A Block Decomposition Algorithm for Sparse Optimization

Ganzhao Yuan[1], Li Shen[2], Wei-Shi Zheng[3,1]
[1]Peng Cheng Laboratory, China [2]Tencent AI Lab, China [3]Sun Yat-sen University, China
yuangzh@pcl.ac.cn, mathshenli@gmail.com, zhwshi@mail.sysu.edu.cn

ABSTRACT

Sparse optimization is a central problem in machine learning and computer vision. However, this problem is inherently NP-hard and thus difficult to solve in general. Combinatorial search methods find the global optimal solution but are confined to small-sized problems, while coordinate descent methods are efficient but often suffer from poor local minima. This paper considers a new block decomposition algorithm that combines the effectiveness of combinatorial search methods and the efficiency of coordinate descent methods. Specifically, we consider a random strategy or/and a greedy strategy to select a subset of coordinates as the working set, and then perform a global combinatorial search over the working set based on the original objective function. We show that our method finds stronger stationary points than Amir Beck et al.'s coordinate-wise optimization method. In addition, we establish the convergence rate of our algorithm. Our experiments on solving sparse regularized and sparsity constrained least squares optimization problems demonstrate that our method achieves state-of-the-art performance in terms of accuracy. For example, our method generally outperforms the well-known greedy pursuit method.

CCS CONCEPTS

• **Mathematics of computing** → **Combinatorial optimization**; • **Theory of computation** → **Nonconvex optimization**.

KEYWORDS

Sparse Optimization; NP-hard; Block Coordinate Descent; Nonconvex Optimization; Convex Optimization

ACM Reference Format:
Ganzhao Yuan[1], Li Shen[2], Wei-Shi Zheng[3,1]. 2020. A Block Decomposition Algorithm for Sparse Optimization. In *Proceedings of the 26th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD '20), August 23–27, 2020, Virtual Event, CA, USA.* ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/3394486.3403070

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
KDD '20, August 23–27, 2020, Virtual Event, CA, USA
© 2020 Association for Computing Machinery.
ACM ISBN 978-1-4503-7998-4/20/08…$15.00
https://doi.org/10.1145/3394486.3403070

1 INTRODUCTION

This paper mainly focuses on the following nonconvex sparsity constrained / sparse regularized optimization problem:

$$\boxed{\min_{\mathbf{x}} \ f(\mathbf{x}),\ s.t.\ \|\mathbf{x}\|_0 \le s} \ \text{or} \ \boxed{\min_{\mathbf{x}} \ f(\mathbf{x}) + \lambda\|\mathbf{x}\|_0,} \qquad (1)$$

where $\mathbf{x} \in \mathbb{R}^n$, λ is a positive constant, $s \in [n]$ is a positive integer, $f(\cdot)$ is assumed to be convex, and $\|\cdot\|_0$ is a function that counts the number of nonzero elements in a vector. Problem (1) can be rewritten as the following unified composite minimization problem ('$\triangleq$' means define):

$$\min_{\mathbf{x}} \ F(\mathbf{x}) \triangleq f(\mathbf{x}) + h(\mathbf{x}),\ \text{with}\ h(\mathbf{x}) \triangleq h_{\text{cons}}\ \text{or}\ h_{\text{regu}}.$$

Here, $h_{\text{cons}}(\mathbf{x}) \triangleq I_\Psi(\mathbf{x})$, $\Psi \triangleq \{\mathbf{x} \mid \|\mathbf{x}\|_0 \le s\}$, $I_\Psi(\cdot)$ is an indicator function on the set Ψ with $I_\Psi(\mathbf{x}) = \begin{cases} 0, & \mathbf{x} \in \Psi \\ \infty, & \mathbf{x} \notin \Psi \end{cases}$, and $h_{\text{regu}}(\mathbf{x}) \triangleq \lambda\|\mathbf{x}\|_0$. Problem (1) captures a variety of applications of interest in both machine learning and computer vision (e.g., sparse coding [1, 2, 15], sparse subspace clustering [16]).

This paper proposes a block decomposition algorithm using a proximal strategy and a combinatorial search strategy for solving the sparse optimization problem as in (1). We review existing methods in the literature and summarize the merits of our approach.

► **The Relaxed Approximation Method.** One popular method to solve Problem (1) is the convex or nonconvex relaxed approximation method [11, 40, 47]. Many approaches such as ℓ_1 norm, top-k norm, Schatten ℓ_p norm, re-weighted ℓ_1 norm, capped ℓ_1 norm, and half quadratic function have been proposed for solving sparse optimization problems in the last decade. It is generally believed that nonconvex methods often achieve better accuracy than the convex counterparts [8, 41, 42]. However, minimizing the approximate function does not necessarily lead to the minimization of the original function in Problem (1). ♠ Our method directly controls the sparsity of the solution and minimize the original objective function.

► **The Greedy Pursuit Method.** This method is often used to solve sparsity constrained optimization problems. It greedily selects at each step one coordinate of the variables which have some desirable benefits [9, 14, 27, 28, 36]. This method has a monotonically decreasing property and achieves optimality guarantees in some situations, but it is limited to solving problems with smooth objective functions (typically the square function). Furthermore, the solutions must be initialized to zero and may cause divergence when being incorporated to solve the bilinear matrix factorization problem [2]. ♠ Our method is a greedy coordinate descent algorithm without forcing the initial solution to zero.

► **The Combinatorial Search Method.** This method is typically concerned with NP-hard problems [13]. A naive strategy is an exhaustive search which systematically enumerates all possible candidates for the solution and picks the best candidate corresponding to the lowest objective value. The cutting plane method solves the convex linear programming relaxation and adds linear constraints to drive the solution towards binary variables, while the branch-and-cut method performs branches and applies cuts at the nodes of the tree having a lower bound that is worse than the current solution. Although in some cases these two methods converge without much effort, in the worse case they end up solving all 2^n convex subproblems. ♠ Our method leverages the effectiveness of combinatorial search methods.

► **The Proximal Gradient Method.** Based on the current gradient $\nabla f(\mathbf{x}^k)$, the proximal gradient method [3, 10, 20, 24, 31, 32] iteratively performs a gradient update followed by a proximal operation: $\mathbf{x}^{k+1} = \text{prox}(\mathbf{x}^k - \beta\nabla f(\mathbf{x}^k); \beta, h)$. Here the proximal operator $\text{prox}(\mathbf{a}; \beta, h) = \arg\min_{\mathbf{x}} \frac{1}{2}\|\mathbf{x} - \mathbf{a}\|_2^2 + \beta h(\mathbf{x})$ can be evaluated analytically, and $\beta = 1/L$ is the step size with L being the Lipschitz constant. This method is closely related to (block) coordinate descent [17, 25, 29, 34, 39] in the literature. Due to its simplicity, many strategies (e.g., variance reduction [21, 22, 38], asynchronous parallelism [23, 35], and non-uniform sampling [46]) have been proposed to accelerate proximal gradient method. However, existing works use a scalar step size and solve a first-order majorization/surrogate function via closed-form updates. Since Problem (1) is nonconvex, such a simple majorization function may not necessarily be a good approximation. ♠ Our method significantly outperforms proximal gradient method and inherits its computational advantages.

Contributions. The contributions of this paper are threefold. **(i)** Algorithmically, we introduce a novel block decomposition method for sparse optimization (See Section 2). **(ii)** Theoretically, we establish the optimality hierarchy of our algorithm and show that it always finds stronger stationary points than existing methods (See Section 3). Furthermore, we prove the convergence rate of our algorithm (See Section 4). Additional discussions for our method is provided in Section 5. **(iii)** Empirically, we have conducted experiments on some sparse optimization tasks to show the superiority of our method (See Section 6).

Notations. All vectors are column vectors and superscript T denotes transpose. For any vector $\mathbf{x} \in \mathbb{R}^n$ and any $i \in \{1, 2, ..., n\}$, we denote by $\mathbf{x}_i$ the i-th component of $\mathbf{x}$. The Euclidean inner product between $\mathbf{x}$ and $\mathbf{y}$ is denoted by $\langle \mathbf{x}, \mathbf{y} \rangle$ or $\mathbf{x}^\mathsf{T}\mathbf{y}$. $\mathbf{1}$ is an all-one column vector, and $\mathbf{e}_i$ is a unit vector with a 1 in the ith entry and 0 in all other entries. When β is a constant, β^t denotes the t-th power of β, and when β is an optimization variable, β^t denotes the value of β in the t-th iteration. The number of possible combinations choosing k items from n without repetition is denoted by C_n^k. For any $B \in \mathbb{N}^k$ containing k unique integers selected from $\{1, 2, ..., n\}$, we define $\bar{B} \triangleq \{1, 2, ..., n\} \setminus B$ and denote $\mathbf{x}_B$ as the sub-vector of $\mathbf{x}$ indexed by B.

2 THE PROPOSED BLOCK DECOMPOSITION ALGORITHM

This section presents our block decomposition algorithm for solving (1). Our algorithm is an iterative procedure. In every iteration, the index set of variables is separated into two sets B and $\bar{B}$, where B is the working set. We fix the variables corresponding to $\bar{B}$, while minimizing a sub-problem on variables corresponding to B. The proposed method is summarized in Algorithm 1.

Algorithm 1 The Proposed Block Decomposition Algorithm

1: Input: the size of the working set $k \in [n]$, the proximal point parameter $\theta > 0$, and an initial feasible solution $\mathbf{x}^0$. Set $t = 0$.
2: **while** not converge **do**
3: (S1) Employ some strategy to find a working set B of size k. We define $\bar{B} \triangleq \{1, 2, ..., n\} \setminus B$.
4: (S2) Solve the following subproblem globally using combinatorial search methods:

$$\mathbf{x}^{t+1} \Leftarrow \arg\min_{\mathbf{z}} \; f(\mathbf{z}) + h(\mathbf{z}) + \tfrac{\theta}{2}\|\mathbf{z} - \mathbf{x}^t\|^2, s.t.\ \mathbf{z}_{\bar{B}} = \mathbf{x}^t_{\bar{B}} \quad (2)$$

5: (S3) Increment t by 1
6: **end while**

At first glance, Algorithm 1 might seem to be merely a block coordinate descent algorithm [37] applied to (1). However, it has some interesting properties that are worth commenting on.

► **Two New Strategies. (i)** Instead of using majorization techniques for optimizing over the block of the variables, we consider minimizing the original objective function. Although the subproblem is NP-hard and admits no closed-form solution, we can use an exhaustive search to solve it exactly. **(ii)** We consider a proximal point strategy for the subproblem in (2). This is to guarantee sufficient descent condition for the optimization problem and global convergence of Algorithm 1 (refer to Proposition 2).

► **Solving the Subproblem Globally.** The subproblem in (2) essentially contains k unknown decision variables and can be solved exactly within sub-exponential time $\mathcal{O}(2^k)$. Using the variational reformulation of ℓ_0 pseudo-norm [1], Problem (2) can be reformulated as a mixed-integer optimization problem and solved by some global optimization solvers such as 'CPLEX' or 'Gurobi'. For simplicity, we consider a simple exhaustive search (a.k.a. generate and test method) to solve it. Specifically, for every coordinate of the k-dimensional subproblem, it has two states, i.e., zero/nonzero. We systematically enumerate the full binary tree to obtain all possible candidate solutions and then pick the best one that leads to the lowest objective value as the optimal solution.

► **Finding the Working Set.** We observe that it contains C_n^k possible combinations of choice for the working set. One may use a cyclic strategy to alternatingly select all the choices

[1] For all $\mathbf{x} \in \mathbb{R}^n$ with $\|\mathbf{x}\|_\infty \leq \rho$, it always holds that $\|\mathbf{x}\|_0 = \min_{\mathbf{v}} \langle \mathbf{1}, \mathbf{v} \rangle,\ s.t.\ \mathbf{v} \in \{0, 1\}^n,\ |\mathbf{x}| \leq \rho\mathbf{v}$.

of the working set. However, past results show that the coordinate gradient method results in faster convergence when the working set is chosen in an arbitrary order [18] or in a greedy manner [19, 37]. This inspires us to use a random strategy or a greedy strategy for finding the working set. We remark that the combination of the two strategies is preferred in practice.

Random strategy. We uniformly select one combination (which contains k coordinates) from the whole working set of size C_n^k. One good benefit of this strategy is that our algorithm is ensured to find a block-k stationary point (discussed later) in expectation.

