A Hybrid Method of Combinatorial Search and Gradient Descent for Discrete Optimization

Ganzhao Yuan * Li Shen † Wei-Shi Zheng ‡

Abstract

Discrete optimization is a central problem in machine learning with a broad range of applications, among which binary optimization and sparse optimization are two common ones. However, these problems are NP-hard and thus difficult to solve in general. Combinatorial search methods such as branch-and-bound and exhaustive search find the global optimal solution but are confined to small-sized problems, while gradient descent methods such as proximal gradient descent and coordinate gradient descent are efficient but often suffer from poor local minima. In this paper, we consider a hybrid method that combines the effectiveness of combinatorial search and the efficiency of gradient descent. Specifically, we randomly select a subset of coordinates and then perform global combinatorial search over the small subset of coordinates based on the gradient information of the smooth objective function. The proposed method can be viewed as a randomized block proximal Newton method. In addition, we provide some optimality analysis and convergence analysis for the proposed method. Finally, we demonstrate the efficacy of our method on some sparse optimization and binary optimization applications. As a result, our method achieves state-of-the-art performance in terms of accuracy.

1 Introduction

In this paper, we mainly focus on the following nonconvex composite minimization problem:

$$\min_{\mathbf{x}} \; F(\mathbf{x}) \triangleq f(\mathbf{x}) + h(\mathbf{x}), \; f(\mathbf{x}) \triangleq \tfrac{1}{2}\mathbf{x}^T\mathbf{Q}\mathbf{x} + \langle \mathbf{x}, \mathbf{p}\rangle, \tag{1}$$

where $\mathbf{x} \in \mathbb{R}^n$, $\mathbf{p} \in \mathbb{R}^n$, $\mathbf{Q} \in \mathbb{R}^{n\times n}$ is a symmetric (not necessarily positive semidefinite) matrix, and $h(\mathbf{x})$ is a piecewise separable function. We consider two cases for $h(\cdot)$:

$$\left\{h_{\text{binary}}(\mathbf{x}) \triangleq I_\Psi(\mathbf{x}), \; \Psi \triangleq \{-1,+1\}^n\right\} \text{ or } \left\{h_{\text{sparse}}(\mathbf{x}) \triangleq \lambda\|\mathbf{x}\|_0 + I_\Omega(\mathbf{x}), \; \Omega \triangleq [-\rho\mathbf{1}, \rho\mathbf{1}]\right\},$$

where $I_\Psi(\cdot)$ is an indicator function on Ψ with $I_\Psi(\mathbf{x}) = \left\{ \begin{smallmatrix} 0, & \mathbf{x} \in \Psi; \\ +\infty, & \mathbf{x} \notin \Psi. \end{smallmatrix} \right.$, $\|\cdot\|_0$ is a function that counts the number of nonzero elements in a vector, λ and ρ are strictly positive constants. When $h \triangleq h_{\text{binary}}$, (1) refers to the binary optimization problem; when $h \triangleq h_{\text{sparse}}$, (1) corresponds to the sparse optimization problem. We note that the solution set in (1) is compact.

Binary optimization and sparse optimization capture a variety of applications of interest in both machine learning and computer vision, including binary hashing [37, 36], dense subgraph discovery [45, 43], Markov random fields [8], compressive sensing [10, 15], sparse coding [21, 1, 2], sparse logistic regression, subspace clustering [16], to name a few. Although we mainly focus on separable $h(\cdot)$ and quadratic $f(\cdot)$ in our presentation, the proposed method can be extended to handle non-quadratic but convex $f(\cdot)$ and non-separable but simple $h(\cdot)$ as well.

Binary optimization and sparse optimization are closely related to each other. A binary optimization problem can be reformulated as a sparse optimization problem using the fact that $\mathbf{x} \in$

*School of Data & Computer Science, Sun Yat-sen University, China. Email: yuanganzhao@gmail.com.
†Tencent AI Lab, Shenzhen, China. Email: shen.li@mail.scut.edu.cn.
‡School of Data & Computer Science, Sun Yat-sen University, China. Email: wszheng@ieee.org (Corresponding Author).

$\{-1,+1\}^n \Leftrightarrow \|\mathbf{x}-\mathbf{1}\|_0+\|\mathbf{x}+\mathbf{1}\|_0 \leq n$ [42, 41], and the reverse is also true using the variational reformulation of ℓ_0 norm: $\forall \|\mathbf{x}\|_\infty \leq \rho,\ \|\mathbf{x}\|_0 = \min_{\mathbf{v}\in\{0,1\}^n}\ \langle \mathbf{1}, \mathbf{v}\rangle,\ s.t.\ |\mathbf{x}| \leq \rho\mathbf{v}$ [7]. There are generally three classes of methods for solving the binary or sparse optimization problem as in (1) in the literature, which we present below.

(**Relaxed Approximation Algorithm**) One popular method to solve (1) is (convex or nonconvex) relaxed approximation method. Box constrained relaxation, SDP relaxation and spherical relaxation are often used for solving binary optimization, while ℓ_1 norm, top-k norm, Schatten ℓ_p norm and others (such as re-weighted ℓ_1 norm, Capped ℓ_1, Half Quadratic et al.) are often used for solving sparse optimization. It is generally accepted that nonconvex method often achieves better accuracy than the convex counterpart. However, this class of method fails to directly control the sparsity or binary property of the solution and they are not the focus of this paper.

(**Combinatorial Search Algorithm**) Combinatorial search algorithm is typically concerned with problems that are NP-hard. It is a non-heuristic and global optimization algorithm in nonconvex optimization. A naive method is exhaustive search (a.k.a generate and test method). It systematically enumerates all possible candidates for the solution and pick the best candidate that leads to the lowest objective value. The cutting plane method [13] solves the convex linear programming relaxation and adds linear constraints to drive the solution towards binary variables. The branch-and-cut method works by performing branches and applying cuts at the nodes of the tree having a lower bound that is worse than the current solution. It maintains a probable upper bound and lower bound on the globally optimal objective value and terminates with some certificate. However, both the cutting plane method and branch-and-cut method could also be slow. Although in some cases we are lucky and these methods converge with much less effort, in the worse case they end up solving all 2^n convex subproblems.

(**Proximal Gradient Descent Algorithm**) Another popular method is the proximal gradient descent [3, 24, 19, 29, 30, 31, 22]. Based on the current gradient $\nabla f(\mathbf{x}^k)$, it works by iteratively performing a gradient update followed by a hard proximal/thresholding operation: $\mathbf{x}^{k+1} = \text{prox}_{\gamma h}(\mathbf{x}^k - \gamma\nabla f(\mathbf{x}^k))$. Here the proximal operator $\text{prox}_{\tilde{h}}(\mathbf{a}) = \arg\min_{\mathbf{x}}\ \frac{1}{2}\|\mathbf{x}-\mathbf{a}\|_2^2 + \tilde{h}(\mathbf{x})$ can be evaluated analytically, $\gamma = 1/L$ is the step size with L being the local (or global) Lipschitz constant. We remark that this algorithm is closed related to coordinate descent [27, 11, 9], block coordinate descent [14, 33, 5, 18, 25, 40], asynchronous parallel stochastic coordinate descent[23, 34], accelerated proximal gradient method [12], and mini-batch variance reduction gradient descent [46] in the literature. However, most of existing methods use first-order gradient-descent type iterations and a constant Lipschitz step size. They work by solving a first-order majorization/surrogate function via closed form updates. When the problem is nonconvex, this scaled identity quadratic function may not necessarily be a good majorization/surrogate function for the original problem.

Above all, the proposed method is also related to the recent matrix splitting method for composite function minimization [44]. Incorporating with a new triangle proximal operator procedure, the matrix splitting method achieves state-of-the-art performance. One good merit of this method is to explore the specific second-order information of the problem. Inspired by this observation, we propose a hybrid method which can better capture the second-order information of the optimization problem. It combines the effectiveness of combinatorial search and the efficiency of gradient descent, leading to a more accurate and practical method for discrete optimization.

The contributions of this paper are three-fold. (i) Algorithmically, we introduce a novel hybrid method (denoted as HYBRID) for sparse or binary optimization which combines the effectiveness of combinatorial search and the efficiency of gradient descent (See Section 2). (ii) Theoretically, we establish the optimality hierarchy and the convergence rate of our proposed algorithm (See Section 3). Our algorithm finds a stronger stationary point than existing methods. (iii) Empirically, we have conducted extensive experiments on some synthetic and real data to show the superiority of our method (See Section 4).

2 Proposed Algorithm

In this section, we present our hybrid method for solving the optimization problem in (1). It can be viewed as a randomized block proximal Newton method. For any partition of the index vector $[1, 2, ..., n]$ into $[B_1|B_2|...|B_m]$ with $B_i \in \mathbb{N}^{n_i},\ \forall i = 1, ..., m$, the $n \times n$ identity matrix can be partitioned as $\mathbf{I} = [\mathbf{U}_1, ..., \mathbf{U}_m]$, where $\mathbf{U}_i \in \mathbb{R}^{n \times n_i}$, such that $\mathbf{x} = \sum_{i=1}^{m} \mathbf{U}_i \mathbf{x}_{B_i},\ \mathbf{x}_{B_i} = (\mathbf{U}_i)^T \mathbf{x},\ i = 1, ..., m$. Sometimes, we use $\mathbf{x}_i$ to denote $\mathbf{x}_{B_i}$ for brevity.

Following the approach of [35], we keep the non-smooth non-convex function $h(\mathbf{z})$ and build a quadratic Newton approximation around any solution $\mathbf{x}^t$ for the objective function $f(\mathbf{x})$ by considering its second-order Taylor expansion:

$$T(\mathbf{z}, \mathbf{x}^t) \triangleq \tfrac{1}{2}(\mathbf{z} - \mathbf{x}^t)^T \nabla^2 f(\mathbf{x}^t)(\mathbf{z} - \mathbf{x}^t) + \langle \mathbf{z} - \mathbf{x}^t, \nabla f(\mathbf{x}^t) \rangle + h(\mathbf{z}) \tag{2}$$

where $\nabla f(\mathbf{x}^t)$ and $\nabla^2 f(\mathbf{x}^t)$ denote the first-order and second-order gradient of f, respectively. Each time a single block variable is optimized while the rest of the variables remain fixed. The proposed method is summarized in Algorithm 1.

Algorithm 1 A Hybrid Approach for Sparse or Binary Optimization

1: Input k as the size of the working set. Find $\mathbf{x}^0$ as the initial feasiable solution. Set $t = 0$.
2: **while** not converge **do**
3: (S1) Sample i from $\{1, ..., m\}$ with equal probability. Denote $B = B_i,\ N = \{1, ..., n\} \setminus B$.
4: (S2) Solve the following subproblem *globally* using combinatorial search:

$$\mathbf{x}^{t+1} \Leftarrow \arg\min_{\mathbf{z}}\ T(\mathbf{z}, \mathbf{x}^t) + \tfrac{\theta}{2}\|\mathbf{z} - \mathbf{x}^t\|^2,\ s.t.\ \mathbf{z}_N = \mathbf{0} \tag{3}$$

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

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

Two New Ingredients. The proposed algorithm has two new ingredients. (i) Instead of using majorization techniques for optimizing the block of variables, we consider the original second-order gradient information. Although the subproblem is NP-hard and admits no closed form solution, we consider an exhaustive search to solve it. (ii) We introduce a new proximal point strategy for the subproblem. This is to guarantee sufficient decrease condition and global convergence of the proposed algorithm. The introduction of proximal point strategies guarantees sufficient decrease of the objective and global convergence of the algorithm (refer to Lemma 1 and Theorem 2).

