Hindawi
Complexity
Volume 2020, Article ID 8323865, 14 pages
https://doi.org/10.1155/2020/8323865

Research Article

A New Conjugate Gradient Projection Method for Convex Constrained Nonlinear Equations

Pengjie Liu,[1] **Jinbao Jian**,[2] **and Xianzhen Jiang**[2]

[1]*College of Mathematics and Information Science, Guangxi University, Nanning, Guangxi 530004, China*
[2]*College of Mathematics and Physics, Guangxi University for Nationalities, Nanning, Guangxi 530006, China*

Correspondence should be addressed to Xianzhen Jiang; yl2811280@163.com

Received 3 April 2020; Accepted 8 June 2020; Published 9 July 2020

Academic Editor: Chongyang Liu

The conjugate gradient projection method is one of the most effective methods for solving large-scale monotone nonlinear equations with convex constraints. In this paper, a new conjugate parameter is designed to generate the search direction, and an adaptive line search strategy is improved to yield the step size, and then, a new conjugate gradient projection method is proposed for large-scale monotone nonlinear equations with convex constraints. Under mild conditions, the proposed method is proved to be globally convergent. A large number of numerical experiments for the presented method and its comparisons are executed, which indicates that the presented method is very promising. Finally, the proposed method is applied to deal with the recovery of sparse signals.

1. Introduction

Solving a system of nonlinear equations can be transformed as an optimization problem, which is widely applied in many fields of sciences and engineering, for instance, the economic equilibrium problem [1], the neural networks problem [2], the financial problem [3, 4], the chemical equilibrium system [5], and the compressed sensing problem [6, 7].

In this paper, the following system of constrained monotone nonlinear equations is considered:

$$F(x) = 0, \quad \text{s.t. } x \in \mathscr{C}, \tag{1}$$

where $\mathscr{C}$ is a nonempty closed convex set of $\mathbb{R}^n$ and $F: \mathbb{R}^n \longrightarrow \mathbb{R}^n$ is a monotone mapping, namely,

$$(F(x) - F(y))^T (x - y) \geq 0, \quad \forall x, y \in \mathbb{R}^n. \tag{2}$$

It follows from (2) that the solution set of system (1) is convex [8]. Throughout this paper, the symbol $\|\cdot\|$ stands for the Euclidean norm. Denote $F_k := F(x_k)$ for $x_k \in \mathbb{R}^n$, $P_{\mathscr{C}}[x] = \arg\min\{\|y - x\| \mid y \in \mathscr{C}, \forall x, y \in \mathbb{R}^n\}$.

Many algorithms have been proposed to deal with (1) during the past few decades (see, e.g., [9–16]), such as the projected Newton method [9], the projected quasi-Newton method [10–13], the Levenberg–Marquardt method [14], the trust region method [15], and the Lagrangian global method [16]. As we know, these methods converge rapidly if the sufficiently good initial points are chosen. However, they are not well-suited for solving large-scale constrained nonlinear equations due to the computation of the Jacobian matrix or its approximation at each iteration. Therefore, in the past few years, the projected derivative-free method (PDFM) has become more and more popular, i.e., the spectral gradient projection method [17–19], the multivariate spectral gradient-type projection method [20, 21], and the conjugate gradient projection method (CGPM) [22–31], and more other PDFM can be seen in references [32, 33].

In this work, we concentrate on studying CGPM for large-scale nonlinear equations with convex constraints. We aim to establish a more efficient CGPM by improving the search direction and the line search rule and use the proposed CGPM to deal with the problem of sparse signals recovery.

The contributions of this article are listed as follows:

(i) To guarantee the search direction satisfying the sufficient descent condition and trust region property independent of any line searches, a new conjugate parameter is proposed;

(ii) Based on classic line searches for nonlinear equations, an adaptive line search is improved to seek suitable step size easily;

(iii) Under general assumptions, the convergence analysis of the proposed algorithm is proved;

(iv) The reported numerical experiments show that our method is promising for solving large-scale nonlinear constrained equations and handling the problem of recovering sparse signals.

The remainder of this paper is organized as follows. In Section 2, a new search direction and an adaptive line search are proposed, and the corresponding algorithm is given. The global convergence is studied in Section 3. In Section 4, the numerical experiments for the proposed algorithm and its comparisons are performed, and the corresponding results are reported. Application of the proposed algorithm in compressed sensing is introduced in Section 5. In Section 6, a conclusion for this work is given.

2. A New CGPM Algorithm

The CGPM has been attracting extensive attention, since it not only inherits the advantages of the conjugate gradient method (CGM) with a simple algorithm structure, rapid convergence, and low storage requirements but also uses no any jacobian information of the equation in practice. To the best of our knowledge, the computation cost of CGPM mainly exists in the process of generating the search direction and computing the step size. Therefore, in the following part, we design our search direction by the CGM and give an improved inexact line search to yield the step size.

2.1. The New Search Direction Yielded by CGM. It is well-known that the search direction of the classical CGM is generated by

$$d_k = \begin{cases} -F_k, & k = 0, \\ -F_k + \beta_k d_{k-1}, & k \geq 1, \end{cases} \tag{3}$$

which is decided by the conjugate parameter β_k. Usually, a different β_k leads to a different search direction.

In the recent years, many scholars have made efforts to extend the CGM to solve the large-scale nonlinear monotone equations system. For example, based on the hybrid conjugate parameter β_k^{JHJ} in [34], Sun and Liu [28] proposed a modified conjugate parameter as follows:

$$\beta_k^N = \frac{\|F_k\|^2 - \max\{0, \|F_k\|/\|F_{k-1}\|F_k^T F_{k-1}\}}{\max\{\mu\|d_{k-1}\|\|F_k\|, d_{k-1}^T y_{k-1}\}}, \quad (\mu > 1), \tag{4}$$

where $y_{k-1} = F_k - F_{k-1}$ and extended it to solve problem (1).

Recently, Tsegay et al. [35] gave a new CGM with sufficient descent property, that is,

$$\beta_k^{\mathrm{SDCG}} = \frac{\|F_k\|^2}{v\left(\|F_k\|^2 + \|d_{k-1}\|^2\right)}, \quad v > \frac{1}{2}. \tag{5}$$

Inspired by [28, 35], we propose a new hybrid formula as follows:

$$\beta_k^{\mathrm{LJJ}} = \frac{\|F_k\|^2 - \max\{0, \|F_k\|/\|F_{k-1}\|F_k^T F_{k-1}\}}{\nu\left(\|F_k\|^2 + \|d_{k-1}\|^2\right)}, \tag{6}$$

where $\nu > (1/2)$. Clearly, the parameter β_k^{LJJ} satisfies

$$0 \leq \beta_k^{\mathrm{LJJ}} \leq \frac{\|F_k\|^2}{2\nu\|F_k\|\|d_{k-1}\|} = \frac{\|F_k\|}{2\nu\|d_{k-1}\|}. \tag{7}$$

It is interesting that the search direction with β_k^{LJJ} have better theoretical properties, that is, it satisfies the sufficient descent condition and trust region property, automatically.

2.2. An Improved Adaptive Line Search Strategy. For an efficient CGPM, choosing an inexpensive line search is a key technique. To this end, many researchers try to exploit an inexact line search strategy to obtain step size with minimal cost. Zhang and Zhou [18] adopt an Armijo-type line search procedure, that is, the step size $\alpha_k = \max\{\beta\rho^i \mid i = 0, 1, 2, \cdots\}$, such that

$$-F(x_k + \alpha_k d_k)^T d_k \geq \sigma\alpha_k\|d_k\|^2, \tag{8}$$

where β is an initial guess for the step size, $\rho \in (0, 1)$, and σ is a positive constant. Li and Li [31] obtained the step size $\alpha_k = \max\{\beta\rho^i \mid i = 0, 1, 2, \cdots\}$ by the following line search:

$$-F(x_k + \alpha_k d_k)^T d_k \geq \sigma\alpha_k\|F(x_k + \alpha_k d_k)\|\|d_k\|^2, \tag{9}$$

which was originally proposed by Solodov and Svaiter [36]. Guo and Wan [26] proposed an adaptive line search, i.e., the step size $\alpha_k = \max\{\beta\rho^i \mid i = 0, 1, 2, \cdots\}$ satisfied the following inequality:

$$-F(x_k + \alpha_k d_k)^T d_k \geq \sigma\alpha_k\|d_k\|^2 \min\{1, \|F(x_k + \alpha_k d_k)\|\}. \tag{10}$$

Obviously, when x_k is far from the solutions of problem (1) and the $\|F(x_k + \alpha_k d_k)\|$ is too large, it follows from (9) that the step size α_k becomes small, which increases the computation cost. A similar case can appear for the line search (8) when x_k is close to the solution set of problem (1) and $\|d_k\|$ is too large. However, it is worth noting that the line search (10) can overcome the previously mentioned weaknesses and take advantage of line searches (8) and (9).

Inspired by [26], we introduce another new adaptive line search strategy with a disturbance factor, that is, taking $\alpha_k = \max\{\beta\rho^i \mid i = 0, 1, 2, \cdots\}$, such that

$$-F(x_k + \alpha_k d_k)^T d_k \geq \sigma\alpha_k\gamma_k\|d_k\|^2, \tag{11}$$

where $\gamma_k = ((\|F(x_k + \alpha_k d_k)\|)/(\max\{\|F(x_k + \alpha_k d_k)\|, \mu\}))$, β is an initial guess for the step size, $\sigma, \rho \in (0,1)$, and $\mu \geq 1$. Here, μ is a disturbance factor, which can adjust the size of the right side of the line search (11) and further reduce the computation cost.

Remark 1. In fact, for a given μ, $\|F(x_k + \alpha_k d_k)\|$ is too large if x_k is far away from the solution set, namely, $((\|F(x_k + \alpha_k d_k)\|)/(\max\{\|F(x_k + \alpha_k d_k)\|, \mu\})) = 1$, and then, the new line search (11) is similar to (8) in performance. Otherwise, when x_k is close to the solution set, $\|F(x_k + \alpha_k d_k)\|$ approaches 0 and so γ_k approaches $\|F(x_k + \alpha_k d_k)\|$, and then, the new line search (11) comes back to (9).

2.3. Algorithm. Based on the new search direction with β_k^{LJJ} (6) and adaptive line search (11), we describe our algorithm in detail for solving problem (1) as follows (Algorithm 1).

3. Convergence Property

In order to obtain some important properties and convergence property of Algorithm 1, the following basic assumptions are necessary.

Assumption H:

(H1) The solution set of system (1), denoted by $\mathcal{S}$, is nonempty, and the mapping F is monotone on $\mathbb{R}^n$.

(H2) The mapping F is L-Lipschitz continuous on $\mathbb{R}^n$, i.e., there exists a constant $L > 0$ such that

$$\|F(x) - F(y)\| \leq L\|x - y\|, \quad \forall x, y \in \mathbb{R}^n. \tag{12}$$

The well-known nonexpansive property of the projection operator [37] is reviewed in the following lemma.

