Nonconvex fraction function recovery sparse signal by convex optimization

Angang Cui, Jigen Peng, Haiyang Li and Meng Wen

Abstract—The problem of recovering a sparse signal from the linear constraints, known as the \( \ell_0 \)-norm minimization problem, has been attracting extensive attention in recent years. However, the \( \ell_0 \)-norm minimization problem is a NP-hard problem. In our latest work, a non-convex fraction function is studied to approximate the \( \ell_0 \)-norm in \( \ell_0 \)-norm minimization problem and translate this NP-hard problem into a fraction function minimization problem. The FP thresholding algorithm is generated to solve the regularization fraction function minimization problem. However, we find that there are some drawbacks for our previous proposed FP algorithm. One is that the FP algorithm always convergent to a local minima due to the non-convexity of fraction function. The other one is that the parameter \( a \), which influences the behaviour of non-convex fraction function \( \rho_a \), needs to be determined manually in every simulation, and how to determine the best parameter \( a \) is not an easy problem. To avoid these drawbacks, here instead, in this paper we generate an adaptive convex FP algorithm to solve the problem \((FP_{a,\lambda})\). When doing so, our adaptive convex FP algorithm will not only convergent to a global minima but also intelligent both for the choice of the regularization parameter \( \lambda \) and the parameter \( a \). These are the advantages for our convex algorithm compared with our previous proposed FP algorithm.

Index Terms—The \( \ell_0 \)-norm minimization problem, Regularization fraction function minimization problem, FP algorithm, Adaptive convex FP algorithm

I. INTRODUCTION

In information processing, many practical problems can be formulated as the following \( \ell_0 \)-minimization problem[1], [2], [3], [4]:

\[
(P_0) \quad \min_{x \in \mathbb{R}^n} \|x\|_0 \quad \text{subject to} \quad Ax = b. \tag{1}
\]

where \( A \in \mathbb{R}^{m \times n} \) is a real matrix of full row rank with \( m \ll n \), \( b \in \mathbb{R}^m \) is a nonzero real column vector, and \( \|x\|_0 \) is the \( \ell_0 \)-norm of real vector \( x \in \mathbb{R}^n \), which counts the number of the non-zero entries in vector \( x \) [5], [6]. The \( \ell_0 \)-minimization problem aims to seek the sparsest signals which satisfy the underdetermined linear equations. However, it is NP-hard [7], [8] because of the discrete and discontinuous nature of the \( \ell_0 \)-norm.

In [4], we substitute the discontinuous \( \ell_0 \)-norm \( \|x\|_0 \) by the sparsity promoting penalty function

\[
P_a(x) = \sum_{i=1}^{n} \frac{a|x_i|}{a|x_i| + 1}, \quad a > 0, \tag{2}
\]

where

\[
\rho_a(t) = \frac{a|t|}{a|t| + 1} \tag{3}
\]

is the fraction function and concave in \( t \in [0, +\infty) \). It is clear that the non-convex function \( P_a(x) \) interpolates the \( \ell_0 \)-norm:

\[
\lim_{a \to +\infty} \rho_a(x_i) = \begin{cases} 0, & \text{if } x_i = 0; \\ 1, & \text{if } x_i \neq 0. \end{cases} \tag{4}
\]

Then, we translate problem \((P_0)\) into the following fraction function minimization problem

\[
(FP_a) \quad \min_{x \in \mathbb{R}^n} P_a(x) \quad \text{subject to} \quad Ax = b \tag{5}
\]

for the constrained form and

\[
(FP_{a,\lambda}) \quad \min_{x \in \mathbb{R}^n} \left\{ \|Ax - b\|^2 + \lambda P_a(x) \right\} \tag{6}
\]

for the regularized form, where \( \lambda > 0 \) is the regularized parameter.