Greedy strategy. Generally speaking, we pick the top-k coordinates that lead to the greatest descent when one variable is changed and the rest variables are fixed based on the current solution $\mathbf{x}^t$. We denote $Z \triangleq \{i : \mathbf{x}_i^t = 0\}$ and $\bar{Z} \triangleq \{j : \mathbf{x}_j^t \neq 0\}$. For Z, we solve a one-variable subproblem to compute the possible decrease for all $i \in Z$ of $\mathbf{x}^t$ when changing from zero to nonzero:

$$\forall i = 1, ..., |Z|,\ \mathbf{c}_i = \min_\alpha F(\mathbf{x}^t + \alpha \mathbf{e}_i) - F(\mathbf{x}^t).$$

For $\bar{Z}$, we compute the decrease for each coordinate $j \in \bar{Z}$ of $\mathbf{x}^t$ when changing from nonzero to exactly zero:

$$\forall j = 1, ..., |\bar{Z}|,\ \mathbf{d}_j = F(\mathbf{x}^t + \alpha \mathbf{e}_j) - F(\mathbf{x}^t),\ \alpha = \mathbf{x}_j^t.$$

We sort the vectors $\mathbf{c}$ and $\mathbf{d}$ in increasing order and then pick the top-k coordinates as the working set.

3 OPTIMALITY ANALYSIS

This section provides an optimality analysis of our method. We assume that $f(\mathbf{x})$ is a smooth convex function with its gradient being L-Lipschitz continuous. In the sequel, we present some necessary optimal conditions for (1). Since the block-k optimality condition is novel in this paper, it is necessary to clarify its relations with existing optimality conditions formally. We use $\breve{\mathbf{x}}$, $\grave{\mathbf{x}}$, and $\bar{\mathbf{x}}$ to denote an arbitrary basic stationary point, an L-stationary point, and a block-k stationary point, respectively.

Definition 1. *(Basic Stationary Point) A solution $\breve{\mathbf{x}}$ is called a basic stationary point if the following holds. $h \triangleq h_{cons}$: $\breve{\mathbf{x}} = \arg\min_{\mathbf{y}} f(\mathbf{y})$, s.t. $|\bar{Z}| \leq k$, $\mathbf{y}_Z = \mathbf{0}$; $h \triangleq h_{regu}$: $\breve{\mathbf{x}} = \arg\min_{\mathbf{y}} f(\mathbf{y})$, s.t. $\mathbf{y}_Z = \mathbf{0}$. Here, $Z \triangleq \{i|\breve{\mathbf{x}}_i = 0\}$, $\bar{Z} \triangleq \{j|\breve{\mathbf{x}}_j \neq 0\}$.*

Remarks. The basic stationary point states that the solution achieves its global optimality when the support set is restricted. The number of basic stationary points for $h \triangleq h_{\text{cons}}$ and $h \triangleq h_{\text{regu}}$ is $\sum_{i=0}^k C_n^i$ and $\sum_{i=0}^n C_n^i$, respectively. One good feature of the basic stationary condition is that the solution set is enumerable, which makes it possible to validate whether a solution is optimal for the original sparse optimization problem.

Definition 2. *(L-Stationary Point) A solution $\grave{\mathbf{x}}$ is an L-stationary point if it holds that: $\grave{\mathbf{x}} = \arg\min_{\mathbf{y}}\ g(\mathbf{y}, \grave{\mathbf{x}}) + h(\mathbf{y})$ with $g(\mathbf{y}, \mathbf{x}) \triangleq f(\mathbf{x}) + \langle \nabla f(\mathbf{x}), \mathbf{y} - \mathbf{x} \rangle + \frac{L}{2}\|\mathbf{y} - \mathbf{x}\|_2^2$.*

Remarks. This is the well-known proximal thresholding operator. The term $g(\mathbf{y}, \mathbf{x})$ is a majorization function of $f(\mathbf{y})$ and it always holds that $f(\mathbf{y}) \leq g(\mathbf{y}, \mathbf{x})$ for all $\mathbf{x}$ and $\mathbf{y}$. Although it has a closed-form solution, this simple surrogate function may not be a good majorization/surrogate function for the non-convex problem.

Definition 3. *(Block-k Stationary Point) A solution $\bar{\mathbf{x}}$ is a block-k stationary point if it holds that:*

$$\bar{\mathbf{x}} \in \arg\min_{\mathbf{z}} \mathcal{P}(\mathbf{z}; \bar{\mathbf{x}}, B) \triangleq \{F(\mathbf{z}), s.t. \mathbf{z}_{\bar{B}} = \bar{\mathbf{x}}_{\bar{B}}\}, \forall |B| = k. \quad (3)$$

Remarks. (i) The concept of the block-k stationary point is novel in this paper. Our method can inherently better explore the second-order / curvature information of the objective function. **(ii)** The sub-problem $\min_{\mathbf{z}}\ \mathcal{P}(\mathbf{z}; \bar{\mathbf{x}}, B)$ is NP-hard, and it takes sub-exponential time $\mathcal{O}(2^k)$ to solve it. However, since k is often very small, it can be tackled by some practical global optimization methods. **(iii)** Testing whether a solution $\bar{\mathbf{x}}$ is a block-k stationary point deterministically requires solving C_n^k subproblems, therefore leading to a total time complexity of $C_n^k \times \mathcal{O}(2^k)$. However, using a random strategy for finding the working set B from C_n^k combinations, we can test whether a solution $\bar{\mathbf{x}}$ is the block-k stationary point in expectation within a time complexity of $T \times \mathcal{O}(2^k)$ with the constant T being the number of times which is related to the confidence of the probability.

The following proposition states the relations between the three types of the stationary point.

Proposition 1. ***Optimality Hierarchy between the Optimality Conditions.*** *The following relationship holds:* *Basic Stat. Point* $\overset{(a)}{\Leftarrow}$ *L-Stat. Point* $\overset{(b)}{\Leftarrow}$ *Block-k Stat. Point* $\overset{(c)}{\Leftarrow}$ *Block-(k+1) Stat. Point* $\Leftarrow ... \Leftarrow$ *Block-n Stat. Point* $\overset{(d)}{\Leftrightarrow}$ *Optimal Point* *for sparse regularized (resp., for sparsity constrained) optimization problems with $k \geq 1$ (resp., $k \geq 2$).*

Proof. We denote $\Gamma_s(\mathbf{x})$ as the operator that sets all but the largest (in magnitude) s elements of $\mathbf{x}$ to zero.

(a) First, we prove that an L-stationary point $\grave{\mathbf{x}}$ is also a basic stationary point $\breve{\mathbf{x}}$ when $h \triangleq h_{\text{cons}}$. For an L-stationary point, we have $\grave{\mathbf{x}} = \Gamma_s(\grave{\mathbf{x}} - (\nabla f(\grave{\mathbf{x}}))/L)$. This implies that there exists an index set S such that $\breve{\mathbf{x}}_S = \breve{\mathbf{x}}_S - (\nabla f(\breve{\mathbf{x}}))_S/L$ and $\breve{\mathbf{x}}_{\{1,...,n\}\setminus S} = \mathbf{0}$, which is the optimal condition for a basic stationary point.

Second, we prove that an L-stationary point $\grave{\mathbf{x}}$ is also a basic stationary point $\breve{\mathbf{x}}$ when $h \triangleq h_{\text{regu}}$. Using Definition 2, we have the following closed-form solution for $\grave{\mathbf{x}}$:

$$\grave{\mathbf{x}}_i = \begin{cases} \grave{\mathbf{x}}_i - \nabla_i f(\grave{\mathbf{x}})/L, & (\grave{\mathbf{x}}_i - \nabla_i f(\grave{\mathbf{x}})/L)^2 > 2\lambda/L; \\ 0, & \text{else.} \end{cases}$$

This implies that there exists a support set S such that $\breve{\mathbf{x}}_S = \breve{\mathbf{x}}_S - (\nabla f(\breve{\mathbf{x}}))_S/L$, which is the optimal condition for a basic stationary point. Defining $Z \triangleq \{i|\grave{\mathbf{x}}_i = 0\}$, $\bar{Z} \triangleq \{j|\grave{\mathbf{x}}_j \neq 0\}$, we note that $\forall i \in Z,\ |\nabla f(\grave{\mathbf{x}})|_i \leq \sqrt{2\lambda L}$, and $\forall j \in \bar{Z},\ (\nabla f(\grave{\mathbf{x}}))_j = 0,\ |\grave{\mathbf{x}}_j| \geq \sqrt{2\lambda/L}$.

(b) First, we prove that a block-2 stationary point is also an L-stationary point for $h \triangleq h_{\text{cons}}$. Given a vector $\mathbf{a} \in \mathbb{R}^n$,

	Basic Stat.	L-Stat.	Block-1 Stat.	Block-2 Stat.	Block-3 Stat.	Block-4 Stat.	Block-5 Stat.	Block-6 Stat.
$h = h_{\text{cons}}$	57	14	–	2	1	1	1	1
$h = h_{\text{regu}}$	64	56	9	3	1	1	1	1

Table 1: Number of points satisfying optimality conditions.

we consider the following optimization problem:

$$\mathbf{z}_B^* = \arg \min_{\mathbf{z}_B,\ \forall |B|=2} \|\mathbf{z}-\mathbf{a}\|_2^2,\ s.t.\ \|\mathbf{z}_B\|_0 + \|\mathbf{z}_{\bar{B}}\|_0 \le s, \quad (4)$$

which essentially contains C_n^2 2-dimensional subproblems. It is not hard to validate that (4) achieves the optimal solution with $\mathbf{z}^* = \Gamma_s(\mathbf{a})$. For any block-2 stationary point $\bar{\mathbf{x}}$, we have $\bar{\mathbf{x}}_B = \arg\min_{\mathbf{z}_B} \|\mathbf{z}-(\bar{\mathbf{x}}-\nabla f(\bar{\mathbf{x}})/L)\|_2^2,\ s.t.\ \|\mathbf{z}_B\|_0+\|\mathbf{z}_{\bar{B}}\|_0 \le s$. Applying this conclusion with $\mathbf{a} = \bar{\mathbf{x}} - \nabla f(\bar{\mathbf{x}})/L$, we have $\bar{\mathbf{x}} = \Gamma_s(\bar{\mathbf{x}} - \nabla f(\bar{\mathbf{x}})/L)$.

Second, we prove that a block-1 stationary point is also an L-stationary point for $h \triangleq h_{\text{regu}}$. Assume that the convex objective function $f(\cdot)$ has coordinate-wise Lipschitz continuous gradient with constant $\mathbf{s}_i,\ \forall i = 1, 2, ..., n$. For all $\mathbf{x} \in \mathbb{R}^n,\ t \in \mathbb{R},\ i = 1, 2, ...n$, it holds that [29]: $f(\mathbf{x} + t\mathbf{e}_i) \le Q_i(\mathbf{x}, t) \triangleq f(\mathbf{x}) + \langle \nabla f(\mathbf{x}), t\mathbf{e}_i\rangle + \frac{\mathbf{s}_i}{2}\|t\mathbf{e}_i\|_2^2$. Any block-1 stationary point must satisfy the following relation: $0 \in \arg\min_t\ Q_i(\bar{\mathbf{x}}, t) + \lambda\|\bar{\mathbf{x}}_i + t\|_0,\ \forall i$. We have the following optimal condition for $\bar{\mathbf{x}}$ with $k = 1$: $\bar{\mathbf{x}}_i = \left\{ \begin{array}{ll} (\bar{\mathbf{x}}_i - \nabla_i f(\bar{\mathbf{x}})/\mathbf{s}_i), & (\bar{\mathbf{x}}_i - \nabla_i f(\bar{\mathbf{x}})/\mathbf{s}_i)^2 > 2\lambda/\mathbf{s}_i; \\ 0, & \text{else.} \end{array} \right\}$. Since $\forall i,\ \mathbf{s}_i \le L$, the latter formulation implies the former one.

(c) Assume $k_1 \ge k_2$. The subproblem for the block-k_2 stationary point is a subset of those of the block-k_1 stationary point. Therefore, the block-k_1 stationary point implies the block-k_2 stationary point.

(d) Obvious.

□

Remarks. It is worthwhile to point out that the seminal works of Amir Beck et al. present a coordinate-wise optimality condition for sparse optimization [3–5, 7]. However, our block-k condition is stronger since their optimality condition corresponds to $k = 1$ in our optimality condition framework.

A Running Example. We consider the following sparsity constrained / sparse regularized optimization problem: $\min_{\mathbf{x}\in\mathbb{R}^n}\ \frac{1}{2}\mathbf{x}^\intercal\mathbf{Q}\mathbf{x} + \mathbf{x}^\intercal\mathbf{p} + h(\mathbf{x})$. Here, $n = 6$, $\mathbf{Q} = \mathbf{c}\mathbf{c}^\intercal + \mathbf{I}$, $\mathbf{p} = \mathbf{1}$, $\mathbf{c} = [1\ 2\ 3\ 4\ 5\ 6]^\intercal$. The parameters for $h_{\text{cons}}(\mathbf{x})$ and $h_{\text{regu}}(\mathbf{x})$ are set to $(s,\ \lambda) = (4,\ 0.01)$. The stationary point distribution of this example can be found in Table 1. This problem contains $\sum_{i=0}^4 C_6^i = 57$ basic stationary points for $h \triangleq h_{\text{cons}}$, while it has $\sum_{i=0}^6 C_6^i = 2^6 = 64$ basic stationary points for $h \triangleq h_{\text{regu}}$. As k becomes large, the newly introduced type of local minimizer (i.e., block-k stationary point) becomes more restricted in the sense that it has a smaller number of stationary points. Moreover, any block-3 stationary point is also the unique global optimal solution for this example.

