



Research article

A linearly convergent self-adaptive gradient projection algorithm for sparse signal reconstruction in compressive sensing

Hengdi Wang¹, Jiakang Du³, Honglei Su¹ and Hongchun Sun^{2,*}

¹ School of Electronic Information, Qingdao University, Qingdao 266071, China

² School of Mathematics and Statistics, Linyi University, Linyi 276005, China

³ School of Management Science, Qufu Normal University, Rizhao 276800, China

* **Correspondence:** Email: hcsun68@126.com.

Abstract: For sparse signal reconstruction (SSR) problem in compressive sensing (CS), by the splitting technique, we first transform it into a continuously differentiable convex optimization problem, and then a new self-adaptive gradient projection algorithm is proposed to solve the SSR problem, which has fast solving speed and pinpoint accuracy when the dimension increases. Global convergence of the proposed algorithm is established in detail. Without any assumptions, we establish global R -linear convergence rate of the proposed algorithm, which is a new result for constrained convex (rather than strictly convex) quadratic programming problem. Furthermore, we can also obtain an approximate optimal solution in a finite number of iterations. Some numerical experiments are made on the sparse signal recovery and image restoration to exhibit the efficiency of the proposed algorithm. Compared with the state-of-the-art algorithms in SSR problem, the proposed algorithm is more accurate and efficient.

Keywords: self-adaptive gradient projection algorithm; global convergence; global R -linear convergence rate; SSR problem

Mathematics Subject Classification: 90C30, 90C33

1. Introduction

Sparse signal reconstruction (SSR), i.e., using the least entries from a complete dictionary to represent the signal of interest, is a popular technique for recovering a sparse signal or an image from an under-determined linear system [1, 2], which can be formulated as the following non-smooth minimization problem

$$\begin{aligned} \min \quad & \|x\|_0 \\ \text{s.t.} \quad & Ax = b, \end{aligned} \tag{1.1}$$

where $A \in R^{m \times n}$ ($m \ll n$) is the design matrix, $b \in R^n$ is the observation vector and $\|x\|_0$ denotes the number of nonzero elements of vector $x \in R^n$ to be estimated. Since the problem (1.1) is NP hard [3,4], it cannot be solved approximately within a fixed ratio unless $P = NP$. In addition, as for $Ax = b$, it is often the case that is ill-posed due to possible ill-conditionedness of A . To deal with the ill-posed case, and take into consideration sparse structure of (1.1), the ℓ_1 -norm regularization techniques have successfully been used for inducing sparsity of solutions [5]. Thus, the problem (1.1) is transformed into the following Lagrangian form or penalty form, which is usually written as the following form

$$\min_{x \in R^n} f(x) = \frac{1}{2} \|Ax - b\|_2^2 + \rho \|x\|_1, \quad (1.2)$$

where ρ is a positive parameter and $\|x\|_1 = \sum_{j=1}^n |x_j|$ denotes the ℓ_1 -norm of x .

Based on the convex relaxation form (1.2), many researchers have developed various algorithms for solving this problem, including interior-point methods [6], projected gradient methods [7], spectral gradient methods [8], spectral residual methods [9], conjugate gradient projection methods [10], alternating direction algorithms [11], and iterative thresholding [12] and so on.

A large class of the first order ℓ_1 -minimization algorithms (abbreviated as the first order algorithms) for problem (1.2) is based on the iterative shrinkage/thresholding (IST) method or on variants of IST [12, 13]. Recently, many extensions and modifications of these basic algorithms have been introduced and investigated. For instance, Hale et al. [14] present a fixed-point continuation method by shrinkage operator. Beck and Teboulle [15] propose a fast iterative shrinkage-thresholding algorithm (FISTA), which has virtually the same complexity as IST, but has better convergence properties. Bioucas-Dias and Figueiredo [16] present a two-step IST method, which has exhibited much faster convergence rate than IST. Another closely related method to IST is the proximal thresholding algorithm by Combettes et al. [17]. A gradient based algorithm is especially one of the most popular methods for solving (1.2). One of the earliest gradient projection method for sparse reconstruction was developed by Figueiredo et al. [7]. Another method, SPGL1, was proposed by van den Berg and Friedlander [18]. Other gradient based methods include Nesterov's algorithm (NESTA) [19] and Sparse Reconstruction by Separable Approximation method (SpaRSA) [20] all have good performance. A large class of the second order information of the underlying function based algorithms (abbreviated as the second order algorithms) is also popular for solving problem (1.2), which include Orthant-Based adaptive method (OBA) [21], FBS-Newton [22], Semismooth Newton method with multidimensional filter globalization (SNF) [23], Inexact successive quadratic approximation (SQA) method [24] and so on. Although the second order algorithms often outperform the first order algorithms in seeking high-precision solutions, the computational cost of each step is usually much higher than that of the first order algorithms.

In recent years, a lot of numerical algorithms about the equivalent forms of (1.2) have been extensively developed. For instance, Xiao and Zhu [25] transformed (1.2) into convex constrained monotone equations, and presented a conjugate gradient method for this equivalent form. Sun and Tian [26] gave a class of derivative-free conjugate gradient projection method for the equivalent form of (1.2). Sun et al. [27] reformulated (1.2) as variational inequality problem, and proposed a novel inverse matrix-free proximal point algorithm for the equivalent form of (1.2). Based on the same transformation, Feng and Wang [28] also proposed a projection-type algorithm without any backtracking line search. Xiao et al. [29] proposed a spectral gradient method for solving an equivalent

non-smooth equation of (1.2). Liu et al. [30, 31] presented a projection method to solve monotone nonlinear equations with convex constraints, and applied it to solve an equivalent non-smooth equation of (1.2). In [32], Wang et al. proposed a smooth and convex Lagrange-dual reformulation of (1.2), and many state of the art gradient-type algorithms can be used to solve the reformulation. Landi [33] considers the nonnegative constrained quadratic program reformulation of (1.2), and proposes modified Newton projection method to solve it. On the other hand, since (1.2) can be equivalently converted into a separable convex programming by introducing an auxiliary variable, the numerical methods which can solve the separable convex programming are applicable to solving (1.2), such as the alternating direction method of multipliers and its linearized version [11, 34], the Peaceman-Rachford splitting method (PRSM) or Douglas-Rachford splitting method (DRSM) of multipliers [35, 36], the symmetric alternating direction method of multipliers [37], etc. Certainly, problem (1.1) can be formulated as a splitting feasible problem (SFP) [38], and the numerical methods which are designed for SFP are also applicable to solve (1.1) [38, 39].

As can be seen from this brief review, existing methods are able to solve (1.2). However, the solving speed and accuracy are adversely affected when the dimension increases greatly. In the paper, we will establish a new self-adaptive gradient projection method for solving (1.2), which have closed-form solutions, the motivation for this approach is to achieve better numerical performance, especially for large-scale problem (1.1). From theoretical perspectives, the global linear rate convergence result for new algorithms is established without any assumptions, which is a new result for constrained convex (rather than strictly convex) quadratic programming problem. In addition, an approximate optimal solution is obtained in a finite number of iterations, and these stronger convergence results are one of the most important motivations for this work.

The remainder of this paper is organized as follows. In Section 2, we use the separating technique to transform (1.2) into a differentiable problem and present some related properties, which are the basis of our analysis. In Section 3, we propose a new self-adaptive gradient projection method, in which line search step is different from that in Qu [39]. The global convergence of the proposed method is discussed in detail. We establish a global R -linear rate convergence theorem without any assumptions, which is weaker than the condition in Theorem 3.3 in [34]. Furthermore, we can obtain an approximate optimal solution in a finite number of iterations. In Section 4, some numerical experiments on sparse signal reconstruction or image restoration are given, and compare the CPU time and the mean of squared error among the Algorithm 2.1 in [30], the CQ algorithm in [39], FISTA [15], the primal alternating direction method [11] and our proposed algorithm, and show that our algorithm is much faster than other algorithms, and also have higher accuracy. Finally, some remarks and conclusions are presented in Section 5 and Section 6, respectively.

To end this section, some notations used in this paper are in order. We use R_+^n to denote the nonnegative quadrant in R^n , and the x_+ denotes the orthogonal projection of vector $x \in R^n$ onto R_+^n , that is, $(x_i)_+ := \max\{x_i, 0\}$, $1 \leq i \leq n$; the norm $\|\cdot\|$ and $\|\cdot\|_1$ denote the Euclidean 2-norm and 1-norm, respectively. For $x, y \in R^n$, use $(x; y)$ to denote the column vector $(x^\top, y^\top)^\top$. We denote by $\det(M)$ the determinant of a matrix M and, by I_n the identity matrix of order n .

2. Preliminaries

In this section, a smooth equivalence transformation of the problem (1.2) is established.