Solving the Subproblem Globally. Problem (3) in Algorithm 1 is equivalent to the following small-sized composite optimization problem: $\min_{\mathbf{z}_B}\ \frac{1}{2}(\mathbf{z}_B - \mathbf{x}_B^t)^T \mathbf{Q}_{B,B}(\mathbf{z}_B - \mathbf{x}_B^t) + \langle (\mathbf{z}_B - \mathbf{x}_B^t), (\nabla f(\mathbf{x}^t))_B \rangle + h(\mathbf{z}_B) + \frac{\theta}{2}\|\mathbf{z}_B - \mathbf{x}_B^t\|_2^2$. By rearranging terms we obtain the following equivalent optimization problem:

$$\min_{\mathbf{z} \in \mathbb{R}^k}\ \tfrac{1}{2}\mathbf{z}^T \mathbf{A}\mathbf{z} + \langle \mathbf{z}, \mathbf{b} \rangle + h(\mathbf{z}) \tag{4}$$

with $\mathbf{A} = \mathbf{Q}_{B,B} + \theta\mathbf{I}$ and $\mathbf{b} = \mathbf{Q}_{B,B}\mathbf{x}_B^t - (\nabla f(\mathbf{x}^t))_B - \theta\mathbf{x}_B^t$. We remark that for $h = h_{\text{sparse}}$, it can be reformulated as a mixed-integer optimization problem: $\min_{\|\mathbf{z}\|_\infty \le \rho,\ \mathbf{y} \in \{0,1\}^k} \frac{1}{2}\mathbf{z}^T \mathbf{A}\mathbf{z} + \langle \mathbf{z}, \mathbf{b} \rangle + \lambda \mathbf{y}^T \mathbf{1},\ s.t.\ |\mathbf{z}| \le \rho\mathbf{y}$, which can be solved by branch-and-bound solver such as CPLEX. For simplicity, we use a simple exhaustive search method to solve it. For every variable $\mathbf{z}_i, i = 1, 2, ..., k$, it has two states, i.e., $-1/+1$ and zero/nonzero. We systematically enumerate the full binary tree for $\mathbf{z}$ to obtain all possible candidate solutions and then pick the best one that leads to the lowest objective value as the optimal solution.

Finding a Working Set. We observe that it contains C_n^k possible combinations of choice for the working set. Although in our algorithm design and theoretical analysis, we assume the partition blocks $\{B_1, B_2, ..., B_m\}$ are predefined. However, to better capture the second-order information of

problem (1), it is beneficial to change the partition block and sample one block coordinate uniformly in every iteration.

Computational Iteration Complexity. In every iteration, one needs to solve a NP-hard problem of size k which take $\mathcal{O}(2^k)$. Computing the partial gradient and partial Hessian take nk and k^2, respectively. We assume that it takes $\#it$ for Algorithm 1 to converge. The total time computational complexity of Algorithm 1 is $\#it \times (nk + \mathcal{O}(2^k))$.

Extensions to General Problems. (i) Our algorithm can solve composite discrete optimization problem even when $f(\cdot)$ is a non-quadratic but convex function. Since the second-order Taylor series is a majorizer of any twice differentiable convex function, one can minimize (approximately) the upper bound of the non-quadratic function $q(\mathbf{y})$ (i.e. the quadratic surrogate function) at the current estimate $\mathbf{x}$ and drive the objective downward until a stationary point is reached. (ii) The proposed algorithm can still be applied when $h(\mathbf{x})$ contains linear/nonlinear non-separable constraint. What one needs is to ensure that the next solution $\mathbf{x}^{t+1}$ is also within the non-separable constraint. Sufficient decrease condition and global convergence can still be guaranteed.

3 Theoretical Analysis

This section establishes the optimality hierarchy and the convergence rate of our proposed algorithm.

3.1 Optimality Analysis

In the sequel, we present some necessary optimality conditions for (1).

Definition 1. *(Basic Stationary Point) We denote L as the global gradient Lipschitz constant of $f(\cdot)$ and $S \triangleq supp(\mathbf{x}^*)$. A vector $\mathbf{x}^*$ is called an basic stationary point if the following condition holds:*

$$h_{binary} : \mathbf{x}^* \in \{-1, +1\}^n;\ h_{sparse} : \mathbf{x}^*_S = \arg\min_{\mathbf{z}\in\Omega} \tfrac{L}{2}\|\mathbf{z} - (\mathbf{x}^* - \nabla f(\mathbf{x}^*)/L)_S\|_2^2 \tag{5}$$

Remarks: For binary optimization, any binary solution is a basic stationary point. For sparse optimization, basic stationary point states that the solution obtains the global optimal when the support set is restricted. One remarkable feature of the basic stationary condition is that the solution set is enumerable and its size is 2^n. It makes it possible to validate whether a solution is optimal for the original discrete optimization problem.

Definition 2. *(L-Stationary Point) We denote L as the global gradient Lipschitz constant of $f(\cdot)$. A vector $\mathbf{x}^*$ is called an L-stationary point if the following condition holds:*

$$\mathbf{x}^* = \arg\min_{\mathbf{z}} \ \tfrac{L}{2}\|\mathbf{z} - (\mathbf{x}^* - \nabla f(\mathbf{x}^*)/L)\|_2^2 + h(\mathbf{z}) \tag{6}$$

Remarks: This is the well-known proximal hardsholding operator. Previous studies use first-order gradient-descent type iterations and a constant Lipschitz step size. Due to non-convexity, this scaled identity quadratic function may not necessarily be a good majorization/surrogate function for solving problem (1).

Definition 3. *(Block-k Stationary Point) Assume that $\mathbf{x}^*$ is the global optimal solution. The following always holds for all B with $|B| = k$:*

$$\mathbf{x}^*_B \in \arg\min_{\mathbf{z}} \ \tfrac{1}{2}(\mathbf{z} - \mathbf{x}^*_B)^T(\nabla^2 f(\mathbf{x}^*))_{BB}(\mathbf{z} - \mathbf{x}^*_B) + \langle \mathbf{z} - \mathbf{x}^*_B, (\nabla f(\mathbf{x}^*))_B\rangle + h(\mathbf{z}) \tag{7}$$

Remarks: Block-k stationary point is novel in this paper. It involves solving a small-sized NP-hard problem as in (7). However, this problem can be solved by some practical global optimization methods.

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

Theorem 1. ***Proof of the Hierarchy between the Necessary Optimality Condition.*** *We have the following optimality hierarchy:* $\boxed{\textit{Basic Stat. Point}} \overset{(1)}{\Leftarrow} \boxed{\textit{L-Stat. Point}} \overset{(2)}{\Leftarrow} \boxed{\textit{Block-1 Stat. Point}} \overset{(3)}{\Leftarrow} \boxed{\textit{Block-2 Stat. Point}} \Leftarrow \ldots \Leftarrow \boxed{\textit{Block-}n\textit{ Stat. Point}} \overset{(4)}{\Leftrightarrow} \boxed{\textit{Optimal Point}}$.

Proof. (1) We now prove that L-stationary point implies basic stationary point. For binary optimization, this conclusion clearly holds. We now consider sparse optimization (i.e. $h = h_{\text{sparse}}$). Since the optimization problem in (5) is separable, we have the following closed form solution: $\mathbf{x}_S^* = \min(\rho, \max(-\rho, (\mathbf{x}^* - \nabla f(\mathbf{x}^*)/L)_S))$. For the optimization in (6), we have the following closed form solution: $\mathbf{x}_i^* = \begin{cases} \min(\rho, \max(-\rho, (\mathbf{x}_i^* - \nabla_i f(\mathbf{x}^*)/L))), & (\mathbf{x}_i^* - \nabla_i f(\mathbf{x}^*)/L)^2 \geq 2\lambda; \\ 0, & \text{else.} \end{cases}$. Clearly, the latter formulation implies the former one.

(2) Note that the optimization problem in (7) is separable when $k = 1$. For $h = h_{\text{binary}}$, we have the following result: $\mathbf{x}_i^* = \begin{cases} 1, & \mathbf{x}_i^* - \nabla_i f(\mathbf{x}^*)/\mathbf{Q}_{i,i} \geq 0; \\ -1, & \text{else.} \end{cases}$. For $h = h_{\text{sparse}}$, we have the following result: $\mathbf{x}_i^* = \begin{cases} \min(\rho, \max(-\rho, \mathbf{x}_i^* - \nabla_i f(\mathbf{x}^*)/\mathbf{Q}_{i,i})), & (\mathbf{x}_i^* - \nabla_i f(\mathbf{x}^*)/\mathbf{Q}_{i,i})^2 \geq 2\lambda; \\ 0, & \text{else.} \end{cases}$. Since $\mathbf{Q}_{i,i} \leq L$ for all i, we conclude that block-1 stationary point implies L-stationary point.

(3) We now show that block-k_1 stationary point implies block-k_2 stationary point when $k_1 \geq k_2$. Note that to guarantee block-k stationary condition, one need to solve the problem in (7) for $\sum_{i=0}^{k} C_n^k$ times, i.e. all the combination which is at most of size k. Clearly, when $k_1 \geq k_2$, the subproblem for block-k_2 stationary point is a subset of that of block-k_1 stationary point.

(4) When $B = \{1, 2, ..., n\}$, we have $\min_{\mathbf{z}} \frac{1}{2}(\mathbf{z}-\mathbf{x}^*)^T\mathbf{Q}(\mathbf{z}-\mathbf{x}^*) + \langle \mathbf{z}-\mathbf{x}^*, \mathbf{Q}\mathbf{x}^* + \mathbf{p}\rangle + \lambda\|\mathbf{z}\|_0$. Rearranging terms, we have that this optimization problem is completely equivalent to the original problem as in (1).

□

Remarks: It is worthwhile to point out that the seminal work of [3] also presents optimality conditions for sparse optimization. However, our block-k condition is stronger than their coordinate-wise optimality condition since their optimal condition corresponds to $k = 1$ in our optimality condition framework.

A Running Example. We consider the quadratic optimization problem $\min_{\mathbf{x}\in\mathbb{R}^n} \frac{1}{2}\mathbf{x}^T\mathbf{Q}\mathbf{x} + \langle \mathbf{x}, \mathbf{p}\rangle + h(\mathbf{x})$ with $n = 6$, where $\mathbf{Q} = \mathbf{c}\mathbf{c}^T + \mathbf{I}$, $\mathbf{p} = \mathbf{1}$, $\mathbf{c} = [1\ 2\ 3\ 4\ 5\ 6]^T$. We consider two cases for $h(\cdot)$, $h_{\text{sparse}}(\mathbf{x})$ and $h_{\text{binary}}(\mathbf{x})$. The parameters for $h_{\text{sparse}}(\mathbf{x})$ are set to $\lambda = 0.01$ and $\rho = +\infty$. The local minima distribution on this example can be found in Table 1. This problem contains $\sum_{i=0}^{6} C_6^i = 64$ local minima. 56 and 58 local minimizers satisfy the L-stationary condition for binary optimization and sparse optimization problem, respectively. 9 and 11 local minimizers satisfy the block-1 stationary condition. Moreover, as k becomes large, the newly introduced classes of local minimizers (i.e. block-k stationary point) become more restricted in the sense that they have small number of local minima.

	basic-Stat.	L-Stat.	Block-1 Stat.	Block-2 Stat.	Block-3 Stat.	Block-4 Stat.	Block-5 Stat.	Block-6 Stat.
$h \triangleq h_{\text{binary}}$	64	56	9	3	1	1	1	1
$h \triangleq h_{\text{sparse}}$	64	58	11	2	1	1	1	1

Table 1: Points satisfying optimality conditions.

3.2 Convergence Analysis

In this subsection, we provide some convergence analysis for Algorithm 1. All proofs can be found in the **Appendix**.

The following sufficient decrease condition is useful in our proof.

Lemma 1. ***(Sufficient Decrease Condition)*** *Assume B is the working set at the tth iteration. Suppose $\{F(x^t)\}_{t=1}^n$ is generated by Algorithm 1, the following inequality holds:*

$$F(\mathbf{x}^{t+1}) - F(\mathbf{x}^t) \leq -\frac{\theta}{2}\|\mathbf{x}_B^{t+1} - \mathbf{x}_B^t\|^2 \tag{8}$$

Remarks: The introduction of the strongly convex parameter $\theta > 0$ is necessary for our nonconvex optimization problem since it guarantees sufficient decrease condition which is very important for convergence.

