

ASYNCHRONOUS STOCHASTIC COORDINATE DESCENT: PARALLELISM AND CONVERGENCE PROPERTIES

JI LIU* AND STEPHEN J. WRIGHT†

Abstract. We describe an asynchronous parallel stochastic proximal coordinate descent algorithm for minimizing a composite objective function, which consists of a smooth convex function added to a separable convex function. In contrast to previous analyses, our model of asynchronous computation accounts for the fact that components of the unknown vector may be written by some cores simultaneously with being read by others. Despite the complications arising from this possibility, the method achieves a linear convergence rate on functions that satisfy an optimal strong convexity property and a sublinear rate $(1/k)$ on general convex functions. Near-linear speedup on a multicore system can be expected if the number of processors is $O(n^{1/4})$. We describe results from implementation on ten cores of a multicore processor.

Key words. stochastic coordinate descent, asynchronous parallelism, inconsistent read, composite objective

AMS subject classifications. 90C25, 68W20, 68W10, 90C05

1. Introduction. We consider the convex optimization problem

$$\min_x F(x) := f(x) + g(x), \quad (1.1)$$

where $f : \mathbb{R}^n \mapsto \mathbb{R}$ is a smooth convex function and $g : \mathbb{R}^n \mapsto \mathbb{R} \cup \{\infty\}$ is a separable, closed, convex, and extended real-valued function. “Separable” means that $g(x)$ can be expressed as $g(x) = \sum_{i=1}^n g_i((x)_i)$, where $(x)_i$ denotes the i th element of x and each $g_i : \mathbb{R} \mapsto \mathbb{R} \cup \{\infty\}$, $i = 1, 2, \dots, n$ is a closed, convex, and extended real-valued function.

Formulations of the type (1.1) arise in many data analysis and machine learning problems, for example, the linear primal or nonlinear dual formulation of support vector machines [9], the LASSO approach to regularized least squares, and regularized logistic regression. Algorithms based on gradient and approximate / partial gradient information have proved effective in these settings. We mention in particular gradient projection and its accelerated variants [29], proximal gradient [45] and accelerated proximal gradient [4] methods for regularized objectives, and stochastic gradient methods [28, 38]. These methods are inherently serial, in that each iteration depends on the result of the previous iteration. Recently, parallel multicore versions of stochastic gradient and stochastic coordinate descent have been described for problems involving large data sets; see for example [31, 34, 3, 21, 39, 22].

This paper proposes an asynchronous stochastic proximal coordinate-descent algorithm, called ASYSPCD, for composite objective functions. The basic step of ASYSPCD, executed repeatedly by each core of a multicore system, is as follows: Choose an index $i \in \{1, 2, \dots, n\}$; read x from shared memory and evaluate the i th element of ∇f ; subtract a short, constant, positive of this partial gradient from

*Department of Computer Sciences, University of Wisconsin-Madison, 1210 W. Dayton St., Madison, WI 53706-1685, US (ji.liu.uwisc@gmail.edu). This author was supported in part by NSF Awards DMS-0914524 and DMS-1216318 and ONR Award N00014-13-1-0129.

†Department of Computer Sciences, University of Wisconsin-Madison, 1210 W. Dayton St., Madison, WI 53706-1685, US (swright@cs.wisc.edu). This author was supported in part by NSF Awards DMS-0914524, DMS-1216318, and IIS-1447449, ONR Award N00014-13-1-0129, AFOSR Award FA9550-13-1-0138, and Subcontract 3F-30222 from Argonne National Laboratory.

$(x)_i$; and perform a proximal operation on $(x)_i$ to account for the regularization term $g_i(\cdot)$. We use a simple model of computation that matches well to modern multicore architectures. Each core performs its updates on centrally stored vector x in an asynchronous, uncoordinated fashion, without any form of locking. A consequence of this model is that the version of x that is read by a core in order to evaluate its gradient is usually not the same as the version to which the update is made later, because x is updated in the interim by other cores. (Generally, we denote by \hat{x} the version of x that is used by a core to evaluate its component of $\nabla f(\hat{x})$.) We assume, however, that indefinite delays do not occur between reading and updating: There is a bound τ such no more than τ component-wise updates to x are missed by a core, between the time at which it reads the vector \hat{x} and the time at which it makes its update to the chosen element of x . A similar model of parallel asynchronous computation was used in HOGWILD! [31] and ASYSCD [21]. However, there is a key difference in this paper: *We do not assume that the evaluation vector \hat{x} is a version of x that actually existed in the shared memory at some point in time.* Rather, we account for the fact that the components of x may be updated by multiple cores while in the process of being read by another core, so that \hat{x} may be a “hybrid” version that never actually existed in memory. Our new model, which we call an “inconsistent read” model, is significantly closer to the reality of asynchronous computation, and dispenses with the somewhat unsatisfying “consistent read” assumption of previous work. It also requires a quite distinct style of analysis; our proofs differ substantially from those in previous related works.

We show that, for suitable choices of steplength, our algorithm converges at a linear rate if an “optimal strong convexity” property (1.2) holds. It attains sublinear convergence at a “ $1/k$ ” rate for general convex functions. Our analysis also defines a sufficient condition for near-linear speedup in the number of cores used. This condition relates the value of delay parameter τ (which corresponds closely to the number of cores / threads used in the computation) to the problem dimension n . A parameter that quantifies the cross-coordinate interactions in ∇f also appears in this relationship. When the Hessian of f is nearly diagonal, the minimization problem (1.1) is almost separable, so higher degrees of parallelism are possible.

We review related work in Section 2. Section 3 specifies the proposed algorithm. Convergence results are described in Section 4, with proofs given in the appendix. Computational experience is reported in Section 5. A summary and conclusions appear in Section 6.

Notation and Assumption. We use the following notation in the remainder of the paper.

- Ω denotes the intersection of $\text{dom}(f)$ and $\text{dom}(g)$
- S denotes the set on which F attains its optimal value, which is denoted by F^* .
- $\mathcal{P}_S(\cdot)$ denotes Euclidean-norm projection onto S .
- e_i denotes the i th natural basis vector in \mathbb{R}^n .
- Given a matrix A , we use $A_{\cdot j}$ to denote its j th column and A_i to denote its i th row.
- $\|\cdot\|$ denotes the Euclidean norm $\|\cdot\|_2$.
- $x_j \in \mathbb{R}^n$ denotes the j th iterate generated by the algorithm.
- $f_j^* := f(\mathcal{P}_S(x_j))$ and $g_j^* := g(\mathcal{P}_S(x_j))$.
- $F^* := F(\mathcal{P}_S(x))$ denotes the optimal objective value. (Note that $F^* = f_j^* + g_j^*$ for any j .)

- We use $(x)_i$ for the i th element of x , and $\nabla_i f(x)$ for the i th element of $\nabla f(x)$.
- Given a scalar function $h : \mathbb{R} \rightarrow \mathbb{R}$, define the componentwise proximal operator

$$\mathcal{P}_{i,h}(y) := \arg \min_x \frac{1}{2} \|x - y\|^2 + h((x)_i).$$

Similarly, for the vector function g , we denote

$$\mathcal{P}_g(y) := \arg \min_x \frac{1}{2} \|x - y\|^2 + g(x).$$

Note that the proximal operator is nonexpansive, that is, $\|\mathcal{P}_g(x) - \mathcal{P}_g(y)\| \leq \|x - y\|$.

We define the following *optimal strong convexity* condition for a convex function f with respect to the optimal set S , with parameter $l > 0$:

$$F(x) - F(\mathcal{P}_S(x)) \geq \frac{l}{2} \|x - \mathcal{P}_S(x)\|^2 \quad \forall x \in \Omega. \quad (1.2)$$

This condition is significantly weaker than the usual strong convexity condition; a strongly convex function $F(\cdot)$ is an optimally strongly convex function, but the converse is not true in general. We provide several examples of optimally strongly convex functions that are not strongly convex:

- $F(x) = \text{constant}$.
- $F(x) = f(Ax)$, where f is a strongly convex function and A is any matrix, possibly one with a nontrivial kernel.
- $F(x) = f(Ax) + \mathbf{1}_X(x)$ with strongly convex f , and arbitrary A , where $\mathbf{1}_X(x)$ is an indicator function defined on a polyhedron set X . Note first that $y^* := Ax^*$ is unique for any $x^* \in S$, from the strong convexity of f . The optimal solution set S is defined by

$$Ax = y^*, \quad x \in X.$$

The inequality (1.2) clearly holds for $x \notin X$, since the left-hand side is infinite in this case. For $x \in X$, we have by the famous theorem of Hoffman [19] that there exists $c > 0$ such that

$$\|Ax - y^*\|^2 = \|A(x - \mathcal{P}_S(x))\|^2 \geq c \|x - \mathcal{P}_S(x)\|^2.$$

Then from the strong convexity of $f(x)$, we have that there exists a positive number l such that for any $x \in X$

$$\begin{aligned} F(Ax) - F(A\mathcal{P}_S(x)) &= f(Ax) - f(A\mathcal{P}_S(x)) \\ &\geq \frac{l}{2} \|A(x - \mathcal{P}_S(x))\|^2 \geq \frac{lc}{2} \|x - \mathcal{P}_S(x)\|^2. \end{aligned}$$

- Squared hinge loss $F(x) = \sum_i \max(0, a_i^T x - y_i)^2$. To verify optimal strong convexity, we reformulate this problem as

$$\min_{t,x} \|t\|^2 \quad \text{subject to } t_i \geq a_i^T x - y_i \quad \forall i,$$

and apply the result just derived.

Note that optimal strong convexity (1.2) is a weaker version of the “essential strong convexity” condition used in [21]. A concept called “restricted strong convexity” proposed in [20] (See Lemma 4.6) is similar in that it requires a certain quantity to increase quadratically with distance from the solution set, but different in that the objective is assumed to be differentiable. Anitescu [2] defines a “quadratic growth condition” for (smooth) nonlinear programming in which the objective is assumed to grow at least quadratically with distance to a local solution in some feasible neighborhood of that solution. Since our setting (unconstrained, nonsmooth, convex) is quite different, we believe the use of a different term is warranted here.

Throughout this paper, we make the following assumption.

ASSUMPTION 1. *The solution set S of (1.1) is nonempty.*

Lipschitz Constants. We define two different Lipschitz constants L_{res} and L_{max} that are critical to the analysis, as follows. L_{res} is the *restricted Lipschitz constant* for ∇f along the coordinate directions: For any $x \in \Omega$, for any $i = 1, 2, \dots, n$, and any $t \in \mathbb{R}$ such that $x + te_i \in \Omega$, we have

$$\|\nabla f(x) - \nabla f(x + te_i)\| \leq L_{\text{res}}|t|.$$

The *coordinate Lipschitz constant* L_{max} is defined for x, i, t satisfying the same conditions as above:

$$\|\nabla f(x) - \nabla f(x + te_i)\|_{\infty} \leq L_{\text{max}}|t|.$$

Note that

$$f(x + te_i) - f(x) \leq \langle \nabla_i f(x), t \rangle + \frac{L_{\text{max}}}{2}t^2. \quad (1.3)$$

We denote the ratio between these two quantities by Λ :

$$\Lambda := L_{\text{res}}/L_{\text{max}}. \quad (1.4)$$

Making the implicit assumption that L_{res} and L_{max} are chosen to be the smallest values that satisfy their respective definitions, we have from standard relationships between the ℓ_2 and ℓ_{∞} norms that

$$1 \leq \Lambda \leq \sqrt{n}.$$

Besides bounding the nonlinearity of f along various directions, the quantities L_{res} and L_{max} capture the interactions between the various components in the gradient ∇f . In the case of twice continuously differentiable f , we can understand these interactions by observing the diagonal and off-diagonal terms of the Hessian $\nabla^2 f(x)$. Let us consider upper bounds on the ratio Λ in various situations. For simplicity, we suppose that f is quadratic with positive semidefinite Hessian Q .

- If Q is sparse with at most p nonzeros per row/column, we have that

$$L_{\text{res}} = \max_i \|Q_{\cdot i}\|_2 \leq \sqrt{p} \max_i \|Q_{\cdot i}\|_{\infty} = \sqrt{p} L_{\text{max}},$$

so that $\Lambda \leq \sqrt{p}$ in this situation.

- If Q is diagonally dominant, we have for any column i that

$$\|Q_{\cdot i}\|_2 \leq Q_{ii} + \|[Q_{ji}]_{j \neq i}\|_2 \leq Q_{ii} + \sum_{j \neq i} |Q_{ji}| \leq 2Q_{ii},$$

which, by taking the maximum of both sides, implies that $\Lambda \leq 2$ in this case.

