression of $\rho(M)$, we obtain the desired inequality

$$\rho(M) \leq \max \left\{ 1 - \frac{3}{4}(4 - 3\tau_{\max}), 1 - \frac{1}{2}\tau_{\min} \right\}.$$

7.2 Proof of Lemma 4

This is one of the two main lemmas of proving the expected convergence rate of RP-ADMM (the other is the expected convergence rate of RP-CD).

The proof outline of Lemma 4 and the main techniques are described below. The previous proof for the expected convergence of RP-ADMM in Section 6 is not strong enough to prove a convergence rate.

We have to obtain a more refined estimate of the spectral radius of AQA^T . To do so, we transform the induction formula in Proposition 3 to a “dual” form: instead of AQA^T , we consider a similar matrix $QA^T A$. We then apply the two simple techniques used in the proof of Lemma 1: factorize and rearrange, and reduce the dimension by eliminating a variable from the eigenvalue equation. We obtain a somewhat complicated inequality relating $\lambda_{\max}(QA^T A)$ and its lower-dimensional analog $\lambda_{\max}(\hat{Q}\hat{A}^T\hat{A})$. Finally, we perform a detailed analysis of the inequality to prove the desired bound.

7.2.1 Step 1: Mathematical Induction and Induction Formula

Define a sequence $\{\alpha_k\}_{k=1}^{\infty}$ such that

$$\alpha_1 = 1/3, \quad \alpha_{k+1} = h(\alpha_k) \triangleq \frac{\alpha_k}{8} \frac{16 - 3\alpha_k}{2 + 3\alpha_k}. \quad (80)$$

It is easy to verify that $0 < \alpha_{k+1} < \alpha_k \leq 1/3$ for all k . The following claim provides a bound of α_k (the proof will be given in Section 7.2.4).

Claim 7.1 *Suppose the sequence $\{\alpha_k\}_{k=1}^{\infty}$ satisfies (80), then $\alpha_k \geq \frac{4}{9(k+1)}, \forall k \geq 1$.*

According to this claim, to prove the desired result $\lambda_{\max}(AQA^T) \leq \frac{4}{3} - \frac{4}{9(k+1)}$, we only need to prove the following result:

$$\text{eig}(AQA^T) \subseteq (0, \frac{4}{3} - \alpha_n]. \quad (81)$$

We prove this result by mathematical induction. When $n = 1$, since $A^T A = A_1^T A_1 = I$, we have $\lambda_{\min}(AQA^T) = \lambda_{\max}(AQA^T) = 1 = \frac{4}{3} - \alpha_1$.

Suppose the result holds for $n - 1$, i.e., for a problem with $n - 1$ blocks, the eigenvalues of the corresponding matrix $\hat{A}\hat{Q}\hat{A}^T$ lie in the region $(0, \frac{4}{3} - \alpha_{n-1})$.

Next, we build the induction formula, which is the dual form of the one we derived before. According to (75), we have

$$2(I - AQA^T) = \frac{1}{n} \sum_{k=1}^n (P_k(I - \hat{A}_k \hat{Q}_k \hat{A}_k) + (I - \hat{A}_k \hat{Q}_k \hat{A}_k)P_k),$$

which can be rewritten as

$$AQA^T = \frac{1}{n} \sum_{k=1}^n \left[I - \frac{1}{2} P_k (I - \hat{A}_k \hat{Q}_k \hat{A}_k) - \frac{1}{2} (I - \hat{A}_k \hat{Q}_k \hat{A}_k) P_k \right] \quad (82)$$

Note that

$$\begin{aligned} I - P_k(I - \hat{A}_k \hat{Q}_k \hat{A}_k) &= I - (I - A_k A_k^T)(I - \hat{A}_k \hat{Q}_k \hat{A}_k^T) \\ &= A_k A_k^T + \hat{A}_k \hat{Q}_k \hat{A}_k^T - A_k A_k^T \hat{A}_k \hat{Q}_k \hat{A}_k^T \\ &= [\hat{A}_k, A_k] \begin{bmatrix} \hat{Q}_k & 0 \\ -A_k^T \hat{A}_k \hat{Q}_k & I \end{bmatrix} [\hat{A}_k, A_k]^T \end{aligned}$$

Thus the symmetrized version

$$I - \frac{1}{2}P_k(I - \hat{A}_k\hat{Q}_k\hat{A}_k) - \frac{1}{2}(I - \hat{A}_k\hat{Q}_k\hat{A}_k)P_k \quad (83)$$

$$= [\hat{A}_k, A_k] \begin{bmatrix} \hat{Q}_k & -\frac{1}{2}\hat{Q}_k^T\hat{A}_k^TA_k \\ -\frac{1}{2}A_k^T\hat{A}_k\hat{Q}_k & I \end{bmatrix} [\hat{A}_k, A_k]^T \quad (84)$$

$$= \bar{A}_k Q_k \bar{A}_k^T, \quad (85)$$

where in the last step we use the definitions

$$\bar{A}_k \triangleq [\hat{A}_k, A_k], \quad Q_k \triangleq \begin{bmatrix} \hat{Q}_k & -\frac{1}{2}\hat{Q}_k W_k \\ -\frac{1}{2}W_k^T \hat{Q}_k & I_{d_k \times d_k} \end{bmatrix} \quad (86)$$

Sum up (85) for $k = 1, \dots, n$ and applying (82), we have

$$AQ A^T = \frac{1}{n} \sum_{k=1}^n \bar{A}_k Q_k \bar{A}_k^T. \quad (87)$$

Consequently,

$$\frac{1}{n} \sum_{k=1}^n \lambda_{\min}(\bar{A}_k Q_k \bar{A}_k^T) \leq \lambda_{\min}(AQ A^T) \leq \lambda_{\max}(AQ A^T) \leq \frac{1}{n} \sum_{k=1}^n \lambda_{\max}(\bar{A}_k Q_k \bar{A}_k^T). \quad (88)$$

To prove $\text{eig}(AQ A^T) \subseteq (0, \frac{4}{3} - \alpha_n]$, we only need to prove for any $k = 1, \dots, n$,

$$\text{eig}(\bar{A}_k Q_k \bar{A}_k^T) \subseteq (0, \frac{4}{3} - \alpha_n). \quad (89)$$

Note that \hat{Q}_k only depends on the entries of $\hat{A}_k^T \hat{A}_k \in \mathbb{R}^{(N-d_k) \times (N-d_k)}$ which has $(n-1) \times (n-1)$ blocks, thus by the induction hypothesis, we have

$$\text{eig}(\hat{Q}_k \hat{A}_k^T \hat{A}_k) \subseteq (0, \frac{4}{3} - \alpha_{n-1}). \quad (90)$$

Proposition 4 Suppose $A = [\hat{A}_n, A_n] \in \mathbb{R}^{N \times N}$ is a non-singular matrix, where $\hat{A}_n \in \mathbb{R}^{N \times (N-d_n)}$, and $A_n \in \mathbb{R}^{N \times d_n}$ satisfies $A_n^T A_n = I_{d_n \times d_n}$. Suppose $\hat{Q}_n \in \mathbb{R}^{(N-d_n) \times (N-d_n)}$ is symmetric, satisfying

$$\text{eig}(A \hat{Q}_n A^T) \subseteq (0, \frac{4}{3} - \alpha_{n-1}), \quad (91)$$

where $\{\alpha_k\}$ is defined in (80). Define

$$W_n \triangleq \hat{A}_n^T A_n \in \mathbb{R}^{(N-d_n) \times d_n}, \quad Q_n \triangleq \begin{bmatrix} \hat{Q}_n & -\frac{1}{2}\hat{Q}_n W_n \\ -\frac{1}{2}W_n^T \hat{Q}_n & I_{d_n \times d_n} \end{bmatrix}. \quad (92)$$

Then $\text{eig}(AQ_n A^T) \subseteq (0, \frac{4}{3} - \alpha_n]$.

The proof of Proposition 4 will be divided into two parts, and given in Section 7.2.2 and Section 7.2.3.

We claim that (89) follows from the induction hypothesis (90) and the expressions of \bar{A}_k and Q_k in (86). In fact, the above proposition directly proves (89) for $k = n$. If we replace $A, \hat{A}_n, A_n, \hat{Q}_n, Q_n$ by $\bar{A}_k, \hat{A}_k, A_k, \hat{Q}_k, Q_k$ respectively in the following proposition, we will obtain (89) for any k . Finally, as mentioned earlier, the desired result $\text{eig}(AQ A^T) \subseteq (0, \frac{4}{3} - \alpha_n]$ in Lemma 2 follows immediately from (89) and (88).

7.2.2 Step 2: Relation Between $\lambda_{\max}(A_n Q A_n^T)$ and its analog

In this subsection, we provide a proof of a weaker result $\text{eig}(A Q_n A^T) \subseteq (0, \frac{4}{3})$ under the conditions of Prop. 4; the proof of the desired result $\text{eig}(A Q_n A^T) \subseteq (0, \frac{4}{3} - \alpha_n]$ will be provided in the next subsection.

