Circuits Syst Signal Process (2017) 36:4022–4049
DOI 10.1007/s00034-017-0498-5

A Primal Douglas–Rachford Splitting Method for the Constrained Minimization Problem in Compressive Sensing

Yongchao Yu[1,2] · **Jigen Peng**[1,2] · **Xuanli Han**[1] · **Angang Cui**[1]

Received: 17 May 2016 / Revised: 15 January 2017 / Accepted: 17 January 2017 /
Published online: 31 January 2017

Abstract Compressive sensing has achieved great success in many scientific research fields. It has revealed that sparse signals can be stably recovered from a small number of noisy measurements by solving the constrained convex ℓ_1-minimization problem. In practice, a faster algorithm for solving this optimization problem is the key to compressive sensing. The Douglas–Rachford splitting method is a well-known operator splitting method that has been widely applied for solving a certain class of convex composite problems. In particular, its dual application results in the popular alternating direction method of multipliers (ADMM). In this paper, we reformulate the constrained convex ℓ_1-minimization problem as a convex composite problem with a special structure and then apply the primal Douglas–Rachford splitting method to solve it. The computational cost of the developed algorithm in each iteration is dominated by the projection onto the constraint set. A fast and efficient method of computing the projection is proposed. Numerical results show that the developed algorithm performs better than the popular NESTA and LADMM (inexact ADMM) in terms of accuracy and run time for large-scale sparse signal recovery.

✉ Jigen Peng
jgpengxjtu@126.com

Yongchao Yu
sparseelad@126.com

Xuanli Han
hanxuanli@xust.edu.cn

Angang Cui
cuiangang@163.com

[1] School of Mathematics and Statistics, Xi'an Jiaotong University, Xi'an 710049, China

[2] Beijing Center for Mathematics and Information Interdisciplinary Sciences (BCMIIS), Beijing 100048, China

Keywords Compressive sensing · Sparse signal recovery · Douglas–Rachford splitting method

1 Introduction

Over the past decade, compressive sensing (CS), first introduced by Donoho [17], Candès et al. [11,12], has established itself as a rapidly growing research area with wide applications in signal and image processing [28], medicine [40], astronomy [7], seismology [35], and so on. Indeed, the original work [11] on compressive sensing was motivated by medical imaging. The application of compressive sensing techniques to magnetic resonance imaging (MRI) was investigated in [40,44,58], and its application to the related problem of nuclear magnetic resonance spectroscopy was also studied in [36,56]. It is well known that the single-pixel camera [26] was designed to verify that the underlying idea of compressive sensing can be implemented in hardware. Applications of compressive sensing to radar, sonar and the channel estimation problem in wireless communications can also be found in [37,51,53]. Compressive sensing is closely related to sampling theory [5], error correction [13] and high-dimensional statistics [6]. Additional applications of compressive sensing include hyperspectral imaging [52], analog-to-digital conversion [45,57] and low-rank matrix recovery [14,32]. We refer the reader to two recent monographs [29,31] for further details on compressive sensing and its applications.

The fundamental problem in compressive sensing is to recover a high-dimensional sparse signal from a small number of linear measurements. We consider linear measurements of a sparse signal $\bar{x} \in \mathbb{R}^n$:

$$b = A\bar{x}, \tag{1}$$

where $A \in \mathbb{R}^{m\times n}$ ($m \ll n$) is a known sensing matrix. A naïve idea is to recover the sparse signal $\bar{x}$ from the available measurements b by solving the ℓ_0 minimization problem:

$$(\mathrm{P}_0) \quad \min_{x\in\mathbb{R}^n} \{||x||_0 : b = Ax\}. \tag{2}$$

In (P_0), $||x||_0$ denotes the number of nonzero entries of x and is often called the ℓ_0-'norm' of x. (P_0) is a non-convex problem and is generally NP-hard. A widely studied problem is its convex relaxation:

$$(\mathrm{BP}) \quad \min_{x\in\mathbb{R}^n} \{||x||_1 : b = Ax\}, \tag{3}$$

which is known as basis pursuit (BP) [10]. Compressive sensing has revealed that under certain conditions with regard to the sensing matrix A, the minimizer of (BP) is identical to that of (P_0), and thus, (BP) can exactly recover a sparse signal.

In practice, measurements are often perturbed by noise, i.e.,

$$b = A\bar{x} + e, \tag{4}$$

where $e \in \mathbb{R}^m$ is a noise term. In this case, the quadratically constrained ℓ_1-minimization problem is considered:

$$(\mathrm{BP}_\varepsilon) \quad \min_{x\in\mathbb{R}^n} \{||x||_1 : ||Ax - b||_2 \le \varepsilon\}, \tag{5}$$

where ε is an estimated upper bound on the noise level, i.e., $\|e\|_2 \le \varepsilon$, with $\|\cdot\|_2$ being the Euclidean norm. It follows from compressive sensing [31] that sparse signals can be stably recovered from noisy measurements by solving $(\mathrm{BP}_\varepsilon)$. Note that in the presence of noise-free measurements, $(\mathrm{BP}_\varepsilon)$ with $\varepsilon = 0$ reduces to basis pursuit. Thus, in this paper, we consider only $(\mathrm{BP}_\varepsilon)$ and develop efficient algorithms for solving it.

(BP) and $(\mathrm{BP}_\varepsilon)$ play central roles in sparse recovery; however, these two non-smooth constrained optimization problems are challenging to solve, especially $(\mathrm{BP}_\varepsilon)$. It is well known that (BP) and $(\mathrm{BP}_\varepsilon)$ can be cast as a linear programming problem (LP) and a second-order cone programming problem (SOCP), respectively. The relatively simple primal-dual interior-point method for solving (LP) and the log-barrier method [46] for solving (SOCP) can thus be applied to (BP) and $(\mathrm{BP}_\varepsilon)$, respectively. However, these two methods are typically problematic for large-scale problems because large systems of linear equations must be solved in each iteration. Many efforts have been made to develop more efficient algorithms for solving (BP) and $(\mathrm{BP}_\varepsilon)$. Here, we recall only a few widely used methods; we do not attempt to give a complete review of all proposed methods. The Bregman iteration algorithms were proposed in [61] for solving (BP) and were thoroughly studied and improved in [50]. In recent years, the alternating direction method of multipliers (ADMM) has been extensively investigated for various applications arising in different areas [30,42,60]. Inexact ADMM-based algorithms were developed by Yang and Zhang [59] to solve the two constrained ℓ_1-minimization problems. The developed algorithms are efficient, stable and robust for sparse recovery. Moreover, a fast and accurate first-order algorithm named NESTA was proposed in [8]; this algorithm smooths the ℓ_1-norm and then implements Nesterov's optimal gradient method to solve the smoothed problem.

In the sparse recovery literature, (BP) and $(\mathrm{BP}_\varepsilon)$ have been well studied as an ℓ_1-regularized least-squares problem:

$$\min_{x\in\mathbb{R}^n} \left\{ \frac{1}{2}||Ax - b||_2^2 + \lambda||x||_1 \right\}, \tag{6}$$

where $\lambda > 0$ is a regularization parameter. The problem expressed in (6) is called basis pursuit denoising (BPDN) in [10] and is often called the least absolute shrinkage and selection operator (LASSO) in statistics. Furthermore, since problem (6) can be cast as a quadratic programming problem (QP), it may also be denoted by (QP_λ). From convex optimization theory, it is well known that the solutions to $(\mathrm{BP}_\varepsilon)$ and (QP_λ) are the same when one can select appropriate parameter values ε and λ. Based on the above considerations, many researchers have designed efficient algorithms for solving (QP_λ) instead of $(\mathrm{BP}_\varepsilon)$. Among these algorithms, the iterative shrinkage-thresholding algorithm (ISTA) [15] is the most well-known method because of its simplicity. Since ISTA converges slowly and needs many iterations to obtain an optimal solution, several accelerated versions have been proposed. By using a continuation strategy, Hale et

al. developed a fixed-point continuation method (FPC) in [34]. A more elegant accelerated method known as the fast iterative shrinkage-thresholding algorithm (FISTA), which enjoys a better theoretical rate of convergence, was proposed in [3]. Because, in general, we do not know how to determine in advance the relationship between the parameters ε and λ, these algorithms designed specifically for (QP_λ) may not be able to solve (BP_ε) in practice. In addition, efficient hybrid random/deterministic parallel algorithms were proposed in [19,20] for solving a general optimization problem in which the objective function is the sum of a smooth function and a non-smooth convex one. The work presented in [21,22] used the smoothing technique and then developed second-order methods of handling large-scale optimization problems such as sparse signal recovery using coherent and redundant dictionaries as well as large-scale support vector machines [47].

Recently, operator splitting methods [38] have received considerable attention for solving many structured convex minimization problems. Among these splitting methods, the Douglas–Rachford splitting method originally introduced in [25] possesses many favorable properties. It is well known that for a class of convex problems with a particular structure [27], ADMM is the dual application of the Douglas–Rachford splitting method. It was further elucidated in [27] that the Douglas–Rachford splitting method is a special case of the fundamental proximal point algorithm. Its rate of convergence has recently been studied in [33]. The Douglas–Rachford splitting method has proven to be suitable for several practical applications, such as the traffic equilibrium problem [43], TV-based image recovery [16], semidefinite programming [23], multiplicative noise removal [55], matrix completion [1] and robust principal component analysis [64]. Moreover, primal-dual decomposition algorithms based on the Douglas–Rachford splitting method have recently been proposed for image deblurring [49]. We also note that in [24], this method was applied to (BP), with a focus on analyzing the asymptotic linear convergence rate. The Douglas–Rachford splitting method has also been extended to address several non-convex problems [39].

In this paper, we apply the primal Douglas–Rachford splitting method to (BP_ε). The resulting algorithm mainly consists of computing the proximity operator of the ℓ_1-norm and its projection onto the constraint set in (BP_ε) in each iteration. The proximity operator of the ℓ_1-norm is the well-known soft-thresholding operator [18], which is very simple and incurs only a small computational cost. Thus, the computational cost of the resulting algorithm in each iteration is dominated by the projection onto the constraint set. It is worth noting that in various applications of compressive sensing, the sensing matrix A satisfies $AA^* = I$. By utilizing this special structure, we can derive a closed-form solution for the projection. For cases in which the sensing matrix A does not satisfy $AA^* = I$, a fast and efficient method is proposed for computing the projection onto the constraint set. We list several advantages of the proposed algorithm compared with some popular methods. First, in contrast to the dual application of the Douglas–Rachford splitting method, i.e., the alternating direction method of multipliers (ADMM), the proposed algorithm offers a closed-form solution for each subproblem when the sensing matrix A satisfies $AA^* = I$. Second, the proposed algorithm involves only one parameter, whereas the inexact ADMM, that is, the linearized alternating direction method of multipliers (LADMM), involves more tuned parameters. Third, the proposed algorithm can solve (BP_ε), whereas NESTA was proposed

merely to solve an approximation of (BP_ε). Thus, both its simplicity and efficiency make the proposed algorithm more suitable for sparse signal recovery.