Firstly, we briefly review the process for constructing a convex quadratic program problem of Figueiredo et al. [7]. For any vector $x \in R^n$, it can be formulated for $x = \mu - \nu$, $\mu \geq 0$, $\nu \geq 0$, where $\mu \in R^n$, $\nu \in R^n$, and $\mu_i = (x_i)_+$, $\nu_i = (-x_i)_+$ for all $i = 1, 2, \dots, n$. Subsequently, the ℓ_1 -norm of the vector can be represented as $\|x\|_1 = (e^\top, e^\top)(\mu; \nu)$, where $e \in R^n$ denotes the vector composed by elements 1, i.e., $e = (1, 1, \dots, 1)^\top$. Thus, we can reformulate (1.2) into

$$\begin{aligned} \min_{(\mu; \nu) \in R^{2n}} \quad & \frac{1}{2} \|(A, -A)(\mu; \nu) - b\|_2^2 + \rho(e^\top, e^\top)(\mu; \nu) \\ \text{s.t.} \quad & (\mu; \nu) \geq 0. \end{aligned} \quad (2.1)$$

In order to make the description more concise, we let $\omega = (\mu; \nu)$. Thus, the problem (2.1) can be further written as

$$\begin{aligned} \min \quad & f(\omega) = \frac{1}{2}(\omega^\top M \omega - 2p^\top \omega + b^\top b) \\ \text{s.t.} \quad & \omega \in R_+^{2n}, \end{aligned} \quad (2.2)$$

where $M = \begin{pmatrix} A^\top A & -A^\top A \\ -A^\top A & A^\top A \end{pmatrix}$, $p = \begin{pmatrix} A^\top \\ -A^\top \end{pmatrix} b - \rho \begin{pmatrix} e \\ e \end{pmatrix}$. We assume the solution of the problem is nonempty and denote it by Ω^* .

Although the dimension of (2.2) is twice of the original problems, it is shown that this increases only slightly the impact of computation. In addition, the computation of the function values or the gradient requires two matrix-vector multiplications involving A and A^\top [7].

Obviously, the problem (2.2) is a convex optimization problem, then the stationary set of (2.2) coincides with its solution set which also coincides with the solution set of the following problem: find $\omega^* \in R_+^{2n}$ such that

$$(\omega - \omega^*)^\top \nabla f(\omega^*) \geq 0, \quad \forall \omega \in R_+^{2n}, \quad (2.3)$$

where $\nabla f(\omega) = M\omega - p$.

Secondly, we will give the definition of projection operator and some related properties [40], which are the basis of our analysis.

For a nonempty closed convex set $K \subset R^n$ and vector $z \in R^n$, the orthogonal projection of z onto K , i.e., $\arg \min \{\|z - y\| \mid y \in K\}$, is denoted by $P_K(z)$.

Proposition 2.1. *Let K be a closed convex subset of R^n , then $\|P_K(z_1) - P_K(z_2)\| \leq \|z_1 - z_2\|$, $\forall z_1, z_2 \in R^n$.*

For (2.3) and $\omega \in R^{2n}$, define the projection residue

$$r(\omega, \hat{\rho}) = \|\omega - P_{R_+^{2n}}\{\omega - \hat{\rho}(M\omega - p)\}\|, \quad (2.4)$$

where $\hat{\rho} > 0$ is some constant. The projection residue is intimately related to the solution of (2.3) as shown by the following well-known result, which is due to Noor [41].

Proposition 2.2. *The vector ω^* is a solution of (2.3) if and only if $r(\omega^*, \hat{\rho}) = 0$ with some $\hat{\rho} > 0$.*

Finally, we recall the following lemma, which is the well-known fundamental property for a smooth function in the class $C^{1,1}$.

Lemma 2.1. *(Lemma 3.2 [42]) Let $F : R^n \rightarrow R$ be a continuously differentiable function with Lipschitz continuous gradient and Lipschitz constant L_F . Then, for any $L \geq L_F$, one has*

$$F(y) \leq F(\bar{y}) + \langle y - \bar{y}, \nabla F(\bar{y}) \rangle + \frac{L}{2} \|y - \bar{y}\|^2, \quad \forall y, \bar{y} \in R^n. \quad (2.5)$$

For any $\omega, \bar{\omega} \in R^{2n}$ and the function f defined in (2.2), one has $\|\nabla f(\omega) - \nabla f(\bar{\omega})\| \leq \|M\|\|\omega - \bar{\omega}\|$, i.e., the gradient function $\nabla f(\omega)$ be Lipschitz continuous and Lipschitz constant $\|M\|$. Applying Lemma 2.1, it follows that

$$\begin{aligned} f(p_L(\bar{\omega})) &\leq f(\bar{\omega}) + \langle p_L(\bar{\omega}) - \bar{\omega}, \nabla f(\bar{\omega}) \rangle + \frac{L}{2} \|p_L(\bar{\omega}) - \bar{\omega}\|^2 \\ &= Q(p_L(\bar{\omega}), \bar{\omega}), \quad \text{for } L \geq \|M\|, \end{aligned} \quad (2.6)$$

where

$$Q_L(\omega, \bar{\omega}) := f(\bar{\omega}) + \langle \omega - \bar{\omega}, \nabla f(\bar{\omega}) \rangle + \frac{L}{2} \|\omega - \bar{\omega}\|^2, \quad (2.7)$$

and

$$p_L(\bar{\omega}) = \arg \min_{\omega \in R_+^{2n}} \{Q_L(\omega, \bar{\omega})\}. \quad (2.8)$$

From (2.7) and (2.8), it is easy to calculate that

$$p_L(\bar{\omega}) = \arg \min_{\omega \in R_+^{2n}} \left\{ \left\| \omega - \left(\bar{\omega} - \frac{1}{L} \nabla f(\bar{\omega}) \right) \right\|^2 \right\}. \quad (2.9)$$

3. Algorithm and global convergence

In this section, we will propose a new algorithm for solving the model (2.3), and prove global convergence and R -linear convergence rate of the new algorithm in detail.

Now, we formally state our algorithm.

Algorithm 3.1

Step 0. Select $\beta > 0, \eta > 1, \epsilon \geq 0, \gamma \in (0, 1), \omega^0 \in R^{2n}$, and let $k := 0$.

Step 1. For the current iterate points ω^{k-1} , compute

$$\begin{aligned} \omega^k : &= p_{L_k}(\omega^{k-1}) \\ &= \arg \min_{\omega \in R_+^{2n}} \left\{ \left\| \omega - \left(\omega^{k-1} - \frac{1}{L_k} \nabla f(\omega^{k-1}) \right) \right\|^2 \right\} \\ &= \max \left\{ \omega^{k-1} - \frac{1}{L_k} \nabla f(\omega^{k-1}), 0 \right\}, \end{aligned} \quad (3.1)$$

where $L_k = \eta^{m_k} \beta$, and m_k is the smallest nonnegative integer m such that

$$f(\omega^k) \leq f(\omega^{k-1}) - \gamma \langle \omega^k - \omega^{k-1}, \nabla f(\omega^{k-1}) \rangle, \quad (3.2)$$

and

$$0 \leq (1 + \gamma) \langle \omega^k - \omega^{k-1}, \nabla f(\omega^{k-1}) \rangle + \frac{\eta^m \beta}{2} \|\omega^k - \omega^{k-1}\|^2. \quad (3.3)$$

Step 2. If $\|\omega^k - \omega^{k-1}\| \leq \epsilon$, stop. Then, ω^k is an approximate solution of (2.3). Otherwise, go to Step 1 with $k := k + 1$.

From (3.2) and (3.3), one has

$$f(\omega^k) \leq f(\omega^{k-1}) + \left\langle \omega^k - \omega^{k-1}, \nabla f(\omega^{k-1}) \right\rangle + \frac{\eta^m \beta}{2} \|\omega^k - \omega^{k-1}\|^2. \quad (3.4)$$

If $L_k \geq \|M\|$, using (3.4) and (2.6), we get

$$f(\omega^k) = f(p_{L_k}(\omega^{k-1})) \leq Q(p_{L_k}(\omega^{k-1}), \omega^{k-1}) = Q(\omega^k, \omega^{k-1}). \quad (3.5)$$

Furthermore,

$$f(\omega^k) \leq Q_{L_k}(\omega^k, \omega^{k-1}) \leq Q_{L_k}(\omega^{k-1}, \omega^{k-1}) = f(\omega^{k-1}), \quad k \geq 1. \quad (3.6)$$

Thus, the sequence $\{f(\omega^k)\}$ produced by Algorithm 3.1 is nonincreasing.

The termination criteria in step 2 of Algorithm 3.1 is reasonable, which can be proved by (3.20) and Theorem 3.2 below. It can make Algorithm 3.1 terminate in a finite step, which can be proved in Theorem 3.1 below.

Remark 3.1. By (3.2) and (3.3), we assure the existence of upper and lower bounds of the parameter L_k , see details from Lemma 3.1 below. Furthermore, we adopted Armijo-like technique in a new way, such that the iteration stepsize $\frac{1}{L_k}$ can be self-adaptive updated in Algorithm 3.1.

Lemma 3.1. Assume parameters $\beta > 0, \eta > 1$ defined as in Algorithm 3.1, then $\beta \leq L_k < \eta \|M\|, \forall k \geq 1$.