4 CONVERGENCE ANALYSIS

This section provides some convergence analysis for Algorithm 1. We assume that $f(\mathbf{x})$ is a smooth convex function with its gradient being L-Lipschitz continuous, and the working set of size k is selected randomly and uniformly (sample with replacement). Due to space limitations, some proofs are placed into the **Appendix**.

PROPOSITION 2. ***Global Convergence.*** *Letting $\{\mathbf{x}^t\}_{t=0}^\infty$ be the sequence generated by Algorithm 1, we have the following results.* ***(a)*** *It holds that: $F(\mathbf{x}^{t+1}) \le F(\mathbf{x}^t) - \frac{\theta}{2}\|\mathbf{x}^{t+1} - \mathbf{x}^t\|^2,\ \lim_{t\to\infty}\mathbb{E}[\|\mathbf{x}^{t+1} - \mathbf{x}^t\|] = 0$.* ***(b)*** *As $t \to \infty$, $\mathbf{x}^t$ converges to the block-k stationary point $\bar{\mathbf{x}}$ of (1) in expectation.*

Remarks. Coordinate descent may cycle indefinitely if each minimization step contains multiple solutions [33]. The introduction of the proximal point parameter $\theta > 0$ is necessary for our nonconvex problem since it guarantees sufficient decrease condition, which is essential for global convergence. Our algorithm is guaranteed to find the block-k stationary point, but it is in expectation.

We prove the convergence rate of our algorithm for sparsity constrained optimization with $h \triangleq h_{\text{cons}}$.

THEOREM 1. ***Convergence Rate for Sparsity Constrained Optimization.*** *Assume that $f(\cdot)$ is σ-strongly convex, and Lipschitz continuous such that $\forall t,\ \|\nabla f(\mathbf{x}^t)\|_2^2 \le \tau$ for some positive constant τ. Denoting $\alpha \triangleq \frac{n\theta}{k\sigma}/(1 + \frac{n\theta}{k\sigma})$, we have the following results:*

$$\mathbb{E}[F(\mathbf{x}^t) - F(\bar{\mathbf{x}})] \le (F(\mathbf{x}^0) - F(\bar{\mathbf{x}}))\alpha^t + \tfrac{\tau}{2\theta}\tfrac{\alpha}{1-\alpha},$$
$$\mathbb{E}[\tfrac{\sigma}{4}\|\mathbf{x}^{t+1} - \bar{\mathbf{x}}\|_2^2] \le \tfrac{2n\theta}{k}(F(\mathbf{x}^0) - F(\bar{\mathbf{x}}))\alpha^t + \tfrac{n}{k}\tfrac{\tau}{1-\alpha}.$$

PROOF. **(a)** First of all, we define the zero set and nonzero set of the solution $\mathbf{x}^{t+1}$ as follows:

$$S \triangleq \{i \mid i \in B,\ \mathbf{x}_i^{t+1} \ne 0\},\ Q \triangleq \{i \mid i \in B,\ \mathbf{x}_i^{t+1} = 0\}.$$

Using the optimality of $\mathbf{x}^{t+1}$ for the subproblem, we obtain

$$(\nabla f(\mathbf{x}^{t+1}))_S + \theta(\mathbf{x}_S^{t+1} - \mathbf{x}_S^t) = 0 \quad (5)$$

We derive the following inequalities:

$$\begin{aligned}
& \mathbb{E}[f(\mathbf{x}^{t+1}) - f(\bar{\mathbf{x}})] \\
\overset{(a)}{\le}\ & \mathbb{E}[\langle \mathbf{x}^{t+1} - \bar{\mathbf{x}},\ \nabla f(\mathbf{x}^{t+1})\rangle - \tfrac{\sigma}{2}\|\mathbf{x}^{t+1} - \bar{\mathbf{x}}\|_2^2] \\
\overset{(b)}{=}\ & \tfrac{n}{k}\mathbb{E}[\langle \mathbf{x}_B^{t+1} - \bar{\mathbf{x}}_B,\ (\nabla f(\mathbf{x}^{t+1}))_B\rangle - \tfrac{\sigma}{2}\|\mathbf{x}_B^{t+1} - \bar{\mathbf{x}}_B\|_2^2] \\
\overset{(c)}{\le}\ & \tfrac{n}{k}\tfrac{\sigma}{2}\mathbb{E}[\|(\nabla f(\mathbf{x}^{t+1}))_B/\sigma\|_2^2] \\
\overset{(d)}{=}\ & \tfrac{n}{k}\tfrac{1}{2\sigma}\left(\mathbb{E}[\|(\nabla f(\mathbf{x}^{t+1}))_S\|_2^2] + \mathbb{E}[\|(\nabla f(\mathbf{x}^{t+1}))_Q\|_2^2]\right) \\
\overset{(e)}{\le}\ & \mathbb{E}[\tfrac{n}{k}\tfrac{1}{2\sigma}[\|\theta(\mathbf{x}_S^t - \mathbf{x}_S^{t+1})\|_2^2 + \|(\nabla f(\mathbf{x}^{t+1}))_Q\|_2^2]] \\
\overset{(f)}{\le}\ & \mathbb{E}[\tfrac{n}{k}\tfrac{\theta^2}{2\sigma}\|\mathbf{x}^t - \mathbf{x}^{t+1}\|_2^2 + \tfrac{n\tau}{2\sigma k}] \\
\overset{(g)}{\le}\ & \mathbb{E}[\tfrac{n\theta}{\sigma k}[f(\mathbf{x}^t) - f(\mathbf{x}^{t+1})] + \tfrac{n\tau}{2\sigma k}] \\
\overset{(h)}{=}\ & \mathbb{E}[\tfrac{n\theta}{\sigma k}[(f(\mathbf{x}^t) - f(\bar{\mathbf{x}})) - (f(\mathbf{x}^{t+1}) - f(\bar{\mathbf{x}}))] + \tfrac{n\tau}{2\sigma k}] \quad (6)
\end{aligned}$$

where step (a) uses the strongly convexity of $f(\cdot)$; step (b) uses the fact that the working set B is selected with $\frac{k}{n}$

probability; step (c) uses the inequality that $\langle \mathbf{x},\ \mathbf{a}\rangle - \frac{\sigma}{2}\|\mathbf{x}\|_2^2 = \frac{\sigma}{2}\|\mathbf{a}/\sigma\|_2^2 - \frac{\sigma}{2}\|\mathbf{x}-\mathbf{a}/\sigma\|_2^2 \leq \frac{\sigma}{2}\|\mathbf{a}/\sigma\|_2^2$ for all $\mathbf{a},\ \mathbf{x}$; step (d) uses the fact that $B = S \cup Q$, step (e) uses (5); step (f) uses the fact that $\forall \mathbf{x},\ \|\mathbf{x}_S\|_2^2 \leq \|\mathbf{x}\|_2^2$ and the Lipschitz continuity of $f(\cdot)$ that $\forall t,\ \|\nabla f(\mathbf{x}^{t+1})\|_2^2 \leq \tau$; step (g) uses the sufficient decrease condition that $\frac{\theta}{2}\|\mathbf{x}^{t+1} - \mathbf{x}^t\|^2 \leq F(\mathbf{x}^t) - F(\mathbf{x}^{t+1})$; step (h) uses $f(\mathbf{x}^t) - f(\mathbf{x}^{t+1}) = (f(\mathbf{x}^t) - f(\bar{\mathbf{x}})) - (f(\mathbf{x}^{t+1}) - f(\bar{\mathbf{x}}))$. From (6), we have the following inequalities:

$$\mathbb{E}[(1 + \tfrac{n\theta}{ks})(f(\mathbf{x}^{t+1}) - f(\bar{\mathbf{x}}))] \leq \mathbb{E}[\tfrac{n\theta}{ks} \cdot (f(\mathbf{x}^t) - f(\bar{\mathbf{x}})) + \tfrac{n\tau}{2ks}]$$

$$\mathbb{E}[f(\mathbf{x}^{t+1}) - f(\bar{\mathbf{x}})] \leq \mathbb{E}[\alpha(f(\mathbf{x}^t) - f(\bar{\mathbf{x}})) + \tfrac{\frac{n}{2ks}}{\frac{n\theta}{k\sigma}}\alpha\tau]$$

$$\mathbb{E}[f(\mathbf{x}^{t+1}) - f(\bar{\mathbf{x}})] \leq \mathbb{E}[\alpha(f(\mathbf{x}^t) - f(\bar{\mathbf{x}})) + \tfrac{\alpha\tau}{2\theta}]$$

Solving this recursive formulation, we have:

$$\begin{aligned}\mathbb{E}[f(\mathbf{x}^t) - f(\bar{\mathbf{x}})] &\leq \mathbb{E}[\alpha^t(f(\mathbf{x}^0) - f(\bar{\mathbf{x}}))] + \tau\sum_{i=1}^{t}\alpha^i \\ &= \mathbb{E}[\alpha^t(f(\mathbf{x}^0) - f(\bar{\mathbf{x}}))] + \tfrac{\tau}{2\theta}\cdot\tfrac{\alpha(1-\alpha^t)}{1-\alpha} \\ &\leq \mathbb{E}[\alpha^t(f(\mathbf{x}^0) - f(\bar{\mathbf{x}}))] + \tfrac{\tau}{2\theta}\cdot\tfrac{\alpha}{1-\alpha}\end{aligned}$$

Since $\mathbf{x}^t$ is always a feasible solution for all $t = 1, 2, ...\infty$, we have $F(\mathbf{x}^t) = f(\mathbf{x}^t)$.

(b) We now prove the second part of this theorem. First, we derive the following inequalities:

$$\begin{aligned}\mathbb{E}[\|\mathbf{x}^{t+1} - \mathbf{x}^t\|_2^2] &\overset{(a)}{\leq} \mathbb{E}[\tfrac{2}{\theta}\left(F(\mathbf{x}^t) - F(\mathbf{x}^{t+1})\right)] \\ &\overset{(b)}{\leq} \mathbb{E}[\tfrac{2}{\theta}\left(F(\mathbf{x}^t) - F(\bar{\mathbf{x}})\right)] \\ &\overset{(c)}{\leq} \mathbb{E}[\tfrac{2}{\theta}\alpha^t(f(\mathbf{x}^0) - f(\bar{\mathbf{x}}))] + \tfrac{\tau\alpha}{\theta^2(1-\alpha)} \quad (7)\end{aligned}$$

where step (a) uses the sufficient decrease condition; step (b) uses the fact that $F(\bar{\mathbf{x}}) \leq F(\mathbf{x}^{t+1})$; step (c) uses the result in (7).

Second, we have the following results:

$$\begin{aligned}&\mathbb{E}[\tfrac{\sigma}{2}\|\mathbf{x}^{t+1} - \bar{\mathbf{x}}\|_2^2] \\ \overset{(a)}{\leq}\ &\mathbb{E}[\langle \mathbf{x}^{t+1} - \bar{\mathbf{x}},\ \nabla f(\mathbf{x}^{t+1})\rangle + f(\bar{\mathbf{x}}) - f(\mathbf{x}^{t+1})] \\ \overset{(b)}{\leq}\ &\mathbb{E}[\langle \mathbf{x}^{t+1} - \bar{\mathbf{x}},\ \nabla f(\mathbf{x}^{t+1})\rangle] \\ \overset{(c)}{\leq}\ &\mathbb{E}[\|\mathbf{x}^{t+1} - \bar{\mathbf{x}}\| \cdot \|\nabla f(\mathbf{x}^{t+1})\|] \quad (8)\end{aligned}$$

where step (a) uses the strongly convexity of $f(\cdot)$; step (b) uses the fact that $f(\bar{\mathbf{x}}) \leq f(\mathbf{x}^{t+1})$; step (c) uses the Cauchy-Schwarz inequality.

