Author et al.: A Second Order Algorithm for MCP Regularized Optimization
I. INTRODUCTION

In this paper, we focus on the MCP-regularized optimization problem

\[ \min \varphi(x) := f(x) + \sum_{i=1}^{n} r_{i}(x_{i}), \tag{1} \]

where \( f \) is continuously differentiable and

\[ r_{i}(x_{i}) = \begin{cases} \lambda |x_{i}| - \frac{\lambda^{2}}{2\theta}, & \text{if } |x_{i}| < \theta \lambda, \\ \frac{|x_{i}|}{\theta}, & \text{if } |x_{i}| \geq \theta \lambda, \end{cases} \]

where \( \lambda > 0 \) and \( \theta > 1 \). The MCP penalty is a fast, continuously unbiased and sparse penalty function [43]. An important application of (1), found in the signal processing literature, is the problem

\[ \min \frac{1}{2} \|Ax - b\|^{2} + \sum_{i=1}^{n} r_{i}(x_{i}), \tag{2} \]

which approximates the known \( \ell_{2}-\ell_{0} \)

\[ \min \frac{1}{2} \|Ax - b\|^{2} + \|x\|_{0}, \tag{3} \]

where \( A \in \mathbb{R}^{m \times n} \) and \( b \in \mathbb{R}^{m} \) and the \( \ell_{0} \) norm counts the nonzero entries of \( x \). The problems (1)-(3) are used in many applications, such as in compressive sensing, high-dimensional variable selection, data fitting and image reconstruction ([5], [28], [40]).

A. PREVIOUS WORKS

Problem (3) is known to be NP-hard ([12], [30]) since it involves the \( \ell_{0} \) norm and hence is a nonconvex and discontinuous optimization problem. Several different classes of approaches have been proposed to solve problem (3). Convex relaxation is the first class of methods in which the \( \ell_{0} \)-norm is replaced by the convex \( \ell_{1} \)-norm. One of the most popular methods in convex relaxation is the class of iterative shrinkage/thresholding algorithms (ISTA) and various variants ([2], [13], [25], [26], [40], [41]), where each iteration requires a matrix-vector multiplication involving \( A \) and \( A^{T} \) followed by a shrinkage/soft-threshold step. Other algorithms for the \( \ell_{1} \) minimization include alternating direction method of multipliers SALSA [1]; coordinate-wise descent method [39]; projected gradient method [17]; reduced-space algorithm [10]; active set methods ([11], [26], [33]). We refer to...
papers ([10], [26]) for recent advances in this area.

Another class of methods to deal with \(\ell_0\) optimization problem are greedy methods such as matching pursuit [29], orthogonal matching pursuit (OMP) [32], CoSaMP [31] and single best replacement [35]. These greedy methods update the support of \(x\) by exploiting the residual and then solve a least-squares problem on the support at each iteration. These type of algorithms are widely used in practice due to their relatively fast computational speed. Under a sufficient incoherence assumption, Tropp [38] proved that OMP recovers the sparsest representation of the input signal. We refer to a survey of greedy algorithms [37] for recent advances in this area.

The convexity of the \(\ell_0\) penalty allows designing fast and global convergent algorithms. However, it has a drawback which requires more restrictive conditions on the matrix \(A\) to establish the equivalent and unique global solution to model (3). In addition, it tends to produce biased estimated for larger coefficients and lacks oracle property [13,14]. To overcome the drawback produced by the \(\ell_1\) penalty, many continuous nonconvex approximations have been proposed to approach the \(\ell_0\)-norm. Among the variety of \(\ell_0\)-like continuous penalties, one can find the nonnegative garrote [6], log-sum penalty [7] and capped-\(\ell_1\) [42]. Fan and Li [16] defined necessary conditions to obtain a 'good' penalty function (unbiasedness, continuity in data and sparsity) and proposed the smoothly clipped absolute deviation (SCAD) penalty. Fan and Li [16] discussed a variable splitting and penalty decomposition minimization technique for (2), along with other approximations of the \(\ell_0\)-norm. Lv and Fan [27] used concave function on \(\mathcal{R}_+\) to approximate the \(\ell_0\) norm and proved a nonasymptotic nearly oracle property of the resultant estimator. Zhang [43] introduced the notion of sparse convexity to compare penalties and proposed the minimax concave penalty (MCP). Fourcat and Lai [19] investigated the use of \(\ell_p\)-norm and an interesting work on the equivalence between minimal \(\ell_0\) and \(\ell_p(0 < p < 1)\) solutions of linear systems of equations (equality and inequality) was given in [20]. A general iterative shrinkage thresholding algorithm in [21] was proposed and illustrated on the SCAD, MCP and capped \(\ell_1\) penalties. They also proved that the algorithm converged to a critical point. Fornasier and Ward [18] proposed an iterative thresholding algorithm for minimizing (2) where the \(\ell_0\) norm is replaced by a reasonable sparsity-promoting relaxing and convergence to a local minimizer is established. In [44], zhang et al. proved that the transformed \(\ell_1\) penalty can recover the exact solution of constrained \(\ell_0\) norm with the null space property (NSP) and the restricted isotropy property (RIP).

They also presented two difference of convex algorithms for the transformed \(\ell_1\) optimization which guarantee the convergence to a stationary point. Recently, Jiao et al. [24] developed a unified primal dual active set algorithm for a class of nonconvex sparsity penalties which includes \(\ell_0\), \(\ell_p\), SCAD, capped \(\ell_1\) and MCP. Under the restricted isotropy property, they proved the proposed algorithms converges to the underlying regression target. We refer to papers ([3], [4], [9], [24], [28], [34], [36], [45]) for recent advances in this area.

B. OUR WORK

In this paper, we propose a second-order algorithm to solve problem (1). The main contributions of our paper can be summarized as follows.

1. We provide two optimal property of problem (1).
2. We introduce an active set identification technique. An attractive advantage of the identification technique is that it can accurately identify the zero components in a neighbourhood of a stationary point of problem (1).
3. By means of the identification technique, we propose a second-order algorithm for solving (1.1). To accelerate the convergence of the algorithm, the nonmonotone line search [22] is used. We prove that every accumulation point of the sequence generated by the new algorithm is a stationary point of problem (1).
4. We do comprehensive numerical experiments to show the efficiency of the proposed algorithm. Numerical results demonstrate that the proposed algorithm is competitive with several known methods.

The remainder of the paper is organized as follows. Two optimal properties related to (1) are given in Section II. The active set estimate is presented in Section III. In Section IV, we propose the algorithm and establish the global convergence of the algorithm. We report some numerical results in Section V and make some conclusions in section VI.
The set $\mathcal{G}(\bar{x})$ can be subdivided into the following two sets

$$\mathcal{G}_+(\bar{x}) = \{i \in \mathcal{G}(\bar{x}) : |g_i(\bar{x})| < \lambda\}$$

and

$$\mathcal{G}_-(\bar{x}) = \{i \in \mathcal{G}(\bar{x}) : |g_i(\bar{x})| = \lambda\}.$$ 

The support $\mathcal{T}(\bar{x})$ can be subdivided into the following three sets

$$\mathcal{T}_1(\bar{x}) = \{i : 0 < |\bar{x}_i| < \lambda\theta\}, \quad \mathcal{T}_2(\bar{x}) = \{i : |\bar{x}_i| > \lambda\theta\}$$

and

$$\mathcal{T}_3(\bar{x}) = \{i : |\bar{x}_i| = \lambda\theta\}.$$ 

The following property shows that if $\bar{x}$ is a local minimizer of problem (2), then the columns of $A_{\mathcal{T}(\bar{x})}$ are linearly independent. Namely, the sparsity of the local minimizer is at most $\text{rank}(A)$. 