Proof. By $L_k = \eta^{m_k} \beta$ and $\eta > 1$, one has $L_k \geq \beta, \forall k \geq 1$. Since (3.2) and (3.3) are satisfied for some $\eta^m \beta \geq \|M\|$, combining this with the definition of m_k , we ensure that $L_k/\eta = \eta^{m_k-1} \beta$ must violate (3.2) or (3.3), it follows that (3.4) does not hold. Combining this with Lemma 2.1, we obtain that $L_k/\eta < \|M\|$ for every $k \geq 1$. Hence, the desired result follows. \square

Lemma 3.2. For any $\omega \in \mathbb{R}_+^{2n}$, we have

$$f(\omega) - f(\omega^k) \geq \frac{L_k}{2} \|\omega^k - \omega^{k-1}\|^2 + L_k \langle \omega - \omega^{k-1}, \omega^{k-1} - \omega^k \rangle, \quad \forall k \geq 1. \quad (3.7)$$

Proof. By (3.5), one has

$$f(\omega) - f(\omega^k) \geq f(\omega) - Q_{L_k}(\omega^k, \omega^{k-1}). \quad (3.8)$$

By (2.7) with $\omega = \omega^k, \bar{\omega} = \omega^{k-1}, L = L_k$, then

$$Q_{L_k}(\omega^k, \omega^{k-1}) = f(\omega^{k-1}) + \langle \omega^k - \omega^{k-1}, \nabla f(\omega^{k-1}) \rangle + \frac{L_k}{2} \|\omega^k - \omega^{k-1}\|^2. \quad (3.9)$$

Since f is convex, one has

$$f(\omega) \geq f(\omega^{k-1}) + \langle \omega - \omega^{k-1}, \nabla f(\omega^{k-1}) \rangle. \quad (3.10)$$

Combining (3.8) and (3.10) with (3.9), one has

$$\begin{aligned} f(\omega) - f(\omega^k) &\geq f(\omega) - Q_{L_k}(\omega^k, \omega^{k-1}) \\ &= f(\omega) - f(\omega^{k-1}) - \langle \omega^k - \omega^{k-1}, \nabla f(\omega^{k-1}) \rangle - \frac{L_k}{2} \|\omega^k - \omega^{k-1}\|^2 \\ &\geq \langle \omega - \omega^{k-1}, \nabla f(\omega^{k-1}) \rangle - \langle \omega^k - \omega^{k-1}, \nabla f(\omega^{k-1}) \rangle - \frac{L_k}{2} \|\omega^k - \omega^{k-1}\|^2 \\ &= \langle \omega - \omega^k, \nabla f(\omega^{k-1}) \rangle - \frac{L_k}{2} \|\omega^k - \omega^{k-1}\|^2. \end{aligned} \quad (3.11)$$

Applying the first-order optimality condition of (3.1), one has

$$\langle \omega - \omega^k, \nabla f(\omega^{k-1}) + L_k(\omega^k - \omega^{k-1}) \rangle \geq 0, \quad \forall \omega \in \mathbb{R}_+^{2n}, \quad (3.12)$$

i.e., $\langle \omega - \omega^k, \nabla f(\omega^{k-1}) \rangle \geq L_k \langle \omega - \omega^k, \omega^{k-1} - \omega^k \rangle$, $\forall \omega \in R_+^{2n}$. Combining this with (3.11), one has

$$\begin{aligned} f(\omega) - f(\omega^k) &\geq L_k \langle \omega - \omega^k, \omega^{k-1} - \omega^k \rangle - \frac{L_k}{2} \|\omega^k - \omega^{k-1}\|^2 \\ &= L_k \langle (\omega - \omega^{k-1}) + (\omega^{k-1} - \omega^k), \omega^{k-1} - \omega^k \rangle - \frac{L_k}{2} \|\omega^k - \omega^{k-1}\|^2 \\ &= L_k \langle \omega - \omega^{k-1}, \omega^{k-1} - \omega^k \rangle + \frac{L_k}{2} \|\omega^k - \omega^{k-1}\|^2. \end{aligned}$$

We next prove that the sequence $\{\omega^k\}$ terminates in a finite number of steps at an approximate optimal solution of (2.2).

Theorem 3.1. *Assume that ω^* is a solution of (2.3), and the sequence $\{\omega^k\}$ generated by Algorithm 3.1. Then, for any $k \geq 1$, one has*

$$f(\omega^k) - f(\omega^*) \leq \frac{\eta \|M\|}{2k} \|\omega^* - \omega^0\|^2. \quad (3.13)$$

Proof. In order to make the description more concise. In (3.7), let $\omega = \omega^*$, and replace ω^k with ω^m , we obtain

$$\begin{aligned} \frac{2}{L_m}(f(\omega^*) - f(\omega^m)) &\geq \|\omega^m - \omega^{m-1}\|^2 + 2\langle \omega^* - \omega^{m-1}, \omega^{m-1} - \omega^m \rangle \\ &= \|\omega^m - \omega^{m-1}\|^2 + \langle (\omega^* - \omega^m) + (\omega^m - \omega^{m-1}), \omega^{m-1} - \omega^m \rangle \\ &\quad + \langle \omega^* - \omega^{m-1}, (\omega^{m-1} - \omega^*) + (\omega^* - \omega^m) \rangle \\ &= \langle \omega^* - \omega^m, \omega^{m-1} - \omega^m \rangle \\ &\quad + \langle \omega^* - \omega^{m-1}, \omega^{m-1} - \omega^* \rangle + \langle \omega^* - \omega^{m-1}, \omega^* - \omega^m \rangle \\ &= \langle \omega^* - \omega^m, \omega^{m-1} - \omega^* + \omega^* - \omega^m \rangle \\ &\quad + \langle \omega^* - \omega^{m-1}, \omega^{m-1} - \omega^* \rangle + \langle \omega^* - \omega^{m-1}, \omega^* - \omega^m \rangle \\ &= \|\omega^* - \omega^m\|^2 - \|\omega^* - \omega^{m-1}\|^2. \end{aligned} \quad (3.14)$$

Since ω^* be a solution of (2.3), then it be also a solution of (2.2), and one has $f(\omega^*) - f(\omega^m) \leq 0$. Combining this with Lemma 3.1, we obtain

$$\frac{2}{\eta \|M\|} (f(\omega^*) - f(\omega^m)) \geq \frac{2}{L_m} (f(\omega^*) - f(\omega^m)) \geq \|\omega^* - \omega^m\|^2 - \|\omega^* - \omega^{m-1}\|^2, \quad (3.15)$$

Summing (3.15) over $m = 1, 2, \dots, k$ gives

$$\frac{2}{\eta \|M\|} [kf(\omega^*) - \sum_{m=1}^k f(\omega^m)] = \sum_{m=1}^k \frac{2}{\eta \|M\|} (f(\omega^*) - f(\omega^m)) \geq \|\omega^* - \omega^k\|^2 - \|\omega^* - \omega^0\|^2, \quad (3.16)$$

We note that Lemma 3.2 holds for any $\omega \in R_+^{2n}$. Hence, we take $\omega = \omega^{m-1}$, and replace ω^k with ω^m in (3.7), we get

$$2L_m^{-1}(f(\omega^{m-1}) - f(\omega^m)) \geq \|\omega^m - \omega^{m-1}\|^2. \quad (3.17)$$

Since $L_m \geq \beta$ and $f(\omega^{m-1}) - f(\omega^m) \geq 0$, it follows that

$$2\beta^{-1}(f(\omega^{m-1}) - f(\omega^m)) \geq \|\omega^m - \omega^{m-1}\|^2, \quad (3.18)$$

i.e., $2\beta^{-1}[(m-1)f(\omega^{m-1}) - mf(\omega^m) + f(\omega^m)] \geq (m-1)\|\omega^m - \omega^{m-1}\|^2$. Multiplying both sides of the above inequality by $\frac{\beta}{\eta\|M\|}$ and summing over $m = 1, 2, \dots, k$, we have

$$\frac{2}{\eta\|M\|}[-kf(\omega^k) + \sum_{m=1}^k f(\omega^m)] \geq \frac{\beta}{\eta\|M\|} \sum_{m=1}^k (m-1)\|\omega^m - \omega^{m-1}\|^2. \quad (3.19)$$

Adding (3.16) and (3.19), we have

$$\frac{2k}{\eta\|M\|}(f(\omega^*) - f(\omega^k)) \geq \|\omega^* - \omega^k\|^2 + \frac{\beta}{\eta\|M\|} \sum_{m=1}^k (m-1)\|\omega^m - \omega^{m-1}\|^2 - \|\omega^* - \omega^0\|^2 \geq -\|\omega^* - \omega^0\|^2.$$

Combining this with $f(\omega^*) - f(\omega^k) \leq 0$, the desired result follows. \square

Remark 3.2. Theorem 3.1 above means that we can obtain an ϵ -optimal solution, denoted by $\hat{\omega}$, requires the number of iterations at most $\lceil c/\epsilon \rceil + 1$, such that $f(\hat{\omega}) - f(\omega^*) \leq \epsilon$, where $c := \frac{1}{2}\eta\|M\|\|\omega^0 - \omega^*\|^2$. In fact, using (3.13), and set $f(\omega^k) - f(\omega^*) \leq \frac{\eta\|M\|}{2k}\|\omega^* - \omega^0\|^2 \leq \epsilon$, then one has $k \geq c/\epsilon$, i.e., requiring the number of iterations at most $\lceil c/\epsilon \rceil + 1$.

In the following convergence analysis, we assume that Algorithm 3.1 generates an infinite sequence, and first prove the global convergence of the proposed algorithm.

Theorem 3.2. Suppose that the solution set of (2.3) is bounded. Then, the sequence ω^k generated by Algorithm 3.1 converges globally to a solution of (2.3).