Lemma 1 (see [37]). *Let $\Omega \subseteq \mathbb{R}^n$ be a nonempty closed convex set. Then,*

$$\|P_\Omega[x] - P_\Omega[y]\| \leq \|x - y\|, \quad \forall x, y \in \mathbb{R}^n. \tag{13}$$

Therefore, the projection operator P is L-Lipschitz continuous on $\mathbb{R}^n$.

The following lemma shows that the search direction yielded by equation in step 2 in Algorithm 1 satisfies the sufficient descent condition and possesses some important properties.

Lemma 2. *Suppose that Assumption H holds, then the search direction generated by equation in step 2 in Algorithm 1 satisfies the sufficient descent condition,*

$$F_k^T d_k \leq -\tau\|F_k\|^2, \tag{14}$$

and $\tau\|F_k\| \leq \|d_k\| \leq \widehat{\tau}\|F_k\|$, for some positive constants τ and $\widehat{\tau}$.

Proof. For $k = 0$, it is easy to know that $F_0^T d_0 = -\|F_0\|^2$ and Lemma 2 holds. To proceed, we consider the case $k \geq 1$. If $F_k^T d_{k-1} = 0$, it follows from equation in step 2 in Algorithm 1 that $F_k^T d_k = -\|F_k\|^2$. Otherwise, multiplying both sides of equation in step 2 in Algorithm 1 by F_k^T, from (6) and (7), we have

$$\begin{aligned} F_k^T d_k &= -\|F_k\|^2 + \beta_k^{\mathrm{LJJ}} F_k^T d_{k-1} \\ &\leq -\|F_k\|^2 + \beta_k^{\mathrm{LJJ}}\|F_k\|\|d_{k-1}\| \\ &\leq -\|F_k\|^2 + \frac{\|F_k\|}{2\nu\|d_{k-1}\|}\|F_k\|\|d_{k-1}\|, \\ &= -\left(1 - \frac{1}{2\nu}\right)\|F_k\|^2 =: -\tau\|F_k\|^2, \end{aligned} \tag{15}$$

which shows that the sufficient descent property (14) holds by taking $\tau := (1 - (1/2\nu))$. Again, according to

$$\|F_k\|\|d_k\|\cos(F_k, d_k) = F_k^T d_k, \tag{16}$$

the following relation holds:

$$-\|F_k\|\|d_k\| \leq F_k^T d_k \leq -\tau\|F_k\|^2, \tag{17}$$

and then, $\|d_k\| \geq \tau\|F_k\|$.

On the other hand, it follows from (7) and equation in step 2 in Algorithm 1 that

$$\begin{aligned} \|d_k\| &\leq \|F_k\| + \beta_k^{\mathrm{LJJ}}\|d_{k-1}\| \\ &\leq \|F_k\| + \frac{\|F_k\|}{2\nu\|d_{k-1}\|}\|d_{k-1}\| \\ &= \left(1 + \frac{1}{2\nu}\right)\|F_k\| =: \widehat{\tau}\|F_k\|, \end{aligned} \tag{18}$$

and the proof is completed.

The next lemma not only indicates that the line search strategy (11) is well-defined but also provides a lower bound for step size α_k. □

Lemma 3

(i) Let the sequences $\{d_k\}$ and $\{x_k\}$ be generated by Algorithm 1; then, there exists a step size satisfying the line search (11)

(ii) Suppose that Assumption H holds; then, the step size yielded by Algorithm 1 satisfies

$$\alpha_k \geq \min\left\{\beta, \frac{\tau\rho}{L + \sigma}\frac{\|F_k\|^2}{\|d_k\|^2}\right\}. \tag{19}$$

Proof

(i) Suppose that for any nonnegative integer i, (11) does not hold at the k_0-th iterate,then

$$-F\left(x_{k_0} + \beta\rho^i d_{k_0}\right)^T d_{k_0} < \sigma\beta\rho^i\left\|d_{k_0}\right\|^2 \frac{\left\|F\left(x_{k_0} + \beta\rho^i d_{k_0}\right)\right\|}{\max\left\{\left\|F\left(x_{k_0} + \beta\rho^i d_{k_0}\right)\right\|, \mu\right\}}. \tag{20}$$

Step 0: given $x_0 \in \mathscr{C}$, the constants $\varepsilon > 0, \sigma, \rho \in (0,1), \mu \geq 1, \nu > (1/2)$ and an initial stepsize β. Set $k := 0$
Step 1: if a given terminate criterion is satisfied, e.g., $\|F(x_k)\| \leq \varepsilon$, terminate. Otherwise, go to Step 2
Step 2: compute search direction d_k by $d_k = \begin{cases} -F_k := F(x_k) & k = 0 \\ -F_k + \beta_k^{\mathrm{LJJ}} d_{k-1} & k \geq 1 \end{cases}$ and yield the conjugate parameter β_k^{LJJ} by (6)
Step 3: compute the step size α_k using the adaptive line search strategy (11) and set $z_k = x_k + \alpha_k d_k$
Step 4: compute x_{k+1} by

$$x_{k+1} = P_{\mathscr{C}}[x_k - \xi_k F(z_k)],$$

where

$$\xi_k = ((F(z_k)^T (x_k - z_k))/\|F(z_k)\|^2)$$

Set $k := k + 1$, and return to Step 1

Algorithm 1: LJJ CGPM.

From the continuity of F and $\rho \in (0,1)$, let $i \longrightarrow \infty$, and it is clear that

$$-F\left(x_{k_0}\right)^T d_{k_0} \leq 0, \tag{21}$$

which contradicts (14). The proof is completed.

(ii) For the second part, it is clear that if $\alpha_k = \beta$, then (19) holds. If $\alpha_k \neq \beta$, α_k is computed by the backtracking process in Algorithm 1. Let $\widehat{\alpha}_k := \rho^{-1}\alpha_k$, and then, $\widehat{\alpha}_k$ does not satisfy (11), namely,

$$-F(\widehat{z}_k)^T d_k < \sigma\widehat{\alpha}_k \|d_k\|^2 \frac{\|F(\widehat{z}_k)\|}{\max\{\|F(\widehat{z}_k)\|, \mu\}} \leq \sigma\widehat{\alpha}_k \|d_k\|^2, \tag{22}$$

where $\widehat{z}_k = x_k + \widehat{\alpha}_k d_k$. It follows from (12), (14), and (22) that

$$\begin{aligned} \tau\|F_k\|^2 \leq -F_k^T d_k &= (F(\widehat{z}_k) - F_k)^T d_k - F(\widehat{z}_k)^T d_k \\ &< \widehat{\alpha}_k (L + \sigma)\|d_k\|^2 = \rho^{-1}\alpha_k (L + \sigma)\|d_k\|^2. \end{aligned} \tag{23}$$

Then,

$$\alpha_k > \frac{\tau\rho}{L + \sigma} \frac{\|F_k\|^2}{\|d_k\|^2}, \tag{24}$$

which completes the proof.

The following lemma is necessary to analyze the global convergence of Algorithm 1. □

Lemma 4. *Suppose that Assumptions H holds, let $\{x_k\}$ and $\{z_k\}$ be generated by Algorithm 1, and let x^* be any given solution for system (1), i.e., $x^* \in \mathcal{S}$. Then, the sequence $\{\|x_k - x^*\|\}$ is convergent, and sequences $\{x_k\}$ and $\{z_k\}$ are both bounded. Furthermore, it holds that*

$$\lim_{k \longrightarrow \infty} \alpha_k \|d_k\| = 0. \tag{25}$$

Proof. In view of the definition of z_k and (11), we know that

$$\begin{aligned} F(z_k)^T (x_k - z_k) &= -\alpha_k F(z_k)^T d_k \\ &\geq \sigma\alpha_k^2 \|d_k\|^2 \gamma_k \\ &= \sigma\|x_k - z_k\|^2 \gamma_k. \end{aligned} \tag{26}$$

On the other hand, taking Assumption (H1) and $x^* \in \mathcal{S}$ into consideration, we have

$$\begin{aligned} F(z_k)^T (x_k - x^*) &= F(z_k)^T (x_k - z_k) + F(z_k)^T (z_k - x^*) \\ &\geq F(z_k)^T (x_k - z_k) + F(x^*)^T (z_k - x^*) \\ &= F(z_k)^T (x_k - z_k). \end{aligned} \tag{27}$$

According to equation in step 4 in Algorithm 1, Assumption (H1), Lemma 1, and (26) and (27), it follows that

$$\begin{aligned} \|x_{k+1} - x^*\|^2 &= \|P_{\mathscr{C}}[x_k - \xi_k F(z_k)] - P_{\mathscr{C}}[x^*]\|^2 \\ &\leq \|x_k - \xi_k F(z_k) - x^*\|^2 \\ &= \|x_k - x^*\|^2 - 2\xi_k F(z_k)^T (x_k - x^*) + \xi_k^2 \|F(z_k)\|^2 \\ &\leq \|x_k - x^*\|^2 - 2\xi_k F(z_k)^T (x_k - z_k) + \xi_k^2 \|F(z_k)\|^2 \\ &= \|x_k - x^*\|^2 - \frac{\left(F(z_k)^T (x_k - z_k)\right)^2}{\|F(z_k)\|^2} \\ &\leq \|x_k - x^*\|^2 - \frac{\sigma^2 \|x_k - z_k\|^4}{\|F(z_k)\|^2} \gamma_k^2, \end{aligned} \tag{28}$$

which shows that the inequalities $0 \leq \|x_{k+1} - x^*\| \leq \|x_k - x^*\|$ hold, that is, the sequence $\{\|x_k - x^*\|\}$ is monotone nonincreasing and bounded below. Hence, $\{\|x_k - x^*\|\}$ is convergent. Furthermore, the boundedness of $\{x_k\}$ is obtained.

By Lemma 2, it holds that $\{d_k\}$ is bounded and so is $\{z_k\}$. Without the loss of generality, there exists a constant $\overline{M} > 0$ such that $\|F(z_k)\| \leq \overline{M}, \forall k \geq 0$. If $\max\{\|F(z_k)\|, \mu\} = \|F(z_k)\|$, then $\max\{\|F(z_k)\|, \mu\} \leq \overline{M}$; otherwise, $\max\{\|F(z_k)\|, \mu\} = \mu$. Hence, the following relation holds:

$$\frac{\gamma_k}{\|F(z_k)\|} = \frac{1}{\max\{\|F(z_k)\|, \mu\}} \geq \frac{1}{\max\{\overline{M}, \mu\}} := A. \tag{29}$$

This together with (28) implies that

$$A^2\sigma^2 \sum_{k=0}^{\infty} \|x_k - z_k\|^4 \leq \sum_{k=0}^{\infty} \left(\|x_k - x^*\|^2 - \|x_{k+1} - x^*\|^2 \right) = \|x_0 - x^*\|^2 - \lim_{k\longrightarrow\infty} \|x_{k+1} - x^*\|^2 < \infty. \tag{30}$$

Thus, this further implies $\lim_{k\longrightarrow\infty}\alpha_k\|d_k\| = \lim_{k\longrightarrow\infty}\|x_k - z_k\| = 0$, and the proof is completed.