**Theorem 1:** Suppose that $\bar{x}$ is a local minimizer of problem (2), then $A_{\mathcal{T}(\bar{x})}$ is of full column rank, i.e., columns of $A_{\mathcal{T}(\bar{x})}$ are linearly independent.

**Proof:** Since $\bar{x}$ is a local minimizer of problem (2), there exists a neighbourhood $N(\bar{x}, r)$ of $\bar{x}$ such that

$$\varphi(x) \geq \varphi(\bar{x}), \quad \forall x \in N(\bar{x}, r).$$

Assume that columns of $A_{\mathcal{T}(\bar{x})}$ are not linearly independent. Then there exists a vector $0 \neq p = (p_{\mathcal{T}(\bar{x})}, 0)$ such that

$$A_{\mathcal{T}(\bar{x})}p_{\mathcal{T}(\bar{x})} = 0.$$ 

Consequently, we have

$$Ap = 0 \quad \text{and} \quad \mathcal{T}(p) \subseteq \mathcal{T}(\bar{x}).$$

Further, we can scale $p$ such that

$$\|p\| < \min\{r, |\bar{x}_i|, i \in \mathcal{T}(\bar{x})\}. \quad (5)$$

Let $\xi^1 = \bar{x} + p$ and $\xi^2 = \bar{x} - p$. By (5) and the definition of $p$, we have

$$\text{sgn}(\bar{x}_i) = \text{sgn}(\xi^1_i) = \text{sgn}(\xi^2_i),$$

for any index $i$ which means that

$$\frac{1}{2} |\xi^1_i| + \frac{1}{2} |\xi^2_i| = \frac{1}{2} |\xi^1_i| + |\xi^2_i|.\quad (6)$$

Since $r_i(t)$ is strictly concave for $t > 0$, we have

$$\frac{1}{2} r_i(|\xi^1_i|) + \frac{1}{2} r_i(|\xi^2_i|) < r_i(\frac{|\xi^1_i| + |\xi^2_i|}{2}) = r_i(\bar{x}_i).$$

Thus, we have

$$\min(\varphi(\xi^1), \varphi(\xi^2)) \leq \frac{1}{2} \varphi(\xi^1) + \frac{1}{2} \varphi(\xi^2)$$

$$= \frac{1}{4} \|A\xi^1 - b\|^2 + \frac{1}{4} \|A\xi^2 - b\|^2$$

$$+ \sum_{i=1}^n \left(\frac{1}{2} r_i(\xi^1_i) + \frac{1}{2} r_i(\xi^2_i)\right)$$

$$= \frac{1}{2} \|A\xi^1 - b\|^2 + \sum_{i=1}^n \left(\frac{1}{2} r_i(\xi^1_i) + \frac{1}{2} r_i(\xi^2_i)\right)$$

$$= \frac{1}{2} \|A\xi^1 - b\|^2 + \sum_{i=1}^n r_i(\bar{x}_i)$$

$$= \varphi(\bar{x}).$$

which contradicts the assumption that $\bar{x}$ is a local minimizer of problem (2). The proof is completed.

Chen et al. [8] gave an affine-scaled second-order sufficient condition for local minimizers of nonconvex penalties which include the MCP penalty. However, it is difficult to verify the condition. Huang et al. [24] gave a second-order sufficient condition for a stationary point to be a local minimizer point of (2). Below we provide two sufficient conditions for a stationary point to be a local minimizer point of (1).

**Condition (6) is an extension of the result [24].**

**Theorem 2:** Suppose that $f$ is twice continuously differentiable and $\bar{x}$ is a stationary point of problem (1). If

$$\hat{\lambda}_{\min} > \frac{1}{\theta} \quad \text{and} \quad ||g_{\mathcal{G}(\bar{x})}(\bar{x})\|_{\infty} < \lambda \quad (6)$$

or $\lambda_{\min} > \frac{1}{\theta}$ where $\hat{\lambda}_{\min}$ denotes the minimizer eigenvalue of the matrix $\nabla^2 f(\bar{x})_{\mathcal{T}(\bar{x}), \mathcal{T}(\bar{x})}$ and $\lambda_{\min}$ denotes the minimizer eigenvalue of the matrix $\nabla^2 f(\bar{x})$, then $\bar{x}$ is a local minimizer point of problem (1).

**Proof:** Suppose condition (6) holds. If $\mathcal{T}(\bar{x}) = \emptyset$, the conclusion is clearly. Suppose that $\mathcal{T}(\bar{x}) \neq \emptyset$. We shall prove that $\varphi(\bar{x} + d) \geq \varphi(\bar{x})$ for any sufficiently small $0 \neq d \in \mathbb{R}^n$.

We shall consider two different cases. Note that

$$|\bar{x}_i + d_i| - |\bar{x}_i| = \text{sgn}(\bar{x}_i)d_i, \quad \text{as} \quad ||d|| \to 0 \quad \text{and} \quad \bar{x}_i \neq 0. \quad (7)$$

Case (1) $d_{\mathcal{G}(\bar{x})} \neq 0$. For any sufficiently small vector $0 \neq d \in \mathbb{R}^n$, we have

$$\varphi(\bar{x} + d) - \varphi(\bar{x})$$

$$= f(\bar{x} + d) - f(\bar{x}) + \sum_{i=1}^n (r_i(\bar{x}_i + d_i) - r_i(\bar{x}_i))$$

$$= g(\bar{x})^T d + \sum_{|\bar{x}_i| \leq \theta \lambda} (r_i(\bar{x}_i + d_i) - r_i(\bar{x}_i)) + o(||d||)$$

$$= \sum_{\bar{x}_i = 0} (g_i(\bar{x})^T d + \lambda d_i)$$

$$+ \sum_{0 < |\bar{x}_i| \leq \theta \lambda} (g_i(\bar{x}) + \nabla r_i(\bar{x})) d_i + o(||d||)$$

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/.
\[ \sum_{i=0}^{\lambda} (\lambda - \lambda_i |d_i|) > 0. \]

Case (2) \( d_{g(x)} = 0 \). For any sufficiently small vector \( 0 \neq d \in \mathbb{R}^n \), we have

\[ \varphi(\bar{x} + d) - \varphi(\bar{x}) = f(\bar{x} + d) - f(\bar{x}) + \sum_{i=1}^{n} (r_i(\bar{x}_i + d_i) - r_i(\bar{x}_i)) \]
\[ = g(\bar{x})^T d + \frac{1}{2} d^T \nabla^2 f(\bar{x}) d + o(||d||^2) \]
\[ + \sum_{|\bar{x}_i| \leq \lambda} (\lambda |\bar{x}_i| + |d_i| - \lambda |\bar{x}_i| + \frac{\bar{x}_i^2 - (\bar{x}_i + d_i)^2}{2\theta}) \]
\[ = \sum_{\bar{x}_i = 0} (g(\bar{x}) d_i + \lambda |d_i|) \]
\[ + \sum_{0 < |\bar{x}_i| \leq \lambda} (g(\bar{x}) + \lambda sgn(\bar{x}_i) - \frac{\bar{x}_i}{\theta}) d_i \]
\[ + \frac{1}{2} d^T \nabla^2 f(\bar{x}) d + \sum_{|\bar{x}_i| \leq \lambda} - \frac{d_i^2}{2\theta} + o(||d||^2) \]
\[ = \frac{1}{2} d^T \nabla^2 f(\bar{x}) d + \sum_{0 < |\bar{x}_i| \leq \lambda} - \frac{d_i^2}{2\theta} + o(||d||^2) \]
\[ \geq \frac{\lambda_{\min}}{2} ||d_{\mathcal{T}_2(\bar{x})}||^2 + o(||d||^2) \]
\[ + \sum_{0 < |\bar{x}_i| \leq \lambda} \left( \frac{\lambda_{\min}}{2} - \frac{1}{2\theta} \right) d_i^2 > 0 \]

where the fifth equality uses (4) and the fourth equality uses (7).

