

## A DAI-LIAO-LIKE PROJECTION METHOD FOR SOLVING CONVEX CONSTRAINED NONLINEAR MONOTONE EQUATIONS AND MINIMIZING THE $\ell_1$ -REGULARIZED PROBLEM

ABUBAKAR BAKOJI MUHAMMAD<sup>1,3</sup>, CHRISTIANE TAMMER<sup>1,\*</sup>, ALIYU MUHAMMED AWWAL<sup>2,3</sup>,  
ROSALIND ELSTER<sup>4</sup>

<sup>1</sup>*Faculty of Natural Sciences II, Institute of Mathematics,*

*Martin Luther University Halle-Wittenberg, Halle (Saale), Germany*

<sup>2</sup>*Department of Mathematics, Faculty of Science, King Mongkut's University of Technology Thonburi (KMUTT),  
Bangkok 10140, Thailand*

<sup>3</sup>*Department of Mathematics, Faculty of Science, Gombe State University, Gombe 760214, Nigeria*

<sup>4</sup>*06116 Halle (Saale), Germany*

**Abstract.** In this paper, a three-term derivative-free method for solving a nonlinear system of equations with convex constraints is proposed. In addition, by reformulating an  $\ell_1$ -regularized problem into a nonlinear system of equations, the proposed method is applicable to solving signal recovery and image deblurring problems. Our method is based on the projection technique of Solodov and Svaiter (1998) by incorporating a quasi-Newton-like direction with the Dai-Liao conjugate gradient parameter. The proposed method is matrix-free and the search direction satisfies a certain descent condition. Under the assumption that the underlying function is monotone and Lipschitzian, the global convergence of the proposed method is established. Preliminary numerical experiments on some large-scale nonlinear system of equations with convex constraints show that the proposed method is efficient. Furthermore, we apply the proposed method to the  $\ell_1$ -regularization problem in compressive sensing.

**Keywords.** Derivative-free method; Nonlinear monotone equations; Projection method; Quasi-Newton method; Signal and image recovery.

### 1. INTRODUCTION

Iterative methods, such as the Newton method, the quasi-Newton method, and conjugate gradient methods, have been widely investigated for general unconstrained optimization problem

$$\min\{f(x) : x \in \mathbb{R}^n\}, \quad (1.1)$$

where  $f : \mathbb{R}^n \rightarrow \mathbb{R}$  is a continuously differentiable function, and bounded from below. Unlike quasi-Newton methods, which require the storage of  $n \times n$  matrices in every iteration, conjugate gradient (CG) methods require the storage of  $n \times 1$  vectors at each iteration. Moreover, the CG

---

\*Corresponding author.

E-mail addresses: [abubakar.muhammad@mathematik.uni-halle.de](mailto:abubakar.muhammad@mathematik.uni-halle.de) (A.B. Muhammad), [christiane.tammer@mathematik.uni-halle.de](mailto:christiane.tammer@mathematik.uni-halle.de) (C. Tammer), [aliyumagsu@gmail.com](mailto:aliyumagsu@gmail.com) (A.M. Awwal), [r.elster@t-online.de](mailto:r.elster@t-online.de) (R. Elster).

Received March 19, 2021; Accepted August 24, 2021.

methods are of particular interest due to their global convergence property and simplicity in implementation. The CG methods generate an iterative sequence  $\{x_k\}$  in the following manner

$$x_{k+1} := x_k + \alpha_k d_k, \quad \forall k \geq 0. \quad (1.2)$$

The scalar  $\alpha_k > 0$  denotes the step-length that is obtained via a suitable line-search strategy, and the vector  $d_k$  is a search direction given by  $d_0 := -F(x_0)$  and  $d_k := -F(x_k) + \beta_k d_{k-1}$ ,  $\forall k \in \mathbb{N}$ , where  $F := \nabla f$ , and  $\beta_k$  (called the CG parameter) is a scalar that is updated in every iteration.

Under the differentiability assumption, a well-known necessary optimality condition for problem (1.1) is:

$$x^* \text{ is a minimizer of } f \implies \nabla f(x^*) = 0.$$

This fact is called the first-order necessary optimality condition for the minimizer  $x^*$  of problem (1.1). Therefore, we consider the following system of nonlinear equations

$$F(x) = 0. \quad (1.3)$$

In this context, the gradient of  $f$  in problem (1.1) is viewed as the function  $F : \mathbb{R}^n \rightarrow \mathbb{R}^n$  in problem (1.3), i.e.,  $F := \nabla f$ . Hence, algorithms for solving the nonlinear system of equations (1.3) can be used to solve problem (1.1).

In this paper, we consider the problem (1.3), where constraints given by a set  $\Psi \subseteq \mathbb{R}^n$  are involved, i.e., we look for a vector  $x \in \Psi \subseteq \mathbb{R}^n$  for which (1.3) holds. The feasible set  $\Psi$  is assumed to be nonempty, closed, and convex. Thus, we refer problem (1.3) as nonlinear system of equations with convex constraints. Many mathematical problems arising from various applications, such as fixed point problems, differential equations, variational inequality problems and so on, can be reformulated into problem (1.3) (see, e.g., [1, 2, 3, 4]). Moreover, problem (1.3) also appears as a sub-problem in generalized proximal algorithms with Bregman distances [5]. Optimization problems including least square errors and  $\ell_1$ -regularizations can equally be translated into problem (1.3) [6, 7]. These underline the importance of problem (1.3) and efficient algorithms with minimal computational costs for solving it.

Let us recall the quasi-Newton algorithm for solving problem (1.1) with its iterative sequence  $\{x_k\}$  being updated via (1.2), and the vector  $d_k$  being given by

$$d_k := -A_k F(x_k), \quad k \geq 0, \quad (1.4)$$

where  $A_k$  is an  $n \times n$ -matrix for each  $k$ . The search direction  $d_k$  in (1.4) needs to satisfy the following inequality

$$\langle F(x_k), d_k \rangle \leq -c \|F(x_k)\|^2, \quad c > 0. \quad (1.5)$$

The matrix  $A_k$  in (1.4) is an approximation of the inverse Hessian of  $f$  at  $x_k$  which can be computed by using different types of an updating formula. One of the famous formula for computing  $A_k$  is the following Broyden-Fletcher-Goldfarb-Shanno (BFGS) update

$$A_k := A_{k-1} + \left( 1 + \frac{y_{k-1}^T A_{k-1} y_{k-1}}{y_{k-1}^T s_{k-1}} \right) \frac{s_{k-1} s_{k-1}^T}{y_{k-1}^T s_{k-1}} - \left( \frac{s_{k-1} y_{k-1}^T A_{k-1} + A_{k-1} y_{k-1} s_{k-1}^T}{y_{k-1}^T s_{k-1}} \right), \quad (1.6)$$

where  $s_{k-1} := x_k - x_{k-1}$ ,  $y_{k-1} := F(x_k) - F(x_{k-1})$ , and it is assumed that  $y_{k-1}^T s_{k-1} \neq 0$ . By replacing the matrix  $A_{k-1}$  in (1.6) by an identity matrix, that is,  $A_{k-1} = I$ , (1.6) becomes

$$A_k = I + \left( 1 + \frac{y_{k-1}^T y_{k-1}}{y_{k-1}^T s_{k-1}} \right) \frac{s_{k-1} s_{k-1}^T}{y_{k-1}^T s_{k-1}} - \left( \frac{s_{k-1} y_{k-1}^T + y_{k-1} s_{k-1}^T}{y_{k-1}^T s_{k-1}} \right). \quad (1.7)$$

Multiplying  $F(x_k)$  from the right on (1.7) gives

$$A_k F(x_k) = F(x_k) + \left[ \left( 1 + \frac{\|y_{k-1}\|^2}{\langle y_{k-1}, s_{k-1} \rangle} \right) \frac{\langle s_{k-1}, F(x_k) \rangle}{\langle y_{k-1}, s_{k-1} \rangle} - \frac{\langle y_{k-1}, F(x_k) \rangle}{\langle y_{k-1}, s_{k-1} \rangle} \right] s_{k-1} - \frac{\langle s_{k-1}, F(x_k) \rangle}{\langle y_{k-1}, s_{k-1} \rangle} y_{k-1}. \quad (1.8)$$

Substituting (1.8) into (1.4) gives

$$d_k = -F(x_k) + \left[ \frac{\langle y_{k-1}, F(x_k) \rangle}{\langle y_{k-1}, s_{k-1} \rangle} - \left( 1 + \frac{\|y_{k-1}\|^2}{\langle y_{k-1}, s_{k-1} \rangle} \right) \frac{\langle s_{k-1}, F(x_k) \rangle}{\langle y_{k-1}, s_{k-1} \rangle} \right] s_{k-1} + \frac{\langle s_{k-1}, F(x_k) \rangle}{\langle y_{k-1}, s_{k-1} \rangle} y_{k-1}. \quad (1.9)$$

Since the coefficient of  $s_{k-1}$  and  $y_{k-1}$  are scalars, the search direction  $d_k$  in (1.9) can be written in the form of a three-term CG direction, i.e.,

$$d_k := -F(x_k) + \beta_k s_{k-1} + \frac{\langle s_{k-1}, F(x_k) \rangle}{\langle y_{k-1}, s_{k-1} \rangle} y_{k-1},$$

where the scalar  $\beta_k$  is a suitable CG parameter.