The remainder of the paper is organized as follows. In Sect. 2, the primal Douglas–Rachford splitting method is reviewed and applied to a general convex composite optimization problem. The application of the Douglas–Rachford splitting method to (BP_ε) is presented in Sect. 3. Numerical experiments and conclusions are presented in Sects. 4 and 5, respectively.

2 Douglas–Rachford Splitting Method

Let $\mathbb{R}^d$ denote the usual d-dimensional Euclidean space with the standard inner product $\langle u, v\rangle = \sum_{i=1}^{d} u_i v_i$ for $u, v \in \mathbb{R}^d$ and the corresponding norm. We consider the problem of finding a zero of the sum of two maximally monotone operators in $\mathbb{R}^d$, that is,

$$0 \in T_1(x) + T_2(x). \tag{7}$$

Before applying the Douglas–Rachford splitting method to (7), we need to introduce the resolvent operator of any maximally monotone operator denoted by T. In fact, the resolvent operator with index $\alpha > 0$ of T is defined as $J_T^\alpha = (I + \alpha T)^{-1}$. The resolvent operator J_T^α has several very important properties, as presented below.

Proposition 1 *Let T be a maximally monotone operator. Its resolvent operator J_T^α is a single-valued operator that is defined everywhere on $\mathbb{R}^d$ and is firmly non-expansive, that is,*

$$\|J_T^\alpha(u) - J_T^\alpha(v)\|^2 \le \langle J_T^\alpha(u) - J_T^\alpha(v), u - v\rangle, \ \forall\, u, v \in \mathbb{R}^d. \tag{8}$$

More details on a maximally monotone operator and its resolvent can be found in [4,54].

For solving (7), the Douglas–Rachford splitting method generates a sequence $\{y_k\}$ with an arbitrary initial point $y_0 \in \mathbb{R}^d$ by means of the following iterative scheme:

$$y_{k+1} = [J_{T_2}^\alpha \circ (2J_{T_1}^\alpha - I) + (I - J_{T_1}^\alpha)](y_k), \tag{9}$$

where $\circ$ denotes the composition of operators. The following result regarding the convergence of this algorithm has been established in [43].

Theorem 1 *Let T_1 and T_2 be maximally monotone operators in $\mathbb{R}^d$. If the set of solutions of (7) is non-empty, then the iterative sequence $\{y_k\}$ generated by the scheme given in (9) converges to a point $y \in \mathbb{R}^d$ such that $x = J_{T_1}^\alpha(y)$ is a solution of (7).*

Proof Assume that x is a solution of (7). For $\alpha > 0$, it follows from the inclusion problem expressed in (7) that

$$2x \in (I + \alpha T_1)x + (I + \alpha T_2)x. \tag{10}$$

By choosing a point $y \in (I + \alpha T_1)x$, that is, $x = J^{\alpha}_{T_1}(y)$, we can rewrite (10) as

$$2x - y \in (I + \alpha T_2)x, \tag{11}$$

which indicates that $x = J^{\alpha}_{T_2}(2x - y)$. Based on the equality $x = J^{\alpha}_{T_1}(y)$, we further find that

$$y + J^{\alpha}_{T_2}(2J^{\alpha}_{T_1}(y) - y) = y + J^{\alpha}_{T_1}(y), \tag{12}$$

that is,

$$y = [J^{\alpha}_{T_2} \circ (2J^{\alpha}_{T_1} - I) + (I - J^{\alpha}_{T_1})](y). \tag{13}$$

Let

$$F = J^{\alpha}_{T_2} \circ (2J^{\alpha}_{T_1} - I) + (I - J^{\alpha}_{T_1}).$$

Obviously, the point y is a fixed point of the operator F.

To prove that the iterative sequence generated by the scheme given in (9) converges to a fixed point of F, let us first prove that F is firmly non-expansive. To this end, given two points $x_1,\ x_2 \in \mathbb{R}^d$, we set

$$y_1 = J^{\alpha}_{T_1}(x_1),\ y_2 = J^{\alpha}_{T_1}(x_2),\ z_1 = J^{\alpha}_{T_2}(2y_1 - x_1),\ z_2 = J^{\alpha}_{T_2}(2y_2 - x_2).$$

Thus, $F(x_1) = x_1 + z_1 - y_1$ and $F(x_2) = x_2 + z_2 - y_2$. Since the resolvent operators $J^{\alpha}_{T_1}$ and $J^{\alpha}_{T_2}$ are firmly non-expansive. the following two inequalities hold:

$$\langle y_1 - y_2, x_1 - x_2\rangle \geq \|y_1 - y_2\|^2, \tag{14}$$

$$\langle z_1 - z_2, 2y_1 - x_1 - 2y_2 + x_2\rangle \geq \|z_1 - z_2\|^2. \tag{15}$$

It follows from (14) and (15) that

$$\begin{aligned}
&\langle F(x_1) - F(x_2), x_1 - x_2\rangle \\
&\quad \geq \langle F(x_1) - F(x_2), x_1 - x_2\rangle - \langle y_1 - y_2, x_1 - x_2\rangle + \|y_1 - y_2\|^2 \\
&\quad = \langle z_1 - z_2, x_1 - x_2\rangle + \|x_1 - y_1 - x_2 + y_2\|^2 \\
&\quad = \langle z_1 - z_2, 2y_1 - x_1 - 2y_2 + x_2\rangle - \|z_1 - z_2\|^2 + \|F(x_1) - F(x_2)\|^2 \\
&\quad \geq \|F(x_1) - F(x_2)\|^2.
\end{aligned}$$

Thus, the operator F is firmly non-expansive.

Assume that $\hat{y}$ is a fixed point of F, that is, $\hat{y} = F(\hat{y})$. Furthermore, we have

$$\begin{aligned}\|y_{k+1} - \hat{y}\|^2 &= \|y_{k+1} - y_k + y_k - \hat{y}\|^2 \\ &= \|y_{k+1} - y_k\|^2 + 2\langle y_{k+1} - y_k, y_k - \hat{y}\rangle + \|y_k - \hat{y}\|^2 \end{aligned} \tag{16}$$

and

$$\begin{aligned}\langle y_{k+1} - y_k, y_k - \hat{y}\rangle &= \langle y_{k+1} - \hat{y} + \hat{y} - y_k, y_k - \hat{y}\rangle \\ &= \langle F(y_k) - F(\hat{y}), y_k - \hat{y}\rangle - \|y_k - \hat{y}\|^2 \\ &\geq \|F(y_k) - F(\hat{y})\|^2 - \|y_k - \hat{y}\|^2 \\ &= \|y_{k+1} - \hat{y}\|^2 - \|y_k - \hat{y}\|^2. \end{aligned} \tag{17}$$

(16) and (17) lead to

$$\|y_{k+1} - \hat{y}\|^2 \leq \|y_k - \hat{y}\|^2 - \|y_{k+1} - y_k\|^2. \tag{18}$$

Summing the inequalities in (18) from $k = 1$ to N yields

$$\|y_{N+1} - \hat{y}\|^2 \leq \|y_1 - \hat{y}\|^2 - \sum_{k=1}^{N} \|y_{k+1} - y_k\|^2. \tag{19}$$

First, it follows from the inequality expressed in (19) that the iterative sequence $\{y_k\}$ is bounded, which indicates that the iterative sequence $\{y_k\}$ has at least one cluster point, denoted by $\bar{y}$. We also assume that $\{y_{k_i}\}$ is a subsequence that converges to $\bar{y}$. Second, (19) also indicates that

$$\sum_{k=1}^{N} \|y_{k+1} - y_k\|^2 < +\infty, \tag{20}$$

which further implies that

$$\lim_{k\to\infty} \|y_{k+1} - y_k\|^2 = 0. \tag{21}$$

Upon setting $k = k_i$ in (21), we obtain

$$\lim_{i\to\infty} y_{k_i+1} = \lim_{i\to\infty} y_{k_i} = \bar{y}. \tag{22}$$

Since F is firmly non-expansive and thus is continuous, we find that

$$\bar{y} = \lim_{i\to\infty} y_{k_i+1} = \lim_{i\to\infty} F(y_{k_i}) = F(\bar{y}). \tag{23}$$

Hence, $\bar{y}$ is a fixed point of F.

We now replace $\hat{y}$ in (19) with $\bar{y}$. Obviously, the iterative sequence $\{\|y_k - \bar{y}\|^2\}$ is monotonically decreasing. Since the subsequence $\{y_{k_i}\}$ converges to $\bar{y}$, we find that

$$\lim_{k\to\infty} y_k = \bar{y}.$$

The proof is complete. □

We now focus on the convex composite optimization problem of the form

$$\min_{x\in\mathbb{R}^d} \{f(x) + g(x)\}, \tag{24}$$

where $f, g : \mathbb{R}^d \to \bar{\mathbb{R}} = \mathbb{R} \cup \{+\infty\}$ are proper lower-semicontinuous convex functions. It follows from Fermat's rule that the optimality condition for solving (24) is

$$0 \in \partial f(x) + \partial g(x), \tag{25}$$

where ∂f and ∂g are the subdifferentials of functions f and g, respectively.

The subdifferential of a proper lower-semicontinuous convex function f at a given point $x \in \mathbb{R}^d$ is the set defined by

$$\partial f(x) = \{z \in \mathbb{R}^d | \, \forall u \in \mathbb{R}^d, \ f(u) \geq f(x) + \langle z, u - x\rangle\}.$$

In particular, $\partial f(x) = \{\nabla f(x)\}$ if the function f is differentiable at the point x.

The problem defined in (24) can be considered a problem of the type represented by (7) since the optimality condition (25) can be expressed as (7) by letting $T_1 = \partial f$ and $T_2 = \partial g$. Moreover, the subdifferential operator of any proper lower-semicontinuous convex function is maximally monotone [4]. When the iterative scheme described in (9) is applied to the monotone inclusion problem defined in (25), it is necessary to compute the resolvent of the subdifferential operator. In fact, the resolvent operator with index $\alpha > 0$ of the subdifferential operator of any proper lower-semicontinuous convex function f is the proximal operator with index $\alpha > 0$ of f, that is,

$$\text{prox}_{\alpha f} = (I + \alpha\partial f)^{-1},$$

where the proximal operator $\text{prox}_{\alpha f}$ from $\mathbb{R}^d$ to $\mathbb{R}^d$ is defined as

$$\text{prox}_{\alpha f}(x) = \underset{z\in\mathbb{R}^d}{\text{argmin}} \left\{ \frac{1}{2\alpha}||z - x||^2 + f(z) \right\}. \tag{26}$$

Building on the proximity operators of f and g, we rewrite the Douglas–Rachford splitting method applied to (24) as follows:

Algorithm 1 The Douglas–Rachford splitting method

Input: Initialization: $y_0, \alpha > 0$.
1: **for** $k = 1, 2, \ldots, K$ **do**
2: $x_{k+1} = \text{prox}_{\alpha f}(y_k)$;
3: $z_{k+1} = \text{prox}_{\alpha g}(2x_{k+1} - y_k)$;
4: $y_{k+1} = z_{k+1} + y_k - x_{k+1}$.
5: **end for**
Output: x_K

Note that the above Douglas–Rachford splitting method is a simple iterative scheme that requires only the computation of the proximity operators of the functions f and g. In the remainder of this section, the proximity operators of several functions that we will need in later sections are presented.