If \( \lambda_{\min} > \frac{1}{\theta} \), by (7) and (8) we have

\[ \varphi(\bar{x} + d) - \varphi(\bar{x}) \]
\[ = \sum_{\bar{x}_i = 0} (g(\bar{x}) d_i + \lambda |d_i|) + o(||d||^2) \]
\[ + \sum_{0 < |\bar{x}_i| \leq \lambda} (g(\bar{x}) + \lambda sgn(\bar{x}_i) - \frac{\bar{x}_i}{\theta}) d_i \]
\[ + \frac{1}{2} d^T \nabla^2 f(\bar{x}) d + \sum_{|\bar{x}_i| \leq \lambda} - \frac{d_i^2}{2\theta} \]
\[ \geq \sum_{\bar{x}_i = 0} (\lambda - g(\bar{x})) |d_i| + \frac{\lambda_{\min}}{2} ||d||^2 \]
\[ - \sum_{|\bar{x}_i| \leq \lambda} \frac{d_i^2}{2\theta} + o(||d||^2) \]
\[ \geq \frac{\lambda_{\min}}{2} ||d_{\mathcal{T}_2(\bar{x})}||^2 + o(||d||^2) \]
\[ + \sum_{|\bar{x}_i| \leq \lambda} \left( \frac{\lambda_{\min}}{2} - \frac{1}{2\theta} \right) d_i^2 > 0 \]

for any sufficiently small \( d \neq 0 \). The proof is completed. \( \blacksquare \)

**III. ACTIVE SET ESTIMATE**

In this section, we propose the active set estimate and give some properties of the active set estimate. We use the following sets \( \mathcal{Z}(x) \) and \( \mathcal{S}(x) \) to approximate \( \mathcal{G}(\bar{x}) \) and \( \mathcal{T}(\bar{x}) \), respectively.

\[ \mathcal{Z}(x) = \{ i : |x_i| \leq \lambda \theta \}, \]
\[ \mathcal{S}(x) = \{ i : |x_i| > \lambda \theta \}. \]

Furthermore, we partition \( \mathcal{Z}(x) \) into two parts. That is,

\[ \mathcal{Z}_1(x) = \{ i \in \mathcal{Z}(x) : |x_i - g_i(x)| \leq \lambda \}
\]
and

\[ \mathcal{Z}_2(x) = \{ i \in \mathcal{Z}(x) : |x_i - g_i(x)| > \lambda \}. \]

Below the theorem shows that when the point \( \bar{x} \) is sufficiently close to a stationary point \( \bar{x} \), the set \( \mathcal{Z}_1(x) \) is a subset of the set \( \mathcal{G}(\bar{x}) \). Furthermore, when \( \mathcal{G}_-(\bar{x}) = \emptyset \) holds, the set \( \mathcal{Z}_1(x) \) is actually equal to the set \( \mathcal{G}(\bar{x}) \).

**Theorem 3:** Assume that \( \bar{x} \) is a stationary point of problem (1) and \( g(x) \) is continuous on \( \mathbb{R}^n \). Then we have the following statement.

1. There exists a neighbourhood \( \mathcal{N}(\bar{x}) \) of \( \bar{x} \) such that \( \mathcal{G}_+(\bar{x}) \subseteq \mathcal{Z}_1(x) \subseteq \mathcal{G}(\bar{x}) \).

In particular, if \( \mathcal{G}_-(\bar{x}) = \emptyset \), then we have

\[ \mathcal{Z}_1(x) = \mathcal{G}(\bar{x}), \quad \forall x \in \mathcal{N}(\bar{x}). \]

2. There exists a neighbourhood \( \mathcal{N}(\bar{x}) \) of \( \bar{x} \) such that \( \mathcal{T}_2(\bar{x}) \subseteq \mathcal{S}(x) \subseteq \mathcal{T}_2(\bar{x}) \cup \mathcal{T}_3(\bar{x}), \quad \forall x \in \mathcal{N}(\bar{x}). \)

3. There exists a neighbourhood \( \mathcal{N}(\bar{x}) \) of \( \bar{x} \) such that \( \mathcal{T}_1(\bar{x}) \subseteq \mathcal{Z}_2(x) \subseteq \mathcal{G}_-(\bar{x}) \cup \mathcal{T}_1(\bar{x}) \cup T_3(\bar{x}), \quad \forall x \in \mathcal{N}(\bar{x}). \)

Proof: If \( \mathcal{G}_-(\bar{x}) = \emptyset \) holds at \( \bar{x} \), then the second statement in (1) follows from the first statement (1). Thus, it suffices to prove the statement (1). For each \( i \in \mathcal{G}_+(\bar{x}) \), we have

\[ \bar{x}_i = 0 \quad \text{and} \quad |g_i(\bar{x})| < \lambda. \]

By the continuity of \( g(x) \), we have

\[ |x_i| \leq \lambda \theta \quad \text{and} \quad |x_i - g_i(x)| \leq \lambda \]

as \( x \to \bar{x} \). Thus \( i \in \mathcal{Z}_1(x) \) as \( x \to \bar{x} \). For each \( i \in \mathcal{Z}_1(x) \), by the continuity of \( g(x) \), we have

\[ |\bar{x}_i| \leq \lambda \theta \quad \text{and} \quad |\bar{x}_i - g_i(\bar{x})| \leq \lambda. \]

(9)

We claim that it must hold \( \bar{x}_i = 0 \). In fact, by (4), we get

\[ |\bar{x}_i| = \lambda \theta \quad \text{and} \quad |\bar{x}_i - g_i(\bar{x})| = |(\frac{1}{\theta})\bar{x}_i + sgn(\bar{x}_i)\lambda| > \lambda \]

and

\[ 0 < |\bar{x}_i| < \lambda \theta \quad \text{and} \quad |\bar{x}_i - g_i(\bar{x})| = |(\frac{1}{\theta})\bar{x}_i + sgn(\bar{x}_i)\lambda| > \lambda \]
which each $i \in T_2(\bar{x})$, by the continuity of $g(x)$, we have

$$|x_i| > \lambda \theta \ and \ |x_i - g_i(x)| \to |\bar{x}_i| > \lambda \theta$$

as $x \to \bar{x}$. Thus $i \in S(x)$. On the other hand, for each $i \in S(x)$, by (4) we have

$$|\bar{x}_i| \geq \lambda \theta \ and \ g_i(\bar{x}) = 0.$$

Thus $i \in T_2(\bar{x}) \cup T_1(\bar{x})$. Then we get the statement (2).

For each $i \in T_1(\bar{x})$, by (4) we have

$$\lambda < |\bar{x}_i - g_i(\bar{x})| = |(1 - \frac{1}{\theta})\bar{x}_i + sgn(\bar{x}_i)\lambda| < \lambda \theta.$$ 

By the continuity of $g(x)$, we have $i \in Z_2(x)$ as $x \to \bar{x}$. On the other hand, for each $i \in Z_2(x)$, by the continuity of $g(x)$ we have

$$|\bar{x}_i| \leq \theta \lambda \ and \ \lambda \leq |\bar{x}_i - g_i(\bar{x})| \leq \lambda \theta.$$

Then we get $i \in G_-(\bar{x}) \cup T_1(\bar{x}) \cup T_2(\bar{x})$ as $x \to \bar{x}$. Then we get the statement (3). The proof is completed. ■