From (8), we further have the following results:

$$\begin{aligned}\mathbb{E}[\tfrac{\sigma}{4}\|\mathbf{x}^{t+1} - \bar{\mathbf{x}}\|_2^2] &\overset{(a)}{\leq} \mathbb{E}[\|\nabla f(\mathbf{x}^{t+1})\|_2^2] \\ &= \tfrac{n}{k}\mathbb{E}[\|\nabla_B f(\mathbf{x}^{t+1})\|_2^2] \\ &\overset{(b)}{=} \tfrac{n}{k}\mathbb{E}[\|\nabla_S f(\mathbf{x}^{t+1})\|_2^2 + \|\nabla_Q f(\mathbf{x}^{t+1})\|_2^2] \\ &\overset{(c)}{\leq} \tfrac{n}{k}\mathbb{E}[\theta^2\|\mathbf{x}_S^{t+1} - \mathbf{x}_S^t\|_2^2] + \tfrac{n}{k}\tau \\ &\overset{(d)}{=} \tfrac{n}{k}\mathbb{E}[2\theta\alpha^t(f(\mathbf{x}^0) - f(\bar{\mathbf{x}})) + \tfrac{\tau}{1-\alpha}]\end{aligned}$$

where step (a) uses the strongly convexity of $f(\cdot)$; step (b) uses the fact that $B = S \cup Q$; step (c) uses the assumption that $\|\nabla f(\mathbf{x}^t)\|_2^2 \leq \tau$ for all $\mathbf{x}^t$ and the optimality of $\mathbf{x}^{t+1}$ in (5); step (d) uses (7). Therefore, we finish the proof of this theorem.

□

Remarks. Our results of convergence rate are similar to those of the gradient hard thresholding pursuit as in [45]. The first term and the second term for our convergence rate are called parameter estimation error and statistical error, respectively. While their analysis relies on the conditions of restricted strong convexity/smoothness, our study relies on the requirements of generally strong convexity/smoothness.

We prove the convergence rate of our algorithm for sparse regularized optimization with $h \triangleq h_{\text{regu}}$.

THEOREM 2. ***Convergence Rate for Sparse Regularized Optimization.*** *For $h \triangleq h_{regu}$, we have the following results:*

(a) *It holds that $\forall i,\ |\mathbf{x}_i^t| \geq \delta > 0$ with $\mathbf{x}_i^t \neq 0$. Whenever $\mathbf{x}^{t+1} \neq \mathbf{x}^t$, we have $\|\mathbf{x}^{t+1} - \mathbf{x}^t\|_2^2 \geq \frac{k\delta^2}{n}$ and the objective value is decreased at least by D. The solution changes at most $\bar{J}$ times in expectation for finding a block-k stationary point $\bar{\mathbf{x}}$. Here δ, D, and $\bar{J}$ are respectively defined as:*

$$\delta \triangleq \min(\sqrt{\tfrac{2\lambda}{\theta+L}}, \min(|\mathbf{x}^0|)),\ D \triangleq \tfrac{k\theta\delta^2}{2n},\ \bar{J} \triangleq \tfrac{F(\mathbf{x}^0) - F(\bar{\mathbf{x}})}{D}. \quad (9)$$

(b) *Assume that $f(\cdot)$ is generally convex, and the solution is always bounded with $\|\mathbf{x}^t\|_\infty \leq \rho,\ \forall t$. If the support set of $\mathbf{x}^t$ does not changes for all $t = 0, 1, ..., \infty$, Algorithm 1 takes at most V_1 iterations in expectation to converge to a stationary point $\bar{\mathbf{x}}$ satisfying $F(\mathbf{x}^t) - F(\bar{\mathbf{x}}) \leq \epsilon$. Moreover, Algorithm 1 takes at most $V_1 \times \bar{J}$ iterations in expectation to converge to a stationary point $\bar{\mathbf{x}}$ satisfying $F(\mathbf{x}^t) - F(\bar{\mathbf{x}}) \leq \epsilon$. Here, V_1 is defined as:*

$$V_1 = \max(\tfrac{4\nu^2}{\theta}, \sqrt{\tfrac{2\nu^2(F(\mathbf{x}^0) - F(\bar{\mathbf{x}}))}{\theta}})/\epsilon,\ \text{with } \nu \triangleq \tfrac{2n\rho\sqrt{k}\theta}{k}. \quad (10)$$

(c) *Assume that $f(\cdot)$ is σ-strongly convex. If the support set of $\mathbf{x}^t$ does not changes for all $t = 0, 1, ..., \infty$, Algorithm 1 takes at most V_2 iterations in expectation to converge to a stationary point $\bar{\mathbf{x}}$ satisfying $F(\mathbf{x}^t) - F(\bar{\mathbf{x}}) \leq \epsilon$. Moreover, Algorithm 1 takes at most $V_2 \times \bar{J}$ iterations in expectation to converge to a stationary point $\bar{\mathbf{x}}$ satisfying $F(\mathbf{x}^t) - F(\bar{\mathbf{x}}) \leq \epsilon$. Here, V_2 is defined as:*

$$V_2 = \log_\alpha(\epsilon/(F(\mathbf{x}^0) - F(\bar{\mathbf{x}}))),\ \text{with } \alpha \triangleq \tfrac{n\theta}{k\sigma}/(1 + \tfrac{n\theta}{k\sigma}). \quad (11)$$

Remarks. (i) When the support set is fixed, the optimization problem reduces to a convex problem. **(ii)** We derive a upper bound for the number of changes $\bar{J}$ for the support set in **(a)**, and a upper bound on the number of iterations V_1 (or V_2) performed after the support set is fixed in **(b)** (or **(c)**). Multiplying these two bounds, we can establish the upper bound of the number of iterations for Algorithm 1 to converge. However, these bounds are not tight enough.

The following theorem establishes an improved convergence rate of our algorithm with $h \triangleq h_{\text{regu}}$.

THEOREM 3. ***Improved Convergence Rate for Sparse Regularized Optimization.*** *For $h \triangleq h_{regu}$, we have the following results:*

(a) *Assume that $f(\cdot)$ is generally convex, and the solution is always bounded with $\|\mathbf{x}^t\|_\infty \leq \rho,\ \forall t$. Algorithm 1 takes at most N_1 iterations in expectation to converge to a block-k stationary point $\bar{\mathbf{x}}$ satisfying $F(\mathbf{x}^t) - F(\bar{\mathbf{x}}) \leq \epsilon$, where* $N_1 = (\frac{\bar{J}}{D} + \frac{1}{\epsilon}) \times \max(\frac{4\nu^2}{\theta}, \sqrt{\frac{2\nu^2(F(\mathbf{x}^0)-F(\bar{\mathbf{x}})-D)}{\theta}})$.

(b) *Assume that $f(\cdot)$ is σ-strongly convex. Algorithm 1 takes at most N_2 iterations in expectation to converge to a block-k stationary point $\bar{\mathbf{x}}$ satisfying $F(\mathbf{x}^t) - F(\bar{\mathbf{x}}) \leq \epsilon$, where* $N_2 = \bar{J}\log_\alpha(\frac{D}{(F(\mathbf{x}^0)-F(\bar{\mathbf{x}}))}) + \log_\alpha(\frac{\epsilon}{F(\mathbf{x}^0)-D-F(\bar{\mathbf{x}})})$.

Remarks. (i) Our proof of Theorem 3 is based on the results in Theorem 2 and a similar iterative bounding technique as in [24]. **(ii)** If $\bar{J} \geq 2$ and the accuracy ϵ is sufficiently small such that $\epsilon \leq \frac{D}{2}$, we have $\frac{\bar{J}}{D} + \frac{1}{\epsilon} \leq \frac{\bar{J}}{2\epsilon} + \frac{1}{\epsilon} \leq \frac{\bar{J}}{2\epsilon} + \frac{\bar{J}/2}{\epsilon} = \frac{\bar{J}}{\epsilon}$, leading to $(\frac{\bar{J}}{D} + \frac{1}{\epsilon}) \times \max(\frac{4\nu^2}{\theta}, \sqrt{2\nu^2(F(\mathbf{x}^0) - F(\bar{\mathbf{x}}) - D)/\theta}) \leq \frac{\bar{J}}{\epsilon} \times \max(\frac{4\nu^2}{\theta}, \sqrt{2\nu^2(F(\mathbf{x}^0) - F(\bar{\mathbf{x}}))/\theta})$. Using the same assumption and strategy, we have $\bar{J}\log_\alpha(D/(F(\mathbf{x}^0) - F(\bar{\mathbf{x}}))) + \log_\alpha(\epsilon/(F(\mathbf{x}^0) - D - F(\bar{\mathbf{x}}))) \leq \bar{J} \times \log_\alpha(\epsilon/(F(\mathbf{x}^0) - F(\bar{\mathbf{x}})))$. In this situation, the bounds in Theorem 3 are tighter than those in Theorem 2.

5 DISCUSSIONS

This section provides additional discussions for the proposed method.

► **When the objective function is complicated.** In step (S2) of the proposed algorithm, a global solution is to be found for the subproblem. When $f(\cdot)$ is simple (e.g., a quadratic function), we can find efficient and exact solutions to the subproblems. We now consider the situation when f is complicated (e.g., logistic regression, maximum entropy models). One can still find a quadratic majorizer $Q(\mathbf{x}, \mathbf{z})$ for the convex smooth function $f(\mathbf{x})$ with

$$\forall\ \mathbf{z},\ \mathbf{x},\ f(\mathbf{x}) \leq Q(\mathbf{x}, \mathbf{z}) \triangleq f(\mathbf{z}) + (\mathbf{x} - \mathbf{z})^\intercal \nabla f(\mathbf{z}) + \tfrac{1}{2}(\mathbf{x} - \mathbf{z})^\intercal \mathbf{M}(\mathbf{z})(\mathbf{x} - \mathbf{z}),\ \mathbf{M}(\mathbf{z}) \succ \nabla f^2(\mathbf{z}).$$

By minimizing the upper bound of $f(\mathbf{x})$ (i.e., the quadratic surrogate function) at the current estimate $\mathbf{x}^t$, i.e.,

$$\mathbf{x}^{t+1} \Leftarrow \arg\min_{\mathbf{x}}\ Q(\mathbf{x}, \mathbf{x}^t) + h(\mathbf{x}),$$

we can drive the objective downward until a stationary point is reached. We will obtain a stationary point $\ddot{\mathbf{x}}$ satisfying:

$$\ddot{\mathbf{x}} = \arg\min_{\mathbf{z}}\ h(\mathbf{z}) + f(\ddot{\mathbf{x}}) + (\mathbf{z} - \ddot{\mathbf{x}})^\intercal \nabla f(\ddot{\mathbf{x}}) + \tfrac{1}{2}(\mathbf{z} - \ddot{\mathbf{x}})^\intercal \mathbf{M}(\ddot{\mathbf{x}})(\mathbf{z} - \ddot{\mathbf{x}}),\ s.t.\ \ddot{\mathbf{x}}_{\bar{B}} = \mathbf{z}_{\bar{B}}$$

for all $\bar{B}$. We denote this optimality condition as the Newton block-k stationary point. It is weaker than the full block-k stationary point as in Definition 3. However, it is stronger than the L-Stationary Point as in Definition 2.

► **Computational efficiency.** Block coordinate descent is shown to be very efficient for solving convex problems (e.g., support vector machines [12, 18], LASSO problems [37], nonnegative matrix factorization [19]). The main difference of our block coordinate descent from existing ones is that our method needs to solve a small-sized NP-hard subproblem globally which takes subexponential time $\mathcal{O}(2^k)$. As a result, our algorithm finds a block-k approximation solution for the original NP-hard problem within $\mathcal{O}(2^k)$ time. When k is large, it is hard to enumerate the full binary tree since the subproblem is equally NP-hard. However, k is relatively small in practice (e.g., 2 to 20). In addition, real-world applications often have some special (e.g., unbalanced, sparse, local) structure and block-k stationary point could also be the global stationary point (refer to Table 1 of this paper).

6 EXPERIMENTAL VALIDATION

This section demonstrates the effectiveness of our algorithm on two sparse optimization tasks, namely the sparse regularized least squares problem and the sparsity constrained least squares problem.

