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

Ganzhao Yuan\(^1\), Li Shen\(^2\), Wei-Shi Zheng\(^1\)

\(^1\) School of Data and Computer Science, Sun Yat-sen University (SYSU), P.R. China
\(^2\) Tencent AI Lab, Shenzhen, P.R. China

yuanganzhao@gmail.com, mathshenli@gmail.com, zhwshi@mail.sysu.edu.cn

Abstract

Discrete optimization is a central problem in mathematical optimization 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 coordinate descent methods such as 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 coordinate descent. Specifically, we consider random strategy or/and greedy strategy to select a subset of coordinates as the working set, and then perform global combinatorial search over the working set based on the original objective function. 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:

\[
\min_{x \in \mathbb{R}^n} F(x) \triangleq f(x) + h(x)
\]

(1)

where \(f(x)\) is a smooth convex function with its gradient being \(L\)-Lipschitz continuous, and \(h(x)\) is a piecewise separable function with \(h(x) = \sum^n_{i=1} h_i(x_i)\). We consider two cases for \(h(\cdot)\):

\[
\{h_b(x) \triangleq I_{\Psi}(x), \ \Psi \triangleq \{-1, 1\}^n\} \quad \text{or} \quad \{h_s(x) \triangleq \lambda \|x\|_0 + I_{\Omega}(x), \ \Omega \triangleq [-\rho, \rho]^n\}
\]

where \(I_{\Psi}(\cdot)\) is an indicator function on \(\Psi\) with \(I_{\Psi}(x) = \begin{cases} 0, & x \in \Psi \ \\
\infty, & x \notin \Psi \end{cases}\). \(\|\cdot\|_0\) is a function that counts the number of nonzero elements in a vector, \(\lambda\) and \(\rho\) are strictly positive constants. When \(h \triangleq h_b\), (1) refers to the binary optimization problem; when \(h \triangleq h_s\), (1) corresponds to the sparse optimization problem.

Binary optimization and sparse optimization capture a variety of applications of interest in both machine learning and computer vision, including binary hashing \([46, 45]\), dense subgraph discovery \([54, 52]\), Markov random fields \([8]\), compressive sensing \([11, 17]\), sparse coding \([25, 1, 2]\), subspace clustering \([18]\), to name only a few. In addition, 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 \(\{x \in \{-1, 1\}^n : \|x - 1\|_0 + \|x + 1\|_0 \leq n\} \iff \{x \in \mathbb{R}^n : \|x\|_\infty \leq \rho, \|x\|_0 = n\}\) for \(\rho > 0\) and \(n \leq 2\rho\). The reverse is also true using the variational reformulation of \(\ell_0\) pseudo-norm \([8]\): \(\forall \|x\|_\infty \leq \rho, \|x\|_0 = 1\) for \(\rho > 0\).

Preprint. Work in progress.
min $\langle 1, v \rangle$, s.t. $v \in \{0, 1\}^n$, $|x| \leq \rho v$. There are generally four classes of methods for solving the binary or sparse optimization problem in the literature, which we present below.

**Relaxed Approximation Method.** One popular method to solve (1) is convex or nonconvex relaxed approximation method. Box constrained relaxation, semi-definite programming relaxation and spherical relaxation are often used for solving binary optimization problems, while $\ell_1$ norm, top-$k$ norm, Schatten $\ell_p$ norm and others (such as re-weighted $\ell_1$ norm, capped $\ell_1$, half quadratic et al.) are often used for solving sparse optimization problems. It is generally accepted that nonconvex method often achieves better accuracy than the convex counterpart. Despite its merits, this class of method fails to directly control the sparse or binary property of the solution.

**Greedy Pursuit Method.** This method is often used to solve cardinality constrained discrete optimization problems. For sparse optimization, this method greedily selects at each step one atom of the variables which have some desirable benefits [43, 15, 33, 7, 32]. It has a monotonically decreasing property and optimality guarantees in some situations, but it is limited to solving problems with smooth objective functions (typically the square function). For binary optimization, this method is strongly related to submodular optimization [10] as minimizing a set function can be reformulated as a binary optimization problem.

**Combinatorial Search Method.** Combinatorial search method [14] is typically concerned with problems that are NP-hard. 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 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.

**Proximal Point Method.** Based on the current gradient $\nabla f(x^k)$, proximal point method [3, 28, 23, 36, 38, 26] iteratively performs a gradient update followed by a proximal operation: $x^{k+1} = \text{prox}_{h_k}(x^k - \gamma \nabla f(x^k))$. Here the proximal operator $\text{prox}_{h_k}(a) = \arg \min_{x} \frac{1}{2} \|x - a\|^2 + h(x)$ can be evaluated analytically, and $\gamma \geq 1/L$ is the step size with $L$ being the Lipschitz constant. This method is closely related to (block) coordinate descent [34, 12, 3, 16, 41, 5, 20, 29, 49] in the literature. Due to its simplicity, many strategies (such as variance reduction [24, 48, 13], asynchronous parallelism [27, 42], non-uniform sampling [55]) have been proposed to accelerate proximal point 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, the scaled identity quadratic majorization function may not necessarily be a good approximation for the original problem.

Compared to existing solutions mentioned above, our method has the following merits. (i) It can directly control the sparse or binary property of the solution. (ii) It is a greedy coordinate descent algorithm. (iii) It leverages the effectiveness of combinatorial search. (iv) It significantly outperforms proximal point method and inherits its computational advantages.

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 coordinate descent (See Section 2). (ii) Theoretically, we establish the optimality hierarchy and the convergence rate of our proposed algorithm. Our algorithm finds a stronger stationary point than existing methods (See Section 3). (iii) Empirically, we have conducted extensive experiments on some some binary optimization and sparse optimization tasks to show the superiority of our method (See Section 4).

## 2 Proposed Algorithm

This section presents our hybrid method for solving the optimization problem in (1). Our algorithm is an iterative procedure. In every iteration, the index set of variables is separated to two sets $B$ and $N$, where $B$ is the working set. Each time variables corresponding to $N$ are fixed while a sub-
We use the following variable selection strategy for sparse optimization, and similarly consider a proximal point strategy for the subproblem. This is to guarantee sufficient descent conditions of the objective function and global convergence of Algorithm 1 (refer to Theorem 1).