To get a better descent direction of the new algorithm, we partition $\mathbb{Z}_2(x)$ into $\mathbb{Z}_{21}(x)$ and $\mathbb{Z}_{22}(x)$, respectively,

$$\mathbb{Z}_{21}(x) = \{i \in \mathbb{Z}_2(x) : x_i - g_i(x) > \lambda\}$$

and

$$\mathbb{Z}_{22}(x) = \{i \in \mathbb{Z}_2(x) : x_i - g_i(x) < -\lambda\}.$$

Furthermore, we partition $\mathbb{Z}_{21}(x)$ and $\mathbb{Z}_{22}(x)$ into two parts, respectively,

$$\mathbb{Z}_{21,1}(x) = \{i \in \mathbb{Z}_{21}(x) : 0 < x_i \leq \lambda \theta\},$$

$$\mathbb{Z}_{21,f}(x) = \{i \in \mathbb{Z}_{21}(x) : -\lambda \theta \leq x_i \leq 0\},$$

$$\mathbb{Z}_{22}(x) = \{i \in \mathbb{Z}_{22}(x) : -\lambda \theta < x_i < 0\},$$

$$\mathbb{Z}_{22,f}(x) = \{i \in \mathbb{Z}_{22}(x) : 0 \leq x_i \leq \lambda \theta\}.$$ 

The following theorem shows that $\mathbb{Z}_{21,f} \cup \mathbb{Z}_{22,f}$ is a subset of the set $G_-(\bar{x})$ when $x$ is sufficiently close to $\bar{x}$.

**Theorem 4:** Assume that $\bar{x}$ is a stationary point of problem (1) and $g(x)$ is continuous on $\mathbb{R}^n$. Then there exists a neighborhood $\mathcal{N}(\bar{x})$ of $\bar{x}$ such that, $\forall x \in \mathcal{N}(\bar{x})$, the following statements hold,

$$\mathbb{Z}_{21,f}(x) \subset \{i : \bar{x}_i = 0, \ g_i(\bar{x}) = \lambda\},$$

$$\mathbb{Z}_{22,f}(x) \subset \{i : \bar{x}_i = 0, \ g_i(\bar{x}) = -\lambda\},$$

$$\mathbb{Z}_{21,f}(x) \cup \mathbb{Z}_{22,f}(x) \subseteq G_-(\bar{x}).$$

**Proof:** It is easy to see that (10)-(11) give (12). In the following, we will only prove (10) since by a similar analysis, we can show (11). For each $i \in \mathbb{Z}_{21,f}(x)$, by the continuity of $g(x)$, we have

$$-\lambda \theta \leq \bar{x}_i \leq 0 \ and \ \bar{x}_i - g_i(\bar{x}) \geq \lambda$$

(13)

Then we must have $\bar{x}_i = 0$ and $g_i(\bar{x}) = \lambda$. In fact,

$$\bar{x}_i = -\lambda \theta \ and \ \bar{x}_i - g_i(\bar{x}) = -\lambda \theta$$

and

$$-\lambda \theta \leq \bar{x}_i < 0 \ and \ \bar{x}_i - g_i(\bar{x}) = (1 - \frac{1}{\theta})\bar{x}_i + sgn(\bar{x}_i)\lambda < 0.$$ 

which contradicts (9). Then we get the statement (1).
Then it follows that
\[ c_1 \| g_{2x_1} \| \leq \| d_{2x_1} \| \leq c_2 \| g_{2x_1} \|. \] (20)

For each \( i \in I_2^k \), we have
\[ 0 \leq x_i^k \leq \lambda \theta, \quad g_i(x^k) > x_i^k + \lambda > \frac{1}{\theta} x_i^k + \lambda, \quad i \in Z_{21f}(x^k) \]
and
\[ -\lambda \theta \leq x_i^k \leq 0, \quad g_i(x^k) < x_i^k - \lambda < \frac{1}{\theta} x_i^k - \lambda, \quad i \in Z_{22f}(x^k). \]

Then we have
\[
\begin{align*}
|d_i^k| &= |g_i(x^k) - \frac{1}{\theta} x_i^k - \text{sgn}(g_i(x^k) - \frac{1}{\theta} x_i^k)\lambda| \\
&= |g_i(x^k) - \frac{1}{\theta} x_i^k - \lambda| > 0, \quad i \in I_2^k.
\end{align*}
\] (21)

From Theorem 3, we know that \( Z_i^k \) approximates the set \( \mathcal{G}(x) \) when \( x^k \rightarrow \bar{x} \). Thus, it is a better choice to fix those variables with indices in \( Z_i^k \) to 0. Thus, we set \( d_i^k = -x_i^k \), \( \forall i \in Z_i^k \). We will get \( x_i^{k+1} = 0 \) for all \( i \in Z_i^k \) if the unit steplength is accepted (see Algorithm 1). Below the theorem establishes that \( d^k = 0 \) if and only if \( x^k \) is a stationary point of problem (1).

**Theorem 6:** Assume that \( d^k \) is determined by (16)-(19). Then \( d^k = 0 \) if and only if \( x^k \) is a stationary point of problem (1).

Proof: Suppose that \( d^k = 0 \). For each \( i \in Z_1(x^k) \) and \( d_i^k = -x_i^k \), we have
\[ x_i^k = 0 \quad \text{and} \quad |g_i(x^k)| \leq \lambda. \]

For each \( i \in S(x^k) \), by (20), we get
\[ |x_i^k| > \lambda \theta \quad \text{and} \quad g_i(x^k) = 0, \quad i \in S(x^k). \]

By (21) and the assumption \( d^k = 0 \), we note that \( Z_i^k = 0 \). For each \( i \in Z_i^k \), by the definition of \( d_i^k \) and the assumption \( d^k = 0 \), we get
\[ 0 < |x_i^k| \leq \lambda \theta \quad \text{and} \quad g_i(x^k) + \text{sgn}(x_i^k)\lambda - \frac{1}{\theta} x_i^k = 0. \]

On the contrary, suppose that \( x^k \) is a stationary point of problem (1). By Theorem 5, we get \( Z_i^k = 0 \). For each \( i \in S(x^k) \), by (4) we get \( g_i(x^k) = 0 \). By (20), we get \( d_i^k = 0 \), for all \( i \in S(x^k) \). For each \( i \in Z_1(x^k) \), we must have \( d_i^k = -x_i^k = 0 \). In fact, if \( 0 < |x_i^k| \leq \lambda \theta \), by (4) we get
\[ |x_i^k - g_i(x^k)| = |(1 - \frac{1}{\theta}) x_i^k + \text{sgn}(x_i^k)\lambda| > \lambda \]
which contradicts the definition of \( Z_1(x^k) \). For each \( i \in Z_i^k \), by (4) we get
\[ 0 < |x_i^k| \leq \lambda \theta \quad \text{and} \quad d_i^k = -g_i(x^k) + \text{sgn}(x_i^k)\lambda - \frac{1}{\theta} x_i^k = 0. \]

The proof is completed.

The following theorem together with Theorem 6 demonstrates that \( d^k \) is a descent direction of \( \varphi \) at \( x_k \) if \( x_k \) is not a stationary point of problem (1). Consequently, it guarantees that Algorithm 1 is well-defined.