Theorem 2. ***Proof of Convergence for $h = h_{sparse}$ or $h = h_{binary}$.*** *Let $\mathbf{x}^t$ be the sequence generated by Algorithm 1. We have the following results. (i) It holds that $\lim_{t\to\infty} \mathbb{E}[\|\mathbf{x}^{t+1} - \mathbf{x}^t\|] = 0$. Moreover, there exists a scalar F^* such that $\lim_{t\to\infty} \mathbb{E}[F(\mathbf{x}^t)] = F^*$. (ii) When $h = h_{binary}$, we have $\mathbb{E}[\|\mathbf{x}^{t+1} - \mathbf{x}^t\| \mid \mathbf{x}^t] \geq \sqrt{2}/m$ if $\mathbf{x}^t \neq \mathbf{x}^{t+1}$ for all $t \geq 0$. The current solution changes at most $\sqrt{2}m(F(\mathbf{x}^0) - F(\mathbf{x}^*))/\theta$ times in expectation. (iii) When $h = h_{sparse}$, we have $i \in supp(\mathbf{x}^t)$, $|\mathbf{x}_i^t| \geq \delta$ for all $t \geq 0$, where $\delta \triangleq \min_j \{\min(\rho, \mathbf{x}_j^0, \sqrt{2\lambda/(\theta + \mathbf{Q}_{j,j})})\}$. Moreover, it holds that: $\mathbb{E}[\|\mathbf{x}^{t+1} - \mathbf{x}^t\| \mid \mathbf{x}^t] \geq \delta/m$ if $\mathbf{x}^t \neq \mathbf{x}^{t+1}$. The index set changes at most $\frac{2m(F(\mathbf{x}^0)-F(\mathbf{x}^*))}{\theta\delta}$ times in expectation.*

In what follows, we study the convergence rate of Algorithm 1 for binary optimization (i.e. $h \triangleq h_{\text{binary}}$). We define $\Pi(\mathbf{a}) = \arg\min_{\mathbf{x}} \|\mathbf{x} - \mathbf{a}\|,\ s.t.\ \mathbf{x} \in \{-1, +1\}^n$. Our key observation is that when $\Pi(\mathbf{x}) \neq \Pi(\mathbf{y})$, it holds $\|\Pi(\mathbf{x}) - \mathbf{x}\|_2 \leq (1-\kappa)\|\Pi(y) - \mathbf{x}\|_2$ for some constant $\kappa > 0$. Our analysis combines the above observation with the strongly convex property of $f(\cdot)$ to provide Q-linear convergence rate for Algorithm 1.

Theorem 3. ***Proof of Convergence Rate when $f(\cdot)$ is s-Strongly Convex and $h \triangleq f_{binary}$.*** *Let $\mathbf{x}^t$ be the sequence generated by Algorithm 1. We have the following result: $E[f(\mathbf{x}^{k+1})|\mathbf{x}^k] - f(\mathbf{x}^*) \leq (1-C)(f(\mathbf{x}^k) - f(\mathbf{x}^*))$, where $C \triangleq (\frac{1}{2L}\sqrt{(L-s)(1-\kappa)\kappa} + 1 - \kappa)/m$. Moreover, it takes at most $\log_{(1-C)}(\frac{\epsilon}{F(\mathbf{x}^0)-F(\mathbf{x}^*)})$ times to find a local optimal solution satisfying $F(\mathbf{x}^k) - F(\mathbf{x}^*) \leq \epsilon$. Here $L \triangleq \max_{i=1}^m \|\mathbf{U}_i^T \mathbf{Q} \mathbf{U}_i\|$.*

Remarks: Notice that the assumption that $f(\cdot)$ is strongly convex always holds for binary optimization. This is because one can append an additional term $\frac{\eta}{2}\|\mathbf{x}\|_2^2$ to the objective function $f(\cdot)$ and the objective function becomes strongly convex with sufficiently large η.

In what follows, we establish the convergence rate for sparse optimization. In Theorem 2, we have derived the bound of the number of change for the support set. Now we need to derive a bound on the number of iterations performed after the support is fixed. Combining these two bounds, we complete our proof.

Theorem 4. ***Proof of Convergence Rate when $f(\cdot)$ is Convex and $h = f_{sparse}$.*** *Algorithm 1 at most takes $(m(0.5\|\mathbf{x}^0 - \mathbf{x}^*\|_2^2(L+\theta)^2 + F(\mathbf{x}^0) - F(\mathbf{x}^*))/\epsilon - m - 1) \cdot (2mF(\mathbf{x}^0) - 2mF(\mathbf{x}^*))/(\theta\delta))$ iterations in expectation to find a local optimal solution satisfying $F(\mathbf{x}^k) - F(\mathbf{x}^*) \leq \epsilon$. Here $L \triangleq \max_{i=1}^m \|\mathbf{U}_i^T \mathbf{Q} \mathbf{U}_i\|$.*

4 Experimental Validation

In this section, we demonstrate the effectiveness of our proposed algorithm (denoted as HYBRID) on 5 discrete optimization tasks, namely sparse regularized least square problem, binary constrained least square problem, dense subgraph discovery, sparse constrained least square problem, and sparse PCA (Principle Component Analysis). Unless otherwise specified, we use the default parameters ($k = 30$ and $\theta = 0.01$) for Algorithm 1 throughout our experiments. All codes are implemented in Matlab on an Intel 3.20GHz CPU with 8 GB RAM.

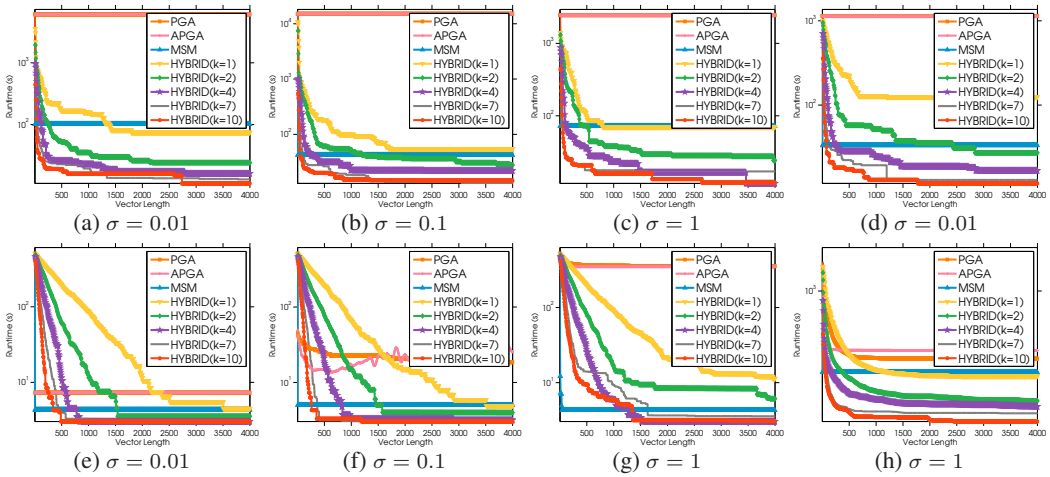
4.1 Sparse Regularized / Binary Constrained Least Squares Problem

Given a design matrix $\mathbf{A} \in \mathbb{R}^{p\times n}$ and an observation vector $\mathbf{b} \in \mathbb{R}^m$, sparse regularized / binary constrained least squares problem involves solving the following optimization problem:

$$\left\{\min_{\mathbf{x}} \tfrac{1}{2}\|\mathbf{A}\mathbf{x} - \mathbf{b}\|_2^2,\ s.t.\ \mathbf{x} \in \{-1, +1\}^n\right\} \text{ or } \left\{\min_{\mathbf{x}} \tfrac{1}{2}\|\mathbf{A}\mathbf{x} - \mathbf{b}\|_2^2 + \lambda\|\mathbf{x}\|_0,\ s.t.\ \|\mathbf{x}\|_\infty \leq \rho\right\} \quad (9)$$

In our experiments, we generate $\mathbf{A} \in \mathbb{R}^{50\times 500}$ and $\mathbf{b} \in \mathbb{R}^{50}$ from a (0-1) uniform distribution. For sparse optimization, we set $\lambda = 1$ and $\rho = +\infty$.

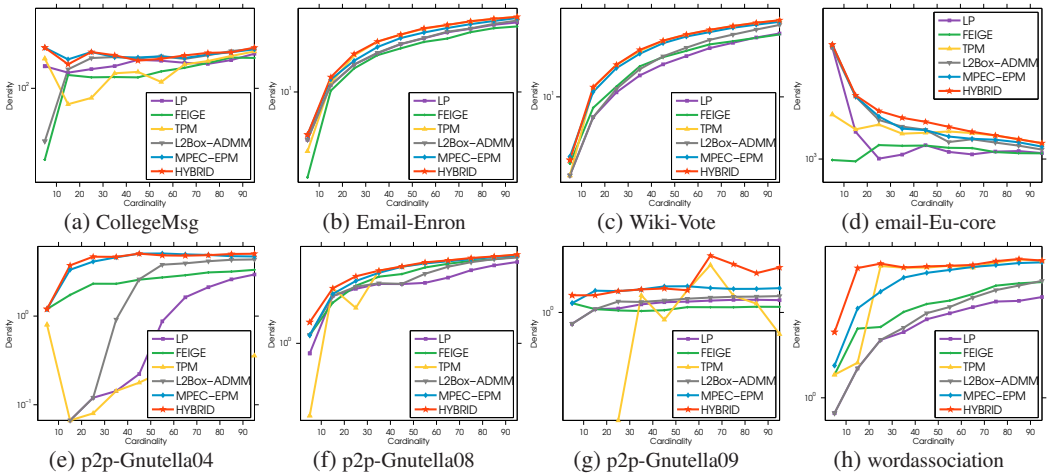
Compared Methods. We compare the proposed method (HYBRID) with three state-of-the-art methods: (i) proximal gradient algorithm (PGA) [28], (ii) accelerated proximal gradient algorithm(APGA) [28, 4], (iii) matrix splitting method (MSM) [44].



(a) $\sigma = 0.01$ (b) $\sigma = 0.1$ (c) $\sigma = 1$ (d) $\sigma = 0.01$

(e) $\sigma = 0.01$ (f) $\sigma = 0.1$ (g) $\sigma = 1$ (h) $\sigma = 1$

Figure 1 First Row: Binary Constrained Optimization Problem with $h = h_{\text{binary}}$; Second Row: Sparse Regularized Optimization Problem with $h = h_{\text{sparse}}$. Convergence behavior for solving the sparse regularized least square optimization problem with different initializations. Denoting $\tilde{\mathbf{o}}$ as an arbitrary standard Gaussian random matrix of suitable size, we consider the following strategy for different initiations $\mathbf{x}$. First Row: $\mathbf{x} = sign(\sigma \times \tilde{\mathbf{o}})$. Second Row: $\mathbf{x} = \sigma \times \tilde{\mathbf{o}}$.