For a non-empty closed convex set $C \subseteq \mathbb{R}^d$, its indicator function is defined as

$$\iota_C(x) = \begin{cases} 0, & \text{if } x \in C, \\ +\infty, & \text{if } x \notin C. \end{cases}$$

Clearly, for any non-empty closed convex set C, we have $\text{prox}_{\alpha\iota_C} = \mathcal{P}_C$, where $\mathcal{P}_C : \mathbb{R}^d \to C, \mathcal{P}_C(x) = \text{argmin}_{z\in C} \|z - x\|$, denotes the projection operator onto C. For example, for the indicator function of a closed ℓ_2-ball in $\mathbb{R}^d$ with radius ε, denoted by B_2^ε, it is very easy to compute its proximity operator $\text{prox}_{\alpha\iota_{B_2^\varepsilon}}$, that is, the projection onto B_2^ε. In fact, its proximity operator can be written as

$$\text{prox}_{\alpha\iota_{B_2^\varepsilon}}(x) = \mathcal{P}_{B_2^\varepsilon}(x) = \min\left\{1, \frac{\varepsilon}{\|x\|_2}\right\} \cdot x. \tag{27}$$

We can also compute proximity operators of the ℓ_1-norm and ℓ_2-norm in $\mathbb{R}^d$. The proximity operator $\text{prox}_{\alpha||\cdot||_1}$ is the well-known soft-thresholding operator [18], defined as

$$\text{prox}_{\alpha||\cdot||_1}(x) = \text{sgn}(x) \odot \max\{|x| - \alpha, 0\}, \tag{28}$$

where $\odot$ represents elementwise multiplication.

The proximity operator $\text{prox}_{\alpha||\cdot||_2}$ can be expressed as

$$\text{prox}_{\alpha\|\cdot\|_2}(x) = \begin{cases} \max\left\{1 - \frac{\alpha}{||x||_2}, 0\right\} x, & x \neq 0, \\ 0, & x = 0. \end{cases} \tag{29}$$

Proof We let $\mathcal{J}$ denote the proximal function:

$$\mathcal{J}(z) = \frac{1}{2}||z - x||^2 + \alpha||z||_2.$$

Since $\mathcal{J}(z)$ is non-differentiable only at 0, $\text{prox}_{\alpha||\cdot||_2}(x)$ must be 0 or a point $\bar{z}$ that satisfies the following first-order condition:

$$\bar{z} - x + \frac{\alpha}{||\bar{z}||_2}\bar{z} = 0.$$

The above equation implies that the following two equalities hold:

$$||\bar{z}||_2 = ||x||_2 - \alpha, \quad \bar{z} = \frac{||x||_2 - \alpha}{||x||_2}x.$$

By comparing $\mathcal{J}\left(\frac{||x||_2-\alpha}{||x||_2}x\right)$ with $\mathcal{J}(0)$, we find that

$$\mathcal{J}\left(\frac{||x||_2 - \alpha}{||x||_2}x\right) - \mathcal{J}(0) = \begin{cases} -\frac{1}{2}(||x||_2 - \alpha)^2 < 0, & ||x||_2 > \alpha, \\ \frac{1}{2}(\alpha - ||x||_2)(3\alpha + ||x||_2) \geq 0, & ||x||_2 \leq \alpha. \end{cases}$$

Hence, we obtain

$$\text{prox}_{\alpha||\cdot||_2}(x) = \begin{cases} (1 - \alpha/||x||_2)x, & ||x||_2 > \alpha, \\ 0, & ||x||_2 \leq \alpha, \end{cases}$$

which is equivalent to (29). The proof is complete. □

3 Application to (BP_ε)

In this section, the primal Douglas–Rachford splitting method is applied to solve (BP_ε), and a fast and efficient algorithm is developed to address the subproblems of the proposed algorithm. The relation of the proposed algorithm to the popular ADMM and LADMM is also discussed.

(BP_ε) can be rewritten as the following unconstrained convex composite problem:

$$\min_{x\in\mathbb{R}^n} \{\|x\|_1 + \iota_{B_2^\varepsilon}(Ax - b)\}, \tag{30}$$

where $\iota_{B_2^\varepsilon}$ is the indicator function of a closed ℓ_2-ball in $\mathbb{R}^m$ with radius ε (B_2^ε). If the functions f and g in (24) are taken to be $\|x\|_1$ and $\iota_{B_2^\varepsilon}(Ax - b)$, respectively, then the Douglas–Rachford splitting method for solving (24) can be applied to (30). To this end, we need to compute the proximity operators of $\|\cdot\|_1$ and $\iota_{B_2^\varepsilon}(A\cdot - b)$. The proximity operator of $\|\cdot\|_1$ has previously been presented in (28). For the function $\iota_{B_2^\varepsilon}(A\cdot - b)$, its proximity operator is the projection onto the constraint set in (BP_ε), which is computationally difficult to solve in general. However, a fast and efficient method of solving it can be designed. Furthermore, a closed-form solution for the projection can also be obtained when the matrix A satisfies $AA^* = I$. These results are presented in Theorem 2.

Theorem 2 *The proximity operator of the function $\iota_{B_2^\varepsilon}(A\cdot - b)$ with index $\alpha > 0$, that is,*

$$prox_{\alpha\iota_{B_2^\varepsilon}(A\cdot-b)}(x) = \underset{z\in\mathbb{R}^n}{\operatorname{argmin}} \left\{ \frac{1}{2\alpha}||z-x||^2 + \iota_{B_2^\varepsilon}(Az-b) \right\}, \tag{31}$$

can be computed as follows:

$$prox_{\alpha\iota_{B_2^\varepsilon}(A\cdot-b)}(x) = x - A^*\tilde{y}. \tag{32}$$

In (32), $\tilde{y}$ is a minimizer of the following optimization problem:

$$\min_{y\in\mathbb{R}^m} \left\{ \frac{1}{2}\|A^*y\|^2 - \langle y, Ax-b\rangle + \varepsilon\|y\|_2 \right\}. \tag{33}$$

In particular, when the matrix A satisfies $AA^* = I$, the equality given in (32) can be written as

$$prox_{\alpha\iota_{B_2^\varepsilon}(A\cdot-b)}(x) = x - A^* prox_{\epsilon\|\cdot\|_2}(Ax-b). \tag{34}$$

Proof Since $\text{prox}_{\alpha\iota_{B_2^\varepsilon}(A\cdot-b)}$ is the projection onto the constraint set in (BP_ε), we can write that $\text{prox}_{\alpha\iota_{B_2^\varepsilon}(A\cdot-b)} = \text{prox}_{\iota_{B_2^\varepsilon}(A\cdot-b)}$ for any index $\alpha > 0$. Therefore, we consider the following problem:

$$\min_{z\in\mathbb{R}^n} \left\{ \frac{1}{2}||z-x||^2 + \iota_{B_2^\varepsilon}(Az-b) \right\}. \tag{35}$$

The dual problem of (35) can be derived as follows. First, (35) is rewritten in the following constrained form:

$$\min_{z\in\mathbb{R}^n, w\in\mathbb{R}^m} \left\{ \frac{1}{2}||z-x||^2 + \iota_{B_2^\varepsilon}(w) : w = Az-b \right\}. \tag{36}$$

By introducing a dual variable $y \in \mathbb{R}^m$, we can construct the Lagrangian function of the problem expressed in (36):

$$L(z,w,y) = \frac{1}{2}||z-x||^2 + \iota_{B_2^\varepsilon}(w) - \langle y, w-(Az-b)\rangle. \tag{37}$$

By minimizing (37) with respect to the primal variables z and w, we find that the dual function $d(y)$ is

$$\begin{aligned} d(y) &= \min_{z\in\mathbb{R}^n, w\in\mathbb{R}^m} L(z, w, y) \\ &= \min_{z\in\mathbb{R}^n} \left\{ \frac{1}{2}||z - x||^2 + \langle A^* y, z\rangle \right\} \\ &\quad + \min_{w\in\mathbb{R}^m} \{\iota_{B_2^\varepsilon}(w) - \langle y, w\rangle\} - \langle y, b\rangle. \end{aligned} \tag{38}$$

To obtain $d(y)$, we need to solve the following two optimization problems:

$$\min_{z\in\mathbb{R}^n} \left\{ \frac{1}{2}||z - x||^2 + \langle A^* y, z\rangle \right\} \tag{39}$$

and

$$\min_{w\in\mathbb{R}^m} \{\iota_{B_2^\varepsilon}(w) - \langle y, w\rangle\}. \tag{40}$$

Let us first consider problem (39). It is easy to compute the minimizer of (39) denoted by $\tilde{z}$, that is, $\tilde{z} = x - A^* y$. Thus,

$$\begin{aligned} &\min_{z\in\mathbb{R}^n} \left\{ \frac{1}{2}||z - x||^2 + \langle A^* y, z\rangle \right\} \\ &= \frac{1}{2}||\tilde{z} - x||^2 + \langle A^* y, \tilde{z}\rangle \\ &= -\frac{1}{2}\|A^* y\|^2 + \langle y, Ax\rangle. \end{aligned} \tag{41}$$

For problem (40), we have

$$\begin{aligned} \min_{w\in\mathbb{R}^m} \{\iota_{B_2^\varepsilon}(w) - \langle y, w\rangle\} &= \min_{w\in\mathbb{R}^m} \{-\langle y, w\rangle : \|w\|_2 \le \varepsilon\} \\ &= -\max_{w\in\mathbb{R}^m} \{\langle y, w\rangle : \|w\|_2 \le \varepsilon\} \\ &= -\max_{w'\in\mathbb{R}^m} \{\langle \varepsilon y, w'\rangle : \|w'\|_2 \le 1,\ w = \varepsilon w'\} \\ &= -\varepsilon\|y\|_2. \end{aligned}$$

The last equality above is derived from the fact that the dual norm of the ℓ_2-norm is itself.