**Theorem 7:** Suppose that \( x^k \) is not a stationary point of problem (1) and \( d^k \) determined by (16)-(19). Then we have
\[ \lim_{\beta \to 0^+} \varphi(x^k + \beta d^k) - \varphi(x^k) < 0. \]

Proof: By the convexity of \( \|x\|_1 \), we get for any \( \beta \in [0, 1] \)
\[ |x_i^k + \beta d_i^k| - |x_i^k| \leq \beta(|x_i^k + d_i^k| - |x_i^k|). \] (22)

For each \( i \in I_2^k \), by (21), we get
\[
\begin{align*}
g_i(x^k)d_i^k + \lambda(|x_i^k + d_i^k| - |x_i^k|) - \frac{1}{\theta} d_i^k x_i^k \\
&\leq g_i(x^k)d_i^k + \lambda|d_i^k| - \frac{1}{\theta} d_i^k x_i^k \\
&= -(g_i(x^k) - \frac{1}{\theta} x_i^k)^2 + 2g_i(x^k) - \frac{1}{\theta} x_i^k|\lambda - (\lambda)^2 \\
&= -(|g_i(x^k) - \frac{1}{\theta} x_i^k| - \lambda)^2 \\
&= -(d_i^k)^2. \end{align*}
\] (23)

For each \( i \in I_1^k \), without loss of generality, we assume \( 0 \leq x_i^k \leq \lambda \theta \). By the definition of \( Z_i^k \), we have
\[ x_i^k \leq g_i(x^k) + \lambda = g_i(x^k) + \text{sgn}(x_i^k)\lambda. \]

Thus,
\[ (d_i^k)^2 \leq (g_i(x^k) + \text{sgn}(x_i^k)\lambda)x_i^k. \]

Then we have
\[ -g_i(x^k) + \text{sgn}(x_i^k)\lambda x_i^k \leq -(d_i^k)^2, \quad i \in Z_i^k. \]

For each \( i \in Z_1^k \) and \( d_i^k = -x_i^k \), by the last inequality we have
\[
\begin{align*}
g_i(x^k)d_i^k + \lambda(|x_i^k + d_i^k| - |x_i^k|) - \frac{1}{\theta} d_i^k x_i^k \\
&= g_i(x^k)d_i^k - \lambda|x_i^k| + \frac{1}{\theta}(d_i^k)^2 \\
&= -(g_i(x^k) + \lambda \text{sgn}(x_i^k))x_i^k + \frac{1}{\theta}(d_i^k)^2 \\
&\leq -(1 - \frac{1}{\theta})(d_i^k)^2. \end{align*}
\] (24)

By the definition of \( d^k \), we can get
\[ \varphi(x^k + \beta d^k) - \varphi(x^k) \]
\[ = f(x^k + \beta d^k) - f(x^k) + \sum_{i=1}^{n} (r_i(x_i^k + \beta d_i^k) - r_i(x_i^k)) \\
= \beta g(x^k)^T d^k + \sum_{|x_i^k| \leq \theta} (\lambda |x_i^k + \beta d_i^k| - \lambda |x_i^k| + (x_i^k)^2 - (x_i^k + \beta d_i^k)^2) \end{align*}
\[ + \frac{2\theta}{\beta} (x_i^k)^2 - (x_i^k + \beta d_i^k)^2 + o(\beta) \]
\[ = \beta g(x^k)^T d^k + o(\beta) \]
\[ + \sum_{|x_i^k| > \theta} (\lambda |x_i^k + \beta d_i^k| - \lambda |x_i^k| - \frac{1}{\theta} d_i^k x_i^k) \]
\[ \leq \beta g(x^k)^T d^k + \beta \sum_{Z_1^k \cup Z_2^k \cup Z_3^k} (\lambda |x_i^k + d_i^k| - \lambda |x_i^k| - \frac{1}{\theta} d_i^k x_i^k) \]
Assume that been successfully incorporated into a variety of optimization produced by Grippo, Lampariello, and Lucidi (see [23]). It has

\[ \sum_{i} (\lambda|x|^2 + \beta d_k^i) - |\lambda|x|^2 - \frac{\beta}{\theta} d_k^i x_i^k) + o(\beta) \]

\[ \leq \beta \sum_{x_k^i} (g(x_k^i) d_k^i + \lambda|x|^2 + \beta d_k^i) - |\lambda|x|^2 - \frac{\beta}{\theta} d_k^i x_i^k) \]

\[ + \sum_{x_k^i} \beta(g(x_k^i) - \frac{1}{\theta} d_k^i x_i^k) + \lambda \beta \text{sgn}(x_k^i) d_k^i x_i^k) \]

where \( \beta \) is generated by Algorithm 1. Then

\[ c||d_k^i||^2 + o(\beta) \]

\[ \leq -c||d_k^i||^2 - \beta \frac{1}{\theta} ||d_k^i||^2 - \beta ||d_k^i||^2 \]

\[ + \beta \sum_{x_k^i} (g(x_k^i) - \frac{1}{\theta} d_k^i x_i^k) + \lambda \text{sgn}(x_k^i) d_k^i + o(\beta) \]

\[ = -c||d_k^i||^2 - \beta \frac{1}{\theta} ||d_k^i||^2 - \beta ||d_k^i||^2 \]

\[ = -c||d_k^i||^2 - \beta \frac{1}{\theta} ||d_k^i||^2 - \beta ||d_k^i||^2 \]

\[ \leq 0 \]

\[ |\lambda|^2 \leq \lambda \theta \]

\[ g_i(x^*) + \text{sgn}(x_i^*) \lambda - \frac{1}{\theta} x_i^* = 0 \]

\[ \text{for each } i \in I_2, \text{we have} \]

\[ -\lambda \theta \leq x_i^* \leq 0 \text{ and } x_i^* - g_i(x^*) \geq \lambda \]

or

\[ 0 \leq x_i^* \leq \lambda \theta \text{ and } x_i^* - g_i(x^*) \leq -\lambda \]

By \( d_k^i = -(g_i(x_k^i) - \text{sgn}(g_i(x_k^i)) \lambda - \frac{1}{\theta} x_i^k) \to 0 \), we have

\[ \beta \leq x_i^k \leq -\lambda \theta \]

\[ \frac{1}{\theta} x_i^k = g_i(x^*) + \lambda \leq x_i^* \]

and

\[ 0 \leq x_i^* \leq \lambda \theta + \frac{1}{\theta} x_i^k = g_i(x^*) - \lambda \geq x_i^* \]

Thus we have

\[ x_i^* = 0 \text{ and } |g_i(x^*)| = \lambda \]

Thus, for each \( i \in I, \) by (20) we get

\[ |x_i^*| \geq \lambda \theta \text{ and } g_i(x^*) = 0. \]

Thus, \( x^* \) is a stationary point of problem (1). The proof is completed.

\[ \text{Proof: Taking into account that the number of distinct sets } Z_k^1, I_k^1, I_k^2, I_k \text{ is finite, there exists a subsequence (without loss of generality, we label again } \{x_k\} \text{ such that } \text{index sets } Z_k, I_k, I_k^1 \text{ and } I_k^2 \text{ are constant and hence we can write} \]

\[ Z_k^1 = Z_1, I_k^1 = I_1, I_k^2 = I_2 \text{ and } I_k = I \]

\[ \text{for each } i \in I_2, \text{we have} \]

\[ -\lambda \theta \leq x_i^* \leq 0 \text{ and } x_i^* - g_i(x^*) \geq \lambda \]

or

\[ 0 \leq x_i^* \leq \lambda \theta \text{ and } x_i^* - g_i(x^*) \leq -\lambda \]

By \( d_k^i = -(g_i(x_k^i) - \text{sgn}(g_i(x_k^i)) \lambda - \frac{1}{\theta} x_i^k) \to 0 \), we have

\[ \beta \leq x_i^k \leq -\lambda \theta \]

\[ \frac{1}{\theta} x_i^k = g_i(x^*) + \lambda \leq x_i^* \]

and

\[ 0 \leq x_i^* \leq \lambda \theta + \frac{1}{\theta} x_i^k = g_i(x^*) - \lambda \geq x_i^* \]

Thus we have

\[ x_i^* = 0 \text{ and } |g_i(x^*)| = \lambda \]

Thus, for each \( i \in I, \) by (20) we get

\[ |x_i^*| \geq \lambda \theta \text{ and } g_i(x^*) = 0. \]

Thus, \( x^* \) is a stationary point of problem (1). The proof is completed.

\[ \text{B. THE ALGORITHM AND ITS GLOBAL CONVERGENCE} \]

Now, we state the steps of the new algorithm for solving (1) as follows.