(a) CollegeMsg (b) Email-Enron (c) Wiki-Vote (d) email-Eu-core

(e) p2p-Gnutella04 (f) p2p-Gnutella08 (g) p2p-Gnutella09 (h) wordassociation

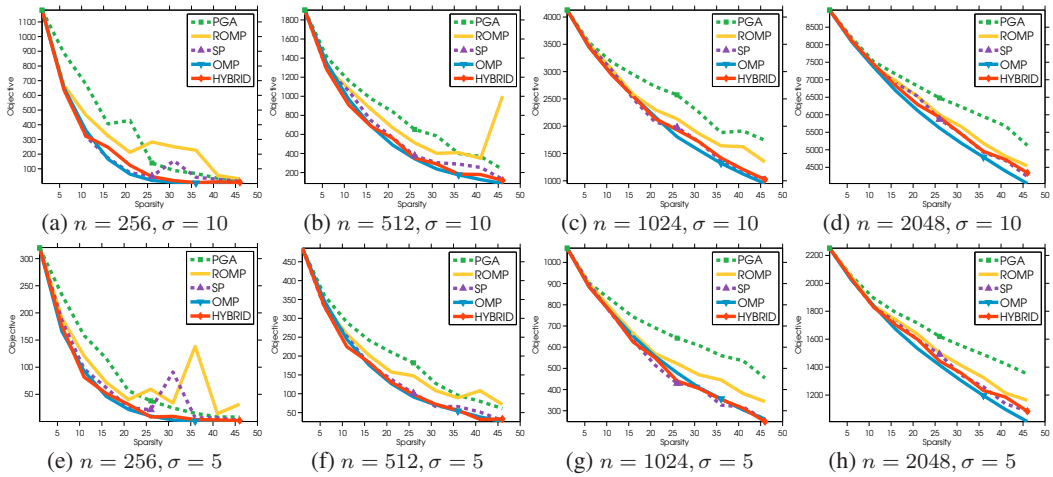
Figure 2: Experimental results on dense subgraph discovery.

Experimental Results. Several observations can be drawn from Figure 1. (i) Classical proxima point method and accelerated proximal gradient method achieves similar performance and they lead to poor accuracy. MSM significantly improves over PGA and APGA. This is consistent with the findings in the work of [44]. (iii) Our proposed block-k hybrid method is more effective than the proximal method. In addition, we find that as the parameter k becomes larger, more higher accuracy is achieved. (iii) The proposed method HYBRID appears to be less sensitive to their initializations and it converges to similar objective values.

4.2 Binary Optimization Application: Dense Subgraph Discovery

Dense subgraphs discovery [32, 17, 45] aims at finding the maximum density subgraph on k vertices, which can be formulated as the following binary program:

$$\min_{\mathbf{x}\in\{0,1\}^n} \ \mathbf{x}^T(\lambda\mathbf{I} - \mathbf{W})\mathbf{x}, \ s.t. \ \mathbf{x}^T\mathbf{1} = k \tag{10}$$



(a) $n = 256, \sigma = 10$ (b) $n = 512, \sigma = 10$ (c) $n = 1024, \sigma = 10$ (d) $n = 2048, \sigma = 10$

(e) $n = 256, \sigma = 5$ (f) $n = 512, \sigma = 5$ (g) $n = 1024, \sigma = 5$ (h) $n = 2048, \sigma = 5$

Figure 3: Convergence behavior for solving the sparse constrained least squares optimization problem.

where $\mathbf{W} \in \mathbb{R}^{n \times n}$ is the adjacency matrix of the graph. We set $\lambda = \|\mathbf{W}\|$ to ensure that the objective function is convex. This is equivalent to adding a constant to the objective since $\lambda \mathbf{x}^T \mathbf{x} = \lambda k$ in the effective domain. Although Algorithm 1 is not ready to solve the problem in (10) that contains non-separable structure. However, following the work of [26], we can ensure that the current solution always satisfies the non-separable constraint in every iteration. Sufficient decrease condition, global convergence, and block-k global optimality can still be guaranteed.

Compared Methods. We compare our HYBRID method on 8 datasets (CollegeMsg, Email-Enron, Wiki-Vote, email-Eu-core, p2p-Gnutella04, p2p-Gnutella08, p2p-Gnutella09, wordassociation) [1] against 6 methods: (i) Feige's greedy algorithm (GEIGE) [17]. (ii) LP relaxation [43]. (iii) L2box-ADMM [38]. (iv) Truncated Power Method (TPM) [2] [45]. (v) MPEC-EPM [43]. For more description of these methods, we refer to [43, 41].

Experimental Results. Several observations can be drawn from Figure 2. (i) FEIGE generally fails to solve the dense subgraph discovery problem and it leads to solutions with low density. (ii) LP relaxation gives better performance than state-of-the-art technique TPM in some cases. (iii) L2box-ADMM outperforms LP relaxation for all cases. (iv) Our proposed HYBRID outperforms all the compared methods.

4.3 Sparse Optimization Application: Sparse Constrained Least Squares Problem

We consider the following sparse constrained least squares problem:

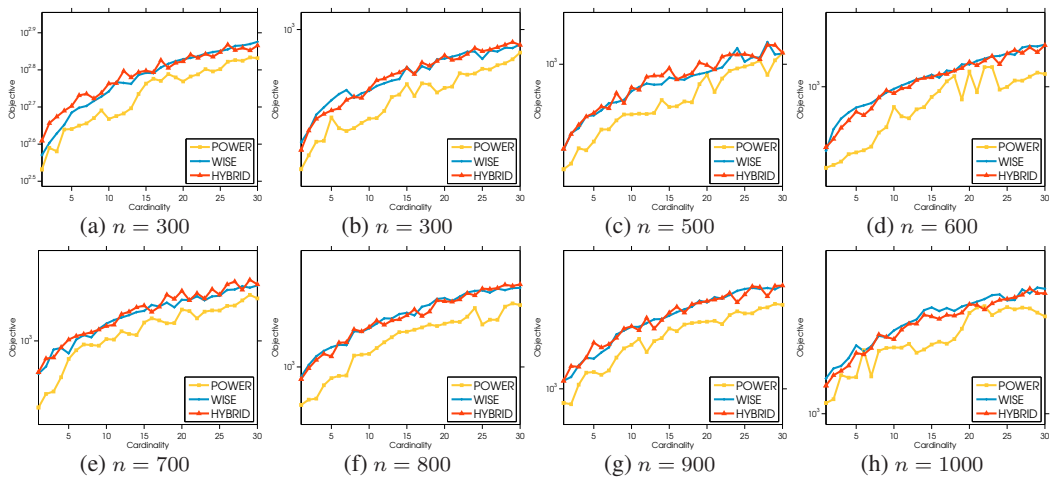
$$\min_{\mathbf{x}} \ \tfrac{1}{2}\|\mathbf{A}\mathbf{x} - \mathbf{b}\|_2^2, \ s.t. \ \|\mathbf{x}\|_0 \leq k \tag{11}$$

In our experiments, we first generate an design matrix $\mathbf{A}$ with independent Gaussian entries and then normalize each column to unit vector. We then select a support set of size 50 uniformly at random. We finally set $\mathbf{b} = \mathbf{A}\mathbf{x} + \boldsymbol{\epsilon}$ with $\epsilon \sim \mathcal{N}(0, \sigma^2 \mathbf{I})$. We vary n from $\{256, 512, 1024, 2048\}$ and set $m = n/8$. We vary the number of support set k from 1 to 50.

Compared Methods. We compare the proposed hybrid algorithm with four state-of-the-art sparse optimization algorithms: (i) Orthogonal Matching Pursuit (OMP) (ii) Regularized Orthogonal Matching Pursuit (ROMP) (iii) Subspace Pursuit (SP) and (iv) Proximal Gradient Algorithm (PGA). We remark that OMP, ROMP and SP are greedy pursuit algorithms and their support sets need to be selected greedily. They are non-gradient type algorithms and it is hard to incorporate these methods

[1] https://snap.stanford.edu/data/

[2] https://sites.google.com/site/xtyuan1980/publications



(a) $n = 300$ (b) $n = 300$ (c) $n = 500$ (d) $n = 600$

(e) $n = 700$ (f) $n = 800$ (g) $n = 900$ (h) $n = 1000$

Figure 4: Convergence behavior for solving the sparse PCA problem with different dimensions.

into other gradient-type based optimization algorithms. Such limitations have been pointed out in [2] and motivate the use of PGA.

Experimental Results. Several conclusions can be drawn from Figure 3. (i) Proximal gradient algorithm general leads to the worst performance. This is because the simple choice for constant step size does not exploit the specific structure of the original optimization problem. (ii) ROMP and SP are not stable and sometimes they present worse performance than the proximal point algorithm. (iii) OMP generally leads to better performance than PGA, SOMP and SP. (iv) The proposed method is at least comparable to the well-known sparse optimization method OMP.

4.4 Sparse Optimization Application: Sparse Principle Component Analysis

We consider the following sparse principle component analysis problem:

$$\max_{\mathbf{x}} \ \mathbf{x}^T(\mathbf{A}\mathbf{A}^T)\mathbf{x}, \ .s.t. \ \|\mathbf{x}\|_2^2 = 1, \ \|\mathbf{x}\|_0 \leq k \tag{12}$$

In our experiments, we generate a random matrix $\mathbf{A} \in \mathbb{R}^{n\times n}$ whose entries are sample standard Gaussian distribution. We report the objective value as in (12) with varying the parameter k from 1 to 30 for different n.

Compared Methods. We compare with two state-of-the-art methods, power method (POWER) [45] and coordinate-wise descent algorithm (WISE) [6].

Experimental Results. As can be seen in Figure 4, two conclusions can be drawn. (i) The power method presents the worst results. (ii) The proposed hybrid method is, if not better than, at least comparable to state-of-the-art sparse PCA solver WISE.

4.5 Computational Efficiency of Algorithm 1

Generally speaking, our HYBRID is effective and practical for large-scale discrete optimization. (i) Although it takes longer time to converge than the popular method such as proximal gradient algorithm, the computational time is acceptable and it generally takes less than 2 minutes to converge in all our instances. (ii) We think this computation time pays off as HYBRID achieves significantly higher accuracy than the proximal gradient algorithm. (iii) The parameter k in Algorithm 1 can be viewed as a tuning parameter to balance the efficacy and efficiency. (iv) Once can further accelerate the algorithm using asynchronous parallelism or mini-batch optimization techniques.

5 Conclusions and Future Work

This paper presents an effective and practical method for solving discrete optimization problems. Our method takes advantage of the effectiveness of combinatorial search and the efficiency of gradient descent. We also provided rigorous optimality analysis and convergence analysis for the proposed algorithm. The extensive experiments show that our method achieves state-of-the-art performance.

Our future work focuses on several directions. (i) We will extend the proposed method to deal with other nonconvex optimization applications (e.g., sparse coding [21], sparse phrase retrieval [3], nonnegative matrix factorization, deep learning, etc.). (ii) It is interesting to develop more efficient and robust branch-and-bound methods for solving medium-sized sub-problems globally. (iii) We are interested in incorporating other strategies (such as variance reduction [20, 39] and asynchronous parallelism [23]) into accelerating the proposed algorithm.