Recently, the hyperplane projection of Solodov and Svaiter [8] has influenced many researchers to extend CG methods for problem (1.1) to the nonlinear system of equations (1.3). For example, Dai and Liao [9] introduced two CG methods, which are based on the modified conjugacy condition proposed by Perry [10]. Their numerical results show that one of their methods is promising. However, the sequence of the search directions generated by their algorithms may not satisfy the descent condition (1.5). As a result, Babaie-Kafaki and Ghanbari [11] proposed a family of descent Dai-Liao CG methods for solving the general unconstrained optimization problem (1.1). Subsequently, Abubakar and Kumam [12] extended one of the methods in [11] to solve unconstrained monotone nonlinear equations by combining their method with the hyperplane projection technique of Solodov and Svaiter. Under some suitable assumptions, they proved the global convergence of the method. Preliminary numerical experiments conducted on some test problems show that their method works well. This method was later modified by Abubakar, Awwal and Kumam [13] to solve nonlinear monotone equations with convex constraints. Their algorithm was shown to have good numerical performance and successfully used to deal with signal recovery problems in compressive sensing. Motivated by the approach of Babaie-Kafaki and Ghanbari [11], Ahmed, Sabiu and Waziri proposed another modified Dai-Liao CG method for solving a nonlinear system of equations based on an extended secant equation in [14]. Under the assumption that the function  $F$  is continuously differentiable and Lipschitzian, they proved the global convergence of their method. Numerical results show that their method performs numerically better than the existing method developed by Fang and Ni [15]. Dai-Liao CG-like methods for solving unconstrained optimization problems (1.1) have been extensively studied over the years. However, to the best of our knowledge, not much study has been done in the literature on using them to solve the system of nonlinear monotone equations of the form (1.3). In this paper, motivated by the above discussion, particularly, the three-term CG search direction (1.9), we propose a Dai-Liao-like projection method for solving a system of nonlinear monotone equations (1.3) with convex constraints. The proposed method neither requires differentiability assumption nor matrix storage. The sequence

of the search directions generated by the developed algorithm satisfies the sufficient descent condition (1.5). Under the assumption that the function  $F$  is Lipschitzian and monotone, we establish the global convergence of the method. We present preliminary numerical experiments on some nonlinear monotone equations with convex constraints to demonstrate the efficiency of the proposed method. Subsequently, we successfully apply the proposed method to solve signal and image recovery problems. Throughout this paper, we denote by  $\mathbb{R}_+^n$ ,  $\|\cdot\|_2$  and  $\langle \cdot, \cdot \rangle$  the set  $\{(x_1, x_2, \dots, x_n)^T \in \mathbb{R}^n \mid x_i \geq 0, i = 1, 2, \dots, n\}$ , the Euclidean norm in  $\mathbb{R}^n$ , and the Euclidean inner product in  $\mathbb{R}^n$ , respectively.

The remaining part of this paper is organized as follows. In Section 2, we describe the three-term Dai-Liao-like projection method and its global convergence. In Section 3, we report numerical experiments to show the efficiency of the proposed method. We describe the application of the proposed algorithm in Section 4, and give our conclusions in Section 5, the last section.

## 2. THREE-TERM DAI-LIAO LIKE PROJECTION METHOD AND ITS GLOBAL CONVERGENCE

We begin this section by giving some preliminaries, and then we describe the details of the proposed algorithm.

**Definition 2.1.** A function  $F : \mathbb{R}^n \rightarrow \mathbb{R}^n$  is said to be monotone if, for all  $x, y \in \mathbb{R}^n$ ,

$$\langle F(x) - F(y), x - y \rangle \geq 0. \quad (2.1)$$

**Definition 2.2.** A function  $F : \mathbb{R}^n \rightarrow \mathbb{R}^n$  is said to be Lipschitzian if there exists some positive constant, say  $L$ , such that, for all  $x, y \in \mathbb{R}^n$ ,

$$\|F(x) - F(y)\| \leq L\|x - y\|. \quad (2.2)$$

**Assumption 2.1.** The solution set of problem (1.3) is nonempty, and the function  $F : \mathbb{R}^n \rightarrow \mathbb{R}^n$  is monotone and Lipschitzian. That is,  $F$  satisfies (2.1) and (2.2).

We recall that the Dai and Liao two-term CG method in [9] generates a sequence  $\{x_k\}$  via (1.2), where the CG parameter  $\beta_k := \beta_k^{DL}$  is obtained by the modified conjugacy condition and is given by

$$\beta_k^{DL} := \frac{\langle F(x_k), y_{k-1} - t s_{k-1} \rangle}{\langle y_{k-1}, d_{k-1} \rangle}, \quad t \geq 0, \quad (2.3)$$

$y_{k-1} := F(x_k) - F(x_{k-1})$ , and  $s_{k-1} := x_k - x_{k-1}$ . However, as mentioned above, the Dai-Liao method with the CG parameter given by (2.3) may not generate descent directions. In addition, for some functions, the inner product  $\langle y_{k-1}, d_{k-1} \rangle$  may become zero which will make the parameter  $\beta_k^{DL}$  (2.3) undefined. See Remark 2.1 for more details of the scalar product  $\langle y_{k-1}, d_{k-1} \rangle$  at the denominator of the CG parameter (2.3).

Motivated by the three-term search direction (1.9), (2.3), and the approach of Zhang, Zhou and Li [16], we propose the following search direction

$$d_k := \begin{cases} -F(x_k), & \text{if } k = 0, \\ -\lambda_k F(x_k) + \beta_k^{TDLP} s_{k-1} + \frac{\langle s_{k-1}, F(x_k) \rangle}{\langle \eta_{k-1}, s_{k-1} \rangle} \vartheta_{k-1}, & \text{if } k > 0, \end{cases} \quad (2.4)$$

where  $\lambda_k$  is a scalar parameter to be determined in such a way that the search direction  $d_k$  given by (2.4) satisfies inequality (1.5). The CG parameter  $\beta_k := \beta_k^{TDLP}$  (TDLP stands for **T**hree-term **D**ai-**L**iao **P**rojection algorithm) is given by

$$\beta_k^{TDLP} := \frac{\langle F(x_k), \vartheta_{k-1} - ts_{k-1} \rangle}{\langle \eta_{k-1}, s_{k-1} \rangle}, \quad (2.5)$$

where

$$\eta_{k-1} := \vartheta_{k-1} + \left( 1 + \max \left\{ 0, -\frac{\langle s_{k-1}, \vartheta_{k-1} \rangle}{\|s_{k-1}\|^2} \right\} \right) s_{k-1}, \quad (2.6)$$

$\vartheta_{k-1} := F(x_k) - F(x_{k-1}) + rs_{k-1}$ ,  $r > 0$ ,  $s_{k-1} := x_k - x_{k-1}$ , and  $x_k \neq x_{k-1}$ . Obviously, it is clear that the CG parameter  $\beta_k^{TDLP}$  given by (2.5) is well-defined (see Remark 2.1 for more details).

**Remark 2.1.** To show that  $\beta_k^{TDLP}$  is well-defined, by the definition of  $\eta_{k-1}$  given by (2.6), we consider two cases as follows:

If  $\max \left\{ 0, -\frac{\langle s_{k-1}, \vartheta_{k-1} \rangle}{\|s_{k-1}\|^2} \right\} \neq 0$  in the definition of  $\eta_{k-1}$  in (2.6), then

$$\begin{aligned} \langle s_{k-1}, \eta_{k-1} \rangle &= \left\langle s_{k-1}, \vartheta_{k-1} + s_{k-1} - \frac{\langle s_{k-1}, \vartheta_{k-1} \rangle}{\|s_{k-1}\|^2} s_{k-1} \right\rangle \\ &= \langle s_{k-1}, \vartheta_{k-1} \rangle + \|s_{k-1}\|^2 - \frac{\langle s_{k-1}, \vartheta_{k-1} \rangle}{\|s_{k-1}\|^2} \|s_{k-1}\|^2 \\ &= \|s_{k-1}\|^2 > 0, \text{ if } x_k \neq x_{k-1}. \end{aligned}$$

Otherwise,

$$\begin{aligned} \langle s_{k-1}, \eta_{k-1} \rangle &= \langle s_{k-1}, \vartheta_{k-1} \rangle + \|s_{k-1}\|^2 \\ &= \langle s_{k-1}, F(x_k) - F(x_{k-1}) + rs_{k-1} \rangle + \|s_{k-1}\|^2 \\ &= \langle s_{k-1}, F(x_k) - F(x_{k-1}) \rangle + r\|s_{k-1}\|^2 + \|s_{k-1}\|^2 \\ &\geq (r+1)\|s_{k-1}\|^2 > 0. \end{aligned}$$

The last inequality follows from the definition of  $s_{k-1}$ , and the monotonicity of  $F$ . This means that  $\langle s_{k-1}, \eta_{k-1} \rangle > 0$ . Hence,  $\beta_k^{TDLP}$  is well-defined.