\[ \text{Algorithm 1: (The new algorithm)} \]

\[ \text{Step 0. Initialize an initial guess } x^0 \in \mathbb{R}^n \text{ and constants } M > 0, \eta > 0 \text{ and } \delta \in (0, 1). \text{ Set } k = 0. \]

\[ \text{Step 1. If } x^k \text{ satisfies the stopping condition, then stop. Otherwise, go to Step 2.} \]

\[ \text{Step 2. Compute } d_k \text{ by (16) and (17).} \]

\[ \text{Step 3. Determine } \beta_k := \max\{\eta^j, j = 0, 1, \cdots\} \text{ satisfying} \]

\[ \varphi(x^k + \beta_k d_k) \leq \varphi_{\text{max}} - \delta(\beta_k ||d_k||^2)^2, \]  

\[ \text{where } \varphi_{\text{max}} = \max\{\varphi(x^{k-j}) : 0 \leq j \leq \min(k, M - 1)\}. \]

\[ \text{Step 4. Set } x^{k+1} := x^k + \beta_k d_k. \]

\[ \text{Step 5. Set } k := k + 1 \text{ and go to Step 1.} \]

The nonmonotone line search condition (25) was introduced by Grippo, Lampariello, and Lucidi (see [23]). It has been successfully incorporated into a variety of optimization algorithms. Now, we need the following two lemmas to establish the convergence of Algorithm 1. Using a similar proof as the one in [22], we get Lemma 1.

\[ \text{Lemma 1: Assume that } f \text{ satisfies Assumption 4.1 and } \{x_k\} \text{ is} \]

\[ \text{generated by Algorithm 1. Then} \]

\[ \lim_{k \to \infty} \beta_k d_k = 0. \]  

The following theorem shows that if \( x_k \to x^* \) and \( d_k \to 0 \), then \( x^* \) is a stationary point of problem (1).

\[ \text{Lemma 2: Assume that } x_k \to x^* \text{ and } d_k \to 0, \text{ where } d_k \text{ is determined by (16)-(19). Then } x^* \text{ is a stationary point of problem (1).} \]
for any \( k \in \mathcal{K} \). We consider two different cases. Case (1) \( \lim_{k \to \infty} \beta_k \neq 0 \). By (26), we obtain

\[
\lim_{k \to \infty} d^k = 0.
\]

By Lemma 2, we get the conclusion. Case (2) \( \lim_{k \to \infty} \beta_k = 0 \).

By the line search condition (25), we get

\[
\phi(x^k + \frac{\beta_k}{\eta} d^k) - \phi(x^k) > -\delta \left( \frac{\beta_k}{\eta} \| d^k \| \right)^2
\]

for all \( k \geq 1 \). By the mean-value theorem, we get for sufficiently large \( k \in \mathcal{K} \) that

\[
\phi(x^k + \frac{\beta_k}{\eta} d^k) - \phi(x^k) = \frac{\beta_k}{\eta} \left( g(x^k + \theta \frac{\beta_k}{\eta} d^k) - g(x^k) \right)^T d^k + \sum_{i \in \mathcal{I}_k^1} \frac{\beta_k}{\eta} g_i(x^k) d^k_i + \frac{\beta_k}{\eta} \| x^k_i + \frac{\beta_k}{\eta} d^k_i \| - \| x^k_i \| - \frac{\beta_k}{\eta} \| d^k_i \| \| x^k_i \|
\]

\[
\leq \theta_k L \left( \frac{\beta_k}{\eta} \right)^2 \| d^k \|^2 - c_1 \frac{\beta_k}{\eta} \| d^k \|^2
\]

where \( \theta_k \in (0, 1) \), the first inequality uses (15), the convexity of \( \| x \|_1 \) and the last equality use (23) and (24). Substituting (27) into the last inequality, we have

\[
-c_1 \| d^k \|_1^2 - (1 - \frac{1}{\theta}) \| d^k \|_2^2 - \sum_{i \in \mathcal{I}_k^1} \| g_i(x^k) \| d^k_i + \lambda \| x^k_i + d^k_i \| - \| x^k_i \| - \frac{\beta_k}{\eta} \| d^k_i \| \| x^k_i \|
\]

and the solution (or an approximate solution) \( x_{k+1} \) as the initial estimate of the solution to the next problem. We set \( \lambda_0 = 0.5 \| A^T b \|_\infty \) and \( \lambda_N = \lambda = 10^{-10} \lambda_0 \). The interval \( [\lambda_N, \lambda_0] \) is divided into \( N \) equally distributed subintervals in logarithmic scale. The sequence \( \{ \lambda_k \} \) are chosen to be the endpoints of the subintervals. We set \( N = 30 \) in all tested algorithms. In our implementation, at the end of iteration \( k \), the next parameter \( \lambda_k \) is set to a value smaller than \( \lambda_{k-1} \) if the point \( x^k \) satisfies the condition

\[
\text{max}(|\phi_{\lambda_1}(x^k)|, |\phi_{\lambda_{k-1}}(x^k-1)|) \leq \epsilon_f
\]

where \( \epsilon_f = 0.01 \). We implemented the CG algorithm in [23] to solve (17) with parameters \( c = \sigma = 10^{-8} \). In many applications, it is not necessary and wasteful to obtain an

V. NUMERICAL EXPERIMENTS

In this section, computational experiments for testing the performance of the proposed algorithm on the compressed sensing problems (2) are presented. We compare the proposed algorithm with the following three existing solvers: GIST [21], PDASC [24] and MSCR [42]. The MATLAB code (package Unified-PDASC) is available at http://www0.cs.ucl.ac.uk/staff/b.jin/software/updasc.zip. We note that package Unified-PDASC contains the codes GIST [21] and MSCR [42]. "MCP" form of the three codes are implemented with default parameters. All codes are written in MATLAB R2013a and all tests described in this section were performed on a PC with Intel I5-3230 2.6GHz CPU processor and 16GB RAM memory with a Windows operating system.

In Subsection VI-A, parameter settings and data generated are presented. In Subsection VI-B, the effect of different concave parameters \( \theta \) on the proposed algorithm is presented. We compare our method GIST [21], PDASC [24] and MSCR [42] by solving middle-scale problem (2) in Subsection VI-C and by solving large-scale problem (2) in Subsection VI-D. The results in Subsections VI-C and VI-D show that our method outperforms GIST [21], PDASC [24] and MSCR [42] both in solution quality and in running time for all most tested problems.

A. PARAMETERS SETTING AND DATA GENERATION

We now briefly describe the implementation details of Algorithm 1. For convenience, we abbreviate Algorithm 1 as AMCP. We implemented AMCP with the following parameters \( M = 10 \), \( \delta = 10^{-2} \) and \( \eta = 0.6 \). For all the tests, the computation is initialized with zero vector. To obtain a good initial guess, we embed AMCP in a continuation procedure. Specifically, instead of solving problem (1) directly from scratch, we solve a sequence of problems \( x_{N+1}^k := \arg \min_{x \in \mathbb{R}^n} \| \phi \|_1 \) \( k = 0, \ldots, N \), where \( \lambda_0 > \lambda_1 > \cdots > \lambda_N \) is a positive integer and

\[
r_{i}^{k+1}(x_i) = \begin{cases} 
\lambda_k |x_i| - \frac{x_i^2}{2 \lambda_k}, & \text{if } |x_i| < \theta \lambda_k, \\
\frac{\lambda_k}{2}, & \text{if } |x_i| \geq \theta \lambda_k,
\end{cases}
\]

and use the solution (or an approximate solution) \( x_{N+1}^k \) as the initial estimate of the solution to the next problem. This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/.
exact solution at each iteration of (17) when \( |T^k| \) is large. Thus we use only an approximate solution of system \((17)\) and also terminate the CG algorithm if one of the following is satisfied:

1. \( r^j \leq \max\{10^{-1}r^0, 10^{-12}\}; \)
2. \( \|d^j\| \geq \max\{0.05, \min\{10^3, 10\|x^k - x^{k-1}\|\}\}; \)
3. the maximum number of CG iterations exceeds 10.