References

[1] Michal Aharon, Michael Elad, and Alfred Bruckstein. K-svd: An algorithm for designing overcomplete dictionaries for sparse representation. *IEEE Transactions on Signal Processing*, 54(11):4311–4322, 2006. 1

[2] Chenglong Bao, Hui Ji, Yuhui Quan, and Zuowei Shen. Dictionary learning for sparse coding: Algorithms and convergence analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 38(7):1356–1369, 2016. 1, 9

[3] Amir Beck and Yonina C Eldar. Sparsity constrained nonlinear optimization: Optimality conditions and algorithms. *SIAM Journal on Optimization (SIOPT)*, 23(3):1480–1509, 2013. 2, 5, 10

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

[5] Amir Beck and Luba Tetruashvili. On the convergence of block coordinate descent type methods. *SIAM journal on Optimization (SIOPT)*, 23(4):2037–2060, 2013. 2

[6] Amir Beck and Yakov Vaisbourd. The sparse principal component analysis problem: Optimality conditions and algorithms. *Journal of Optimization Theory and Applications*, 170(1):119–143, 2016. 9

[7] Daniel Bienstock. Computational study of a family of mixed-integer quadratic programming problems. *Mathematical programming*, 74(2):121–140, 1996. 2

[8] Yuri Boykov, Olga Veksler, and Ramin Zabih. Fast approximate energy minimization via graph cuts. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 23(11):1222–1239, 2001. 1

[9] Patrick Breheny and Jian Huang. Coordinate descent algorithms for nonconvex penalized regression, with applications to biological feature selection. *The Annals of Applied Statistics*, 5(1):232, 2011. 2

[10] Emmanuel J Candes and Terence Tao. Decoding by linear programming. *IEEE Transactions on Information Theory*, 51(12):4203–4215, 2005. 1

[11] Kai-Wei Chang, Cho-Jui Hsieh, and Chih-Jen Lin. Coordinate descent method for large-scale l2-loss linear support vector machines. *Journal of Machine Learning Research (JMLR)*, 9(Jul):1369–1398, 2008. 2

[12] Jinghui Chen and Quanquan Gu. Accelerated stochastic block coordinate gradient descent for sparsity constrained nonconvex optimization. In *Conference on Uncertainty in Artificial Intelligence (UAI)*, 2016. 2

[13] Michele Conforti, Gérard Cornuéjols, and Giacomo Zambelli. *Integer programming*, volume 271. Springer, 2014. 2

[14] Marianna De Santis, Stefano Lucidi, and Francesco Rinaldi. A fast active set block coordinate descent algorithm for \ell_1-regularized least squares. *SIAM Journal on Optimization (SIOPT)*, 26(1):781–809, 2016. 2

[15] David L. Donoho. Compressed sensing. *IEEE Transactions on Information Theory*, 52(4):1289–1306, 2006. 1

[16] Ehsan Elhamifar and Rene Vidal. Sparse subspace clustering: Algorithm, theory, and applications. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 35(11):2765–2781, 2013. 1

[17] Uriel Feige, David Peleg, and Guy Kortsarz. The dense k-subgraph problem. *Algorithmica*, 29(3):410–421, 2001. 7, 8

[18] Mingyi Hong, Xiangfeng Wang, Meisam Razaviyayn, and Zhi-Quan Luo. Iteration complexity analysis of block coordinate descent methods. *Mathematical Programming*, pages 1–30, 2013. 2

[19] Prateek Jain, Ambuj Tewari, and Purushottam Kar. On iterative hard thresholding methods for high-dimensional m-estimation. In *Advances in Neural Information Processing Systems*, pages 685–693, 2014. 2

[20] Rie Johnson and Tong Zhang. Accelerating stochastic gradient descent using predictive variance reduction. In *Advances in Neural Information Processing Systems (NIPS)*, pages 315–323, 2013. 10

[21] Honglak Lee, Alexis Battle, Rajat Raina, and Andrew Y Ng. Efficient sparse coding algorithms. *Advances in Neural Information Processing Systems (NIPS)*, 19:801, 2007. 1, 10

[22] Xingguo Li, Raman Arora, Han Liu, Jarvis Haupt, and Tuo Zhao. Nonconvex sparse learning via stochastic optimization with progressive variance reduction. *arXiv preprint arXiv:1605.02711*, 2016. 2

[23] Ji Liu, Stephen J Wright, Christopher Ré, Victor Bittorf, and Srikrishna Sridhar. An asynchronous parallel stochastic coordinate descent algorithm. *Journal of Machine Learning Research (JMLR)*, 16(285-322):1–5, 2015. 2, 10

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

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

[26] Ion Necoara. Random coordinate descent algorithms for multi-agent convex optimization over networks. *IEEE Transactions on Automatic Control*, 58(8):2001–2012, 2013. 8

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

[28] Yurii Nesterov. *Introductory lectures on convex optimization: A basic course*, volume 87. Springer Science & Business Media, 2013. 6

[29] Nam Nguyen, Deanna Needell, and Tina Woolf. Linear convergence of stochastic iterative greedy algorithms with sparse constraints. *arXiv preprint arXiv:1407.0088*, 2014. 2

[30] Andrei Patrascu and Ion Necoara. Efficient random coordinate descent algorithms for large-scale structured nonconvex optimization. *Journal of Global Optimization*, 61(1):19–46, 2015. 2

[31] Andrei Patrascu and Ion Necoara. Random coordinate descent methods for ℓ_0 regularized convex optimization. *IEEE Transactions on Automatic Control*, 60(7):1811–1824, 2015. 2

[32] Sekharipuram S Ravi, Daniel J Rosenkrantz, and Giri K Tayi. Heuristic and special case algorithms for dispersion problems. *Operations Research*, 42(2):299–310, 1994. 7

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

[34] Benjamin Recht, Christopher Re, Stephen Wright, and Feng Niu. Hogwild: A lock-free approach to parallelizing stochastic gradient descent. In *Advances in Neural Information Processing Systems (NIPS)*, pages 693–701, 2011. 2

[35] Paul Tseng and Sangwoon Yun. A coordinate gradient descent method for nonsmooth separable minimization. *Mathematical Programming*, 117(1):387–423, 2009. 3

[36] Jingdong Wang, Ting Zhang, Jingkuan Song, Nicu Sebe, and Heng Tao Shen. A survey on learning to hash. In *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI), to appear*, 2017. 1

[37] Jun Wang, Wei Liu, Sanjiv Kumar, and Shih-Fu Chang. Learning to hash for indexing big data - a survey. *Proceedings of the IEEE*, 104(1):34–57, 2016. 1

[38] Baoyuan Wu and Bernard Ghanem. ℓ_p-box admm: A versatile framework for integer programming. *arXiv preprint arXiv:1604.07666*, 2016. 8

[39] Lin Xiao and Tong Zhang. A proximal stochastic gradient method with progressive variance reduction. *SIAM Journal on Optimization (SIOPT)*, 24(4):2057–2075, 2014. 10

[40] Yangyang Xu and Wotao Yin. 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):1758–1789, 2013. 2

[41] Ganzhao Yuan and Bernard Ghanem. Binary optimization via mathematical programming with equilibrium constraints. In *arXiv preprint*, 2016. 2, 8

[42] Ganzhao Yuan and Bernard Ghanem. Sparsity constrained minimization via mathematical programming with equilibrium constraints. In *arXiv preprint*, 2016. 2

[43] Ganzhao Yuan and Bernard Ghanem. An exact penalty method for binary optimization based on MPEC formulation. In *AAAI Conference on Artificial Intelligence (AAAI)*, pages 2867–2875, 2017. 1, 8

[44] Ganzhao Yuan, Wei-Shi Zheng, and Bernard Ghanem. A matrix splitting method for composite function minimization. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 2, 6, 7

[45] Xiao-Tong Yuan and Tong Zhang. Truncated power method for sparse eigenvalue problems. *Journal of Machine Learning Research (JMLR)*, 14(Apr):899–925, 2013. 1, 7, 8, 9

[46] Aston Zhang and Quanquan Gu. Accelerated stochastic block coordinate descent with optimal sampling. In *ACM International Conference on Knowledge Discovery and Data Mining (SIGKDD)*, pages 2035–2044, 2016. 2

Appendix

A Proof of Lemma 1 (Sufficient Decrease Condition)

Lemma 1. *(**Sufficient Decrease Condition**) Let B be the working set selected at the t-th iteration. Suppose $\{F(x^t)\}_{t=1}^{\infty}$ is generated by Algorithm 1, the following inequality holds:*

$$F(\mathbf{x}^{t+1}) - F(\mathbf{x}^t) \leq -\tfrac{\theta}{2}\|\mathbf{x}_B^{t+1} - \mathbf{x}_B^t\|^2 \tag{13}$$

Proof. We define $N \triangleq \{1, 2, ..., n\} \setminus B$ and $\mathbf{s} \triangleq \mathbf{x}^{t+1} - \mathbf{x}^t$. Using the structure of $F(\cdot)$, we have the following results:

$$\begin{aligned} & F(\mathbf{x}^{t+1}) - F(\mathbf{x}^t) \\ = \; & \frac{1}{2}\mathbf{x}^{t+1^T}\mathbf{Q}\mathbf{x}^{t+1} + \mathbf{p}^T\mathbf{x}^{t+1} + \lambda\|\mathbf{x}^{t+1}\|_0 - \frac{1}{2}\mathbf{x}^{t^T}\mathbf{Q}\mathbf{x}^t - \mathbf{p}^T\mathbf{x}^t - \lambda\|\mathbf{x}^t\|_0 \\ = \; & \frac{1}{2}\mathbf{s}_B{}^T\mathbf{Q}_{BB}\mathbf{s}_B + \langle \mathbf{s}_B, (\mathbf{Q}\mathbf{x}^t)_B + \mathbf{p}_B\rangle + \lambda\|\mathbf{x}_B^{t+1}\|_0 - \lambda\|\mathbf{x}_B^t\|_0 \end{aligned} \tag{14}$$

Moreover, we have $\mathbf{s}_N = \mathbf{0}$. In the t-th iteration, we solve the optimization problem in (3), we have $\frac{1}{2}(\mathbf{x}_B^{t+1} - \mathbf{x}_B^t)^T\mathbf{Q}_{BB}^t(\mathbf{x}_B^{t+1} - \mathbf{x}_B^t) + \langle \mathbf{x}_B^{t+1} - \mathbf{x}_B^t, (\mathbf{Q}\mathbf{x}^t + \mathbf{p})_B\rangle + \frac{\theta}{2}\|\mathbf{x}_B^{t+1} - \mathbf{x}_B^t\|^2 \leq \lambda\|\mathbf{x}_B^t\|_0 - \lambda\|\mathbf{x}_B^{t+1}\|_0$. Combining this inequality with (14), we have the following result: $F(\mathbf{x}^{t+1}) - F(\mathbf{x}^t) \leq -\frac{\theta}{2}\|\mathbf{x}_B^{t+1} - \mathbf{x}_B^t\|_2^2$. Thus, we finish the proof of this lemma.

□

B Proof of Theorem 2 (Convergence for Binary or Sparse Optimization)

Theorem 1. ***Proof of Convergence for $h = h_{sparse}$ or $h = h_{binary}$.*** *Let $\mathbf{x}^t$ be the sequence generated by Algorithm 1. We have the following results. (i) It holds that $\lim_{t\to\infty}\mathbb{E}[\|\mathbf{x}^{t+1} - \mathbf{x}^t\|] = 0$. Moreover, there exists a scalar F^* such that $\lim_{t\to\infty}\mathbb{E}[F(\mathbf{x}^t)] = F^*$. (ii) When $h = h_{binary}$, we have $\mathbb{E}[\|\mathbf{x}^{t+1} - \mathbf{x}^t\| \mid \mathbf{x}^t] \geq \sqrt{2}/m$ if $\mathbf{x}^t \neq \mathbf{x}^{t+1}$ for all $t \geq 0$. The current solution changes at most $\sqrt{2}m(F(\mathbf{x}^0) - F(\mathbf{x}^*))/\theta$ times in expectation. (iii) When $h = h_{sparse}$, we have $i \in supp(\mathbf{x}^t)$, $|\mathbf{x}_i^t| \geq \delta$ for all $t \geq 0$, where $\delta \triangleq \min_j \{\min(\rho, \mathbf{x}_j^0, \sqrt{2\lambda/(\theta + \mathbf{Q}_{j,j})})\}$. Moreover, it holds that: $\mathbb{E}[\|\mathbf{x}^{t+1} - \mathbf{x}^t\| \mid \mathbf{x}^t] \geq \delta/m$ if $\mathbf{x}^t \neq \mathbf{x}^{t+1}$. The index set changes at most $\frac{2m(F(\mathbf{x}^0) - F(\mathbf{x}^*))}{\theta\delta}$ times in expectation.*

Proof. Firstly, taking the expectation of B for the sufficient descent inequality in Lemma 1, we have

$$\mathbb{E}[F(\mathbf{x}^{t+1}) \mid \mathbf{x}^t] \leq F(\mathbf{x}^t) - \mathbb{E}[\tfrac{\theta}{2}\|\mathbf{x}^{t+1} - \mathbf{x}^t\|\mathbf{x}^t] \tag{15}$$

Summing (15) over $i = 0, 1, 2, ..., t-1$, we have:

$$\begin{aligned} & \mathbb{E}[F(\mathbf{x}^t) - F(\mathbf{x}^0)] \leq \tfrac{\theta}{2}\textstyle\sum_{i=0}^t \|\mathbf{x}^{i+1} - \mathbf{x}^i\|_2^2 \\ \Rightarrow \quad & \mathbb{E}[F(\mathbf{x}^*) - F(\mathbf{x}^0)] \leq \tfrac{\theta}{2}\textstyle\sum_{i=0}^t \|\mathbf{x}^{i+1} - \mathbf{x}^i\|_2^2 \\ \Rightarrow \quad & \mathbb{E}[(2F(\mathbf{x}^0) - 2F(\mathbf{x}^*))/(t\theta)] \geq \min_{i=1,...,t}\|\mathbf{x}^{i+1} - \mathbf{x}^i\|_2^2 \end{aligned} \tag{16}$$

Therefore, we have $\lim_{t\to\infty}\|\mathbf{x}^{t+1} - \mathbf{x}^t\| = 0$. Moreover, there exist a scalar F^* such that $\lim_{t\to\infty} F(\mathbf{x}^t) = F^*$ almost sure.