**Finding a Working Set.** We observe that it contains \( C^k \) possible combinations of choice for the working set. One may use a cyclic strategy to alternatingly select all the choices of the working set. However, past results show that coordinate gradient method results in faster convergence when the working set in selected in an arbitrary order \([21]\) or in a greedy manner \([44, 22]\). This inspires us to consider a proximal point strategy for the subproblem. This is to guarantee sufficient descent condition of the objective function and global convergence of Algorithm 1 (refer to Theorem 1).

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

**Greedy strategy.** We use the following variable selection strategy for sparse optimization, and similar strategy can be directly applied to the binary case. Generally speaking, we pick 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 \( x^t \). We denote \( I \triangleq \{i : x^t_i = 0\} \) and \( J \triangleq \{j : x^t_j \neq 0\} \). We expect the working set is balanced and pick \( k/2 \) coordinates from \( I \) and \( k/2 \) coordinates from \( J \). For \( I \), we solve a one-variable subproblem to compute the possible decrease for all \( i \in I \) of \( x^t \) when changing from zero to nonzero:

\[
\forall i = 1, ..., |I|, \quad c_i = \min_{\alpha} F(x^t + \alpha e_i) - F(x^t)
\]

For \( J \), we compute the decrease for each coordinate \( j \in J \) of \( x^t \) when changing from nonzero to exactly zero:

\[
\forall j = 1, ..., |J|, \quad d_j = F(x^t + \alpha e_j) - F(x^t), \quad \alpha = x^t_j
\]
Here $e_i$ is a unit vector with a 1 in the $i$th entry and 0 in all other entries. We sort the vectors $c$ and $d$ in increasing order and then pick top-(k/2) coordinates form $I$ and top-(k/2) coordinates from $J$ as the working set. If either $|I| < k/2$ or $|J| < k/2$, one can pick the whole set of $I$ or $J$ as a part of the working set. Here we assume that $k$ is an even number. We remark that when $f(x) \triangleq \frac{1}{2}x^T Q x + \langle x, p \rangle$ and $h(x) \triangleq h_s(x)$, we can further simplify $c_i$ and $d_j$ as $c_i = \alpha(Qx + p)_i + 0.5\alpha^2 Q_{i,i} + \lambda$ and $d_j = \alpha(Qx + p)_j + 0.5\alpha^2 Q_{j,j} - \lambda$, respectively.

- **Extensions to Cardinality Constrained Problems.** In many applications, it is desirable to directly control the cardinality of the solution using the following constraints:

$$\{h_{bc}(x) \triangleq I_{\mathcal{T}}(x), \; \mathcal{T} \triangleq \{ x \mid x \in \{0, 1\}^n, \; x^T 1 = s \}\} \cup \{h_{sc}(x) \triangleq I_{\Phi}(x)\}, \; \Phi \triangleq \{ x \mid \|x\|_0 \leq s\}.$$ 

The proposed block coordinate method (including the exhaustive search algorithm and the working set selection strategies) can still be applied even when $h(x)$ contains one non-separable constraint. What one needs is to ensure that the solution $x^t$ is a feasible solution for all $t = 0, 1, \ldots, \infty$. This is similar to the prior work of [3].

3 Theoretical Analysis

This section provides some optimality analysis and convergence analysis of our method. All proofs of this paper are placed into the supplementary material.

3.1 Optimality Analysis

In the sequel, we present some necessary optimal conditions for [1]. Since the block-$k$ optimality is novel in this paper, it is necessary to clarify formally its relations with existing optimality conditions. We use $\bar{x}$, $\hat{x}$, and $\bar{x}$ to denote a basic stationary point, an $L$-stationary point, and a block-$k$ stationary point, respectively.

**Definition 1. (Basic Stationary Point)** A solution $\bar{x}$ is called a basic stationary point if the following holds: $h \triangleq h_b : \bar{x} \in \{-1, +1\}^n$; $h \triangleq h_s : \bar{x}_S = \arg \min_{x \in [-\rho, 1, \rho]} \frac{1}{2}\|z - (\bar{x} - \nabla f(\bar{x})/L)\|_2^2$, where $S \triangleq \{ i \mid \bar{x}_i \neq 0 \}$.

**Remarks:** For binary optimization, any binary solution is a basic stationary point. For sparse optimization, basic stationary point states that the solution achieves its global optimality 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)** A solution $\bar{x}$ is an $L$-stationary point if it holds that: $\hat{x} = \arg \min_{x} g(x, \bar{x}) + h(x)$ with $g(x, z) \triangleq f(z) + \langle \nabla f(z), x - z \rangle + \frac{L}{2}\|x - z\|_2^2$, $\forall x, \; z, \; f(x) \leq g(x, z)$.

**Remarks:** This is the well-known proximal thresholding operator [3]. Although it has a closed-form solution, this scaled identity quadratic function may not be a good majorization/surrogate function for the non-convex problem.

**Definition 3. (Block-$k$ Stationary Point)** A solution $\bar{x}$ is a block-$k$ stationary point if it holds that: $\bar{x} \in \arg \min_{x \in \mathbb{R}^n} P(z; x, B) \triangleq \{ F(z), \; s.t. \; z_N = \bar{x}_N \}, \; \forall |B| = k, \; N \triangleq \{1, \ldots, n\} \setminus B$.

**Remarks:** Block-$k$ stationary point is novel in this paper. One remarkable feature of this concept is that it involves solving a small-sized NP-hard problem which can be tackled by some practical global optimization method. The following proposition states the relations between the three types of stationary point.

**Proposition 1. Proof of the Hierarchy between the Necessary Optimality Conditions.** We have the following optimality hierarchy: \begin{align*}
\text{Block-2 Stat. Point} & \subset \ldots \subset \text{Block-n Stat. Point} \overset{(1)}{\subset} \text{Block-1 Stat. Point} \overset{(2)}{\subset} \text{Optimal Point} \overset{(3)}{\subset}
\end{align*}

**Remarks:** It is worthwhile to point out that the seminal work of [3] also presents an optimality condition for sparse optimization. However, our block-$k$ condition is stronger than their coordinate-wise optimality since their optimal condition corresponds to $k = 1$ in our optimality condition framework.
A Running Example. We consider the quadratic optimization problem \( \min_{x \in \mathbb{R}^n} \frac{1}{2} x^T Q x + x^T p + h(x) \) where \( n = 6, Q = cc^T + I, p = 1, c = [1 2 3 4 5 6]^T \) when \( h(\cdot) = h_1(\cdot) \) and \( h(\cdot) = h_3(\cdot) \). The parameters for \( h_i(x) \) are set to \( \lambda = 0.01 \) and \( \rho = +\infty \). The stationary point distribution on this example can be found in Table 1. This problem contains \( \sum_{i=0}^{56} C_6^i = 64 \) stationary points. There are 56 and 58 local minimizers satisfying the \( L \)-stationary condition for binary optimization and sparse optimization problem, respectively. There are 9 and 11 local minimizers satisfying the block-1 stationary condition. Moreover, as \( k \) becomes large, the newly introduced type of local minimizers (i.e. block-\( k \) stationary point) become more restricted in the sense that they have small number of stationary points.