In [4], the FP algorithm is proposed to solve the problem \((FP_{a,\lambda})\). A large number of numerical experiments on some sparse signal recovery problems have shown that the FP algorithm performances very well in recovering a sparse signal compared with some state-of-art methods. In the following study, we find that there are some drawbacks in our previous proposed FP algorithm. One is that the FP algorithm always converges to a local minima due to the non-convexity of function function. The other one is that the parameter \( a \), which influences the behaviour of non-convex fraction function \( \rho_a \), needs to be determined manually in every simulation. In fact, how to determine the best parameter \( a \) is not an easy problem.

The rest of this paper is organized as follows. In Section II, we review some known results about our previous proposed FP algorithm for solving the problem \((FP_{a,\lambda})\). In Section III, an convex FP algorithm is proposed to solve the regularized problem \((FP_{a,\lambda})\), and also establish the global convergence for our convex algorithm. In Section IV, a series of experiments on some sparse signal recovery problems are demonstrated. In V, we conclude some remarks in this paper.
II. The FP Algorithm for Solving the Problem 
\( (FP_{\lambda}) \)

In this section, we just review some known results from our latest work [4] for our new proposed FP algorithm to solve the problem \( (FP_{\lambda}) \).

**Lemma 1.** (44) Define a function of \( \beta \in \mathbb{R} \) as

\[
f_{a,\lambda}(\beta) = (\beta - \gamma)^2 + \lambda \rho_a(\beta)
\]

where \( \gamma \in \mathbb{R} \) and \( \lambda > 0 \), the optimal solution to

\[
\min_{\beta \in \mathbb{R}} f_{a,\lambda}(\beta)
\]

can be described as

\[
\beta^* = \begin{cases} 
  g_{a,\lambda}(\gamma), & \text{if } |\gamma| > t_{a,\lambda}; \\
  0, & \text{if } |\gamma| \leq t_{a,\lambda},
\end{cases}
\]

where \( g_{a,\lambda}(\gamma) \) is defined as

\[
g_{a,\lambda}(\gamma) = \text{sign}(\gamma) \left( \frac{1 + a|\gamma|}{2} \left( 1 + 2 \cos \left( \frac{\phi(\gamma)}{a} \right) \right) - 1 \right)
\]

and the threshold value \( t_{a,\lambda} \) satisfies

\[
t_{a,\lambda} = \begin{cases} 
  \frac{\lambda_1}{\sqrt{a} - \frac{1}{\sqrt{a}}}, & \text{if } \lambda \leq \frac{1}{\sqrt{a}}; \\
  \lambda_0, & \text{if } \lambda > \frac{1}{\sqrt{a}}.
\end{cases}
\]

**Definition 1.** (88) The nonincreasing rearrangement of the vector \( x \in \mathbb{R}^n \) is the vector \( |x| \in \mathbb{R}^n \) for which

\[
|x|_1 \geq |x|_2 \geq \cdots \geq |x|_n \geq 0
\]

and there is a permutation \( \pi : [n] \to [n] \) with \( |x|_i = |x_{\pi(i)}| \) for all \( i \in [n] \).

Now, we consider the following regularization function

\[
C_\lambda(x) = \|Ax - b\|_2^2 + \lambda P_a(x)
\]

and its surrogate function

\[
C_{\lambda,\mu}(x, z) = \mu C_\lambda(x) - \|Ax - Az\|_2^2 + \|x - z\|_2^2
\]

for any \( \lambda > 0 \), \( \mu > 0 \) and \( z \in \mathbb{R}^n \). When we set \( 0 < \mu \leq \|A\|_2^{-2} \), we can get that

\[
\|x - z\|_2^2 - \mu \|Ax - Az\|_2^2 \geq 0.
\]

Therefore, we have

\[
C_{\lambda,\mu}(x, z) = \mu C_\lambda(x) - \|Ax - Az\|_2^2 + \|x - z\|_2^2 \geq \mu C_\lambda(x)
\]

\[
\geq \mu C_\lambda(x^*)
\]

\[
= C_\mu(x^*, x^*),
\]

which implies that \( x^* \) is also a minimizer of \( C_{\lambda,\mu}(x, x^*) \) on \( x \in \mathbb{R}^n \).