Next, based on Assumption H and Lemmas 1–4, the global convergence of the proposed algorithm is established. □

Theorem 1. *Suppose that Assumption H holds and the sequences $\{x_k\}$ be generated by the Algorithm 1; then,*

$$\liminf_{k\longrightarrow\infty} \|F_k\| = 0. \tag{31}$$

Furthermore, the whole sequence $\{x_k\}$ converges to a solution of system (1).

Proof. First, by contradiction, suppose that relation (31) is not true; then, there exists a constant $\varepsilon > 0$ such that

$$\|F_k\| \geq \varepsilon, \quad \forall k \geq 0. \tag{32}$$

Again, the following inequality comes directly from Lemma 2:

$$\|d_k\| \geq \tau\|F_k\| \geq \tau\varepsilon, \quad \forall k \geq 0. \tag{33}$$

This together with (25) shows that

$$\lim_{k\longrightarrow\infty} \alpha_k = 0. \tag{34}$$

In addition, from Lemmas 2 and 3 (ii) and the boundedness of $\{\|d_k\|\}$, the following relation holds:

$$\alpha_k \geq \min\left\{\beta, \frac{\tau\rho}{L+\sigma}\frac{\|F_k\|^2}{\|d_k\|^2}\right\} \geq \min\left\{\beta, \frac{\tau\rho}{(L+\sigma)\tilde{\tau}^2}\right\} > 0, \tag{35}$$

which contradicts (34). Therefore, (31) is true.

Second, (31) shows that there exists an infinite index set $\mathcal{K}$ such that $\lim_{k\in\mathcal{K}}\|F_k\| = 0$. Again, the $\{x_k\}$ is bounded and $\mathcal{C}$ is a closed set, so without the loss of generality, suppose that $\lim_{k\in\mathcal{K}} x_k = \overline{x} \in \mathcal{C}$. It follows from the continuity of F that

$$\|F(\overline{x})\| = \lim_{k\in\mathcal{K}} \|F_k\| = 0, \tag{36}$$

which shows that $\overline{x} \in \mathcal{S}$.

Finally, noticing that $x^* := \overline{x}$ from Lemma 4, it follows that the sequence $\{\|x_k - \overline{x}\|\}$ is convergent, namely,

$$\lim_{k\longrightarrow\infty} \|x_k - \overline{x}\| = \lim_{k\in\mathcal{K}} \|x_k - \overline{x}\| = 0, \tag{37}$$

which implies that the whole sequence $\{x_k\}$ converges to $\overline{x} \in \mathcal{S}$, and the proof is completed. □

4. Numerical Experiments

In this section, the numerical performances of Algorithm 1 (LJJ CGPM) for solving convex constrained nonlinear equations are tested and reported by the following two subsections.

4.1. Experimental Setup. In order to illustrate the effectiveness of the LJJ CGPM, we compare it with two recent CGPMs. Specifically, eight large-scale examples are solved by the LJJ CGPM method, PDY method [22], and ATTCGP method [23] in the same calculating environment. All codes were written in Matlab R2014a and run on a DELL with 4 GB RAM memory and Windows 10 operating system.

For the LJJ CGPM, we use (6) and (11) as the conjugate parameter of search direction and the line search rule, respectively, and the parameters in the LJJ CGPM are chosen as $\rho = 0.4, \beta = 0.99, \sigma = 0.01, \mu = 1.25$, and $\nu = 2.6$. For the PDY [22] and the ATTCGP [23] methods, the search direction, line search rule, and selection of parameters are consistent with the original literature, respectively.

For all methods, the computation will be terminated when one of the following criteria are satisfied:

$$\begin{aligned} &\text{(i) } \|F_k\| \leq 10^{-6}, \\ &\text{(ii) } \|d_k\| \leq 10^{-7}, \\ &\text{or (iii) Itr} > 2000, \end{aligned} \tag{38}$$

where "Itr" refers to the total number of iterations. Define

$$F(x) = \left(f_1(x), f_2(x), \cdots, f_n(x)\right)^T, \tag{39}$$

and the tested functions are listed as follows.

Problem 1 (see Wang et al. [11]). Set $f_i(x) = e^{x_i} - 1$, for $i = 1, 2, \cdots, n$ and $\mathcal{C} = \mathbb{R}^n_+$. Obviously, Problem 1 has a unique solution $x^* = (0, 0, \cdots, 0)^T$.

Problem 2 (see Yu et al. [20]). Set

$$\begin{aligned} f_1(x) &= x_1 - e^{\cos\left((x_1+x_2)/n+1\right)}, \\ f_i(x) &= x_i - e^{\cos\left((x_{i-1}+x_i+x_{i+1})/n+1\right)}, \\ f_n(x) &= x_n - e^{\cos\left((x_{n-1}+x_n)/n+1\right)}, \end{aligned} \tag{40}$$

for $i = 2, 3, \cdots, n-1$ and $\mathcal{C} = \mathbb{R}^n_+$.

Problem 3 (see Yu et al. [20]). Set $f_i(x) = x_i - \sin|x_i - 1|$, for $i = 1, 2, \cdots, n$ and $\mathcal{C} = \{x \in \mathbb{R}^n \mid \sum_{i=1}^n x_i \leq n, x_i \geq -1, i = 1, 2, \cdots, n\}$. Obviously, Problem 3 is nonsmooth at $x = (1, \cdots, 1)^T$.

Problem 4 (see Zhou and Li [13]). Set $f_i(x) = 2x_i - \sin|x_i|$, for $i = 1, 2, \cdots, n$ and $\mathcal{C} = \mathbb{R}^n_+$.

Problem 5 (see Gao and He [29]). Set

$$\begin{aligned} f_1(x) &= 2x_1 + \sin(x_1) - 1, \\ f_i(x) &= 2x_{i-1} + 2x_i + \sin(x_i) - 1, \\ f_n(x) &= 2x_n + \sin(x_n) - 1, \end{aligned} \tag{41}$$

for $i = 2, 3, \cdots, n-1$ and $\mathscr{C} = \mathbb{R}^n_+$.

Problem 6 (see Ou and Li [19]). Set

$$\begin{aligned} f_1(x) &= e^{x_1} - 1, \\ f_i(x) &= e^{x_i} + x_{i-1} - 1, \end{aligned} \tag{42}$$

for $i = 2, 3, \cdots, n$ and $\mathscr{C} = \mathbb{R}^n_+$.

Problem 7 (see Gao and He [29]). Set $f_i(x) = (e^{x_i})^2 + 3\sin(x_i)\cos(x_i) - 1$, for $i = 1, 2, \cdots, n$ and $\mathscr{C} = \mathbb{R}^n_+$.

Problem 8 (see Gao and He [29]). Set

$$\begin{aligned} f_1(x) &= x_1 - e^{\cos((x_1+x_2)/2)}, \\ f_i(x) &= x_i - e^{\cos((x_{i-1}+x_i+x_{i+1})/i)}, \\ f_n(x) &= x_n - e^{\cos((x_{n-1}+x_n)/n)}, \end{aligned} \tag{43}$$

for $i = 2, 3, \cdots, n-1$ and $\mathscr{C} = \mathbb{R}^n_+$.

The new iterate points yielded by the quadratic program solver quadprog. m are taken from the Matlab optimization toolbox. Problems 1–8 are tested with seven initial points $x_1 = (1, 1, \cdots, 1)^T$, $x_2 = (0.1, 0.1, \cdots, 0.1)^T$, $x_3 = (1/2, 1/2^2, \cdots, 1/2^n)^T$, $x_4 = (0, 1/n, \cdots, (n-1)/n)^T$, $x_5 = (1, 1/2, \cdots, 1/n)^T$, $x_6 = (1/n, 2/n, \cdots, 1)^T$, and $x_7 = (1 - (1/n), 1 - (2/n), \cdots, 0)^T$. Here, the dimension of problems is chosen as 10^4, 5×10^4 and 10^5, respectively.

The comparison of data is listed in Tables 1–8, where "Init" means the initial point, "n" is the dimension of the problem, "NF" denotes the number of function evaluations, "Tcpu" denotes the CPU time, and "$\|F^*\|$" is the final value of $\|F_k\|$ when the program is stopped.

In addition, in order to show the numerical performance clearly, we adopt the profiles introduced by Dolan and Morè [38] to compare the performance on Tcpu, NF, and Itr, respectively. A brief explanation of the performance figures is as follows. Denote the whole set of n_p test problems by P and the set of solvers by S. Let $t_{p,s}$ be the Tcpu (or the NF or the Itr) required to solve problem $p \in P$ by solver $s \in S$, and the comparison results between different solvers are based on the performance ratio defined from

$$r_{p,s} = \frac{t_{p,s}}{\min_{s \in S} t_{p,s}}, \tag{44}$$

and the performance profile for each solver s is defined by

$$\rho_s(\tau) = \frac{1}{n_p} \text{size}\left\{p \in \mathscr{P}: \quad \log_2 r_{p,s} \le \tau\right\}, \tag{45}$$

where size A means the number of elements in the set A. Then, $\rho_s(\tau)$ is the probability for solver $s \in \mathscr{S}$ that a performance ratio $r_{p,s}$ is within a factor $\tau \in R^n$. ρ_s is the (cumulative) distribution function for the performance ratio. Clearly, the top curved shape of the method is a winner. For details about the performance profile, see [38].

4.2. Numerical Testing Reports. The specific results of the numerical tests are displayed in Tables 1–8. Their corresponding performance profiles are plotted in Figures 1–3 in terms of Tcpu, NF, and Itr, respectively. From Tables 1–8 and Figures 1–3, the following results are obtained.

(i) The LJJ CGPM, PDY, and ATTCGP can solve all test problems completely, which shows that above-mentioned three methods are effective.

(ii) From Figures 1–3, the LJJ CGPM is the top performer among the three algorithms, and this implies that the LJJ CGPM is superior to the PDY and ATTCGP totally, at least for this set of numerical experiments.

Reasonably, the advantages of the LJJ CGPM are attributed to the choice of techniques (6) and (11) for the conjugate parameter of search direction and the adaptive line search strategy. At the first glance, the parameter β_k^{LJJ} (6) and line search strategy (11) are a bit complicated; however, the LJJ CGPM is very effective for solving constrained equations. Furthermore, numerical results indicate that the proposed method is competitive to similar methods for large-scale problems.