### 3.2 Convergence Analysis

This subsection provides some convergence analysis for Algorithm \( \Pi \). We assume that the working set of size \( k \) is selected uniformly. We let \( x' \) be the sequence generated by Algorithm \( \Pi \) and \( \bar{x} \) be the block-\( k \) stationary of \( \Pi \).

**Theorem 1.** Convergence Properties for \( h = h_1 \) and \( h = h_3 \). We have the following results. (i)

\[
F(x^{t+1}) - F(x^t) \leq -\frac{\theta}{2} \|x^{t+1} - x^t\|^2
\]  

(ii) It holds that \( \lim_{t \to \infty} \mathbb{E}[\|x^{t+1} - x^t\|] = 0 \). (iii) As \( t \to \infty \), \( x' \) converges to the block-\( k \) stationary point \( \bar{x} \) of \( \Pi \). (iv) For \( h = h_1 \), we have \( \mathbb{E}[\|x^{t+1} - x^t\| | x^t] \geq \sqrt{2} / C_n^k \).

The solution changes at most \( \sqrt{2} C_n^k (F(x^0) - F(\bar{x}))/\theta \) times in expectation for finding a block-\( k \) stationary point. (iv) For \( h = h_3 \), we have \( \|x^t_i\| \geq \delta \) for all \( i \) with \( x^t_i \neq 0 \), where \( \delta = \min_j \{\min(\rho, \sqrt{2}/(\theta + L))\} \). Moreover, it holds that: \( \mathbb{E}[\|x^{t+1} - x^t\| | x^t] \geq \delta / C_n^k \).

The solution changes at most \( T_{out} = 2 C_n^k (F(x^0) - F(\bar{x}))/\theta \) times in expectation for finding a block-\( k \) stationary point.

**Proof.** (i) Due to the optimality of \( x^{t+1} \), we have: \( F(x^{t+1}) + \frac{\theta}{2} \|x^{t+1} - x^t\|^2 \leq F(u) + \frac{\theta}{2} \|u - x^t\|^2 \) for all \( u \in x^t \). Letting \( u = x^t \), we obtain the sufficient decrease condition in (3).

(ii) Taking the expectation of \( B \) for the sufficient descent inequality, we have \( \mathbb{E}[F(x^{t+1}) | x^t] \leq F(x^t) - \frac{\theta}{2} \|x^{t+1} - x^t\|^2 \). Summing this inequality over \( t = 0, 1, 2, \ldots, T \), we have:

\[
\sum_{t=0}^{T} \mathbb{E}[\|x^{t+1} - x^t\|^2 | x^t] \leq F(x^0) - F(x^T)\]  

Using the fact that \( F(\bar{x}) \leq F(x^t) \), we obtain:

\[
\min_{t=1, \ldots, T} \mathbb{E}[\|x^{t+1} - x^t\|^2 | x^t] \leq \frac{T}{T} \sum_{t=0}^{T} \mathbb{E}[\|x^{t+1} - x^t\|^2 | x^t] \leq \frac{F(x^0) - F(\bar{x})}{T} \]  

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

(iii) We assume the conclusion does not hold. In expectation there exists a block of coordinates \( B \) such that \( x^t \notin \text{arg} \min_{\mathcal{P}} \mathcal{P}(z; x^t, B) \) for some \( B \), where \( \mathcal{P}(\cdot) \) is defined in Definition 3. However, according to the fact that \( x^t = x^{t+1} \) and step S2 in Algorithm \( \Pi \), we have \( x^t \notin \text{arg} \min_{\mathcal{P}} \mathcal{P}(z; x^t, B) \). Hence, we have \( x^{t+1}_B \neq x^t_B \). This contradicts with the fact that \( x^t = x^{t+1} \) as \( t \to \infty \). We conclude that \( x^t \) converges to the block-\( k \) stationary point.

(iv) We observe that when the current solution \( x^t \) changes, we have \( \|x^{t+1} - x^t\| \geq \sqrt{2} \) if \( x^t \neq x^{t+1} \). Noticing that there are \( C_n^k \) possible combinations of choice for the working set of size \( k \). Thus, we have \( \mathbb{E}[\|x^{t+1} - x^t\| | x^t] \geq \sqrt{2} / C_n^k \). From (4), we obtain: \( [F(x^0) - F(\bar{x})]/(\theta) \geq \frac{\sqrt{2} / C_n^k}{\theta} \) Therefore, the number of iterations is upper bounded by \( t \leq \sqrt{2} C_n^k (F(x^0) - F(\bar{x}))/\theta \) times in expectation.

(iv) Note that Algorithm \( \Pi \) solves problem (2) in every iteration. Using Proposition 1 we have that the solution \( x^{t+1}_B \) is also a \( L \)-stationary point. Therefore, we have \( |x^{t+1}_i| \geq \min(\rho, \sqrt{2}/(\theta + L)) \) for all \( x^{t+1}_i \neq 0 \). Taking the initial point of \( x \) for consideration, we have that: \( |x^{t+1}_i| \geq \)
We now prove the convergence rate of Algorithm 1 for binary optimization. We define $\Pi(x) = \arg\min_x \|x - a\|$, s.t. $x \in \{-1, +1\}^n$. Our key observation is that when $\Pi(x) \neq \Pi(y)$, it holds $\|\Pi(x) - x\|_2 \leq (1 - \kappa)\|\Pi(y) - x\|_2$ with $0 < \kappa < 1$ for any $y$. Our analysis combines this observation with the strongly convex property of $f(\cdot)$ to provide $Q$-linear convergence rate of Algorithm 1.