Proof. By (3.6) and $f(\omega) \geq 0$, we know that the nonnegative sequence $f(\omega^k)$ is monotonically decreasing, so it converges. Combining this with (3.18), one has

$$\lim_{k \rightarrow \infty} \|\omega^k - \omega^{k-1}\| = 0. \quad (3.20)$$

Applying (3.14) and the fact $f(\omega^*) - f(\omega^k) \leq 0$, we have

$$\|\omega^* - \omega^k\| \leq \|\omega^* - \omega^{k-1}\|, \quad (3.21)$$

i.e., the nonnegative sequence $\{\|\omega^k - \omega^*\|\}$ is monotonically decreasing, so it converges. Thus, the sequence $\{\omega^k\}$ is bounded, and there exists a convergent subsequence $\{\omega^{k_i}\}$ of $\{\omega^k\}$, we assume that $\lim_{k \rightarrow \infty} \omega^{k_i} = \bar{\omega}$. Combining this with (3.20), we have

$$\lim_{k \rightarrow \infty} \|\omega^{k_i-1} - \bar{\omega}\| \leq \lim_{k \rightarrow \infty} \|\omega^{k_i} - \omega^{k_i-1}\| + \lim_{k \rightarrow \infty} \|\omega^{k_i} - \bar{\omega}\| = 0. \quad (3.22)$$

From the second equality of (3.1), we obtain $\langle \omega - \omega^{k_i}, \nabla f(\omega^{k_i-1}) + L_{k_i}(\omega^{k_i} - \omega^{k_i-1}) \rangle \geq 0, \forall \omega \in R_+^{2n}$. Combining this with (3.20) and (3.22), we have $\langle \omega - \bar{\omega}, \nabla f(\bar{\omega}) \rangle \geq 0, \forall \omega \in R_+^{2n}$, i.e., $\bar{\omega}$ is a solution of (2.3). So, the $\bar{\omega}$ can be used as ω^* to discuss in Theorem 3.1 above, we obtain that the sequence $\{\|\omega^k - \bar{\omega}\|\}$ is also convergent, combining this with $\lim_{k \rightarrow \infty} \|\omega^{k_i} - \bar{\omega}\| = 0$, one has $\lim_{k \rightarrow \infty} \|\omega^k - \bar{\omega}\| = 0$. Thus, the desired result follows. \square

Theorem 3.3. The sequence $\{x^k\}$ converges globally to a solution of (1.2), where $x^k = \mu^k - \nu^k, (\mu^k; \nu^k) = \omega^k$.

Proof. From Theorem 3.2, we know that ω^k converges globally to a solution of (2.3), say ω^* , and assume $\omega^* = (\mu^*; \nu^*)$, $x^* = \mu^* - \nu^*$. A direct computation yields that

$$\begin{aligned} \|x^k - x^*\| &= \|(\mu^k - \nu^k) - (\mu^* - \nu^*)\| \\ &\leq \|(\mu^k - \mu^*)\| + \|(\nu^k - \nu^*)\| \\ &\leq \|(\mu^k - \mu^*)\|_1 + \|(\nu^k - \nu^*)\|_1 \\ &= \|\mu^k - \mu^*; \nu^k - \nu^*\|_1 \\ &\leq \sqrt{2n} \|\mu^k - \mu^*; \nu^k - \nu^*\| \\ &= \sqrt{2n} \|\omega^k - \omega^*\| \rightarrow 0, \end{aligned} \tag{3.23}$$

as $k \rightarrow \infty$. Thus, the desired result follows. \square

In the end of this section, we discuss the convergence rate of Algorithm 3.1. To this end, the following lemmas are needed.

Lemma 3.3. *For the sequence $\{\omega^k\}$ generated by Algorithm 3.1. Then, there exists a positive constant τ such that*

$$\text{dist}(\omega^k, \Omega^*) \leq \tau r(\omega^k, L_k^{-1}), \tag{3.24}$$

where $\text{dist}(\omega^k, \Omega^*)$ denotes the distance from point ω^k to the solution set Ω^* .

Proof. By Theorem 3.2, we obtain that the sequence $\{\omega^k\}$ is bounded, i.e., there exists a constant $c_0 > 0$ such that $\|\omega^k\| \leq c_0$. The following proof uses a similar technique to that of Corollary 3.2 in [43]. \square

Lemma 3.4. *For the sequence $\{\omega^k\}$ generated by Algorithm 3.1, it holds that*

$$\text{dist}(\omega^k, \Omega^*) \leq \tau \left(1 + \frac{\|M\|}{\beta}\right) \|\omega^k - \omega^{k-1}\|, \tag{3.25}$$

where positive constant τ defined in Lemma 3.3.

Proof. By (3.1), one has

$$\omega^k - P_{R_+^{2n}} \left\{ \omega^{k-1} - \frac{1}{L_k} \nabla f(\omega^{k-1}) \right\} = 0. \tag{3.26}$$

Then, by (2.4), one has

$$\begin{aligned} r(\omega^k, \frac{1}{L_k}) &= \left\| \omega^k - P_{R_+^{2n}} \left\{ \omega^k - \frac{1}{L_k} (M\omega^k - p) \right\} \right\| \\ &= \left\| \left[\omega^k - P_{R_+^{2n}} \left\{ \omega^k - \frac{1}{L_k} (M\omega^k - p) \right\} \right] - \left[\omega^{k-1} - P_{R_+^{2n}} \left\{ \omega^{k-1} - \frac{1}{L_k} (M\omega^{k-1} - p) \right\} \right] \right\| \\ &= \left\| P_{R_+^{2n}} \left\{ \omega^k - \frac{1}{L_k} (M\omega^k - p) \right\} - P_{R_+^{2n}} \left\{ \omega^{k-1} - \frac{1}{L_k} (M\omega^{k-1} - p) \right\} \right\| \\ &\leq \left\| \left\{ \omega^k - \frac{1}{L_k} (M\omega^k - p) \right\} - \left\{ \omega^{k-1} - \frac{1}{L_k} (M\omega^{k-1} - p) \right\} \right\| \\ &\leq \|\omega^k - \omega^{k-1}\| + \frac{1}{L_k} \|M(\omega^k - \omega^{k-1})\| \\ &\leq \left(1 + \frac{1}{L_k} \|M\|\right) \|\omega^k - \omega^{k-1}\| \\ &\leq \left(1 + \frac{\|M\|}{\beta}\right) \|\omega^k - \omega^{k-1}\|, \end{aligned} \tag{3.27}$$

where the second equality follow from (3.26), the second inequality follows from Proposition 2.1. Combining (3.27) with Lemma 3.3 yields (3.25). \square

Lemma 3.5. For the sequence $\{\omega^k\}$ generated by Algorithm 3.1, it holds that

$$f(\omega^k) - f(\bar{\omega}^k) \leq \sigma \|\omega^k - \omega^{k-1}\|^2, \quad (3.28)$$

where σ is a positive constant, $\bar{\omega}^k \in \Omega^*$ is the point closest to ω^k .

Proof. By (2.2), and using the Mean Value Theorem with some n -vector z^k lying on the line segment joining $\bar{\omega}^k$ with ω^k , we have

$$\begin{aligned} f(\omega^k) - f(\bar{\omega}^k) &= (\omega^k - \bar{\omega}^k)^\top (Mz^k - p) \\ &\leq (\omega^k - \bar{\omega}^k)^\top (Mz^k - p) + (\bar{\omega}^k - \omega^k)^\top (M\omega^{k-1} - p) + L_k(\bar{\omega}^k - \omega^k)^\top (\omega^k - \omega^{k-1}) \\ &= (\omega^k - \bar{\omega}^k)^\top (Mz^k - p) + (\bar{\omega}^k - \omega^k)^\top (M\bar{\omega}^k - p) \\ &\quad + (\bar{\omega}^k - \omega^k)^\top M(\omega^{k-1} - \bar{\omega}^k) + L_k(\bar{\omega}^k - \omega^k)^\top (\omega^k - \omega^{k-1}) \\ &= (\omega^k - \bar{\omega}^k)^\top M(z^k - \bar{\omega}^k) + L_k(\bar{\omega}^k - \omega^k)^\top (\omega^k - \omega^{k-1}) \\ &\quad + (\bar{\omega}^k - \omega^k)^\top M(\omega^{k-1} - \omega^k) + (\bar{\omega}^k - \omega^k)^\top M(\omega^k - \bar{\omega}^k) \\ &\leq \|M\| \|\omega^k - \bar{\omega}^k\| \|z^k - \bar{\omega}^k\| + L_k \|\bar{\omega}^k - \omega^k\| \|\omega^k - \omega^{k-1}\| \\ &\quad + \|M\| \|\bar{\omega}^k - \omega^k\| \|\omega^{k-1} - \omega^k\| + \|M\| \|\omega^k - \bar{\omega}^k\|^2 \\ &\leq 2\|M\| \|\omega^k - \bar{\omega}^k\|^2 + (\eta + 1) \|M\| \|\bar{\omega}^k - \omega^k\| \|\omega^k - \omega^{k-1}\| \\ &= 2\|M\| \text{dist}(\omega^k, \Omega^*)^2 + (\eta + 1) \|M\| \|\omega^k - \omega^{k-1}\| \text{dist}(\omega^k, \Omega^*) \\ &\leq \left\{ 2\|M\| \left[\tau \left(1 + \frac{\|M\|}{\beta} \right) \right]^2 + (\eta + 1) \|M\| \left[\tau \left(1 + \frac{\|M\|}{\beta} \right) \right] \right\} \|\omega^k - \omega^{k-1}\|^2, \end{aligned} \quad (3.29)$$

where the first inequality follows from (3.12) with $\bar{\omega}^k \in R_+^{2n}$, and the second inequality is by the Cauchy-Schwartz inequality, the third inequality follows from the fact that $\|z^k - \bar{\omega}^k\| \leq \|\omega^k - \bar{\omega}^k\|$ and Lemma 3.1, the last inequality is by (3.25), let $\sigma = 2\|M\| \left[\tau \left(1 + \frac{\|M\|}{\beta} \right) \right]^2 + (\eta + 1) \|M\| \left[\tau \left(1 + \frac{\|M\|}{\beta} \right) \right]$, and thus, the desired result follows. \square

Theorem 3.4. The generated sequence $\{\omega^k\}$ by Algorithm 3.1 converges global linearly to an element of Ω^* .