**Remark 2.2.** In order to determine  $\lambda_k$  such that (1.5) is fulfilled for  $d_k$  given by (2.4), we proceed as follows: Let  $k > 0$ . By taking the inner product of  $F(x_k)$ , and  $d_k$  defined in (2.4), we obtain

$$\begin{aligned} \langle F(x_k), d_k \rangle &= -\lambda_k \|F(x_k)\|^2 + 2 \frac{\langle F(x_k), \vartheta_{k-1} \rangle \langle F(x_k), s_{k-1} \rangle}{\langle \eta_{k-1}, s_{k-1} \rangle} - t \frac{\langle F(x_k), s_{k-1} \rangle^2}{\langle \eta_{k-1}, s_{k-1} \rangle} \\ &\leq -\lambda_k \|F(x_k)\|^2 + 2 \frac{\langle F(x_k), \vartheta_{k-1} \rangle \langle F(x_k), s_{k-1} \rangle}{\langle \eta_{k-1}, s_{k-1} \rangle} \\ &\leq -\lambda_k \|F(x_k)\|^2 + 2 \frac{\|F(x_k)\|^2 \|\vartheta_{k-1}\| \|s_{k-1}\|}{\langle \eta_{k-1}, s_{k-1} \rangle} \\ &= -\left( \lambda_k - 2 \frac{\|\vartheta_{k-1}\| \|s_{k-1}\|}{\langle \eta_{k-1}, s_{k-1} \rangle} \right) \|F(x_k)\|^2. \end{aligned} \quad (2.7)$$

The first inequality in (2.7) was obtained by dropping the third term in the preceding line since the denominator is positive from Remark (2.1). The second inequality was obtained by applying

the Cauchy-Schwarz inequality. Therefore, for the search direction (2.4) to satisfy inequality (1.5), we only need

$$\lambda_k \geq c + 2 \frac{\|\vartheta_{k-1}\| \|s_{k-1}\|}{\langle \eta_{k-1}, s_{k-1} \rangle}, \quad c > 0, \text{ for all } k \in \mathbb{N}.$$

Without loss of generality, we choose the following value of  $\lambda_k$  in the proposed method as

$$\lambda_k := c + 2 \frac{\|\vartheta_{k-1}\| \|s_{k-1}\|}{\langle \eta_{k-1}, s_{k-1} \rangle}, \text{ for all } k \in \mathbb{N}. \quad (2.8)$$

On the other hand, let  $x \in \mathbb{R}^n$  be any point and define its projection onto the feasible set  $\Psi$  as  $P_\Psi(x) := \arg \min\{\|x - y\| : y \in \Psi\}$ . This projection operator  $P_\Psi(x)$  is nonexpansive as shown in the following Lemma.

**Lemma 2.1.** [17] *Let  $\Psi$  be a nonempty closed convex subset of  $\mathbb{R}^n$ . For any  $x, y \in \mathbb{R}^n$ , the following inequality holds*

$$\|P_\Psi(x) - P_\Psi(y)\| \leq \|x - y\|. \quad (2.9)$$

If  $y \in \Psi$ , then  $P_\Psi(y) = y$ . Hence,

$$\|P_\Psi(x) - y\| \leq \|x - y\|. \quad (2.10)$$

In what follows, we present the steps of the proposed **Three-term Dai-Liao like Projection (TDLP)** algorithm for solving a system of nonlinear monotone equations according to (1.3).

---

**Algorithm 1: Three-term Dai-Liao Projection (TDLP)**

---

**Input:** Give an initial point  $x_0 \in \Psi$ , parameters  $\sigma > 0, r > 0, t > 0, c > 0, \kappa > 0, \rho \in (0, 1)$ , and stopping tolerance  $Tol \geq 0$ . Set  $k := 0$ .

**Step 1:** Compute  $F(x_k)$  and set  $d_k := -F(x_k)$ .

**Step 2:** If  $\|F(x_k)\| \leq Tol$ , then  $x_k$  is a solution and the iteration process stops.

**Step 3:** Set

$$w_k := x_k + \alpha_k d_k, \quad (2.11)$$

where  $\alpha_k := \kappa \rho^i$  such that  $i$  is the smallest nonnegative integer satisfying

$$-\langle F(x_k + \kappa \rho^i d_k), d_k \rangle \geq \sigma \kappa \rho^i \|d_k\|^2 \|F(x_k + \kappa \rho^i d_k)\|^{1/q}, \quad q \geq 1. \quad (2.12)$$

**Step 4:** If  $\|F(w_k)\| = 0$ , then stop. Else, compute the next iterate

$$x_{k+1} := P_\Psi \left[ x_k - \frac{\langle F(w_k), x_k - w_k \rangle}{\|F(w_k)\|^2} F(w_k) \right], \quad F(w_k) \neq 0. \quad (2.13)$$

**Step 5:** Set  $k := k + 1$  and update  $d_k$  by using (2.4), (2.5) and (2.8), and repeat the process from **Step 2**.

---

Next, we give a pictorial description of the metric projection of Algorithm 1 where  $\zeta_k := \frac{\langle F(w_k), x_k - w_k \rangle}{\|F(w_k)\|^2}$ .

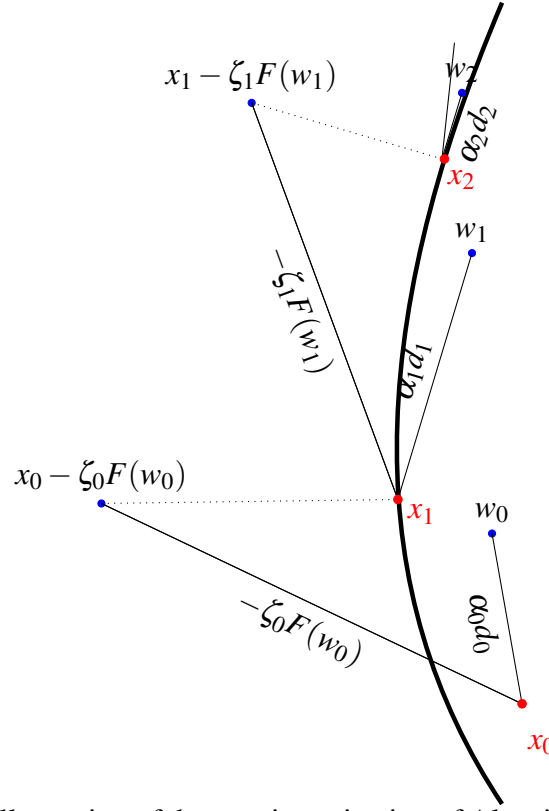


FIGURE 1. The illustration of the metric projection of Algorithm 1 adopted from the Figure 1 of Abubakar et al. [18]

**Remark 2.3.** The search direction  $d_k$  generated by Algorithm 1 satisfies the sufficient descent condition

$$\langle F(x_k), d_k \rangle \leq -c \|F(x_k)\|^2 < 0, \quad \text{for all } k = 0, 1, 2, \dots, \quad c > 0. \quad (2.14)$$

**Remark 2.4.** The line search strategy (2.12), proposed in [19], is well defined. That is, for all  $k \geq 0$ , there always exists a step-size  $\alpha_k$  satisfying (2.12) in a finite number of iterations.

Suppose on the contrary that there exists some  $k_0$  such that, for any  $i = 0, 1, 2, \dots$ , (2.12) does not hold, that is,  $-\langle F(x_{k_0} + \kappa \rho^i d_{k_0}), d_{k_0} \rangle < \sigma \kappa \rho^i \|d_{k_0}\|^2 \|F(x_{k_0} + \kappa \rho^i d_{k_0})\|^{1/q}$ ,  $q \geq 1$ . By the continuity of  $F$  and  $0 < \rho^i < 1$ , ( $i = 0, 1, 2, \dots$ ), letting  $i \rightarrow \infty$  yields  $\langle F(x_{k_0}), d_{k_0} \rangle \geq 0$ , which contradicts (2.14). Hence, line search (2.12) is well-defined.

**Lemma 2.2.** Let  $\hat{x}$  be a solution of problem (1.3) and let  $F$  be monotone and Lipschitzian function. Let the sequences  $\{d_k\}$ ,  $\{w_k\}$  and  $\{x_k\}$  be generated by (2.4), (2.11), and (2.13) as well as the sequence of scalars  $\{\alpha_k\}$  be in Algorithm 1. Then the following assertions hold:

- (i)  $\{x_k\}$  and  $\{\|F(x_k)\|\}$  are bounded and  $\lim_{k \rightarrow \infty} \|x_k - \hat{x}\|$  exists;
- (ii)  $\{w_k\}$  and  $\{\|F(w_k)\|\}$  are bounded;
- (iii)  $\lim_{k \rightarrow \infty} \alpha_k \|d_k\| = 0$ ;
- (iv)  $\lim_{k \rightarrow \infty} \|x_{k+1} - x_k\| = 0$ .