(ii) We observe that when the current solution $\mathbf{x}^t$ changes, we have $\mathbb{E}[\|\mathbf{x}^{t+1} - \mathbf{x}^t\| \mid \mathbf{x}^t] \geq \sqrt{2}/m$ if $\mathbf{x}^t \neq \mathbf{x}^{t+1}$ for all $t \geq 0$. From (16), we obtain: $\mathbb{E}[(2F(\mathbf{x}^0) - 2F(\mathbf{x}^*))/(t\theta)] \geq \frac{\sqrt{2}}{m}$. Therefore, the number of iterations is upper bounded by $t \leq \sqrt{2}m(F(\mathbf{x}^0) - F(\mathbf{x}^*))/\theta$ times in expectation.

(iii) Note that in every iteration Algorithm 1 solves the the optimization problem as in (4). Using Theorem 1, we have that the optimal solution implies that $\mathbf{x}^*$ is also a block-1 stationary point.

Therefore, we have $|\mathbf{x}_{t+1}| \geq \sqrt{2\lambda/(\theta + \mathbf{Q}_{i,i})}$, $\forall i \in supp(\mathbf{x}_{t+1})$. Note that this is also true for all $t \geq 0$. Therefore, we have that, for any $i \in B$, we have: $|\mathbf{x}_i^t| \geq \sqrt{2\lambda/(\theta + \mathbf{Q}_{i,i})}$ or $|\mathbf{x}_i^t| = \mathbf{x}_i^0$. Therefore,we have the following results: $\mathbb{E}[\|\mathbf{x}^{k+1} - \mathbf{x}^k\|_2 \mid \mathbf{x}^k] = \frac{\delta}{m}$ if $\mathbf{x}^t \neq \mathbf{x}^{t+1}$. Moreover, from (16), we obtain: $\mathbb{E}[(2F(\mathbf{x}^0) - 2F(\mathbf{x}^*))/(t\theta)] \geq \frac{\delta}{m}$. Therefore, the number of iterations is upper bounded by $t \leq \frac{2m(F(\mathbf{x}^0) - F(\mathbf{x}^*))}{\theta\delta}$.

□

C Proof of Theorem 3 (Convergence Rate for Binary Optimization)

This section provides the proof of convergence rate for binary optimization. The following lemmas are useful in our proof.

Lemma 2. *We define* $\Pi(\mathbf{a}) = \arg\min_{\mathbf{x}} \|\mathbf{x} - \mathbf{a}\|,\ s.t.\ \mathbf{x} \in \{-1, +1\}^n$. *The following inequality always holds for all* $\mathbf{x}$ *and* $\mathbf{y}$*:*

$$\|\Pi(\mathbf{x}) - \mathbf{x}\|_2^2 \leq (1 - \kappa)\|\Pi(y) - \mathbf{x}\|_2^2 \tag{17}$$

with $\kappa = 0$. *Moreover, if* $\Pi(\mathbf{x}) \neq \Pi(\mathbf{y})$, *there exist a small* $\kappa > 0$ *such that (17) holds.*

Proof. Since $\|\Pi(\mathbf{x})\|_2^2 = n$, we have the following results:

$$\begin{aligned}
& \|\Pi(\mathbf{x}) - \mathbf{x}\|_2^2 \leq (1 - \kappa)\|\Pi(y) - \mathbf{x}\|_2^2 \\
\Leftrightarrow\ & \kappa\|\Pi(y) - \mathbf{x}\|_2^2 + \|\Pi(\mathbf{x})\|_2^2 + \|\mathbf{x}\|_2^2 - 2\langle\Pi(\mathbf{x}), \mathbf{x}\rangle \leq \|\Pi(y)\|_2^2 + \|\mathbf{x}\|_2^2 - 2\langle\Pi(y), \mathbf{x}\rangle \\
\Leftrightarrow\ & \kappa\|\Pi(y) - \mathbf{x}\|_2^2 \leq 2\langle\Pi(x) - \Pi(y), \mathbf{x}\rangle
\end{aligned}$$

Note that $\Pi(\mathbf{x}) \neq \Pi(\mathbf{y})$ also implies $\mathbf{x} \neq \Pi(\mathbf{y})$ and we have $\|\Pi(y) - \mathbf{x}\|_2^2 > 0$. Moreover, we notice that $\mathbf{x}_i \cdot sign(\mathbf{y}_i) \leq \mathbf{x}_i \cdot sign(\mathbf{x}_i)$ for all i and we have $\langle\Pi(x) - \Pi(y), \mathbf{x}\rangle > 0$. Therefore, there exists a sufficient small κ such that $\kappa\|\Pi(y) - \mathbf{x}\|_2^2 \leq 2\langle\Pi(x) - \Pi(y), \mathbf{x}\rangle$ holds. This finishes the proof of this lemma.

□

Lemma 3. *Assume that* $f(\cdot)$ *is* α*-strongly convex. The following inequality holds for any* $\mathbf{x}$ *and* $\mathbf{y}$*:*

$$\tfrac{\alpha^2}{4}\|\mathbf{y} - \mathbf{x}\|_2^2 - \|\nabla f(\mathbf{x})\|_2^2 \leq -\tfrac{\alpha}{2}(f(\mathbf{x}) - f(\mathbf{y})) \tag{18}$$

Proof. We naturally derive the following results:

$$\begin{aligned}
& \|\nabla f(\mathbf{x})\|_2^2 - \tfrac{\alpha^2}{4}\|\mathbf{y} - \mathbf{x}\|_2^2 \\
=\ & \big(\|\nabla f(\mathbf{x})\|_2 - \tfrac{\alpha}{2}\|\mathbf{x}^* - \mathbf{x}\|_2\big) \cdot \big(\|\nabla f(\mathbf{x})\|_2 + \tfrac{\alpha}{2}\|\mathbf{y} - \mathbf{x}\|_2\big) \\
=\ & \big(\|\nabla f(\mathbf{x})\|_2 \cdot \|\mathbf{y} - \mathbf{x}\|_2 - \tfrac{\alpha}{2}\|\mathbf{y} - \mathbf{x}\|_2^2\big) \cdot \big(\|\nabla f(\mathbf{x})\|_2 + \tfrac{\alpha}{2}\|\mathbf{y} - \mathbf{x}\|_2\big)/\|\mathbf{y} - \mathbf{x}\|_2 \\
\geq\ & \big(\langle\nabla f(\mathbf{x}), \mathbf{y} - \mathbf{x}\rangle - \tfrac{\alpha}{2}\|\mathbf{y} - \mathbf{x}\|_2^2\big) \cdot \big(\|\nabla f(\mathbf{x})\|_2 + \tfrac{\alpha}{2}\|\mathbf{y} - \mathbf{x}\|_2\big)/\|\mathbf{y} - \mathbf{x}\|_2 \\
\geq\ & \tfrac{\alpha}{2}(f(\mathbf{x}) - f(\mathbf{y}))
\end{aligned}$$

where the first inequality uses the Cauchy-Schwarz inequality, the second inequality uses the α-strongly convexity condition.

□

Theorem 2. ***Proof of Convergence Rate when*** $f(\cdot)$ ***is*** s***-Strongly Convex and*** $h \triangleq f_{\textbf{binary}}$. *Let* $\mathbf{x}^t$ *be the sequence generated by Algorithm 1. We have the following result:*

$$E[f(\mathbf{x}^{k+1})|\mathbf{x}^k] - f(\mathbf{x}^*) \leq (1 - C)(f(\mathbf{x}^k) - f(\mathbf{x}^*)) \tag{19}$$

where $C \triangleq (\frac{1}{2L}\sqrt{(L - s)(1 - \kappa)\kappa} + 1 - \kappa)/m$. *Moreover, it takes at most* $\log_{(1-C)}(\frac{\epsilon}{F(\mathbf{x}^0) - F(\mathbf{x}^*)})$ *times to find a local optimal solution satisfying* $F(\mathbf{x}^k) - F(\mathbf{x}^*) \leq \epsilon$. *Here* $L \triangleq \max_{i=1}^m \|\mathbf{U}_i^T\mathbf{Q}\mathbf{U}_i\|$.

Proof. We define

$$
\begin{aligned}
&d(\mathbf{x}) = \arg\min_{\mathbf{d}\in\mathbb{R}^n} H(\mathbf{x},\mathbf{d}),\\
&H(\mathbf{x},\mathbf{d}) \triangleq f(\mathbf{x}) + \langle \nabla f(\mathbf{x}),\mathbf{d}\rangle + \tfrac{1}{2}\mathbf{d}^T(\mathbf{A}+\theta\mathbf{I})\mathbf{d} + h_{\text{binary}}(\mathbf{x}+\mathbf{d})\\
&d'(\mathbf{x}) = \arg\min_{\mathbf{d}\in\mathbb{R}^n} H'(\mathbf{x},\mathbf{d}),\\
&H'(\mathbf{x},\mathbf{d}) \triangleq f(\mathbf{x}) + \langle \nabla f(\mathbf{x}),\mathbf{d}\rangle + \tfrac{1}{2}\mathbf{d}^T(L\mathbf{I}+\theta\mathbf{I})\mathbf{d} + h_{\text{binary}}(\mathbf{x}+\mathbf{d})\\
&d_i(\mathbf{x}) = \arg\min_{\mathbf{d}_i\in\mathbb{R}^{n_i}} H_i(\mathbf{x},\mathbf{d}_i),\\
&H_i(\mathbf{x},\mathbf{d}_i) \triangleq f(\mathbf{x}) + \langle \nabla_i f(\mathbf{x}),\mathbf{d}_i\rangle + \tfrac{1}{2}\mathbf{d}_i^T(\mathbf{U}_i^T\mathbf{Q}\mathbf{U}_i+\theta\mathbf{I})\mathbf{d}_i + h_{\text{binary}}(\mathbf{x}_i+\mathbf{d}_i)
\end{aligned}
\tag{20}
$$

where $\mathbf{A}$ is define as follows:

$$
\mathbf{A} = \begin{pmatrix} \mathbf{U}_1^T\mathbf{Q}\mathbf{U}_1 & 0 & 0 & 0\\ 0 & \mathbf{U}_2^T\mathbf{Q}\mathbf{U}_2 & 0 & 0\\ 0 & 0 & ... & 0\\ 0 & 0 & 0 & \mathbf{U}_m^T\mathbf{Q}\mathbf{U}_m \end{pmatrix}
$$