Proof. For $\omega^*, \bar{\omega}^k \in \Omega^*$, by (2.2), one has $f(\omega^*) = f(\bar{\omega}^k)$. Combining this with (3.28) and (3.18), we obtain

$$\begin{aligned} f(\omega^k) - f(\omega^*) &= f(\omega^k) - f(\bar{\omega}^k) \\ &\leq \sigma \|\omega^k - \omega^{k-1}\|^2 \\ &\leq \frac{2\sigma}{\beta} (f(\omega^{k-1}) - f(\omega^k)) \\ &= \frac{2\sigma}{\beta} (f(\omega^{k-1}) - f(\omega^*)) - \frac{2\sigma}{\beta} (f(\omega^k) - f(\omega^*)), \end{aligned}$$

i.e.,

$$f(\omega^{k+1}) - f(\omega^*) \leq \varrho (f(\omega^k) - f(\omega^*)), \quad (3.30)$$

where $0 < \varrho := \frac{2\sigma}{1 + \frac{2\sigma}{\beta}} < 1$. Since $f(\omega^k) - f(\omega^*) > 0$, $f(\omega^{k+1}) - f(\omega^*) > 0$, so (3.30) implies that $\{f(\omega^k)\}$ converges global linearly to $f(\omega^*)$.

By (3.18) and (3.30), we obtain

$$\begin{aligned} \|\omega^k - \omega^{k-1}\|^2 &\leq \frac{2}{\beta}(f(\omega^k) - f(\omega^{k-1})) \\ &\leq \frac{2}{\beta}(f(\omega^k) - f(\omega^*)) \\ &\leq \frac{2}{\beta} \varrho (f(\omega^{k-1}) - f(\omega^*)) \\ &\leq \dots\dots\dots \\ &\leq \left[\frac{2}{\beta}(f(\omega^0) - f(\omega^*))\right] \varrho^k. \end{aligned}$$

Combining this with (3.25), we obtain

$$\begin{aligned} \text{dist}(\omega^k, \Omega^*) &\leq \tau(1 + \frac{\|M\|}{\beta}) \|\omega^k - \omega^{k-1}\| \\ &\leq \left\{ \tau(1 + \frac{\|M\|}{\beta}) \sqrt{\frac{2}{\beta}(f(\omega^0) - f(\omega^*))} \right\} \sqrt{\varrho^k}, \end{aligned}$$

where $0 < \sqrt{\varrho} < 1$. Thus, the sequence $\{\omega^k\}$ converges global linearly an element of Ω^* . \square

Remark 3.3. Riahi et al. (Theorem 2 [44]) give a linear rate of convergence of the Fletcher-Reeves' nonlinear conjugate gradient method (NCG), satisfying strong Wolfe conditions when the Hessian matrix of the cost function is positive definite and continuous and has bounded eigenvalues, which is the first result with a linear rate of convergence reported for such NCG in a general framework. In this paper, we propose a gradient projection method with global R-linear rate convergence. Our method is very different because first, the iterative method and line search step of the proposed algorithm are different from that in [44], and second because the Hessian matrix of objective function in (2.2) is semi-positive definite instead of positive definite. In fact, since $m \ll n$ in (1.1), we know that the rank of the sensing matrix A is less than m , then the matrix $A^T A$ is semi-positive definite instead of positive definite, and according to (2.2), one has

$$\begin{pmatrix} I_n & 0 \\ I_n & I_n \end{pmatrix} M = \begin{pmatrix} A^T A & -A^T A \\ 0 & 0 \end{pmatrix},$$

it follows that $\det(M) = 0$. Hence, the desired result follows. Finally because the proof method of linear convergence in this paper is different from that of Theorem 2 in [44].

Thus, the global R-linear rate convergence of the proposed Algorithm 3.1 is a new result for constrained convex (rather than strictly convex) quadratic programming problem.

4. Numerical experiments

In this section, some numerical experiments on sparse signal recovery were provided to prove the efficiency of the proposed method. All codes are written by version of Matlab 9.20.538062 and performed on a Windows 10 PC with Intel(R) Core(TM) i3-10105F CPU @ 3.70GHz 3.70 GHz and 16GB of memory. For experiments below, we set the matrix A is generated by Matlab scripts:

$$[Q, R] = \text{qr}(A', \emptyset); A = Q'.$$

The original signal \bar{x} is generated by

$$p = \text{randperm}(n); \quad x(p(1:k)) = \text{randn}(k, 1).$$

Then, the observed signal is $y = A\bar{x} + \bar{n}$, where \bar{n} is generated by a standard Gaussian distribution $N(0, 1)$ and then it is normalized. The initial points $\omega^0 = (\mu^0; \nu^0)$, where $\mu^0 = \max\{0, A^\top y\}$, $\nu^0 = \max\{0, -A^\top y\}$.

The stop criterion is

$$\frac{\|f_k - f_{k-1}\|}{\|f_{k-1}\|} < 10^{-5},$$

where f_k denotes the objective value of (2.2) at iteration ω^k .

In our experiment, we assess the quality of restoration by mean of squared error (MSE) to the original signal \bar{x} , i.e.,

$$MSE = \frac{1}{n} \|\bar{x} - x^*\|^2$$

where x^* denotes the recovery signal.

We compare the proposed Algorithm 3.1 with several state-of-the-art algorithms, all of which were the most popular in iterative algorithms for solving the ℓ_1 -norm regularization problem arising from compressive sensing. In Section 4.1 and 4.2, we compare the proposed Algorithm 3.1 with PCG [30]—a projection conjugate gradient algorithm for monotone nonlinear equations with convex constraints, and GCQ [39]—a CQ algorithm for splitting feasible problems. In Section 4.3, we compare the proposed Algorithm 3.1 with FISTA [15]—a fast iterative shrinkage-thresholding algorithm for linear inverse problems, and PADM [11]—a primal alternating direction method for ℓ_1 -problems in compressive sensing. In Section 4.4, we compare the proposed Algorithm 3.1 with PCG for deblurring image. In Section 4.5, we present convergence curve of the proposed Algorithm 3.1.

4.1. Test on additive Gaussian white noise

In this subsection, we will illustrate the feasibility and effectiveness of the proposed Algorithm 3.1 by recovering sparse signal of which observation data is corrupted by additive Gaussian white noise, and give some comparisons with PCG and GCQ.

We set $n = 2^{13}$, $m = 2^{11}$, $k = 2^9$, and the parameters in the three tested algorithms are listed as follows:

$$\begin{aligned} \text{Algorithm 3.1} : & \quad \beta = 0.6, \gamma = 0.5, \eta = 1.1; \\ \text{GCQ} : & \quad \beta = 0.6, \gamma = 0.5, \eta = 1.1; \\ \text{PCG} : & \quad \xi = 1, \rho = 0.55, \gamma = 0.1. \end{aligned}$$

The original signal, the measurement and the reconstructed signal (marked by red point) by Algorithm 3.1, GCQ and PCG are given in Figure 1.

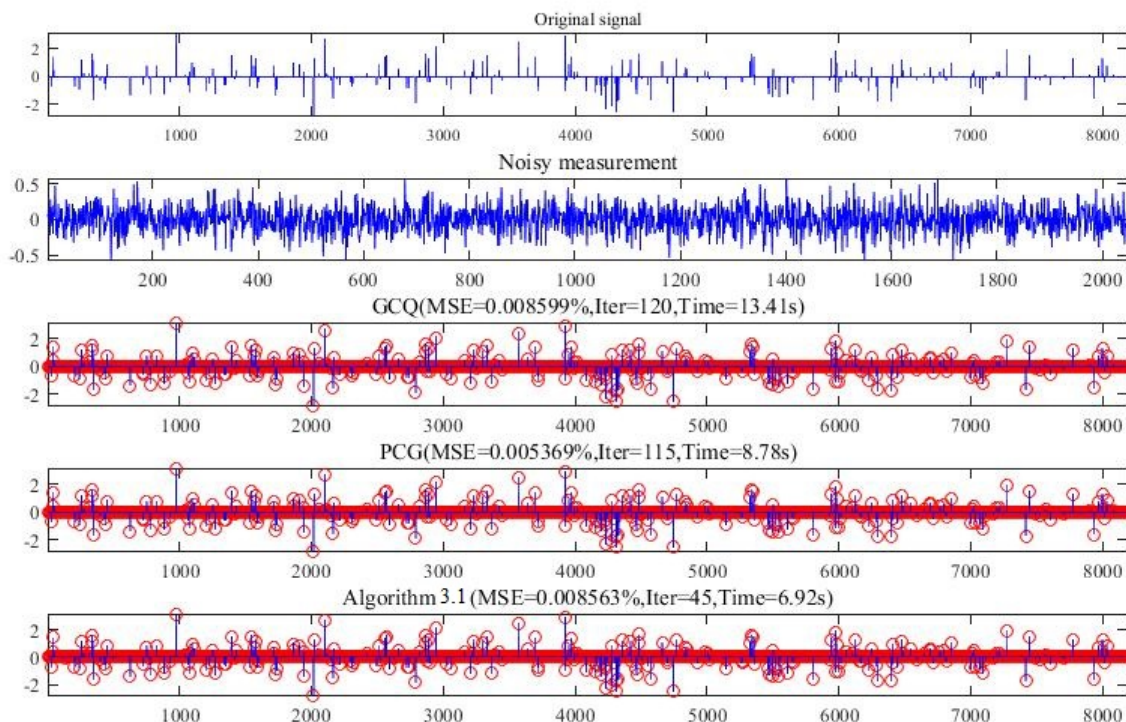


Figure 1. Compare the signal restoration results of the three algorithms.