Given a design matrix $\mathbf{A} \in \mathbb{R}^{m\times n}$ and an observation vector $\mathbf{b} \in \mathbb{R}^m$, we solve the following optimization problem:

$$\min_{\mathbf{x}}\ \tfrac{1}{2}\|\mathbf{A}\mathbf{x} - \mathbf{b}\|_2^2,\ s.t.\ \|\mathbf{x}\|_0 \leq s,\quad \text{or}\ \min_{\mathbf{x}}\ \tfrac{1}{2}\|\mathbf{A}\mathbf{x} - \mathbf{b}\|_2^2 + \lambda\|\mathbf{x}\|_0,$$

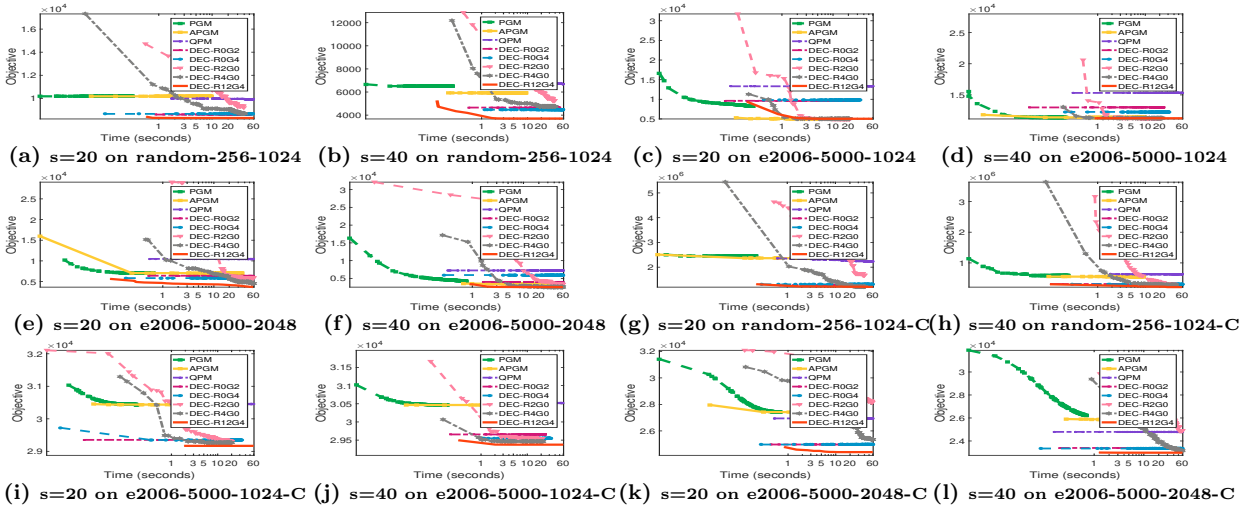
where s and λ are given parameters.

Experimental Settings. We use **DEC-R**i**G**j to denote our block decomposition method along with selecting i coordinates using the **R**andom strategy and j coordinates using the **G**reedy strategy. We keep a record of the relative changes of the objective function values by $r_t = (f(\mathbf{x}^t) - f(\mathbf{x}^{t+1}))/f(\mathbf{x}^t)$. We let **DEC** run up to T iterations and stop it at iteration $t < T$ if $\text{mean}([r_{t-\min(t,\varrho)+1},\ r_{t-min(t,\varrho)+2}, ..., r_t]) \leq \epsilon$. We use the default value $(\theta,\ \epsilon,\ \varrho,\ T) = (10^{-3},\ 10^{-5},\ 50,\ 1000)$ for **DEC**. All codes were implemented in Matlab on an Intel 3.20GHz CPU with 8 GB RAM. We measure the quality of the solution by comparing the objective values for different methods. Note that although **DEC** uses the randomized strategy to find the working set, we can always measure the quality of the solution by computing the deterministic objective value.

Data Sets. Four types of data sets for $\{\mathbf{A}, \mathbf{b}\}$ are considered in our experiments. **(i)** 'random-m-n': We generate the design matrix as $\mathbf{A} = \text{randn}(m, n)$, where $\text{randn}(m, n)$ is a function that returns a standard Gaussian random matrix of size $m \times n$. To generate the sparse original signal $\ddot{\mathbf{x}} \in \mathbb{R}^n$, we select a support set of size 100 uniformly at random and set them to arbitrary number sampled from standard Gaussian distribution, the observation vector is generated via $\mathbf{b} = \mathbf{A}\ddot{\mathbf{x}} + \mathbf{o}$ with $\mathbf{o} = 10 \times \text{randn}(m, 1)$. **(ii)** 'e2006-m-n': We use the real-world data set 'e2006' [2] which has been used in sparse optimization [46]. We uniformly select m examples and n dimensions from the original data set. **(iii)** 'random-m-n-C': To verify the robustness of **DEC**, we generate design matrices containing outliers by $\mathcal{P}(\mathbf{A})$. Here, $\mathcal{P}(\mathbf{A}) \in \mathbb{R}^{m\times n}$ is a noisy version of $\mathbf{A} \in \mathbb{R}^{m\times n}$ where 2% of the entries of $\mathbf{A}$ are corrupted uniformly by scaling the original values by 100 times [3]. We use the same sampling strategy to generate $\mathbf{A}$ as in 'random-m-n'. Note that the Hessian matrix can be

[2] https://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/

[3] Matlab script: I=randperm(m*n,round(0.02*m*n)); A(I)=A(I)*100.

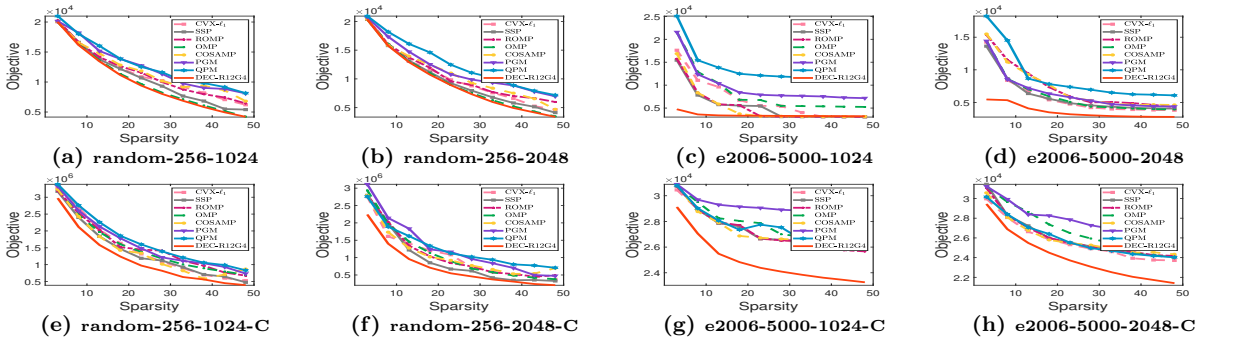


(a) s=20 on random-256-1024 (b) s=40 on random-256-1024 (c) s=20 on e2006-5000-1024 (d) s=40 on e2006-5000-1024

(e) s=20 on e2006-5000-2048 (f) s=40 on e2006-5000-2048 (g) s=20 on random-256-1024-C (h) s=40 on random-256-1024-C

(i) s=20 on e2006-5000-1024-C (j) s=40 on e2006-5000-1024-C (k) s=20 on e2006-5000-2048-C (l) s=40 on e2006-5000-2048-C

Figure 1: Convergence curve and computional efficiency for solving sparsity constrained least squares problems on different data sets with different s.



(a) random-256-1024 (b) random-256-2048 (c) e2006-5000-1024 (d) e2006-5000-2048

(e) random-256-1024-C (f) random-256-2048-C (g) e2006-5000-1024-C (h) e2006-5000-2048-C

Figure 2: Experimental results on sparsity constrained least squares problems on different data sets with varying the sparsity of the solution.

ill-conditioned. **(iv)** 'e2006-m-n-C': We use the same corrupting strategy to generate the corrupted real-world data as in 'random-m-n-C'.

► **Sparsity Constrained Least Squares Problem.** We compare **DEC** with 8 state-of-the-art sparse optimization algorithms. **(i)** Proximal Gradient Method (PGM) [2], **(ii)** Accerlated Proximal Gradient Method (APGM), and **(iii)** Quadratic Penalty Method (QPM) [26] are gradient-type methods based on iterative hard thresholding. **(iv)** Subspace Pursuit (SSP) [14], **(v)** Regularized Orthogonal Matching Pursuit (ROMP) [28], **(vi)** Orthogonal Matching Pursuit (OMP) [36], and **(vii)** Compressive Sampling Matched Pursuit (CoSaMP)[27] are greedy algorithms based on iterative support set detection. We use the Matlab implementation in the 'sparsify' toolbox[4]. We also include the comparison with **(viii)** Convex ℓ_1 Approximation Method (CVX-ℓ_1). We use PGM to solve the convex ℓ_1 regularized problem, with the regulation parameter being swept over $\lambda = \{2^{-10}, 2^{-8}, ..., 2^{10}\}$. The solution that leads to smallest objective after a hard thresholding projection and re-optimization over the support set is selected. Since the optimal solution is expected to be sparse, we initialize the solutions of {PGM, APGM, QPM, CVX-ℓ_1, **DEC**} to $10^{-7} \times \text{randn}(n, 1)$ and project them to feasible solutions. The initial solution of greedy pursuit methods are initialized to zero points implicitly. We vary $s = \{3,\ 8,\ 13,\ 18,\ ..., 50\}$ on different data sets and show the average results of using 5 random initial points.

First, we show the convergence curve and computational efficiency of **DEC** by comparing with gradient-type methods {PGM, APGM, QPM}. Several observations can be drawn from Figure 1. **(i)** PGM and APGM achieve similar performance and they get stuck into poor local minima. **(ii) DEC** is more effective than {PGM, APGM}. In addition, we find that as the parameter k becomes larger, more higher accuracy is achieved. **(iii) DEC-R**0**G**2 converges quickly but it generally leads to worse solution quality than **DEC-R**2**G**0. Based on this observation, we conclude that a combined random and greedy strategy is preferred. **(iv) DEC** generally takes

[4] http://www.personal.soton.ac.uk/tb1m08/sparsify/sparsify.html

	PGM-ℓ_0	APGM-ℓ_0	PGM-ℓ_1	PGM-ℓ_p	DEC-R10G2
results on random-256-1024					
$\lambda = 10^0$	6.9e+2	2.4e+4	7.8e+2	4.0e+2	4.8e+2
$\lambda = 10^1$	2.3e+3	3.8e+4	3.3e+3	1.9e+3	2.2e+3
$\lambda = 10^2$	2.0e+4	1.3e+5	1.8e+4	1.1e+4	9.4e+3
$\lambda = 10^3$	2.5e+4	1.0e+6	2.4e+4	2.4e+4	2.4e+4
results on random-256-2048					
$\lambda = 10^0$	1.3e+3	2.7e+4	1.4e+3	6.0e+2	5.4e+2
$\lambda = 10^1$	2.9e+3	4.5e+4	4.9e+3	2.2e+3	2.2e+3
$\lambda = 10^2$	2.2e+4	2.3e+5	2.1e+4	1.1e+4	9.5e+3
$\lambda = 10^3$	2.7e+4	2.1e+6	2.6e+4	2.7e+4	2.6e+4
results on e2006-5000-1024					
$\lambda = 10^0$	8.5e+3	3.3e+4	1.1e+4	1.8e+4	7.3e+3
$\lambda = 10^1$	9.4e+3	4.2e+4	3.2e+4	3.2e+4	8.6e+3
$\lambda = 10^2$	3.2e+4	1.3e+5	3.2e+4	3.2e+4	1.3e+4
$\lambda = 10^3$	1.8e+4	1.1e+6	3.2e+4	3.2e+4	1.1e+4
results on e2006-5000-2048					
$\lambda = 10^0$	3.1e+3	3.4e+4	4.4e+3	1.4e+4	2.6e+3
$\lambda = 10^1$	5.2e+3	5.3e+4	1.2e+4	1.2e+4	4.5e+3
$\lambda = 10^2$	3.2e+4	2.4e+5	3.2e+4	3.2e+4	7.0e+3
$\lambda = 10^3$	1.8e+4	2.1e+6	3.2e+4	3.2e+4	1.3e+4
results on random-256-1024-C					
$\lambda = 10^0$	9.6e+2	5.7e+6	1.0e+3	1.0e+3	8.9e+2
$\lambda = 10^1$	8.1e+3	3.5e+6	1.0e+4	8.2e+3	7.3e+3
$\lambda = 10^2$	5.8e+4	6.2e+6	8.9e+4	5.4e+4	5.1e+4
$\lambda = 10^3$	2.5e+5	5.3e+6	3.7e+5	2.2e+5	2.0e+5
results on random-256-2048-C					
$\lambda = 10^0$	1.9e+3	5.7e+6	2.0e+3	1.9e+3	1.2e+3
$\lambda = 10^1$	1.7e+4	7.7e+6	2.0e+4	1.6e+4	9.2e+3
$\lambda = 10^2$	8.4e+4	4.2e+6	1.6e+5	6.4e+4	5.3e+4
$\lambda = 10^3$	2.5e+5	9.6e+6	6.3e+5	2.5e+5	2.4e+5
results on e2006-5000-1024-C					
$\lambda = 10^0$	3.0e+4	3.3e+4	2.8e+4	2.9e+4	2.2e+4
$\lambda = 10^1$	3.2e+4	4.2e+4	3.2e+4	3.2e+4	2.3e+4
$\lambda = 10^2$	3.2e+4	1.3e+5	3.2e+4	3.2e+4	2.9e+4
$\lambda = 10^3$	3.2e+4	1.1e+6	3.2e+4	3.2e+4	3.2e+4
results on e2006-5000-2048-C					
$\lambda = 10^0$	2.9e+4	3.4e+4	2.6e+4	2.7e+4	1.7e+4
$\lambda = 10^1$	3.2e+4	5.3e+4	3.2e+4	3.2e+4	2.1e+4
$\lambda = 10^2$	3.2e+4	2.4e+5	3.2e+4	3.2e+4	2.7e+4
$\lambda = 10^3$	3.2e+4	2.1e+6	3.2e+4	3.2e+4	3.2e+4

Table 2: Comparisons of objective values of all the methods for solving the sparse regularized least squares problem. The 1^{st}, 2^{nd}, and 3^{rd} best results are colored with red, blue and green, respectively.

less than 30 seconds to converge in *all* our instances with obtaining reasonably good accuracy.