On the other hand, \( C_{\lambda,\mu}(x, z) \) with \( z = x^* \) can be reexpressed as

\[
C_{\lambda,\mu}(x, x^*) = \|x - (x^* - \mu A^T Ax^* + \mu A^T b)\|_2^2
\]

\[
+ \lambda \mu P_a(x) + \mu \|b\|_2^2 + \|x^*\|_2^2 - \mu \|Ax^*\|_2^2
\]

\[
- \|x - B_{\mu}(x^*)\|_2^2 + \lambda \mu P_a(x) + \mu \|b\|_2^2
\]

\[
+ \|x^*\|_2^2 - \mu \|Ax^*\|_2^2 - \|B_{\mu}(x^*)\|_2^2,
\]

where \( B_{\mu}(x^*) = x^* + \mu A^T (b - Ax^*) \). This implies that for any fixed \( \lambda > 0 \), \( \mu > 0 \), minimizing \( C_{\lambda,\mu}(x, x^*) \) on \( x \in \mathbb{R}^n \) is equivalent to solve

\[
\min_{x \in \mathbb{R}^n} \left\{ \|x - B_{\mu}(x^*)\|_2^2 + \lambda \mu P_a(x) \right\}.
\]

The objective function in (14) is separable, so, \( x^* = (x_1^*, x_2^*, \ldots, x_n^*) \in \mathbb{R}^n \) is the minimizer of \( C_{\lambda,\mu}(x, x^*) \) if and only if, for any \( i = 1, 2, \ldots, n \), \( x_i^* \) solves the problem

\[
\min_{x_i \in \mathbb{R}} \left\{ \|x_i - (B_{\mu}(x_i^*))_i\|_2^2 + \lambda \mu P_a(x_i) \right\}.
\]

By Lemma 1, the minimizer \( x_i^* \) of minimization (15) is given by

\[
x_i^* = \begin{cases} 
  g_{a,\lambda \mu}(\mu (B_{\mu}(x_i^*))_i), & \text{if } \|B_{\mu}(x_i^*)\|_2^2 > t_{a,\lambda \mu}; \\
  0, & \text{if } \|B_{\mu}(x_i^*)\|_2^2 \leq t_{a,\lambda \mu}.
\end{cases}
\]

for \( i = 1, 2, \ldots, n \), where \( g_{a,\lambda} \) and \( t_{a,\lambda} \) are obtained by replacing \( \lambda \) with \( \lambda \mu \) in \( g_{a,\lambda} \) and \( t_{a,\lambda} \), which are defined in (9) and (10).

With the representation (16), the FP algorithm for solving the problem \( (FP_{\lambda}) \) can be naturally given by

\[
x_{k+1} \leftarrow \begin{cases} 
  g_{a,\lambda \mu}(\mu (B_{\mu}(x_k^*))_i), & \text{if } \|B_{\mu}(x_k^*)\|_2^2 > t_{a,\lambda \mu}; \\
  0, & \text{if } \|B_{\mu}(x_k^*)\|_2^2 \leq t_{a,\lambda \mu}.
\end{cases}
\]

for \( i = 1, 2, \ldots, n \).