- Suppose that $Q = A^T A$, where $A \in \mathbb{R}^{m \times n}$ is a random matrix whose entries are i.i.d from $\mathcal{N}(0, 1)$. (For example, f could be the linear least-squares objective $f(x) = \frac{1}{2} \|Ax - b\|^2$.) We show in [21] that Λ is upper-bounded roughly by $1 + \sqrt{n/m}$ in this case.

2. Related Work. We have surveyed related work on coordinate descent and stochastic gradient methods in a recent report [21]. Our discussion there included non-stochastic, cyclic coordinate descent methods [40, 24, 44, 5, 42, 43, 35], synchronous parallel methods that distribute the work of function and gradient evaluation [16, 25, 18, 7, 11, 1, 10, 37], and asynchronous parallel stochastic gradient methods (including the randomized Kaczmarz algorithm) [31, 22]. We make some additional comments here on related topics, and include some recent references from this active research area.

Stochastic coordinate descent can be viewed as a special case of stochastic gradient, so analysis of the latter approach can be applied, to obtain for example a sublinear $1/k$ rate of convergence in expectation for strongly convex functions; see, for example [28]. However, stochastic coordinate descent is “special” in that it is possible to guarantee improvement in the objective at every step. Nesterov [30] studied the convergence rate for a stochastic block coordinate descent method for unconstrained and separably constrained convex smooth optimization, proving linear convergence for the strongly convex case and a sublinear $1/k$ rate for the convex case. Richtárik and Takáč [33] and Lu and Xiao [23] extended this work to composite minimization, in which the objective is the sum of a smooth convex function and a separable nonsmooth convex function, and obtained similar (slightly stronger) convergence results. Stochastic coordinate descent is extended by Necoara and Patrascu [27] to convex optimization with a single linear constraint, randomly updating *two* coordinates at a time to maintain feasibility.

In the class of *synchronous parallel methods* for coordinate descent, Richtárik and Takáč [34] studied a synchronized parallel block (or minibatch) coordinate descent algorithm for composite optimization problems of the form (1.1), with a block separable regularizer g . At each iteration, processors update the randomly selected coordinates concurrently and synchronously. Speedup depends on the sparsity of the data matrix that defines the loss functions. A similar synchronous parallel method was studied in [26] and [8]; the latter focuses on the case of $g(x) = \|x\|_1$. Scherrer et al. [36] make greedy choices of multiple blocks of variables to update in parallel. Another greedy way of selecting coordinates was considered by Peng et al. [32], who also describe a parallel implementation of FISTA, an accelerated first-order algorithm due to Beck and Teboulle [4]. Fercoq and Richtárik [15] consider a variant of (1.1) in which f is allowed to be nonsmooth. They apply Nesterov’s smoothing scheme to obtain a smoothed version and update multiple blocks of coordinates using block coordinate descent in parallel. Sublinear convergence rate is established for both strongly convex and weakly convex cases. Fercoq and Richtárik [14] proposed a variant of Nesterov’s accelerated scheme to accelerate the synchronous parallel block coordinate algorithm of [34], proving an improved sublinear convergence rate for weakly convex problems. This variant avoids the disadvantage of the original Nesterov acceleration scheme [30], which requires $O(n)$ complexity per iteration, even on sparse data. Facchinei, Sagratella, and Scutari [13] consider a general framework for synchronous block coordinate descent methods with separable regularizers, in which the block subproblems may be solved inexactly. However, the block to be updated at each step is not chosen randomly; it must contain a component that is furthest from optimality, in some

sense.

We turn now to *asynchronous parallel methods*. Bertsekas and Tsitsiklis [6] described an asynchronous method for fixed-point problems $x = q(x)$ over a separable convex closed feasible region. (The optimization problem (1.1) can be formulated in this way by defining $q(x) := \mathcal{P}_{\alpha g}[(I - \alpha \nabla f)(x)]$ for a fixed $\alpha > 0$.) They use an inconsistent-read model of asynchronous computation, and establish linear convergence provided that components are not neglected indefinitely and that the iteration $x = q(x)$ is a maximum-norm contraction. The latter condition is quite strong. In the case of g null and f convex quadratic in (1.1) for instance, it requires the Hessian to satisfy a diagonal dominance condition — a stronger condition than strong convexity. By comparison, ASYSCD [21] guarantees linear convergence under an “essential strong convexity” condition, though it assumes a consistent-read model of asynchronous computation. Elsner et al. [12] considered the same fixed point problem and architecture as [6], and describe a similar scheme. Their scheme appears to require locking of the shared-memory data structure for x to ensure consistent reading and writing. Frommer and Szyld [17] give a comprehensive survey of asynchronous methods for solving fixed-point problems.

Liu et al. [21] followed the asynchronous consistent-read model of HOGWILD! to develop an asynchronous stochastic coordinate descent (ASYSCD) algorithm and proved sublinear ($1/k$) convergence on general convex functions and a linear convergence rate on functions that satisfy an “essential strong convexity” property. Sridhar et al. [39] developed an efficient LP solver by relaxing an LP problem into a bound-constrained QP problem, which is then solved by ASYSCD.

Liu et al. [22] developed an asynchronous parallel variant of the randomized Kaczmarz algorithm for solving a general consistent linear system $Ax = b$, proving a linear convergence rate. Avron et al. [3] proposed an asynchronous solver for the system $Qx = c$ where Q is a symmetric positive definite matrix, proving a linear convergence rate. This method is essentially an asynchronous stochastic coordinate descent method applied to the strongly convex quadratic optimization problem $\min_x \frac{1}{2}x^T Qx - c^T x$. The paper considers both inconsistent- and consistent-read cases are considered, with slightly different convergence results.

3. Algorithm. In our algorithm ASYSPCD, multiple processors have access to a shared data structure for the vector x , and each processor is able to compute a randomly chosen element of the gradient vector $\nabla f(x)$. Each processor repeatedly runs the following proximal coordinate descent process. (Choice of the steplength parameter γ is discussed further in the next section.)

- R: Choose an index $i \in \{1, 2, \dots, n\}$ at random, read x into the local storage location \hat{x} , and evaluate $\nabla_i f(\hat{x})$;
- U: Update component i of the shared x by taking a step of length γ/L_{\max} in the direction $-\nabla_i f(\hat{x})$, follows by a proximal operation defined as follows:¹

$$x \leftarrow \mathcal{P}_{i, \frac{\gamma}{L_{\max}} g_i} \left(x - \frac{\gamma}{L_{\max}} e_i \nabla_i f(\hat{x}) \right).$$

¹Our analysis assumes that no other process modifies x_i while this proximal operation is being computed. As we explain in Section 5, our practical implementation actually assigns each coordinate x_i to a single core, and allows only that core to update x_i , so this issue does not arise. An alternative implementation, pointed out by a referee, would be to use a “compare-and-swap” atomic instruction to implement the update. This operation would perform the update only if x_i was not changed while the update was being computed.

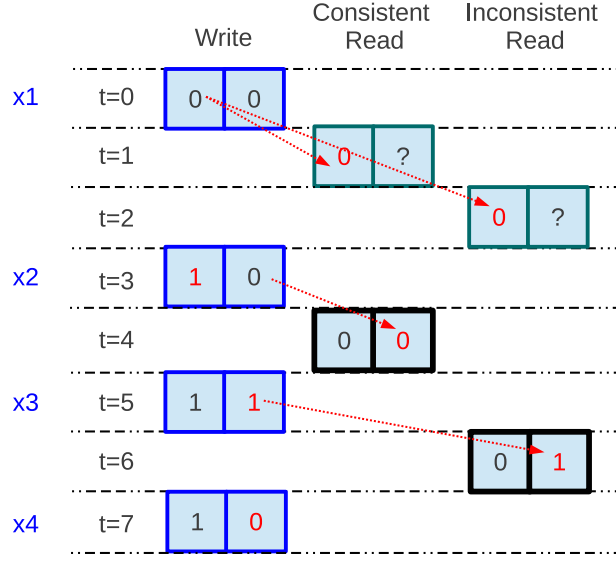


Fig. 3.1: Time sequence of writes and reads of a two-variable vector, showing instances of consistent and inconsistent reading. The left column shows the initial vector at time 0, stored in shared memory, with updates to single components at times 3, 5, and 7. The middle column shows a consistent read, in which the first component is read at time 1 and the second component is read at time 4. The read vector is equal to the shared-memory vector at time 0. The right column shows an inconsistent read, in which the first component is read at time 2 and the second component is read at time 6. Because of intervening writes to these components, the read vector does not match the versions that appeared in shared memory at any time point.

Notice that each step changes just a single element of x , that is, the i th element. Unlike standard proximal coordinate descent, the value \hat{x} at which the coordinate gradient is calculated usually differs from the value of x to which the update is applied, because while the processor is evaluating its gradient, other processors may repeatedly update the value of x stored in memory. As mentioned above, we use an “inconsistent read” model of asynchronous computation here, in contrast to the “consistent read” models of ASYSCD [21] and HOGWILD! [31]. Figure 3.1 shows how inconsistent reading can occur, as a result of updating of components of x while it is being read. Consistent reading can be guaranteed by means of a software lock, but such a mechanism degrades parallel performance significantly. In fact, the implementations of HOGWILD! and ASYSCD described in the papers [31, 21] do not use any software lock, and in this respect the computations in those papers are not quite compatible with their analysis.

The “global” view of algorithm ASYSPCD is shown in **Algorithm 1**. To obtain this version from the “local” version, we introduce a counter j to track the total number of updates applied to x , so that x_j is the state of x in memory after update j is performed. We use $i(j)$ to denote the component that is updated at iteration j , and \hat{x}_j for value of x that is used in the calculation of the gradient element $\nabla f_{i(j)}$. The components of \hat{x}_j may have different ages. Some components may be current at iteration j , others may not reflect recent updates made by other processors. We

Algorithm 1 Asynchronous Stochastic Coordinate Descent Algorithm $x_J = \text{ASYSPCD}(x_0, \gamma, J)$

Require: x_0 , γ , and J

Ensure: x_J

- 1: Initialize $j \leftarrow 0$;
 - 2: **while** $j < J$ **do**
 - 3: Choose $i(j)$ from $\{1, 2, \dots, n\}$ with equal probability;
 - 4: $x_{j+1} \leftarrow \mathcal{P}_{i(j), \frac{\gamma}{L_{\max}} g_{i(j)}} \left(x_j - \frac{\gamma}{L_{\max}} e_{i(j)} \nabla_{i(j)} f(\hat{x}_j) \right)$;
 - 5: $j \leftarrow j + 1$;
 - 6: **end while**
-

assume however that there is an upper bound of τ on the age of each component, measured in terms of updates. $K(j)$ defines an iterate set such that

$$x_j = \hat{x}_j + \sum_{d \in K(j)} (x_{d+1} - x_d).$$

One can see that $d \leq j - 1, \forall d \in K(j)$. Here we assume τ to be the upper bound on the age of all elements in $K(j)$, for all j , so that $\tau \geq j - \min\{d \mid d \in K(j)\}$. We assume further that $K(j)$ is ordered from oldest to newest index (that is, smallest to largest). Note that $K(j)$ is empty if $x_j = \hat{x}_j$, that is, if the step is simply an ordinary stochastic coordinate gradient update. The value of τ corresponds closely to the number of cores involved in the computation provided that computation of the update for each component of x costs roughly the same.

4. Main Results. This section presents results on convergence of ASYSPCD. The theorem encompasses both the linear rate for optimally strongly convex f and the sublinear rate for general convex f . The result depends strongly on the delay parameter τ . The proofs are highly technical, and are relegated to Appendix A. We note the proof techniques differ significantly from those used for the consistent-read algorithms of [31] and [21].