5. Application in Compressed Sensing

In this section, the LJJ CGPM is applied to a typical compressed sensing scenario and compared with the PDY method [22] in terms of the mean of squared error (MSE), iterative number, and CPU time. The parameters for the LJJ CGPM are set as follows: $\rho = 0.55, \beta = 10, \sigma = 10^{-4}$, and $\mu = 1.25, \nu = 2.6$. Also, the parameters of the PDY method are taken from Section 4 in [22].

5.1. Compressed Sensing. Compressed sensing, also called compressed sampling or sparse sampling, is a typical signal sampling technique. In electronic engineering, especially in signal processing, compressed sensing is often used to acquire and reconstruct sparse or compressible signals.

In this section, the attention is focused on recovering the unknown vector $x_0 \in \mathbb{R}^n$ from an incomplete or disturbed observation

$$b = Ax_0 + e, \tag{46}$$

where $b \in \mathbb{R}^k$ is the observation data, $A \in \mathbb{R}^{k \times n}$ $(k \ll n)$ is a linear mapping, and $e \in \mathbb{R}^k$ is an error term. In fact, the sparse result of x_0 can be obtained by solving the following convex unconstrained optimization problem:

$$\min_{x \in \mathbb{R}^n} \tau \|x\|_1 + \frac{1}{2}\|Ax - b\|_2^2, \tag{47}$$

where parameter $\tau > 0$, $\|\cdot\|_1$ and $\|\cdot\|_2$ mean the l_1-norm and l_2-norm, respectively. There are many algorithms to solve problem (47). One of them is the gradient projection method

Table 1: Numerical results on Problem 1.

Init (n)	LJJ CGPM Itr/NF/Tcpu/$\|F^*\|$	PDY Itr/NF/Tcpu/$\|F^*\|$	ATTCGP Itr/NF/Tcpu/$\|F^*\|$
$x_1(10^4)$	8/23/0.025/3.939$e-07$	16/49/0.043/9.317$e-07$	20/62/0.047/6.199$e-07$
$x_2(10^4)$	7/19/0.016/4.751$e-07$	15/46/0.035/5.458$e-07$	18/55/0.036/6.052$e-07$
$x_3(10^4)$	10/28/0.011/4.686$e-07$	12/37/0.015/8.832$e-07$	14/43/0.017/9.585$e-07$
$x_4(10^4)$	16/43/0.030/1.287$e-08$	17/52/0.036/5.220$e-07$	25/79/0.035/7.025$e-07$
$x_5(10^4)$	11/31/0.021/4.987$e-07$	14/44/0.024/4.552$e-07$	16/50/0.021/7.025$e-07$
$x_6(10^4)$	16/43/0.030/1.291$e-08$	17/52/0.039/5.225$e-07$	25/79/0.031/5.167$e-07$
$x_7(10^4)$	16/43/0.029/1.287$e-08$	17/52/0.037/5.220$e-07$	25/79/0.034/7.025$e-07$
$x_1(5\times10^4)$	8/23/0.096/8.808$e-07$	18/57/0.229/9.221$e-07$	21/65/0.238/5.544$e-07$
$x_2(5\times10^4)$	8/21/0.078/1.062$e-08$	16/49/0.185/4.039$e-07$	19/58/0.207/5.413$e-07$
$x_3(5\times10^4)$	10/28/0.059/4.686$e-07$	12/37/0.080/8.832$e-07$	14/43/0.093/9.585$e-07$
$x_4(5\times10^4)$	16/43/0.150/2.881$e-08$	18/57/0.212/6.743$e-07$	27/86/0.187/8.395$e-07$
$x_5(5\times10^4)$	11/31/0.105/4.987$e-07$	14/44/0.129/4.552$e-07$	16/50/0.123/7.025$e-07$
$x_6(5\times10^4)$	16/43/0.151/2.883$e-08$	18/57/0.215/6.743$e-07$	27/86/0.191/8.174$e-07$
$x_7(5\times10^4)$	16/43/0.151/2.881$e-08$	18/57/0.215/6.743$e-07$	27/86/0.193/8.395$e-07$
$x_1(10^5)$	9/25/0.185/1.246$e-08$	19/63/0.500/7.471$e-07$	21/65/0.496/7.841$e-07$
$x_2(10^5)$	8/21/0.159/1.502$e-08$	16/49/0.377/5.712$e-07$	19/58/0.412/7.655$e-07$
$x_3(10^5)$	10/28/0.129/4.686$e-07$	12/37/0.193/8.832$e-07$	14/43/0.186/9.585$e-07$
$x_4(10^5)$	16/43/0.307/4.076$e-08$	18/57/0.436/9.535$e-07$	28/89/0.412/5.734$e-07$
$x_5(10^5)$	11/31/0.207/4.987$e-07$	14/44/0.251/4.552$e-07$	16/50/0.236/7.025$e-07$
$x_6(10^5)$	16/43/0.306/4.077$e-08$	18/57/0.430/9.536$e-07$	28/89/0.443/6.548$e-07$
$x_7(10^5)$	16/43/0.320/4.076$e-08$	18/57/0.429/9.535$e-07$	28/89/0.419/5.734$e-07$

Table 2: Numerical results on Problem 2.

Init (n)	LJJ CGPM Itr/NF/Tcpu/$\|F^*\|$	PDY Itr/NF/Tcpu/$\|F^*\|$	ATTCGP Itr/NF/Tcpu/$\|F^*\|$
$x_1(10^4)$	5/11/0.018/1.718$e-08$	18/55/0.078/3.324$e-07$	21/64/0.087/7.557$e-07$
$x_2(10^4)$	5/11/0.017/2.617$e-08$	20/65/0.088/3.998$e-07$	22/67/0.092/4.606$e-07$
$x_3(10^4)$	5/11/0.017/2.717$e-08$	20/65/0.091/4.150$e-07$	22/67/0.096/4.782$e-07$
$x_4(10^4)$	5/11/0.017/2.236$e-08$	19/60/0.087/5.876$e-07$	21/64/0.094/9.838$e-07$
$x_5(10^4)$	5/11/0.018/2.717$e-08$	20/65/0.092/4.149$e-07$	22/67/0.096/4.780$e-07$
$x_6(10^4)$	5/11/0.017/2.236$e-08$	19/60/0.086/5.876$e-07$	21/64/0.093/9.838$e-07$
$x_7(10^4)$	5/11/0.018/2.236$e-08$	19/60/0.086/5.876$e-07$	21/64/0.094/9.838$e-07$
$x_1(5\times10^4)$	5/11/0.085/3.842$e-08$	21/69/0.497/3.654$e-07$	22/67/0.479/6.759$e-07$
$x_2(5\times10^4)$	5/11/0.086/5.855$e-08$	23/82/0.574/7.076$e-07$	23/70/0.494/4.120$e-07$
$x_3(5\times10^4)$	5/11/0.085/6.078$e-08$	23/82/0.577/7.347$e-07$	23/70/0.502/4.277$e-07$
$x_4(5\times10^4)$	5/11/0.087/5.002$e-08$	22/77/0.561/7.324$e-07$	22/67/0.505/8.800$e-07$
$x_5(5\times10^4)$	5/11/0.099/6.078$e-08$	23/82/0.611/7.346$e-07$	23/70/0.541/4.277$e-07$
$x_6(5\times10^4)$	5/11/0.086/5.002$e-08$	22/77/0.549/7.324$e-07$	22/67/0.487/8.800$e-07$
$x_7(5\times10^4)$	5/11/0.087/5.002$e-08$	22/77/0.549/7.324$e-07$	22/67/0.490/8.800$e-07$
$x_1(10^5)$	5/11/0.171/5.434$e-08$	23/82/1.148/6.568$e-07$	22/67/1.083/9.559$e-07$
$x_2(10^5)$	5/11/0.184/8.280$e-08$	28/114/1.723/5.929$e-07$	23/70/1.219/5.826$e-07$
$x_3(10^5)$	5/11/0.188/8.596$e-08$	28/114/1.564/6.155$e-07$	23/70/1.007/6.049$e-07$
$x_4(10^5)$	5/11/0.171/7.074$e-08$	25/94/1.385/6.443$e-07$	23/70/1.045/4.978$e-07$
$x_5(10^5)$	5/11/0.180/8.596$e-08$	28/114/1.677/6.155$e-07$	23/70/1.077/6.049$e-07$
$x_6(10^5)$	5/11/0.171/7.074$e-08$	25/94/1.310/6.443$e-07$	23/70/1.023/4.978$e-07$
$x_7(10^5)$	5/11/0.178/7.074$e-08$	25/94/1.319/6.443$e-07$	23/70/1.030/4.978$e-07$

for sparse reconstruction proposed by Figueiredo et al. [39]. The first step of this method is to convert (47) into a convex quadratic programming problem.

For any vector $x \in \mathbb{R}^n$, it can be decomposed into the following two parts:

$$x = u - v, \quad u \geq 0,\ v \geq 0, \tag{48}$$

where $u_i = \max\{x_i, 0\}$ and $v_i = \max\{-x_i, 0\}$ for all $i = 1, 2, \cdots, n$. In this way, the l_1−norm of a vector can be expressed as $\|\cdot\|_1 = e_n^T u + e_n^T v$, where $e_n^T = (1, 1, \cdots, 1) \in \mathbb{R}^n$. Therefore, problem (47) can be expressed as the following convex quadratic programming problem with nonnegative constraints

$$\min_{u,v} \frac{1}{2}\|b - A(u - v)\|_2^2 + \tau e_n^T u + \tau e_n^T v, \tag{49}$$

where $u \geq 0, v \geq 0$. Furthermore, it is easy to get the following standard form:

Table 3: Numerical results on Problem 3.