In [4], the cross-validation method is accepted to select the proper regularization parameter \( \lambda \). Suppose that the vector \( x^* \) of sparsity \( r \) is the optimal solution to the problem \( (FP_{\lambda}) \). In each iteration, the regularization parameter \( \lambda \) can be selected as

\[
\lambda = \begin{cases} 
  \lambda^1_k = \frac{2 \|B_{\mu}(x^*_k)\|_2}{4 \|x^*_k\|_2^2 - (2a \|B_{\mu}(x^*_k)\|_2 + 1)^2}, & \text{if } \lambda^1_k \leq \frac{1}{\sigma^p}; \\
  \lambda^2_k = \frac{(1 - \epsilon) (2a \|B_{\mu}(x^*_k)\|_2 + 1)^2}{4 \|x^*_k\|_2^2}, & \text{if } \lambda^2_k > \frac{1}{\sigma^p},
\end{cases}
\]

where \( \epsilon > 0 \) is a very small positive number such as 0.01 or 0.001.

There is one more thing needed to be mentioned that, in each iteration, the threshold value function is set to \( t_{a,\lambda \mu} = \frac{\lambda \mu}{2} \) when \( \lambda = \lambda^1_k \), and \( t_{a,\lambda \mu} = \sqrt{\lambda^1_k - \frac{1}{\sigma^p}} \) when \( \lambda = \lambda^2_k \). The FP algorithm for solving the problem \( (FP_{\lambda}) \) is summarized in Algorithm 1.

A large number of numerical experiments on some sparse recovery problems have shown that the FP algorithm performances very well in recovering a sparse signal compared with some state-of-art methods. It is worth to noting that there are some drawbacks in our FP algorithm. One is that the FP algorithm always converges to a local minima due to the non-convexity of fraction function. The other one is that the parameter \( \alpha \), which influences the behaviour of non-convex
fraction function $\rho_\alpha$ needs to be determined manually in every simulation. How to ensure the FP algorithm converges to the global minima and determine the best parameter $a$ for the FP algorithm are two really challenging problem.

**Algorithm 1**: FP algorithm [4])

**Input**: $A \in \mathbb{R}^{m \times n}$, $b \in \mathbb{R}^m$, $0 < \mu < \|A\|_2^{-2}$, $\alpha > 0$ and $\epsilon > 0$;

**Initialize**: Given $x^0 \in \mathbb{R}^n$;

while not converged do

- $B(x^k) = x^k + \mu A^T(b - Ax^k)$;
- $\lambda_1^k = \frac{2(B(x^k))_{i+1}}{4}\lambda = \frac{(1-\epsilon)(2a|B(x^k)|_{i+1})^2}{4\alpha^2}\mu$;
- if $\lambda_1^k \leq \frac{1}{\alpha^2}\mu$ then
  - $\lambda = \lambda_1^k$, $t_{a,\lambda} = \frac{\lambda a}{\mu}$
  - for $i = 1 : n$
    - if $B_{\mu}(x^k)_i > t_{a,\lambda}$, then
      - $x^k_{i+1} = g_{a,\lambda}(B_{\mu}(x^k)_i)$;
    - $B_{\mu}(x^k)_i \leq t_{a,\lambda}$, then
      - $x^k_{i+1} = 0$;
else
  - $\lambda = \lambda_2^k$, $t_{a,\lambda} = \sqrt{\lambda_2^k \frac{1}{\alpha^2}}$
  - for $i = 1 : n$
    - if $B_{\mu}(x^k)_i > t_{a,\lambda}$, then
      - $x^k_{i+1} = g_{a,\lambda}(B_{\mu}(x^k)_i)$;
    - $B_{\mu}(x^k)_i \leq t_{a,\lambda}$, then
      - $x^k_{i+1} = 0$;
end

$k \rightarrow k + 1$;

end while

return $x^*$

### III. A CONVEX FP ALGORITHM FOR SOLVING THE PROBLEM ($FP_\alpha, \lambda$)

In this section, here instead, we will generate a convex FP algorithm to solve the problem ($FP_\alpha, \lambda$). The convexity of our FP algorithm is obtained by the convexity of the objective function in problem (15) which is ensured by constraining the parameter $a$ of the non-convex fraction function. Our convex FP algorithm will not only converges to the global minima but also intelligent both for the choice of the parameter $a$ and the regularization parameter $\lambda$, which are the advantages for our convex FP algorithm compared with the our previous proposed FP algorithm.