We start by describing the key idea of the algorithm, which is reflected in the way that it chooses the steplength parameter γ . Denoting \bar{x}_{j+1} by

$$\bar{x}_{j+1} := \mathcal{P}_{\frac{\gamma}{L_{\max}} g} \left(x_j - \frac{\gamma}{L_{\max}} \nabla f(\hat{x}_j) \right), \quad (4.1)$$

we can see that

$$(x_{j+1})_{i(j)} = (\bar{x}_{j+1})_{i(j)}, \quad (x_{j+1})_i = (x_j)_i \text{ for } i \neq i(j), \quad (4.2)$$

so that $x_{j+1} - x_j = [(\bar{x}_{j+1})_{i(j)} - (x_j)_{i(j)}]e_{i(j)}$. Thus, we have

$$\mathbb{E}_{i(j)}(x_{j+1} - x_j) = \frac{1}{n} \sum_{i=1}^n [(\bar{x}_{j+1})_i - (x_j)_i]e_i = \frac{1}{n} [\bar{x}_{j+1} - x_j].$$

Therefore, we can view $\bar{x}_{j+1} - x_j$ as capturing the expected behavior of $x_{j+1} - x_j$. Note that when $g(x) = 0$, we have $\bar{x}_{j+1} - x_j = -(\gamma/L_{\max})\nabla f(\hat{x}_j)$, a standard negative-gradient step. The choice of steplength parameter γ entails a tradeoff: We would like γ to be long enough that significant progress is made at each step, but not so long

that the gradient information computed at \hat{x}_j is stale and irrelevant by the time the update is applied to x_j . We enforce this tradeoff by means of a bound on the ratio of expected squared norms on $x_j - \bar{x}_{j+1}$ at successive iterates; specifically,

$$\mathbb{E}\|x_{j-1} - \bar{x}_j\|^2 \leq \rho \mathbb{E}\|x_j - \bar{x}_{j+1}\|^2, \quad (4.3)$$

where $\rho > 1$ is a user defined parameter. The analysis becomes a delicate balancing act in the choice of ρ and steplength γ between aggression and excessive conservatism. We find, however, that these values can be chosen to ensure steady convergence for the asynchronous method at a linear rate, with rate constants that are almost consistent with a standard short-step proximal full-gradient descent, when the optimal strong convexity condition (1.2) is satisfied.

Our main convergence result is the following.

THEOREM 4.1. *Suppose that Assumption 1 is satisfied. Let ρ be a constant that satisfies $\rho > 1 + 4/\sqrt{n}$, and define the quantities θ , θ' , and ψ as follows:*

$$\theta := \frac{\rho^{(\tau+1)/2} - \rho^{1/2}}{\rho^{1/2} - 1}, \quad \theta' := \frac{\rho^{(\tau+1)} - \rho}{\rho - 1}, \quad \psi := 1 + \frac{\tau\theta'}{n} + \frac{2\Lambda\theta}{\sqrt{n}}. \quad (4.4)$$

Suppose that the steplength parameter $\gamma > 0$ satisfies the following two bounds:

$$\gamma \leq \frac{1}{\psi}, \quad \gamma \leq \frac{\sqrt{n}(1 - \rho^{-1}) - 4}{4(1 + \theta)\Lambda}. \quad (4.5)$$

Then we have

$$\mathbb{E}\|x_{j-1} - \bar{x}_j\|^2 \leq \rho \mathbb{E}\|x_j - \bar{x}_{j+1}\|^2, \quad j = 1, 2, \dots \quad (4.6)$$

If the optimal strong convexity property (1.2) holds with $l > 0$, we have for $j = 1, 2, \dots$ that

$$\begin{aligned} & \mathbb{E}\|x_j - \mathcal{P}_S(x_j)\|^2 + \frac{2\gamma}{L_{\max}}(\mathbb{E}F(x_j) - F^*) \\ & \leq \left(1 - \frac{l}{n(l + \gamma^{-1}L_{\max})}\right)^j \left(\|x_0 - \mathcal{P}_S(x_0)\|^2 + \frac{2\gamma}{L_{\max}}(F(x_0) - F^*)\right), \end{aligned} \quad (4.7)$$

while for general smooth convex function f , we have

$$\mathbb{E}F(x_j) - F^* \leq \frac{n(\|x_0 - \mathcal{P}_S(x_0)\|^2 L_{\max} + 2\gamma(F(x_0) - F^*))}{2\gamma(n + j)}. \quad (4.8)$$

The following corollary proposes an interesting particular choice for the parameters for which the convergence expressions become more comprehensible. The result requires a condition on the delay bound τ in terms of n and the ratio Λ .

COROLLARY 4.2. *Suppose that Assumption 1 holds and that*

$$4e\Lambda(\tau + 1)^2 \leq \sqrt{n}. \quad (4.9)$$

If we choose

$$\rho = \left(1 + \frac{4e\Lambda(\tau + 1)}{\sqrt{n}}\right)^2, \quad (4.10)$$

then the steplength $\gamma = 1/2$ will satisfy the bounds (4.5). In addition, when the optimal strong convexity property (1.2) holds with $l > 0$, we have for $j = 1, 2, \dots$ that

$$\mathbb{E}F(x_j) - F^* \leq \left(1 - \frac{l}{n(l + 2L_{\max})}\right)^j (L_{\max}\|x_0 - \mathcal{P}_S(x_0)\|^2 + F(x_0) - F^*), \quad (4.11)$$

while for the case of general convex f , we have

$$\mathbb{E}F(x_j) - F^* \leq \frac{n(L_{\max}\|x_0 - \mathcal{P}_S(x_0)\|^2 + F(x_0) - F^*)}{j + n}. \quad (4.12)$$

We note that the linear rate (4.11) is broadly consistent with the linear rate for the classical steepest descent method applied to strongly convex functions, which has a rate constant of $(1 - 2l/L)$, where L is the standard Lipschitz constant for ∇f . Suppose we assume (not unreasonably) that n steps of stochastic coordinate descent cost roughly the same as one step of steepest descent, and that $l \leq L_{\max}$. It follows from (4.11) that n steps of stochastic coordinate descent would achieve a reduction factor of about

$$1 - \frac{l}{2L_{\max} + l} \leq 1 - \frac{l}{3L_{\max}},$$

so a standard argument would suggest that stochastic coordinate descent would require about $6L_{\max}/L$ times more computation. Since $L_{\max}/L \in [1/n, 1]$, the stochastic asynchronous approach may actually require less computation. It may also gain an advantage from the parallel asynchronous implementation. A parallel implementation of standard gradient descent would require synchronization and careful division of the work of evaluating ∇f , whereas the stochastic approach can be implemented in an asynchronous fashion.

For the general convex case, (4.12) defines a sublinear rate, whose relationship with the rate of standard gradient descent for general convex optimization is similar to the previous paragraph.

Note that the results in Theorem 4.1 and Corollary 4.2 are consistent with the analysis for constrained ASySCD in [21], but this paper considers the more general case of composite optimization and the inconsistent-read model of parallel computation.

As noted in Section 1, the parameter τ corresponds closely to the number of cores that can be involved in the computation, since if all cores are working at the same rate, we would expect each other core to make one update between the times at which x is read and (later) updated. If τ is small enough that (4.9) holds, the analysis indicates that near-linear speedup in the number of processors is achievable. A small value for the ratio Λ (not much greater than 1) implies a greater degree of potential parallelism. As we note at the end of Section 1, this ratio tends to closer to 1 than to \sqrt{n} in some important applications. In these situations, the bound (4.9) indicates that τ can vary like $n^{1/4}$ without affecting the iteration-wise convergence rate, and yielding near-linear speedup in the number of cores. This quantity is consistent with the analysis for constrained ASySCD in [21] but weaker than the unconstrained ASySCD (which allows the maximal number of cores being $O(n^{1/2})$). A further comparison is with the asynchronous randomized Kaczmarz algorithm [22] which allows $O(m)$ cores to be used efficiently when solving a consistent sparse linear system.

We conclude this section with a high-probability bound. The result follows immediately from Markov’s inequality. See Theorem 3 in [21] for a related result and complete proof.

THEOREM 4.3. *Suppose that the conditions of Corollary 4.2 hold, including the choice of ρ . Then for $\epsilon > 0$ and $\eta \in (0, 1)$, we have that*

$$\mathbb{P}(F(x_j) - F^* \leq \epsilon) \geq 1 - \eta, \quad (4.13)$$

provided that one of the following conditions holds. In the optimally strongly convex case (1.2) with $l > 0$, we require

$$j \geq \frac{n(l + 2L_{\max})}{l} \left\lceil \log \frac{L_{\max}\|x_0 - \mathcal{P}_S(x_0)\|^2 + F(x_0) - F^*}{\epsilon\eta} \right\rceil,$$

iterations, while in the general convex case, it suffices that

$$j \geq \frac{n(L_{\max}\|x_0 - \mathcal{P}_S(x_0)\|^2 + F(x_0) - F^*)}{\epsilon\eta} - n.$$

5. Experiments. This section presents some results to illustrate the effectiveness of ASYSPCD, in particular, the fact that near-linear speedup can be observed on a multicore machine. We note that more comprehensive experiments can be found in [21] and [39], for unconstrained and box-constrained problems. Although the analysis in [21] assumes consistent read, it is not enforced in the implementation, so apart from the fact that we now include a prox-step to account for the regularization term, the implementations in [21] and [39] are quite similar to the one employed in this section.

We apply our code for ASYSPCD to the following “ ℓ_2 - ℓ_1 ” problem:

$$\min_x \frac{1}{2}\|Ax - b\|^2 + \lambda\|x\|_1 \equiv \frac{1}{2}x^T A^T A x - b^T A x + \frac{1}{2}b^T b + \lambda\|x\|_1.$$

The elements of $A \in \mathbb{R}^{m \times n}$ are selected i.i.d. from a Gaussian $\mathcal{N}(0, 1)$ distribution. To construct a sparse true solution $x^* \in \mathbb{R}^n$, given the dimension n and sparsity s , we select s entries of x^* at random to be nonzero and $\mathcal{N}(0, 1)$ normally distributed, and set the rest to zero. The measurement vector $b \in \mathbb{R}^m$ is obtained by $b = Ax^* + \epsilon$, where elements of the noise vector $\epsilon \in \mathbb{R}^m$ are i.i.d. $\mathcal{N}(0, \sigma^2)$, where the value of σ controls the signal-to-noise ratio.

Our experiments run on 1 to 10 threads of an Intel Xeon machine, with all threads sharing a single memory socket. Our implementations deviate modestly from the version of ASYSPCD described in Section 3. We compute $Q := A^T A \in \mathbb{R}^{n \times n}$ and $c := A^T b \in \mathbb{R}^n$ offline. Q and c are partitioned into slices (row submatrices) and subvectors (respectively) of equal size, and each thread is assigned one submatrix from Q and the corresponding subvector from c . During the algorithm, each thread updates the elements of x corresponding to its slice of Q , in order. After one scan, or “epoch” is complete, it reorders the indices randomly, then repeats the process. This scheme essentially changes the scheme from sampling with replacement (as analyzed) to sampling without replacement, which has demonstrated empirically better performance on many related problems. (The same advantage is noted in the implementations of HOGWILD! [31].)

We choose $\sigma = 0.01$ with $m = 6000$, $n = 10000$, and $s = 10$ in Figure 5.1 and $m = 12000$, $n = 20000$, and $s = 20$ in Figure 5.2. We set $\lambda = 20\sqrt{m \log(n)}\sigma$ (a

value of the order of $\sqrt{m \log(n)}\sigma$ is suggested by compressed sensing theory) and the steplength γ is set as 1 in both figures. In both cases, we can estimate the ratio $\Lambda = L_{\text{res}}/L_{\text{max}}$ roughly by $1 + \sqrt{n/m} \approx 2.3$, as suggested at the end of Section 1. Our final computed values of x have nonzeros in the same locations as the chosen solution x^* , though the values differ, because of the noise in b .

The left-hand graph in each figure indicates the number of threads / cores and plots objective function value vs epoch count, where one epoch is equivalent to n iterations. Note that the curves are almost overlaid, indicating that the total workload required for AsySPCD is almost independent of the number of cores used in the computation. This observation validates our result in Corollary 4.2, which indicates that provided τ is below a certain threshold, it does not seriously affect the rate of convergence, as a function of total computation performed. The right-hand graph in each figure shows speedup when executed on different numbers of cores. Near-linear speedup is observed in Figure 5.2, while there is a slight dropoff for the larger numbers of cores in Figure 5.1. The difference can be explain by the smaller dimension of the problem illustrated in Figure 5.1. Referring to our threshold value (4.9) that indicates dimensions above which linear speedup should be expected, we have by setting $\Lambda \approx 2.3$ (as discussed above) and $\tau = 10$ (the maximum number of threads used in this experiment) that the left-hand side of (4.9) is approximately 3000, while the right-hand side is 100 (for Figure 5.1) and approximately 141 (for Figure 5.1). As expected, our analysis is quite conservative; near-linear speedup is observed even when the threshold (4.9) is violated significantly.