*Proof.* (i) From property (2.10), we have

$$\left\| P_{\Psi} \left( x_k - \frac{\langle F(w_k), x_k - w_k \rangle}{\|F(w_k)\|^2} F(w_k) \right) - \hat{x} \right\| \leq \left\| x_k - \frac{\langle F(w_k), x_k - w_k \rangle}{\|F(w_k)\|^2} F(w_k) - \hat{x} \right\|. \quad (2.15)$$

Since  $\hat{x}$  is a solution of problem (1.3), it holds that  $F(\hat{x}) = 0$ . Therefore,  $\langle F(\hat{x}), w_k - \hat{x} \rangle = 0$ . By the monotonicity of  $F$ , we have  $\langle F(\hat{x}), w_k - \hat{x} \rangle \leq \langle F(w_k), w_k - \hat{x} \rangle$ . This means

$$\begin{aligned} \langle F(w_k), x_k - w_k \rangle &= \langle F(w_k), x_k - w_k \rangle + \langle F(\hat{x}), w_k - \hat{x} \rangle \\ &\leq \langle F(w_k), x_k - w_k \rangle + \langle F(w_k), w_k - \hat{x} \rangle \\ &= \langle F(w_k), x_k - w_k + w_k - \hat{x} \rangle \\ &= \langle F(w_k), x_k - \hat{x} \rangle. \end{aligned} \quad (2.16)$$

By (2.15), (2.16) and the definition of  $x_{k+1}$  in (2.13), we obtain

$$\begin{aligned} \|x_{k+1} - \hat{x}\|^2 &\leq \left\| x_k - \hat{x} - \frac{\langle F(w_k), x_k - w_k \rangle}{\|F(w_k)\|^2} F(w_k) \right\|^2 \\ &= \|x_k - \hat{x}\|^2 - 2 \frac{\langle F(w_k), x_k - w_k \rangle}{\|F(w_k)\|^2} \langle F(w_k), x_k - \hat{x} \rangle + \frac{\langle F(w_k), x_k - w_k \rangle^2}{\|F(w_k)\|^2} \\ &\leq \|x_k - \hat{x}\|^2 - 2 \frac{\langle F(w_k), x_k - w_k \rangle}{\|F(w_k)\|^2} \langle F(w_k), x_k - w_k \rangle + \frac{\langle F(w_k), x_k - w_k \rangle^2}{\|F(w_k)\|^2} \\ &= \|x_k - \hat{x}\|^2 - \frac{\langle F(w_k), x_k - w_k \rangle^2}{\|F(w_k)\|^2} \\ &\leq \|x_k - \hat{x}\|^2. \end{aligned} \quad (2.17)$$

This implies that, for all  $k \geq 0$ ,  $\|x_{k+1} - \hat{x}\| \leq \|x_k - \hat{x}\| \leq \|x_{k-1} - \hat{x}\| \leq \dots \leq \|x_0 - \hat{x}\|$ , where  $x_0$  is one of the given starting points. Therefore,  $\lim_{k \rightarrow \infty} \|x_k - \hat{x}\|$  exists, and  $\{x_k\}$  is bounded.

Let  $u_1 := L\|x_0 - \hat{x}\|$ . Since  $F$  is Lipschitzian, for all  $k \geq 0$ , we have

$$\begin{aligned} \|F(x_k)\| &= \|F(x_k) - F(\hat{x})\| \\ &\leq L\|x_k - \hat{x}\| \\ &\vdots \\ &\leq L\|x_0 - \hat{x}\| \\ &= u_1. \end{aligned} \quad (2.18)$$

(ii) By the definition of  $w_k$  in (2.11), (2.16), and the boundedness of  $\{x_k\}$ , we obtain that, for all  $k \geq 0$ ,  $\{w_k\}$  is bounded.

By the Lipschitzian of  $F$  and the boundedness of  $\{w_k\}$ , there exists some constant, say  $u_2$ , such that, for all  $k \geq 0$ ,

$$\begin{aligned} \|F(w_k)\| &= \|F(w_k) - F(\hat{x})\| \\ &\leq L\|w_k - \hat{x}\| \\ &\vdots \\ &\leq L\|w_0 - \hat{x}\| \\ &= u_2. \end{aligned} \quad (2.19)$$



where  $u_2 := L\|w_0 - \hat{x}\|$ .

(iii) From (2.17), we can deduce that

$$\langle F(w_k), \alpha_k d_k \rangle^2 \leq \|F(w_k)\|^2 (\|x_k - \hat{x}\|^2 - \|x_{k+1} - \hat{x}\|^2). \quad (2.20)$$

By the definition of  $\alpha_k$  in **Step 3** of Algorithm 1 and (2.12), we have

$$\sigma^2 \alpha_k^4 \|d_k\|^4 \|F(w_k)\|^{2/q} \leq \langle F(w_k), \alpha_k d_k \rangle^2. \quad (2.21)$$

Combining (2.20) and (2.21), we obtain

$$\sigma^2 \alpha_k^4 \|d_k\|^4 \|F(w_k)\|^{2/q} \leq \|F(w_k)\|^2 (\|x_k - \hat{x}\|^2 - \|x_{k+1} - \hat{x}\|^2). \quad (2.22)$$

Since  $\lim_{k \rightarrow \infty} \|x_k - \hat{x}\|$  exists, we conclude from (2.19) and (2.22) that

$$\begin{aligned} \sigma^2 \lim_{k \rightarrow \infty} \alpha_k^4 \|d_k\|^4 &\leq \lim_{k \rightarrow \infty} \|F(w_k)\|^{2-2/q} (\|x_k - \hat{x}\|^2 - \|x_{k+1} - \hat{x}\|^2) \\ &\leq u_2^{2-2/q} \lim_{k \rightarrow \infty} (\|x_k - \hat{x}\|^2 - \|x_{k+1} - \hat{x}\|^2) \\ &= 0. \end{aligned}$$

This implies,

$$\lim_{k \rightarrow \infty} \alpha_k \|d_k\| = 0. \quad (2.23)$$

(iv) Using (2.9), (2.11), (2.13), (2.23), and the Cauchy–Schwarz inequality, we have

$$\begin{aligned} \lim_{k \rightarrow \infty} \|x_{k+1} - x_k\| &= \lim_{k \rightarrow \infty} \left\| P_{\Psi} \left[ x_k - \frac{\langle F(w_k), x_k - w_k \rangle}{\|F(w_k)\|^2} F(w_k) \right] - P_{\Psi}(x_k) \right\| \\ &\leq \lim_{k \rightarrow \infty} \left\| x_k - \frac{\langle F(w_k), x_k - w_k \rangle}{\|F(w_k)\|^2} F(w_k) - x_k \right\| \\ &\leq \lim_{k \rightarrow \infty} \frac{\|F(w_k)\| \|x_k - w_k\|}{\|F(w_k)\|^2} \|F(w_k)\| \\ &= \lim_{k \rightarrow \infty} \|x_k - w_k\| \\ &= \lim_{k \rightarrow \infty} \alpha_k \|d_k\| \\ &= 0. \end{aligned}$$

This completes the proof.  $\square$

**Theorem 2.1.** *Let  $\{x_k\}$  be a sequence generated by Algorithm 1. Suppose that  $F$  is Lipschitzian and the solution set of problem (1.3) is nonempty. Then,*

$$\liminf_{k \rightarrow \infty} \|F(x_k)\| = 0. \quad (2.24)$$

Furthermore,  $\{x_k\}$  converges to the solution of problem (1.3).

*Proof.* First, we show that the search direction  $d_k$  generated by Algorithm 1 is bounded for all  $k \geq 0$ .

For  $k = 0$ , we have from (2.4) and (2.18) that  $\|d_0\| = \|F(x_0)\| \leq u_1$ . For  $k > 0$ , by the definition of  $\vartheta_{k-1}$  and the Lipschitz continuity of  $F$ , we have

$$\begin{aligned}\|\vartheta_{k-1}\| &= \|F(x_k) - F(x_{k-1}) + r(x_k - x_{k-1})\| \\ &\leq \|F(x_k) - F(x_{k-1})\| + r\|x_k - x_{k-1}\| \\ &\leq L\|x_k - x_{k-1}\| + r\|x_k - x_{k-1}\| \\ &= (L + r)\|s_{k-1}\|, \quad r > 0.\end{aligned}$$

Substituting (2.5) and (2.8) into (2.4), we obtain

$$\begin{aligned}\|d_k\| &= \left\| -\lambda_k F(x_k) + \beta_k^{TDLP} s_{k-1} + \frac{\langle s_{k-1}, F(x_k) \rangle}{\langle \eta_{k-1}, s_{k-1} \rangle} \vartheta_{k-1} \right\| \\ &\leq c\|F(x_k)\| + 2 \frac{\|\vartheta_{k-1}\| \|s_{k-1}\|}{\langle \eta_{k-1}, s_{k-1} \rangle} \|F(x_k)\| + 2 \frac{\|F(x_k)\| \|\vartheta_{k-1}\| \|s_{k-1}\|}{\langle \eta_{k-1}, s_{k-1} \rangle} + \frac{t\|F(x_k)\| \|s_{k-1}\|^2}{\langle \eta_{k-1}, s_{k-1} \rangle} \\ &\leq c\|F(x_k)\| + 4 \frac{\|\vartheta_{k-1}\| \|s_{k-1}\|}{\|s_{k-1}\|^2} \|F(x_k)\| + \frac{t\|F(x_k)\| \|s_{k-1}\|^2}{\|s_{k-1}\|^2} \\ &\leq c\|F(x_k)\| + 4 \frac{(L + r)\|s_{k-1}\|^2}{\|s_{k-1}\|^2} \|F(x_k)\| + \frac{t\|F(x_k)\| \|s_{k-1}\|^2}{\|s_{k-1}\|^2} \\ &= c\|F(x_k)\| + 4(L + r)\|F(x_k)\| + t\|F(x_k)\| \\ &= [c + 4(L + r) + t]\|F(x_k)\| \\ &\leq [c + 4(L + r) + t]u_1.\end{aligned}$$