**Theorem 2. Proof of Convergence Rate when $f(\cdot)$ is $s$-Strongly Convex and $h \triangleq f_b$.** We define $\beta \triangleq \left[\frac{1}{2(L + \theta - s)}(1 - \kappa) \kappa + 1 - \kappa\right]_+^k < 1$. In expectation, we have: $F(x^t) - F(\bar{x}) \leq (1 - \beta^t)(F(x^0) - F(\bar{x}))$. In other words, it takes at most $\log_\beta(1 - \beta)(\frac{F(x^0) - F(\bar{x})}{F(x^0) - F(\bar{x})})$ times to find a solution $x^t$ satisfying $F(x^t) - F(\bar{x}) \leq \epsilon$.

**Remarks:** The strongly convexity assumption always holds for binary optimization, since one can append an additional term $\frac{\eta}{2}\|x\|^2$ to $f(\cdot)$ with sufficiently large $\eta$.

We now prove the convergence rate of Algorithm 1 for sparse optimization. We have derived the upper bound for the number of changes $T_{\text{out}}$ for the support set in Theorem 2. We now need to derive a bound on the number of iterations $T_{\text{in}}$ performed after the support set is fixed. Combining these two bounds, we complete our proof.

**Theorem 3. Proof of Convergence Rate when $f(\cdot)$ is Convex and $h \triangleq f_b$.** We define $\nu \triangleq \frac{2\kappa\rho}{\sqrt{2\lambda}}$ and $T_{\text{in}} \triangleq \max\{4\nu^2/\theta, \nu\sqrt{2\nu^2/\theta}\}/\epsilon$. We have the following results: (i) When the support set does not change, it takes at most $T_{\text{in}}$ iterations for Algorithm 2 to converge to a point $x^t$ satisfying $F(x^t) - F(\bar{x}) \leq \epsilon$. (ii) It takes at most $T_{\text{in}} \times T_{\text{out}}$ iterations to find a point $x^t$ satisfying $F(x^t) - F(\bar{x}) \leq \epsilon$.

4 Experimental Validation

In this section, we demonstrate the effectiveness of our proposed algorithm on three discrete optimization tasks, namely sparse regularized least square problem, binary constrained least square problem, and sparse constrained least square problem. We use HYBRID (Ri-$G_j$) to denote our hybrid method along with selecting $i$ coordinate using Random strategy and $j$ coordinates using the Greedy strategy. Since these two strategies may select the same coordinates, the working set at most contains $i + j$ coordinates. We set $\theta = 10^{-5}$ for HYBRID. We keep a record of the relative changes of the objective function values by $r_t = \{f(x^t) - f(x^{t+1})\}/f(x^t)$. We let our algorithms run up to $T$ iterations and stop them at iteration $t < T$ if mean($|r_{t \leftarrow \min(t, M) + 1}, r_{t \leftarrow \min(t, M) + 2}, \ldots, r_t|\) $ $\leq \epsilon$. The defaults values of $\epsilon, M,$ and $T$ are $10^{-5}, 50$ and $1000$, respectively. All codes were implemented in Matlab on an Intel 3.20GHz CPU with 8 GB RAM. For further evaluation, we provide some Matlab code in the supplementary material.

We also apply our method to solve dense subgraph discovery problem and our experiments have shown that our method achieves state-of-the-art performance. Due to space limitation, we place them into the supplementary material.
We consider the following sparse constrained least squares problem:

\[
\min_{\mathbf{x}} \frac{1}{2} \| \mathbf{A} \mathbf{x} - \mathbf{b} \|^2_2 \quad \text{s.t.} \quad \| \mathbf{x} \|_0 \leq s.
\]

We generate the working set in our forthcoming experiments. Hybrid(R0-G10) converges quickly but it generally leads to worse solution quality than Hybrid(R10-G0). From the experiments, we observe that although HYBRID uses some randomized strategies to find the working set, we can always measure the quality of the solution by computing the deterministic objective value.

**Compared Methods.** We compare the proposed method (HYBRID) with four state-of-the-art methods: (i) Proximal Point Algorithm (PPA) [53], (ii) Accelerated PPA (APPA) [53, 4], (iii) Matrix Splitting Method (MSM) [53], and (iv) Greedy Pursuit Method (GPM).

**Experimental Results.** Several observations can be drawn from Figure 1. (i) PPA and APPA achieve nearly the same performance and they get stuck into poor local minima. (ii) MSM improves over PGA and APGA, and GPM consistently outperforms MSM. (iii) Our proposed hybrid method is more effective than MSM and GPM. In addition, we find that as the parameter \( k \) becomes larger, more higher accuracy is achieved. (iv) HYBRID appears to be less sensitive to initialization and it converges to similar objective values when using different initializations. (iv) We notice that Hybrid(R0-G10) converges quickly but it generally leads to worse solution quality than Hybrid(R10-G0). Based on this observation, we consider a combined random and greedy strategies for finding the working set in our forthcoming experiments.

### 4.1 Binary Constrained / Sparse Regularized Least Squares Problem

Given a design matrix \( \mathbf{A} \in \mathbb{R}^{m \times n} \) and an observation vector \( \mathbf{b} \in \mathbb{R}^m \), sparse regularized / binary constrained least square problem is to solve the following optimization problem:

\[
\min_{\mathbf{x} \in \{-1, 1\}^n} \frac{1}{2} \| \mathbf{A} \mathbf{x} - \mathbf{b} \|^2_2 \quad \text{or} \quad \min_{\mathbf{x}} \frac{1}{2} \| \mathbf{A} \mathbf{x} - \mathbf{b} \|^2_2 + \lambda \| \mathbf{x} \|_0
\]

We generate \( \mathbf{A} \in \mathbb{R}^{200 \times 500} \) and \( \mathbf{b} \in \mathbb{R}^{200} \) from a (0-1) uniform distribution. We set \( \lambda = 0.1 \). Note that although HYBRID uses some randomized strategies to find the working set, we can always measure the quality of the solution by computing the deterministic objective value.

**Compared Methods.** We compare the proposed method (HYBRID) with four state-of-the-art methods: (i) Proximal Point Algorithm (PPA) [53], (ii) Accelerated PPA (APPA) [53, 4], (iii) Matrix Splitting Method (MSM) [53], and (iv) Greedy Pursuit Method (GPM).

**Experimental Results.** Several observations can be drawn from Figure 1. (i) PPA and APPA achieve nearly the same performance and they get stuck into poor local minima. (ii) MSM improves over PGA and APGA, and GPM consistently outperforms MSM. (iii) Our proposed hybrid method is more effective than MSM and GPM. In addition, we find that as the parameter \( k \) becomes larger, more higher accuracy is achieved. (iv) HYBRID appears to be less sensitive to initialization and it converges to similar objective values when using different initializations. (iv) We notice that Hybrid(R0-G10) converges quickly but it generally leads to worse solution quality than Hybrid(R10-G0). Based on this observation, we consider a combined random and greedy strategies for finding the working set in our forthcoming experiments.