Comparing the first, third, fourth and the last plots in Figure 1, all elements in the original signal are circled by the red points, which indicates that the three methods can recover the original signal quite well. Furthermore, we record the number of iterations (abbreviated as Iter), the CPU time in seconds (abbreviated as CPUTime) and MSE of three methods. Figure 1 indicates that Algorithm 3.1 is always faster than GCQ and PCG methods, and also have higher accuracy. Thus, Algorithm 3.1 is an efficient method for sparse signal recovery.

4.2. Comparison with GCQ and PCG in different dimension

In this subsection, we compare MSE, CPUTime and Iter among Algorithm 3.1, GCQ and PCG in different dimension n , where some parameters in the tested algorithms are given above. We set $m = \frac{n}{4}, k = \frac{m}{8}$. The following experiments is divided into two cases: adding noise and not adding noise.

The numerical results are listed in Table 1. From Table 1, whether it is Free noise or Gaussian noise, we can see that the CPU time of Algorithm 3.1 are always less than that of both GCQ and PCG in different dimensions except for $n = 1024$, which shows that Algorithm 3.1 converges faster than both GCQ and PCG. In the not noise case, the MSE of Algorithm 3.1 is almost smaller than that of GCQ. In other cases, although the MSE of Algorithm 3.1 is occasionally bigger than that of GCQ or PCG, the difference between them is very small. Specially, as the dimensionality of the recovered signal increases, the recovery speed advantage of Algorithm 3.1 becomes more obvious.

Table 1. Compare Iter, CPUTime and MSE among Algorithm 3.1, GCQ and PCG in different dimension n ($k = \frac{n}{32}$).

Dimension	Tested method	NO-NOISE			NOISE		
		Iter	CPUTime	MSE	Iter	CPUTime	MSE
1024	Algorithm 3.1	40	0.1406	2.33e-04	58	0.1606	2.75e-04
	GCQ	106	0.2031	2.35e-04	153	0.2500	2.73e-04
	PCG	117	0.1250	1.71e-04	197	0.1875	1.13e-04
2048	Algorithm 3.1	52	0.5781	1.77e-04	58	0.6469	1.90e-04
	GCQ	135	1.0781	1.79e-04	130	1.1094	1.90e-04
	PCG	128	0.6250	1.12e-04	163	0.7344	8.52e-05
3072	Algorithm 3.1	45	1.2375	1.59e-04	45	1.4031	1.53e-04
	GCQ	147	2.6563	1.59e-04	122	2.8500	1.55e-04
	PCG	141	2.1656	9.65e-05	111	2.6094	1.77e-04
4096	Algorithm 3.1	41	1.9844	1.40e-04	52	2.4688	1.33e-04
	GCQ	116	4.0313	1.40e-04	138	4.5000	1.34e-04
	PCG	145	4.1719	8.27e-05	150	4.9063	7.68e-05
5120	Algorithm 3.1	54	3.5156	1.27e-04	57	3.3281	1.18e-04
	GCQ	140	7.4375	1.27e-04	145	6.9531	1.19e-04
	PCG	100	4.1250	1.81e-04	160	6.1094	5.73e-05
6144	Algorithm 3.1	43	3.4844	1.16e-04	56	5.1250	1.07e-04
	GCQ	117	7.8281	1.17e-04	125	10.1250	1.07e-04
	PCG	121	6.6250	7.42e-05	145	8.6406	6.05e-05
7168	Algorithm 3.1	45	5.1500	9.65e-05	49	5.7313	9.94e-05
	GCQ	120	10.9531	9.74e-05	128	11.1094	9.03e-05
	PCG	117	7.3438	8.07e-05	155	8.0938	5.26e-05
8192	Algorithm 3.1	43	6.1563	8.56e-05	45	6.9844	8.56e-05
	GCQ	115	12.1250	8.60e-05	120	14.6406	8.60e-05
	PCG	110	8.6563	5.37e-05	115	9.5875	5.37e-05
9216	Algorithm 3.1	50	8.8281	8.57e-05	55	9.5469	8.76e-05
	GCQ	122	16.3750	8.39e-05	132	18.3906	8.84e-05
	PCG	128	11.4219	5.36e-05	128	12.3750	5.96e-05
10240	Algorithm 3.1	47	10.7500	8.01e-05	47	11.8456	8.95e-05
	GCQ	122	24.3750	8.86e-05	132	28.3906	8.84e-05
	PCG	117	14.8281	4.99e-05	117	15.8125	4.86e-05

4.3. Comparison with FISTA and PADM in different dimension

In this subsection, we present comparison results of the proposed Algorithm 3.1 and FISTA [15], PADM [11] on Gaussian noise, where some parameters in FISTA and PADM are listed as follows:

$$FISTA : \beta = 0.6, \eta = 6, \sigma = 0.01, \gamma = 0.5;$$

$$PADM : \beta = 1.5, \alpha = 1.5, \tau = 2,$$

some parameters in the proposed Algorithm 3.1 are given Subsection 4.1. Comparison with Subsection 4.2, The sparse signal is double that of Subsection 4.2, i.e., $m = \frac{n}{4}, k = \frac{m}{4}$.

The numerical results are listed in Table 2. From Table 2, we obtain that the CPU time of Algorithm 3.1 are obviously less than that of FISTA and PADM in different k-Sparse signal, which shows that the recovery speed of Algorithm 3.1 is faster than that of FISTA and PADM, respectively. The MSE of Algorithm 3.1 is smaller than that of FISTA. Although the MSE of algorithm 3.1 is larger than that of PADM, the difference between the two is only slightly. Especially, the recovery speed advantage of Algorithm 3.1 becomes more obvious with the increase of dimension. All of these indicate that Algorithm 3.1 is an efficient method for different k-Sparse signal recovery.

Table 2. Compare Iter, CPUTime and MSE among Algorithm 3.1, FISTA and PADM in different dimension n ($k = \frac{n}{16}$).

Dimension	k -Sparse signal	Tested method	Iter	CPUTime	MSE
3072	192	Algorithm 3.1	92	1.2281	1.49e-04
		FISTA	89	2.0625	1.51e-04
		PADM	100	1.2500	1.47e-04
4096	256	Algorithm 3.1	90	1.9063	1.32e-04
		FISTA	87	3.3125	1.34e-04
		PADM	103	2.4063	1.31e-04
5120	320	Algorithm 3.1	99	2.7969	1.18e-04
		FISTA	97	5.1563	1.19e-04
		PADM	112	4.0156	1.17e-04
6144	384	Algorithm 3.1	88	3.5000	9.87e-05
		FISTA	86	6.6406	9.90e-05
		PADM	102	5.3125	9.73e-05
7168	448	Algorithm 3.1	89	4.2500	9.33e-05
		FISTA	87	8.3125	9.36e-05
		PADM	102	6.8906	9.17e-05
8192	512	Algorithm 3.1	94	6.5625	9.00e-05
		FISTA	91	11.2031	9.03e-05
		PADM	107	9.3906	8.38e-05

4.4. Noisy image restoration

In this subsection, we will use the proposed Algorithm 3.1 and PCG to deblur image with salt-and-pepper noise. Some parameters in tested algorithms are given above. We choose Lena (512×512),

Barbara (512×512) and Baboon (512×512) as the test images, and assess the restoration performance by applying the peak signal to noise ratio (PSNR), that is

$$PSNR = 10 \times \log_{10} \frac{(N-1)^2}{\frac{1}{MN} \sum_{i,j} (u_{i,j}^* - x_{i,j})^2}.$$

For three different test images, we use different noise levels for numerical comparison to verify the resilience of the Algorithm 3.1 and PCG, and calculate numerical result for each noise sample of each image for each noise level. The detailed numerical results are listed in Table 3, where r denotes the noise level, where $r = 1$ means that the picture is all noise. Iter, CPUTime and PSNR denote numerical results for the proposed Algorithm 3.1 denoising, Iter_PCG, CPUTime_PCG and PSNR_PCG denote numerical results for PCG denoising, PSNR_Noise denotes peak signal to noise ratio with noise.

Table 3. Algorithm 3.1 and PCG restore images with different salt and pepper noise $r = 0.1, 0.3, 0.5, 0.7$ and different test images.

r	Image	Iter/Iter_PCG	CPUTime/CPUTime_PCG	PSNR_Noise	PSNR/PSNR_PCG
0.1					
	Lena	1/13	0.7188/2.7344	15.3759	48.2876 / 45.0947
	Barbaba	1/13	0.5938/2.4063	15.2665	39.2428 / 34.8734
	Baboon	2/13	1.0781/2.2969	15.5923	37.1912 / 32.9240
0.3					
	Lena	1/20	0.7813/3.0625	10.5662	41.0399 / 40.5378
	Barbaba	2/13	1.3125/3.5938	10.5093	33.4399 / 32.4564
	Baboon	3/19	1.2656/5.0469	10.7793	31.5139 / 30.5613
0.5					
	Lena	3/11	1.5000/3.0469	8.3642	37.0211 / 36.9201
	Barbaba	2/8	1.8125/3.5469	8.2810	30.3528 / 30.1411
	Baboon	5/15	2.1250/4.2031	8.5936	28.4595 / 28.2282
0.7					
	Lena	1/9	1.3750/3.6875	6.9164	33.5518 / 33.5431
	Barbaba	2/8	1.6250/3.0781	6.8222	27.9891 / 27.9719
	Baboon	4/8	2.2500/3.5005	7.1119	26.0362 / 26.0117

From Table 3, using different salt-and-pepper noise level $r = 0.1, 0.3, 0.5, 0.7$, we can see that the Algorithm 3.1 and PCG could restore the image very well. Compared with PCG, the Algorithm 3.1 not only has a faster denoising speed, but also a higher PSNR, which means the effective of Algorithm 3.1 to remove salt and pepper noise.