For simplicity, throughout this proof, we denote

$$W \triangleq W_n \in \mathbb{R}^{(N-d_n) \times d_n}, \quad \hat{Q} \triangleq \hat{Q}_n \in \mathbb{R}^{(N-d_n) \times (N-d_n)}, \quad \hat{A} \triangleq \hat{A}_n \in \mathbb{R}^{N \times (N-d_n)}.$$

According to the assumption of Prop. 4, we have

$$\hat{\lambda} \triangleq \lambda_{\max}(A \hat{Q} A^T) \in (0, \frac{4}{3} - \alpha_{n-1}]. \quad (93)$$

We first prove

$$0 \preceq \Theta \triangleq W^T \hat{Q} W \prec \frac{4}{3} I. \quad (94)$$

Since $\text{eig}(\hat{Q} \hat{A}^T \hat{A}) \subseteq (0, \infty)$ and \hat{A} is non-singular, thus $\hat{Q} \succ 0$. Then we have $\Theta = W^T \hat{Q} W \succeq 0$, which proves the first relation of (94). By the definition $W = \hat{A}^T A_n$ we have

$$\begin{aligned} \rho(\Theta) &= \rho(A_n^T \hat{A} \hat{Q} \hat{A}^T A_n) = \max_{v \in \mathbb{R}^{d_n \times 1}, \|v\|=1} v^T A_n^T \hat{A} \hat{Q} \hat{A}^T A_n v \\ &\leq \rho(\hat{A} \hat{Q} \hat{A}^T) \max_{v \in \mathbb{R}^{d_n \times 1}, \|v\|=1} \|A_n v\|^2 = \rho(\hat{A} \hat{Q} \hat{A}^T) \|A_n\|^2 = \rho(\hat{A} \hat{Q} \hat{A}^T) < \frac{4}{3}, \end{aligned} \quad (95)$$

where the last equality is due to the assumption $A_n^T A_n = I$, and the last inequality is due to the assumption (91). By (95) we have $\Theta \prec \frac{4}{3} I$, thus (94) is proved.

We apply a trick that we have previously used: factorize Q_n and change the order of multiplication. To be specific, Q_n defined in (92) can be factorized as

$$Q_n = \begin{bmatrix} I & 0 \\ -\frac{1}{2} W^T & I \end{bmatrix} \begin{bmatrix} \hat{Q} & 0 \\ 0 & I - \frac{1}{4} W^T \hat{Q} W \end{bmatrix} \begin{bmatrix} I & -\frac{1}{2} W \\ 0 & I \end{bmatrix} = J \begin{bmatrix} \hat{Q} & 0 \\ 0 & C \end{bmatrix} J^T, \quad (96)$$

where $J \triangleq \begin{bmatrix} I & 0 \\ -\frac{1}{2} W^T & I \end{bmatrix}$, I in the upper left block denotes the $(N - d_n)$ -dimensional identity matrix, I in the lower right block denotes the d_n -dim identity matrix, and

$$C \triangleq I - \frac{1}{4} W^T \hat{Q} W \in \mathbb{R}^{d_n \times d_n}. \quad (97)$$

It is easy to prove

$$\text{eig}(A Q_n A^T) \subseteq (0, \infty). \quad (98)$$

In fact, we only need to prove $Q_n \succ 0$. According to (96), we only need to prove $\begin{bmatrix} \hat{Q} & 0 \\ 0 & C \end{bmatrix} \succ 0$. This follows from $\hat{Q} \succ 0$ and the fact $C = I - \frac{1}{4} W^T \hat{Q} W \stackrel{(94)}{\succ} I - \frac{1}{3} I \succ 0$. Thus (98) is proved.

It remains to prove

$$\rho(A Q_n A^T) < \frac{4}{3}. \quad (99)$$

Denote $\hat{B} \triangleq \hat{A}^T \hat{A} \in \mathbb{R}^{(N-d_n) \times (N-d_n)}$, then we can write $A^T A$ as

$$A^T A = \begin{bmatrix} \hat{B} & W \\ W^T & I \end{bmatrix}. \quad (100)$$

We simplify the expression of $\rho(AQ_n A^T)$ as follows:

$$\rho(AQ_n A^T) = \rho \left(AJ \begin{bmatrix} \hat{Q} & 0 \\ 0 & C \end{bmatrix} J^T A^T \right) = \rho \left(\begin{bmatrix} \hat{Q} & 0 \\ 0 & C \end{bmatrix} J^T A^T AJ \right). \quad (101)$$

By algebraic computation, we have

$$\begin{aligned} J^T A^T AJ &= \begin{bmatrix} I & -\frac{1}{2}W \\ 0 & I \end{bmatrix} \begin{bmatrix} \hat{B} & W \\ W^T & I \end{bmatrix} \begin{bmatrix} I & 0 \\ -\frac{1}{2}W^T & I \end{bmatrix} \\ &= \begin{bmatrix} I & -\frac{1}{2}W \\ 0 & I \end{bmatrix} \begin{bmatrix} \hat{B} - \frac{1}{2}WW^T & W \\ \frac{1}{2}W^T & I \end{bmatrix} = \begin{bmatrix} \hat{B} - \frac{3}{4}WW^T & \frac{1}{2}W \\ \frac{1}{2}W^T & I \end{bmatrix}, \end{aligned} \quad (102)$$

thus

$$Y \triangleq \begin{bmatrix} \hat{Q} & 0 \\ 0 & C \end{bmatrix} J^T A^T AJ = \begin{bmatrix} \hat{Q} & 0 \\ 0 & C \end{bmatrix} \begin{bmatrix} \hat{B} - \frac{3}{4}WW^T & \frac{1}{2}W \\ \frac{1}{2}W^T & I \end{bmatrix} = \begin{bmatrix} \hat{Q}\hat{B} - \frac{3}{4}\hat{Q}WW^T & \frac{1}{2}\hat{Q}W \\ \frac{1}{2}CW^T & C \end{bmatrix}. \quad (103)$$

Suppose $\lambda > 0$ is the maximal eigenvalue of Y . According to (101) that $\rho(AQ_n A^T) = \rho(Y)$, we also have $\lambda = \lambda_{\max}(AQ_n A^T)$. To prove (99), we only need to prove

$$\lambda < \frac{4}{3}. \quad (104)$$

Suppose $v \in \mathbb{R}^{N \times 1} \setminus \{0\}$ is the eigenvector corresponding to λ , i.e. $Zv = \lambda v$. Partition v into $v = \begin{bmatrix} v_1 \\ v_0 \end{bmatrix}$, where $v_1 \in \mathbb{R}^{N-d_n}$, $v_0 \in \mathbb{R}^{d_n}$. According to the expression of Z in (103), $Zv = \lambda v$ implies

$$(\hat{Q}\hat{B} - \frac{3}{4}\hat{Q}WW^T)v_1 + \frac{1}{2}\hat{Q}Wv_0 = \lambda v_1, \quad (105a)$$

$$\frac{1}{2}CW^T v_1 + Cv_0 = \lambda v_0. \quad (105b)$$

If $\lambda I - C$ is singular, i.e. λ is an eigenvalue of C , then by (94) we have $\frac{2}{3}I \prec C = 1 - \frac{1}{4}\Theta \preceq I$, which implies $\lambda \leq 1$, thus (104) holds. In the following, we assume

$$\lambda I - C \text{ is non-singular.} \quad (106)$$

An immediate consequence is

$$v_1 \neq 0,$$

since otherwise (105b) implies $Cv_0 = \lambda v_0$, which combined with (106) leads to $v_0 = 0$ and thus $v = 0$, a contradiction.

By (105b) we get

$$v_0 = \frac{1}{2}(\lambda I - C)^{-1}CW^T v_1.$$

Plugging into (105a), we obtain

$$\lambda v_1 = (\hat{Q}\hat{B} - \frac{3}{4}\hat{Q}WW^T)v_1 + \frac{1}{2}\hat{Q}W\frac{1}{2}(\lambda I - C)^{-1}CW^Tv_1 = (\hat{Q}\hat{B} + \hat{Q}W\Phi W^T)v_1, \quad (107)$$

where

$$\begin{aligned} \Phi &\triangleq -\frac{3}{4}I + \frac{1}{4}(\lambda I - C)^{-1}C = -I + \frac{1}{4}[I + (\lambda I - C)^{-1}C] \\ &= -I + \frac{\lambda}{4}(\lambda I - C)^{-1} = -I + \lambda[(4\lambda - 4)I + \Theta]^{-1}. \end{aligned} \quad (108)$$

Here we have used the definition $C = I - \frac{1}{4}W^T\hat{Q}W = I - \frac{1}{4}\Theta$. Since Θ is a symmetric matrix, Φ is also a symmetric matrix.