By combining the work presented above, we obtain the following dual problem:

$$\max_{y\in\mathbb{R}^m} \left\{ -\frac{1}{2}\|A^* y\|^2 + \langle y, Ax\rangle - \varepsilon\|y\|_2 - \langle y, b\rangle \right\}. \tag{42}$$

Problem (36) is convex, and its constraints are all affine. By Slater's condition, as long as problem (36) is feasible, the strong duality condition for convex problems

(see, e.g., [54]) holds, and the optimal solution of the dual problem expressed in (42) is attained. Let $\tilde{z}$ and $\tilde{w}$ denote the optimal primal variables, and let $\tilde{y}$ denote the optimal dual variable. From the KKT condition on problem (36), we find that

$$0 = \partial_z L(\tilde{z}, \tilde{w}, \tilde{y}) = \tilde{z} - x + A^*\tilde{y}, \tag{43}$$

$$0 \in \partial_w L(\tilde{z}, \tilde{w}, \tilde{y}) = \partial \iota_{B_2^\varepsilon}(\tilde{w}) - \tilde{y}, \tag{44}$$

$$0 = \partial_y L(\tilde{z}, \tilde{w}, \tilde{y}) = \tilde{w} - (A\tilde{z} - b). \tag{45}$$

To obtain an optimal solution $\tilde{y}$, we can also solve the following problem:

$$\min_{y\in\mathbb{R}^m} \left\{ \frac{1}{2}\|A^*y\|^2 - \langle y, Ax - b\rangle + \varepsilon\|y\|_2 \right\}. \tag{46}$$

From (43), the equality expressed in (32) can be obtained. If the matrix A satisfies $AA^* = I$, then the optimal solution to (46) is $\tilde{y} = \text{prox}_{\epsilon\|\cdot\|_2}(Ax - b)$, which leads to (34). The proof is complete. □

Remark 1 Since z can be taken to be $\bar{x}$, which is the original sparse signal, $w = A\bar{x} - b \in B_2^\varepsilon$ holds under the assumption that ε is an estimated upper bound on the noise level, i.e., $\|A\bar{x} - b\|_2 \le \varepsilon$. Thus, problem (36) is feasible.

Remark 2 In many practical applications, as shown in [59], the sensing matrices A are often formed by randomly taking a subset of rows from orthonormal transform matrices, such as discrete cosine transform (DCT), discrete Fourier transform (DFT) or discrete Walsh–Hadamard transform (DWHT) matrices. Such sensing matrices do not require storage and permit a fast algorithm for matrix-vector multiplication. Since the rows of A are orthonormal, the condition $AA^* = I$ holds for such sensing matrices.

Remark 3 Other widely used types of sensing matrices A, such as random Gaussian matrices, are often full spark matrices [2], which have the property that any m columns of A are linearly independent. In fact, a matrix whose entries are continuous random variables drawn from independent and identical distributions will be a full spark matrix with probability one; this assertion has recently been proved in [9]. A full spark matrix is necessarily also of full rank; that is, the m rows of A are linearly independent. Thus, an orthonormal basis for the span of the rows of A can be formulated from linear combinations of those rows. If we multiply A by an elementary matrix denoted by E, then the matrix EA will have orthonormal rows. Since an elementary matrix is invertible, $Ax = b$ is equivalent to $EAx = Eb$. Moreover, the condition of every $m \times m$ submatrix of A being invertible implies that every $m \times m$ submatrix of EA is also invertible, i.e., EA is a full spark matrix. Thus, in the noise-free case, we can always reformulate (BP) such that the rows of A are orthonormal, i.e., $AA^* = I$.

Remark 4 The results of Theorem 2 can be generalized to evaluate the proximity operator of a general composite function $f \circ B$, where the function f has an inexpensive proximity operator and B is a matrix or linear operator. Evaluating the proximity operator of $f \circ B$ is critical for several practical problems, such as total-variation-based image denoising or deblurring [3] and compressive sensing problems with coherent and redundant dictionaries [21].

For a case in which the measurements are perturbed by noise and the sensing matrix A does not satisfy $AA^* = I$, Theorem 2 provides an approach for computing the value of $\text{prox}_{\alpha \iota_{B_2^\varepsilon}(A\cdot - b)}$ at any point x. Specifically, a solution $\tilde{y}$ to problem (33) is first found, and then, $\text{prox}_{\alpha \iota_{B_2^\varepsilon}(A\cdot - b)}(x)$ is identified as

$$\text{prox}_{\alpha \iota_{B_2^\varepsilon}(A\cdot - b)}(x) = x - A^*\tilde{y}. \tag{47}$$

We now develop a fast and efficient method of finding solutions to problem (33). Clearly, problem (33) is a special case of a minimization problem of the following form:

$$\min_{y \in \mathbb{R}^m} \{F(y) = \Phi(y) + \Psi(y)\}, \tag{48}$$

where $\Psi : \mathbb{R}^m \to \bar{\mathbb{R}} = \mathbb{R} \cup \{+\infty\}$ is a proper lower-semicontinuous convex function and $\Phi : \mathbb{R}^m \to \mathbb{R}$ is a continuously differentiable convex function with an L-Lipschitz continuous gradient, i.e.,

$$\|\nabla\Phi(u) - \nabla\Phi(v)\| \le L\|u - v\| \quad \forall u,\ v \in \mathbb{R}^m.$$

The well-known FISTA mentioned above can be applied to solve the minimization problem expressed in (48). FISTA can be described as follows:

Algorithm 2 FISTA

Input: Initialization: $z_1 = y_0$, $t_1 = 1$.
1: **for** $k = 1, 2, \ldots, K$ **do**
2: $y_k = \text{prox}_{\frac{1}{L}\Psi}\left(z_k - \frac{1}{L}\nabla\Phi(z_k)\right)$;
3: $t_{k+1} = \frac{1+\sqrt{1+4t_k^2}}{2}$;
4: $z_{k+1} = y_k + \frac{t_k - 1}{t_{k+1}}(y_k - y_{k-1})$.
5: **end for**
Output: y_K

The remarkable property of this algorithm is that it guarantees a rate of convergence of $\mathcal{O}(k^{-2})$ for the iterative sequence of the function values, where k is the number of

iterations. In fact, it has been verified in [3] that for any $k \geq 1$,

$$F(y_k) - F(y^*) \leq \frac{2L\|y_0 - y^*\|^2}{(k+1)^2},$$

where y^* is a solution to problem (48).

To enable the application of the above-described Algorithm 2 to problem (33), Φ and Ψ in problem (48) are, respectively, taken to be

$$\Phi(y) := \frac{1}{2}\|A^*y\|^2 - \langle y, Ax - b\rangle \quad \text{and} \quad \Psi(y) := \varepsilon\|y\|_2. \tag{49}$$

Obviously, the function Φ in (49) is differentiable, and its gradient, given by

$$\nabla\Phi(y) = AA^*y - (Ax - b),$$

is Lipschitz continuous with constant $\|A\|_2^2$. As a result, the algorithm for computing $\text{prox}_{\alpha\iota_{B_2^\varepsilon}(A\cdot - b)}$ can be written as follows:

Algorithm 3 FISTA for $\text{prox}_{\alpha\iota_{B_2^\varepsilon}(A\cdot - b)}(\text{x})$

Input: Initialization: $x, b, z_1 = y_0, t_1 = 1$.
1: **for** $k = 1, 2, \ldots, K$ **do**
2: $y_k = \text{prox}_{\frac{\varepsilon}{L}\|\cdot\|_2}\left(z_k - \frac{1}{L}\left(A(A^*z_k - x) + b\right)\right)$;
3: $t_{k+1} = \frac{1+\sqrt{1+4t_k^2}}{2}$;
4: $z_{k+1} = y_k + \frac{t_k - 1}{t_{k+1}}(y_k - y_{k-1})$.
5: **end for**
Output: $\text{prox}_{\alpha\iota_{B_2^\varepsilon}(A\cdot - b)}(x) = x - A^*y_K$.

The Douglas–Rachford splitting method for (BP_ε) can be summarized as follows:

Algorithm 4 The Douglas–Rachford splitting method for (BP_ε)

Input: Initialization: y_0; $\alpha > 0$.
1: **for** $k = 0, 1, 2, \ldots, K - 1$ **do**
2: Compute $x_{k+1} = \text{prox}_{\alpha\|\cdot\|_1}(y_k)$ using (28);
3: If $AA^* = I$, compute $z_{k+1} = \text{prox}_{\alpha\iota_{B_2^\varepsilon}(A\cdot - b)}(2x_{k+1} - y_k)$ using (34);
otherwise,
compute $z_{k+1} = \text{prox}_{\alpha\iota_{B_2^\varepsilon}(A\cdot - b)}(2x_{k+1} - y_k)$ using Algorithm 3;
4: $y_{k+1} = z_{k+1} + y_k - x_{k+1}$.
5: **end for**
Output: x_K.

In the remainder of this section, ADMM and LADMM are reviewed and compared with Algorithm 4. Based on the work presented in [59], we can introduce a variable r to reformulate (30) as follows:

$$\min_{x\in\mathbb{R}^n, r\in\mathbb{R}^m} \{\|x\|_1 + \iota_{B_2^\varepsilon}(r) : r = Ax - b\}. \tag{50}$$

The augmented Lagrangian function of (50) is

$$L(x, r, y) = \|x\|_1 + \iota_{B_2^\varepsilon}(r) - \langle y,\ Ax - r - b\rangle + \frac{\alpha}{2}\|Ax - r - b\|^2. \tag{51}$$

Here, $y \in \mathbb{R}^m$ is the Lagrangian multiplier vector, and $\alpha > 0$ is a given penalty parameter. The classical ADMM applied to (51) is written as follows:

$$\begin{cases} x_{k+1} = \operatorname{argmin}_{x\in\mathbb{R}^n} L(x, r_k, y_k), \\ r_{k+1} = \operatorname{argmin}_{r\in\mathbb{R}^m} L(x_{k+1}, r, y_k), \\ y_{k+1} = y_k - \gamma\alpha(Ax_{k+1} - r_{k+1} - b), \end{cases} \tag{52}$$

where $\gamma \in (0, 2)$ is a relaxation parameter. When $\gamma = 1$, the ADMM is the dual application of the Douglas–Rachford splitting method. Clearly, (52) can be equivalent to

$$\begin{cases} x_{k+1} = \operatorname{argmin}_{x\in\mathbb{R}^n} \left\{\|x\|_1 + \frac{\alpha}{2}\|Ax - r_k - b - y_k/\alpha\|^2\right\}, \\ r_{k+1} = \operatorname{argmin}_{r\in\mathbb{R}^m} \left\{\iota_{B_2^\varepsilon}(r) + \frac{\alpha}{2}\|Ax_{k+1} - r - b - y_k/\alpha\|^2\right\}, \\ y_{k+1} = y_k - \gamma\alpha(Ax_{k+1} - r_{k+1} - b). \end{cases} \tag{53}$$