In order to benefit from convex optimization principles in solving (15), we first seek to ensure the objective function

$$h_{a,\lambda} (x_i) := (x_i - (B_{\mu}(x^*)_i))^2 + \lambda \rho_a (x_i)$$

in (16) is convex by controlling the parameter $a$.

**Theorem 1.** For any $0 < a \leq \frac{1}{\sqrt{\lambda}}$, the function $f_{a,\lambda}$ defined in equation (7) is strictly convex.

**Proof.** It is clear to see that the function

$$f_{a,\lambda} (\beta) = (\beta - \gamma)^2 + \lambda \rho_a (\beta)$$

is differentiable on $\mathbb{R}\setminus\{0\}$. For $x \neq 0$, the derivative of $f_{a,\lambda}$ is given by

$$f'_{a,\lambda} (\beta) = 2(\beta - \gamma) + \frac{\lambda a}{(a|\beta| + \lambda)^2} \text{sign} (\beta), \quad \beta \neq 0. \quad (20)$$

Let us find the range of $a$ for which $f_{a,\lambda}$ is convex. Notice that

$$f'_{a,\lambda} (0^+) = -\lambda a > \rho'_a (\lambda) (0^-) \quad (21)$$

and

$$f''_{a,\lambda} (\beta) = 2 - \frac{2\lambda a^2}{(a\beta + \lambda)^2}, \quad \beta > 0. \quad (22)$$

We can get that while $0 < a \leq \frac{1}{\sqrt{\lambda}}$, the function $f'_{a,\lambda} (\beta)$ defined in (20) is increasing, then the function $f_{a,\lambda}$ in (7) is strictly convex. This completes the proof.

![Fig. 1. The behavior of the function $f_{a,\lambda} (\beta) = (\beta - \gamma)^2 + \lambda \rho_a (\beta)$ with $\lambda = 0.49$, $a = 1.1$ and $\gamma = 0.$](image1.png)

![Fig. 2. The behavior of the function $f_{a,\lambda} (\beta) = (\beta - \gamma)^2 + \lambda \rho_a (\beta)$ with $\lambda = 0.49$, $a = 50$ and $\gamma = 0.$](image2.png)
when \( \lambda = 0.49 \) and \( a = 50 \), the graph presented in Fig. 2 show that the function \( f_{a,\lambda}(\beta) = (\beta - \gamma)^2 + \lambda \rho_a(\beta) \) is non-convex.

**Theorem 2.** Suppose \( 0 < a \leq \frac{1}{\sqrt{\lambda}} \), the optimal solution to \( \min_{\beta \in \mathbb{R}} f_{a,\lambda}(\beta) \) can be described as
\[
\beta^* = \begin{cases} 
  g_{a,\lambda}(\gamma), & \text{if } |\gamma| > t_{a,\lambda}; \\
  0, & \text{if } |\gamma| \leq t_{a,\lambda},
\end{cases} \tag{23}
\]
where \( g_{a,\lambda}(\gamma) \) is defined as
\[
g_{a,\lambda}(\gamma) = \text{sign}(\gamma) \left( 1 + \frac{1}{a} \cos\left( \frac{\lambda \rho_a(\gamma)}{\lambda \mu} \right) \right), \tag{24}
\]
and the threshold value \( t_{a,\lambda} \) satisfies
\[
t_{a,\lambda} = \frac{\lambda a}{2}. \tag{25}
\]

**Proof.** We can see that the condition \( 0 < a \leq \frac{1}{\sqrt{\lambda}} \) implies that \( \lambda \leq \frac{1}{\sqrt{\tau}} \), which reduces (10) into (25). This completes the proof. \( \square \)

By Theorem 1, if the constant \( a \) satisfies \( 0 < a \leq \frac{1}{\sqrt{\lambda \mu}} \), then we have the following corollary.