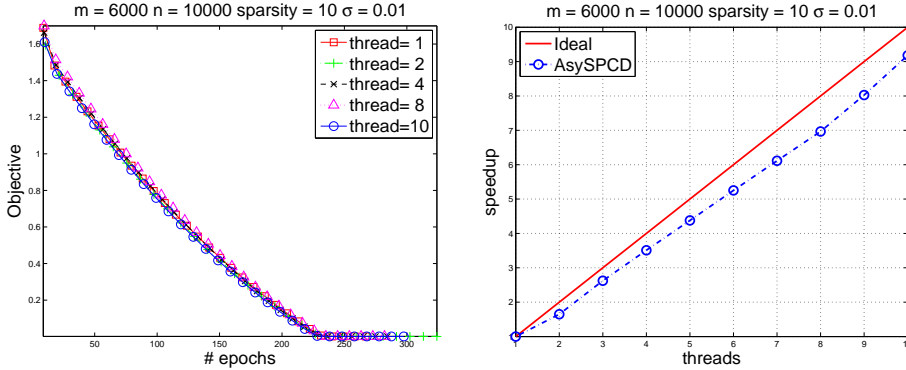


Fig. 5.1: The left graph plots objective function vs epochs for 1, 2, 4, 8, and 10 cores. The right graph shows speedup obtained for implementation on 1-10 cores, plotted against the ideal (linear) speedup.

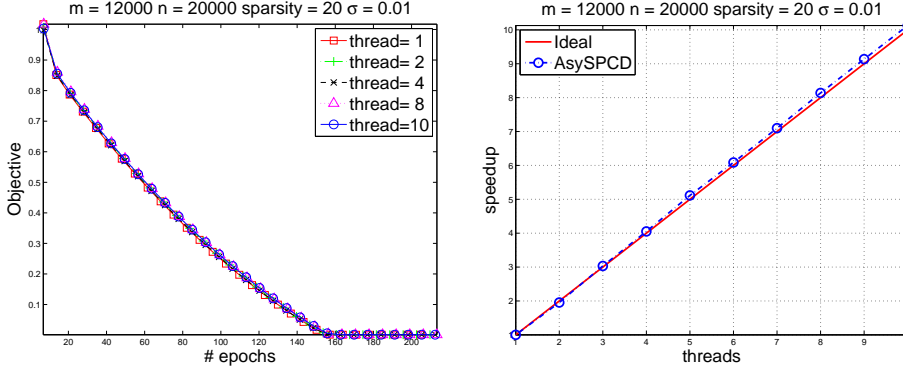


Fig. 5.2: The left graph plots objective function vs epochs for 1, 2, 4, 8, and 10 cores. The right graph shows speedup obtained for implementation on 1-10 cores, plotted against the ideal (linear) speedup.

6. Conclusions. This paper proposes an asynchronous parallel proximal stochastic coordinate descent algorithm for minimizing composite objectives of the form (1.1). Sublinear convergence (at rate $1/k$) is proved for general convex functions, with stronger linear convergence results for problems that satisfy the optimal strong convexity property (1.2). Our analysis indicates the extent to which parallel implementations can be expected to yield near-linear speedup, in terms of a parameter that quantifies the cross-coordinate interactions in the gradient ∇f and a parameter τ that bounds the delay in updating. Our computational experience confirms that the linear speedup properties suggested by the analysis can be observed in practice.

Acknowledgments. The authors thank the editor and both referees for their valuable comments. Special thanks to Dr. Yijun Huang for her implementation of AsySPCD, which was used here to obtain computational results.

Appendix A. Proofs of Main Results.

This section provides the proofs for the main convergence results. We start with some preliminaries, then proceed to proofs of Theorem 4.1 and Corollary 4.2.

A.1. Preliminaries. Note that the component indices $i(0), i(1), \dots, i(j), \dots$ in Algorithm 1 are independent random variables. We use \mathbb{E} to denote the expectation over all random variables, and $\mathbb{E}_{i(j)}$ to denote the conditional expectation in term of $i(j)$ given $i(0), i(1), \dots, i(j-1)$. We also denote

$$(\Delta_j)_{i(j)} := (x_j - x_{j+1})_{i(j)}, \quad (\text{A.1})$$

and formulate the update in Step 4 of Algorithm 1 in the following way:

$$x_{j+1} = \arg \min_x \langle \nabla_{i(j)} f(\hat{x}_j), (x - x_j)_{i(j)} \rangle + \frac{L_{\max}}{2\gamma} \|x - x_j\|^2 + g_{i(j)}((x)_{i(j)}).$$

(Note that $(x_{j+1})_i = (x_j)_i$ for $i \neq i(j)$.) From the optimality condition for this formulation (see (41) in [41]), we have for all x that

$$\left\langle (x - x_{j+1})_{i(j)}, \nabla_{i(j)} f(\hat{x}_j) - \frac{L_{\max}}{\gamma} (\Delta_j)_{i(j)} \right\rangle + g_{i(j)}((x)_{i(j)}) - g_{i(j)}((x_{j+1})_{i(j)}) \geq 0.$$

By rearranging this expression and substituting $\mathcal{P}_S(x)$ for x , we find that the following inequality is true for all x :

$$\begin{aligned} g_{i(j)}((\mathcal{P}_S(x))_{i(j)}) - g_{i(j)}((x_{j+1})_{i(j)}) + \langle (\mathcal{P}_S(x) - x_{j+1})_{i(j)}, \nabla_{i(j)} f(\hat{x}_j) \rangle \\ \geq \frac{L_{\max}}{\gamma} \langle (\mathcal{P}_S(x) - x_{j+1})_{i(j)}, (\Delta_j)_{i(j)} \rangle. \end{aligned} \quad (\text{A.2})$$

From the definition of L_{\max} , and using the notation (A.1), we have

$$f(x_{j+1}) \leq f(x_j) + \langle \nabla_{i(j)} f(x_j), -(\Delta_j)_{i(j)} \rangle + \frac{L_{\max}}{2} |(\Delta_j)_{i(j)}|^2,$$

or equivalently,

$$\langle \nabla_{i(j)} f(x_j), (\Delta_j)_{i(j)} \rangle \leq f(x_j) - f(x_{j+1}) + \frac{L_{\max}}{2} |(\Delta_j)_{i(j)}|^2. \quad (\text{A.3})$$

From the definition of \bar{x}_{j+1} in (4.1), we have

$$\bar{x}_{j+1} = \arg \min_x \langle \nabla f(\hat{x}_j), x - x_j \rangle + \frac{L_{\max}}{2\gamma} \|x - x_j\|^2 + g(x),$$

so, using (41) from [41] again, we have

$$g(x) - g(\bar{x}_{j+1}) + \left\langle x - \bar{x}_{j+1}, \nabla f(\hat{x}_j) + \frac{L_{\max}}{\gamma} (\bar{x}_{j+1} - x_j) \right\rangle \geq 0, \quad \forall x. \quad (\text{A.4})$$

We now define

$$\Delta_j := x_j - \bar{x}_{j+1}, \quad (\text{A.5})$$

and note that this definition is consistent with $(\Delta_j)_{i(j)}$ defined in (A.1). From (4.2), we have

$$\mathbb{E}_{i(j)}(\|x_{j+1} - x_j\|^2) = \frac{1}{n} \|\bar{x}_{j+1} - x_j\|^2. \quad (\text{A.6})$$

Recalling that the indices in $K(j)$ are sorted in the increasing order from smallest (oldest) iterate to largest (newest) iterate, we use $K(j)_t$ to denote the t -th smallest entry in $K(j)$. For $T = 0, 1, \dots, |K(j)|$, we define

$$\hat{x}_{j,T} := \hat{x}_j + \sum_{t=1}^T (x_{K(j)_t+1} - x_{K(j)_t}).$$

We have the following relations:

$$\begin{aligned} \hat{x}_j &= \hat{x}_{j,0} \\ x_j &= \hat{x}_{j,|K(j)|} \\ x_j - \hat{x}_j &= \sum_{t=0}^{|K(j)|-1} (\hat{x}_{j,t+1} - \hat{x}_{j,t}) \\ \nabla f(x_j) - \nabla f(\hat{x}_j) &= \sum_{t=0}^{|K(j)|-1} (\nabla f(\hat{x}_{j,t+1}) - \nabla f(\hat{x}_{j,t})). \end{aligned}$$

Furthermore, we have

$$\begin{aligned}
& \|\nabla f(x_j) - \nabla f(\hat{x}_j)\| \\
&= \left\| \sum_{t=0}^{|K(j)|-1} (\nabla f(\hat{x}_{j,t}) - \nabla f(\hat{x}_{j,t+1})) \right\| \\
&\leq \sum_{t=0}^{|K(j)|-1} \|\nabla f(\hat{x}_{j,t}) - \nabla f(\hat{x}_{j,t+1})\| \\
&\leq L_{\text{res}} \sum_{t=0}^{|K(j)|-1} \|\hat{x}_{j,t} - \hat{x}_{j,t+1}\| \\
&= L_{\text{res}} \sum_{t=1}^{|K(j)|} \|x_{K(j)_t} - x_{K(j)_{t+1}}\| \\
&= L_{\text{res}} \sum_{d \in K(j)} \|x_{d+1} - x_d\|, \tag{A.7}
\end{aligned}$$

where the second inequality holds because $\hat{x}_{j,t}$ and $\hat{x}_{j,t+1}$ differ in only a single coordinate.

A.2. Proof of Theorem 4.1. *Proof.* We prove (4.6) by induction. First, note that for any vectors a and b , we have

$$\begin{aligned}
\|a\|^2 - \|b\|^2 &= 2\|a\|^2 - (\|a\|^2 + \|b\|^2) \\
&\leq 2\|a\|^2 - 2\langle a, b \rangle \\
&= 2\langle a, a - b \rangle \\
&\leq 2\|a\|\|b - a\|.
\end{aligned}$$

Thus for all j , we have

$$\begin{aligned}
& \|x_{j-1} - \bar{x}_j\|^2 - \|x_j - \bar{x}_{j+1}\|^2 \\
&\leq 2\|x_{j-1} - \bar{x}_j\|\|x_j - \bar{x}_{j+1} - x_{j-1} + \bar{x}_j\|. \tag{A.8}
\end{aligned}$$

The second factor in the r.h.s. of (A.8) is bounded as follows:

$$\begin{aligned}
& \|x_j - \bar{x}_{j+1} - x_{j-1} + \bar{x}_j\| \\
&= \left\| x_j - \mathcal{P}_{\frac{\gamma}{L_{\max}}g} \left(x_j - \frac{\gamma}{L_{\max}} \nabla f(\hat{x}_j) \right) - \right. \\
&\quad \left. \left(x_{j-1} - \mathcal{P}_{\frac{\gamma}{L_{\max}}g} \left(x_{j-1} - \frac{\gamma}{L_{\max}} \nabla f(\hat{x}_{j-1}) \right) \right) \right\| \\
&\leq \|x_j - x_{j-1}\| + \\
&\quad \left\| \mathcal{P}_{\frac{\gamma}{L_{\max}}g} \left(x_j - \frac{\gamma}{L_{\max}} \nabla f(\hat{x}_j) \right) - \mathcal{P}_{\frac{\gamma}{L_{\max}}g} \left(x_{j-1} - \frac{\gamma}{L_{\max}} \nabla f(\hat{x}_{j-1}) \right) \right\| \\
&\leq 2\|x_j - x_{j-1}\| + \frac{\gamma}{L_{\max}} \|\nabla f(\hat{x}_j) - \nabla f(\hat{x}_{j-1})\| \\
&\quad \left(\text{by the nonexpansive property of } \mathcal{P}_{\frac{\gamma}{L_{\max}}g} \right) \\
&= 2\|x_j - x_{j-1}\| + \frac{\gamma}{L_{\max}} \|\nabla f(\hat{x}_j) - \nabla f(x_j) + \nabla f(x_j) - \nabla f(x_{j-1}) \\
&\quad + \nabla f(x_{j-1}) - \nabla f(\hat{x}_{j-1})\| \\
&\leq 2\|x_j - x_{j-1}\| + \frac{\gamma}{L_{\max}} (\|\nabla f(\hat{x}_j) - \nabla f(x_j)\| + \|\nabla f(x_j) - \nabla f(x_{j-1})\| \\
&\quad + \|\nabla f(x_{j-1}) - \nabla f(\hat{x}_{j-1})\|) \\
&\leq (2 + \Lambda\gamma) \|x_j - x_{j-1}\| + \frac{\gamma}{L_{\max}} \|\nabla f(\hat{x}_j) - \nabla f(x_j)\| \\
&\quad + \frac{\gamma}{L_{\max}} \|\nabla f(x_{j-1}) - \nabla f(\hat{x}_{j-1})\| \\
&\leq (2 + \Lambda\gamma) \|x_j - x_{j-1}\| + \Lambda\gamma \sum_{d \in K(j)} \|x_d - x_{d+1}\| \\
&\quad + \Lambda\gamma \sum_{d \in K(j-1)} \|x_d - x_{d+1}\| \quad (\text{from (A.7)}) \tag{A.9} \\
&\leq (2 + \Lambda\gamma) \|x_j - x_{j-1}\| + \Lambda\gamma \sum_{d=j-\tau}^{j-1} \|x_d - x_{d+1}\| + \Lambda\gamma \sum_{d=j-1-\tau}^{j-2} \|x_d - x_{d+1}\| \\
&\leq (2 + 2\Lambda\gamma) \|x_j - x_{j-1}\| + 2\Lambda\gamma \sum_{d=j-1-\tau}^{j-2} \|x_d - x_{d+1}\|, \tag{A.10}
\end{aligned}$$

where the fourth inequality uses $\|\nabla f(x_j) - \nabla f(x_{j-1})\| \leq L_{\text{res}} \|x_j - x_{j-1}\|$, since x_j and x_{j-1} differ in just one component.