The x-related subproblem of (53) is difficult to solve; one strategy for alleviating this difficulty is to linearize the quadratic term $\frac{1}{2}\|Ax - r_k - b - y_k/\alpha\|^2$, resulting in the following approximate x-related subproblem for (53):

$$\operatorname{argmin}_{x\in\mathbb{R}^n} \left\{\|x\|_1 + \alpha\left(\langle g_k,\ x - x_k\rangle + \frac{1}{2\beta}\|x - x_k\|^2\right)\right\}. \tag{54}$$

In (54), $g_k = A^*(Ax_k - r_k - b - y_k/\alpha)$ is the gradient of the quadratic term $\frac{1}{2}\|Ax - r_k - b - y_k/\alpha\|^2$ at $x = x_k$, and $\beta > 0$ is a proximal parameter. The solution to (54) is given explicitly by

$$x_{k+1} = \operatorname{prox}_{\frac{\beta}{\alpha}\|\cdot\|_1}(x_k - \beta g_k). \tag{55}$$

It is very easy to obtain the solution to the r-related subproblem of (53), which can be expressed as

$$r_{k+1} = \operatorname{prox}_{\frac{1}{\alpha}\iota_{B_2^\varepsilon}}(Ax_{k+1} - b - y_k/\alpha). \tag{56}$$

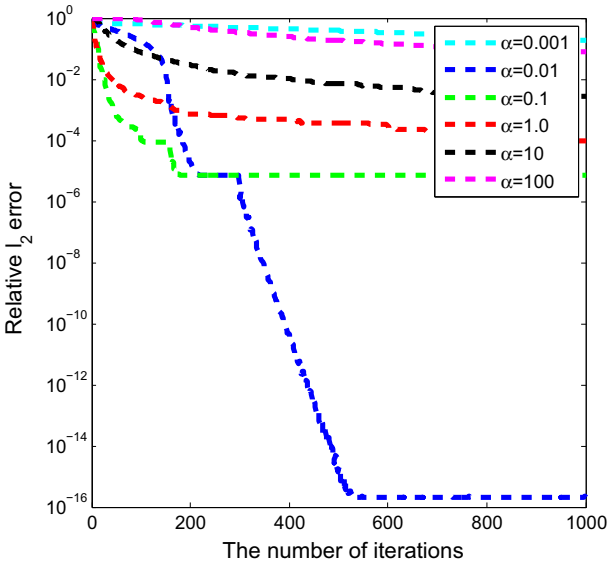

Fig. 1 Comparison of performance of Algorithm 4 with different parameter values of α

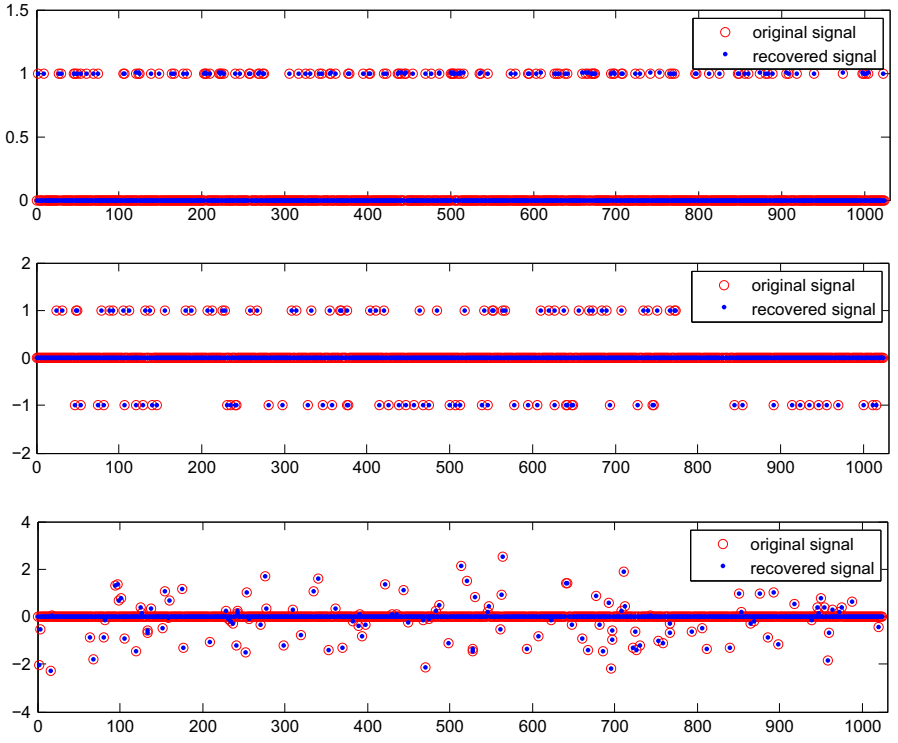

Fig. 2 Perfect recovery of three types of signals with (1024, 512, 128) using Algorithm 4 in the noiseless case

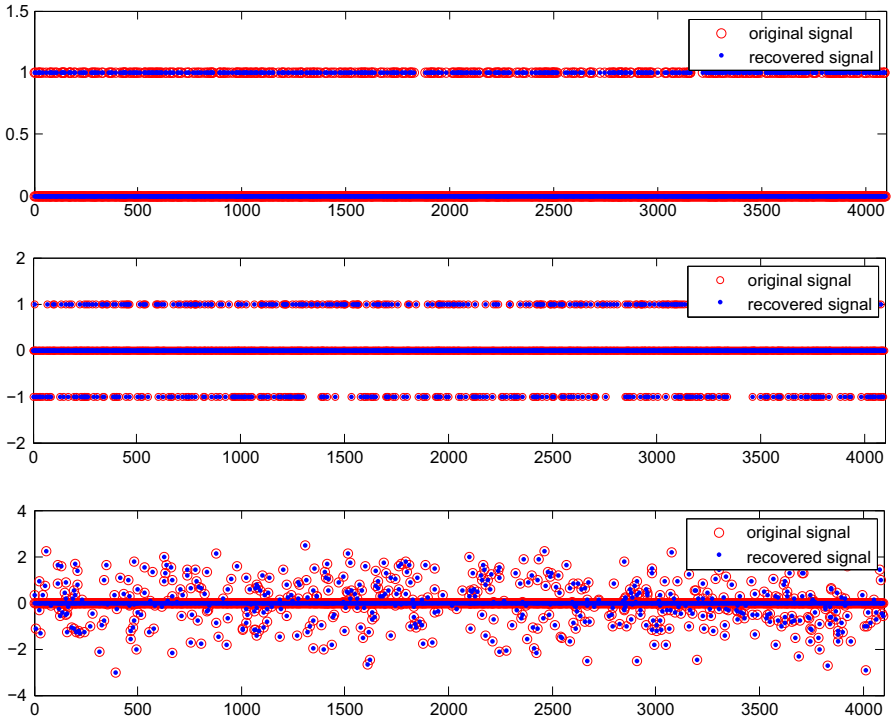

Fig. 3 Perfect recovery of three types of signals with (4096, 2048, 512) using Algorithm 4 in the noiseless case

Hence, LADMM can be written as

$$\begin{cases} x_{k+1} = \operatorname{prox}_{\frac{\beta}{\alpha}\|\cdot\|_1}(x_k - \beta g_k), \\ r_{k+1} = \operatorname{prox}_{\frac{1}{\alpha}\iota_{B_2^\varepsilon}}(Ax_{k+1} - b - y_k/\alpha), \\ y_{k+1} = y_k - \gamma\alpha(Ax_{k+1} - r_{k+1} - b). \end{cases} \tag{57}$$

Note that (57) is an inexact version of ADMM because the x-related subproblem is solved approximately.

A few remarks on Algorithm 4, ADMM and LADMM are in order. First, unlike ADMM, each subproblem of Algorithm 4 has a closed-form solution when the sensing matrix A satisfies $AA^* = I$. Second, when a general sensing matrix is used, we can more exactly solve the second subproblem (projection problem) of Algorithm 4, whereas in LADMM, a gradient approach is used to approximately solve the x-related subproblem of ADMM. Third, our algorithm involves only one parameter, α, whereas LADMM requires three parameters, α, β and γ, which are more difficult to tune to ensure the efficiency of LADMM.

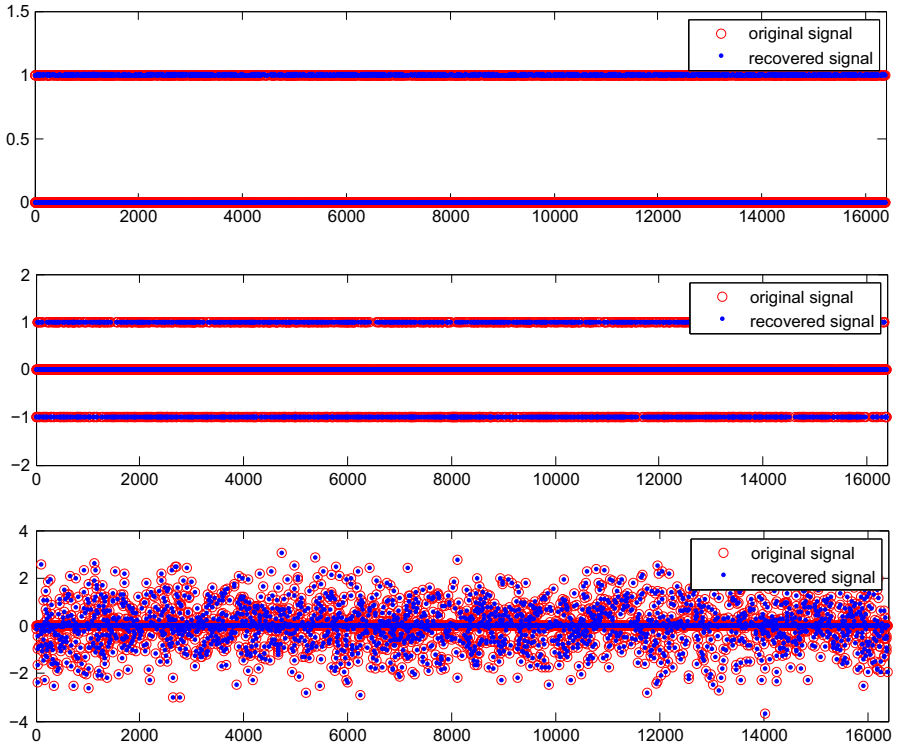

Fig. 4 Perfect recovery of three types of signals with $(16,384, 8192, 2048)$ using Algorithm 4 in the noiseless case

4 Numerical Results

This section reports the implementation of the proposed Algorithm 4 for handling sparse signal recovery problems and the verification of its numerical efficiency. It is compared with the well-known NESTA and LADMM. All experiments were performed on a Thinkpad SL510 with an Intel(R) Core(TM)2 Duo CPU @ 2.20 GHz with 1.99 GB of RAM running Microsoft Windows XP 2002.