In the above, \( d^j \) denotes the direction of the \( j \)th CG iteration and \( r^j \) denotes the residual of the \( j \)th CG iteration. It is easy to see that Theorem 2.2 (c) of [23] ensures that there exist positive constants \( c_1 \) and \( c_2 \) such that the direction \( d^j \) produced by the CG algorithm satisfies conditions (18)-(20).

In the following subsections, we demonstrate the viability of our approach and focus on the problems from compressed sensing. Namely, we consider the problem (2), where the goal is to reconstruct a length-\( n \) sparse signal from \( m \) observations, where \( m < n \). Given the dimension of the signal \( n \) and the number of observations \( m \), we generated a random matrix \( A \) as follows. The type of matrix \( A \) was chosen from the following types:

1. DCT matrix.
2. Orthogonalized Gaussian matrix whose rows are orthogonalized using a QR.
3. Sparse normally distributed random matrix.

Types (1) and (2) of matrix \( A \) were generated by the code [\( A, b, x, xsn, picks \]= getDatam\( n, k, Ameth, xmeth, varargin \) [40]. For each test problem, we stop all tested methods if the condition

\[
\|Ax^k - b\| \leq \epsilon
\]

where \( \epsilon = \|e\| \) and \( e \) is the noise of the data \( b \).

**B. EFFECT OF THE CONCAVE PARAMETER \( \theta \)**

In the experiment, we examine the sensitivity of the algorithm with respect to the concavity parameter \( \theta \). The type of measure matrix \( A \) is DCT matrix. The observation \( b \) is generated by \( b = Ax^* + e \), where \( e \) is drawn according to the Gaussian distribution with zero mean and variance \( 10^{-2} \).

To generate the true signal \( x^* \), we first generated the support \( A^* = \{i | x_i^* \neq 0\} \) by randomly selecting \( T \) indices between 1 and \( n \) and then assigned the sign of a normally distributed random variable to \( x_i^* \) for each \( i \) in \( A^* \). The performance is evaluated in terms of CPU time in seconds (time) and the error (error= \( \|x^k - x^*\|_\infty \)), which are computed as the average of 10 independent realizations of the problem. Table 1 lists the average CPU time and the average error for each problem. From Table 1, we see that the CPU time is fairly robust with respect to the concavity of parameter \( \theta \). Further, the average error varies little with the parameter \( \theta \), indicating the robustness of AMCP. In all the following tests, we set the parameter \( \theta = 2.7 \).

**C. RECONSTRUCTION FROM NOISE OBSERVATION FOR MIDDLE-SCALE PROBLEMS**

In the experiment, the type of the measure matrix \( A \) is orthogonalized Gaussian matrix whose rows are orthogonalized using a QR. The true signal \( x^* \) is a random sparse vector with \( T \)-sparse signal with a support \( A^* \). The dynamical range \( R \) of the true signal \( x^* \) is defined by \( R = \frac{M}{m} \) with \( M = \max\{|x_i^*| : i \in A^*\} \) and \( m = \min\{|x_i^*| : i \in A^*\} \). The \( T \) nonzero spike positions are chosen randomly and the nonzeros values are chosen randomly from \([m, M]\). The observation data \( b \) is generated by

\[
b = Ax^* + e
\]

where \( e \) is drawn according to the Gaussian distribution with zero mean and variance \( 10^{-2} \). In the experiment, we consider a range of degrees of sparseness: the number \( T \) of nonzero spikes in \( x^* \) ranges from 1 to 100 in the support. We set \( n = 10^2 \) and \( m = \text{round}(0.2 \times n) \). In the following figures, we use "CPU time" to denote the CPU time in seconds and "error" to denote the error (error= \( \|x^* - x^k\|_\infty \)). "CPU time" and "error" are computed on the average of 10 independent realizations of the problem. To have more insights into the numerical results, we utilize the performance profiles by Dolan and Moré [14] to evaluate "CPU time" and "error" in Figures 1-4. From Figures 1-4, we observe that AMCP is faster than UPDASC, and UPDASC is faster than GIST and MSCR. Moreover, for most problems, we also observe that AMCP obtains the similar error as the one of UPDASC for most problems and the error obtained by AMCP and UPDASC is better than the error obtained by GIST and MSCR.

**D. RECONSTRUCTION FROM NOISE OBSERVATION FOR LARGE-SCALE PROBLEMS**

In the experiment, we use the MATLAB command \texttt{sprandn(m,n,density)} to generate the random matrices \( A \). We also adjusted each column of \( A \) such that the norm of each column of \( A \) is one. The true signal \( x^* \) and the observation \( b \) were generated by the same way as the second experiment. We set the parameter "density"= 0.01 for \( n = 2^{16}, 2^{17}, 2^{18} \) and "density"= 0.001 for \( 2^{19} \). Table 2 lists the CPU time (time) and the error (error= \( \|x^k - x^*\|_\infty \)) from Table 2, we can see that AMCP, PDASC and GIST obtain the similar error when \( n \leq 2^{17} \) and both AMCP and PDASC obtain the similar error for all tested problems. In summary, we can observe clearly that AMCP obtains the best CPU time and the best error among all tested algorithms for most problems.

**VI. CONCLUSION**

In this paper, an active set second-order algorithm for MCP regularized optimization is proposed. The active sets are estimated by an identification technique that can accurately identify the zero components in a neighborhood of a stationary point. The search direction consists of two parts: some of the components are simply defined; the other components are determined by a second-order algorithm. A nonmonotone
line search strategy that guarantees global convergence is used. The numerical comparisons with several state-of-art methods demonstrate the efficiency of the proposed method.

REFERENCES

[1] M. Afonso, J. Bioucas-Dias and M. Figueiredo, Fast image recovery using variable splitting and constrained optimization, IEEE Trans. Image Process., 19 (2010), pp. 2345-2356.

[2] A. Beck and M. Teboulle, A fast iterative shrinkage-thresholding algorithm for linear inverse problems, SIAM J. Imaging Sci., 2 (2009), pp. 183-202.

[3] A. Beck and N. Hallak, Proximal mapping for symmetric penalty and sparsity, SIAM J. Optim., 28 (2018), pp. 490-527.

[4] W. Bian and X. Chen, A smoothing proximal gradient algorithm for nonsmooth convex regression with cardinality penalty, SIAM J. Numer. Anal., 58 (2020), pp. 858-883.

[5] A.M. Bruckstein, D.L. Donoho and M. Elad, From sparse solutions of linear systems of equations to sparse modeling of signals and images, SIAM J. Math. Anal., 37 (2005), pp. 31-39.

[6] L. Breiman, Better subset regression using the nonnegative garrote, Technometrics, 28 (1986), pp. 301-308.

[7] E.J. Candès, M.B. Wakin, and S.P. Boyd, Enhancing sparsity by reweighted l1 minimization, J. Fourier Anal. Appl., 14 (2008), pp. 877-907.
This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.3020834, IEEE Access

G.M. Fung and O.L. Mangasarian, Equivalence of minimal

J. Fan and R. Li, Variable selection via nonconcave penalized likelihood

E.D. Dolan and J. J. Moré, Benchmarking optimization software with

G. Davis, S. Mallat, and M. Avellaneda, Adaptive greedy approximations,

M. Fornasier and R. Ward, Iterative thresholding meets free-discontinuity

W.Y. Cheng and Y.H. Dai, Gradient-based method with active set strategy

T.Y. Chen, F.E. Curtis and D.P. Robinson, A reduced-space algorithm for

R = 10^4

CPU time (seconds)

% of Problems

MSCR

GIST

AMCP

PDASC

0 1 2 3 4 5 6 7

FIGURE 3. Performance profiles based on CPU time in log2 scale for

R = 10^4

0 1 2 3 4 5 6 7

FIGURE 4. Performance profiles based on error in log2 scale for R = 10^2

8. X Chen, L. Niu, and Y. Yuan, Optimality conditions and a smoothing trust

9. Y.C. Chen, D.D. Ge, M.D. Wang, Z.Z. Wang, Y.Y. Ye and Y. Hao, Strong

574-747.