Clearly, the spectral norm of $\mathbf{A}$ is upper bounded by $\|\mathbf{A}\| \le \max_{i=1}^m \|\mathbf{U}_i^T\mathbf{Q}\mathbf{U}_i\| \triangleq L$. Since each i is picked randomly with probability $1/m$, we have the following inequality:

$$
\begin{aligned}
\mathbb{E}[F(\mathbf{x}^{t+1}) \mid \mathbf{x}^t] &= \mathbb{E}[F(\mathbf{x}^t + \mathbf{U}_i\mathbf{d}_i(\mathbf{x}^t))]\\
&\le \tfrac{1}{m}\textstyle\sum_{i=1}^m [f(\mathbf{x}^t) + \langle \nabla_i f(\mathbf{x}^t), d_i(\mathbf{x}^t)\rangle + \tfrac{1}{2} d_i(\mathbf{x}^t)^T\mathbf{U}_i^T\mathbf{A}\mathbf{U}_i d_i(\mathbf{x}^t) + h_{\text{binary}}(\mathbf{x}_i^{t+1})]\\
&= f(\mathbf{x}^t) + \tfrac{1}{m}\textstyle\sum_{i=1}^m [\langle \nabla_i f(\mathbf{x}^t), d_i(\mathbf{x}^t)\rangle + \tfrac{1}{2} d_i(\mathbf{x}^t)^T\mathbf{U}_i^T\mathbf{A}\mathbf{U}_i d_i(\mathbf{x}^t) + 0\\
&= f(\mathbf{x}^t) + \tfrac{1}{m}(H(\mathbf{x}^t, d(\mathbf{x}^t)) - f(\mathbf{x}^t)) - \tfrac{\theta}{2m}\|d(\mathbf{x}^t)\|_2^2\\
&= \tfrac{1}{m}H(\mathbf{x}^t, d(\mathbf{x}^t)) + \tfrac{m-1}{m}F(\mathbf{x}^t) - \tfrac{\theta}{2m}\|d(\mathbf{x}^t)\|_2^2
\end{aligned}
\tag{21}
$$

where the first inequality uses the convex property of $f(\cdot)$.

It is not hard to validate that the problem of $\arg\min_{\mathbf{d}\in\mathbb{R}^n} H'(\mathbf{x},\mathbf{d})$ admits closed-form solution with

$$
d'(\mathbf{x}) = \Pi\left(\mathbf{x} - \nabla f(\mathbf{x})/L\right) - \mathbf{x} \tag{22}
$$

Using the strongly convex property, we have:

$$
f(\mathbf{x}^t) - f(\mathbf{x}^*) \le \langle \nabla f(\mathbf{x}^t), \mathbf{x}^t - \mathbf{x}^*\rangle - \frac{s}{2}\|\mathbf{x}^t - \mathbf{x}^*\|_2^2 \tag{23}
$$

For any $\mathbf{x}^t \in \Psi$ and $\mathbf{x}^{t+1} \in \Psi$, we naturally derive the following inequalities:

$$
\begin{aligned}
& m\mathbb{E}[f(\mathbf{x}^{k+1})] - mf(\mathbf{x}^k)\\
\le\ & H(\mathbf{x}^k, d(\mathbf{x}^k)) - f(\mathbf{x}^k) - \tfrac{\theta}{2}\|d(\mathbf{x}^t)\|_2^2\\
=\ & \langle \nabla f(\mathbf{x}^k), d(\mathbf{x}^k)\rangle + \tfrac{1}{2} d(\mathbf{x}^k)^T A d(\mathbf{x}^k) + h(\mathbf{x}^k + d(\mathbf{x}^k))\\
\le\ & \langle \nabla f(\mathbf{x}^k), d'(\mathbf{x}^k)\rangle + \tfrac{1}{2} d'(\mathbf{x}^k)^T A d'(\mathbf{x}^k) + h(\mathbf{x}^k + d'(\mathbf{x}^k))\\
\le\ & \langle \nabla f(\mathbf{x}^k), d'(\mathbf{x}^k)\rangle + \tfrac{L}{2} d'(\mathbf{x}^k)^T d'(\mathbf{x}^k) + h(\mathbf{x}^k + d'(\mathbf{x}^k))\\
=\ & \langle \nabla f(\mathbf{x}^k), \Pi\left(\mathbf{x}^k - \nabla f(\mathbf{x}^k)/L\right) - \mathbf{x}^k\rangle + \tfrac{L}{2}\|\Pi\left(\mathbf{x}^k - \nabla f(\mathbf{x}^k)/L\right) - \mathbf{x}^k\|_2^2\\
=\ & \tfrac{L}{2}\|\Pi\left(\mathbf{x}^k - \nabla f(\mathbf{x}^k)/L\right) - \mathbf{x}^k + \nabla f(\mathbf{x}^k)/L\|_2^2 - \tfrac{L}{2}\|\nabla f(\mathbf{x}^k)/L\|_2^2\\
\le\ & \tfrac{L(1-\kappa)}{2}\|\Pi(\mathbf{x}^*) - \mathbf{x}^k + \nabla f(\mathbf{x}^k)/L\|_2^2 - \tfrac{L}{2}\|\nabla f(\mathbf{x}^k)/L\|_2^2\\
=\ & \tfrac{L(1-\kappa)}{2}\|\mathbf{x}^* - \mathbf{x}^k\|_2^2 + (1-\kappa)\langle \mathbf{x}^* - \mathbf{x}^k, \nabla f(\mathbf{x}^k)\rangle - \tfrac{\kappa}{2}\|\nabla f(\mathbf{x}^k)/L\|_2^2\\
\le\ & \tfrac{(L-s)(1-\kappa)}{2}\|\mathbf{x}^* - \mathbf{x}^k\|_2^2 - \tfrac{\kappa}{2L^2}\|\nabla f(\mathbf{x}^k)\|_2^2 + (1-\kappa)(f(\mathbf{x}^*) - f(\mathbf{x}^t))\\
=\ & (\tfrac{\kappa}{2L^2})\left(\tfrac{(L-s)(1-\kappa)L^2}{\kappa}\|\mathbf{x}^* - \mathbf{x}^k\|_2^2 - \|\nabla f(\mathbf{x}^k)\|_2^2\right) + (1-\kappa)(f(\mathbf{x}^*) - f(\mathbf{x}^t))\\
\le\ & (\tfrac{\kappa}{2L^2})\sqrt{\tfrac{(L-s)(1-\kappa)L^2}{\kappa}}(f(\mathbf{x}^*) - f(\mathbf{x}^t)) + (1-\kappa)(f(\mathbf{x}^*) - f(\mathbf{x}^t))
\end{aligned}
$$

where the first step uses (21); the second step uses the definition of $H(\cdot)$; the third step uses the fact that $d(\mathbf{x}^k)$ is the minimizer of $H(\cdot)$ in (20); the fourth step uses the inequality that $\|\mathbf{A}\| \leq L$; the fifth step uses (22); the seventh step uses the inequality in (17) in Lemma 2; the eighth step uses the fact that $\Pi(\mathbf{x}^*) = \mathbf{x}^*$; the ninth step uses (23) since $f(\cdot)$ is a strongly convex function; the eleventh step uses the inequality in (18). We have the following inequality:

$$E[f(\mathbf{x}^{k+1}) \mid \mathbf{x}^k] - f(\mathbf{x}^k) \leq C(f(\mathbf{x}^*) - f(\mathbf{x}^t)) \tag{24}$$

Rearranging terms, we obtain the inequality in (19). In other words, the sequence $\{f(\mathbf{x}^t)\}$ converges to the stationary point linearly in the quotient sense. Applying the inequality in (19) recursively, we obtain:

$$E[f(\mathbf{x}^{k+1})] - f(\mathbf{x}^*) \leq (1-C)^k (f(\mathbf{x}^0) - f(\mathbf{x}^*)) \tag{25}$$

Therefore, we conclude that it takes at most $\log_{(1-C)}(\frac{\epsilon}{F(\mathbf{x}^0)-F(\mathbf{x}^*)})$ times to find a local optimal solution satisfying $F(\mathbf{x}^k) - F(\mathbf{x}^*) \leq \epsilon$.

□

D Proof of Theorem 4 (Convergence Rate for Sparse Optimization)

This section provide some convergence analysis of Algorithm 1. For notational convenience, we define $p(\mathbf{x}) = I_\Delta(\mathbf{x}),\ \Delta = \{\mathbf{x} \mid \|\mathbf{x}\|_\infty \leq \rho\}$ and

$$J(\mathbf{x}) \triangleq f(\mathbf{x}) + p(\mathbf{x}) \tag{26}$$

$$\bar{H}(\mathbf{x}, \mathbf{d}) \triangleq f(\mathbf{x}) + \langle \nabla f(\mathbf{x}), \mathbf{d} \rangle + \tfrac{1}{2}\mathbf{d}^T(\mathbf{A} + \theta\mathbf{I})\mathbf{d} + p(\mathbf{x} + \mathbf{d}) \tag{27}$$

$$\bar{d}_i(\mathbf{x}) \triangleq \arg\min_{\mathbf{d}_i \in \mathbb{R}^{n_i}} \langle \nabla_i f(\mathbf{x}), \mathbf{d}_i \rangle + \tfrac{1}{2}\mathbf{d}_i^T(\mathbf{U}_i^T\mathbf{Q}\mathbf{U}_i + \theta\mathbf{I})\mathbf{d}_i + p_i(\mathbf{x}_i + \mathbf{d}_i) \tag{28}$$

where $\mathbf{A}$ is defined in (21).

Lemma 4. *For any $\mathbf{x},\ \mathbf{y} \in \mathbb{R}^n$, if we pick $i \in \{1, 2, ..., m\}$ uniformly at random, then*

$$m\mathbb{E}[J(\mathbf{x} + \mathbf{U}_i\bar{d}_i(\mathbf{x}))] + \tfrac{1}{2}\sum_{i=1}^m g_i(\mathbf{x})^T(\mathbf{U}_i^T\mathbf{A}\mathbf{U}_i + \theta\mathbf{I})^{-1}g_i(\mathbf{x}) + \langle g(\mathbf{x}), \mathbf{y} - \mathbf{x} \rangle \leq J(\mathbf{y}) + (m-1)J(\mathbf{x}) \tag{29}$$

where $g_i(\mathbf{x}) \triangleq -(\mathbf{U}_i^T\mathbf{A}\mathbf{U}_i + \theta\mathbf{I})d_i'(\mathbf{x})$.