Define

$$\tilde{H} \triangleq \hat{Q}W\Phi W^T \in \mathbb{R}^{(N-d_n) \times (N-d_n)}, H \triangleq W^T\hat{Q}W\Phi = \Theta\Phi \in \mathbb{R}^{d_n \times d_n}. \quad (109)$$

As a well-known linear algebra result, \tilde{H} and H have the same non-zero eigenvalues. Note that $\lambda_{\max}(H)$ may not be equal to $\lambda_{\max}(\tilde{H})$ due to the possible zero eigenvalues. Nevertheless, we can define $\lambda_{\max}^+(X) \triangleq \max\{\lambda_{\max}(X), 0\}$, and then we have

$$\lambda_{\max}^+(\tilde{H}) = \lambda_{\max}^+(H).$$

According to (109) and (108), we know

$$\begin{aligned} H &= \Theta\Phi = \Theta(-I + \lambda[(4\lambda - 4)I + \Theta]^{-1}) \\ &= -\Theta + \lambda\Theta[(4\lambda - 4)I + \Theta]^{-1} \\ &= -\Theta + \lambda(I - (4\lambda - 4)[(4\lambda - 4)I + \Theta]^{-1}) \\ &= -\Theta + \lambda I - \lambda(4\lambda - 4)[(4\lambda - 4)I + \Theta]^{-1}. \end{aligned}$$

It is well-known that if $\alpha I + \Theta$ is invertible, then Θ has an eigenvalue θ iff $(\alpha I + \Theta)^{-1}$ has an eigenvalue $(\alpha + \theta)^{-1}$, and the corresponding eigen-vectors are the same. Similarly, since we already assumed $(4\lambda - 4)I + \Theta$ is invertible, θ is an eigenvalue of Θ iff $H = -\Theta + \lambda I - \lambda(4\lambda - 4)[(4\lambda - 4)I + \Theta]^{-1}$ has an eigenvalue $-\theta + \lambda - \lambda(4\lambda - 4)[(4\lambda - 4) + \theta]^{-1}$. Recall that $\Theta = W^T\hat{Q}W$ satisfies $0 \preceq \Theta \preceq \hat{\lambda}I$, thus any eigenvalue θ satisfies $0 \leq \theta \leq \hat{\lambda}$. Therefore

$$\lambda_{\max}(H) \leq \max_{\theta \in [0, \hat{\lambda}]} \left\{ -\theta + \lambda - \frac{\lambda(4\lambda - 4)}{(4\lambda - 4) + \theta} \right\} \triangleq g(\theta). \quad (110)$$

Since $v_1 \neq 0$, without loss of generality, we can assume $\|v_1\| = 1$. We have

$$\begin{aligned} \lambda &= v_1^T \hat{Q}\hat{B}v_1 + v_1^T \tilde{H}v_1 \leq \hat{\lambda} + v_1^T \tilde{H}v_1 \leq \hat{\lambda} + \lambda_{\max}^+(\tilde{H}) = \hat{\lambda} + \lambda_{\max}^+(H) \\ &\leq \hat{\lambda} + \max\{0, \max_{\theta \in [0, \hat{\lambda}]} \left\{ -\theta + \lambda - \frac{\lambda(4\lambda - 4)}{(4\lambda - 4) + \theta} \right\}\}, \end{aligned} \quad (111)$$

where the first equality is due to (107), the first inequality is due to the induction hypothesis, the second inequality uses the obvious relation $\lambda_{\max}(\tilde{H}) \leq \lambda_{\max}^+(\tilde{H})$, and the last inequality is due to (110).

To prove (104), we consider two cases.

Case 1: $\max_{\theta \in [0, \hat{\lambda}]} g(\theta) \leq 0$. In this case, $\lambda \leq \hat{\lambda} < 4/3$, where the first inequality is due to (111), and the second inequality is due to the induction hypothesis. Thus in Case 1 (104) holds.

Case 2: $\max_{\theta \in [0, \hat{\lambda}]} g(\theta) > 0$. Then there exists some $\theta \geq 0$ such that $g(\theta) > 0$. Note that $g(\theta)$ can also be expressed as $g(\theta) = \theta(-1 + \frac{\lambda}{(4\lambda-4)+\theta})$, thus

$$-1 + \frac{\lambda}{(4\lambda-4)+\theta} > 0. \quad (112)$$

If $\lambda < 1$, then (104) already holds; so we can assume $\lambda > 1$. Thus (112) implies $1 < \frac{\lambda}{(4\lambda-4)+\theta} \leq \frac{\lambda}{4\lambda-4}$, which leads to $\lambda < \frac{4}{3}$. Thus in Case 2 (104) also holds. This finishes the proof of (104).

Remark: The proof of this subsection can lead to an alternative proof of Lemma 2. In particular, the induction step (Step 2) of Section 6.2 can be replaced by the proof here. The proof presented here is more complicated and less intuitive than the one in Section 6.2 (which is just a straightforward application of Strang's result Lemma 5, but the benefit is that it can help establish a stronger bound of λ , as done in the next subsection.

7.2.3 Step 3: More Precise Bound of λ

We will continue the proof in Section 7.2.2, to further prove

$$\lambda = \lambda_{\max}(AQ_n A^T) \leq 4/3 - \alpha_n. \quad (113)$$

We rewrite (111) as follows:

$$\lambda \leq \hat{\lambda} + \max\{0, \max_{\theta \in [0, \hat{\lambda}]} g(\theta)\}, \text{ where } g(\theta) = \lambda - \frac{\lambda(4\lambda-4)}{4\lambda-4+\theta} - \theta. \quad (114)$$

If $\lambda < 1$, then we are done since $1 \leq 4/3 - \alpha_n$. Assume $1 \leq \lambda < 4/3$ from now on.

We first analyze the function $g(\theta)$. Taking the derivative of g , we get

$$g'(\theta) = \frac{\lambda(4\lambda-4)}{(4\lambda-4+\theta)^2} - 1 = \frac{(\sqrt{\lambda(4\lambda-4)} + 4\lambda-4+\theta)(\sqrt{\lambda(4\lambda-4)} - 4\lambda+4-\theta)}{(4\lambda-4+\theta)^2}.$$

Since $\lambda > 1$ and $\theta \geq 0$, the term in the first bracket in the numerator is positive. Define

$$\theta^* = \sqrt{\lambda(4\lambda-4)} - 4\lambda + 4 > 0,$$

where the inequality holds due to $\lambda < 4/3$. Then we have

$$g'(\theta) \begin{cases} \geq 0, & \theta \leq \theta^*; \\ \leq 0, & \theta > \theta^*. \end{cases}$$

Therefore, $g(\theta)$ is increasing in $[0, \theta^*]$ and decreasing in $[\theta^*, \infty)$. This implies

$$g(\theta) \leq g(\theta^*), \forall \theta \geq 0. \quad (115)$$

According to $0 < \lambda < 4/3$, we have $\lambda > \sqrt{\lambda(4\lambda-4)} = 4\lambda-4+\theta^* \Rightarrow -1 + \frac{\lambda}{4\lambda-4+\theta^*} > 0 \Rightarrow g(\theta^*) > 0$. Together with (115) we obtain $\max\{0, \max_{\theta \in [0, \hat{\lambda}]} g(\theta)\} \leq g(\theta^*)$. Substituting into (114), we obtain

$$\lambda \leq \hat{\lambda} + g(\theta^*).$$

We will derive an inequality on λ and $\hat{\lambda}$ from the above relation as below. Substituting the expression of $g(\cdot)$ into the relation, we obtain

$$\lambda \leq \hat{\lambda} + \lambda - \frac{\lambda(4\lambda - 4)}{4\lambda - 4 + \theta^*} - \theta^* \implies \hat{\lambda} \geq \frac{\lambda(4\lambda - 4)}{4\lambda - 4 + \theta^*} + \theta^* = \sqrt{\lambda(4\lambda - 4)} + \theta^* = 2\sqrt{\lambda(4\lambda - 4)} - 4\lambda + 4.$$

This implies

$$\begin{aligned} & \hat{\lambda}^2 + (4\lambda - 4)^2 + 2\hat{\lambda}(4\lambda - 4) \geq 4\lambda(4\lambda - 4) \\ \iff & \hat{\lambda}^2 - \lambda^2 + 2(\hat{\lambda} - \lambda)(4\lambda - 4) + (\lambda - (4\lambda - 4))^2 \geq 0 \\ \iff & (\hat{\lambda} - \lambda)(\hat{\lambda} + \lambda) + 2(\hat{\lambda} - \lambda)(4\lambda - 4) + (4 - 3\lambda)^2 \geq 0. \end{aligned} \quad (116a)$$

Define

$$\delta = 4/3 - \lambda \in (0, 1/3), \quad \hat{\delta} = 4/3 - \hat{\lambda} \in (0, 4/3). \quad (117)$$

Substituting into (116a), we obtain

$$\begin{aligned} & (\delta - \hat{\delta})(8/3 - \delta - \hat{\delta}) + (\delta - \hat{\delta})(8/3 - 8\delta) + 9\delta^2 \geq 0 \\ \iff & (\delta - \hat{\delta})(16/3 - 9\delta - \hat{\delta}) + 9\delta^2 \geq 0 \\ \iff & \frac{16}{3}\delta - \frac{16}{3}\hat{\delta} + 8\hat{\delta}\delta + \delta^2 \geq 0 \\ \iff & \delta \geq \frac{\hat{\delta}(16 - 3\hat{\delta})}{8(2 + 3\hat{\delta})} = h(\hat{\delta}). \end{aligned}$$

It is easy to verify that $h(t)$ is increasing in $t \in [0, 4/3]$; in fact, $h'(t) = \frac{36}{(2+3t)^2} - 1 = \frac{(8+3t)(4-3t)}{(2+3t)^2} \geq 0$ for $t \in [0, 4/3]$. According to (93), we have $\hat{\delta} = 4/3 - \hat{\lambda} \geq \alpha_{n-1}$. Applying the monotonicity of h , we have

$$\delta \geq h(\hat{\delta}) \geq h(\alpha_{n-1}) = \alpha_n,$$

which combined with (117) leads to (113). This finishes the proof of Proposition 4.