### 4.2 Sparse Constrained Least Squares Problem

We consider the following sparse constrained least squares problem:

\[
\min_{\mathbf{x}} \frac{1}{2} \| \mathbf{A} \mathbf{x} - \mathbf{b} \|^2_2 \quad \text{s.t.} \quad \| \mathbf{x} \|_0 \leq s,
\]
In our experiments, to generate the sparse original signal $\mathbf{x} \in \mathbb{R}^n$, we select a support set of size 100 uniformly at random and set them to arbitrary number sampled from standard Gaussian distribution. In order to verify the robustness of the comparing methods, we generate the design matrix $\mathbf{A}$ and the noise vector $\mathbf{o} \in \mathbb{R}^m$ with and without outliers, as follows:

\begin{align*}
\text{AI: } & \mathbf{A} = \text{randn}(m, n), \quad \text{AII: } \mathbf{A} = \mathcal{P}(\text{randn}(m, n)); \\
& \text{bl: } \mathbf{o} = 10 \times \text{randn}(m, 1), \quad \text{bII: } \mathbf{o} = \mathcal{P}(10 \times \text{randn}(m, 1))
\end{align*}

where randn($m, n$) is a function that returns a standard Gaussian random matrix of size $m \times n$, $\mathcal{P}(\mathbf{X}) \in \mathbb{R}^{m \times p}$ denotes a noisy version of $\mathbf{X} \in \mathbb{R}^{m \times p}$ where 2% of the entries of $\mathbf{X}$ are corrupted uniformly by scaling the original values by 100 times4. The observation vector is generated via $\mathbf{b} = \mathbf{A}\mathbf{x} + \mathbf{o}$. Note that the Hessian matrix can be ill-conditioned for the ‘AII’ type design matrix. We vary $n$ from $\{256, 512, 1024, 2048\}$ and fix $m$ to 512. We swap the parameter $s$ over $\{3, 8, 13, 18...50\}$.

**Compared Methods.** We compare the proposed hybrid algorithm with seven state-of-the-art sparse optimization algorithms: (i) Regularized Orthogonal Matching Pursuit (ROMP) [33], (ii) Subspace Pursuit (SSP) [15], (iii) Orthogonal Matching Pursuit (OMP) [43], (iv) Gradient Pursuit (GP) [7], (v) Compressive Sampling Matched Pursuit (CoSaMP) [32], (vi) Proximal Point Algorithm (PPA) [2], and (vii) Quadratic Penalty Method (QPM) [30]. We remark that ROMP, SSP, OMP, GP and CoSaMP are greedy algorithms and their support sets need to be selected iteratively. They are non-gradient type algorithms, it is hard to incorporate these methods into other gradient-type based optimization algorithms [2]. We use the Matlab implementation in the ‘sparsify’ toolbox4. Both PPA and QPM are based on iterative hard thresholding. Since the optimal solution is expected to be sparse, we initialize the solutions of PPA, QPM and HYBRID to $10^{-12}$ $\times$ randn($n, 1$) and project them to feasible solutions. The initial solution of greedy pursuit methods are initialized to zero points implicitly. We show the average results of using 3 random initial points.

**Experimental Results.** Several conclusions can be drawn from Figure 2. (i) PPA and QPM generally lead to the worst performance. (ii) ROMP and COSAMP are not stable and sometimes they present bad performance. (iii) SSP, OMP and GP generally present comparable performance to HYBRID when the Hessian matrix is well-conditioned (for ‘AI’ type design matrix) but present much worse performance than HYBRID when the Hessian matrix is ill-conditioned (for ‘AII’ type design matrix).

4.3 Computational Efficiency of Algorithm 1

We show the convergence curve of different methods for solving sparse constrained least squares problem with \{m, n, k\} = \{512, 2048, 20\}. Left: ‘AI’+ ‘bI’; Right: ‘AII’+ ‘bII’.

The parameter $k$ in Algorithm 1 can be viewed as a tuning parameter to balance the efficacy and efficiency. One can further accelerate the algorithm using asynchronous parallelism or mini-batch optimization techniques.

5 Conclusions

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.

---

4 Matlab script: I = randperm(m*p,round(0.02*m*p)); X(I) = X(I)*100.

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

[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.

[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.

[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.

[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.

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

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

[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.

[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.

[10] Gruiu Călinescu, Chandra Chekuri, Martin Pál, and Jan Vondrák. Maximizing a monotone submodular function subject to a matroid constraint. *SIAM Journal on Computing (SICOMP)*, 40(6):1740–1766, 2011.

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

[12] 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.

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

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

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

[16] 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.

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

[18] 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.

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

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

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

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

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

[24] 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.

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

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

[27] 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.

[28] Zhaosong Lu. Iterative hard thresholding methods for \(\ell_0\) regularized convex cone programming. *Mathematical Programming*, 147(1-2):125–154, 2014.
[29] Zhaosong Lu and Lin Xiao. On the complexity analysis of randomized block-coordinate descent methods. *Mathematical Programming*, 152(1-2):615–642, 2015.

[30] Zhaosong Lu and Yong Zhang. Sparse approximation via penalty decomposition methods. *SIAM Journal on Optimization (SIOPT)*, 23(4):2448–2478, 2013.

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

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

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

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

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

[36] Nam Nguyen, Deanna Needell, and Tina Woolf. Linear convergence of stochastic iterative greedy algorithms with sparse constraints. *arXiv Preprint*, 2014.

[37] 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.

[38] Andrei Patrascu and Ion Necoara. Random coordinate descent methods for ℓ0 regularized convex optimization. *IEEE Transactions on Automatic Control*, 60(7):1811–1824, 2015.

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

[40] 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.

[41] 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.

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

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

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

[45] 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.

[46] 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.

[47] Baoyuan Wu and Bernard Ghanem. ℓp-box adm: A versatile framework for integer programming. *arXiv Preprint*, 2016.

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

A Proof of proposition 1 (Proof of Optimality Hierarchy)

Proof. For notational convenience, we denote \( \Pi(a) \triangleq \min(\rho 1, \max(-\rho 1, a)) \).