By letting  $\mathbb{M} := [c + 4(L + r) + t]u_1$ , we have

$$\|d_k\| \leq \mathbb{M}, \text{ for all } k \in \mathbb{N}_0. \quad (2.25)$$

Now, we suppose on the contrary that (2.24) does not hold. Then, there exists a sufficiently large  $K$  such that, for all  $k \geq K$ ,  $\|F(x_k)\| \geq \varepsilon$ , where  $\varepsilon > 0$  is a given positive constant. Suppose  $\alpha'_k := \rho^{-1} \alpha_k$ , and define  $w'_k := x_k + \alpha'_k d_k$ . If  $\alpha_k \neq \kappa$ . Then,  $\alpha'_k$  does not satisfy (2.12), i.e.,

$$F(w'_k)^T d_k + \sigma \alpha'_k \|F(w'_k)\|^{1/q} \|d_k\|^2 > 0.$$

Since  $\hat{x}$  is in the solution set of problem (1.3), by the Lipschitzian of  $F$  and the boundedness of  $\|d_k\|$  in (2.25), we obtain, for all  $k \geq 0$ ,

$$\begin{aligned}\|F(w'_k)\| &= \|F(w'_k) - F(\hat{x})\| \\ &\leq L\|x_k + \alpha'_k d_k - \hat{x}\| \\ &\leq L\|x_k - \hat{x}\| + L\alpha'_k \|d_k\| \\ &\leq L\|x_0 - \hat{x}\| + L\alpha'_k \|d_k\| \\ &\leq u_1 + L\rho^{-1} \alpha_k \mathbb{M} \\ &\leq u_1 + L\rho^{-1} \mathbb{M},\end{aligned}$$

where  $u_1 := L\|x_0 - \hat{x}\|$ . The last inequality follows from the fact that  $\alpha_k \leq 1$ , for all  $k \geq 0$ . Now, letting  $u_3 := u_1 + L\rho^{-1} \mathbb{M}$ , we have

$$\|F(w'_k)\| \leq u_3. \quad (2.26)$$

From (1.5), (2.2), and the Cauchy–Schwarz inequality, we have, for all  $k \geq 0$ ,

$$\begin{aligned}
 c\|F(x_k)\|^2 &\leq -\langle F(x_k), d_k \rangle \\
 &< -\langle F(x_k), d_k \rangle + \langle F(w'_k), d_k \rangle + \sigma\alpha'_k\|F(w'_k)\|^{1/q}\|d_k\|^2 \\
 &= \langle F(w'_k) - F(x_k), d_k \rangle + \sigma\alpha'_k\|F(w'_k)\|^{1/q}\|d_k\|^2 \\
 &\leq \|F(w'_k) - F(x_k)\|\|d_k\| + \sigma\alpha'_k\|F(w'_k)\|^{1/q}\|d_k\|^2 \\
 &\leq L\alpha'_k\|d_k\|^2 + \sigma\alpha'_k\|F(w'_k)\|^{1/q}\|d_k\|^2 \\
 &= \rho^{-1}\alpha_k(L + \sigma\|F(w'_k)\|^{1/q})\|d_k\|^2.
 \end{aligned}$$

This further gives

$$\alpha_k\|d_k\| \geq \frac{\rho c\|F(x_k)\|^2}{(L + \sigma\|F(w'_k)\|^{1/q})\|d_k\|} \geq \frac{\rho c\epsilon^2}{(L + \sigma u_3^{1/q})\mathbb{M}}, \quad (2.27)$$

where the last inequality follows from (2.25) and (2.26). Taking the limit of (2.27), we obtain

$$\lim_{k \rightarrow \infty} \alpha_k\|d_k\| \geq \frac{\rho c\epsilon^2}{(L + \sigma u_3^{1/q})\mathbb{M}},$$

which clearly contradicts (2.23). Hence, assertion (2.24) holds.

Furthermore, by the continuity of  $F$  and the boundedness of  $\{x_k\}$ , there exists some accumulation point  $\hat{x}$  of  $\{x_k\}$  such that  $F(\hat{x}) = 0$ . Since  $\{\|x_k - \hat{x}\|\}$  converges (according to Lemma 2.2) and  $\hat{x}$  is an accumulation point of  $\{x_k\}$ , it follows that  $\{x_k\}$  converges to  $\hat{x}$ . This completes the proof.  $\square$

### 3. NUMERICAL EXPERIMENTS

In this section, we present some numerical experiments to assess the performance of the proposed method as well as its computational advantage in comparison with two existing methods. We compare the performance of our method described in Algorithm 1 (TDLP) with the methods of Zheng, Yang and Liang [20] (denoted as CGPM) and Abubakar et al. [21] (denoted as DCG). The executed algorithms were written on Windows 10 ASUSTek personal computer with Intel(R) Core(TM) i7-7500U processor with 8.00GB of RAM and CPU of 2.70GHz using MATLAB R2017a software. In order to get the best possible results, the following parameters:  $\kappa = 1$ ,  $\rho = 0.8$ ,  $\sigma = 0.01$ ,  $r = 0.01$ ,  $q = 5$ ,  $c = 10$  and  $t = 0.1$  were chosen for Algorithm 1 (TDLP). For the other two methods, their parameters were chosen as reported in their respective papers. Furthermore, all runs were terminated whenever  $\|F(x_k)\| < 10^{-6}$ . If this condition is not satisfied after 1000 iterations, failure is declared.

In order to test the performance and efficiency of the methods, the following test problems taken from existing literature are used for the experiment where the function  $F$  is taken as  $F(x) = (f_1(x), f_2(x), \dots, f_n(x))^T$ . The associated initial points for these test problems are given as  $x_1 = (0.1, 0.1, \dots, 0.1)^T$ ,  $x_2 = (\frac{1}{2}, \frac{1}{2^2}, \dots, \frac{1}{2^n})^T$ ,  $x_3 = (2, 2, \dots, 2)^T$ ,  $x_4 = (1, \frac{1}{2}, \dots, \frac{1}{n})^T$ ,  $x_5 = (1 - \frac{1}{n}, 1 - \frac{2}{n}, \dots, 0)^T$ , and  $x_6 = \text{rand}(0, 1)$ . Problems 1-9 are solved with dimensions  $n = 1000, 5000, 10000, 50000$ , and  $100000$  while Problem 10 is solved with dimension  $n = 4$ .

Problem 1 [22]

$$f_1(x) = e^{x_1} - 1,$$

$$f_i(x) = e^{x_i} + x_{i-1} - 1, \text{ for } i = 1, 2, \dots, n-1 \text{ and } \Psi = \mathbb{R}_+^n.$$

Problem 2 [22]

$$f_i(x) = 2x_i - \sin|x_i|, \quad i = 1, 2, \dots, n \text{ and } \Psi = \mathbb{R}_+^n.$$

Problem 3 [23]

$$f_i(x) = e^{x_i} - 1, \quad i = 1, 2, \dots, n \text{ and } \Psi = \mathbb{R}_+^n.$$

Problem 4 [24]

$$f_1(x) = x_1 - e^{\cos(h(x_1+x_2))},$$

$$f_i(x) = x_i - e^{\cos(h(x_{i-1}+x_i+x_{i+1}))}, \quad i = 2, \dots, n-1,$$

$$f_n(x) = x_n - e^{\cos(h(x_{n-1}+x_n))}, \quad h = \frac{1}{n+1} \text{ and } \Psi = \mathbb{R}_+^n.$$

Problem 5 [23]

$$f_i(x) = x_i - \sin(|x_i - 1|), \quad i = 1, 2, \dots, n-1$$

$$\text{and } \Psi = \{x \in \mathbb{R}^n : \sum_{i=1}^n x_i \leq n, \quad x_i \geq -1, \quad i = 1, 2, \dots, n\}.$$

Problem 6 [25]

$$f_i(x) = (e^{x_i})^2 + \frac{3}{2} \sin(2x_i) - 1, \quad i = 1, 2, \dots, n \text{ and } \Psi = \mathbb{R}_+^n.$$

Problem 7 [12]

$$f_1(x) = \frac{5}{2}x_1 + x_2 - 1,$$

$$f_i(x) = x_{i-1} + \frac{5}{2}x_i + x_{i+1} - 1, \quad i = 2, \dots, n-1,$$

$$f_n(x) = x_{n-1} + \frac{5}{2}x_n - 1 \text{ and } \Psi = \mathbb{R}_+^n.$$

Problem 8 [26]

$$f_i(x) = \frac{i}{n}e^{x_i} - 1, \text{ for } i = 1, 2, \dots, n \text{ and } \Psi = \mathbb{R}_+^n.$$

Problem 9 [27]

$$x_i = \sum_{i=1}^n x_i^2, \quad c = 10^{-5},$$

$$f_i(x) = 2c(x_i - 1) + 4(x_i - 0.25)x_i, \quad i = 1, 2, \dots, n \text{ and } \Psi = \mathbb{R}_+^n.$$

Problem 10 [22] Let  $F : \mathbb{R}^4 \rightarrow \mathbb{R}^4$  be defined by

$$F(x) = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & -1 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{pmatrix} + \begin{pmatrix} x_1^3 \\ x_2^3 \\ 2x_3^3 \\ 2x_4^3 \end{pmatrix} + \begin{pmatrix} -10 \\ 1 \\ -3 \\ 0 \end{pmatrix},$$

$$\text{and } \Psi = \{x \in \mathbb{R}^n : \sum_{i=1}^4 x_i = 3, x_i \geq 0, i = 1, 2, 3, 4\}.$$

The metrics used for the comparison are ITER (number of iterations), FVAL (number of function evaluations), and TIME (CPU time in seconds), where their information together with NORM (norm of the objective function at the solution) are reported. The Tables of the numerical results are available in the following link <https://github.com/AMBakoji/CGAIS-CGWOIS>. The NORM values indicate whether an algorithm successfully leads to an approximate solution of a particular problem or not.