7.2.4 Proof of Claim 7.1

Define another sequence as $\omega_k = \frac{16}{3\alpha_k} - 9k$. Then $\alpha_k = \frac{16}{3} \frac{1}{9k + \omega_k}$ and $\omega_1 = 7, \omega_2 = 38/5$. We then derive the recurrence equation of ω_k . According to (80), we have

$$\begin{aligned} & \frac{16}{3} \frac{1}{9k + 9 + \omega_{k+1}} = \frac{2}{3} \frac{1}{9k + \omega_k} \frac{16 - 16/(9k + \omega_k)}{2 + 16/(9k + \omega_k)} = \frac{2}{3} \frac{1}{9k + \omega_k} \frac{16(9k + \omega_k - 1)}{2(9k + \omega_k + 8)} \\ \implies & 9k + 9 + \omega_{k+1} = \frac{(9k + \omega_k)(9k + \omega_k + 8)}{9k + \omega_k - 1} \\ \implies & \omega_{k+1} = \omega_k + \frac{1}{9k + \omega_k - 1} [(9k + \omega_k)(9k + \omega_k + 8) - (9k + \omega_k - 1)(9k + 9 + \omega_k)] \\ \implies & \omega_{k+1} = \omega_k + \frac{9}{9k + \omega_k - 1}. \end{aligned}$$

It is easy to see that $\omega_k > 0 \implies \omega_{k+1} > \omega_k > 0$, thus

$$\omega_k > \omega_1 = 7, \forall k.$$

Furthermore, $\omega_{k+1} = \omega_k + \frac{9}{9k+\omega_k-1} \leq \omega_k + \frac{1}{k}$, thus

$$\omega_k \leq \omega_1 + \sum_{j=1}^{k-1} \frac{1}{j} \leq 8 + \log(k-1).$$

The lower bound and upper bound on ω_k imply upper and lower bounds on α_k :

$$\frac{16}{3} \frac{1}{9k+7} \geq \alpha_k \geq \frac{16}{3} \frac{1}{9k+8+\log(k-1)}. \quad (118)$$

As a side comment, this implies that $\lim_{k \rightarrow \infty} \alpha_k = \frac{16}{27k} \approx \frac{0.59}{k}$. For our purpose, we need a universal lower bound on α_k . When $k \geq 3$, we have $3k \geq 8 + \log(k-1)$, thus $12k \geq 9k + 8 + \log(k-1)$, which further implies

$$\frac{16}{3} \frac{1}{9k+8+\log(k-1)} \geq \frac{4}{9k}, \quad \forall k \geq 3.$$

Combining with the bound (118), we obtain

$$\alpha_k \geq \frac{4}{9k} > \frac{4}{9(k+1)}, \quad \forall k \geq 3.$$

Notice that $\alpha_1 = \frac{1}{3} > \frac{4}{9} \cdot \frac{1}{2}$, and $\alpha_2 = \frac{5}{24} > \frac{4}{9} \cdot \frac{1}{3}$, we have $\alpha_k > \frac{4}{9(k+1)}$ for any $k \geq 1$. This finishes the proof of the claim.

7.3 Proof of Lemma 3

We rewrite the lemma statement below. Suppose $P_i \in \mathbb{R}^{N \times N}$, $i = 1, \dots, n$ are projection matrices, then the lemma claims that

$$\frac{1}{n!} \sum_{\sigma=(\sigma_1, \dots, \sigma_n) \in \Gamma} P_{\sigma_n} P_{\sigma_{n-1}} \dots P_{\sigma_1} \preceq \frac{1}{n} \sum_i P_i. \quad (119)$$

We first prove the case $n = 2$, $n = 3$ and $n = 4$, then prove the general case $n = 2k$ and $n = 2k + 1$ separately.

When $n = 2$, (119) reduces to $P_1 P_2 + P_2 P_1 \preceq P_1 + P_2$. Notice that $P_i = P_i^2$ since P_i is a projection matrix, we have $P_1 + P_2 - P_1 P_2 + P_2 P_1 = P_1^2 + P_2^2 - P_1 P_2 + P_2 P_1 = (P_1 - P_2)^2 = (P_1 - P_2)^T (P_1 - P_2) \succeq 0$.

When $n = 3$, (119) reduces to $\frac{1}{6} \sum_{i,j,k \text{ are distinct}} P_i P_j P_k \preceq \frac{1}{3} (P_1 + P_2 + P_3)$. Note that $(P_i - P_k) P_j (P_i - P_k) \succeq 0$, thus

$$P_i P_j P_i + P_k P_j P_k \succeq P_i P_j P_k + P_k P_j P_i.$$

Summing up the above inequality for all possible triples (i, j, k) , we get

$$\sum_{i \neq j} P_i P_j P_i \succeq \sum_{i,j,k \text{ are distinct}} P_i P_j P_k. \quad (120)$$

We then need to bound the left-hand-side of the above inequality. Since $I - P_j \succeq 0$, we have $P_i (I - P_j) P_i \succeq 0$, which implies $P_i \succeq P_i P_j P_i$. Summing up this inequality for all pairs $i \neq j$, we obtain $\frac{1}{6} \sum_{i \neq j} P_i P_j P_i \preceq \frac{1}{3} (P_1 + P_2 + P_3)$. Combining with (120), we obtain the desired inequality $\frac{1}{6} \sum_{i,j,k \text{ are distinct}} P_i P_j P_k \preceq \frac{1}{3} (P_1 + P_2 + P_3)$.

The proof for $n = 4$ illustrates partially the gist of a general proof, so we present this proof. When $n = 4$, (119) reduces to $\frac{1}{24} \sum_{i,j,k,l \text{ are distinct}} P_i P_j P_k P_l \preceq \frac{1}{4}(P_1 + P_2 + P_3 + P_4)$. Similar to (120) in the $n = 3$ case, we first prove

$$\frac{1}{24} \sum_{i,j,k,l \text{ are distinct}} P_i P_j P_k P_l \leq \frac{1}{12} \sum_{i \neq j} P_i P_j P_i. \quad (121)$$

To prove this inequality, we need the following two basic inequalities:

$$\begin{aligned} (P_i - P_l)(P_j + P_k)^2(P_i - P_l) &\succeq 0, \\ (P_i + P_l)(P_j - P_k)^2(P_i + P_l) &\succeq 0. \end{aligned}$$

Summing up these two inequalities, we can eliminate terms like $P_i P_j P_k P_i$ (with three distinct subscripts) and keep the terms like $P_i P_j P_i$ (with two distinct subscripts) and $P_i P_j P_k P_l$ (with four distinct subscripts), to obtain

$$P_i P_j P_i + P_i P_k P_i + P_l P_j P_l + P_l P_k P_l \succeq P_i P_j P_k P_l + P_i P_k P_j P_l + P_l P_j P_k P_i + P_l P_k P_j P_i.$$

Summing up this inequality for all possible (i, j, k, l) that are distinct, we obtain (121). Similar to the proof of $n = 3$ case, we have $\frac{1}{12} \sum_{i \neq j} P_i P_j P_i \leq \frac{1}{4}(P_1 + P_2 + P_3 + P_4)$, thus combining with (121) we obtain the desired result.

We next prove the case $n = 2k$, where $k \geq 2$ is a positive integer. We will prove that

$$E_{\sigma \in \Gamma}(P_{\sigma_n} P_{\sigma_{n-1}} \dots P_{\sigma_1}) \preceq E_{\pi \in \Gamma_k}(P_{\pi_1} \dots P_{\pi_{k-1}} P_{\pi_k} P_{\pi_{k-1}} \dots P_{\pi_1}), \quad (122)$$

where Γ_k is the set of k -permutations of $1, 2, \dots, n$ (here, a k -permutation is a permutation of k distinct numbers chosen from $1, 2, \dots, n$), and $E_{\sigma \in \Gamma}$ and $E_{\pi \in \Gamma_k}$ denote the expectation over a uniform distribution on Γ and Γ_k respectively.

To prove (122), we need the following fact: for any $\epsilon = (\epsilon_1, \dots, \epsilon_k) \in \{1, -1\}^k$, we have

$$G_{\sigma, \epsilon} \triangleq (P_{\sigma_n} + \epsilon_1 P_{\sigma_1})(P_{\sigma_{n-1}} + \epsilon_2 P_{\sigma_2}) \dots (P_{\sigma_{k+1}} + \epsilon_k P_{\sigma_k})(P_{\sigma_{k+1}} + \epsilon_k P_{\sigma_k}) \dots (P_{\sigma_n} + \epsilon_1 P_{\sigma_1}) \succeq 0. \quad (123)$$

This relation holds because for any positive-semidefinite matrix X and any symmetric matrix Y , we have $YXY = Y^T XY \succeq 0$. Applying this fact k times leads to (123).