(1) We now prove that an \( L \)-stationary point \( \bar{x} \) is also a basic stationary point \( \bar{x} \). When \( h \triangleq h_b \), this conclusion clearly holds. We now consider \( h \triangleq h_s \). Observing that the problem (see Definition 1) is separable, we have the following closed-form solution: \( \bar{x}_S = \Pi(\bar{x}_S - (\nabla f(\bar{x}))_S/L) \).

For the problem in Definition B, we have the following closed form solution for \( x \): \( x_i = \begin{cases} \Pi(x_i - \nabla_f(\bar{x})/L), & (x_i - \nabla_f(\bar{x})/L)^2 > 2\lambda/L; \\ 0, & (x_i - \nabla_f(\bar{x})/L)^2 \leq 2\lambda/L. \end{cases} \)

Clearly, the latter formulation implies the former one. Moreover, it is not hard to notice that for all \( i \) that \( \bar{x}_i = 0 \), we have \( |\nabla_i f(\bar{x})| \leq \sqrt{2\lambda L} \), and for all \( j \) that \( \bar{x}_j \neq 0 \), we have \( \nabla_j f(\bar{x}) = 0 \) and \( |\bar{x}_j| \geq \min(\rho, \sqrt{2\lambda L}) \).

(2) Note that the convex objective function \( f(\cdot) \) has coordinate-wise Lipschitz continuous gradient and it holds that \( [3,4]: f(\bar{x} + \delta e_i) \leq Q_i(\bar{x}, \delta) \triangleq f(\bar{x}) + \langle \nabla_i f(\bar{x}), \delta e_i \rangle + \frac{\lambda}{\rho} \|\delta e_i\|^2, \quad \forall \bar{x} \in \mathbb{R}^n, \delta \in \mathbb{R}, \quad i = 1, 2, \ldots, n \). For notational convenience, we denote \( \Pi(\bar{x}, \delta) \triangleq \Pi(\bar{x}) + \Pi(\bar{x} + \delta) \) for \( h = h_s \), we have the following result: \( \Pi(\bar{x}, \delta) \leq \Pi(\bar{x} + \delta) \).

(3) We now show that block-\( k \) stationary point implies block-\( k \) stationary point when \( k_1 \geq k_2 \).}

(4) It is clearly that any block-\( n \) stationary point is also the global optimal solution.

\( \square \)

B Proof of Convergence Rate

This section provides the proof of convergence rate of Algorithm 1 when the random strategy for finding the working set is used. We notice that the working set contains \( \binom{C_k}{n} \) possible different combinations and each combination contains \( k \) coordinates. We uniformly select one combination from the working set as \( B_i \) in \( i \)-th iteration. Note that every \( B_i \) corresponds to a unique \( U_i \) and \( x_{B_i} \) can be rewritten as \( x_{B_i} = (U_i)^T \bar{x} \) for some suitable binary matrix with \( (1 U)^T = 1 \) and \( U \in \{0, 1\}^n \).

Sometimes, we use \( x_i \) to denote \( x_{B_i} \) for brevity. It is not hard to verify that the following always holds:

\( \frac{1}{n} \sum_{i=1}^{C_k} U_i \) x_{B_i} = \frac{\bar{x}}{n} \cdot \bar{x}.

B.1 Proof of Theorem 2 (Convergence Rate for Binary Optimization)

In what follows, we study the convergence rate of Algorithm 1 for binary optimization (i.e. \( h \triangleq h_b \)).

The following lemmas are useful in our proof.

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

\[ \|\Pi(x) - x\|^2 \leq (1 - \kappa)\|\Pi(y) - x\|^2 \]

with \( \kappa = 0 \). Moreover, if \( \Pi(x) \neq \Pi(y) \), there exist a small \( 0 < \kappa < 1 \) such that (7) holds.

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

\[ \|\Pi(x) - x\|^2 \leq (1 - \kappa)\|\Pi(y) - x\|^2 \]

\[ \iff \kappa\|\Pi(y) - x\|^2 + \|\Pi(x)\|^2 - \|\Pi(x), x\|^2 \leq \|\Pi(y)\|^2 + \|x\|^2 - 2\|\Pi(y), x\| \]

\[ \iff \kappa\|\Pi(y) - x\|^2 \leq 2\|\Pi(x) - \Pi(y), x\| \]
Note that $\Pi(x) \neq \Pi(y)$ also implies $x \neq \Pi(y)$ and we have $\|\Pi(y) - x\|_2^2 > 0$. Moreover, we notice that $x_i \cdot \text{sign}(y_i) \leq x_i \cdot \text{sign}(x_i)$ for all $i$ and we have $(\Pi(x) - \Pi(y), x) > 0$. Therefore, there exists a sufficient small $\kappa$ such that $\kappa \|\Pi(y) - x\|_2^2 \leq 2(\Pi(x) - \Pi(y), x)$ holds. This finishes the proof of this lemma. \hfill \square

**Lemma 2.** Assume that $f(\cdot)$ is $\alpha$-strongly convex. The following inequality holds for any $x$ and $y$:

$$\frac{\alpha^2}{2} \|y - x\|_2^2 - \|\nabla f(x)\|_2^2 \leq -\frac{\alpha}{2}(f(x) - f(y))$$  \hspace{1cm} (7)

**Proof.** We naturally derive the following results:

$$\|\nabla f(x)\|_2^2 - \frac{\alpha^2}{2} \|y - x\|_2^2 = (\|\nabla f(x)\|_2 - 2 \frac{\alpha}{2} \|x - y\|_2) \cdot (\|\nabla f(x)\|_2 + 2 \frac{\alpha}{2} \|y - x\|_2)/\|y - x\|_2$$

$$\geq (\langle \nabla f(x), y - x \rangle - \frac{\alpha^2}{2} \|y - x\|_2^2) \cdot (\|\nabla f(x)\|_2 + 2 \frac{\alpha}{2} \|y - x\|_2)/\|y - x\|_2$$

$$\geq (f(x) - f(y)) \cdot (0 + \frac{\alpha}{2})$$

$$= \frac{\alpha}{2}(f(x) - f(y))$$

where step (a) uses the Cauchy-Schwarz inequality; step (b) uses the $\alpha$-strongly convexity condition that $f(x) - f(y) \leq \langle \nabla f(x), x - y \rangle - \frac{\alpha}{2} \|x - y\|_2^2$. \hfill \square