Now, from the NORM values reported in the Tables, we see that the three methods: TDLP, CGPM, and DCG solved all the test problems successfully. Considering the ITER values reported in the Tables, though all the three methods recorded relatively low ITER, it can be seen that the TDLP method recorded the least values of ITER in most cases. Similarly, if we consider the FVAL values recorded by the TDLP, CGPM, and DCG, the TDLP method needs the smallest FVAL to obtain the solutions of the test problems. Last, the TIME presented in the Tables show that the three methods obtain the solutions of most of the test problems in less than one second, however, for some problems with DIM=100000, the TDLP method requires more computing time than the CGPM and DCG methods. This might be as a result of a number of parameters needed to be computed by the TDLP method. Taking everything into consideration, we see that the TDLP method works well and outperform the CGPM and DCG methods.

In order to have a graphical view on the numerical performance of TDLP relative to CGPM and DCG, we summarized the information on the Tables (<https://github.com/AMBakoji/CGAIS-CGWOIS>) and employ the performance profile proposed by Dolan and Moré in [28]. Figure 2 illustrates the performance profile of the three algorithms based on ITER, while Figure 3 demonstrates the performance profile based on FVAL and lastly, Figure 4 shows the performance profile based on TIME. Based on the performance profile of Dolan and Moré, it can be observed from Figure 2, 3 and 4 that the TDLP method is comparable or superior to the CGPM method and also superior to the DCG method in terms of the number of iterations and the number of function evaluations.

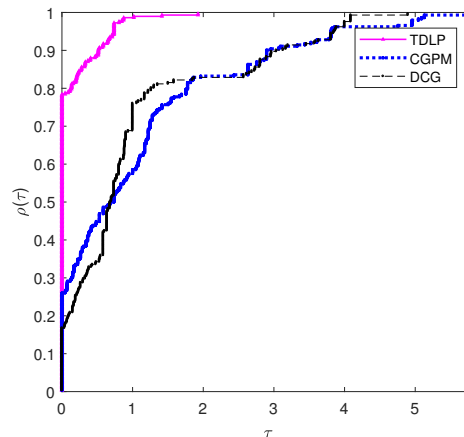


FIGURE 2. Performance profiles based on number of iterations

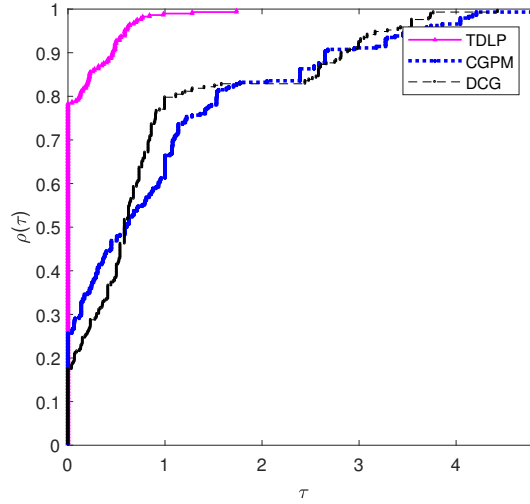


FIGURE 3. Performance profiles based on number of function evaluations

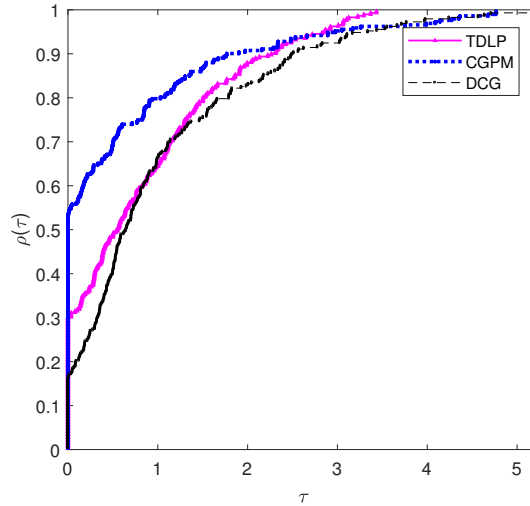


FIGURE 4. Performance profiles based on CPU time (in seconds)

#### 4. THE APPLICATIONS IN COMPRESSIVE SENSING

Compressive sensing (CS) plays an important role in medical sciences, biological engineering and other areas of science and engineering [29, 30]. The CS problem deals with recovering a sparse signal, say  $x$ , from the following linear system  $Ax = b$ , where the vector  $b \in \mathbb{R}^k$  is the observation, and the linear operator  $A \in \mathbb{R}^{k \times n}$  ( $k \ll n$ ) is the sensing matrix. In an attempt to search for the sparsest among the solution set, the above system of the linear equation is converted into the following so-called " $\ell_0$ -norm" optimization problem with constraints  $\min_x \{ \|x\|_0 : Ax = b \}$ , where  $\|x\|_0$  is given by the number of non-zero components of a vector  $x \in \mathbb{R}^n$ . Because  $\|\cdot\|_0$  is not positively homogeneous, it is not a norm. However, due to the

difficulty associated with  $\ell_0$ -norm problems, an alternative approach is to replace it with the  $\ell_1$ -norm (see [31]), that is,

$$\min_x \{\|x\|_1 : Ax = b\}. \quad (4.1)$$

where  $\|x\|_1 := \sum_{i=1}^n |x_i|$ . Under some suitable conditions, problem (4.1) has been shown to produce the desired results with acceptable degrees of accuracy. When the measurements are affected by some noise, the constraint in (4.1) is usually relaxed to the following regularized least squares problem

$$\min_x \tau \|x\|_1 + \frac{1}{2} \|Ax - b\|_2^2, \quad (4.2)$$

where  $\tau$  is a positive regularization parameter, and  $\|x\|_2$  denotes the Euclidean norm of  $x \in \mathbb{R}^n$ .

Iterative methods for solving problem (4.2) have been investigated; see, e.g., [6, 32, 33, 34] and the references therein. Due to the fact that the proposed algorithm (TDLP) is derivative-free, it can be applied to handle nonsmooth problems that are in the form of (1.3) effectively. Therefore, we solve the nonsmooth problem (4.2) in a different way by converting it into the form of (1.3). Fortunately, Xiao, Wang and Hu [7] translated problem (4.2) into a nonlinear system of equations based on the work of Figueiredo, Nowak and Wright [34] as follows:

Let  $x$  be any vector in  $\mathbb{R}^n$ . Then,  $x$  can be split into positive and negative parts, that is,  $x = u - v$ ,  $u \geq 0$ ,  $v \geq 0$ , where  $u \in \mathbb{R}^n$ ,  $v \in \mathbb{R}^n$ , and  $u_i = (x_i)_+$ ,  $v_i = (-x_i)_+$ , for all  $i = 1, 2, \dots, n$  with  $(\cdot)_+ = \max\{0, \cdot\}$ . The  $\ell_1$ -norm of a vector  $x$  can be represented as  $\|x\|_1 = e_n^T u + e_n^T v$ , where  $e_n = (1, 1, \dots, 1)^T \in \mathbb{R}^n$ . Hence, the  $\ell_1$ -norm problem (4.2) can be transformed as

$$\min_{u,v} \frac{1}{2} \|b - A(u - v)\|_2^2 + \tau e_n^T u + \tau e_n^T v, \text{ such that } \tau \geq 0, u \geq 0, v \geq 0. \quad (4.3)$$

However, from [34], problem (4.3) can also be rewritten as a quadratic programming problem with box constraints

$$\min_z \frac{1}{2} z^T D z + c^T z, \text{ such that } z \geq 0, \quad (4.4)$$

where  $z = \begin{bmatrix} u \\ v \end{bmatrix}$ ,  $c = \tau e_{2n} + \begin{bmatrix} -x \\ x \end{bmatrix}$ ,  $x = A^T b$  and  $D = \begin{bmatrix} A^T A & -A^T A \\ -A^T A & A^T A \end{bmatrix}$ .

Clearly,  $D$  is a positive semi-definite matrix, which implies that problem (4.4) is a convex quadratic problem. Xiao, Wang and Hu [7] translated problem (4.4) into a linear variable inequality problem, which is equivalent to a linear complementary problem. Furthermore, they pointed out that  $z$  is a solution of the linear complementary problem if and only if it is a solution of the nonlinear equation:  $F(z) = \min\{z, Dz + c\} = 0$ , where  $F(\cdot)$  is said to be continuous and monotone, see [7, 35]. Hence, solving problem (4.2) is equivalent to solve problem (1.3). Therefore, Algorithm 1 can be applied to solve problem (4.2) effectively.