Second, we show the experimental results on sparsity constrained least squares problems with varying the cardinality s. Several conclusions can be drawn from Figure 2. **(i)** The methods {PGM, APGM, QPM} based on iterative hard thresholding generally lead to bad performance. **(ii)** OMP and ROMP are not stable and sometimes they achieve bad accuracy. **(iii) DEC** presents comparable performance to the greedy methods on {'random-256-1024', ' random-256-2048'} but it significantly and consistently outperforms the greedy methods on the other data sets.

► **Sparse Regularized Least Squares Problem.** We use Proximal Gradient Method (PGM) and Accelerated Proximal Gradient Method (APGM) [6, 30] to solve the ℓ_0 norm problem directly, leading to two compared methods **(i)** PGM-ℓ_0 and **(ii)** APGM-ℓ_0. We apply PGM to solve the convex ℓ_1 relaxation and nonconvex ℓ_p relaxation of the original ℓ_0 norm problem, resulting in additional two methods **(iii)** PGM-ℓ_1 and **(iv)** PGM-ℓ_p. We compare **DEC** with {PGM-ℓ_0, APGM-ℓ_0, PGM-ℓ_1, PGM-ℓ_p}. We initialize the solutions of all the methods to $10^{-7} \times \text{randn}(n, 1)$. For PGM-$\ell_p$, we set $p = \frac{1}{2}$ and use the efficient closed-form solver [40] to compute the proximal operator.

We show the objective values of all the methods with varying the hyper-parameter λ on different data sets in Table 2. Two observations can be drawn. **(i)** PGM-ℓ_p achieves better performance than the convex method PGM-ℓ_1. **(ii) DEC** generally outperforms the other methods in all our data sets.

We demonstrate the average computing time for the compared methods in Table 3. We have two observations. **(i) DEC** takes several times longer to converge than the compared methods. **(ii) DEC** generally takes less than 70 seconds to converge in *all* our instances. We argue that the computational time is acceptable and pays off as **DEC** achieves significantly higher accuracy. The main bottleneck of computation is on solving the small-sized subproblems using sub-exponential time $\mathcal{O}(2^k)$. The parameter k can be viewed as a tuning parameter to balance the efficacy and efficiency.

	PGM-ℓ_0	APGM-ℓ_0	PGM-ℓ_1	PGM-ℓ_p	DEC-R10G2
r.-256-1024	12 ± 3	13 ± 3	5 ± 3	15 ± 3	36 ± 3
r.-256-2048	11 ± 3	11 ± 3	9 ± 3	16 ± 3	66 ± 7
e.-5000-1024	12 ± 3	11 ± 3	8 ± 3	14 ± 3	45 ± 3
e.-5000-2048	12 ± 3	10 ± 3	12 ± 3	5 ± 3	65 ± 8

Table 3: Comparisons of average times (in seconds) of all the methods on different data sets for solving the sparse regularized least squares problem.

7 CONCLUSIONS

This paper presents an effective and practical method for solving sparse optimization problems. Our approach takes advantage of the effectiveness of the combinatorial search and the efficiency of coordinate descent. We provided rigorous optimality analysis and convergence analysis for the proposed algorithm. Our experiments show that our method achieves state-of-the-art performance. Our block decomposition algorithm has been extended to solve binary optimization problems [44] and sparse generalized eigenvalue problems [43].

Acknowledgments. This work was supported by NSFC (U1911401), Key-Area Research and Development Program of Guangdong Province (2019B121204008), NSFC (61772570, U1811461), Guangzhou Research Project (201902010037), Pearl River S&T Nova Program of Guangzhou (201806010056), and Guangdong Natural Science Funds for Distinguished Young Scholar (2018B030306025).

REFERENCES

[1] Michal Aharon, Michael Elad, and Alfred Bruckstein. 2006. K-SVD: An Algorithm for Designing Overcomplete Dictionaries for Sparse Representation. *IEEE Transactions on Signal Processing* 54, 11 (2006), 4311–4322.

[2] Chenglong Bao, Hui Ji, Yuhui Quan, and Zuowei Shen. 2016. Dictionary Learning for Sparse Coding: Algorithms and Convergence Analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)* 38, 7 (2016), 1356–1369.

[3] Amir Beck and Yonina C Eldar. 2013. Sparsity constrained nonlinear optimization: Optimality conditions and algorithms. *SIAM Journal on Optimization (SIOPT)* 23, 3 (2013), 1480–1509.

[4] Amir Beck and Nadav Hallak. 2019. Optimization problems involving group sparsity terms. *Mathematical Programming* 178, 1-2 (2019), 39–67.

[5] Amir Beck and Nadav Hallak. 2020. On the Minimization Over Sparse Symmetric Sets: Projections, Optimality Conditions, and Algorithms. *Mathematics of Operations Research* (2020).

[6] Amir Beck and Marc Teboulle. 2009. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM Journal on Imaging Sciences (SIIMS)* 2, 1 (2009), 183–202.

[7] Amir Beck and Yakov Vaisbourd. 2016. The Sparse Principal Component Analysis Problem: Optimality Conditions and Algorithms. *Journal of Optimization Theory and Applications* 170, 1 (2016), 119–143.

[8] Shujun Bi, Xiaolan Liu, and Shaohua Pan. 2014. Exact Penalty Decomposition Method for Zero-Norm Minimization Based on MPEC Formulation. *SIAM Journal on Scientific Computing (SISC)* 36, 4 (2014).

[9] Thomas Blumensath and Mike E Davies. 2008. Gradient pursuits. *IEEE Trans. on Signal Processing* 56, 6 (2008), 2370–2382.

[10] Thomas Blumensath and Mike E. Davies. 2009. Iterative hard thresholding for compressed sensing. *Applied and Computational Harmonic Analysis* 27, 3 (2009), 265 – 274.

[11] Emmanuel J Candes and Terence Tao. 2005. Decoding by linear programming. *IEEE Transactions on Information Theory* 51, 12 (2005), 4203–4215.

[12] Kai-Wei Chang, Cho-Jui Hsieh, and Chih-Jen Lin. 2008. Coordinate descent method for large-scale l2-loss linear support vector machines. *J. of Machine Learning Research* 9 (2008), 1369–1398.

[13] Michele Conforti, Gérard Cornuéjols, and Giacomo Zambelli. 2014. *Integer programming*. Vol. 271. Springer.

[14] Wei Dai and Olgica Milenkovic. 2009. Subspace pursuit for compressive sensing signal reconstruction. *IEEE Transactions on Information Theory* 55, 5 (2009), 2230–2249.

[15] David L. Donoho. 2006. Compressed sensing. *IEEE Transactions on Information Theory* 52, 4 (2006), 1289–1306.

[16] Ehsan Elhamifar and Rene Vidal. 2013. Sparse subspace clustering: Algorithm, theory, and applications. *IEEE Trans. on Pattern Analysis and Machine Intelligence* 35, 11 (2013), 2765–2781.

[17] Mingyi Hong, Xiangfeng Wang, Meisam Razaviyayn, and Zhi-Quan Luo. 2013. Iteration complexity analysis of block coordinate descent methods. *Mathematical Programming* (2013), 1–30.

[18] Cho-Jui Hsieh, Kai-Wei Chang, Chih-Jen Lin, S Sathiya Keerthi, and Sellamanickam Sundararajan. 2008. A dual coordinate descent method for large-scale linear SVM. In *International Conference on Machine Learning (ICML)*. 408–415.

[19] Cho-Jui Hsieh and Inderjit S Dhillon. 2011. Fast coordinate descent methods with variable selection for non-negative matrix factorization. In *ACM International Conference on Knowledge Discovery and Data Mining (SIGKDD)*. 1064–1072.

[20] Prateek Jain, Ambuj Tewari, and Purushottam Kar. 2014. On iterative hard thresholding methods for high-dimensional m-estimation. In *Neural Information Processing Systems*. 685–693.

[21] Rie Johnson and Tong Zhang. 2013. Accelerating stochastic gradient descent using predictive variance reduction. In *Advances in Neural Information Processing Systems (NeurIPS)*. 315–323.

[22] Xingguo Li, Tuo Zhao, Raman Arora, Han Liu, and Jarvis D. Haupt. 2016. Stochastic Variance Reduced Optimization for Nonconvex Sparse Learning. In *Proceedings of the 33nd International Conference on Machine Learning (ICML)*, Vol. 48. 917–925.

[23] Ji Liu, Stephen J Wright, Christopher Ré, Victor Bittorf, and Srikrishna Sridhar. 2015. An asynchronous parallel stochastic coordinate descent algorithm. *Journal of Machine Learning Research (JMLR)* 16, 285-322 (2015), 1–5.

[24] Zhaosong Lu. 2014. Iterative hard thresholding methods for ℓ_0 regularized convex cone programming. *Mathematical Programming* 147, 1-2 (2014), 125–154.

[25] Zhaosong Lu and Lin Xiao. 2015. On the complexity analysis of randomized block-coordinate descent methods. *Mathematical Programming* 152, 1-2 (2015), 615–642.

[26] Zhaosong Lu and Yong Zhang. 2013. Sparse Approximation via Penalty Decomposition Methods. *SIAM Journal on Optimization (SIOPT)* 23, 4 (2013), 2448–2478.

[27] Deanna Needell and Joel A Tropp. 2009. CoSaMP: Iterative signal recovery from incomplete and inaccurate samples. *Applied and Computational Harmonic Analysis* 26, 3 (2009), 301–321.

[28] Deanna Needell and Roman Vershynin. 2010. Signal recovery from incomplete and inaccurate measurements via regularized orthogonal matching pursuit. *IEEE Journal of Selected Topics in Signal Processing* 4, 2 (2010), 310–316.

[29] Yu Nesterov. 2012. Efficiency of coordinate descent methods on huge-scale optimization problems. *SIAM Journal on Optimization (SIOPT)* 22, 2 (2012), 341–362.

[30] Yurii Nesterov. 2013. *Introductory lectures on convex optimization: A basic course*. Vol. 87. Springer Science & Business Media.

[31] Andrei Patrascu and Ion Necoara. 2015. Efficient random coordinate descent algorithms for large-scale structured nonconvex optimization. *J. of Global Optimization* 61, 1 (2015), 19–46.

[32] Andrei Patrascu and Ion Necoara. 2015. Random Coordinate Descent Methods for ℓ_0 Regularized Convex Optimization. *IEEE Trans. Automat. Control* 60, 7 (2015), 1811–1824.

[33] M. J. D. Powell. 1973. On search directions for minimization algorithms. *Mathematical Programming* 4, 1 (1973), 193–201.

[34] Meisam Razaviyayn, Mingyi Hong, and Zhi-Quan Luo. 2013. A unified convergence analysis of block successive minimization methods for nonsmooth optimization. *SIAM Journal on Optimization (SIOPT)* 23, 2 (2013), 1126–1153.

[35] Benjamin Recht, Christopher Re, Stephen Wright, and Feng Niu. 2011. Hogwild: A lock-free approach to parallelizing stochastic gradient descent. In *Neural Information Processing Systems (NeurIPS)*. 693–701.

[36] Joel A Tropp and Anna C Gilbert. 2007. Signal recovery from random measurements via orthogonal matching pursuit. *IEEE Transactions on Information Theory* 53, 12 (2007), 4655–4666.

[37] Paul Tseng and Sangwoon Yun. 2009. A coordinate gradient descent method for nonsmooth separable minimization. *Mathematical Programming* 117, 1 (2009), 387–423.

[38] Lin Xiao and Tong Zhang. 2014. A proximal stochastic gradient method with progressive variance reduction. *SIAM Journal on Optimization (SIOPT)* 24, 4 (2014), 2057–2075.

[39] Yangyang Xu and Wotao Yin. 2013. A block coordinate descent method for regularized multiconvex optimization with applications to nonnegative tensor factorization and completion. *SIAM Journal on Imaging Sciences (SIIMS)* 6, 3 (2013), 1758–1789.