The expression of $G_{\sigma, \epsilon}$ in (123) involves 2^k terms in the form of $P_{i_1} P_{i_2} \dots P_{i_n}$. To prove (122), only two terms are of interest to us. The strategy is to pick ϵ_i 's properly so that summing up a bunch of relations of the form (123) will eliminate all but the two desired terms. We elaborate this strategy below.

Define

$$\begin{aligned} \Lambda_k &\triangleq \{(\epsilon_1, \dots, \epsilon_k) \in \{1, -1\}^k \mid \text{the number of } -1 \text{ in } \epsilon_1, \dots, \epsilon_k \text{ is odd}\}, \\ \Lambda_k^c &= \{(\epsilon_1, \dots, \epsilon_k) \in \{1, -1\}^k \mid \text{the number of } -1 \text{ in } \epsilon_1, \dots, \epsilon_k \text{ is even}\}. \end{aligned}$$

For example, when $k = 3$, $\Lambda_3 = \{(-1, 1, 1), (1, -1, 1), (1, 1, -1), (-1, -1, -1)\}$, and the complement $\Lambda_3^c = \{(1, 1, 1), (-1, -1, 1), (-1, 1, -1), (1, -1, -1)\}$. As a well-known fact,

$$|\Lambda_k^c| - |\Lambda_k| = \sum_{i \text{ is even}, 0 \leq i \leq n} \binom{n}{i} - \sum_{i \text{ is odd}, 0 \leq i \leq n} \binom{n}{i} = (1 - 1)^k = 0, \quad (124)$$

This matrix $G_{\sigma, \epsilon}$ can be expressed as the sum of 2^k terms, and each term is of the form $\pm P_{\pi_1} \dots P_{\pi_n}$, where $\pi_i \in \{\sigma_i, \sigma_{n+1-i}\}$. For the fixed permutation σ , define a set

$$\Omega(\sigma) = \{(\pi_1, \dots, \pi_n) \mid \pi_i \in \{\sigma_i, \sigma_{n+1-i}\}, \forall i\}$$

We partition the set into three subsets:

$$\begin{aligned}\Omega_0(\sigma) &= \{(\pi_1, \dots, \pi_n) \in \Omega \mid \pi_i = \pi_{n+1-i}, \forall i\}, \\ \Omega_1(\sigma) &= \{(\pi_1, \dots, \pi_n) \in \Omega \mid \pi_i \neq \pi_{n+1-i}, \forall i\}, \\ \Omega_2(\sigma) &= \Omega \setminus (\Omega_0 \cap \Omega_1).\end{aligned}$$

For most of the proof, we will use the abbreviation $\Omega_t = \Omega_t(\sigma)$, $t = 0, 1, 2$. For any $\pi = (\pi_1, \dots, \pi_n) \in \Omega$, define an indicator vector of π as $\delta(\pi) = (\delta_1, \dots, \delta_k)$, where each δ_i is determined by

$$\delta_i = \mathbb{I}(\pi_i - \pi_{n+1-i}) = \begin{cases} 0, & \pi_i = \pi_{n+1-i}, \\ 1, & \pi_i \neq \pi_{n+1-i}, \end{cases} \quad (125)$$

where $\mathbb{I}(z)$ equals 0 if $z = 0$ and equals 1 if $z \neq 0$. For example, when $n = 6$ and $\pi = (\sigma_1, \sigma_2, \sigma_3, \sigma_4, \sigma_2, \sigma_6)$, the corresponding indicator vector is $(0, 0, 1)$; when $\pi = (\sigma_1, \sigma_2, \sigma_3, \sigma_3, \sigma_2, \sigma_1)$, the indicator vector is $(0, 0, 0)$. Clearly, we have

$$\delta(\pi) = (0, 0, \dots, 0), \forall \pi \in \Omega_0; \quad \delta(\pi) = (1, 1, \dots, 1), \forall \pi \in \Omega_1; \quad \delta(\pi) \notin \{0_k, 1_k\}, \forall \pi \in \Omega_2. \quad (126)$$

In the expression of $G_{\sigma, \epsilon}$, half of the terms have coefficient 1 and the other half have coefficient -1 . To understand which terms have coefficient 1 and which have coefficient -1 , consider a special $\epsilon = (-1, 1, \dots, 1)$, i.e., $\epsilon_1 = -1$ and all other $\epsilon_i = 1$. A term with coefficient -1 has the form $P_{\sigma_1} P_{\pi_2} \dots P_{\pi_{n-1}} P_{\sigma_n}$ or $P_{\sigma_n} P_{\pi_2} \dots P_{\pi_{n-1}} P_{\sigma_1}$, i.e., with an indicator vector whose first element $\delta_1 = 1$, and a term with coefficient 1 has the form $P_{\sigma_1} P_{\pi_{n-1}} \dots P_{\pi_2} P_{\sigma_1}$ or $P_{\sigma_n} P_{\pi_{n-1}} \dots P_{\pi_2} P_{\sigma_n}$, i.e., with an indicator vector whose first element $\delta_1 = 0$. We can see that the coefficient is in fact $\epsilon_1^{\delta_1}$. For general $\epsilon \in \Lambda$ and $\pi \in \Omega$, the coefficient of $P_{\pi_n} P_{\pi_{n-1}} \dots P_{\pi_2} P_{\pi_1}$ in $G_{\sigma, \epsilon}$ is $(\epsilon_1)^{\delta_1} \dots (\epsilon_k)^{\delta_k}$, where $\delta = \delta(\pi)$ is defined as in (125). We can then write the expression of $G_{\sigma, \epsilon}$ as

$$G_{\sigma, \epsilon} = \sum_{\pi \in \Omega} \epsilon_1^{\delta_1} \dots \epsilon_k^{\delta_k} P_{\pi_n} P_{\pi_{n-1}} \dots P_{\pi_1}.$$

Summing up this relation for all ϵ in Λ_k , we have

$$\sum_{\epsilon \in \Lambda_k} G_{\sigma, \epsilon} = \sum_{\epsilon \in \Lambda_k} \sum_{\pi \in \Omega} \epsilon_1^{\delta_1} \dots \epsilon_k^{\delta_k} P_{\pi_n} P_{\pi_{n-1}} \dots P_{\pi_1} = \sum_{\pi \in \Omega} P_{\pi_n} P_{\pi_{n-1}} \dots P_{\pi_1} \left(\sum_{\epsilon \in \Lambda_k} \epsilon_1^{\delta_1} \dots \epsilon_k^{\delta_k} \right). \quad (127)$$

Note that in this expression, $\delta_1, \dots, \delta_k$ depend on π .

Denote $0_k = (0, 0, \dots, 0) \in \mathbb{R}^k$, $1_k = (1, \dots, 1) \in \mathbb{R}^k$. Define

$$g_k(\delta) \triangleq \sum_{\epsilon \in \Lambda_k} \epsilon_1^{\delta_1} \dots \epsilon_k^{\delta_k}, \quad h_k(\delta) \triangleq \sum_{\epsilon \in \Lambda_k^c} \epsilon_1^{\delta_1} \dots \epsilon_k^{\delta_k}.$$

For any $\delta \neq 0_k$, we have $g_k(\delta) + h_k(\delta) = \sum_{\epsilon \in \{1, -1\}^k} \epsilon_1^{\delta_1} \dots \epsilon_k^{\delta_k} = (1^{\delta_1} + (-1)^{\delta_1}) \dots (1^{\delta_k} + (-1)^{\delta_k}) = 0$, thus

$$h_k(\delta) = -g_k(\delta), \quad \forall \delta \neq 0_k. \quad (128)$$

It is easy to see that

$$\frac{1}{|\Lambda_k|} g_k(\delta) = \begin{cases} 1, & \delta = (0, 0, \dots, 0), \\ -1, & \delta = (1, 1, \dots, 1). \end{cases} \quad (129)$$

We will prove: for any $\delta \notin \{0_k, 1_k\}$,

$$g_k(\delta) = \sum_{\epsilon \in \Lambda_k} \epsilon_1^{\delta_1} \dots \epsilon_k^{\delta_k} = 0, \quad (130)$$

We prove (130) by induction on k . When $k = 2$, $\Lambda_2 = \{(-1, 1), (1, -1)\}$, we have:

$$\begin{aligned} \text{when } \delta = (0, 1), \quad g_2(\delta) &= (-1)^0 1^1 + 1^0 (-1)^1 = 1 - 1 = 0, \\ \text{when } \delta = (1, 0), \quad g_2(\delta) &= (-1)^1 1^0 + 1^1 (-1)^0 = -1 + 1 = 0. \end{aligned}$$

Assume (130) holds for $k - 1$, i.e.,

$$g_{k-1}(\hat{\delta}) = 0, \quad \forall \hat{\delta} \in \{0, 1\}^{k-1} \setminus \{0_{k-1}, 1_{k-1}\}. \quad (131)$$

According to (128), we have

$$h_{k-1}(\hat{\delta}) = 0, \quad \forall \hat{\delta} \in \{0, 1\}^{k-1} \setminus \{0_{k-1}, 1_{k-1}\}. \quad (132)$$