Next, we employ the TDLP Algorithm in the reconstruction of sparse signal and image restoration problems. We begin the experiment with the reconstruction of one dimensional sparse signal of length  $n$  from  $k$  observation. The quality of the recovery to the original signal  $x$  is assessed by the mean square error (MSE) defined by  $MSE := \frac{1}{n} \|x - \tilde{x}\|^2$ , where  $\tilde{x}$  is the restored signal. Due to the limited memory of the PC, we select a small signal with a signal size of  $n = 4096$ ,  $k = 1024$  and the original signal contains 128 randomly non-zero elements. In addition, the measurement  $b$  is distributed with noise, i.e,  $b = Ax + \mu$ , where  $A$  is a randomly generated Gaussian matrix, and  $\mu$  is the Gaussian noise distributed normally with mean 0 and

variance  $10^{-4}$ . To evaluate the performance of the TDLP Algorithm in signal recovery problems, we compare it with the method of Gao, He and Liu in [36] denoted as AFP Algorithm. However, we choose the parameters:  $\rho = 0.8$ ,  $\kappa = 10$ ,  $q = 5$ ,  $c = 10$ ,  $t = 0.1$ ,  $\sigma = 0.001$ , and  $r = 0.001$  for the TDLP Algorithm while all the parameters for the AFP Algorithm are as contained in the respective paper. For fairness in comparison, each code was run from the same initial point, same continuation on the parameter  $\tau$  and observed only the behaviour of the convergence of each method to have a similar accurate solution. The experiment was initialized with  $x_0 = A^T b$  and terminates when

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

where  $f(x_k) := \frac{1}{2}\|y - Ax_k\|_2^2 + \tau\|x_k\|_1$  is the objective function. In Figure 5, we provide the numerical results consisting of the original sparse signal, the measurement and the reconstructed signal by the TDLP and AFP Algorithms. In addition, in Figure 6, we give a visual illustration of the numerical performance of each Algorithm relative to their convergence behavior from the view of the merit function values and relative error as the number of iterations and computing time increases. Comparing the two algorithms in Figure 5, it is not difficult to see that the original signal is recovered by the two algorithms. However, the TDLS Algorithm won in decoding sparse signal in compressive sensing. This is reflected by the lesser number of iterations, computing time and most importantly lesser MSE. To further illustrate the efficiency of the TDLP Algorithm relative to AFP, we repeated the experiment on 20 different noise samples. Each time the experiment is run, the TDLP Algorithm proves to be more efficient than the AFP Algorithm in almost all the experiments in terms of the number of iterations, CPU time and the MSE. See the summary in Table 1.

Next, we test the performance of the TDLP Algorithm in restoring blurred images. We considered the following well-known inbuilt gray test images: (P1) Lena, (P2) House, (P3) Peppers and (P4) Barbara for the experiments. We used five different Gaussian blur kernels with a standard deviation  $v$  to compare the performance of the TDLP Algorithm with that of the AFP Algorithm proposed in [36]. To assess the performance of each algorithm tested with respect to the metrics that indicate better quality restoration, in Table 2 we reported the objective function (ObjFun) at the approximate solution, the MSE, the signal-to-noise-ratio (SNR), which is defined as

$$SNR := 20 \times \log_{10} \left( \frac{\|\tilde{x}\|}{\|x - \tilde{x}\|} \right),$$

and the structural similarity (SSIM) index that measures the similarity between the original image and the restored image [37] for each of the 20 experiments. The MATLAB implementation of the SSIM index can be obtained at <http://www.cns.nyu.edu/~lcv/ssim/>. The original, blurred, and the restored images by each of the algorithm are given in Figure 7. The figure demonstrate that both algorithms can restore the blurred images. In Table 2, it can be observed that, under the five Gaussian blur kernel, the quality of the restored images by the TDLP Algorithm are much better than the ones restored by the AFP. This is reflected by the smaller values of the ObjFun and MSE. Similarly, the larger SNR and SSIM indicate that the restored images from the blurred images by the TDLP are much more closer to the original ones than the ones recovered by the AFD Algorithm in most cases.



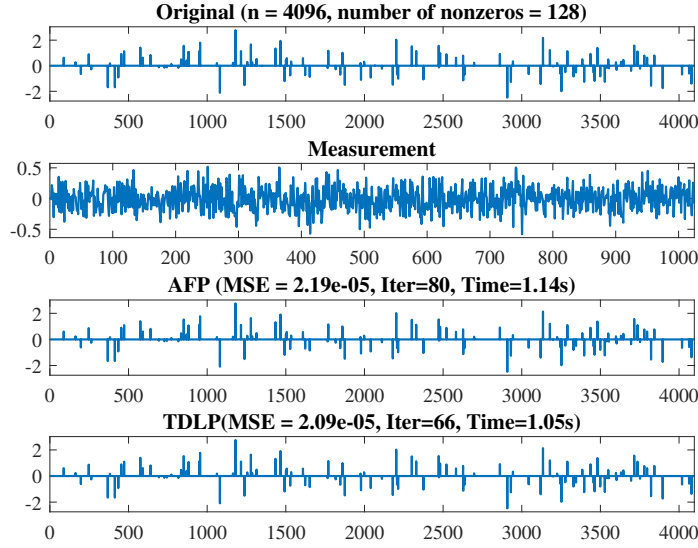


FIGURE 5. Reconstruction of sparse signal. From top to bottom is the original signal (First plot), the measurement (Second plot) and the reconstructed signals by AFP (Third plot) and TDLP (Fourth plot)

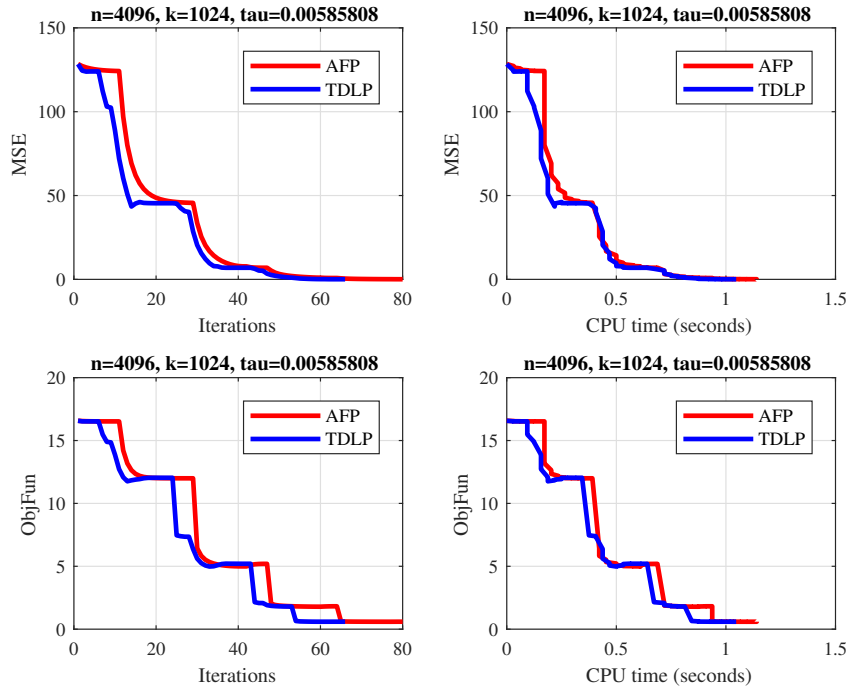


FIGURE 6. Comparison results of AFP and TDLP Algorithms. The  $x$ -axis represent the number of iterations (**top and bottom left**) and the CPU time in seconds (**top and bottom right**), the  $y$ -axis represent the MSE (**top left and right**) and the function values (**bottom left and right**)

TABLE 1. Twenty experimental results of  $\ell_1$ -norm regularization problem

ITER		TIME		MSE		
AFP	TDLP	AFP	TDLP	AFP	TDLP	
90	80	1.41	1.41	2.02E-05	2.21E-05	
80	66	1.41	1.05	2.19E-05	2.09E-05	
88	79	1.36	1.33	1.69E-05	1.69E-05	
87	77	1.38	1.25	1.95E-05	2.51E-05	
83	75	1.28	1.23	2.42E-05	2.43E-05	
77	90	1.23	1.64	2.07E-05	1.90E-05	
89	68	1.36	1.13	2.07E-05	1.19E-05	
93	84	1.42	1.48	2.32E-05	2.33E-05	
94	87	1.33	1.34	1.88E-05	2.08E-05	
89	78	1.28	1.20	2.12E-05	2.12E-05	
86	76	1.27	1.19	3.09E-05	3.13E-05	
89	68	1.28	1.09	2.29E-05	2.87E-05	
91	82	1.30	1.39	2.57E-05	2.58E-05	
89	76	1.34	1.25	2.83E-05	2.83E-05	
81	78	1.20	1.34	1.77E-05	1.76E-05	
84	81	1.49	1.33	3.17E-05	3.18E-05	
83	78	1.13	1.27	2.79E-05	2.99E-05	
79	77	1.19	1.30	1.60E-05	1.60E-05	
79	76	1.13	1.17	4.41E-05	3.32E-05	
89	74	1.28	1.23	4.39E-05	4.37E-05	
Average	86	77.5	1.30	1.28	2.48E-05	2.46E-05



FIGURE 7. The original images (first row), the blurred images (second row), the restored images by the TDLP Algorithm (third row) and the AFP Algorithm (fourth row)

TABLE 2. Efficiency comparison for image restoration between the TDLP and the AFP Algorithms under different Gaussian blur kernels  $Pi(v)$ 