4.1 Experimental Setup and Parameter Selection

This subsection describes how the $m \times n$ sensing matrices A and sparse signals were obtained in our experiments. Two types of sensing matrices were considered: sensing matrices that satisfy $AA^* = I$ and those that do not. Two approaches for generating sensing matrices that satisfy $AA^* = I$ were applied. In the first approach, the $m \times n$ sensing matrix was taken to be the partial DCT matrix generated by randomly choosing m rows from the $n \times n$ DCT matrix. In the second approach, we generated a random Gaussian matrix by independently drawing its elements from a standard normal distribution and then orthonormalizing its rows. In this second case, A was

Table 1 Average performance of the three algorithms in the case of the partial DCT sensing matrix for the noiseless measurements

Method	Algorithm 4		NESTA		LADMM	
Signal	RE	Time (s)	RE	Time (s)	RE	Time (s)
$(n, m, s) = (1024, 512, 128)$						
Type 1	1.18e−16	0.2024	4.27e−11	0.8263	8.35e−14	0.5528
Type 2	2.01e−16	0.2486	8.19e−11	0.8310	2.01e−14	0.5633
Type 3	6.55e−16	0.2254	4.11e−11	0.9372	3.72e−14	0.5882
$(n, m, s) = (4096, 2048, 512)$						
Type 1	4.12e−16	1.1249	7.23e−11	5.3682	2.90e−14	2.5827
Type 2	8.53e−16	1.1478	8.37e−11	5.4821	4.11e−14	2.9153
Type 3	2.20e−16	1.2163	6.16e−11	5.6193	2.85e−14	2.8126
$(n, m, s) = (16{,}384, 8192, 2048)$						
Type 1	5.11e−16	4.0155	3.29e−11	12.3112	2.73e−14	7.5612
Type 2	9.32e−16	4.4786	1.16e−10	12.4728	2.58e−14	8.1247
Type 3	7.45e−16	5.1083	3.82e−10	12.8291	1.99e−13	8.7271

Table 2 Average performance of the three algorithms in the case of the orthonormalized Gaussian sensing matrix for the noiseless measurements

Method	Algorithm 4		NESTA		LADMM	
Signal	RE	Time (s)	RE	Time (s)	RE	Time (s)
$(n, m, s) = (1024, 512, 128)$						
Type 1	2.56e−16	1.5078	9.23e−11	3.0138	3.45e−14	2.2733
Type 2	4.12e−16	1.9786	4.33e−11	3.2745	2.56e−14	2.5291
Type 3	5.61e−16	1.7534	8.15e−11	3.7291	8.90e−14	2.7183
$(n, m, s) = (4096, 2048, 512)$						
Type 1	3.79e−16	17.3191	3.55e−11	25.1583	1.14e−14	19.3957
Type 2	9.12e−16	17.5313	1.25e−11	25.9472	9.27e−14	19.7380
Type 3	6.31e−16	17.6156	4.42e−11	25.7390	9.38e−14	20.0358
$(n, m, s) = (16{,}384, 8192, 2048)$						
Type 1	4.62e−16	38.2703	3.80e−10	49.3792	1.48e−13	42.8473
Type 2	3.16e−16	38.5829	9.17e−10	49.2547	9.02e−13	42.5836
Type 3	8.39e−16	38.7201	2.63e−10	49.5831	3.57e−13	42.0562

directly taken to be a general Gaussian random matrix with entries independently drawn from the normal distribution $\mathcal{N}(0, 1/m)$. We considered three types of signals of length n and sparsity level s, i.e., with s nonzero components. Specifically, for signals of the first type, each nonzero component was equal to 1, whereas for signals of the second type, each nonzero component was equal to either 1 or -1. For signals of the last type, each nonzero component was drawn from a standard normal distribution. The locations of the nonzero components were randomly permuted in all three types of signals. We

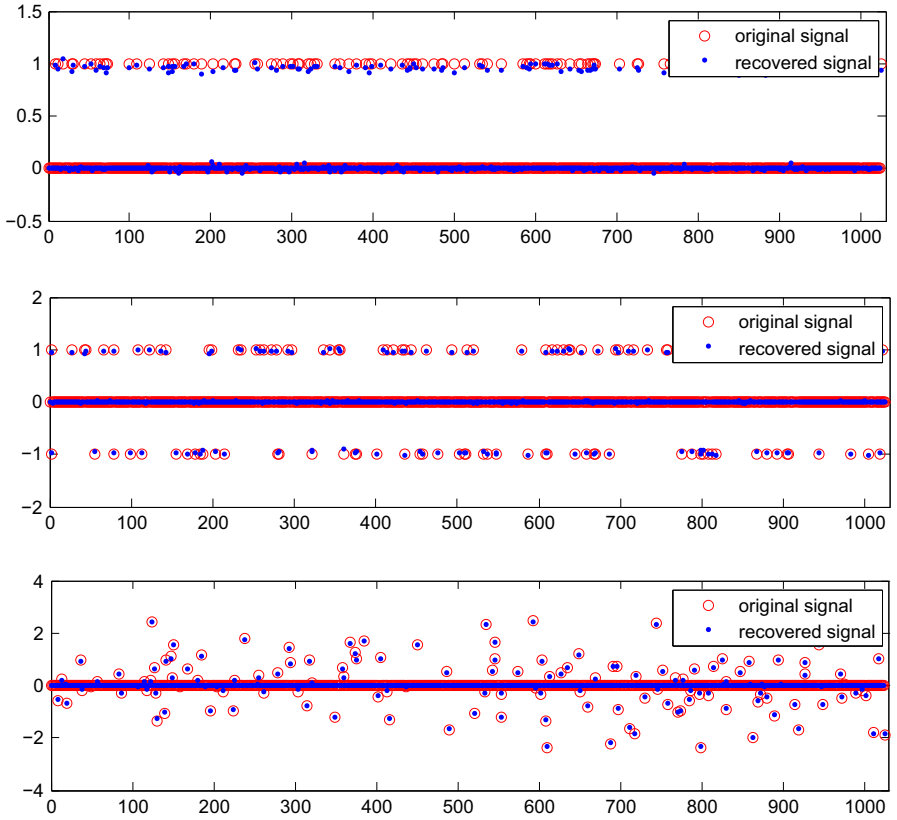

Fig. 5 Accurate recovery of three types of signals with (1024, 512, 128) using Algorithm 4 in the case of the partial DCT sensing matrix for the noise level $\sigma = 0.01$

let $(n,\ m,\ s)$ denote signals of different lengths n, numbers of measurements m and sparsity levels s. In our experiments, we considered $n = 2^{10}, 2^{12}, 2^{14}, m = n/2$ and $s = m/4$, i.e., (1024, 512, 128), (4096, 2048, 512) and (16384, 8192, 2048). The measurements were collected in the form $b = A\bar{x}$ in the noiseless case or $b = A\bar{x} + e$ in the noisy case, where e represents Gaussian noise. The accuracy of a recovered solution can be assessed based on the relative ℓ_2-error, which is defined as follows:

$$\text{RE} = \frac{||\bar{x} - x^*||_2}{||x||_2}. \tag{58}$$

where $\bar{x}$ is the original signal and x^* is the recovered signal.

To implement the proposed Algorithm 4, we need to determine the relevant parameter α. We first observed the impact of different values of α on the performance of Algorithm 4. To this end, 10 sparse signals of the third type with (16,384, 8192, 2048) were generated as described above, and the sensing matrix A was taken to be the partial DCT matrix. The measurements were free of noise. Six different values of α, i.e.,

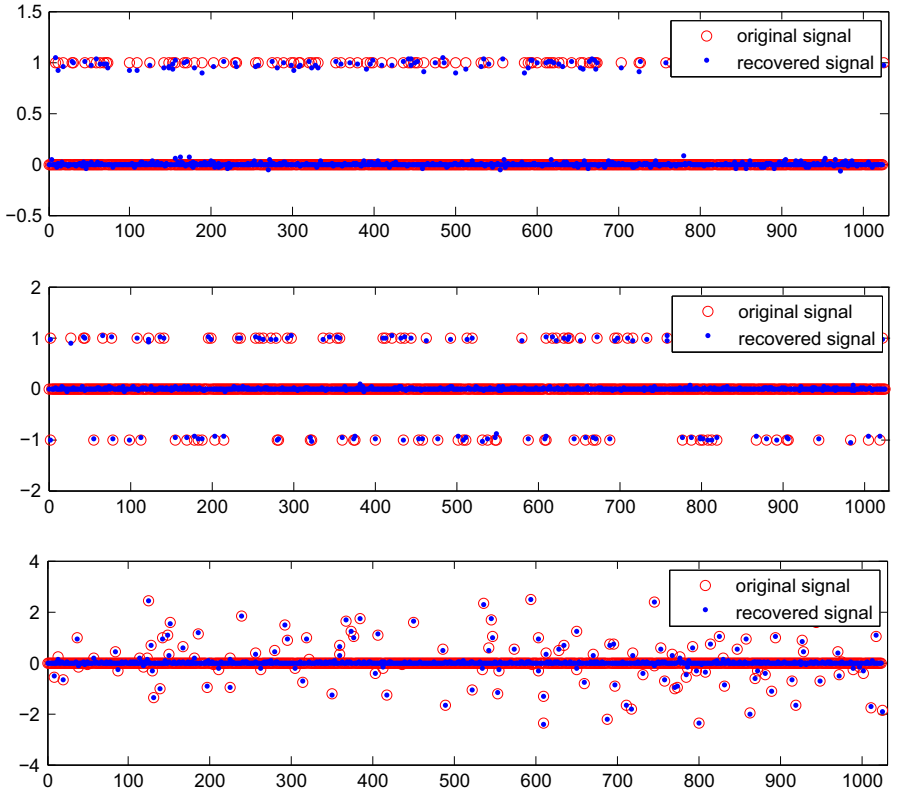

Fig. 6 Accurate recovery of three types of signals with (1024, 512, 128) using Algorithm 4 in the case of the general Gaussian sensing matrix for the noise level $\sigma = 0.01$

0.001, 0.01, 0.1, 1.0, 10 and 100, were considered. One thousand iterations of Algorithm 4 were run. Figure 1 compares the performance of Algorithm 4 for six different values of α. Figure 1 shows that when $\alpha = 0.01$, Algorithm 4 requires only approximately 550 iterations for the relative ℓ_2-error to decrease to a level of 10^{-16}. For other values of α, the levels of the relative ℓ_2-errors are smaller than 10^{-06}. Thus, we can conclude that Algorithm 4 performs best with $\alpha = 0.01$. Therefore, in our experiments, α was set to 0.01. The use of NESTA and LADMM also requires the determination of relevant parameters. It is well known that NESTA was developed by first smoothing the ℓ_1-norm and then applying an accelerated first-order technique to solve the smoothed problem. A parameter denoted by μ is used to control how close the smoothed ℓ_1-norm may be to the ℓ_1-norm. We set $\mu = 10^{-06}$ in our experiments. Another parameter, Tol, which defines the tolerance related to the stopping criterion for NESTA, will be discussed in Sect. 4.2. The three parameters α, β and γ in LADMM were optimally tuned in accordance with the work presented in [59]. The codes for NESTA and LADMM can be obtained from [48] and [62], respectively.