Init (n)	LJJ CGPM Itr/NF/Tcpu/$\|F^*\|$	PDY Itr/NF/Tcpu/$\|F^*\|$	ATTCGP Itr/NF/Tcpu/$\|F^*\|$
$x_1(10^4)$	14/43/0.013/4.708$e-07$	6/19/0.006/8.620$e-08$	17/69/0.016/4.344$e-07$
$x_2(10^4)$	14/43/0.011/5.456$e-07$	17/68/0.021/7.903$e-07$	17/69/0.016/4.628$e-07$
$x_3(10^4)$	14/43/0.012/7.328$e-07$	18/72/0.017/4.221$e-07$	17/69/0.016/6.140$e-07$
$x_4(10^4)$	14/43/0.012/3.474$e-07$	19/76/0.019/4.688$e-07$	16/65/0.017/9.030$e-07$
$x_5(10^4)$	14/43/0.011/7.314$e-07$	18/72/0.018/4.221$e-07$	17/69/0.016/6.128$e-07$
$x_6(10^4)$	14/43/0.011/3.474$e-07$	19/76/0.019/4.690$e-07$	16/65/0.015/9.029$e-07$
$x_7(10^4)$	14/43/0.013/3.474$e-07$	19/76/0.018/4.688$e-07$	16/65/0.014/9.030$e-07$
$x_1(5\times10^4)$	15/46/0.058/2.722$e-07$	19/77/0.095/6.950$e-07$	17/69/0.071/9.714$e-07$
$x_2(5\times10^4)$	15/46/0.053/3.154$e-07$	18/72/0.083/6.418$e-07$	18/73/0.086/3.373$e-07$
$x_3(5\times10^4)$	15/46/0.058/4.238$e-07$	18/72/0.079/9.439$e-07$	18/73/0.076/4.476$e-07$
$x_4(5\times10^4)$	14/43/0.054/7.768$e-07$	20/80/0.096/3.801$e-07$	17/69/0.072/6.582$e-07$
$x_5(5\times10^4)$	15/46/0.055/4.236$e-07$	18/72/0.080/9.438$e-07$	18/73/0.078/4.474$e-07$
$x_6(5\times10^4)$	14/43/0.053/7.768$e-07$	20/80/0.092/3.801$e-07$	17/69/0.072/6.582$e-07$
$x_7(5\times10^4)$	14/43/0.056/7.768$e-07$	20/80/0.091/3.801$e-07$	17/69/0.079/6.582$e-07$
$x_1(10^5)$	15/46/0.114/3.849$e-07$	19/77/0.191/9.829$e-07$	18/73/0.164/4.478$e-07$
$x_2(10^5)$	15/46/0.122/4.461$e-07$	19/77/0.189/7.449$e-07$	18/73/0.176/4.770$e-07$
$x_3(10^5)$	15/46/0.119/5.993$e-07$	19/77/0.196/9.210$e-07$	18/73/0.165/6.330$e-07$
$x_4(10^5)$	15/46/0.124/2.841$e-07$	19/77/0.190/6.666$e-07$	17/69/0.164/9.308$e-07$
$x_5(10^5)$	15/46/0.121/5.992$e-07$	19/77/0.202/9.209$e-07$	18/73/0.172/6.328$e-07$
$x_6(10^5)$	15/46/0.121/2.841$e-07$	19/77/0.190/6.666$e-07$	17/69/0.163/9.308$e-07$
$x_7(10^5)$	15/46/0.118/2.841$e-07$	19/77/0.200/6.666$e-07$	17/69/0.162/9.308$e-07$

Table 4: Numerical results on Problem 4.

Init (n)	LJJ CGPM Itr/NF/Tcpu/$\|F^*\|$	PDY Itr/NF/Tcpu/$\|F^*\|$	ATTCGP Itr/NF/Tcpu/$\|F^*\|$
$x_1(10^4)$	7/18/0.007/7.770$e-08$	17/52/0.017/7.653$e-07$	20/61/0.019/8.152$e-07$
$x_2(10^4)$	4/9/0.003/8.351$e-08$	15/46/0.015/5.283$e-07$	18/55/0.016/6.852$e-07$
$x_3(10^4)$	7/17/0.005/4.317$e-08$	12/37/0.010/8.833$e-07$	15/46/0.012/5.849$e-07$
$x_4(10^4)$	10/26/0.008/9.622$e-07$	17/52/0.015/4.108$e-07$	20/61/0.019/5.400$e-07$
$x_5(10^4)$	8/21/0.006/6.150$e-08$	13/40/0.011/7.409$e-07$	16/49/0.014/4.741$e-07$
$x_6(10^4)$	10/26/0.008/9.639$e-07$	17/52/0.017/4.109$e-07$	20/61/0.016/5.401$e-07$
$x_7(10^4)$	10/26/0.008/9.622$e-07$	17/52/0.015/4.108$e-07$	20/61/0.016/5.400$e-07$
$x_1(5\times10^4)$	7/18/0.025/1.737$e-07$	19/60/0.079/5.509$e-07$	21/64/0.085/7.292$e-07$
$x_2(5\times10^4)$	4/9/0.015/1.867$e-07$	16/49/0.064/3.907$e-07$	19/58/0.065/6.128$e-07$
$x_3(5\times10^4)$	7/17/0.020/4.317$e-08$	12/37/0.038/8.833$e-07$	15/46/0.045/5.849$e-07$
$x_4(5\times10^4)$	11/28/0.037/2.153$e-08$	17/52/0.065/9.187$e-07$	21/64/0.081/4.830$e-07$
$x_5(5\times10^4)$	8/21/0.025/6.151$e-08$	13/40/0.051/7.410$e-07$	16/49/0.055/4.742$e-07$
$x_6(5\times10^4)$	11/28/0.038/2.154$e-08$	17/52/0.071/9.187$e-07$	21/64/0.097/4.830$e-07$
$x_7(5\times10^4)$	11/28/0.035/2.153$e-08$	17/52/0.067/9.187$e-07$	21/64/0.077/4.830$e-07$
$x_1(10^5)$	7/18/0.050/2.457$e-07$	20/65/0.168/4.909$e-07$	22/67/0.162/4.125$e-07$
$x_2(10^5)$	4/9/0.032/2.641$e-07$	16/49/0.132/5.525$e-07$	19/58/0.143/8.667$e-07$
$x_3(10^5)$	7/17/0.040/4.317$e-08$	12/37/0.087/8.833$e-07$	15/46/0.096/5.849$e-07$
$x_4(10^5)$	11/28/0.082/3.045$e-08$	19/60/0.179/4.622$e-07$	21/64/0.174/6.831$e-07$
$x_5(10^5)$	8/21/0.062/6.151$e-08$	13/40/0.106/7.410$e-07$	16/49/0.118/4.742$e-07$
$x_6(10^5)$	11/28/0.077/3.046$e-08$	19/60/0.167/4.622$e-07$	21/64/0.172/6.831$e-07$
$x_7(10^5)$	11/28/0.071/3.045$e-08$	19/60/0.158/4.622$e-07$	21/64/0.161/6.831$e-07$

Table 5: Numerical results on Problem 5.

Init (n)	LJJ CGPM Itr/NF/Tcpu/$\|F^*\|$	PDY Itr/NF/Tcpu/$\|F^*\|$	ATTCGP Itr/NF/Tcpu/$\|F^*\|$
$x_1(10^4)$	19/77/0.060/7.476$e-07$	27/145/0.101/8.743$e-07$	152/787/0.578/7.450$e-07$
$x_2(10^4)$	23/93/0.069/7.916$e-07$	42/226/0.158/8.372$e-07$	204/1047/0.744/6.883$e-07$
$x_3(10^4)$	19/77/0.057/5.319$e-07$	28/151/0.103/5.112$e-07$	170/876/0.610/8.184$e-07$
$x_4(10^4)$	20/81/0.062/9.327$e-07$	27/144/0.103/5.294$e-07$	196/1008/0.699/6.472$e-07$
$x_5(10^4)$	20/81/0.061/5.548$e-07$	26/138/0.098/8.101$e-07$	185/952/0.653/7.283$e-07$

Table 5: Continued.

Init (n)	LJJ CGPM Itr/NF/Tcpu/$\|F^*\|$	PDY Itr/NF/Tcpu/$\|F^*\|$	ATTCGP Itr/NF/Tcpu/$\|F^*\|$
$x_6(10^4)$	20/81/0.062/9.324$e-07$	26/138/0.101/8.724$e-07$	196/1008/0.685/6.471$e-07$
$x_7(10^4)$	19/77/0.057/5.444$e-07$	26/138/0.096/6.535$e-07$	160/827/0.566/6.203$e-07$
$x_1(5\times10^4)$	19/77/0.296/8.171$e-07$	25/133/0.487/5.286$e-07$	141/732/2.615/9.222$e-07$
$x_2(5\times10^4)$	21/85/0.321/7.117$e-07$	26/139/0.503/5.201$e-07$	194/997/3.551/8.547$e-07$
$x_3(5\times10^4)$	19/77/0.294/4.696$e-07$	27/146/0.528/9.869$e-07$	171/882/3.184/5.748$e-07$
$x_4(5\times10^4)$	19/77/0.297/9.870$e-07$	31/167/0.613/6.829$e-07$	185/953/3.447/8.111$e-07$
$x_5(5\times10^4)$	19/77/0.294/8.999$e-07$	28/150/0.543/6.333$e-07$	175/902/3.215/8.894$e-07$
$x_6(5\times10^4)$	19/77/0.293/9.870$e-07$	31/167/0.612/6.807$e-07$	185/953/3.397/8.111$e-07$
$x_7(5\times10^4)$	19/77/0.293/5.847$e-07$	26/138/0.505/9.469$e-07$	151/782/2.788/7.611$e-07$
$x_1(10^5)$	19/77/0.598/8.828$e-07$	30/162/1.216/4.748$e-07$	146/758/5.586/5.631$e-07$
$x_2(10^5)$	20/81/0.635/8.295$e-07$	25/133/0.993/9.282$e-07$	185/952/6.975/9.531$e-07$
$x_3(10^5)$	19/77/0.596/4.759$e-07$	26/137/1.025/6.434$e-07$	162/837/6.107/6.288$e-07$
$x_4(10^5)$	19/77/0.595/8.785$e-07$	27/144/1.100/8.273$e-07$	176/908/6.668/9.015$e-07$
$x_5(10^5)$	19/77/0.598/8.971$e-07$	29/156/1.160/4.997$e-07$	168/867/6.344/9.717$e-07$
$x_6(10^5)$	19/77/0.604/8.784$e-07$	27/143/1.074/7.645$e-07$	176/908/6.682/9.015$e-07$
$x_7(10^5)$	19/77/0.604/6.266$e-07$	27/143/1.072/6.767$e-07$	143/742/5.435/8.344$e-07$

Table 6: Numerical results on Problem 6.