We set $j = 1$, and note that $K(0) = \emptyset$ and $K(1) \subset \{0\}$. In this case, we obtain a bound from (A.9)

$$\|x_1 - \bar{x}_2 + x_0 - \bar{x}_1\| \leq (2 + \Lambda\gamma) \|x_1 - x_0\| + \Lambda\gamma \|x_1 - x_0\| = (2 + 2\Lambda\gamma) \|x_1 - x_0\|.$$

By substituting this bound in (A.8) and setting $j = 1$, and taking expectations, we obtain

$$\begin{aligned}
\mathbb{E}(\|x_0 - \bar{x}_1\|^2) - \mathbb{E}(\|x_1 - \bar{x}_2\|^2) &\leq 2\mathbb{E}(\|x_0 - \bar{x}_1\| \|x_1 - \bar{x}_2 - x_0 + \bar{x}_1\|) \\
&\leq (4 + 4\Lambda\gamma) \mathbb{E}(\|\bar{x}_1 - x_0\| \|x_1 - x_0\|). \tag{A.11}
\end{aligned}$$

For any positive scalars μ_1 , μ_2 , and α , we have

$$\mu_1\mu_2 \leq \frac{1}{2}(\alpha\mu_1^2 + \alpha^{-1}\mu_2^2). \quad (\text{A.12})$$

It follows that

$$\begin{aligned} \mathbb{E}(\|x_j - x_{j-1}\| \|\bar{x}_j - x_{j-1}\|) &\leq \frac{1}{2} \mathbb{E}(n^{1/2}\|x_j - x_{j-1}\|^2 + n^{-1/2}\|\bar{x}_j - x_{j-1}\|^2) \\ &= \frac{1}{2} \mathbb{E}(n^{1/2} \mathbb{E}_{i(j-1)}(\|x_j - x_{j-1}\|^2) + n^{-1/2}\|\bar{x}_j - x_{j-1}\|^2) \\ &= \frac{1}{2} \mathbb{E} \left(n^{-1/2}\|\bar{x}_j - x_{j-1}\|^2 + n^{-1/2}\|\bar{x}_j - x_{j-1}\|^2 \right) \quad (\text{from (A.6)}) \\ &= n^{-1/2} \mathbb{E} \|\bar{x}_j - x_{j-1}\|^2. \end{aligned} \quad (\text{A.13})$$

By taking $j = 1$ in (A.13), and substituting in (A.11), we obtain

$$\mathbb{E}(\|x_0 - \bar{x}_1\|^2) - \mathbb{E}(\|x_1 - \bar{x}_2\|^2) \leq n^{-1/2} (4 + 4\Lambda\gamma) \mathbb{E} \|\bar{x}_1 - x_0\|^2,$$

which implies that

$$\mathbb{E}(\|x_0 - \bar{x}_1\|^2) \leq \left(1 - \frac{4 + 4\gamma\Lambda}{\sqrt{n}} \right)^{-1} \mathbb{E}(\|x_1 - \bar{x}_2\|^2) \leq \rho \mathbb{E}(\|x_1 - \bar{x}_2\|^2).$$

To see the last inequality, one only needs to verify that

$$\rho^{-1} \leq 1 - \frac{4 + 4\gamma\Lambda}{\sqrt{n}} \Leftrightarrow \gamma \leq \frac{\sqrt{n}(1 - \rho^{-1}) - 4}{4\Lambda},$$

where the last inequality follows from the second bound for γ in (4.5). We have thus shown that (4.6) holds for $j = 1$.

To take the inductive step, we assume that (4.6) holds up to index $j - 1$. We have for $j - 1 - \tau \leq d \leq j - 2$ and any $\beta > 0$ (using (A.12) again) that

$$\begin{aligned} &\mathbb{E}(\|x_d - x_{d+1}\| \|\bar{x}_j - x_{j-1}\|) \\ &\leq \frac{1}{2} \mathbb{E}(n^{1/2}\beta\|x_d - x_{d+1}\|^2 + n^{-1/2}\beta^{-1}\|\bar{x}_j - x_{j-1}\|^2) \\ &= \frac{1}{2} \mathbb{E}(n^{1/2}\beta \mathbb{E}_{i(d)}(\|x_d - x_{d+1}\|^2) + n^{-1/2}\beta^{-1}\|\bar{x}_j - x_{j-1}\|^2) \\ &= \frac{1}{2} \mathbb{E}(n^{-1/2}\beta\|x_d - \bar{x}_{d+1}\|^2 + n^{-1/2}\beta^{-1}\|\bar{x}_j - x_{j-1}\|^2) \quad (\text{from (A.6)}) \\ &\leq \frac{1}{2} \mathbb{E}(n^{-1/2}\beta\rho^{j-1-d}\|x_{j-1} - \bar{x}_j\|^2 + n^{-1/2}\beta^{-1}\|\bar{x}_j - x_{j-1}\|^2) \\ &\quad (\text{by the inductive hypothesis}). \end{aligned}$$

Thus by setting $\beta = \rho^{(d+1-j)/2}$, we obtain

$$\mathbb{E}(\|x_d - x_{d+1}\| \|\bar{x}_j - x_{j-1}\|) \leq \frac{\rho^{(j-1-d)/2}}{n^{1/2}} \mathbb{E}(\|\bar{x}_j - x_{j-1}\|^2). \quad (\text{A.14})$$

By substituting (A.10) into (A.8) and taking expectation on both sides of (A.8), we obtain

$$\begin{aligned}
& \mathbb{E}(\|x_{j-1} - \bar{x}_j\|^2) - \mathbb{E}(\|x_j - \bar{x}_{j+1}\|^2) \\
& \leq 2\mathbb{E}(\|\bar{x}_j - x_{j-1}\| \|\bar{x}_j - \bar{x}_{j+1} + x_j - x_{j-1}\|) \\
& \leq 2\mathbb{E} \left(\|\bar{x}_j - x_{j-1}\| \left((2 + 2\Lambda\gamma) \|x_j - x_{j-1}\| + 2\Lambda\gamma \sum_{d=j-1-\tau}^{j-2} \|x_d - x_{d+1}\| \right) \right) \\
& = (4 + 4\Lambda\gamma) \mathbb{E}(\|\bar{x}_j - x_{j-1}\| \|x_j - x_{j-1}\|) + 4\Lambda\gamma \sum_{d=j-1-\tau}^{j-2} \mathbb{E}(\|\bar{x}_j - x_{j-1}\| \|x_d - x_{d+1}\|) \\
& \leq n^{-1/2} (4 + 4\Lambda\gamma) \mathbb{E}(\|\bar{x}_j - x_{j-1}\|^2) \\
& \quad + n^{-1/2} 4\Lambda\gamma \mathbb{E}(\|x_{j-1} - \bar{x}_j\|^2) \sum_{d=j-1-\tau}^{j-2} \rho^{(j-1-d)/2} \quad (\text{from (A.13) and (A.14)}) \\
& \leq n^{-1/2} (4 + 4\Lambda\gamma) \mathbb{E}(\|\bar{x}_j - x_{j-1}\|^2) + n^{-1/2} 4\Lambda\gamma \mathbb{E}(\|x_{j-1} - \bar{x}_j\|^2) \sum_{t=1}^{\tau} \rho^{t/2} \\
& = n^{-1/2} (4 + 4\Lambda\gamma(1 + \theta)) \mathbb{E}(\|x_{j-1} - \bar{x}_j\|^2),
\end{aligned}$$

where the last equality follows from the definition of θ in (4.4). It follows that

$$\begin{aligned}
\mathbb{E}(\|x_{j-1} - \bar{x}_j\|^2) & \leq \left(1 - n^{-1/2} (4 + 4\Lambda\gamma(1 + \theta)) \right)^{-1} \mathbb{E}(\|x_j - \bar{x}_{j+1}\|^2) \\
& \leq \rho \mathbb{E}(\|x_j - \bar{x}_{j+1}\|^2).
\end{aligned}$$

To see the last inequality, one only needs to verify that

$$\rho^{-1} \leq 1 - \frac{4 + 4\gamma\Lambda(1 + \theta)}{\sqrt{n}} \Leftrightarrow \gamma \leq \frac{\sqrt{n}(1 - \rho^{-1}) - 4}{4\Lambda(1 + \theta)},$$

and the last inequality is true because of the upper bound of γ in (4.5). We have thus proved (4.6).

Next we will show the expectation of the objective F is monotonically decreasing. We have by using the definition (A.1) and (4.2) that

$$\begin{aligned}
\mathbb{E}_{i(j)} F(x_{j+1}) & = \mathbb{E}_{i(j)} [f(x_j - (\Delta_j)_{i(j)} e_{i(j)}) + g(x_{j+1})] \\
& \leq \mathbb{E}_{i(j)} \left[f(x_j) + \langle \nabla_{i(j)} f(x_j), (\bar{x}_{j+1} - x_j)_{i(j)} \rangle + \frac{L_{\max}}{2} \|(x_{j+1} - x_j)_{i(j)}\|^2 \right. \\
& \quad \left. + g_{i(j)}((x_{j+1})_{i(j)}) + \sum_{l \neq i(j)} g_l((x_{j+1})_l) \right] \\
& = \mathbb{E}_{i(j)} \left[f(x_j) + \langle \nabla_{i(j)} f(x_j), (\bar{x}_{j+1} - x_j)_{i(j)} \rangle + \frac{L_{\max}}{2} \|(x_{j+1} - x_j)_{i(j)}\|^2 \right. \\
& \quad \left. + g_{i(j)}((x_{j+1})_{i(j)}) + \sum_{l \neq i(j)} g_l((x_j)_l) \right] \\
& = f(x_j) + \frac{n-1}{n} g(x_j) + n^{-1} \left(\langle \nabla f(x_j), \bar{x}_{j+1} - x_j \rangle + \frac{L_{\max}}{2} \|\bar{x}_{j+1} - x_j\|^2 + g(\bar{x}_{j+1}) \right), \blacksquare
\end{aligned}$$

where we used $\mathbb{E}_{i(j)} \sum_{l \neq i(j)} g_l(x_j)_l = \frac{n-1}{n} g(x_j)$ in the last equality. By adding and subtracting a term involving $\nabla f(\hat{x}_j)$, we obtain

$$\begin{aligned}
& \mathbb{E}_{i(j)} F(x_{j+1}) \\
& \leq F(x_j) + \frac{1}{n} \left(\langle \nabla f(\hat{x}_j), \bar{x}_{j+1} - x_j \rangle + \frac{L_{\max}}{2} \|\bar{x}_{j+1} - x_j\|^2 + g(\bar{x}_{j+1}) - g(x_j) \right) \\
& \quad + \frac{1}{n} \langle \nabla f(x_j) - \nabla f(\hat{x}_j), \bar{x}_{j+1} - x_j \rangle \\
& \leq F(x_j) + \frac{1}{n} \left(\frac{L_{\max}}{2} \|\bar{x}_{j+1} - x_j\|^2 - \frac{L_{\max}}{\gamma} \|\bar{x}_{j+1} - x_j\|^2 \right) \\
& \quad + \frac{1}{n} \langle \nabla f(x_j) - \nabla f(\hat{x}_j), \bar{x}_{j+1} - x_j \rangle \quad (\text{from (A.4) with } x = x_j) \\
& = F(x_j) - \left(\frac{1}{\gamma} - \frac{1}{2} \right) \frac{L_{\max}}{n} \|\bar{x}_{j+1} - x_j\|^2 + \frac{1}{n} \langle \nabla f(x_j) - \nabla f(\hat{x}_j), \bar{x}_{j+1} - x_j \rangle. \quad (\text{A.15})
\end{aligned}$$