Table 3 Average performance of the three algorithms in the case of the partial DCT sensing matrix for the different noise levels

Method	Algorithm 4		NESTA		LADMM	
Signal	RE	Time (s)	RE	Time (s)	RE	Time (s)
$\sigma = 0.01$						
$(n, m, s) = (1024, 512, 128)$						
Type 1	3.19e−02	0.2041	5.91e−02	0.8362	9.16e−02	0.4829
Type 2	1.21e−02	0.1236	3.82e−02	0.8253	2.73e−02	0.4371
Type 3	1.01e−02	0.1312	3.18e−02	0.8101	5.82e−02	0.4015
$(n, m, s) = (4096, 2048, 512)$						
Type 1	6.72e−02	0.3917	7.25e−02	3.5728	4.11e−02	0.8401
Type 2	3.25e−02	0.5792	4.32e−02	3.6719	4.79e−02	0.8237
Type 3	1.56e−02	0.4803	3.01e−02	3.1901	9.01e−02	0.8110
$(n, m, s) = (16{,}384, 8192, 2048)$						
Type 1	5.18e−02	1.8790	5.91e−01	6.2736	4.71e−01	2.8638
Type 2	2.47e−02	1.5193	2.01e−01	6.2635	3.52e−01	2.8124
Type 3	2.18e−02	1.7745	1.00e−02	6.1628	3.12e−02	2.8628
$\sigma = 0.1$						
$(n, m, s) = (1024, 512, 128)$						
Type 1	5.91e−01	0.3303	6.29e−01	1.0257	3.47e−00	0.6381
Type 2	3.46e−01	0.2646	4.77e−01	0.9263	5.16e−01	0.6349
Type 3	2.31e−01	0.1726	2.71e−01	0.9125	2.47e−01	0.5283
$(n, m, s) = (4096, 2048, 512)$						
Type 1	3.81e−01	0.7906	9.16e−00	4.2103	5.44e−00	2.5363
Type 2	4.27e−01	0.6437	4.83e−01	4.6281	6.27e−01	2.3137
Type 3	8.19e−01	0.7204	3.51e−01	4.2375	5.24e−01	2.1273
$(n, m, s) = (16{,}384, 8192, 2048)$						
Type 1	2.14e−01	2.9713	1.01e−00	7.4724	1.27e−00	4.9125
Type 2	2.69e−01	2.6170	5.80e−01	7.3619	5.88e−00	4.2739
Type 3	1.52e−01	2.8449	2.48e−01	7.2639	6.81e−01	4.1728

4.2 Numerical Experiments

First, sparse signal recovery in the noiseless case is considered. Based on Remarks refre2 and 3, we discuss both the partial DCT sensing matrix and the orthonormalized Gaussian sensing matrix. Note that since the DCT approach permits a fast algorithm for matrix-vector multiplication, this trick can be utilized to accelerate the three algorithms when the partial DCT sensing matrix is used. Algorithm 4 and LADMM were terminated when the relative error on the recovered signal between successive iterations satisfied $||x_k - x_{k-1}||_2/||x_{k-1}||_2 < 10^{-16}$. The stopping criterion for NESTA was Tol $< 10^{-16}$. One thousand iterations were run for each of the three algorithms. Perfect recovery of signals with (1024, 512, 128), (4096, 2048, 512) and

Table 4 Average performance of Algorithm 4 and LADMM in the case of the general Gaussian sensing matrix for the various noise levels

Method	Algorithm 4		LADMM	
Signal	RE	Time (s)	RE	Time (s)
$\sigma = 0.01$				
$(n, m, s) = (1024, 512, 128)$				
Type 1	2.18e−02	3.1648	4.38e−02	3.7391
Type 2	7.66e−02	3.5701	2.56e−02	3.0127
Type 3	2.02e−02	3.6175	7.99e−02	2.5281
$(n, m, s) = (4096, 2048, 512)$				
Type 1	4.83e−02	32.5162	5.71e−02	26.5629
Type 2	8.66e−02	31.4738	9.28e−02	26.7190
Type 3	1.00e−02	31.3728	3.67e−02	25.9153
$(n, m, s) = (16{,}384, 8192, 2048)$				
Type 1	5.77e−02	58.5680	5.83e−01	50.2675
Type 2	8.36e−02	58.3648	2.22e−01	49.7193
Type 3	4.56e−02	58.2631	4.31e−02	49.3749
$\sigma = 0.1$				
$(n, m, s) = (1024, 512, 128)$				
Type 1	5.63e−01	4.5614	2.74e−00	4.0124
Type 2	2.48e−01	4.7280	2.85e−00	3.9347
Type 3	9.47e−01	4.1023	4.58e−01	3.8103
$(n, m, s) = (4096, 2048, 512)$				
Type 1	6.90e−01	34.8729	1.36e−00	29.5681
Type 2	3.62e−01	33.5628	6.85e−00	28.9201
Type 3	5.72e−01	33.7218	5.82e−01	28.1737
$(n, m, s) = (16{,}384, 8192, 2048)$				
Type 1	5.83e−01	60.5257	4.83e−00	52.7348
Type 2	3.38e−01	60.1372	5.72e−00	52.6317
Type 3	4.12e−01	60.3648	2.90e−01	52.1135

(16,384, 8192, 2048) using our algorithm is exhibited in Figs. 2, 3 and 4, respectively. Tables 1 and 2 report the average performance (accuracy and CPU time) of the three algorithms in recovering the three types of signals. It can be observed that the performance of Algorithm 4 is superior to that of either NESTA or LADMM in terms of both accuracy and CPU time.

In the following, sparse signal recovery in the noisy case is discussed. In the experiment, the partial DCT matrix and the general Gaussian matrix were used as the sensing matrices A. The measurements b were contaminated by random Gaussian noise with mean 0 and standard deviation σ generated using the MATLAB command $\sigma * randn(m, 1)$. The noise power proved to be $\varepsilon = \sqrt{m}\sigma$. We set the noise levels to $\sigma = 0.01$ and 0.1. For the partial DCT sensing matrix satisfying $AA^* = I$, we considered both NESTA and LADMM for comparison. For the general Gaussian sensing matrix, we compared our proposed algorithm only with LADMM because

NESTA is very time consuming. In addition, for the case of the general Gaussian sensing matrix, it was necessary to run the inner loop of our algorithm. In fact, we ran FISTA only 10 times in each iteration. The stopping criterion for both our algorithm and LADMM was that the relative error on the recovered signal between successive iterations must satisfy $||x_k - x_{k-1}||_2/||x_{k-1}||_2 < 10^{-04}$. The stopping criterion for NESTA was Tol $< 10^{-04}$. One thousand iterations were run for each of the three algorithms. To save space, in Figs. 5 and 6, we show only the recovery results obtained using Algorithm 4 for the signals with (1024, 512, 128) in the noisy case ($\sigma = 0.01$) with the partial DCT sensing matrix and the general Gaussian sensing matrix, respectively. It can be seen that it is slightly more difficult to recover signals of the first type than signals of the other two types. Table 3 reports the average performance of the three algorithms in the case of the partial DCT sensing matrix for the different noise levels. Table 4 reports the average performance of Algorithm 4 and LADMM in the case of the general Gaussian sensing matrix for the various noise levels. It can be seen that the performance of Algorithm 4 is superior to that of either NESTA or LADMM when the partial DCT sensing matrix is used. Furthermore, when the general Gaussian sensing matrix is used, the accuracy of the signals recovered using our proposed algorithm is slightly better than that for LADMM, although our proposed algorithm also needs slightly more time than LADMM. Hence, from the two experiments presented above, we can conclude that the proposed algorithm is a fast and efficient method of recovering sparse signals, especially when the sensing matrix A satisfies $AA^* = I$.

5 Conclusions

In this paper, the constrained convex ℓ_1-minimization problem (BP_ε) is recast as a convex composite problem with a special structure by introducing the indicator function related to the constrained set into (BP_ε). The primal Douglas–Rachford splitting method is applied to solve the resulting convex composite problem. The developed algorithm requires the computation of the proximity operator of the ℓ_1-norm and projection onto the constraint set. Since the proximity operator of the ℓ_1-norm has a closed-form solution that is very simple and requires only a few operations, the computational cost of the developed algorithm in each iteration is dominated solely by the projection. A fast and efficient method of computing the projection is proposed. In particular, when the sensing matrix A satisfies $AA^* = I$, which is often the case in compressive sensing, a closed-form solution for the projection can be derived. Compared with the popular NESTA and LADMM techniques, the developed algorithm performs better in terms of accuracy and computation for large-scale sparse signal recovery.

We also note several possible avenues of future research work. Since the performance of the proposed algorithm is sensitive to the proximal parameter α, the question of how to adaptively choose α becomes a practical and important problem. In addition, the primal Douglas–Rachford splitting method can be applied to other sparse recovery problems, such as low-rank matrix completion [14]. Moreover, because several non-convex functions [41,63,65] have been proposed to replace the ℓ_1-norm, we

can further consider whether the primal Douglas–Rachford splitting method can be applied to non-convex optimization problems in compressive sensing.

Acknowledgements The authors would like to thank the anonymous reviewers and the associate editor and editor-in chief for their constructive comments, which significantly improved this paper. This research was supported by the National Natural Science Foundation of China under the Grants 11131006, 41390450, 91330204 and 11271297 and in part supported by the National Basic Research Program of China under the Grant 2013CB329404.