A reduced-space algorithm for minimizing \( \ell_1 \)-regularized convex function, SIAM J. Optim., 3 (2017), pp. 1583-1610.

W.Y. Cheng and Y.H. Dai, Gradient-based method with active set strategy for \( \ell_1 \) optimization, Math. Comp. 87 (2018), pp. 1283-1305

12. G. Davis, S. Mallat, and M. Avellaneda, Adaptive greedy approximations,

Constr. Approx., 13 (1997), pp. 57-98.

I. Daubechies, M. Defrise and C. De. Mol, An iterative thresholding algorithm for

algorithm for linear inverse problems with a sparsity constraint, Comm.

Pure Appl. Math., 57 (2004), pp.

E.D. Dolan and J. J. Moré, Benchmarking optimization software with performance profiles, Math. Program., 91 (2002), pp. 201-213.

15. J. Fan and R. Li, Variable selection via nonconcave penalized likelihood and its oracle properties, J. Amer. Statist. Assoc., 96 (2001), pp. 1348-1360

16. J. Fan and R. Li, Statistical challenges with high dimensionality: Feature

selection in knowledge discovery, in Proceedings of the International Congress of Mathematicians, Vol. 3, Madrid, Spain, European Mathematical Society, Zürich, 2006, pp. 595-622.

17. M.A. T. Figueiredo, R.D. Nowak and S.J. Wright, Gradient projection for sparse reconstruction: approximation to compressed sensing and other inverse problems, IEEE J. Sel. Top. Signal Process., 1 (2007), pp. 586-597.

18. M. Fornasier and R. Ward, Iterative thresholding meets free-discontinuity problems, Found. Comput. Math., 10 (2010), pp. 527-567.

19. S. Foucart and M.J. Lai, Sparsest solutions of underdetermined linear systems via \( \ell_q, 0 < q < 1 \), Appl Comput Harmon A., 26 (2009), pp. 395-407.

20. G.M. Fung and O.L. Mangasarian, Equivalence of minimal \( \ell_0 \)- and \( \ell_1 \)-norm solutions of linear equalities, inequalities and linear programs for sufficiently small \( \rho \), J. Optim. Theory Appl., 151 (2011), pp. 1-10.

21. P. Gong, C. Zhang, Z. Lu, J. Huang and J. Ye, A general iterative shrinkage and thresholding algorithm for non-convex regularized optimization problems, In Proc. 30th Int. Conf. Mach. Learn. (ICML-13), pp. 37-45, 2013.

22. L. Grippo, F. Lampariello and S. Lucidi, A nonmonotone line search technique for Newton’s method. SIAM J. Numer. Anal., 23 (1986), pp. 707-716.

23. L. Grippo, F. Lampariello and S. Lucidi, A truncated Newton method with nonmonotone line search for unconstrained optimization, J. Optim. Theory Appl., 60 (1989), pp. 401-419.

24. Y.L. Jiao, B.T. Jin, X.L. Lu and W. Ren, A unified primal dual active set algorithm for nonconvex sparse recovery, to appear in Statistical Science.

25. E.T. Hale, W. Yin and Y. Zhang, Fixed-point continuation for \( \ell_1 \) minimization: methodology and convergence, SIAM J. Optim., 19 (2008), pp. 1107-1130.

26. J.W. Liang, J. Fadili and G. Peyré, Activity identification and local linear convergence of forward-backward-type methods, SIAM J. Optim., 1 (2017), pp. 408-437.

27. J. Lv and Y. Fan, A unified approach to model selection and sparse recovery using regularized least squares, Ann. Statist., 37 (2009), pp. 3498-3528.

28. B.C. Liu, T.Yao, R.Z. Li and Y.Y. Ye, Folded concave penalized linear regression: sparsity, statistical performance, and algorithmic theory for local solutions, Math. Program., 166 (2017), pp. 207-240.

29. S.G. Mallat and Z. Zhang, Matching pursuits with time-frequency dictionaries, IEEE Trans. Signal Process., 41 (1993), pp. 3397-3415.

30. B.K. Natarajan, Sparse approximate solutions to linear systems, SIAM J. Comput., 24 (1995), pp. 227-234.

31. D. Needell, J.A. Tropp, CoSaMP: iterative signal recovery from incomplete and inaccurate samples, Appl. Comput. Harmon. Anal., 26 (2009), pp. 301-321.

32. Y.C. Pati, R. Rezaifar and P.S. Krishnaprasad, Orthogonal matching pursuit: Recursive function approximation with algorithms to wavelet decomposition, in Proceedings of the 27th Asilomar Conference on Signals, Systems and Computers (Pacific Grove, CA), IEEE, Washington, DC, 1993, pp. 40-44.

33. M.D. Santis, S. Lucidi and F. Rinaldi, A fast active set block coordinate descent algorithm for \( \ell_1 \)-regularized least squares, SIAM. J. Optim., 26 (2016), pp. 781-809.

34. E. Soubies, L. Blanc-Féraud and G. Aubert, A unified view of exact continuous penalties for \( \ell_2 \)-\( \ell_1 \) minimization, SIAM J. Optim., 27 (2017), pp. 2034-2060.

35. C. Sousedík, J. Idier, D. Brie, and J. Duan, From Bernoulli-Gaussian deconvolution to sparse signal restoration, IEEE Trans. Signal Process., 59 (2011), pp. 4572-4584.

36. E. Soubies, L. Blanc-Féraud and G. Aubert, A continuous exact \( \ell_0 \) penalty (CEL0) for least squares regularized problem, SIAM J. Imaging Sci., 8 (2015), pp. 1607-1639.

37. V.N. Temlyakov, Greedy approximation, Acta Numer., 17 (2008), pp. 235-409.

38. J.A. Tropp, Greed is good: Algorithmic results for sparse approximation, IEEE Trans. Inform. Theory, 50 (2004), pp. 2231-2242.

39. P. Tseng and S.W. Yun, A coordinate gradient descent method for nonsmooth separable minimization, Math. Program., 117 (2009), pp. 387-423.

40. Z.W. Wen, W.T. Yin, D. Goldfarb and Z. Zhang, A fast algorithm for sparse reconstruction based on shrinkage, subspace optimization, and continuation, SIAM J. Sci. Comput., 32 (2010), pp. 1832-1857.

41. J. Wright, R.D. Nowak and M.A. T. Figueiredo, Sparse reconstruction by separable approximation, IEEE Trans. Signal Process., 57 (2009), pp. 2479-2493.

42. T. Zhang, Multi-stage convex relaxation for learning with sparse regularization, in Advances in Neural Information Processing Systems, 22 (2009), pp. 1929-1936

43. C.H. Zhang, Nearly unbiased variable selection under minimax concave penalty, Ann. Statist., 38 (2010), pp. 894-942.

44. S. Zhang and J. Xin, Minimization of transformed \( \ell_1 \) penalty: theory difference of convex function algorithm, and robust application in compressive sensing, Math. Program., Ser. B., 169 (2018), pp. 307-336.

45. C. Zhang and X. Chen, A smoothing active set method for linearly constrained non-Lipschitz nonconvex optimization, SIAM J. Optim., 30 (2020), pp. 1-30.