Consider the expectation of the last term on the right-hand side of this expression. We have

$$\begin{aligned}
& \mathbb{E} \langle \nabla f(x_j) - \nabla f(\hat{x}_j), \bar{x}_{j+1} - x_j \rangle \\
& \leq \mathbb{E} (\|\nabla f(x_j) - \nabla f(\hat{x}_j)\| \|\bar{x}_{j+1} - x_j\|) \\
& \leq L_{\text{res}} \mathbb{E} \left(\sum_{d \in K(j)} \|x_{d+1} - x_d\| \|\bar{x}_{j+1} - x_j\| \right) \quad (\text{from (A.7)}) \\
& \leq L_{\text{res}} \sum_{d=j-\tau}^{j-1} \frac{\rho^{(j-d)/2}}{n^{1/2}} \mathbb{E} (\|x_j - \bar{x}_{j+1}\|^2) \quad (\text{from (A.14), and replacing } j \text{ by } j+1) \\
& \leq n^{-1/2} L_{\text{res}} \theta \mathbb{E} (\|x_j - \bar{x}_{j+1}\|^2) \quad (\text{from (4.4)}). \quad (\text{A.16})
\end{aligned}$$

By taking expectations on both sides of (A.15) and substituting (A.16), we obtain

$$\mathbb{E} F(x_{j+1}) \leq \mathbb{E} F(x_j) - \frac{1}{n} \left(\left(\frac{1}{\gamma} - \frac{1}{2} \right) L_{\max} - \frac{L_{\text{res}} \theta}{n^{1/2}} \right) \mathbb{E} \|\bar{x}_{j+1} - x_j\|^2.$$

To see $\left(\frac{1}{\gamma} - \frac{1}{2} \right) L_{\max} - \frac{L_{\text{res}} \theta}{n^{1/2}} \geq 0$ or equivalently $\left(\frac{1}{\gamma} - \frac{1}{2} \right) - \frac{\Lambda \theta}{n^{1/2}} \geq 0$, we note from (4.4) and (4.5) that

$$\gamma^{-1} \geq \psi \geq \frac{1}{2} + \frac{\Lambda \theta}{\sqrt{n}}.$$

Therefore, we have proved the monotonicity of the expectation of the objectives, that is,

$$\mathbb{E} F(x_{j+1}) \leq \mathbb{E} F(x_j), \quad j = 0, 1, 2, \dots \quad (\text{A.17})$$

Next we prove the sublinear convergence rate for the constrained smooth convex

case in (4.8). We have

$$\begin{aligned}
& \|x_{j+1} - \mathcal{P}_S(x_{j+1})\|^2 \leq \|x_{j+1} - \mathcal{P}_S(x_j)\|^2 \\
& = \|x_j - (\Delta_j)_{i(j)} e_{i(j)} - \mathcal{P}_S(x_j)\|^2 \\
& = \|x_j - \mathcal{P}_S(x_j)\|^2 + |(\Delta_j)_{i(j)}|^2 - 2\langle (x_j - \mathcal{P}_S(x_j))_{i(j)}, (\Delta_j)_{i(j)} \rangle \\
& = \|x_j - \mathcal{P}_S(x_j)\|^2 - |(\Delta_j)_{i(j)}|^2 - 2\langle (x_j - \mathcal{P}_S(x_j))_{i(j)} - (\Delta_j)_{i(j)}, (\Delta_j)_{i(j)} \rangle \\
& = \|x_j - \mathcal{P}_S(x_j)\|^2 - |(\Delta_j)_{i(j)}|^2 + 2\langle \mathcal{P}_S(x_j) - x_{j+1} \rangle_{i(j)}, (\Delta_j)_{i(j)} \rangle \quad (\text{from (A.1)}) \\
& \leq \|x_j - \mathcal{P}_S(x_j)\|^2 - |(\Delta_j)_{i(j)}|^2 + \\
& \quad \frac{2\gamma}{L_{\max}} [\langle (\mathcal{P}_S(x_j) - x_{j+1})_{i(j)}, \nabla_{i(j)} f(\hat{x}_j) \rangle + g_{i(j)}((\mathcal{P}_S(x_j))_{i(j)}) - g_{i(j)}((x_{j+1})_{i(j)})] \\
& \quad (\text{from (A.2)}) \\
& = \|x_j - \mathcal{P}_S(x_j)\|^2 - |(\Delta_j)_{i(j)}|^2 + \\
& \quad \frac{2\gamma}{L_{\max}} [\langle (\mathcal{P}_S(x_j) - x_j)_{i(j)}, \nabla_{i(j)} f(\hat{x}_j) \rangle + g_{i(j)}((\mathcal{P}_S(x_j))_{i(j)}) - g_{i(j)}((x_{j+1})_{i(j)})] + \\
& \quad \frac{2\gamma}{L_{\max}} (\langle (\Delta_j)_{i(j)}, \nabla_{i(j)} f(x_j) \rangle + \langle (\Delta_j)_{i(j)}, \nabla_{i(j)} f(\hat{x}_j) - \nabla_{i(j)} f(x_j) \rangle) \\
& \leq \|x_j - \mathcal{P}_S(x_j)\|^2 - |(\Delta_j)_{i(j)}|^2 + \\
& \quad \frac{2\gamma}{L_{\max}} [\langle (\mathcal{P}_S(x_j) - x_j)_{i(j)}, \nabla_{i(j)} f(\hat{x}_j) \rangle + g_{i(j)}((\mathcal{P}_S(x_j))_{i(j)}) - g_{i(j)}((x_{j+1})_{i(j)})] + \\
& \quad \frac{2\gamma}{L_{\max}} \left(f(x_j) - f(x_{j+1}) + \frac{L_{\max}}{2} |(\Delta_j)_{i(j)}|^2 + \langle (\Delta_j)_{i(j)}, \nabla_{i(j)} f(\hat{x}_j) - \nabla_{i(j)} f(x_j) \rangle \right) \\
& \quad (\text{from (A.3)}) \\
& = \|x_j - \mathcal{P}_S(x_j)\|^2 - (1 - \gamma) |(\Delta_j)_{i(j)}|^2 + \frac{2\gamma}{L_{\max}} \underbrace{\langle (\mathcal{P}_S(x_j) - x_j)_{i(j)}, \nabla_{i(j)} f(\hat{x}_j) \rangle}_{T_1} + \\
& \quad \frac{2\gamma}{L_{\max}} \underbrace{\langle (\Delta_j)_{i(j)}, \nabla_{i(j)} f(\hat{x}_j) - \nabla_{i(j)} f(x_j) \rangle}_{T_2} + \\
& \quad \frac{2\gamma}{L_{\max}} \underbrace{[f(x_j) - f(x_{j+1}) + g_{i(j)}((\mathcal{P}_S(x_j))_{i(j)}) - g_{i(j)}((x_{j+1})_{i(j)})]}_{T_3}. \tag{A.18}
\end{aligned}$$

We now seek upper bounds on the quantities T_1 , T_2 , and T_3 in the expectation sense. For simplicity, we construct a vector $\mathbf{b} \in \mathbb{R}^{|K(j)|}$ with $\mathbf{b}_t = \|\hat{x}_{j,t-1} - \hat{x}_{j,t}\|$. We have from elementary arguments that

$$\begin{aligned}
\mathbb{E}(\|\mathbf{b}\|^2) &= \sum_{t=0}^{|K(j)|-1} \mathbb{E}(\|\hat{x}_{j,t} - \hat{x}_{j,t+1}\|^2) = \sum_{t=1}^{|K(j)|} \mathbb{E}(\|x_{K(j)_t} - x_{K(j)_{t+1}}\|^2) \\
&= \sum_{d \in K(j)} \mathbb{E}(\|x_d - x_{d+1}\|^2) = \frac{1}{n} \sum_{d \in K(j)} \mathbb{E}\|x_d - \bar{x}_{d+1}\|^2 \leq \frac{1}{n} \sum_{d=j-\tau}^{j-1} \mathbb{E}\|x_d - \bar{x}_{d+1}\|^2 \\
&\leq \frac{1}{n} \sum_{t=1}^{\tau} \rho^t \mathbb{E}\|x_j - \bar{x}_{j+1}\|^2 \quad (\text{from (4.6)}) \\
&\leq \frac{\theta'}{n} \mathbb{E}\|x_j - \bar{x}_{j+1}\|^2 \quad (\text{from (4.4)}). \tag{A.19}
\end{aligned}$$

For the expectation of T_1 , defined in (A.18), we have

$$\begin{aligned}
\mathbb{E}(T_1) &= \mathbb{E}((\mathcal{P}_S(x_j) - x_j)_{i(j)} \nabla_{i(j)} f(\hat{x}_j)) \\
&= n^{-1} \mathbb{E} \langle \mathcal{P}_S(x_j) - x_j, \nabla f(\hat{x}_j) \rangle \\
&= n^{-1} \mathbb{E} \langle \mathcal{P}_S(x_j) - \hat{x}_j, \nabla f(\hat{x}_j) \rangle + n^{-1} \mathbb{E} \sum_{t=0}^{|K(j)|-1} \langle \hat{x}_{j,t} - \hat{x}_{j,t+1}, \nabla f(\hat{x}_j) \rangle \\
&= n^{-1} \mathbb{E} \langle \mathcal{P}_S(x_j) - \hat{x}_j, \nabla f(\hat{x}_j) \rangle \\
&\quad + n^{-1} \mathbb{E} \sum_{t=0}^{|K(j)|-1} (\langle \hat{x}_{j,t} - \hat{x}_{j,t+1}, \nabla f(\hat{x}_{j,t}) \rangle + \langle \hat{x}_{j,t} - \hat{x}_{j,t+1}, \nabla f(\hat{x}_j) - \nabla f(\hat{x}_{j,t}) \rangle) \\
&\leq n^{-1} \mathbb{E}(f_j^* - f(\hat{x}_j)) \\
&\quad + n^{-1} \mathbb{E} \sum_{t=0}^{|K(j)|-1} \left(f(\hat{x}_{j,t}) - f(\hat{x}_{j,t+1}) + \frac{L_{\max}}{2} \|\hat{x}_{j,t} - \hat{x}_{j,t+1}\|^2 \right) \\
&\quad + n^{-1} \mathbb{E} \sum_{t=0}^{|K(j)|-1} \langle \hat{x}_{j,t} - \hat{x}_{j,t+1}, \nabla f(\hat{x}_j) - \nabla f(\hat{x}_{j,t}) \rangle \quad (\text{from (1.3)}) \\
&= n^{-1} \mathbb{E}(f_j^* - f(x_j)) + \frac{L_{\max}}{2n} \mathbb{E} \|\mathbf{b}\|^2 \\
&\quad + n^{-1} \mathbb{E} \sum_{t=0}^{|K(j)|-1} \langle \hat{x}_{j,t} - \hat{x}_{j,t+1}, \nabla f(\hat{x}_j) - \nabla f(\hat{x}_{j,t}) \rangle \\
&= n^{-1} \mathbb{E}(f_j^* - f(x_j)) + \frac{L_{\max}}{2n} \mathbb{E} \|\mathbf{b}\|^2 \\
&\quad + n^{-1} \mathbb{E} \sum_{t=0}^{|K(j)|-1} \left\langle \hat{x}_{j,t} - \hat{x}_{j,t+1}, \sum_{t'=0}^{t-1} \nabla f(\hat{x}_{j,t'}) - \nabla f(\hat{x}_{j,t'+1}) \right\rangle \\
&\leq n^{-1} \mathbb{E}(f_j^* - f(x_j)) + \frac{L_{\max}}{2n} \mathbb{E} \|\mathbf{b}\|^2 \\
&\quad + n^{-1} \mathbb{E} \sum_{t=0}^{|K(j)|-1} L_{\max} \left(\|\hat{x}_{j,t} - \hat{x}_{j,t+1}\| \sum_{t'=0}^{t-1} \|\hat{x}_{j,t'} - \hat{x}_{j,t'+1}\| \right) \\
&= n^{-1} \mathbb{E}(f_j^* - f(x_j)) + \frac{L_{\max}}{2n} \mathbb{E} \|\mathbf{b}\|^2 + n^{-1} L_{\max} \mathbb{E} \sum_{t=0}^{|K(j)|-1} \left(\mathbf{b}_{t+1} \sum_{t'=0}^{t-1} \mathbf{b}_{t'+1} \right) \\
&= n^{-1} \mathbb{E}(f_j^* - f(x_j)) + \frac{L_{\max}}{2n} \mathbb{E} \|\mathbf{b}\|^2 + \frac{L_{\max}}{2n} \mathbb{E} (\|\mathbf{b}\|_1^2 - \|\mathbf{b}\|^2) \\
&= n^{-1} \mathbb{E}(f_j^* - f(x_j)) + \frac{L_{\max}}{2n} \mathbb{E} (\|\mathbf{b}\|_1^2) \\
&\leq n^{-1} \mathbb{E}(f_j^* - f(x_j)) + \frac{L_{\max} \tau}{2n} \mathbb{E} (\|\mathbf{b}\|^2) \quad (\text{since } \|\mathbf{b}\|_1 \leq \sqrt{|K(j)|} \|\mathbf{b}\| \leq \sqrt{\tau} \|\mathbf{b}\|) \\
&\leq n^{-1} \mathbb{E}(f_j^* - f(x_j)) + \frac{L_{\max} \tau \theta'}{2n^2} \mathbb{E} (\|x_j - \bar{x}_{j+1}\|^2) \quad (\text{from (A.19)}). \tag{A.20}
\end{aligned}$$