Init (n)	LJJ CGPM Itr/NF/Tcpu/$\|F^*\|$	PDY Itr/NF/Tcpu/$\|F^*\|$	ATTCGP Itr/NF/Tcpu/$\|F^*\|$
$x_1(10^4)$	36/110/0.204/9.949$e-07$	29/117/0.075/9.510$e-07$	24/97/0.057/9.931$e-07$
$x_2(10^4)$	11/34/0.026/5.671$e-07$	25/101/0.063/7.366$e-07$	25/101/0.058/8.628$e-07$
$x_3(10^4)$	37/112/0.051/7.576$e-07$	52/209/0.090/9.275$e-07$	34/137/0.056/8.435$e-07$
$x_4(10^4)$	20/62/0.041/7.165$e-07$	17/69/0.047/7.590$e-07$	16/65/0.032/2.846$e-07$
$x_5(10^4)$	39/118/0.069/7.898$e-07$	60/241/0.127/9.418$e-07$	35/141/0.074/9.479$e-07$
$x_6(10^4)$	20/62/0.042/7.506$e-07$	17/69/0.047/7.591$e-07$	16/65/0.029/2.849$e-07$
$x_7(10^4)$	35/107/0.067/7.787$e-07$	31/125/0.076/8.371$e-07$	31/125/0.055/6.861$e-07$
$x_1(5\times10^4)$	37/113/0.376/6.935$e-07$	30/123/0.413/7.657$e-07$	23/93/0.288/8.910$e-07$
$x_2(5\times10^4)$	12/37/0.134/2.594$e-07$	25/101/0.333/6.388$e-07$	25/101/0.309/7.727$e-07$
$x_3(5\times10^4)$	37/112/0.281/7.576$e-07$	52/209/0.490/9.275$e-07$	34/137/0.311/8.435$e-07$
$x_4(5\times10^4)$	17/53/0.190/7.679$e-07$	18/73/0.256/5.629$e-07$	16/65/0.160/5.993$e-07$
$x_5(5\times10^4)$	39/118/0.365/7.898$e-07$	60/241/0.678/9.417$e-07$	35/141/0.389/9.479$e-07$
$x_6(5\times10^4)$	17/53/0.188/6.669$e-07$	18/73/0.259/5.629$e-07$	16/65/0.158/5.995$e-07$
$x_7(5\times10^4)$	35/107/0.352/9.085$e-07$	29/117/0.390/9.530$e-07$	30/121/0.286/8.457$e-07$
$x_1(10^5)$	37/113/0.783/7.173$e-07$	31/130/0.890/8.423$e-07$	23/93/0.589/7.944$e-07$
$x_2(10^5)$	12/37/0.280/3.661$e-07$	24/97/0.661/9.622$e-07$	25/101/0.628/7.387$e-07$
$x_3(10^5)$	37/112/0.558/7.576$e-07$	52/209/1.006/9.275$e-07$	34/137/0.639/8.435$e-07$
$x_4(10^5)$	17/53/0.377/6.159$e-07$	19/79/0.566/7.772$e-07$	16/65/0.334/8.412$e-07$
$x_5(10^5)$	39/118/0.732/7.898$e-07$	60/241/1.351/9.417$e-07$	35/141/0.785/9.479$e-07$
$x_6(10^5)$	17/53/0.374/6.332$e-07$	19/79/0.572/7.772$e-07$	16/65/0.335/8.413$e-07$
$x_7(10^5)$	35/107/0.698/9.072$e-07$	30/123/0.816/8.816$e-07$	30/121/0.599/7.940$e-07$

Table 7: Numerical results on Problem 7.

Init (n)	LJJ CGPM Itr/NF/Tcpu/$\|F^*\|$	PDY Itr/NF/Tcpu/$\|F^*\|$	ATTCGP Itr/NF/Tcpu/$\|F^*\|$
$x_1(10^4)$	1/4/0.007/0.000$e+00$	13/66/0.081/2.258$e-07$	19/116/0.142/8.849$e-07$
$x_2(10^4)$	12/49/0.061/2.701$e-07$	12/61/0.077/6.969$e-07$	17/103/0.122/8.750$e-07$
$x_3(10^4)$	10/41/0.026/5.307$e-07$	11/56/0.030/4.009$e-07$	14/85/0.041/8.584$e-07$
$x_4(10^4)$	15/62/0.072/2.133$e-07$	16/83/0.086/4.425$e-07$	26/159/0.091/4.365$e-07$
$x_5(10^4)$	15/59/0.066/5.118$e-07$	16/84/0.088/4.508$e-07$	18/106/0.101/8.553$e-07$
$x_6(10^4)$	15/62/0.072/2.138$e-07$	16/83/0.085/4.515$e-07$	26/155/0.087/9.530$e-07$
$x_7(10^4)$	15/62/0.073/2.133$e-07$	16/83/0.085/4.425$e-07$	26/159/0.089/4.365$e-07$
$x_1(5\times10^4)$	1/4/0.032/0.000$e+00$	15/81/0.524/3.287$e-07$	20/122/0.733/6.965$e-07$
$x_2(5\times10^4)$	12/49/0.308/6.039$e-07$	13/66/0.409/3.777$e-07$	18/109/0.629/6.887$e-07$
$x_3(5\times10^4)$	10/41/0.115/5.307$e-07$	11/56/0.157/4.009$e-07$	14/85/0.220/8.584$e-07$

Table 7: Continued.

Init (n)	LJJ CGPM Itr/NF/Tcpu/$\|F^*\|$	PDY Itr/NF/Tcpu/$\|F^*\|$	ATTCGP Itr/NF/Tcpu/$\|F^*\|$
$x_4\,(5\times10^4)$	15/62/0.371/4.775$e-07$	15/78/0.454/5.902$e-07$	14/92/0.297/0.000$e+00$
$x_5\,(5\times10^4)$	15/59/0.317/5.108$e-07$	16/84/0.487/4.212$e-07$	18/106/0.497/8.553$e-07$
$x_6\,(5\times10^4)$	15/62/0.375/4.777$e-07$	15/78/0.470/5.902$e-07$	15/97/0.344/0.000$e+00$
$x_7\,(5\times10^4)$	15/62/0.413/4.775$e-07$	15/78/0.575/5.902$e-07$	14/92/0.319/0.000$e+00$
$x_1\,(10^5)$	1/4/0.065/0.000$e+00$	15/81/1.075/4.648$e-07$	20/122/1.634/9.850$e-07$
$x_2\,(10^5)$	12/49/0.642/8.540$e-07$	13/66/1.206/5.342$e-07$	18/109/1.369/9.740$e-07$
$x_3\,(10^5)$	10/41/0.301/5.307$e-07$	11/56/0.390/4.009$e-07$	14/85/0.484/8.584$e-07$
$x_4\,(10^5)$	15/62/0.765/6.754$e-07$	15/78/0.923/8.346$e-07$	18/158/0.959/0.000$e+00$
$x_5\,(10^5)$	15/59/0.627/5.107$e-07$	16/84/0.893/4.176$e-07$	18/106/0.994/8.553$e-07$
$x_6\,(10^5)$	15/62/0.752/6.755$e-07$	15/78/0.926/8.346$e-07$	23/158/0.999/0.000$e+00$
$x_7\,(10^5)$	15/62/0.759/6.754$e-07$	15/78/0.962/8.346$e-07$	18/158/0.941/0.000$e+00$

Table 8: Numerical results on Problem 8.

Init (n)	LJJ CGPM Itr/NF/Tcpu/$\|F^*\|$	PDY Itr/NF/Tcpu/$\|F^*\|$	ATTCGP Itr/NF/Tcpu/$\|F^*\|$
$x_1\,(10^4)$	26/78/0.111/7.185$e-07$	25/99/0.130/7.229$e-07$	25/90/0.117/2.263$e-07$
$x_2\,(10^4)$	25/76/0.106/9.350$e-07$	25/102/0.137/4.114$e-07$	25/90/0.127/2.314$e-07$
$x_3\,(10^4)$	27/82/0.126/2.437$e-07$	25/102/0.132/7.922$e-07$	25/90/0.120/2.314$e-07$
$x_4\,(10^4)$	27/82/0.116/7.926$e-07$	22/87/0.119/8.501$e-07$	25/90/0.123/2.321$e-07$
$x_5\,(10^4)$	27/82/0.115/4.741$e-07$	25/102/0.136/4.135$e-07$	25/90/0.121/2.312$e-07$
$x_6\,(10^4)$	27/82/0.118/8.941$e-07$	22/87/0.115/8.500$e-07$	25/90/0.121/2.321$e-07$
$x_7\,(10^4)$	28/84/0.119/3.899$e-07$	22/87/0.115/9.315$e-07$	25/90/0.117/2.265$e-07$
$x_1\,(5\times10^4)$	28/84/0.604/1.713$e-07$	25/102/0.691/9.284$e-07$	26/93/0.638/2.269$e-07$
$x_2\,(5\times10^4)$	29/88/0.622/2.518$e-07$	28/118/0.791/4.552$e-07$	26/93/0.637/2.296$e-07$
$x_3\,(5\times10^4)$	28/85/0.599/3.829$e-07$	28/118/0.793/5.796$e-07$	26/93/0.638/2.296$e-07$
$x_4\,(5\times10^4)$	28/85/0.605/2.585$e-07$	27/113/0.767/6.928$e-07$	26/93/0.637/2.299$e-07$
$x_5\,(5\times10^4)$	27/82/0.584/9.499$e-07$	27/114/0.769/9.214$e-07$	26/93/0.642/2.295$e-07$
$x_6\,(5\times10^4)$	28/85/0.608/2.580$e-07$	27/113/0.764/6.927$e-07$	26/93/0.641/2.299$e-07$
$x_7\,(5\times10^4)$	26/78/0.555/4.908$e-07$	28/116/0.792/4.679$e-07$	26/93/0.641/2.271$e-07$
$x_1\,(10^5)$	26/78/1.127/8.270$e-07$	26/109/1.495/5.490$e-07$	26/93/1.302/2.271$e-07$
$x_2\,(10^5)$	27/81/1.160/4.953$e-07$	31/141/1.881/7.790$e-07$	27/96/1.320/2.292$e-07$
$x_3\,(10^5)$	28/84/1.218/2.377$e-07$	31/141/1.874/5.803$e-07$	27/96/1.338/2.292$e-07$
$x_4\,(10^5)$	28/85/1.212/6.599$e-07$	33/141/1.895/3.126$e-07$	27/96/1.332/2.294$e-07$
$x_5\,(10^5)$	28/84/1.230/3.886$e-07$	30/137/1.810/8.937$e-07$	27/96/1.333/2.291$e-07$
$x_6\,(10^5)$	28/85/1.233/6.600$e-07$	33/141/1.884/3.132$e-07$	27/96/1.336/2.294$e-07$
$x_7\,(10^5)$	25/75/1.111/8.550$e-07$	28/121/1.715/7.546$e-07$	27/96/1.391/2.273$e-07$

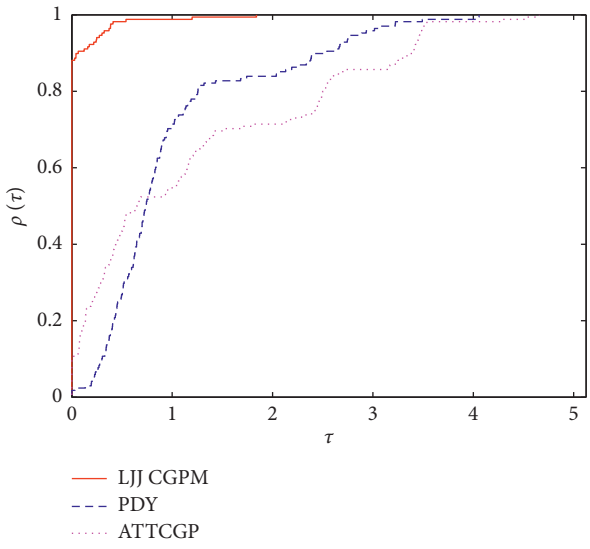

Figure 1: Performance profiles on Tcpu.