Now consider k . Since $\delta \neq 0_k$, there must exist some j such that $\delta_j = 1$; without loss of generality, we assume

$$\delta_k = 1. \quad (133)$$

Partition Γ_k into two sets:

$$\Lambda_{k,1} \triangleq \{\epsilon \in \Lambda_k \mid \epsilon_k = 1\}, \quad \Lambda_{k,2} \triangleq \{\epsilon \in \Lambda_k \mid \epsilon_k = -1\}. \quad (134)$$

If ϵ contains an odd number of -1 and the last element $\epsilon_k = 1$ (or $\epsilon_k = -1$), then the first $k - 1$ elements contain an odd (or even) number of -1 . Thus

$$\Lambda_{k,1} = \{(\hat{\epsilon}, 1) \mid \hat{\epsilon} \in \Lambda_{k-1}\}, \quad \Lambda_{k,2} = \{(\hat{\epsilon}, -1) \mid \hat{\epsilon} \in \Lambda_{k-1}^c\}.$$

Split $g_k(\delta)$ into two parts $g_k(\delta) = g_{k,1}(\delta) + g_{k,2}(\delta)$, where

$$g_{k,1}(\delta) = \sum_{\epsilon \in \Lambda_{k,1}} \epsilon_1^{\delta_1} \dots \epsilon_k^{\delta_k}, \quad g_{k,2}(\delta) = \sum_{\epsilon \in \Lambda_{k,2}} \epsilon_1^{\delta_1} \dots \epsilon_k^{\delta_k}.$$

Denote $\hat{\delta} = (\delta_1, \dots, \delta_{k-1})$. We already assume $\delta \neq 1_k$ and $\delta_k = 1$, so we know

$$\hat{\delta} \neq 1_{k-1}. \quad (135)$$

But it is possible that $\hat{\delta} = 0_{k-1}$. Consider two cases.

Case 1: $\hat{\delta} = 0_{k-1}$, i.e., $\delta = (0_{k-1}, 1)$.

In this case

$$\begin{aligned} g_{k,1}(\delta) &= \sum_{\epsilon \in \Lambda_{k,1}} \epsilon_1^{\delta_1} \dots \epsilon_k^{\delta_k} = \sum_{\epsilon \in \Lambda_{k,1}} \epsilon_1^0 \dots \epsilon_{k-1}^0 \epsilon_k^1 = \sum_{\epsilon \in \Lambda_{k,1}} \epsilon_k^1 = \sum_{\epsilon \in \Lambda_{k,1}} 1^1 = |\Lambda_{k,1}| = |\Lambda_{k-1}|, \\ g_{k,2}(\delta) &= \sum_{\epsilon \in \Lambda_{k,2}} \epsilon_1^{\delta_1} \dots \epsilon_k^{\delta_k} = \sum_{\epsilon \in \Lambda_{k,2}} \epsilon_1^0 \dots \epsilon_{k-1}^0 \epsilon_k^1 = \sum_{\epsilon \in \Lambda_{k,2}} \epsilon_k^1 = \sum_{\epsilon \in \Lambda_{k,2}} (-1)^1 = -|\Lambda_{k,2}| = -|\Lambda_{k-1}^c|, \end{aligned}$$

Thus

$$g_k(\delta) = g_{k,1}(\delta) + g_{k,2}(\delta) = |\Lambda_{k-1}| - |\Lambda_{k-1}^c| = 0,$$

where the last step is due to (124).

Case 2: $\hat{\delta} \neq 0_{k-1}$. Together with (135), we have

$$\hat{\delta} \notin \{0_{k-1}, 1_{k-1}\}.$$

which enables us to apply the induction hypothesis (131) and its corollary (132). In fact,

$$\begin{aligned} g_{k,1}(\delta) &= \sum_{\epsilon \in \Lambda_{k,1}} \epsilon_1^{\delta_1} \dots \epsilon_k^{\delta_k} \stackrel{(133),(134)}{=} \sum_{\epsilon \in \Lambda_{k,1}} \epsilon_1^{\delta_1} \dots \epsilon_{k-1}^{\delta_{k-1}} 1^1 = \sum_{\hat{\epsilon} \in \Lambda_{k-1}} \hat{\epsilon}_1^{\delta_1} \dots \hat{\epsilon}_{k-1}^{\delta_{k-1}} = g_{k-1}(\hat{\delta}) \stackrel{(131)}{=} 0, \\ g_{k,2}(\delta) &= \sum_{\epsilon \in \Lambda_{k,2}} \epsilon_1^{\delta_1} \dots \epsilon_k^{\delta_k} \stackrel{(133),(134)}{=} \sum_{\epsilon \in \Lambda_{k,2}} \epsilon_1^{\delta_1} \dots \epsilon_{k-1}^{\delta_{k-1}} (-1)^1 = - \sum_{\hat{\epsilon} \in \Lambda_{k-1}^c} \hat{\epsilon}_1^{\delta_1} \dots \hat{\epsilon}_{k-1}^{\delta_{k-1}} = h_{k-1}(\hat{\delta}) \stackrel{(132)}{=} 0. \end{aligned}$$

Thus $g_k(\delta) = g_{k,1}(\delta) + g_{k,2}(\delta) = 0$.

In both cases, we have proved $g_k(\delta) = 0$, which finishes the induction step. Therefore (130) holds for any k .

Next, we analyze the sum $\sum_{\epsilon \in \Lambda_k} G_{\sigma, \epsilon}$. According to (127), we have

$$\begin{aligned} \sum_{\epsilon \in \Lambda_k} G_{\sigma, \epsilon} &= \sum_{\pi \in \Omega} P_{\pi_n} P_{\pi_{n-1}} \dots P_{\pi_1} \left(\sum_{\epsilon \in \Lambda_k} \epsilon_1^{\delta_1} \dots \epsilon_k^{\delta_k} \right) \\ &= \sum_{\pi \in \Omega} P_{\pi_n} P_{\pi_{n-1}} \dots P_{\pi_1} g_k(\delta(\pi)) \\ &\stackrel{(i)}{=} \sum_{\pi \in \Omega_0} P_{\pi_n} P_{\pi_{n-1}} \dots P_{\pi_1} \cdot g_k(0_k) + \sum_{\pi \in \Omega_1} P_{\pi_n} P_{\pi_{n-1}} \dots P_{\pi_1} \cdot g_k(1_k) + \sum_{\pi \in \Omega_2} P_{\pi_n} P_{\pi_{n-1}} \dots P_{\pi_1} \cdot g_k(\delta(\pi)) \\ &\stackrel{(ii)}{=} \sum_{\pi \in \Omega_0} P_{\pi_n} P_{\pi_{n-1}} \dots P_{\pi_1} \cdot |\Gamma_k| + \sum_{\pi \in \Omega_1} P_{\pi_n} P_{\pi_{n-1}} \dots P_{\pi_1} \cdot (-1) |\Gamma_k| + \sum_{\pi \in \Omega_2} P_{\pi_n} P_{\pi_{n-1}} \dots P_{\pi_1} \cdot 0 \\ &= |\Gamma_k| \left(\sum_{\pi \in \Omega_0} P_{\pi_n} P_{\pi_{n-1}} \dots P_{\pi_1} - \sum_{\pi \in \Omega_1} P_{\pi_n} P_{\pi_{n-1}} \dots P_{\pi_1} \right). \end{aligned}$$

where (i) is due to (126) and (ii) is due to (129), (130). According to (123), any $G_{\sigma, \epsilon} \succeq 0$, thus the above relation implies the following important relation

$$\sum_{\pi \in \Omega_0} P_{\pi_n} P_{\pi_{n-1}} \dots P_{\pi_1} \succeq \sum_{\pi \in \Omega_1} P_{\pi_n} P_{\pi_{n-1}} \dots P_{\pi_1} \quad (136)$$

Note that this relation holds for a fixed permutation σ and the corresponding set $\Omega_0 = \Omega(\sigma)$ and $\Omega_1(\sigma)$.

Each $\pi \in \Omega_0$ corresponds to a k -permutation χ of $(12 \dots n)$ determined by $\pi = (\chi_1 \dots \chi_{k-1} \chi_k \chi_k \chi_k \chi_{k-1} \dots \chi_1)$ and each $\pi \in \Omega_1$ corresponds to a permutation of $(12 \dots n)$. We rewrite (136) as

$$\sum_{\pi \in \Omega_0(\sigma)} P_{\pi_n} P_{\pi_{n-1}} \dots P_{\pi_1} \succeq \sum_{\pi \in \Omega_1(\sigma)} P_{\pi_n} P_{\pi_{n-1}} \dots P_{\pi_1}$$

and summing up this relation for all possible permutations $\sigma \in \Gamma$ leads to

$$E_{\chi \in \Gamma_k} (P_{\chi_1} \dots P_{\chi_{k-1}} P_{\chi_k} P_{\chi_k} P_{\chi_{k-1}} \dots P_{\chi_1}) \succeq E_{\sigma \in \Gamma} (P_{\sigma_n} P_{\sigma_{n-1}} \dots P_{\sigma_1}),$$

which is exactly (122).