[40] Zongben Xu, Xiangyu Chang, Fengmin Xu, and Hai Zhang. 2012. $L_{1/2}$ regularization: A thresholding representation theory and a fast solver. *IEEE Transactions on Neural Networks and Learning Systems* 23, 7 (2012), 1013–1027.

[41] Ganzhao Yuan and Bernard Ghanem. 2016. Sparsity Constrained Minimization via Mathematical Programming with Equilibrium Constraints. In *arXiv:1608.04430*.

[42] Ganzhao Yuan and Bernard Ghanem. 2019. ℓ_0TV: A Sparse Optimization Method for Impulse Noise Image Restoration. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)* 41, 2 (2019), 352–364.

[43] Ganzhao Yuan, Li Shen, and Wei-Shi Zheng. 2019. A Decomposition Algorithm for the Sparse Generalized Eigenvalue Problem. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 6113–6122.

[44] Ganzhao Yuan, Li Shen, and Wei-Shi Zheng. June, 2017. A Hybrid Method of Combinatorial Search and Coordinate Descent for Discrete Optimization. *arXiv:1706.06493* (June, 2017).

[45] Xiao-Tong Yuan, Ping Li, and Tong Zhang. 2017. Gradient Hard Thresholding Pursuit. *Journal of Machine Learning Research* 18 (2017), 166:1–166:43.

[46] Aston Zhang and Quanquan Gu. 2016. Accelerated Stochastic Block Coordinate Descent with Optimal Sampling. In *ACM International Conference on Knowledge Discovery and Data Mining (SIGKDD)*. 2035–2044.

[47] Tong Zhang. 2010. Analysis of Multi-stage Convex Relaxation for Sparse Regularization. *Journal of Machine Learning Research (JMLR)* 11, 35 (2010), 1081–1107.

Appendix

The appendix section is organized as follows. Section 8 presents a useful lemma. Section 9, 10, and 11 present respectively the proof of Proposition 1, Theorem 2, and Theorem 3.

8 A USEFUL LEMMA

The following lemma is useful in our proof.

LEMMA 1. *Assume a nonnegative sequence* $\{u^t\}_{t=0}^{\infty}$ *satisfies* $(u^{t+1})^2 \leq C(u^t - u^{t+1})$ *for some nonnegative constant* C. *We have:*

$$u^t \leq \frac{\max(2C, \sqrt{Cu^0})}{t} \tag{12}$$

PROOF. We denote $C_1 \triangleq \max(2C, \sqrt{Cu^0})$. Solving this quadratic inequality, we have:

$$u^{t+1} \leq -\frac{C}{2} + \frac{C}{2}\sqrt{1 + \frac{4u^t}{C}} \tag{13}$$

We now show that $u^{t+1} \leq \frac{C_1}{t+1}$, which can be obtained by mathematical induction. (i) When $t = 0$, we have $u^1 \leq -\frac{C}{2} + \frac{C}{2}\sqrt{1 + \frac{4u^0}{C}} \leq -\frac{C}{2} + \frac{C}{2}(1 + \sqrt{\frac{4u^0}{C}}) = \frac{C}{2}\sqrt{\frac{4u^0}{C}} = \sqrt{Cu^0} \leq \frac{C_1}{t+1}$. (ii) When $t \geq 1$, we assume that $u^t \leq \frac{C_1}{t}$ holds. We derive the following results: $t \geq 1 \Rightarrow \frac{t+1}{t} \leq 2 \overset{(a)}{\Rightarrow} C\frac{t+1}{t} \leq C_1 \overset{(b)}{\Rightarrow} C(\frac{1}{t} - \frac{1}{t+1}) \leq \frac{C_1}{(t+1)^2} \Rightarrow \frac{C}{t} \leq \frac{C}{t+1} + \frac{C_1}{(t+1)^2} \Rightarrow \frac{CC_1}{t} \leq \frac{CC_1}{t+1} + \frac{C_1^2}{(t+1)^2} \Rightarrow \frac{C^2}{4} + \frac{CC_1}{t} \leq \frac{CC_1}{t+1} + \frac{C_1^2}{(t+1)^2} + \frac{C^2}{4} \Rightarrow \frac{C^2}{4}(1 + \frac{4C_1}{Ct}) \leq (\frac{C}{2} + \frac{C_1}{t+1})^2 \Rightarrow \frac{C}{2}\sqrt{1 + \frac{4C_1}{Ct}} \leq \frac{C}{2} + \frac{C_1}{t+1} \Rightarrow -\frac{C}{2} + \frac{C}{2}\sqrt{1 + \frac{4C_1}{Ct}} \leq \frac{C_1}{t+1} \overset{(c)}{\Rightarrow} -\frac{C}{2} + \frac{C}{2}\sqrt{1 + \frac{4u^t}{C}} \leq \frac{C_1}{t+1} \Rightarrow u^{t+1} \leq \frac{C_1}{t+1}$. Here, step (a) uses $2C \leq C_1$; step (b) uses $\frac{1}{t(t+1)} = \frac{1}{t} - \frac{1}{t+1}$; step (c) uses $u^t \leq \frac{C_1}{t}$.

□

9 PROOF OF PROPOSITION 1

PROOF. **(a)** Due to the optimality of $\mathbf{x}^{t+1}$, we have: $F(\mathbf{x}^{t+1}) + \frac{\theta}{2}\|\mathbf{x}^{t+1} - \mathbf{x}^t\|_2^2 \leq F(\mathbf{u}) + \frac{\theta}{2}\|\mathbf{u} - \mathbf{x}^t\|_2^2$ for all $\mathbf{u}_{\bar{B}} = (\mathbf{x}^t)_{\bar{B}}$. Letting $\mathbf{u} = \mathbf{x}^t$, we obtain the sufficient decrease condition:

$$F(\mathbf{x}^{t+1}) \leq F(\mathbf{x}^t) - \frac{\theta}{2}\|\mathbf{x}^{t+1} - \mathbf{x}^t\|^2 \tag{14}$$

Taking the expectation of B for the sufficient descent inequality, we have $\mathbb{E}[F(\mathbf{x}^{t+1})] \leq F(\mathbf{x}^t) - \mathbb{E}[\frac{\theta}{2}\|\mathbf{x}^{t+1} - \mathbf{x}^t\|]$. Summing this inequality over $i = 0, 1, 2, ..., t-1$, we have: $\frac{\theta}{2}\sum_{i=0}^{t}\mathbb{E}[\|\mathbf{x}^{i+1} - \mathbf{x}^i\|_2^2] \leq F(\mathbf{x}^0) - F(\mathbf{x}^t)$.
Using the fact that $F(\bar{\mathbf{x}}) \leq F(\mathbf{x}^t)$, we obtain:

$$\begin{aligned} \min_{i=1,...,t}\mathbb{E}[\tfrac{\theta}{2}\|\mathbf{x}^{i+1} - \mathbf{x}^i\|_2^2] &\leq \tfrac{\theta}{2t}\textstyle\sum_{i=0}^{t}\mathbb{E}[\|\mathbf{x}^{i+1} - \mathbf{x}^i\|_2^2] \\ &\leq \tfrac{F(\mathbf{x}^0) - F(\bar{\mathbf{x}})}{t} \end{aligned} \tag{15}$$

Therefore, we have $\lim_{t\to\infty}\mathbb{E}[\|\mathbf{x}^{t+1} - \mathbf{x}^t\|] = 0$.

(b) We assume that the stationary point is not a block-k stationary point. In expectation there exists a block of coordinates B such that $\mathbf{x}^t \notin \arg\min_{\mathbf{z}} \mathcal{P}(\mathbf{z}; \mathbf{x}^t, B)$ for some B, where $\mathcal{P}(\cdot)$ is defined in Definition 3. However, according to the fact that $\mathbf{x}^t = \mathbf{x}^{t+1}$ and subproblem (2) in Algorithm 1, we have $\mathbf{x}^{t+1} \in \arg\min_{\mathbf{z}} \mathcal{P}(\mathbf{z}; \mathbf{x}^t, B)$. Hence, we have $\mathbf{x}_B^t \neq \mathbf{x}_B^{t+1}$. This contradicts with the fact that $\mathbf{x}^t = \mathbf{x}^{t+1}$ as $t \to \infty$. We conclude that $\mathbf{x}^t$ converges to the block-k stationary point.

□

10 PROOF OF THEOREM 2

PROOF. **(a)** Note that Algorithm 1 solves problem (2) in every iteration. Using Proposition 1, we have that the solution $\mathbf{x}_B^{t+1}$ is also a L-stationary point. Therefore, we have $|\mathbf{x}^{t+1}|_i \geq \sqrt{2\lambda/(\theta + L)}$ for all $\mathbf{x}_i^{t+1} \neq 0$. Taking the initial point of $\mathbf{x}$ for consideration, we have that:

$$|\mathbf{x}_i^{t+1}| \geq \min(|\mathbf{x}_i^0|, \sqrt{2\lambda/(\theta + L)}),\ \forall\ i = 1,\ 2, ..., n.$$

Therefore, we have: $\|\mathbf{x}^{t+1} - \mathbf{x}^t\|_2 \geq \delta$. Taking the expectation of B, we have the following results: $\mathbb{E}[\|(\mathbf{x}^{t+1} - \mathbf{x}^t)_B\|_2^2] = \frac{k}{n}\|\mathbf{x}^{t+1} - \mathbf{x}^t\|_2^2 \geq \frac{k}{n}\delta^2$. Every time the support set of $\mathbf{x}$ is changed, the objective value is decreased at least by $\mathbb{E}[\frac{\theta}{2}\|\mathbf{x}^{t+1} - \mathbf{x}^t\|^2] \geq \frac{k\theta\delta^2}{2n} \triangleq D$. Combining with the result in (15), we obtain: $\frac{[2F(\mathbf{x}^0) - 2F(\bar{\mathbf{x}})]}{t\theta} \geq \frac{\delta^2 k}{n}$. Therefore, the number of iterations is upper bounded by $\bar{J}$.

(b) We notice that when the support set is fixed, the original problem reduces to a convex problem. Since the algorithm solves the following subproblem: $\mathbf{x}^{t+1} \Leftarrow \arg\min_{\mathbf{z}}\ f(\mathbf{z}) + \frac{\theta}{2}\|\mathbf{z} - \mathbf{x}^t\|^2,\ s.t.\ \mathbf{z}_{\bar{B}} = \mathbf{x}_{\bar{B}}^t$, we have the following optimality condition for $\mathbf{x}^{t+1}$:

$$(\nabla f(\mathbf{x}^{t+1}))_B + \theta(\mathbf{x}^{t+1} - \mathbf{x}^t)_B = \mathbf{0},\ (\mathbf{x}^{t+1})_{\bar{B}} = (\mathbf{x}^t)_{\bar{B}}, \tag{16}$$

We now consider the case when $f(\cdot)$ is generally convex. We derive the following inequalities:

$$\begin{aligned} & \mathbb{E}[F(\mathbf{x}^{t+1})] - F(\bar{\mathbf{x}}) \\ \overset{(a)}{\leq}\ & \mathbb{E}[\langle \nabla f(\mathbf{x}^{t+1}), \mathbf{x}^{t+1} - \bar{\mathbf{x}}\rangle], \\ \overset{(b)}{\leq}\ & \mathbb{E}[\tfrac{n}{k}\langle (\nabla f(\mathbf{x}^{t+1}))_B, (\mathbf{x}^{t+1} - \bar{\mathbf{x}})_B\rangle], \\ \overset{(c)}{=}\ & \mathbb{E}[\tfrac{n}{k}\langle -\theta(\mathbf{x}^{t+1} - \mathbf{x}^t)_B, (\mathbf{x}^{t+1} - \bar{\mathbf{x}})_B\rangle] \\ \overset{(d)}{\leq}\ & \mathbb{E}[\tfrac{n}{k}\theta\|(\mathbf{x}^{t+1} - \mathbf{x}^t)_B\|_2 \cdot \|(\mathbf{x}^{t+1} - \bar{\mathbf{x}})_B\|_2] \\ \overset{(e)}{=}\ & \mathbb{E}[\tfrac{n}{k}\theta\|(\mathbf{x}^{t+1} - \mathbf{x}^t)\|_2 \cdot \|(\mathbf{x}^{t+1} - \bar{\mathbf{x}})_B\|_2] \\ \overset{(f)}{\leq}\ & \mathbb{E}[\tfrac{n}{k}2\theta\rho\sqrt{k}\|\mathbf{x}^{t+1} - \mathbf{x}^t\|_2] = \mathbb{E}[\nu\|\mathbf{x}^{t+1} - \mathbf{x}^t\|_2] \end{aligned} \tag{17}$$

where step (a) uses the convexity of $F(\cdot)$; step (b) uses the fact that each block B is picked randomly with probability k/n; step (c) uses the optimality condition of $\mathbf{x}^{t+1}$ in (16); step (d) uses the Cauchy-Schwarz inequality; step (e) uses $\|(\mathbf{x}^{t+1} - \mathbf{x}^t)\|_2 = \|(\mathbf{x}^{t+1} - \mathbf{x}^t)_B\|_2$; step (f) uses $\|(\mathbf{x}^{t+1} - \bar{\mathbf{x}})_B\| \leq \sqrt{k}\|(\mathbf{x}^{t+1} - \bar{\mathbf{x}})_B\|_\infty \leq \sqrt{k}\|\mathbf{x}^{t+1} - \bar{\mathbf{x}}\|_\infty \leq \sqrt{k}(\|\mathbf{x}^{t+1}\|_\infty + \|\bar{\mathbf{x}}\|_\infty) \leq 2\rho\sqrt{k}$.