Figure 2 shows the restoration results obtained by using the tested methods to the test images corrupted with 30%. From Figure 2, we could obtain that the Algorithm 3.1 and PCG could restore the image with salt-and-pepper noise very well.

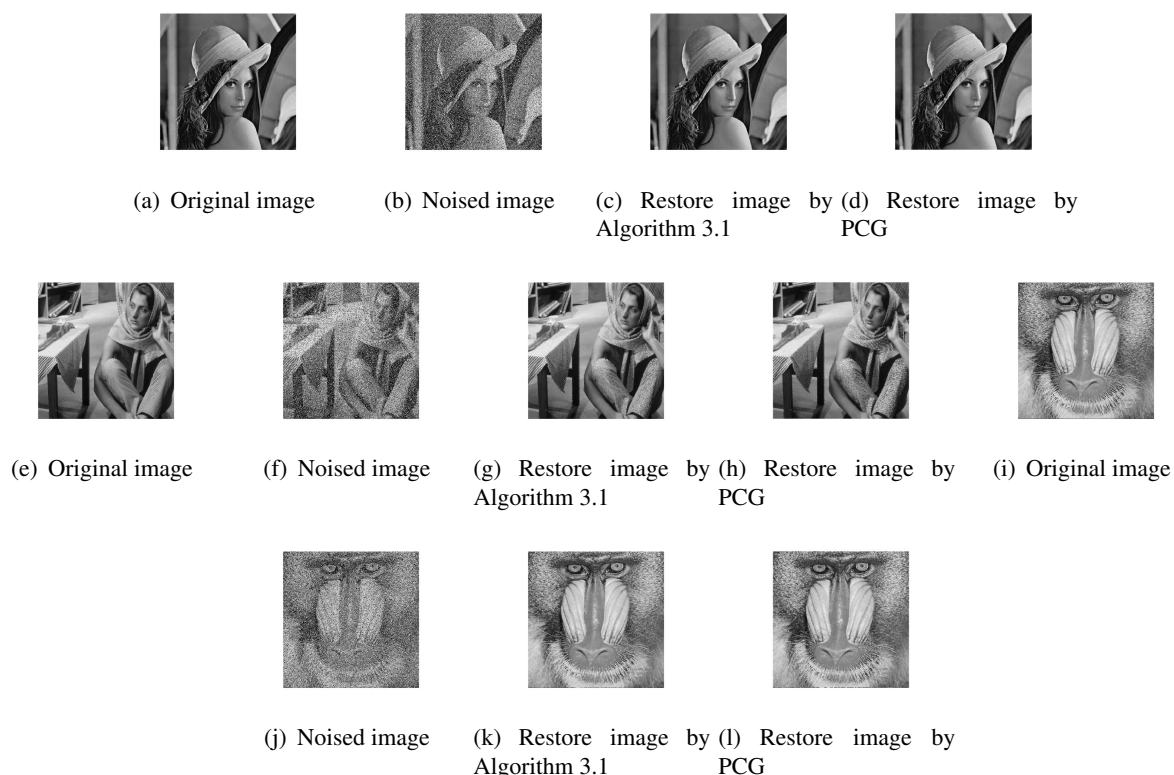


Figure 2. Algorithm 3.1 and PCG restore images with the noise level 0.3 (Lena, Barbaba and Baboon).

4.5. Convergence curve with different dimension

In this subsection, we respectively draw convergence curve of the proposed Algorithm 3.1 for $n = 4096, 6144, 8192$, where $m = \frac{n}{4}, k = \frac{m}{4}$. The experiments divided into two cases: adding Gaussian noise and not adding Gaussian noise. The numerical results are drawn in Figure 3. As can be seen from Figure 3, $\|x^k - x^*\|$ is approximately an exponential function of the number of iterations k whether it is free noise or Gaussian noise, i.e., $\|x^k - x^*\| \approx c_0 a^k$ where c_0 and a are two positive constants, and $0 < a < 1$, which means R -linearly convergence behavior of proposed algorithm.

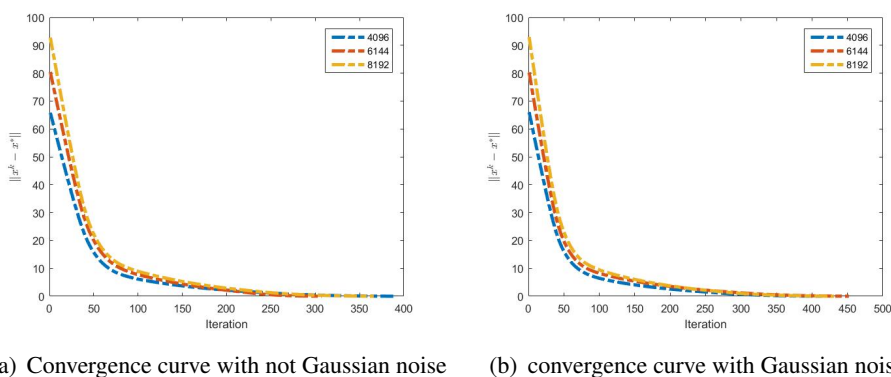


Figure 3. Convergence curve of Algorithm 3.1 with different dimensions.

5. Discussion

This work has possible extensions. First, the parameters $\beta = 0.6, \gamma = 0.5$ of Algorithm 3.1 is adjusted dynamically to further enhance the efficiency of the corresponding method. Second, Riahi [45] present a new steepest-descent type algorithm for the optimization of a positive definite quadratic form, which can effectively handle ill-conditioned quadratic programming problem. Motivated by this technique, we will investigate the feasibility of applying the algorithm presented by Riahi to (2.2) with ill-conditioned the matrix A and the semi-positive definite Hessian matrix. Finally, since the regularized nuclear norm minimization (RNNM) model defined in [46, 47] is a convex program, we explore the possibility of the proposed algorithm developed for (1.2) model to solve the RNNM model from theoretical results and numerical experiments. These will be our further research directions.

6. Conclusions

In this paper, we propose a new self-adaptive gradient projection algorithm for solving sparse signal reconstruction problem in compressive sensing. Its global convergence is established in detail. Without any assumptions, we establish global R -linear convergence rate, which is a new result for constrained convex (rather than strictly convex) quadratic programming problem, and also obtain an approximate optimal solution in a finite number of iterations. Thus, it has fast solving speed and pinpoint accuracy when the dimension increases. It makes the proposed algorithm very attractive for solving large-scale problems. We present numerical a comparison of five tested methods on sparse signal recovery. The proposed method is competitive with other existing ones in terms of recovery speed and computational efficiency.

Acknowledgments

This study was funded by the Natural Science Foundation of China (Nos. 12071250, 11801309), and Shandong Provincial Natural Science Foundation (ZR2021MA088).

Conflict of interest

The authors declare no conflict of interest.

References

1. E. J. Candès, J. Romberg, T. Tao, Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information, *IEEE T. Inform. Theory*, **52** (2006), 489–509. <https://doi.org/10.1109/TIT.2005.862083>
2. E. J. Candès, M. B. Wakin, An introduction to compressive sampling, *IEEE Signal Proc. Mag.*, **25** (2008), 21–30. <https://doi.org/10.1109/MSP.2007.914731>
3. D. L. Donoho, For most large underdetermined systems of equations, the minimal ℓ_1 -norm near-solution approximates the sparsest near-solution, *Commun. Pur. Appl. Math.*, **59** (2006), 907–934. <https://doi.org/10.1002/cpa.20131>