For the expectation of T_2 , we have

$$\begin{aligned}
\mathbb{E}(T_2) &= \mathbb{E}(\Delta_j)_{i(j)} (\nabla_{i(j)} f(\hat{x}_j) - \nabla_{i(j)} f(x_j)) \\
&= n^{-1} \mathbb{E} \langle \Delta_j, \nabla f(\hat{x}_j) - \nabla f(x_j) \rangle \\
&\leq n^{-1} \mathbb{E}(\|\Delta_j\| \|\nabla f(\hat{x}_j) - \nabla f(x_j)\|) \\
&\leq \frac{L_{\text{res}}}{n} \mathbb{E} \left(\sum_{d=j-\tau}^{j-1} \|\Delta_j\| \|x_d - x_{d+1}\| \right) \quad (\text{from (A.7)}) \\
&= \frac{L_{\text{res}}}{n} \mathbb{E} \left(\sum_{d=j-\tau}^{j-1} \|x_j - \bar{x}_{j+1}\| \|x_d - x_{d+1}\| \right) \\
&\leq \frac{L_{\text{res}}}{n^{3/2}} \sum_{d=j-\tau}^{j-1} \rho^{(j-d)/2} \mathbb{E} \|x_j - \bar{x}_{j+1}\|^2 \quad (\text{from (A.14) with } j \text{ replacing } j-1) \\
&\leq \frac{L_{\text{res}} \theta}{n^{3/2}} \mathbb{E} \|x_j - \bar{x}_{j+1}\|^2 \quad (\text{from (4.4)}). \tag{A.21}
\end{aligned}$$

For T_3 , let us look the expectation of several individual terms first

$$\mathbb{E}_{i(j)} g_{i(j)}((\mathcal{P}_S(x_j))_{i(j)}) = n^{-1} g(\mathcal{P}_S(x_j)) = n^{-1} g_j^*,$$

and

$$\begin{aligned}
\mathbb{E}_{i(j)} g_{i(j)}((x_{j+1})_{i(j)}) &= \mathbb{E}_{i(j)} (g(x_{j+1}) - g(x_j) + g_{i(j)}((x_j)_{i(j)})) \\
&= \mathbb{E}_{i(j)} g(x_{j+1}) - g(x_j) + n^{-1} g(x_j) \\
&= \mathbb{E}_{i(j)} g(x_{j+1}) - \frac{n-1}{n} g(x_j).
\end{aligned}$$

Now we take the expectation on T_3 and use the equalities above to obtain:

$$\begin{aligned}
\mathbb{E}(T_3) &= \mathbb{E} f(x_j) - \mathbb{E} f(x_{j+1}) + \mathbb{E} g_{i(j)}((\mathcal{P}_S(x_j))_{i(j)}) - \mathbb{E} g_{i(j)}((x_{j+1})_{i(j)}) \\
&= \mathbb{E} f(x_j) - \mathbb{E} f(x_{j+1}) + n^{-1} \mathbb{E} g_j^* - \mathbb{E} g(x_{j+1}) + \frac{n-1}{n} \mathbb{E} g(x_j). \tag{A.22}
\end{aligned}$$

By substituting the upper bounds from (A.20), (A.21), and (A.22) into (A.18), we obtain

$$\begin{aligned}
\mathbb{E} \|x_{j+1} - \mathcal{P}_S(x_{j+1})\|^2 &\leq \mathbb{E} \|x_j - \mathcal{P}_S(x_j)\|^2 - (1-\gamma) \mathbb{E} |(\Delta_j)_{i(j)}|^2 \\
&\quad + \frac{2\gamma}{L_{\text{max}}} \left(\frac{1}{n} \mathbb{E}(f_j^* - f(x_j)) + \frac{L_{\text{max}} \tau \theta'}{2n^2} \mathbb{E}(\|x_j - \bar{x}_{j+1}\|^2) \right) \\
&\quad + \frac{2\gamma}{L_{\text{max}}} \left(\frac{L_{\text{res}} \theta}{n^{3/2}} \mathbb{E} \|x_j - \bar{x}_{j+1}\|^2 \right) \\
&\quad + \frac{2\gamma}{L_{\text{max}}} \left(\mathbb{E} f(x_j) - \mathbb{E} f(x_{j+1}) + n^{-1} \mathbb{E} g_j^* - \mathbb{E} g(x_{j+1}) + \frac{n-1}{n} \mathbb{E} g(x_j) \right).
\end{aligned}$$

By using

$$\mathbb{E}_{i(j)} (|(\Delta_j)_{i(j)}|^2) = n^{-1} \|x_j - \bar{x}_{j+1}\|^2,$$

it follows that

$$\begin{aligned}
& \mathbb{E}\|x_{j+1} - \mathcal{P}_S(x_{j+1})\|^2 \leq \mathbb{E}\|x_j - \mathcal{P}_S(x_j)\|^2 \\
& \quad - \frac{1}{n} \left(1 - \gamma - \frac{\tau\theta'}{n} \gamma - \frac{2\Lambda\theta}{n^{1/2}} \gamma \right) \mathbb{E}\|x_j - \bar{x}_{j+1}\|^2 \\
& \quad + \frac{2\gamma}{L_{\max}n} (\mathbb{E}f_j^* - \mathbb{E}f(x_j) + \mathbb{E}g_j^*) \\
& \quad + \frac{2\gamma}{L_{\max}} (\mathbb{E}f(x_j) + \frac{n-1}{n} \mathbb{E}g(x_j) - \mathbb{E}f(x_{j+1}) - \mathbb{E}g(x_{j+1})) \\
& \leq \mathbb{E}\|x_j - \mathcal{P}_S(x_j)\|^2 + \frac{2\gamma}{L_{\max}n} (\mathbb{E}f_j^* - \mathbb{E}f(x_j) + \mathbb{E}g_j^*) \\
& \quad + \frac{2\gamma}{L_{\max}} \left(\mathbb{E}f(x_j) + \frac{n-1}{n} \mathbb{E}g(x_j) - \mathbb{E}f(x_{j+1}) - \mathbb{E}g(x_{j+1}) \right) \\
& \leq \mathbb{E}\|x_j - \mathcal{P}_S(x_j)\|^2 + \frac{2\gamma}{L_{\max}n} (F^* - \mathbb{E}F(x_j)) + \frac{2\gamma}{L_{\max}} (\mathbb{E}F(x_j) - \mathbb{E}F(x_{j+1})).
\end{aligned} \tag{A.23}$$

In the second inequality, we were able to drop the term involving $\mathbb{E}\|x_j - \bar{x}_{j+1}\|^2$ by using the fact that

$$1 - \gamma \left(1 + \frac{\tau\theta'}{n} + \frac{\Lambda\theta}{\sqrt{n}} \right) = 1 - \gamma\psi \geq 0,$$

which follows from the definition (4.4) of ψ and from the first upper bound on γ in (4.5). It follows from (A.23) that

$$\begin{aligned}
& \mathbb{E}\|x_{j+1} - \mathcal{P}_S(x_{j+1})\|^2 + \frac{2\gamma}{L_{\max}} (\mathbb{E}F(x_{j+1}) - F^*) \\
& \leq \mathbb{E}\|x_j - \mathcal{P}_S(x_j)\|^2 + \frac{2\gamma}{L_{\max}} (\mathbb{E}F(x_j) - F^*) - \frac{2\gamma}{L_{\max}n} (\mathbb{E}F(x_j) - F^*).
\end{aligned} \tag{A.24}$$

Defining

$$S_j := \mathbb{E}(\|x_j - \mathcal{P}_S(x_j)\|^2) + \frac{2\gamma}{L_{\max}} \mathbb{E}(F(x_j) - F^*), \tag{A.25}$$

we have from (A.24) that

$$S_{j+1} \leq S_j - \frac{2\gamma}{L_{\max}n} \mathbb{E}(F(x_j) - F^*), \tag{A.26}$$

so by induction, we have

$$S_{j+1} \leq S_0 - \frac{2\gamma}{L_{\max}n} \sum_{t=0}^j (\mathbb{E}F(x_t) - F^*) \leq S_0 - \frac{2\gamma(j+1)}{L_{\max}n} (F(x_0) - F^*), \tag{A.27}$$

where the second inequality follows from monotonicity of $\mathbb{E}F(x_j)$ (A.17). Note that

$$S_0 := \|x_0 - \mathcal{P}_S(x_0)\|^2 + \frac{2\gamma}{L_{\max}} (F(x_0) - F^*).$$

By substituting the definition of S_{j+1} into (A.27), we obtain

$$\begin{aligned} \mathbb{E}\|x_{j+1} - \mathcal{P}_S(x_{j+1})\|^2 &+ \frac{2\gamma}{L_{\max}}(\mathbb{E}F(x_{j+1}) - F^*) + \frac{2\gamma(j+1)}{L_{\max}n}(\mathbb{E}F(x_{j+1}) - F^*) \\ &\leq \|x_0 - \mathcal{P}_S(x_0)\|^2 + \frac{2\gamma}{L_{\max}}(F(x_0) - F^*). \end{aligned}$$

The sublinear convergence expression (4.8) follows when we drop the (nonnegative) first term on the left-hand side of this expression, and rearrange.

Finally, we prove the linear convergence rate (4.7) for the optimally strongly convex case. All bounds proven above continue to hold, and we make use the optimal strong convexity property in (1.2):

$$F(x_j) - F^* \geq \frac{l}{2}\|x_j - \mathcal{P}_S(x_j)\|^2.$$

By using this result together with some elementary manipulation, we obtain

$$\begin{aligned} F(x_j) - F^* &= \left(1 - \frac{L_{\max}}{l\gamma + L_{\max}}\right)(F(x_j) - F^*) + \frac{L_{\max}}{l\gamma + L_{\max}}(F(x_j) - F^*) \\ &\geq \left(1 - \frac{L_{\max}}{l\gamma + L_{\max}}\right)(F(x_j) - F^*) + \frac{L_{\max}l}{2(l\gamma + L_{\max})}\|x_j - \mathcal{P}_S(x_j)\|^2 \\ &= \frac{L_{\max}l}{2(l\gamma + L_{\max})} \left(\|x_j - \mathcal{P}_S(x_j)\|^2 + \frac{2\gamma}{L_{\max}}(F(x_j) - F^*)\right). \end{aligned} \quad (\text{A.28})$$

By taking expectations of both sides in this expression, and comparing with (A.25), we obtain

$$\mathbb{E}(F(x_j) - F^*) \geq \frac{L_{\max}l}{2(l\gamma + L_{\max})}S_j.$$

By substituting into (A.26), we obtain

$$\begin{aligned} S_{j+1} &\leq S_j - \left(\frac{2\gamma}{L_{\max}n}\right) \frac{L_{\max}l}{2(l\gamma + L_{\max})}S_j \\ &= \left(1 - \frac{l\gamma}{n(l\gamma + L_{\max})}\right) S_j \\ &\leq \left(1 - \frac{l\gamma}{n(l\gamma + L_{\max})}\right)^{j+1} S_0, \end{aligned}$$

where the last inequality follows from induction over j . We obtain (4.7) by substituting the definition (A.25) of S_j . \square