Using the result in (17) and the sufficient decent condition in (14), we derive the following results:

$$\mathbb{E}[F(\mathbf{x}^{t+1}) - F(\bar{\mathbf{x}})] \leq \mathbb{E}[\nu\sqrt{\tfrac{2}{\theta}\left(F(\mathbf{x}^t) - F(\mathbf{x}^{t+1})\right)}] \tag{18}$$

Denoting $\Delta^t \triangleq \mathbb{E}[F(\mathbf{x}^t) - F(\bar{\mathbf{x}})]$ and $C \triangleq \frac{2\nu^2}{\theta}$, we have the following inequality:

$$(\Delta^{t+1})^2 \leq C(\Delta^t - \Delta^{t+1})$$

Combining with Lemma 1, we have:

$$\mathbb{E}[F(\mathbf{x}^t) - F(\bar{\mathbf{x}})] \leq \max(\tfrac{4\nu^2}{\theta}, \sqrt{\tfrac{2\nu^2\Delta^0}{\theta}})/t$$

Therefore, we obtain the upper bound of the number of iterations to converge to a stationary point $\bar{\mathbf{x}}$ satisfying $F(\mathbf{x}^t) - F(\bar{\mathbf{x}}) \leq \epsilon$ with fixing the support set. Combing the upper bound for the number of changes $\bar{J}$ for the support set in **(a)**, we naturally establish the actual number of iterations for Algorithm 1.

(**c**) We now consider the case when $f(\cdot)$ is σ-strongly convex. We derive the following results:

$$\begin{aligned}
& \mathbb{E}[F(\mathbf{x}^{t+1}) - F(\bar{\mathbf{x}})] \\
\overset{(a)}{\leq} & \mathbb{E}[-\tfrac{\sigma}{2}\|\bar{\mathbf{x}} - \mathbf{x}^{t+1}\|_2^2 - \langle \bar{\mathbf{x}} - \mathbf{x}^{t+1}, \nabla f(\mathbf{x}^{t+1})\rangle] \\
\overset{(b)}{\leq} & \mathbb{E}[\tfrac{1}{2\sigma}\|\nabla f(\mathbf{x}^{t+1})\|_2^2] \\
\overset{(c)}{=} & \mathbb{E}[\tfrac{1}{2\sigma}\|(\nabla f(\mathbf{x}^{t+1}))_B\|_2^2 \times \tfrac{n}{k}] \\
\overset{(d)}{=} & \mathbb{E}[\tfrac{1}{2\sigma}\|\theta(\mathbf{x}^{t+1} - \mathbf{x}^t)_B\|_2^2 \times \tfrac{n}{k}] \\
= & \mathbb{E}[-\tfrac{\theta^2 n}{2\sigma k}\|\mathbf{x}^{t+1} - \mathbf{x}^t\|_2^2] \\
\overset{(e)}{\leq} & \mathbb{E}[\tfrac{\theta^2 n}{2\sigma k}\tfrac{2}{\theta}\left(F(\mathbf{x}^t) - F(\mathbf{x}^{t+1})\right)] \\
\overset{(f)}{=} & \mathbb{E}[\varpi\left([F(\mathbf{x}^t) - F(\bar{\mathbf{x}})] - [F(\mathbf{x}^{t+1}) - F(\bar{\mathbf{x}})]\right)] \quad (19)
\end{aligned}$$

where step (a) uses the strong convexity of $f(\cdot)$; step (b) uses $\forall \mathbf{x}, \mathbf{y},\ -\frac{\sigma}{2}\|\mathbf{x}\|_2^2 - \langle \mathbf{x}, \mathbf{y}\rangle \leq \frac{1}{2\sigma}\|\mathbf{y}\|_2^2$; step (c) uses $\mathbb{E}[\|\mathbf{w}_B\|_2^2] = \frac{k}{n}\|\mathbf{w}\|_2^2$; step (d) uses the optimality of $\mathbf{x}^{t+1}$; step (e) uses the sufficient condition in (14); step (f) uses $\varpi \triangleq \frac{n\theta}{k\sigma}$.

Rearranging terms for (19), we have: $\frac{\mathbb{E}[F(\mathbf{x}^{t+1}) - F(\bar{\mathbf{x}})]}{\mathbb{E}[F(\mathbf{x}^t) - F(\bar{\mathbf{x}})]} \leq \frac{\varpi}{1+\varpi} = \alpha$. Solving the recursive formulation, we obtain:

$$\mathbb{E}[F(\mathbf{x}^t) - F(\bar{\mathbf{x}})] \leq \mathbb{E}[\alpha^t[F(\mathbf{x}^0) - F(\bar{\mathbf{x}})]],$$

and it holds that $t \leq \log_\alpha\left(\frac{F(\mathbf{x}^t) - F(\bar{\mathbf{x}})}{F(\mathbf{x}^0) - F(\bar{\mathbf{x}})}\right)$ in expectation. Using similar techniques as in **(b)**, we obtain (11).

□

11 PROOF OF THEOREM 3

Proof. **(a)** We first consider the case when $f(\cdot)$ is generally convex. We denote $Z_t = \{i : \mathbf{x}_i^t = 0\}$ and known that the Z_t only changes for a finite number of times. We assume that Z_t only changes at $t = c_1, c_2, ..., c_{\bar{J}}$ and define $c_0 = 0$. Therefore, we have:

$$\begin{aligned}
Z_0 = Z_1 =, ..., Z_{-1+c_1} \neq Z_{c_1} = Z_{1+c_1} = Z_{2+c_1} =, \\
..., = Z_{-1+c_j} \neq Z_{c_j} = ... \neq Z_{c_{\bar{J}}} = ...
\end{aligned}$$

with $j = 1, ..., \bar{J}$. We denote $\bar{\mathbf{x}}^{c_j}$ as the optimal solution of the following optimization problem:

$$\min_{\mathbf{x}}\ f(\mathbf{x}),\ s.t.\ \mathbf{x}_{Z_{c_j}} = 0 \quad (20)$$

with $1 \leq j \leq \bar{J}$.

The solution $\mathbf{x}^{c_j}$ changes j times, the objective values decrease at least by jD, where D is defined in (9). Therefore, we have:

$$F(\mathbf{x}^{c_j}) \leq F(\mathbf{x}^0) - j \times D$$

Combing with the fact that $F(\bar{\mathbf{x}}) \leq F(\bar{\mathbf{x}}^{c_j})$, we obtain:

$$0 \leq F(\mathbf{x}^{c_j}) - F(\bar{\mathbf{x}}^{c_j}) \leq F(\mathbf{x}^0) - F(\bar{\mathbf{x}}) - j \times D \quad (21)$$

We now focus on the intermediate solutions $\mathbf{x}_{c_{j-1}}$, $\mathbf{x}_{1+c_{j-1}}$, ..., $\mathbf{x}_{-1+c_j}, \mathbf{x}_{c_j}$. Using part **(b)** in Theorem 2, we conclude that to obtain an accuracy such that $F(\mathbf{x}^{c_j}) - F(\bar{\mathbf{x}}^{c_j}) \leq D$, it takes at most $\max\left(\frac{4\nu^2}{\theta}, \sqrt{\frac{2\nu^2(F(\mathbf{x}^{c_j}) - F(\bar{\mathbf{x}}^{c_j}))}{\theta}}\right)/D$ iterations to converge to $\bar{\mathbf{x}}^{c_j}$, that is,

$$\begin{aligned}
c_j - c_{j-1} &\leq \max(\tfrac{4\nu^2}{\theta}, \sqrt{\tfrac{2\nu^2(F(\mathbf{x}^{c_j}) - F(\bar{\mathbf{x}}^{c_j}))}{\theta}})/D \\
&\leq \max(\tfrac{4\nu^2}{\theta}, \sqrt{\tfrac{2\nu^2(F(\mathbf{x}^0) - F(\bar{\mathbf{x}}) - j\times D)}{\theta}})/D \quad (22)
\end{aligned}$$

Summing up the inequality in (22) for $j = 1, 2, ..., \bar{J}$ and using the fact that $j \geq 1$ and $c_0 = 0$, we obtain that:

$$c_{\bar{J}} \leq \tfrac{\bar{J}}{D} \times \max(\tfrac{4\nu^2}{\theta}, \sqrt{\tfrac{2\nu^2(F(\mathbf{x}^0) - F(\bar{\mathbf{x}}) - D)}{\theta}})$$

After $c_{\bar{J}}$ iterations, Algorithm 1 becomes the proximal gradient method applied to the problem as in (20). Therefore, the total number of iterations for finding a block-k stationary point $\bar{N}_1$ is bounded by:

$$\begin{aligned}
\bar{N}_1 &\overset{(a)}{\leq} c_{\bar{J}} + \max(\tfrac{4\nu^2}{\theta}, \sqrt{\tfrac{2\nu^2[F(\mathbf{x}^{c_{\bar{J}}}) - F(\bar{\mathbf{x}})]}{\theta}})/\epsilon \\
&\overset{(b)}{\leq} c_{\bar{J}} + \max(\tfrac{4\nu^2}{\theta}, \sqrt{\tfrac{2\nu^2[F(\mathbf{x}^0) - F(\bar{\mathbf{x}}) - D]}{\theta}})/\epsilon \\
&= (\tfrac{\bar{J}}{D} + \tfrac{1}{\epsilon}) \times \max(\tfrac{4\nu^2}{\theta}, \sqrt{\tfrac{2\nu^2(F(\mathbf{x}^0) - F(\bar{\mathbf{x}}) - D)}{\theta}})
\end{aligned}$$

where step (a) uses the fact that the total number of iterations for finding a stationary point after $\mathbf{x}_{c_{\bar{J}}}$ is upper bounded by $\max(\frac{4\nu^2}{\theta}, \sqrt{\frac{2\nu^2[F(\mathbf{x}^{c_{\bar{J}}}) - F(\bar{\mathbf{x}})]}{\theta}})/\epsilon$; step (b) uses (21) and $j \geq 1$.

(b) We now discuss the case when $f(\cdot)$ is strongly convex. Using part **(c)** in Theorem 2, we have:

$$c_j - c_{j-1} \leq \log_\alpha \tfrac{D}{F(\mathbf{x}^0) - F(\bar{\mathbf{x}})}, \quad (23)$$

Summing up the inequality (23) for $j = 1, 2, ..., \bar{J}$, we obtain:

$$c_{\bar{J}} \leq \log_\alpha(\tfrac{D^{\bar{J}}}{(F(\mathbf{x}^0) - F(\bar{\mathbf{x}}))^{\bar{J}}}) = \bar{J}\log_\alpha(\tfrac{D}{(F(\mathbf{x}^0) - F(\bar{\mathbf{x}}))})$$

Therefore, the total number of iterations $\bar{N}_2$ is bounded by:

$$\bar{N}_2 \overset{(a)}{\leq} c_{\bar{J}} + \log_\alpha(\tfrac{\epsilon}{F(\mathbf{x}^{c_{\bar{J}}}) - F(\bar{\mathbf{x}})}) \overset{(b)}{\leq} c_{\bar{J}} + \log_\alpha(\tfrac{\epsilon}{F(\mathbf{x}^0) - D - F(\bar{\mathbf{x}})})$$

where step (a) uses the fact that the total number of iterations for finding a stationary point after $\mathbf{x}_{c_{\bar{J}}}$ is upper bounded by $\log_\alpha\left(\frac{\epsilon}{F(\mathbf{x}^{c_{\bar{J}}}) - F(\bar{\mathbf{x}})}\right)$; step (b) uses (21) that $0 \leq F(\mathbf{x}^0) - F(\bar{\mathbf{x}}) - t \times D$ and $t \geq 1$.

□

REPRODUCIBLE RESEARCH

We provide our code in the authors' research webpage at: https://yuangzh.github.io.