4. B. K. Natarajan, Sparse approximate solutions to linear systems, *SIAM J. Comput.*, **24** (1995), 227–234. <https://doi.org/10.1137/S0097539792240406>
5. S. S. Chen, D. L. Donoho, M. A. Saunders, Automatic decomposition by basis pursuit, *SIAM Rev.*, **43** (2001), 129–159. <https://doi.org/10.1137/S003614450037906X>
6. S. J. Kim, K. Koh, M. Lustig, S. Boyd, D. Gorinevsky, An interior-point method for large-scale ℓ_1 -regularized least squares, *IEEE J-STSP*, **1** (2007), 606–617. <https://doi.org/10.1109/JSTSP.2007.910971>
7. M. A. T. Figueiredo, R. D. Nowak, S. J. Wright, Gradient projection for sparse reconstruction: Application to compressed sensing and other inverse problems, *IEEE J-STSP*, **1** (2007), 586–597. <https://doi.org/10.1109/JSTSP.2007.910281>
8. Y. H. Dai, Y. K. Huang, X. W. Liu, A family of spectral gradient methods for optimization, *Comput. Optim. Appl.*, **74** (2019), 43–65. <https://doi.org/10.1007/s10589-019-00107-8>
9. S. Huang, Z. Wan, A new nonmonotone spectral residual method for nonsmooth nonlinear equations, *J. Comput. Appl. Math.*, **313** (2017), 82–101. <https://doi.org/10.1016/j.cam.2016.09.014>
10. L. Zheng, L. Yang, Y. Liang, A conjugate gradient projection method for solving equations with convex constraints, *J. Comput. Appl. Math.*, **375** (2020), 112781. <https://doi.org/10.1016/j.cam.2020.112781>
11. J. F. Yang, Y. Zhang, Alternating direction algorithms for ℓ_1 -problems in compressive sensing, *SIAM J. Sci. Comput.*, **33** (2011), 250–278. <https://doi.org/10.1137/090777761>
12. I. Daubechies, M. Defrise, C. D. Mol, An iterative thresholding algorithm for linear inverse problems with a sparsity constraint, *Commun. Pur. Appl. Math.*, **57** (2004), 1413–1457. <https://doi.org/10.1002/cpa.20042>
13. M. A. T. Figueiredo, R. D. Nowak, An EM algorithm for wavelet-based image restoration, *IEEE T. Image Process.*, **12** (2003), 906C916. <https://doi.org/10.1109/TIP.2003.814255>
14. E. T. Hale, W. T. Yin, Y. Zhang, Fixed-point continuation for ℓ_1 -Minimization: Methodology and convergence, *SIAM J. Optim.*, **19** (2008), 1107–1130. <https://doi.org/10.1137/070698920>
15. A. Beck, M. Teboulle, A fast iterative shrinkage-thresholding algorithm for linear inverse problems, *SIAM J. Imaging Sci.*, **2** (2009), 183–202. <https://doi.org/10.1137/080716542>
16. J. M. Bioucas-Dias, M. A. T. Figueiredo, A new TwIst: Two-step iterative shrinkage/thresholding algorithm for image restoration, *IEEE T. Image Process.*, **16** (2007), 2992–3004. <https://doi.org/10.1109/TIP.2007.909319>
17. P. L. Combettes, J. C. Pesquet, Proximal thresholding algorithm for minimization over orthonormal bases, *SIAM J. Optim.*, **18** (2007), 1351–1376. <https://doi.org/10.1137/060669498>
18. E. van den Berg, M. P. Friedlander, Probing the Pareto frontier for basis pursuit solutions, *SIAM J. Sci. Comput.*, **31** (2008), 890–912. <https://doi.org/10.1137/080714488>
19. S. Becker, J. Bobin, E. J. Cands, NESTA: A fast and accurate first-order method for sparse recovery, *SIAM J. Imaging Sci.*, **4** (2011), 1–39. <https://doi.org/10.1137/090756855>
20. S. J. Wright, R. D. Nowak, M. A. T. Figueiredo, Sparse reconstruction by separable approximation, *IEEE Trans. Signal Proces.*, **57** (2009), 2479–2493. <https://doi.org/10.1109/TSP.2009.2016892>

21. N. Keskar, J. Nocedal, F. Oztoprak, A. Waechter, A second-order method for convex ℓ_1 -regularized optimization with active-set prediction, *Optim. Metod. Softw.*, **31** (2016), 605–621. <https://doi.org/10.1080/10556788.2016.1138222>
22. X. T. Xiao, Y. F. Li, Z. W. Wen, L. W. Zhang, Semi-smooth second-order type methods for composite convex programs, *arXiv:1603.07870v2 [math.OC]*, 2016. <https://doi.org/10.48550/arXiv.1603.07870>
23. A. Milzarek, M. Ulbrich, A semismooth Newton method with multidimensional filter globalization for l_1 -optimization, *SIAM J. Optim.*, **24** (2014), 298–333. <https://doi.org/10.1137/120892167>
24. R. H. Byrd, J. Nocedal, F. Oztoprak, An inexact successive quadratic approximation method for L_1 regularized optimization, *Math. Program.*, **157** (2016), 375–396. <https://doi.org/10.1007/s10107-015-0941-y>
25. Y. H. Xiao, H. Zhu, A conjugate gradient method to solve convex constrained monotone equations with applications in compressive sensing, *J. Math. Anal. Appl.*, **405** (2013), 310–319. <https://doi.org/10.1016/j.jmaa.2013.04.017>
26. M. Sun, M. Y. Tian, A class of derivative-free CG projection methods for nonsmooth equations with an application to the LASSO problem, *B. Iran. Math. Soc.*, **46** (2020), 183–205. <https://doi.org/10.1007/s41980-019-00250-2>
27. H. C. Sun, M. Sun, B. H. Zhang, An inverse matrix-free proximal point algorithm for compressive sensing, *ScienceAsia*, **44** (2018), 311–318. <https://doi.org/10.2306/scienceasia1513-1874.2018.44.311>
28. D. X. Feng, X. Y. Wang, A linearly convergent algorithm for sparse signal reconstruction, *J. Fix. Point Theory Appl.*, **20** (2018), 154. <https://doi.org/10.1007/s11784-018-0635-1>
29. Y. H. Xiao, Q. Y. Wang, Q. J. Hu, Non-smooth equations based method for ℓ_1 -norm problems with applications to compressed sensing, *Nonlinear Anal.*, **74** (2011), 3570–3577. <https://doi.org/10.1016/j.na.2011.02.040>
30. J. K. Liu, S. J. Li, A projection method for convex constrained monotone nonlinear equations with applications, *Comput. Math. Appl.*, **70** (2015), 2442–2453. <https://doi.org/10.1016/j.camwa.2015.09.014>
31. J. K. Liu, Y. M. Feng, A derivative-free iterative method for nonlinear monotone equations with convex constraints, *Numer. Algorithms*, **82** (2019), 245–262. <https://doi.org/10.1007/s11075-018-0603-2>
32. Y. J. Wang, G. L. Zhou, L. Caccetta, W. Q. Liu, An alternative lagrange-dual based algorithm for sparse signal reconstruction, *IEEE Trans. Signal Proces.*, **59** (2011), 1895–1901. <https://doi.org/10.1109/TSP.2010.2103066>
33. G. Landi, A modified Newton projection method for ℓ_1 -regularized least squares image deblurring, *J. Math. Imaging Vis.*, **51** (2015), 195–208. <https://doi.org/10.1007/s10851-014-0514-3>
34. B. Xue, J. K. Du, H. C. Sun, Y. J. Wang, A linearly convergent proximal ADMM with new iterative format for BPDN in compressed sensing problem, *AIMS Mathematics*, **7** (2022), 10513–10533. <https://doi.org/10.3934/math.2022586>

35. H. J. He, D. R. Han, A distributed Douglas-Rachford splitting method for multi-block convex minimization problems, *Adv. Comput. Math.*, **42** (2016), 27–53. <https://doi.org/10.1007/s10444-015-9408-1>
36. M. Sun, J. Liu, A proximal Peaceman-Rachford splitting method for compressive sensing, *J. Appl. Math. Comput.*, **50** (2016), 349–363. <https://doi.org/10.1007/s12190-015-0874-x>
37. B. S. He, F. Ma, X. M. Yuan, Convergence study on the symmetric version of ADMM with larger step sizes, *SIAM J. Imaging Sci.*, **9** (2016), 1467–1501. <https://doi.org/10.1137/15M1044448>
38. H. J. He, C. Ling, H. K. Xu, An implementable splitting algorithm for the ℓ_1 -norm regularized split feasibility problem, *J. Sci. Comput.*, **67** (2016), 281–298. <https://doi.org/10.1007/s10915-015-0078-4>
39. B. Qu, N. H. Xiu, A note on the CQ algorithm for the split feasibility problem, *Inverse Probl.*, **21** (2005), 1655–1665. <https://doi.org/10.1088/0266-5611/21/5/009>
40. E. H. Zarantonello, Projections on convex sets in Hilbert space and spectral theory, In: *Contributions to Nonlinear Functional Analysis*, New York: Academic Press, 1971. <https://doi.org/10.1016/B978-0-12-775850-3.50013-3>
41. M. A. Noor, General variational inequalities, *Appl. Math. Lett.*, **1** (1988), 119–121. [https://doi.org/10.1016/0893-9659\(88\)90054-7](https://doi.org/10.1016/0893-9659(88)90054-7)
42. J. M. Ortega, W. C. Rheinboldt, Iterative solution of nonlinear equations in several variables, *Classics Appl. Math.*, 2000. <https://doi.org/10.1137/1.9780898719468>
43. N. H. Xiu, J. Z. Zhang, Global projection-type error bound for general variational inequalities, *J. Optim. Theory Appl.*, **112** (2002), 213–228. <https://doi.org/10.1023/a:1013056931761>
44. M. K. Riahi, I. A. Qattan, On the convergence rate of Fletcher-Reeves nonlinear conjugate gradient methods satisfying strong Wolfe conditions: Application to parameter identification in problems governed by general dynamics, *Math. Method Appl. Sci.*, **45** (2022), 3644–3664. <https://doi.org/10.1002/mma.8009>
45. M. K. Riahi, A new approach to improve ill-conditioned parabolic optimal control problem via time domain decomposition, *Numer. Algorithms*, **3** (2016), 635–666. <https://doi.org/10.1007/s11075-015-0060-0>
46. E. J. Candès, Y. Plan, Tight oracle inequalities for low-rank matrix recovery from a minimal number of noisy random measurements, *IEEE Trans. Inform. Theory*, **57** (2011), 2342–2359. <https://doi.org/10.1109/TIT.2011.2111771>
47. W. D. Wang, F. Zhang, J. J. Wang, Low-rank matrix recovery via regularized nuclear norm minimization, *Appl. Comput. Harmon. Anal.*, **54** (2021), 1–19. <https://doi.org/10.1016/j.acha.2021.03.001>