Proof. (i) Firstly, since $\bar{d}(\mathbf{x}) = \arg\min_{\mathbf{d}\in\mathbb{R}^n} \bar{H}(\mathbf{x}, \mathbf{d})$, by the first-order optimality condition of the optimization in $\bar{d}(\mathbf{x})$, we have:

$$-\nabla f(\mathbf{x}) + g(\mathbf{x}) \in \partial h(\mathbf{x} + \bar{d}(\mathbf{x})) \tag{30}$$

Note that $\bar{d}(\mathbf{x}) \triangleq \sum_{i=1}^m \mathbf{U}_i\bar{d}_i(\mathbf{x})$ and $g(\mathbf{x}) \triangleq \sum_{i=1}^m \mathbf{U}_i g_i(\mathbf{x})$ by notations, we have:

$$\begin{aligned} & \bar{d}(\mathbf{x})^T(\mathbf{A} + \theta\mathbf{I})\bar{d}(\mathbf{x}) = \textstyle\sum_{i=1}^m \bar{d}_i(\mathbf{x})^T(\mathbf{U}_i^T\mathbf{A}\mathbf{U}_i + \theta\mathbf{I})\bar{d}_i(\mathbf{x}) \\ = \quad & -\textstyle\sum_{i=1}^m \bar{d}_i(\mathbf{x})^T g_i(\mathbf{x}) = \sum_{i=1}^m g_i(\mathbf{x})^T(\theta\mathbf{I} + \mathbf{U}_i^T\mathbf{A}\mathbf{U}_i)^{-1}g_i(\mathbf{x}) \end{aligned} \tag{31}$$

(ii) Secondly, we derive the following inequalities:

$$\begin{aligned} & \bar{H}(\mathbf{x}, d(\mathbf{x})) \\ = \quad & f(\mathbf{x}) + \langle \nabla f(\mathbf{x}), \bar{d}(\mathbf{x}) \rangle + \tfrac{1}{2}\bar{d}(\mathbf{x})^T(\mathbf{A} + \theta\mathbf{I})\bar{d}(\mathbf{x}) + p(\mathbf{x} + \bar{d}(\mathbf{x})) \\ \leq \quad & f(\mathbf{y}) + \langle \nabla f(\mathbf{x}), \mathbf{x} - \mathbf{y} \rangle + \langle \nabla f(\mathbf{x}), \bar{d}(\mathbf{x}) \rangle + \tfrac{1}{2}\bar{d}(\mathbf{x})^T(\mathbf{A} + \theta\mathbf{I})\bar{d}(\mathbf{x}) + p(\mathbf{x} + d(\mathbf{x})) \\ \leq \quad & f(\mathbf{y}) + \langle \nabla f(\mathbf{x}), \mathbf{x} - \mathbf{y} \rangle + \langle \nabla f(\mathbf{x}), \bar{d}(\mathbf{x}) \rangle + \tfrac{1}{2}\bar{d}(\mathbf{x})^T(\mathbf{A} + \theta\mathbf{I})\bar{d}(\mathbf{x}) + p(y) + \langle \partial p(\mathbf{x} + \bar{d}(\mathbf{x})), \mathbf{x} + \bar{d}(\mathbf{x}) - \mathbf{y} \rangle \\ \leq \quad & f(\mathbf{y}) + \langle \nabla f(\mathbf{x}), \mathbf{x} - \mathbf{y} \rangle + \langle \nabla f(\mathbf{x}), \bar{d}(\mathbf{x}) \rangle + \tfrac{1}{2}\bar{d}(\mathbf{x})^T(\mathbf{A} + \theta\mathbf{I})\bar{d}(\mathbf{x}) + p(y) - \langle \nabla f(\mathbf{x}) - g(\mathbf{x}), \mathbf{x} + \bar{d}(\mathbf{x}) - \mathbf{y} \rangle \\ = \quad & J(\mathbf{y}) + \tfrac{1}{2}\bar{d}(\mathbf{x})^T(\mathbf{A} + \theta\mathbf{I})\bar{d}(\mathbf{x}) + \langle g(\mathbf{x}), \mathbf{x} + \bar{d}(\mathbf{x}) - \mathbf{y} \rangle \\ = \quad & J(\mathbf{y}) + \langle g(\mathbf{x}), \mathbf{x} - \mathbf{y} \rangle - \tfrac{1}{2}\textstyle\sum_{i=1}^n g_i(\mathbf{x})^T(\mathbf{U}_i^T\mathbf{A}\mathbf{U}_i + \theta\mathbf{I})^{-1}g_i(\mathbf{x}) \end{aligned} \tag{32}$$

where the first step uses the definition of $\bar{H}(\mathbf{x}, d(\mathbf{x}))$; the second step uses the convexity of $f(\cdot)$; the third step uses the convexity of $p(\mathbf{x})$; the fourth step uses the optimality condition in (30); the last step uses (31).

(iii) Thirdly, we have the following inequalities:

$$\begin{aligned}\mathbb{E}[F(\mathbf{x}^{t+1}) \mid \mathbf{x}^t] &= \tfrac{1}{m}\textstyle\sum_{i=1}^m F(\mathbf{x}^t + \mathbf{U}_i \bar{d}_i(\mathbf{x}^t)) \\ &\leq \tfrac{1}{m}\textstyle\sum_{i=1}^m [f(\mathbf{x}^t) + \langle \nabla_i f(\mathbf{x}^t), \bar{d}_i(\mathbf{x}^t)\rangle + \tfrac{1}{2}\bar{d}_i(\mathbf{x}^t)^T \mathbf{U}_i^T \mathbf{A}\mathbf{U}_i \bar{d}_i(\mathbf{x}^t) + p_i(\mathbf{x}^t)] \\ &= \tfrac{1}{m}\bar{H}(\mathbf{x}^t, \bar{d}(\mathbf{x}^t)) + \tfrac{m-1}{m} f(\mathbf{x}^t) + \tfrac{1}{m}\textstyle\sum_{i=1}^m p_i(\mathbf{x}^t) - \tfrac{\theta}{2m}\|\bar{d}(\mathbf{x}^t)\|_2^2 \\ &= \tfrac{1}{m}\bar{H}(\mathbf{x}^t, \bar{d}(\mathbf{x}^t)) + \tfrac{m-1}{m} F(\mathbf{x}^t) \end{aligned} \tag{33}$$

where the first step uses the fact that $f(\mathbf{x} + \mathbf{U}_i \mathbf{d}_i) \leq f(\mathbf{x}) + \langle \nabla_i f(\mathbf{x}), \mathbf{d}_i\rangle + \frac{1}{2}\mathbf{d}_i^T \mathbf{U}_i^T \mathbf{Q}\mathbf{U}_i \mathbf{d}_i$ holds since the objective function is convex. Combining (32) and (33), we quickly finish the proof of this lemma.

□

Theorem 3. ***Proof of Convergence Rate when*** $f(\cdot)$ ***is Convex and*** $h = f_{\textit{sparse}}$**.** *Algorithm 1 at most takes* $(m(0.5\|\mathbf{x}^0 - \mathbf{x}^*\|_2^2(L+\theta)^2 + F(\mathbf{x}^0) - F(\mathbf{x}^*))/\epsilon - m - 1) \cdot (2mF(\mathbf{x}^0) - 2mF(\mathbf{x}^*))/(\theta\delta))$ *iterations in expectation to find a local optimal solution satisfying* $F(\mathbf{x}^k) - F(\mathbf{x}^*) \leq \epsilon$*. Here* $L \triangleq \max_{i=1}^m \|\mathbf{U}_i^T \mathbf{Q}\mathbf{U}_i\|$.

Proof. When the support set does not change, the optimization problem reduced to the convex optimization as in (26). Let $\mathbf{x}^*$ be the optimal value of the problem in (26) and denote $j = 0, 1, ...$ as the iteration counter after the support set does not change. We now prove the following result:

$$\mathbb{E}[J(\mathbf{x}^j)] - J(\mathbf{x}^*) \leq \frac{m}{m+j+1}\left(\tfrac{1}{2}R_0^2 + J(\mathbf{x}^0) - J(\mathbf{x}^*)\right) \tag{34}$$

where R_0 is defined as $R_0 = \|\mathbf{x}^0 - \mathbf{x}^*\|_{(\mathbf{A}+\theta\mathbf{I})} \leq \|\mathbf{x}^0 - \mathbf{x}^*\|(L+\theta)$.

Let $\mathbf{x}^*$ be an arbitrary optimal solution, For $j \geq 0$, we denote:

$$r_j^2 = \|\mathbf{x}^j - \mathbf{x}^*\|_{(\mathbf{A}+\theta\mathbf{I})}^2 = \textstyle\sum_{i=1}^m \langle (\mathbf{U}_i^T \mathbf{A}\mathbf{U}_i + \theta\mathbf{I})(\mathbf{x}_i^j - \mathbf{x}_i^*), \mathbf{x}_i^j - \mathbf{x}_i^*\rangle \tag{35}$$

Since $\mathbf{x}^{j+1} = \mathbf{x}^j + \mathbf{U}_i d_i(\mathbf{x}_j)$, we obtain:

$$r_{j+1}^2 = r_j^2 + 2\textstyle\sum_{i=1}^m \langle (\mathbf{U}_i^T \mathbf{A}\mathbf{U}_i + \theta\mathbf{I})\bar{d}_i(\mathbf{x}^t), \mathbf{x}_i^t - \mathbf{x}_i^*\rangle + \bar{d}_i(\mathbf{x}^t)^T(\mathbf{A} + \theta\mathbf{I})\bar{d}_i(\mathbf{x}^t) \tag{36}$$

Multiplying both sides by $1/2$ and taking expectation with respect to B, we have:

$$\mathbb{E}[\tfrac{1}{2}r_{k+1}^2] = \tfrac{1}{2}r_k^2 + \tfrac{1}{m}\left(\textstyle\sum_{i=1}^m \langle g_i(\mathbf{x}^k), \mathbf{x}_i^k - \mathbf{x}_i^*\rangle + \tfrac{1}{2}\sum_{i=1}^m g_i(\mathbf{x})^T(\mathbf{U}_i^T \mathbf{A}\mathbf{U}_i + \theta\mathbf{I})^{-1} g_i(\mathbf{x})\right) \tag{37}$$

By Lemma 4, we obtain:

$$\mathbb{E}[\tfrac{1}{2}r_{k+1}^2] \leq \tfrac{1}{2}r_k^2 + \tfrac{1}{m}J(\mathbf{x}^*) + \tfrac{m-1}{m}J(\mathbf{x}^k) - \mathbb{E}[J(\mathbf{x}^{k+1})] \tag{38}$$

By rearranging terms, we have:

$$\forall k \geq 0,\ \mathbb{E}[\tfrac{1}{2}r_{k+1}^2 + J(\mathbf{x}^{k+1}) - J(\mathbf{x}^*)] \leq (\tfrac{1}{2}r_k^2 + J(\mathbf{x}^k) - J(\mathbf{x}^*)) - \tfrac{1}{m}(J(\mathbf{x}^k) - J(\mathbf{x}^*)) \tag{39}$$

Taking expectation on both sides of the above inequality, we have:

$$E[\tfrac{1}{2}r_{t+1}^2 + J(\mathbf{x}^{k+1}) - J(\mathbf{x}^*)] \leq \mathbb{E}[\tfrac{1}{2}r_k^2 + J(\mathbf{x}^t) - F^*] - \tfrac{1}{n}\mathbb{E}[J(\mathbf{x}^k) - J(\mathbf{x}^*)] \tag{40}$$

Notice that $\mathbb{E}[J(\mathbf{x}^j)]$ is monotonically decreasing for $j = 0, 1, 2...t$. Applying the inequality (40) recursively, we obtain:

$$\begin{aligned}\mathbb{E}[J(\mathbf{x}^{t+1})] - J(\mathbf{x}^*) &\leq \mathbb{E}[\tfrac{1}{2}r_{t+1}^2 + \mathcal{J}(\mathbf{x}^{t+1}) - \mathcal{J}(\mathbf{x}^*)] \\ &\leq \tfrac{1}{2}r_0^2 + J(\mathbf{x}^0) - J(\mathbf{x}^*) - \tfrac{1}{m}\sum_{j=0}^{t}(\mathbb{E}[J(\mathbf{x}^j)] - J(\mathbf{x}^*)) \\ &\leq \tfrac{1}{2}r_0^2 + J(\mathbf{x}^0) - J(\mathbf{x}^*) - \tfrac{t+1}{m}(\mathbb{E}[J(\mathbf{x}^{t+1})] - J(\mathbf{x}^*)) \end{aligned}$$

Therefore, we obtain the conclusion in (34).

(iii) From the inequality in (34), we conclude that it takes $T_{\text{in}} = \frac{m(0.5\|\mathbf{x}^0-\mathbf{x}^*\|_2^2(L+\theta)^2+F(\mathbf{x}^0)-F(\mathbf{x}^*))}{\epsilon} - m-1$ iterations in expectation to converge to a local optimal solution that satisfies $F(\mathbf{x}^k)-F(\mathbf{x}^*) \leq \epsilon$. Moreover, from Theorem 2 we have that Algorithm 1 changes at most T_{out} times with $T_{\text{out}} = \frac{2K(F(\mathbf{x}^0)-F(\mathbf{x}^*))}{\theta\delta}$. Therefore, we conclude that it takes $T_{\text{in}} \times T_{\text{out}}$ iteration to converge to the local optimal solution. This finishes the proof of this Theorem.

□