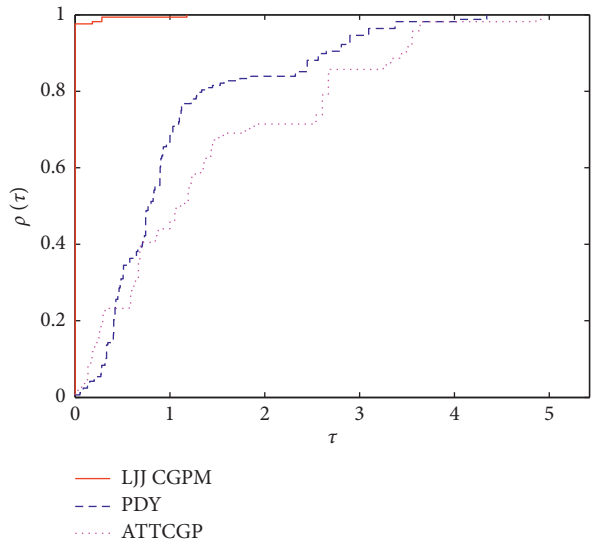

Figure 2: Performance profiles on NF.

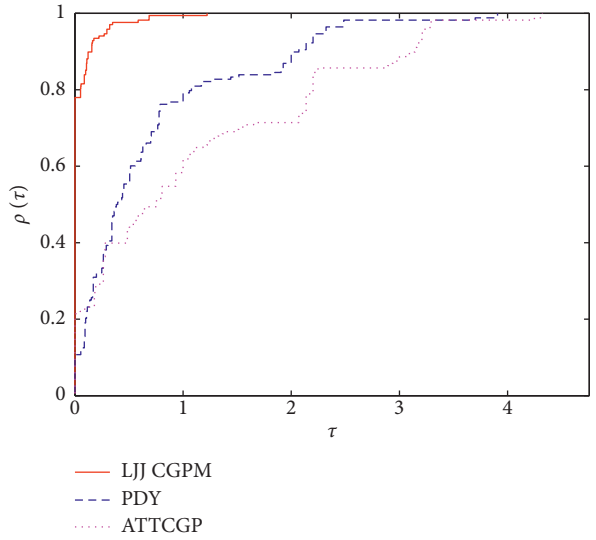

Figure 3: Performance profiles on Itr.

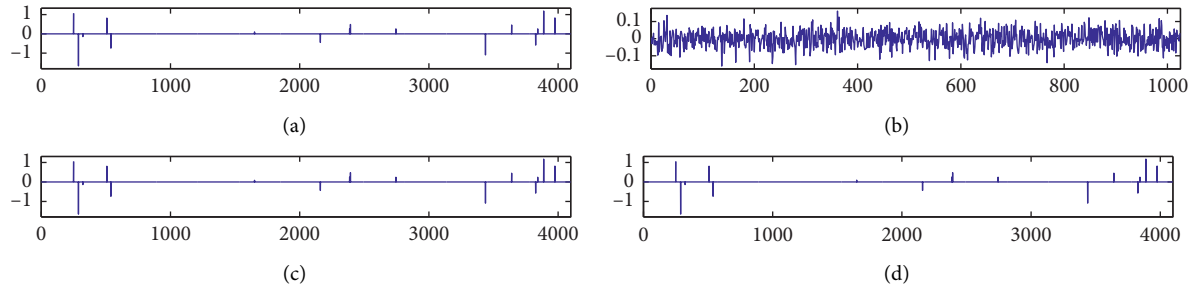

Figure 4: From top to bottom: the original signal, the measurement, and the reconstructed signal by two algorithms. (a) Original ($n = 4096$, number of nonzeros = 16), (b) measurement, (c) LJJ CGPM (MSE = $3.81e-07$, Iter = 42, time = 0.66 s), and (d) PDY (MSE = $3.21e-07$, Iter = 61, time = 1.20 s).

$$\min_{z\geq 0} \frac{1}{2} z^T H z + c^T z, \tag{50}$$

where

$$z = \begin{bmatrix} u \\ v \end{bmatrix},$$
$$c = \begin{bmatrix} \tau e_n - A^T b \\ \tau e_n + A^T b \end{bmatrix}, \tag{51}$$
$$H = \begin{bmatrix} A^T A & -A^T A \\ -A^T A & A^T A \end{bmatrix}.$$

Obviously, H is a semipositive definite matrix. So, problem (50) is a convex quadratic programming problem. In [40], it has been proven to be equivalent to

$$F(z) = \min\{z, Hz + c\} = 0, \quad z \geq 0, \tag{52}$$

where the function F is vector valued, and the "min" is explained as componentwise minimum. From [40] Lemma 3, and [41] Lemma 2.2, we know that F: $\mathbb{R}^{2n} \longrightarrow \mathbb{R}^{2n}$ is Lipschitz continuous and monotone. So, equation (52) can be solved by the LJJ CGPM.

5.2. Numerical Results. In these experiments, the main goal is to recover a one-dimensional sparse signal from its limited measurements with Gaussian noise, that is, to reconstruct a n–length sparse signal from $k\,(k \ll n)$ observations. The Gaussian noise distributed with $N(0, 10^{-4})$ is used in the test. $n = 2^{12}$ and $k = 2^{10}$ are taken for equation (46). The quality of restoration is measured by the MSE, namely,

$$\text{MSE} = \frac{1}{n}\left\|x^* - x_0\right\|, \tag{53}$$

where x^* is the restored signal and x_0 is the original signal. Besides, the original signal x_0 contains 2^4 nonzero elements randomly. The random matrix A is given by the command rand (n, k) in Matlab, and the observed data b is obtained by $b = Ax_0 + e$, where e is the Gaussian noise. $f(x) = \tau\|x\|_1 + (1/2)\|Ax - b\|_2^2$ denotes the merit function, and the value τ is obtained by the same continuation technique for the abovementioned two algorithms (LJJ CGPM and PDY method).

The iterative process starts at the measurement signal, that is, $x_0 = A^T b$, and terminates when the relative change between successive iterative falls below 10^{-5}, that is,

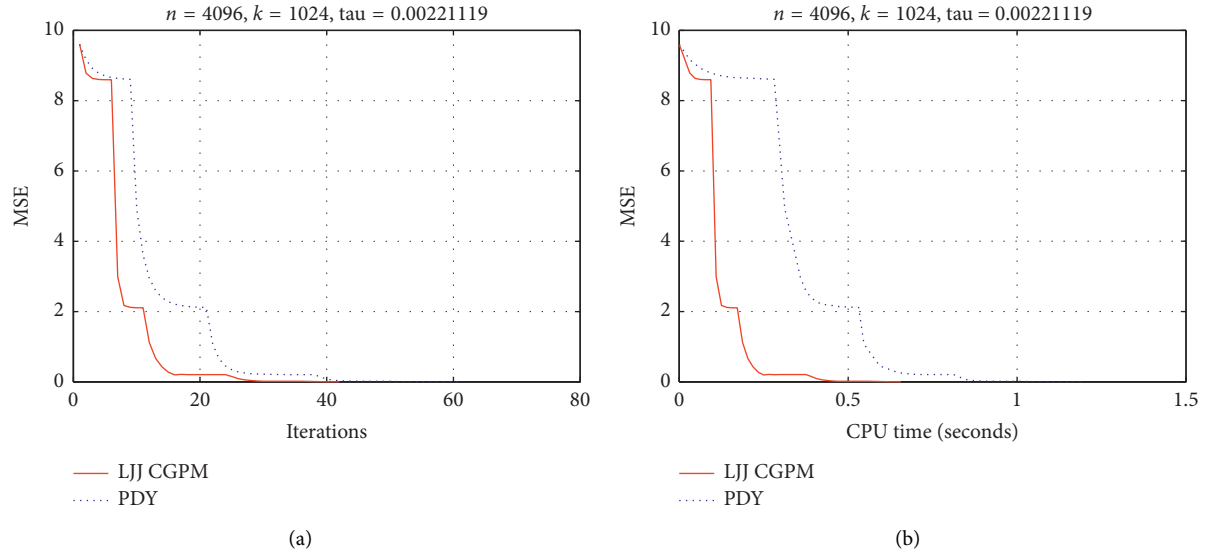

Figure 5: Plots of MSE versus iteration and CPU time.

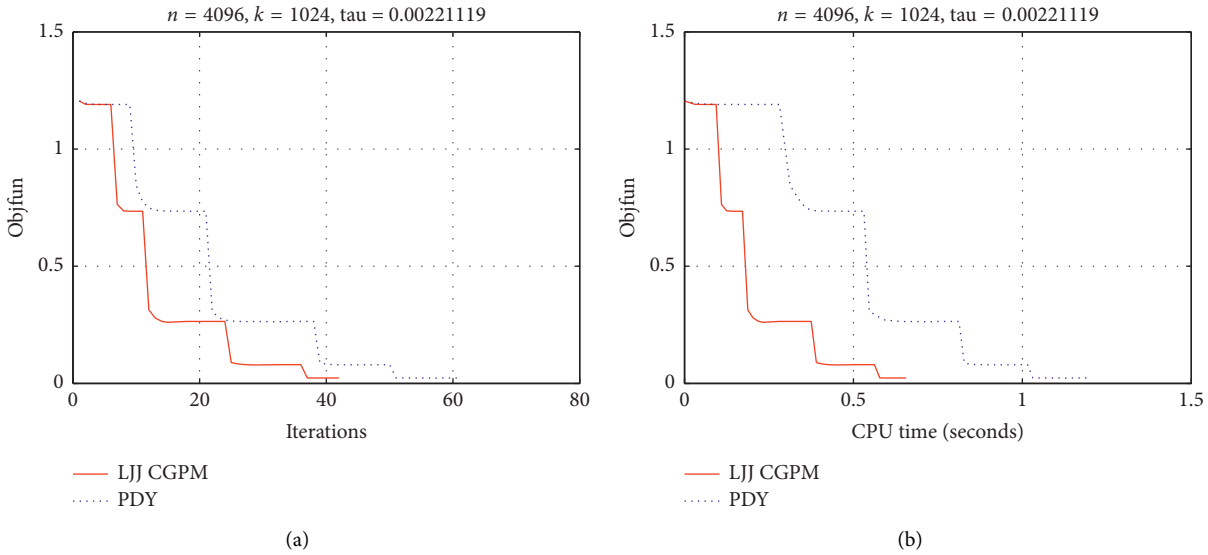

Figure 6: Plots of the objective function value versus iteration and CPU time.

$$\frac{\|f(x_k) - f(x_{k-1})\|}{\|f(x_{k-1})\|} < 10^{-5}. \tag{54}$$

Figure 4 gives the original signal x_0, the measurement b, and the signal x^* reconstructed by the LJJ CGPM and PDY method. It can be seen from Figure 4 that the LJJ CGPM and PDY method almost completely recover the disturbed signal. Moreover, to visualize the performance of the two algorithms, Figures 5 and 6 are plotted, which describe the curves of MSE and objective function values over iterations and CPU time (seconds), respectively. As can be seen from Figures 5 and 6, the LJJ CGPM has advantages over the PDY method in the MSE and objective function values, indeed, and the red curves (LJJ CGPM) drop faster than the blue curves (PDY method). Overall, simple tests show that the LJJ CGPM can effectively decode sparse signals in compressed sensing.