**Corollary 1.** For any \( 0 < a \leq \frac{1}{\sqrt{\lambda \mu}} \), the objective function in (15) is strictly convex.

Corollary 1 shows that the objective function to the problem (15) is a strictly convex by controlling the parameter \( a \) as \( 0 < a \leq \frac{1}{\sqrt{\lambda \mu}} \) in fraction function \( \rho_a \). Therefore, we can get the global minimizer to the problem (15) by controlling the parameter \( a \) as \( 0 < a \leq \frac{1}{\sqrt{\lambda \mu}} \). In the following description, we will discuss a convex FP algorithm to solve the problem (\( F_P \))

Similar as the generation of the FP algorithm, the convex FP algorithm for solving the problem (\( F_P \)) can be expressed as
\[
x_{i+1}^k = \begin{cases} 
  g_{a,\lambda \mu}(\mu_i(x^k)), & \text{if } \mu_i(x^k) > t_{a,\lambda \mu}; \\
  0, & \text{if } \mu_i(x^k) \leq t_{a,\lambda \mu},
\end{cases} \tag{26}
\]
for \( i = 1, 2, \ldots, n \), where the threshold value \( t_{a,\lambda \mu} \) satisfies
\[
t_{a, \lambda \mu} = \frac{\lambda \mu}{2}. \tag{27}
\]
is obtained by replacing \( \lambda \) with \( \lambda \mu \) in \( t_{a,\lambda} \) which is defined in (25). We conclude our convex FP algorithm in Algorithm 2.

**Theorem 3.** Let \( \{x^k\} \) be the sequence generated by convex FP algorithm with \( 0 < \mu < \|A\|^{-2} \). Then
1) The sequence \( \{C(x^k)\} \) is decreasing.
2) \( \{x^k\} \) is asymptotically regular, i.e., \( \lim_{k \to \infty} \|x^{k+1} - x^*\|^2 = 0 \).
3) \( \{x^k\} \) converges to the global stationary point of the iteration (25).

**Proof.** The proof is similar to the proof of Theorem 4.1 in [9]. Due to the convexity of the objective function in (15), the sequence \( \{x^k\} \) converges to the global stationary point of the iteration (26).

An important question we should face is that the quality of convex FP algorithm depends on the setting of the parameters \( \lambda \) and \( a \), and how to select the proper regularization parameters \( \lambda \) and \( a \) for our convex FP algorithm is a very hard problem. In detailed applications, the parameters \( \lambda \) and \( a \) must be carefully chosen. In the following description, we will generate an adaptive convex FP algorithm to solve the problem (\( F_{P,\lambda} \)). This adaptive convex FP algorithm will be intelligent both for the choice of the parameter \( a \) and the regularization parameter \( \lambda \).

1) **Intelligent for the choice of \( a \):** We notice that the parameter \( a \) in convex FP algorithm should be satisfied \( 0 < a \leq \frac{1}{\sqrt{\lambda \mu}} \).

Therefore, we can choose the parameter \( a \) as
\[
a = \frac{\tau}{\sqrt{\lambda \mu}}. \tag{28}
\]
where \( \tau \in (0, 1] \) is a given positive number. While we set \( a = \frac{\tau}{\sqrt{\lambda \mu}} \), the threshold value \( t_{a,\lambda \mu} \) in convex FP algorithm can be expressed as
\[
t_{a, \lambda \mu} = \frac{\tau \sqrt{\lambda \mu}}{2}. \tag{29}
\]
To see clear that once the value of the regularization parameter \( \lambda \) is determined, the parameter \( a \) can be given by (28), and therefore convex FP algorithm will be intelligent for the choice of the parameter \( a \). For the choice of the proper regularization parameter \( \lambda \), here, the cross-validation method (which is accepted to select the proper regularization parameter \( \lambda \) in FP algorithm) is again used to select the proper regularization parameter \( \lambda \) in our convex FP algorithm.