A.3. Proof of Corollary 4.2. *Proof.* Note that for ρ defined by (4.10), and using (4.9), we have

$$\begin{aligned} \rho^{(1+\tau)/2} &= \left(1 + \frac{4e\Lambda(\tau+1)}{\sqrt{n}}\right)^{1+\tau} = \left(\left(1 + \frac{4e\Lambda(\tau+1)}{\sqrt{n}}\right)^{\frac{\sqrt{n}}{4e\Lambda(\tau+1)}}\right)^{\frac{4e\Lambda(\tau+1)^2}{\sqrt{n}}} \\ &\leq e^{\frac{4e\Lambda(\tau+1)^2}{\sqrt{n}}} \leq e. \end{aligned} \quad (\text{A.29})$$

Thus from the definition of ψ (4.4), we have that

$$\begin{aligned}
\psi &= 1 + \frac{\tau\theta'}{n} + \frac{2\Lambda\theta}{\sqrt{n}} \\
&\leq 1 + \frac{\tau^2\rho^\tau}{n} + \frac{2\Lambda\tau\rho^{\tau/2}}{\sqrt{n}} \quad \left(\text{from } \theta = \sum_{t=1}^{\tau} \rho^{t/2} \leq \tau\rho^{\tau/2} \text{ and } \theta' = \sum_{t=1}^{\tau} \rho^t \leq \tau\rho^\tau \right) \\
&\leq 1 + \frac{\tau^2 e^2}{n} + \frac{2\Lambda\tau e}{\sqrt{n}} \quad (\text{from (A.29)}) \\
&\leq 1 + \frac{1}{16} + \frac{1}{2} \leq 2,
\end{aligned}$$

where for the second-last inequality we used (4.9) to obtain

$$\frac{\Lambda\tau e}{\sqrt{n}} \leq \frac{\Lambda\tau e}{4e\Lambda(\tau+1)^2} \leq \frac{1}{4}, \quad \frac{\tau^2 e^2}{n} = \left(\frac{\tau e}{\sqrt{n}} \right)^2 \leq \left(\frac{\Lambda\tau e}{\sqrt{n}} \right) \leq \frac{1}{16}.$$

Thus, the steplength parameter choice $\gamma = 1/2$ satisfies the first bound in (4.5). To show that the second bound in (4.5) holds also, we have

$$\begin{aligned}
&\frac{\sqrt{n}(1 - \rho^{-1}) - 4}{4(1 + \theta)\Lambda} \\
&\geq \frac{\sqrt{n}(1 - \rho^{-1})}{4(1 + \theta)\Lambda} - \frac{1}{2} \quad (\text{from } \theta \geq 1 \text{ and } \Lambda \geq 1) \\
&\geq \frac{\sqrt{n}(1 - \rho^{-1/2})}{4(1 + \theta)\Lambda} - \frac{1}{2} \\
&= \frac{\sqrt{n}(\rho^{1/2} - 1)}{4(1 + \theta)\rho^{1/2}\Lambda} - \frac{1}{2} \\
&\geq \frac{\sqrt{n}(\rho^{1/2} - 1)}{4(\tau + 1)\rho^{(\tau+1)/2}\Lambda} - \frac{1}{2} \quad \left(\text{from } (1 + \theta)\rho^{1/2} \leq (1 + \tau\rho^{\tau/2})\rho^{1/2} \leq (1 + \tau)\rho^{(\tau+1)/2} \right) \\
&\geq \frac{4e\Lambda(\tau + 1)}{4e(\tau + 1)\Lambda} - \frac{1}{2} \quad (\text{from (4.10) and (A.29)}) \\
&\geq 1 - \frac{1}{2} = \frac{1}{2}.
\end{aligned}$$

We can thus set $\gamma = 1/2$, and by substituting this choice into (4.7), we obtain (4.11). We obtain (4.12) by making the same substitution into (4.8). \square

REFERENCES

- [1] A. AGARWAL AND J. C. DUCHI, *Distributed delayed stochastic optimization*, in Proceedings of the Conference on Decision and Control, 2012, pp. 5451–5452.
- [2] M. ANITESCU, *Degenerate nonlinear programming with a quadratic growth condition*, SIAM Journal on Optimization, 10 (2000), pp. 1116–1135.
- [3] H. AVRON, A. DRUINSKY, AND A. GUPTA, *Revisiting asynchronous linear solvers: Provable convergence rate through randomization*, in Proceedings of the IEEE International Parallel and Distributed Processing Symposium, May 2014.
- [4] A. BECK AND M. TEOULLE, *A fast iterative shrinkage-thresholding algorithm for linear inverse problems*, SIAM Journal on Imaging Sciences, 2 (2009), pp. 183–202.
- [5] A. BECK AND L. TETRUASHVILI, *On the convergence of block coordinate descent type methods*, SIAM Journal on Optimization, 23 (2013), pp. 2037–2060.

- [6] D. P. BERTSEKAS AND J. N. TSITSIKLIS, *Parallel and Distributed Computation: Numerical Methods*, Prentice Hall, 1989.
- [7] 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 (2011), pp. 1–122.
- [8] J. K. BRADLEY, A. KYROLA, D. BICKSON, AND C. GUESTRIN, *Parallel coordinate descent for L_1 -regularized loss minimization*, in International Conference on Machine Learning, 2011.
- [9] C. CORTES AND V. VAPNIK, *Support vector networks*, Machine Learning, (1995), pp. 273–297.
- [10] A. COTTER, O. SHAMIR, N. SREBRO, AND K. SRIDHARAN, *Better mini-batch algorithms via accelerated gradient methods*, in Advances in Neural Information Processing Systems, vol. 24, 2011, pp. 1647–1655.
- [11] J. C. DUCHI, A. AGARWAL, AND M. J. WAINWRIGHT, *Dual averaging for distributed optimization: Convergence analysis and network scaling*, IEEE Transactions on Automatic Control, 57 (2012), pp. 592–606.
- [12] L. ELSNER, I. KOLTRACHT, AND M. NEUMANN, *Convergence of sequential and asynchronous paracontractions nonlinear paracontractions*, Numerische Mathematik, 62 (1992), pp. 305–316.
- [13] F. FACCHINEL, S. SAGRATELLA, AND G. SCUTARI, *Flexible parallel algorithms for big data optimization*, technical report, Department of Computer, Control, and Management Engineering, University of Rome "La Sapienza", November 2013. arXiv:1311.2444v1.
- [14] O. FERCOQ AND P. RICHTÁRIK, *Accelerated, parallel, and proximal coordinate descent*, technical report, School of Mathematics, University of Edinburgh, 2013. arXiv: 1312.5799.
- [15] ———, *Smooth minimization of nonsmooth functions by parallel coordinate descent*, technical report, School of Mathematics, University of Edinburgh, 2013. arXiv:1309.5885.
- [16] M. C. FERRIS AND O. L. MANGASARIAN, *Parallel variable distribution*, SIAM Journal on Optimization, 4 (1994), pp. 815–832.
- [17] A. FROMMER AND D. B. SZYLD, *On asynchronous iterations*, Journal of Computational and Applied Mathematics, 123 (2000), pp. 201–216.
- [18] D. GOLDFARB AND S. MA, *Fast multiple-splitting algorithms for convex optimization*, SIAM Journal on Optimization, 22 (2012), pp. 533–556.
- [19] A. J. HOFFMAN, *On approximate solutions of systems of linear inequalities*, Journal of Research of the National Bureau of Standards, 49 (1952), pp. 263–265.
- [20] M. LAI AND W. YIN, *Augmented L_1 and nuclear-norm models with a globally linearly convergent algorithm*, SIAM Journal on Imaging Sciences, 6 (2013), pp. 1059–1091.
- [21] J. LIU, S. J. WRIGHT, C. RÉ, V. BITTORF, AND S. SRIDHAR, *An asynchronous parallel stochastic coordinate descent algorithm*, technical report, Computer Sciences Department, University of Wisconsin-Madison, February 2014. arXiv: 1311.1873.
- [22] J. LIU, S. J. WRIGHT, AND S. SRIDHAR, *An asynchronous parallel randomized Kaczmarz algorithm*, technical report, Computer Sciences Department, University of Wisconsin-Madison, 2014. arXiv: 1401.4780.
- [23] Z. LU AND L. XIAO, *On the complexity analysis of randomized block-coordinate descent methods*, Technical Report MSR-TR-2013-53, Microsoft Research, May 2013. arXiv:1305.4723.
- [24] Z.-Q. LUO AND P. TSENG, *On the convergence of the coordinate descent method for convex differentiable minimization*, Journal of Optimization Theory and Applications, 72 (1992), pp. 7–35.
- [25] O. L. MANGASARIAN, *Parallel gradient distribution in unconstrained optimization*, SIAM Journal on Optimization, 33 (1995), pp. 916–1925.
- [26] I. NECOARA AND D. CLIPICI, *Efficient parallel coordinate descent algorithm for convex optimization problems with separable constraints: application to distributed MPC*, technical report, Automation and Systems Engineering Department, University Politehnica Bucharest, 2013. arXiv: 1302.3092.
- [27] I. NECOARA AND A. PATRASCU, *A random coordinate descent algorithm for optimization problems with composite objective function and linear coupled constraints*, technical report, Automation and Systems Engineering Department, University Politehnica Bucharest, 2013. arXiv: 1302.3074.
- [28] A. NEMIROVSKI, A. JUDITSKY, G. LAN, AND A. SHAPIRO, *Robust stochastic approximation approach to stochastic programming*, SIAM Journal on Optimization, 19 (2009), pp. 1574–1609.
- [29] Y. NESTEROV, *Introductory Lectures on Convex Optimization: A Basic Course*, Kluwer Academic Publishers, 2004.
- [30] ———, *Efficiency of coordinate descent methods on huge-scale optimization problems*, SIAM Journal on Optimization, 22 (2012), pp. 341–362.

- [31] F. NIU, B. RECHT, C. RÉ, AND S. J. WRIGHT, *Hogwild: A lock-free approach to parallelizing stochastic gradient descent*, Advances in Neural Information Processing Systems, 24 (2011), pp. 693–701.
- [32] Z. PENG, M. YAN, AND W. YIN, *Parallel and distributed sparse optimization*, tech. report, Department of Mathematics, UCLA, 2013.
- [33] P. RICHTÁRIK AND M. TAKÁČ, *Iteration complexity of randomized block-coordinate descent methods for minimizing a composite function*, Mathematical Programming, Series A, (2012). (Published Online).
- [34] ———, *Parallel coordinate descent methods for big data optimization*, technical report, Mathematics Department, University of Edinburgh, 2012. arXiv: 1212.0873.
- [35] A. SAHA AND A. TEWARI, *On the nonasymptotic convergence of cyclic coordinate descent methods*, SIAM Journal on Optimization, 23 (2013), pp. 576–601.
- [36] C. SCHERRER, A. TEWARI, M. HALAPPANAVAR, AND D. HAGLIN, *Feature clustering for accelerating parallel coordinate descent*, in Advances in Neural Information Processing, vol. 25, 2012, pp. 28–36.
- [37] S. SHALEV-SHWARTZ AND T. ZHANG, *Accelerated mini-batch stochastic dual coordinate ascent*, in Advances in Neural Information Processing Systems, C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger, eds., vol. 26, 2013, pp. 378–385.
- [38] O. SHAMIR AND T. ZHANG, *Stochastic gradient descent for non-smooth optimization: Convergence results and optimal averaging schemes*, in Proceedings of the International Conference on Machine Learning, 2013.
- [39] S. SRIDHAR, V. BITTORF, J. LIU, C. ZHANG, C. RÉ, AND S. J. WRIGHT, *An approximate efficient solver for LP rounding*, in Advances in Neural Information Processing Systems, vol. 26, 2013.
- [40] P. TSENG, *Convergence of a block coordinate descent method for nondifferentiable minimization*, Journal of Optimization Theory and Applications, 109 (2001), pp. 475–494.
- [41] P. TSENG, *On accelerated proximal gradient methods for convex-concave optimization*, technical report, University of Washington, 2008.
- [42] P. TSENG AND S. YUN, *A coordinate gradient descent method for nonsmooth separable minimization*, Mathematical Programming, Series B, 117 (2009), pp. 387–423.
- [43] ———, *A coordinate gradient descent method for linearly constrained smooth optimization and support vector machines training*, Computational Optimization and Applications, 47 (2010), pp. 179–206.
- [44] P.-W. WANG AND C.-J. LIN, *Iteration complexity of feasible descent methods for convex optimization*, technical report, Department of Computer Science, National Taiwan University, 2013.
- [45] S. J. WRIGHT, R. D. NOWAK, AND M. A. T. FIGUEIREDO, *Sparse reconstruction by separable approximation*, IEEE Transactions on Signal Processing, 57 (2009), pp. 2479–2493.