6. Conclusions

In this work, based on the research studies for the CGMs and some common inexact line searches, a modified conjugate parameter and an adaptive line search strategy are proposed, and then, an effective CGPM is presented for monotone nonlinear equations. Besides, the search direction of our proposed algorithm possesses the properties of sufficient descent and trust region independent of any line search techniques. Furthermore, the presented method inherits the advantages of the CGM with a simple iterative form, rapid

convergence, being derivative-free, and low memory requirements. Under some suitable conditions, the global convergence of the proposed method is established, and its numerical experiments are conducted and reported, which show that our method is very effective in dealing with large-scale monotone nonlinear equations with convex constraints and the ℓ_1−norm regularization problem in compressive sensing.

Regrettably, the parameter μ in a new adaptive line search strategy is fixed. Along with the existing line search rule, our future work is to develop the line search (11) by a dynamic adjustment technology instead of the constant μ. Moreover, summarizing the abovementioned observations and from the fact that the search direction is yielded by the CGM, how to improve the line search rule by combining with the frequently-used inexact line searches for the CGM deserves to be studied.

Data Availability

No data were used to support this study.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

The authors would like to thank J.H. Yin and B. He as without their valuable assistance, this study would not have been successful. This work was supported in part by the National Natural Science Foundation of China under Grant 11771383, in part by the Natural Science Foundation of Guangxi Province under Grant 2016GXNSFAA380028, 2018GXNSFFA281007, and in part by Research Project of Guangxi University for Nationalities under Grant 2018KJQD02.

References

[1] S. P. Dirkse and M. C. Ferris, "Mcplib: a collection of nonlinear mixed complementarity problems," *Optimization Methods and Software*, vol. 5, no. 4, pp. 319–345, 1995.

[2] M. Raissi, P. Perdikaris, and G. E. Karniadakis, "Physics-informed neural networks: a deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations," *Journal of Computational Physics*, vol. 378, pp. 686–707, 2019.

[3] Z. Dai and H. Zhou, "Prediction of stock returns: sum-of-the-parts method and economic constraint method," *Sustainability*, vol. 12, no. 2, p. 541, 2020.

[4] Z. Dai, H. Zhou, F. Wen, and S. He, "Efficient predictability of stock return volatility: the role of stock market implied volatility," *The North American Journal of Economics and Finance*, vol. 52, Article ID 101174, 2020.

[5] K. Meintjes and A. P. Morgan, "A methodology for solving chemical equilibrium systems," *Applied Mathematics and Computation*, vol. 22, no. 4, pp. 333–361, 1987.

[6] X. Chai, X. Zheng, Z. Gan, D. Han, and Y. Chen, "An image encryption algorithm based on chaotic system and compressive sensing," *Signal Processing*, vol. 148, pp. 124–144, 2018.

[7] Z. Wan, J. Guo, J. Liu, and W. Liu, "A modified spectral conjugate gradient projection method for signal recovery," *Signal, Image and Video Processing*, vol. 12, no. 8, pp. 1455–1462, 2018.

[8] J. M. Ortega and W. C. Rheinboldt, *Iterative Solution of Nonlinear Equations in Several Variables*, Academic Press, New York, NY, USA, 1970.

[9] F. R. Oliveira, O. P. Ferreira, and G. N. Silva, "Newton's method with feasible inexact projections for solving constrained generalized equations," *Computational Optimization and Applications*, vol. 72, no. 1, pp. 159–177, 2019.

[10] Z. Wan, Y. Chen, S. Huang, and D. Feng, "A modified nonmonotone BFGS algorithm for solving smooth nonlinear equations," *Optimization Letters*, vol. 8, no. 6, pp. 1845–1860, 2014.

[11] C. Wang, Y. Wang, and C. Xu, "A projection method for a system of nonlinear monotone equations with convex constraints," *Mathematical Methods of Operations Research*, vol. 66, no. 1, pp. 33–46, 2007.

[12] L. Marini, B. Morini, and M. Porcelli, "Quasi-Newton methods for constrained nonlinear systems: complexity analysis and applications," *Computational Optimization and Applications*, vol. 71, no. 1, pp. 147–170, 2018.

[13] W. Zhou and D. Li, "Limited memory BFGS method for nonlinear monotone equations," *Journal of Computational Mathematics*, vol. 25, no. 1, pp. 89–96, 2007.

[14] D. S. Goncalves, M. L. N. Goncalves, and F. R. Oliveira, "Levenberg-Marquardt methods with inexact projections for constrained nonlinear systems," 2019, https://arxiv.org/abs/1908.06118.

[15] M. Kimiaei, "A new class of nonmonotone adaptive trust-region methods for nonlinear equations with box constraints," *Calcolo*, vol. 54, no. 3, pp. 769–812, 2017.

[16] X. Tong, L. Qi, and Y.-F. Yang, "The Lagrangian globalization method for nonsmooth constrained equations," *Computational Optimization and Applications*, vol. 33, no. 1, pp. 89–109, 2006.

[17] T. Li and Z. Wan, "New adaptive barzilai-borwein step size and its application in solving large-scale optimization problems," *The ANZIAM Journal*, vol. 61, no. 1, pp. 76–98, 2019.

[18] L. Zhang and W. Zhou, "Spectral gradient projection method for solving nonlinear monotone equations," *Journal of Computational and Applied Mathematics*, vol. 196, no. 2, pp. 478–484, 2006.

[19] Y. Ou and J. Li, "A new derivative-free SCG-type projection method for nonlinear monotone equations with convex constraints," *Journal of Applied Mathematics and Computing*, vol. 56, no. 1-2, pp. 195–216, 2018.

[20] G. Yu, S. Niu, S. Niu, and J. Ma, "Multivariate spectral gradient projection method for nonlinear monotone equations with convex constraints," *Journal of Industrial & Management Optimization*, vol. 9, no. 1, pp. 117–129, 2013.

[21] J. Liu, S. Li, and S. Li, "Multivariate spectral DY-type projection method for convex constrained nonlinear monotone equations," *Journal of Industrial & Management Optimization*, vol. 13, no. 1, pp. 283–295, 2017.

[22] J. Liu and Y. Feng, "A derivative-free iterative method for nonlinear monotone equations with convex constraints," *Numerical Algorithms*, vol. 82, no. 1, pp. 245–262, 2019.

[23] P. Gao, C. He, and Y. Liu, "An adaptive family of projection methods for constrained monotone nonlinear equations with applications," *Applied Mathematics and Computation*, vol. 359, no. 15, pp. 1–16, 2019.

[24] Y. Xiao, C. Wu, and S.-Y. Wu, "Norm descent conjugate gradient methods for solving symmetric nonlinear

equations," *Journal of Global Optimization*, vol. 62, no. 4, pp. 751–762, 2015.

[25] Z. Dai, X. Chen, and F. Wen, "A modified Perry's conjugate gradient method-based derivative-free method for solving large-scale nonlinear monotone equations," *Applied Mathematics and Computation*, vol. 270, no. 11, pp. 378–386, 2015.

[26] J. Guo and Z. Wan, "A modified spectral PRP conjugate gradient projection method for solving large-scale monotone equations and its application in compressed sensing," *Mathematical Problems in Engineering*, vol. 2019, Article ID 5261830, 17 pages, 2019.

[27] W. Y. Cheng, "A PRP type method for systems of monotone equations," *Mathematical and Computer Modelling*, vol. 50, no. 1-2, pp. 15–20, 2009.

[28] M. Sun and J. Liu, "New hybrid conjugate gradient projection method for the convex constrained equations," *Calcolo*, vol. 53, no. 3, pp. 399–411, 2016.

[29] P. Gao and C. He, "An efficient three-term conjugate gradient method for nonlinear monotone equations with convex constraints," *Calcolo*, vol. 55, no. 4, pp. 1–17, 2018.

[30] Y. Ding, Y. Xiao, and J. Li, "A class of conjugate gradient methods for convex constrained monotone equations," *Optimization*, vol. 66, no. 12, pp. 2309–2328, 2017.

[31] Q. Li and D.-H. Li, "A class of derivative-free methods for large-scale nonlinear monotone equations," *IMA Journal of Numerical Analysis*, vol. 31, no. 4, pp. 1625–1635, 2011.

[32] Y. Ou and Y. Liu, "Supermemory gradient methods for monotone nonlinear equations with convex constraints," *Computational and Applied Mathematics*, vol. 36, no. 1, pp. 259–279, 2017.

[33] Q. Xu, H. C. Lin, and Y. G. Ou, "A derivative-free memory method for systems of nonlinear equations with convex constraints," *Journal of Applied Mathematics*, vol. 29, no. 3, pp. 686–696, 2016.

[34] J. Jian, L. Han, and X. Jiang, "A hybrid conjugate gradient method with descent property for unconstrained optimization," *Applied Mathematical Modelling*, vol. 39, no. 3-4, pp. 1281–1290, 2015.

[35] G. Tsegay, H. Zhang, X. Zhang, and F. Zhang, "A sufficient descent conjugate gradient method for nonlinear unconstrained optimization problems," *Transactions in Operational Research*, vol. 22, no. 3, pp. 59–68, 2018.

[36] M. V. Solodov and B. F. Svaiter, "A globally convergent inexact Newton method for systems of monotone equations," in *Reformulation: Piecewise Smooth, Semi-smooth and Smoothing Methods*, pp. 355–369, Springer, Berlin, Germany, 1998.

[37] E. H. Zarantonello, *Projections on Convex Sets in Hilbert Space and Spectral Theory*, Academic Press, New York, NY, USA, 1971.

[38] E. D. Dolan and J. J. Moré, "Benchmarking optimization software with performance profiles," *Mathematical Programming*, vol. 91, no. 2, pp. 201–213, 2002.

[39] M. A. T. Figueiredo, R. D. Nowak, and S. J. Wright, "Gradient projection for sparse reconstruction: application to compressed sensing and other inverse problems," *IEEE Journal of Selected Topics in Signal Processing*, vol. 1, no. 4, pp. 586–597, 2007.

[40] Y. Xiao, Q. Wang, and Q. Hu, "Non-smooth equations based method for l_1 norm problems with applications to compressed sensing," *Nonlinear Analysis: Theory, Methods & Applications*, vol. 74, no. 11, pp. 3570–3577, 2011.

[41] J.-S. Pang, "Inexact Newton methods for the nonlinear complementarity problem," *Mathematical Programming*, vol. 36, no. 1, pp. 54–71, 1986.