2) **Intelligent for the choice of \( \lambda \):** Let the vector \( x^0 \) of sparsity \( r \) be the optimal solution to the problem (\( F_{P,\lambda} \)). Then, the following inequalities hold
\[
|B_{\mu}(x^*)|_i > t_{a,\lambda \mu} \Rightarrow i \in \{1, 2, \ldots, r\},
\]
\[
|B_{\mu}(x^*)|_j \leq t_{a,\lambda \mu} \Rightarrow j \in \{r + 1, r + 2, \ldots, n\}
\]
which implies that
\[
\frac{4|B_{\mu}(x^*)|_{i+1}^2}{\tau^2 \mu} \leq \lambda \leq \frac{4|B_{\mu}(x^*)|_i^2}{\tau^2 \mu}. \tag{31}
\]
The above estimation provides an exact location of the regularization parameter $\lambda$. We can then take the optimal regularization parameter $\lambda$ as

$$\lambda^* = \frac{4|B_\mu(x^*)|^2}{\tau^2\mu} + \zeta,$$

(32)

where $\zeta > 0$ is a very small positive number such as $10^{-3}$ or $10^{-4}$.

In each iteration, we approximate the optimal solution $x^*$ by $x^k$. Then, the resulting optimal parameters can be selected as

$$\lambda_k = \frac{4|B_\mu(x^k)|^2}{\tau^2\mu} + \zeta, \quad a_k = \frac{1}{\sqrt{\lambda_k\mu}}.$$

(33)

By above operations, our convex FP algorithm will be intelligent both for the choice of the parameters $a$ and $\lambda$. We call this convex FP algorithm which intelligent both for the choice of the parameters $a$ and $\lambda$ as adaptive convex FP algorithm and it is summarized in Algorithm 3.

**Algorithm 3**: Adaptive convex FP algorithm

**Input**: $A \in \mathbb{R}^{m \times n}$, $b \in \mathbb{R}^m$, $0 < \mu < \|A\|^{-2}$, $\lambda > 0$, $\zeta > 0$ and $\tau \in (0, 1]$;  
**Initialize**: Given $x^0 \in \mathbb{R}^n$;  
**while** not converged **do**  
$$B_\mu(x^k) = x^k + \mu A^T (b - Ax^k);$$  
$$\lambda_k = \frac{4|B_\mu(x^k)|^2}{\tau^2\mu} + \zeta, \quad a_k = \frac{1}{\sqrt{\lambda_k\mu}};$$  
**for** $i = 1$ : $n$  
1. $|(B_\mu(x^k))_i| > t_{a_k,\lambda_k,\mu}$, then  
   $$x^k_{i+1} = g_{a_k,\lambda_k,\mu}((B_\mu(x^k))_i);$$  
2. $|(B_\mu(x^k))_i| \leq t_{a_k,\lambda_k,\mu}$, then  
   $$x^k_{i+1} = 0;$$  
$k \to k + 1$;  
**end while**  
**return**: $x^*$

**IV. NUMERICAL EXPERIMENTS**

In this section, we present a series of numerical experiments on some sparse signal recovery problems to demonstrate the performances of our adaptive convex FP algorithm. In these numerical experiments, we compare our convex FP algorithm with our previous proposed FP algorithm [4] and Half algorithm [10]. These numerical experiments are all conducted on a personal computer (3.40GHz, 16.0GB RAM) with MATLAB R2015b.