**(Proof of Theorem 2.)**

**Proof.** We define

$$z(x) \triangleq \arg \min_{x \in \mathbb{R}^n} H(x, z), \quad H(x, z) \triangleq f(x) + \langle \nabla f(x), z \rangle + \frac{L + \alpha}{2} \|z\|_2^2 + h_b(x + z)$$

We have the following inequalities:

$$\mathbb{E}[F(x^{t+1}) \mid x^t]$$

$$\leq(a) \mathbb{E}[F(x^{t+1} + U_t z_t^t(x^t))] + \frac{\alpha}{2} \|z_t^t(x^t)\|_2^2 - \frac{\alpha}{2} \|x^{t+1} - x^t\|_2^2$$

$$\leq(b) \mathbb{E}[F(x^t + U_t z_t^t(x^t))] + \frac{\alpha}{2} \|z_t^t(x^t)\|_2^2$$

$$\leq(c) \mathbb{E}[f(x^t) + \langle \nabla f(x^t), z_t^t(x^t) \rangle + \frac{L + \alpha}{2} \|z_t^t(x^t)\|_2^2 + h_b(x_t^t + z_t^t(x^t))]$$

$$\leq(d) f(x^t) + \mathbb{E}[\langle \nabla f(x^t), z_t^t(x^t) \rangle + \frac{L + \alpha}{2} \|z_t^t(x^t)\|_2^2 + 0]$$

$$\leq(e) f(x^t) + \frac{\alpha}{n} \langle \nabla f(x^t), z_t^t(x^t) \rangle + \frac{\alpha}{n} \frac{L + \alpha}{2} \|z_t^t(x^t)\|_2^2$$  \hspace{1cm} (8)

where step (a) uses the optimality condition of $x^{t+1}$; step (b) uses the fact that $-\frac{\alpha}{2} \|x^{t+1} - x^t\|_2^2 \leq 0$; step (c) uses the Lipschitz continuity of the gradient of $f(\cdot)$; step (d) uses the definition of $h_b(\cdot)$; step (e) uses the fact that each block $B$ is picked randomly with probability $k/n$.  

12
For any \( x^i \in \Psi \) and \( x^{i+1} \in \Psi \), we naturally derive the following inequalities:

\[
\frac{1}{n} \mathbb{E}[f(x^{i+1})|x^i] - \frac{1}{n} f(x^i) 
\leq \langle \nabla f(x^i), z(x^i) \rangle + \frac{L_+ \theta}{2} \|z(x^i)\|^2 \\
\langle \nabla f(x^i), \Pi \left( x^i - \nabla f(x^i)/(L + \theta) \right) - x^i \rangle + \frac{L_+ \theta}{2} \| \Pi \left( x^i - \nabla f(x^i)/(L + \theta) \right) - x^i \|^2 \\
\frac{L_+ \theta}{2} \| \Pi \left( x^i - \nabla f(x^i)/(L + \theta) \right) - x^i + \nabla f(x^i)/(L + \theta) \|^2 - \frac{L_+ \theta}{2} \| \nabla f(x^i)/(L + \theta) \|^2 \\
\leq \frac{(L+\theta)(1-\kappa)}{2} \| \Pi(\tilde{x}) - x^i \|^2 + (1-\kappa) \| \tilde{x} - x^i, \nabla f(x^i) \| - \frac{\kappa}{2} \| \nabla f(x^i)/(L + \theta) \|^2 \\
\leq \frac{(L+\theta)(1-\kappa)}{2} \| x^i - x^i \|^2 + \frac{\kappa}{2} (L + \theta)^2 (f(\tilde{x}) - f(x^i)) + (1-\kappa) (f(\tilde{x}) - f(x^i)) \\
(f) \leq \frac{(L+\theta)(1-\kappa)}{2} \| x^i - x^i \|^2 + \frac{\kappa}{2} (L + \theta)^2 (f(\tilde{x}) - f(x^i)) + (1-\kappa) (f(\tilde{x}) - f(x^i))
\]

where step (a) uses (b); step (b) uses the fact that the problem of \( \arg\min_d \mathbb{E} = H_i(x, d) \) admits a closed-form solution with \( z(x) = \Pi(\tilde{x} - x, \nabla f(x)/(L + \theta) - x) \); step (c) uses the inequality in (d); step (d) uses the fact that \( \Pi(\tilde{x}) = \tilde{x} \); step (e) uses the strongly convexity of \( f(\cdot) \) that \( f(x^t) - f(\tilde{x}) \leq \langle \nabla f(x^t), x^t - \tilde{x} \rangle - \frac{\kappa}{2} \| x^t - \tilde{x} \|^2 \); step (f) uses the inequality in (g).

We have the following inequality: \( E[F(x^{i+1}) | x^i] - F(x^i) \leq \beta (F(\tilde{x}) - F(x^i)) \). Rearranging terms, we obtain that \( F(x^{i+1}) - F(\tilde{x}) \leq (1-\beta) (F(\tilde{x}) - F(x^i)) \). In other words, the sequence \( \{ f(x^i) \} \) converges to the stationary point linearly in the quotient sense. Solving this recursive formulation, we obtain: \( F(x^i) - F(\tilde{x}) \leq (1-\beta)^i (F(\tilde{x}^0) - F(\tilde{x})) \). Therefore, we conclude that it takes at most \( \log_{1-\beta} (F(\tilde{x}^0) - F(\tilde{x})) \) times to find a local optimal solution satisfying \( F(x^i) - F(\tilde{x}) \leq \epsilon \).

\[ \Box \]

B.2 Proof of Theorem 3 (Convergence Rate for Sparse Optimization)

In what follows, we establish the convergence rate for sparse optimization.

We notice that when the support set is fixed, the original problem reduces to the following convex composite optimization problem:

\[ \mathcal{J}(x) \triangleq f(x) + p(x), \quad \text{with} \quad p(x) \triangleq I_{\Omega}(x) \quad (9) \]

The following lemma is useful in our proof.