It remains to prove

$$E_{\chi \in \Gamma_k} (P_{\chi_1} \cdots P_{\chi_{k-1}} P_{\chi_k} P_{\chi_{k-1}} \cdots P_{\pi_1}) \preceq \frac{1}{n} \sum_i P_i. \quad (137)$$

In fact, for any positive-semidefinite matrix X and any symmetric matrix Y , we have $YXY = Y^T XY \succeq 0$. Applying this fact $k-1$ times leads to (137).

Combining (122) and (137), we immediately obtain the desired result (119) for the case $n = 2k$.

The case that $n = 2k - 1$ is an odd number is almost the same, except that the key quantity $G_{\sigma, \epsilon}$ is now defined as

$$G_{\sigma, \epsilon} \triangleq (P_{\sigma_n} + \epsilon_1 P_{\sigma_1}) \cdots (P_{\sigma_{k+1}} + \epsilon_{k-1} P_{\sigma_{k-1}}) P_{\sigma_k} (P_{\sigma_{k+1}} + \epsilon_{k-1} P_{\sigma_{k-1}}) \cdots (P_{\sigma_n} + \epsilon_1 P_{\sigma_1}). \quad (138)$$

In words, we pair P_{σ_i} with $P_{\sigma_{n+1-i}}$ for $i = 1, \dots, k-1$ and leave P_{σ_k} alone (following the same rule it would have been paired with itself). The rest of the proof is almost the same as the even case, so we skip it. **Q.E.D.**

8 Numerical Experiments

In this section, we test the performance of cyclic ADMM and RP-ADMM for solving various kinds of linear systems. As a benchmark, we also test the gradient descent method (GD) with a constant stepsize $\alpha = 1/\lambda_{\max}(A'A)$ for solving the least square problem $\min_{x \in \mathbb{R}^N} \|Ax - b\|^2/2$. Of course there are many other advanced algorithms for solving the least square problem such as the conjugate gradient method, but we do not consider them since our focus is on testing the two ADMM algorithms. These two ADMM algorithms can be used to solve far more general problems than just linear systems, and we believe that the performance comparison for solving linear systems can shed light on more general scenarios.

In the numerical experiments, we set $b = 0$, thus the unique optimal solution is $x^* = 0$. The coefficient matrix A will be generated according to one of the random distributions below:

- Gauss: independent Gaussian entries $A_{i,j} \sim \mathcal{N}(0, 1)$.
- Log-normal: independent log-normal entries $A_{i,j} \sim \exp(\mathcal{N}(0, 1))$.
- Uniform: each entry is drawn independently from a uniform distribution on $[0, 1]$.
- Circulant Hankel: circulant Hankel matrix with independent standard Gaussian entries. More specifically, generate $\delta_1, \delta_2, \dots, \delta_N \sim \mathcal{N}(0, 1)$ and let $A_{i,j} = \delta_{i+j-1}$ (define $\delta_k = \delta_{k-N}$ if $k > N$). Note that the entries of the circulant Hankel matrix are not independent since one δ_i can appear in multiple positions.

For the two ADMM algorithms, we only consider the n -coordinate versions, i.e. each block consists of only one coordinate. We let the three tested algorithms start from the same random initial point

$y^0 = [x^0; \lambda^0]$ (GD will start from x^0). To measure the performance, we define the epoch complexity k to be the minimum k so that the relative error

$$\|Ax^k - b\|/\|Ax^0 - b\| < \epsilon,$$

where ϵ is a desired accuracy (we consider 10^{-2} and 10^{-311}). For the two ADMM algorithms, one epoch refers to one round of primal and dual steps; for GD, one epoch refers to one gradient step. The total computation time should be proportional to the epoch complexity since GD and the two ADMM variants have similar per-epoch cost¹²: a gradient descent step $x^{k+1} = x^k - \alpha A^T(Ax - b)$ contains two matrix-vector multiplications and thus takes time $2N^2 + O(N)$, and an ADMM round also takes time $2N^2 + O(N)$ (the primal update step of ADMM takes time $2N^2 + O(N)$ and the dual update step of ADMM takes time $O(N)$). We test 1000 random instances for $N \in \{3, 10\}$ and 300 random instances for $N = 100$, and record the geometric mean of the number of epochs. In the table, “Diverg. Ratio” represents the percentage of tested instances for which cyclic ADMM diverges and “CycADMM” represents “cyclic ADMM” (note that RP-ADMM converges in all instances we tested, so its divergence ratio is 0). Note that for cyclic ADMM we only report the epoch complexity when it converges, while for RPADMM and GD we report the epoch complexity in all tested instances. If restricting to the successful instances of cyclic ADMM, we find that the epoch complexity of RPADMM does not change too much, while the epoch complexity of GD will be reduced (significantly in some settings).

The simulation results are summarized in Table 2. The main observations from the simulation are:

- For all random distributions of A we tested, cyclic ADMM does not always converge even when N is fixed to be 3. For $N = 100$ and many random distributions, cyclic ADMM diverges with probability 1. This means that the divergence of cyclic ADMM is not merely a “worst-case” phenomenon, but actually quite common. When the dimension increases, the divergence ratio will increase.
- For standard Gaussian entries, cyclic ADMM converges with high probability. When cyclic ADMM converges, it converges faster than RP-ADMM and sometimes much faster.
- RPADMM typically converges faster than the basic gradient descent method and sometimes more than 10 times faster.

We have also tested BR-ADMM for solving the same problems, though the simulation results are not listed in the above table. As expected, BR-ADMM also always converges for solving these linear systems. The convergence speed is usually slower than RP-ADMM. Nevertheless, BR-ADMM can save some sampling time compared to RP-ADMM, and may be more favorable if random permutation is not available due to system architecture constraint. The detailed comparison of BR-ADMM and RP-ADMM, and the design of other randomized schemes or even deterministic schemes that outperform RP schemes are left as future work.

¹¹For high accuracy such as $\epsilon = 10^{-6}$, it takes too many epochs for the algorithms to converge when $n = 100$ as most matrices we generated are highly ill-conditioned, so we do not report the results. Based on the limited experiments for high accuracy, similar gaps between RP-ADMM and GD are observed.

¹²In matlab simulations each epoch of GD takes much less time than a round of ADMM because matlab implements matrix operations much faster than a “for” loop. For a more fair CPU time comparison, one should use other programming languages such as C.

¹³For cyclic ADMM, only record the iteration complexity in convergent instances.

Table 2: *Results of Solving Linear Systems by Cyclic ADMM, RP-ADMM and GD. For the two ADMM variants, one epoch refers to one round of primal and dual steps; for GD, one epoch refers to one gradient step.*

N	Diverg. Ratio	Epochs for $\epsilon = 0.01$			Epochs for $\epsilon = 0.001$		
		CycADMM ¹³	RPADMM	GD	CycADMM	RPADMM	GD
Gaussian							
3	0.7%	1.4e01	3.4e01	5.0e01	3.2e01	8.8e01	1.4e02
10	1.1%	4.1e01	1.8e02	2.0e02	1.2e02	1.1e03	1.5e03
100	3%	1.7e02	4.3e02	3.6e02	1.0e03	7.4e03	6.5e03
Log-normal							
3	0.8%	1.5e01	3.7e01	5.7e01	3.3e01	9.6e01	1.7e02
10	39.2%	1.2e02	3.4e02	6.4e02	3.2e02	2.4e03	6.3e03
100	100%	N/A	5.5e02	5.4e03	N/A	8.8e03	1.0e05
Uniform							
3	3.2%	2.8e01	7.4e01	1.5e02	7.0e01	2.6e02	6.0e02
10	83.0%	2.1e02	4.1e02	1.2e03	5.2e02	3.0e03	9.1e03
100	100%	N/A	9.1e02	1.4e04	N/A	1.4e04	9.7e04
Circulant Hankel							
3	5.6%	1.2e01	1.7e01	1.5e01	1.7e01	2.8e01	2.6e01
10	54.3%	4.2e01	6.0e01	6.5e01	7.5e01	1.3e02	1.7e02
100	100%	N/A	1.3e02	1.7e02	N/A	2.9e02	6.5e02

9 Concluding Remarks

In this paper, we prove the expected convergence of randomly permuted ADMM (RP-ADMM) for solving a non-singular square system of equations (extension to non-square systems is straightforward). We also prove a bound on the expected convergence rate of RP-ADMM for solving linear systems and the expected convergence rate of RP-BCD for solving quadratic problems. The motivation is to resolve the divergence issue of cyclic multi-block ADMM. Our result shows that RP-ADMM may serve as a simple remedy, and we expect RP-ADMM to be one of the important solvers in large-scale optimization. One interesting finding along the path is that the update matrix of RP-BCD has spectrum lying in $(-1/3, 1)$ instead of the commonly seen $(-1, 1)$.

Randomly permutation is widely known to be empirically better than independently randomized versions, but little was known about its theoretical properties in general. Note that most existing analyses of BCD (e.g. [33–35]) are applicable to both the cyclic update rule and the random permutation update rule. However, in light of a recent study which established an up to $O(n^2)$ gap between cyclic CD and R-CD [26], it is unlikely that RP-CD will have the same rate as cyclic CD. Our result in this paper established, for the first time, an $O(n)$ gap between RP-CD and cyclic-CD for general quadratic problems, making some progress towards the conjecture that RP-CD is faster than R-CD.