References

1. F. Artacho, J. Borwein, M. Tam, Douglas–Rachford feasibility methods for matrix completion problems. ANZIAM J. **55**(4), 299–326 (2014)
2. B. Alexeev, J. Cahill, D. Mixon, Full spark frames. J. Fourier Anal. Appl. **18**(6), 1167–1194 (2012)
3. A. Beck, M. Teboulle, A fast iterative shrinkage-thresholding algorithm for linear inverse problems. SIAM J. Imaging Sci. **2**(1), 183–202 (2009)
4. H. Bauschke, P. Combettes, *Convex Analysis and Monotone Operator Theory in Hilbert Spaces* (Springer, New York, 2011). CMS Books in Mathematics
5. J. Benedetto, P. Ferreira, *Modern Sampling Theory: Mathematics and Applications* (Birkhäuser, Boston, 2001). Applied and Numerical Harmonic Analysis
6. P. Bühlmann, S. van de Geer, *Statistics for High-Dimensional Data* (Springer, Berlin, 2011). Springer Series in Statistics
7. J. Bobin, J. Starck, R. Ottensamer, Compressed sensing in astronomy. IEEE J. Sel. Top. Signal Process. **2**(5), 718–726 (2008)
8. S. Becker, J. Bobin, E. Candès, NESTA: a fast and accurate first-order method for sparse recovery. SIAM J. Imaging Sci. **4**(1), 1–39 (2011)
9. T. Blumensath, M. Davies, Sampling theorems for signals from the union of finite-dimensional linear subspaces. IEEE Trans. Inf. Theory **55**(4), 1872–1882 (2009)
10. S. Chen, D. Donoho, M. Saunders, Atomic decomposition by basis pursuit. SIAM J. Sci. Comput. **20**(1), 33–61 (1998)
11. E. Candès, J. Romberg, T. Tao, Robust uncertainty principles: exact signal reconstruction from highly incomplete frequency information. IEEE Trans. Inf. Theory **52**(2), 489–509 (2006)
12. E. Candès, J. Romberg, T. Tao, Stable signal recovery from incomplete and inaccurate measurements. Commun. Pure Appl. Math. **59**(8), 1207–1223 (2006)
13. E. Candès, T. Tao, Decoding by linear programming. IEEE Trans. Inf. Theory **51**(12), 4203–4215 (2005)
14. E. Candès, B. Recht, Exact matrix completion via convex optimization. Found. Comput. Math. **9**(6), 717–772 (2009)
15. P. Combettes, V. Wajs, Signal recovery by proximal forward–backward splitting. SIAM J. Multiscale Model. Simul. **4**(4), 1168–1200 (2005)
16. P. Combettes, J. Pesquet, A Douglas–Rachford splitting approach to nonsmooth convex variational signal recovery. IEEE J. Sel. Topics Signal Process. **1**(4), 564–574 (2007)
17. D. Donoho, Compressed sensing. IEEE Trans. Inf. Theory **52**(8), 1289–1306 (2006)
18. D. Donoho, De-noising by soft-thresholding. IEEE Trans. Inf. Theory **41**(3), 613–627 (1995)
19. A. Daneshmand, F. Facchinei, V. Kungurtsev, G. Scutari, Hybrid random/deterministic parallel algorithms for convex and nonconvex big data optimization. IEEE Trans. Signal Process. **63**(15), 3914–3929 (2015)
20. A. Daneshmand, F. Facchinei, V. Kungurtsev, G. Scutari, Flexible selective parallel algorithms for big data optimization. in *2014 48th Asilomar Conference on Signals, Systems and Computers, IEEE* (2014), pp. 3–7
21. I. Dassios, K. Fountoulakis, J. Gondzio, A second-order method for compressed sensing problems with coherent and redundant dictionaries. http://arxiv.org/abs/1405.4146, (2014)
22. I. Dassios, K. Fountoulakis, J. Gondzio, A preconditioner for a primal-dual newton conjugate gradients method for compressed sensing problems. SIAM J. Sci. Comput. **37**(6), A2783–A2812 (2015)

23. Y. Dong, Douglas–Rachford splitting method for semidefinite programming. J. Appl. Math. Comput. **51**(1), 569–591 (2016)
24. L. Demanet, X. Zhang, Eventual linear convergence of the Douglas–Rachford iteration for basis pursuit. Math. Comput. **85**, 209–238 (2015)
25. J. Douglas, H. Rachford, On the numerical solution of the heat conduction problem in 2 and 3 space variables. Trans. Am. Math. Soc. **82**, 421–439 (1956)
26. M. Duarte, M. Davenport, D. Takhar, J. Laska, S. Ting, K. Kelly, R. Baraniuk, Single-pixel imaging via compressive sampling. IEEE Signal Process. Mag. **25**(2), 83–91 (2008)
27. J. Eckstein, D. Bertsekas, On the Douglas–Rachford splitting method and the proximal point algorithm for maximal monotone operators. Math. Program. **55**(3), 293–318 (1992)
28. M. Elad, *Sparse and Redundant Representations: From Theory to Applications in Signal and Image Processing* (Springer, New York, 2010)
29. Y. Eldar, G. Kutyniok, *Compressed Sensing Theory and Applications* (Cambridge University Press, Cambridge, 2012)
30. J. Eckstein, Augmented Lagrangian and alternating direction methods for convex optimization: a tutorial and some illustrative computational results, RUTCOR Research Report (RRR) (2012)
31. S. Foucart, H. Rauhut, *A Mathematical Introduction to Compressive Sensing* (Birkhäuser, Basel, 2013)
32. M. Fazel, *Matrix Rank Minimization with Applications*. Ph.D. thesis, Stanford University (2002)
33. B. He, X. Yuan, On the convergence rate of Douglas–Rachford operator splitting method. Math. Program. **153**(2), 715–722 (2015)
34. E. Hale, W. Yin, Y. Zhang, Fixed-point continuation for ℓ_1-minimization: methodology and convergence. SIAM J. Optim. **19**(3), 1107–1130 (2008)
35. F. Herrmann, M. Friedlander, O. Yilmaz, Fighting the curse of dimensionality: compressive sensing in exploration seismology. IEEE Signal Process. Mag. **29**(3), 88–100 (2012)
36. D. Holland, M. Bostock, L. Gladden, D. Nietlispach, Fast multidimensional NMR spectroscopy using compressed sensing. Angew. Chem. Int. Ed. **50**(29), 6548–6551 (2011)
37. M. Herman, T. Strohmer, High-resolution radar via compressed sensing. IEEE Trans. Signal Process. **57**(6), 2275–2284 (2009)
38. A. Lenoir, P. Mahey, A survey on operator splitting and decomposition of convex programs. http://www.optimization-online.org/DB_FILE/2015/07/5039 (2015)
39. G. Li, T. Pong, Douglas–Rachford splitting for nonconvex feasibility problems. http://arxiv.org/abs/1409.8444v1 (2014)
40. M. Lustig, D. Donoho, J. Pauly, Sparse MRI: the application of compressed sensing for rapid MR imaging. Magn. Reson. Med. **58**, 1182–1195 (2007)
41. M. Lai, J. Wang, An unconstrained ℓ_q minimization with $0<q \leq 1$ for sparse solution of underdetermined linear systems. SIAM J. Optim. **21**(1), 82–101 (2011)
42. Z. Lu, T. Pong, Y. Zhang, An alternating direction method for finding Dantzig selectors. Comput. Stat. Data Anal. **35**(8), 4037–4046 (2012)
43. F. Masao, The primal Douglas–Rachford splitting algorithm for a class of monotone mappings with application to the traffic equilibrium problem. Math. Program. **72**(1), 1–15 (1996)
44. M. Murphy, M. Alley, J. Demmel, K. Keutzer, S. Vasanawala, M. Lustig, Fast ℓ_1-SPIRiT compressed sensing parallel imaging MRI: scalable parallel implementation and clinically feasible runtime. IEEE Trans. Med. Imaging **31**(6), 1250–1262 (2012)
45. M. Mishali, Y. Eldar, From theory to practice: sub-nyquist sampling of sparse wideband analog signals. IEEE J. Sel. Top. Signal Process. **4**(2), 375–391 (2010)
46. ℓ_1-Magic. http://www.acm.caltech.edu/l1magic/ (2006)
47. T. Ni, J. Zhai, A matrix-free smoothing algorithm for large-scale support vector machines. Inf. Sci. **358–359**(3), 29–43 (2016)
48. NESTA. http://statweb.stanford.edu/~candes/nesta/ (2011)
49. D. O'Connor, L. Vandenberghe, Primal-dual decomposition by operator splitting and applications to image deblurring. SIAM J. Imaging Sci. **7**(3), 1724–1754 (2014)
50. S. Osher, Y. Mao, B. Dong, W. Yin, Fast linearized Bregman iteration for compressive sensing and sparse denoising. Commun. Math. Sci. **8**(1), 93–111 (2010)
51. G. Pfander, H. Rauhut, J. Tanner, Identification of matrices having a sparse representation. IEEE Trans. Signal Process. **56**(11), 5376–5388 (2008)
52. Y. Pfeffer, *Compressive Sensing for Hyperspectral Imaging*. Research thesis, Israel Institute of Technology (2010)

53. H. Rauhut, G. Pfander, Sparsity in time-frequency representations. J. Fourier Anal. Appl. **16**(2), 233–260 (2010)
54. R. Rockafellar, *Convex Analysis* (Princeton University Press, Princeton, 1970)
55. G. Steidl, T. Teuber, Removing multiplicative noise by Douglas–Rachford splitting methods. J. Math. Imaging Vis. **36**(2), 168–184 (2010)
56. Y. Shrot, L. Frydman, Compressed sensing and the reconstruction of ultrafast 2D NMR data: principles and biomolecular applications. J. Magn. Reson. **209**(2), 352–358 (2011)
57. J. Tropp, J. Laska, M. Duarte, J. Romberg, R. Baraniuk, Beyond Nyquist: efficient sampling of sparse bandlimited signals. IEEE Trans. Inf. Theory **56**(1), 520–544 (2010)
58. S. Vasanawala, M. Alley, B. Hargreaves, R. Barth, J. Pauly, M. Lustig, Improved pediatric MR imaging with compressed sensing. Radiology **256**(2), 607–616 (2010)
59. J. Yang, Y. Zhang, Alternating direction algorithms for ℓ_1-problems in compressive sensing. SIAM J. Sci. Comput. **33**(1), 250–278 (2011)
60. J. Yang, X. Yuan, Linearized augmented Lagrangian and alternating direction methods for nuclear norm minimization. Math. Comput. **82**(281), 301–329 (2013)
61. W. Yin, S. Osher, D. Goldfarb, J. Darbon, Bregman iterative algorithms for ℓ_1-minimization with applications to compressed sensing. SIAM J. Imaging Sci. **1**(1), 143–168 (2008)
62. YALL1. http://yall1.blogs.rice.edu/ (2011)
63. C. Zhang, Nearly unbiased variable selection under minimax concave penalty. Ann. Stat. **38**(2), 894–942 (2010)
64. W. Zhu, S. Shu, L. Cheng, Proximity point algorithm for low-rank matrix recovery from sparse noise corrupted data. Appl. Math. Mech. **35**(2), 259–268 (2014)
65. S. Zhang, J. Xin, Minimization of transformed ℓ_1 penalty: closed form representation and iterative thresholding algorithms. arXiv preprint arXiv:1412.5240 (2014)