**Lemma 3.** (Cost-to-Go Condition) Assume that the working set of size \( k \) is selected uniformly at \( t \)-th iteration. We have the following inequality in expectation for some constant \( \nu \triangleq \frac{2k\theta}{n\sqrt{n}} > 0 \):

\[ \mathcal{J}(x^{t+1}) - \mathcal{J}(\tilde{x}) \leq \nu \| x^{t+1} - x^t \|^2 \quad (10) \]

**Proof.** First of all, by the optimality condition of \( x^{t+1} \), we have:

\[ (\partial \mathcal{J}(x^{t+1}))_B + \theta(x^{t+1} - x^t)_B = 0, \quad x_N^{t+1} = x_N^t. \quad (11) \]

We derive the following inequalities:

\[ \mathbb{E}[\mathcal{J}(x^{t+1})|x^t] - \mathcal{J}(\tilde{x}) \quad (a) \leq \mathbb{E}[\langle \partial \mathcal{J}(x^{t+1}), x^{t+1} - \tilde{x} \rangle] \\
\leq \frac{1}{n} \langle (\partial \mathcal{J}(x^{t+1}))_B, (x^{t+1} - \tilde{x})_B \rangle \\
\leq \frac{1}{n} \langle -\theta(x^{t+1} - x^t)_B, (x^{k+1} - \tilde{x})_B \rangle \\
\leq \frac{1}{n} \| (x^{t+1} - x^t)_B \|_2 \cdot \| (x^{t+1} - \tilde{x})_B \|_2 \\
\leq \frac{\nu}{n} \theta \| x^{t+1} - x^t \|_2 \]

\[ \Box \]
where step (a) uses the convexity of $J$; step (b) uses the fact that each block $B$ is picked randomly with probability $k/n$; step (c) uses (11); step (d) uses the Cauchy-Schwarz inequality; step (e) uses that $\|x^{k+1} - \bar{x}\| \leq \frac{1}{\sqrt{n}} \|x^{k+1} - \bar{x}\|_\infty \leq \frac{2}{\sqrt{n}}$.

\[ \]  

Lemma 4. Assume a nonnegative sequence \( \{u^t\}_{t=0}^\infty \) satisfies \((u^{t+1})^2 \leq C(u^t - u^{t+1})\) for some constant $C$. We have:

\[
 u^t \leq \frac{\max(2C, \sqrt{Cu^d})}{t}
\]  

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

\[
 u^{t+1} \leq - \frac{C}{2} + \frac{C}{2}\sqrt{1 + \frac{4u^t}{C}}
\]  

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 $u^1 \leq -\frac{2C}{2} + \frac{2C}{2}\sqrt{1 + \frac{4u^0}{2C}} = \frac{C}{2}\sqrt{\frac{4u^0}{2C}} = \sqrt{Cu^0} \leq \frac{C_1}{1}$. (ii) When $t \geq 1$, we assume that $u^t \leq \frac{C_1}{t}$ holds. We derive the following results: $t \geq 1 \Rightarrow \frac{t+1}{t} \leq 2$ \(\begin{align*}
 C \frac{t^2}{t+1} &\leq C_1 \Rightarrow C \frac{t^2}{t+1} \leq \frac{C_1}{t+1} \Rightarrow C \frac{t^2}{t+1} \leq \frac{C_1}{t+1} \Rightarrow C \frac{t^2}{t+1} \leq \frac{C_1}{t+1} \Rightarrow C \frac{t^2}{t+1} \leq \frac{C_1}{t+1} \Rightarrow \frac{C^2}{t^2} \leq \frac{C_1}{t+1} \Rightarrow -C + \frac{C}{2}\sqrt{1 + \frac{4u^t}{C}} \leq \frac{C_1}{t+1} \Rightarrow -C + \frac{C}{2}\sqrt{1 + \frac{4u^t}{C}} \leq \frac{C_1}{t+1} \Rightarrow u^{t+1} \leq \frac{C_1}{t+1}.
\end{align*} \)

Here, step (a) uses $2C \leq C_1$; step (b) uses $\frac{1}{\sqrt{t(t+1)}} = \frac{1}{t} - \frac{1}{t+1}$; step (c) uses $\frac{u^t}{t} \leq \frac{C_1}{2}$. 

\[ \]  

Proof of Theorem 3.

Proof. When the support set does not change, the optimization problem reduced to the convex optimization as in (9) with $F(x) = J(x)$. We derive the following results:

\[ J(x^{t+1}) - J(x) \leq u||x^{t+1} - x^t||_2 \leq \sqrt{\frac{2}{\theta}[J(x^t) - J(x^{t+1})]} \]  

where the step (a) uses Lemma 3 step (b) uses the sufficient decent condition.

We denote $\Delta^t \equiv J(x^t) - J(\bar{x})$ and $C \equiv \frac{2\sigma^2}{\theta}$ we have the following inequality:

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

Using Lemma 4 we have:

\[ F(x^t) - F(\bar{x}) = J(x^t) - J(\bar{x}) \leq \max\left(\frac{4\sigma^2}{\theta}, \sqrt{\frac{2\sigma^2 u^d}{\theta}}\right) \]  

We conclude that it takes $T_{in}$ iterations in expectation to converge to a local optimal solution that satisfies $F(x^t) - F(x^*) \leq \epsilon$. Moreover, from Theorem 1 we have that Algorithm 1 changes at most $T_{out}$. Therefore, we conclude that it takes $T_{in} \times T_{out}$ iteration to converge to the local optimal solution. This finishes the proof of this Theorem.

\[ \]  

C Additional Experiments

C.1 Dense Subgraph Discovery

Dense subgraph discovery is an important application of binary optimization. It aims at finding the maximum density subgraph on $s$ vertices [40, 19, 54], which can be formulated as the following binary program:

\[
\min_{x \in \{0, 1\}^n} - x^T W x, \text{ s.t. } x^T 1 = s,
\]  

(17)
where $W \in \mathbb{R}^{n \times n}$ is the adjacency matrix of a graph.

**Compared Methods.** We compare our HYBRID method on eight datasets (refer to the sub-captions in Figure 4) against six methods: (i) LP relaxation \([52]\). (ii) Feige’s greedy algorithm (GEIGE) \([19]\). (iii) Truncated Power Method (TPM) \([54]\). (iv) L2box-ADMM \([47]\). (v) MPEC-EPM \([52]\). For more description of these methods, we refer to \([52, 50]\). We show the average results of using 3 random initial points.

**Experimental Results.** Several observations can be drawn from Figure 4: (i) FEIGE generally fails to solve the dense subgraph discovery problem and it leads to solutions with low density. (ii) TPM gives better performance than state-of-the-art technique MPEC-EPM in some cases but it is unstable. (iii) Our proposed HYBRID generally outperforms all the compared methods.

**C.2 More Results for Sparse Constrained Least Squares Problem**

In our experiments, we also vary $m = \{128, 256, 512, 1024\}$ and fix $n = 2048$ on different types of $A$ and $b$. Our results shown in Figure 5 consolidate our conclusions in our submission.