We emphasize that the convergence speed analysis of large-scale optimization has mostly been limited to independently randomized update order in the past decade. Going beyond independent randomized order is an important topic for enlarging the scope of large-scale optimization. Not only the analysis of random permutation is quite challenging, even the analysis of the most classical cyclic order is highly nontrivial [26]. There are quite a few open questions regarding the convergence rate of non-independent-randomized order. Regarding the random permutation order, a very interesting open question is the worst-case convergence rate of RP-BCD for quadratic problems. Due to the close relation with matrix AM-GM inequality, this problem seems to be a quite fundamental problem. Moving to ADMM, the similar questions about the convergence rate of various variants of ADMM, including RP-ADMM and BR-ADMM, are also open.

10 Acknowledgment

We thank an anonymous reviewer for many helpful comments on the manuscript, which enabled us to improve the presentation of the paper.

References

- [1] Stephen J Wright. Coordinate descent algorithms. *Mathematical Programming*, 151(1):3–34, 2015.
- [2] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends® in Machine Learning*, 3(1):1–122, 2011.

- [3] C. Chen, B. He, Y. Ye, and X. Yuan. The direct extension of ADMM for multi-block convex minimization problems is not necessarily convergent. *Mathematical Programming*, pages 1–23, 2014.
- [4] H. Wang, A. Banerjee, and Z.-Q. Luo. Parallel direction method of multipliers. In *Advances in Neural Information Processing Systems*, pages 181–189, 2014.
- [5] R. Glowinski and A. Marroco. Approximation par éléments finis d’ordre un, et la résolution, par pénalisation-dualité d’une classe de problèmes de dirichlet non linéaires. *ESAIM: Mathematical Modelling and Numerical Analysis-Modélisation Mathématique et Analyse Numérique*, 9(R2):41–76, 1975.
- [6] T. F. Chan and R. Glowinski. *Finite element approximation and iterative solution of a class of mildly non-linear elliptic equations*. Computer Science Department, Stanford University Stanford, 1978.
- [7] D. Gabay and B. Mercier. A dual algorithm for the solution of nonlinear variational problems via finite element approximation. *Computers & Mathematics with Applications*, 2(1):17–40, 1976.
- [8] B. He, M. Tao, and X. Yuan. Alternating direction method with Gaussian back substitution for separable convex programming. *SIAM Journal on Optimization*, 22(2):313–340, 2012.
- [9] B. He, M. Tao, and X. Yuan. Convergence rate and iteration complexity on the alternating direction method of multipliers with a substitution procedure for separable convex programming. *Math. Oper. Res.*, under revision, 2, 2012.
- [10] M. Hong and Z.-Q. Luo. On the linear convergence of the alternating direction method of multipliers. *arXiv preprint arXiv:1208.3922*, 2012.
- [11] D. Han and X. Yuan. A note on the alternating direction method of multipliers. *Journal of Optimization Theory and Applications*, 155(1):227–238, 2012.
- [12] C. Chen, Y. Shen, and Y. You. On the convergence analysis of the alternating direction method of multipliers with three blocks. In *Abstract and Applied Analysis*, volume 2013. Hindawi Publishing Corporation, 2013.
- [13] B. He, H.-K. Xu, and X. Yuan. On the proximal jacobian decomposition of alm for multiple-block separable convex minimization problems and its relationship to admm, 2013.
- [14] B. He, L. Hou, and X. Yuan. On full jacobian decomposition of the augmented lagrangian method for separable convex programming. *Preprint*, 2013.
- [15] W. Deng, M.-J. Lai, Z. Peng, and W. Yin. Parallel multi-block ADMM with $o(1/k)$ convergence. *arXiv preprint arXiv:1312.3040*, 2013.
- [16] T. Lin, S. Ma, and S. Zhang. On the convergence rate of multi-block ADMM. *arXiv preprint arXiv:1408.4265*, 2014.
- [17] M. Hong, T.-H. Chang, X. Wang, M. Razaviyayn, S. Ma, and Z.-Q. Luo. A block successive upper bound minimization method of multipliers for linearly constrained convex optimization. *arXiv preprint arXiv:1401.7079*, 2014.

- [18] X. Cai, D. Han, and X. Yuan. The direct extension of ADMM for three-block separable convex minimization models is convergent when one function is strongly convex. *Optimization Online*, 2014.
- [19] D. Sun, K.-C. Toh, and L. Yang. A convergent proximal alternating direction method of multipliers for conic programming with 4-block constraints. *arXiv preprint arXiv:1404.5378*, 2014.
- [20] T. Lin, S. Ma, and S. Zhang. On the global linear convergence of the admm with multi-block variables. *arXiv preprint arXiv:1408.4266*, 2014.
- [21] D. Han, X. Yuan, and W. Zhang. An augmented lagrangian based parallel splitting method for separable convex minimization with applications to image processing. *Mathematics of Computation*, 83(289):2263–2291, 2014.
- [22] X. Li, D. Sun, and K.-C. Toh. A schur complement based semi-proximal admm for convex quadratic conic programming and extensions. *Mathematical Programming*, pages 1–41, 2014.
- [23] M. Li, D. Sun, and K.-C. Toh. A convergent 3-block semi-proximal admm for convex minimization problems with one strongly convex block. *Asia-Pacific Journal of Operational Research*, page 1550024, 2015.
- [24] T. Lin, S. Ma, and S. Zhang. Iteration complexity analysis of multi-block admm for a family of convex minimization without strong convexity. *arXiv preprint arXiv:1504.03087*, 2015.
- [25] Wei Deng, Ming-Jun Lai, Zhimin Peng, and Wotao Yin. Parallel multi-block admm with $o(1/k)$ convergence. *Journal of Scientific Computing*, 71(2):712–736, 2017.
- [26] Ruoyu Sun and Yinyu Ye. Worst-case complexity of cyclic coordinate descent: $o(n^2)$ gap with randomized version. *arXiv preprint arXiv:1604.07130*, 2016.
- [27] D. Blatt, A. O. Hero, and H. Gauchman. A convergent incremental gradient method with a constant step size. *SIAM Journal on Optimization*, 18(1):29–51, 2007.
- [28] M. Schmidt, N. L. Roux, and F. Bach. Minimizing finite sums with the stochastic average gradient. *arXiv preprint arXiv:1309.2388*, 2013.
- [29] Mert Gürbüzbalaban, Asu Ozdaglar, and Pablo Parrilo. Why random reshuffling beats stochastic gradient descent. *arXiv preprint arXiv:1510.08560*, 2015.
- [30] S. Shalev-Shwartz and T. Zhang. Stochastic dual coordinate ascent methods for regularized loss. *The Journal of Machine Learning Research*, 14(1):567–599, 2013.
- [31] B. Recht and C. Ré. Parallel stochastic gradient algorithms for large-scale matrix completion. *Mathematical Programming Computation*, 5(2):201–226, 2013.
- [32] R. Sun. *Matrix Completion via Nonconvex Factorization: Algorithms and Theory*. PhD thesis, University of Minnesota, 2015.
- [33] P. Tseng. Convergence of a block coordinate descent method for nondifferentiable minimization. *Journal of optimization theory and applications*, 109(3):475–494, 2001.

- [34] A. Beck and L. Tetruashvili. On the convergence of block coordinate descent type methods. *SIAM Journal on Optimization*, 23(4):2037–2060, 2013.
- [35] R. Sun and M. Hong. Improved iteration complexity bounds of cyclic block coordinate descent for convex problems. In *NIPS 2015*, pages 638–678, 2015.
- [36] Stephen J Wright and Ching-Pei Lee. Analyzing random permutations for cyclic coordinate descent. *arXiv preprint arXiv:1706.00908*, 2017.
- [37] B. Recht and C. Ré. Beneath the valley of the noncommutative arithmetic-geometric mean inequality: conjectures, case-studies, and consequences. *arXiv preprint arXiv:1202.4184*, 2012.
- [38] Ruoyu Sun, Zhi-Quan Luo, and Yinyu Ye. On the expected convergence of randomly permuted admm. *arXiv preprint arXiv:1503.06387*, 2015.
- [39] Caihu Chen, Min Li, Xin Liu, and Yinyu Ye. Extended ADMM and BCD for nonseparable convex minimization models with quadratic coupling terms: convergence analysis and insights. *Mathematical Programming*, Nov 2017.
- [40] Y. Nesterov. Efficiency of coordinate descent methods on huge-scale optimization problems. *SIAM Journal on Optimization*, 22(2):341–362, 2012.
- [41] Dennis Leventhal and Adrian S Lewis. Randomized methods for linear constraints: convergence rates and conditioning. *Mathematics of Operations Research*, 35(3):641–654, 2010.
- [42] Ching-Pei Lee and Stephen J Wright. Random permutations fix a worst case for cyclic coordinate descent. *arXiv preprint arXiv:1607.08320*, 2016.
- [43] F. Kittaneh. Spectral radius inequalities for Hilbert space operators. *Proceedings of the American Mathematical Society*, pages 385–390, 2006.
- [44] W Gilbert Strang. Eigenvalues of jordan products. *The American Mathematical Monthly*, 69(1):37–40, 1962.