We generate a Gaussian random matrix of size $100 \times 400$ with entries i.i.d. to Gaussian distribution, $\mathcal{N}(0, 1)$ as the measurement matrix $A$. The original $r$-sparse signal $x_0$ with dimension 400 was generated by choosing the non-zero locations uniformly over the support in random, and each nonzero entry is generated as follows:

$$x[i] = \eta_1[i]10^{\eta_2[i]},$$

(34)

where $\eta_1[i] = \pm 1$ with probability 1/2 (a random sign), $\eta_2[i]$ is uniformly distributed in $[0, 1]$ and the parameter $\alpha$ quantifies the dynamic range. We generate the measurement vector $b$ by $b = Ax_0$, and therefore we know the sparsest solution to $Ax_0 = b$. The stopping criterion is defined as

$$\frac{\|x^{k+1} - x^k\|^2}{\|x^k\|^2} \leq 10^{-15}$$
or maximum iteration step equal to 3000. The success recovery of the original sparse vector $x_0$ is measured by computing

$$RE = \|x^* - x_0\|_2.$$

In our experiments, if $RE \leq 10^{-4}$, we say that our algorithm can exact recovery the original $r$-sparse signal $x_0$. For each experiment, we repeatedly perform 30 tests and present average results in this paper. In these numerical experiments, we set $\mu = 0.99\|A\|^{-2}$, $\zeta = 10^{-4}$ in our adaptive convex FP algorithm.

![Fig. 3. The comparison of adaptive convex FP algorithm, FP algorithm and Half algorithm in the recovery of a sparse nonnegative signal with $\alpha = 1.5$.](image3)

![Fig. 4. The behavior of the function $f_{\alpha,\lambda}(\beta) = (\beta - \gamma)^2 + \lambda \rho_\alpha(\beta)$ with $\lambda = 1$, $\alpha = 10$ and $\gamma = 0$.](image4)
V. Conclusion

In this paper, we first review some known results from our lately work for FP algorithm in compressed sensing, and then generate an adaptive convex FP algorithm to solve the problem \((FP_{a,\lambda})\). Our adaptive convex FP algorithm will not only convergent to a global minima but also intelligent both for the choice of the regularization parameter \(\lambda\) and the parameter \(a\), which are the advantages for our adaptive convex FP algorithm compared with our previous proposed FP algorithm. Numerical experiments on some sparse signal recovery problems have shown that our proposed adaptive convex FP algorithm performs very well in recovering a sparse signal.

References

[1] D. L. Donoho. Compressed sensing. IEEE Transaction on Information Theory, 2006, 52(4): 1289–1306.
[2] A. M. Bruckstein, D. L. Donoho, M. Elad. From sparse solutions of systems of equations to sparse modelling of signals and images. SIAM Review, 2009, 51(1): 34–81.
[3] S. Ranjan, M. Vidyasagar. Tight performance bounds for compressed sensing with conventional and group sparsity. IEEE Transactions on Signal Processing, 2019, 67(11): 2854–2867.
[4] H. Li, Q. Zhang, A. Cui, J. Peng. Minimization of fraction function penalty in compressed sensing. arXiv preprint arXiv: 1705.06048, 2019.
[5] M. Elad. Sparse and Redundant Representations: from Theory to Applications in Signal and Image Processing. Springer, New York, 2010.
[6] J. Peng, S. Yue and H. Li. NP/CLP Equivalence: A Phenomenon Hidden among Models for Sparse Information Processing. IEEE Transactions on Information Theory, 2015, 61(7): 4028–4033.
[7] B. K. Natarajan. Sparse approximate solutions to linear systems, SIAM Journal on Computing, 1995, 24(2): 227C234.
[8] S. Foucart, H. Rauhut. A mathematical introduction to compressive sensing[MI]. New York: Springer Science+Business Media, 2013.
[9] D. Peng, N. Xiu, J. Yu. \(S_{1/2}\) regularization methods and fixed point algorithms for affine rank minimization problems, Computational Optimization and Applications, 2017, 67(3): 543–569.
[10] Z. Xu, X. Chang, F. Xu, H. Zhang. \(L_{1/2}\) regularization: a thresholding representation theory and a fast solver. IEEE Transactions on Neural Networks and Learning Systems, 2012, 23(7): 1013-1027.