Image	ObjFunn		MSE		SNR		SSIM	
	TDLP	AFP	TDLP	AFP	TDLP	AFP	TDLP	AFP
$P1(1 \times 10^{-4})$	5.73E+06	6.09E+06	33.60	47.70	27.19	25.66	0.90	0.88
$P1(1 \times 10^{-1})$	5.77E+06	6.10E+06	33.86	47.75	27.15	25.66	0.89	0.88
$P1(0.25)$	5.78E+06	6.12E+06	34.21	47.86	27.11	25.65	0.89	0.88
$P1(1)$	5.86E+06	6.21E+06	35.96	48.28	26.89	25.61	0.80	0.87
$P1(6.25)$	6.41E+06	6.08E+06	48.48	51.14	25.59	25.36	0.80	0.85
$P2(1 \times 10^{-1})$	1.61E+06	1.72E+06	39.99	62.34	27.24	25.31	0.88	0.86
$P2(1 \times 10^{-1})$	1.62E+06	1.72E+06	40.18	62.39	27.10	25.30	0.88	0.86
$P2(0.25)$	1.62E+06	1.73E+06	40.52	62.53	27.18	25.29	0.88	0.85
$P2(1)$	1.64E+06	1.75E+06	42.07	62.88	27.01	25.27	0.87	0.85
$P2(6.25)$	1.78E+06	1.90E+06	53.46	65.92	25.97	25.06	0.80	0.83
$P3(1 \times 10^{-1})$	1.44E+06	1.55E+06	10.02	13.22	22.38	21.17	0.86	0.84
$P3(1 \times 10^{-1})$	1.43E+06	1.55E+06	10.04	13.22	22.37	21.17	0.86	0.84
$P3(0.25)$	1.44E+06	1.55E+06	10.03	13.23	22.37	21.17	0.86	0.84
$P3(1)$	1.46E+06	1.57E+06	10.22	13.29	22.29	21.15	0.85	0.84
$P3(6.25)$	1.60E+06	1.73E+06	11.00	13.53	21.97	21.07	0.82	0.82
$P4(1 \times 10^{-4})$	5.61E+06	5.95E+06	19.43	22.57	19.36	18.71	0.77	0.73
$P4(1 \times 10^{-1})$	5.62E+06	5.96E+06	19.45	22.57	19.35	18.71	0.77	0.73
$P4(0.25)$	5.63E+06	5.98E+06	19.48	22.57	19.35	18.71	0.77	0.73
$P4(1)$	5.71E+06	6.06E+06	19.62	22.61	19.32	18.70	0.76	0.73
$P4(6.25)$	6.28E+06	6.67E+06	20.55	22.82	19.11	18.66	0.72	0.72

## 5. THE CONCLUSION

In this paper, we proposed a three-term Dai-Liao-like projection algorithm for solving a system of nonlinear monotone operator equations with convex constraints. Under some mild conditions, the global convergence of the proposed algorithm is established and the generated sequence of iterations converges to a solution of problem (1.3). Computational results obtained from solving a large-scale nonlinear monotone equation with convex constraints show that our method is robust and effective, and outperforms the CGPM and DCG algorithms. Finally, Algorithm 1 was applied for solving the  $\ell_1$ -regularized problem arising in compressive sensing. Computational results from reconstruction of sparse signal and restoration of blurred images show that the proposed algorithm is practical and efficient.

## Acknowledgments

The authors are grateful to the anonymous reviewers for their suggestions and comments that improved the presentation of this paper. The first author, Abubakar Bakoji Muhammad, was supported by the DAAD (Germany) and PTDF (Nigeria) scholarship at the Martin-Luther-University Halle-Wittenberg, 06099 Halle (Saale), Germany.

## REFERENCES

- [1] F. Kohsaka, W. Takahashi, Fixed point theorems for a class of nonlinear mappings related to maximal monotone operators in Banach spaces, Arch. Math. 91 (2008), 166-177.
- [2] T. L. Magnanti, G. Perakis, Solving variational inequality and fixed point problems by line searches and potential optimization, Math. Program, 101 (2004), 435-461.
- [3] W. Takahashi, Nonlinear Functional Analysis. Fixed Point Theory and its Applications, Yokohama Publishers, Yokohama, 2000.

- [4] W. Takahashi, Introduction to Nonlinear and Convex Analysis, Yokohama Publishers, Yokohama, 2009.
- [5] N. A. Iusem, V. M. Solodov, Newton-type methods with generalized distances for constrained optimization, *Optimization*, 41 (1997), 257-278.
- [6] E.T. Hale, W. Yin, Y. Zhang, A fixed-point continuation method for  $\ell_1$ -regularized minimization with applications to compressed sensing, CAAM Technical Report TR07-07, pp. 1-46, Rice University, Houston Texas, 2007.
- [7] Y. Xiao, Q. Wang, Q. Hu, Non-smooth equations based method for  $\ell_1$ -norm problems with applications to compressed sensing, *Nonlinear Anal.* 11 (2011), 3570-3577.
- [8] M.V. Solodov, B.F. Svaiter, A globally convergent inexact newton method for systems of monotone equations, In *Reformulation: Nonsmooth, Piecewise Smooth, Semismooth and Smoothing Methods*, vol. 22, pp. 355-369, Springer, 1998.
- [9] Y.H. Dai, L.Z. Liao, New conjugacy conditions and related nonlinear conjugate gradient methods, *Appl. Math. Optim.* 43 (2001), 87-101.
- [10] A. Perry, A modified conjugate gradient algorithm, *Oper. Res.* 26 (1978), 1073-1078.
- [11] S. Babaie-Kafaki, R. Ghanbari, A descent family of Dai–Liao conjugate gradient methods, *Optim. Meth. Softw.* 29 (2014), 583-591.
- [12] A.B. Abubakar, P. Kumam, A descent Dai-Liao conjugate gradient method for nonlinear equations, *Numer. Algo.* 81 (2019), 197–210.
- [13] A.B. Abubakar, P. Kumam, A.M. Awwal, A descent Dai-Liao projection method for convex constrained nonlinear monotone equations with applications, *Thai J. Math.* 17 (2019), 128-152.
- [14] M.Y. Waziri, K. Ahmed, J. Sabi’u, A Dai-Liao conjugate gradient method via modified secant equation for system of nonlinear equations, *Arabian J. Math.* 9 (2020), 443–457.
- [15] X. Fang, Q. Ni, A new derivative-free conjugate gradient method for large-scale nonlinear systems of equations, *Bull. Aust. Math. Soc.* 95 (2017), 500–511.
- [16] L. Zhang, W. Zhou, D.H. Li, A descent modified Polak–Ribière–Polyak conjugate gradient method and its global convergence, *IMA J. Numer. Anal.* 26 (2006), 629-640.
- [17] E.H. Zarantonello, Projections on convex sets in Hilbert space and spectral theory, In: *Contributions to Nonlinear Functional Analysis*, vol. 1, pp. 237-424, Academic Press, New York, London, 1971.
- [18] A.B. Abubakar, A.H. Ibrahim, A.B. Muhammad, C. Tammer, A modified descent Dai–Yuan conjugate gradient method for constraint nonlinear monotone operator equations, *Appl. Anal. Optim.* 4 (2020), 1-24.
- [19] A.M. Awwal, L. Wang, P. Kumam, H. Mohammad, A two-step spectral gradient projection method for system of nonlinear monotone equations and image deblurring problems, *Symmetry*, 12 (2020), 874-893.
- [20] L. Zheng, L. Yang, Y. Liang, A conjugate gradient projection method for solving equations with convex constraints, *J. Comput. Appl. Math.* 375 (2020), 112781.
- [21] A.B. Abubakar, P. Kumam, H. Mohammad, A.M. Awwal, An efficient conjugate gradient method for convex constrained monotone nonlinear equations with applications, *Mathematics*, 7 (2019), 767-792.
- [22] A.M. Awwal, P. Kumam, A.B. Abubakar, A modified conjugate gradient method for monotone nonlinear equations with convex constraints, *Appl. Numer. Math.* 145 (2019), 507-520.
- [23] Y. Xiao, H. Zhu, A conjugate gradient method to solve convex constrained monotone equations with applications in compressive sensing, *J. Math. Anal. Appl.* 1 (2013), 310-319.
- [24] Y. Bing, G. Lin, An efficient implementation of merrills method for sparse or partially separable systems of nonlinear equations, *SIAM J. Optim.* 2 (1991), 206-221.
- [25] P. Gao, C. He, An efficient three-term conjugate gradient method for nonlinear monotone equations with convex constraints, *Calcolo*, 55 (2018), 53.
- [26] W.L. Cruz, J. Martínez, M. Raydan, Spectral residual method without gradient information for solving large-scale nonlinear systems of equations, *Math. Comput.* 255 (2006), 1429-1448.
- [27] Y. Ding, Y. Xiao, J. Li, A class of conjugate gradient methods for convex constrained monotone equations, *Optimization* 66 (2017), 2309-2328.
- [28] E.D. Dolan, J.J. Moré, Benchmarking optimization software with performance profiles, *Math. Program*, 2 (2002), 201-213.
- [29] M.R. Banham, A.K. Katsaggelos, Digital image restoration, *IEEE Signal Processing Magazine*, 14 (1997), 24-41.

- [30] C.L. Chan, A.K. Katsaggelos, A.V. Sahakian, Image sequence filtering in quantum-limited noise with application to low-dose fluoroscopy, *IEEE Trans. Medicine Imaging*, 12 (1993), 610-621.
- [31] S.S. Chen, D.L. Donoho, M.A. Saunders, Atomic decomposition by basis pursuit, *SIAM Rev.* 43 (2001), 129-159.
- [32] A. Beck, M. Teboulle, A fast iterative shrinkage-thresholding algorithm for linear inverse problems, *SIAM J. Imaging Sci.* 2 (2009), 183-202.
- [33] E.G. Birgin, J. M. Martínez, M. Raydan, Nonmonotone spectral projected gradient methods on convex sets, *SIAM J. Optim.* 10 (2000), 1196-1211.
- [34] M.A.T. Figueiredo, R.D. Nowak, S.J. Wright, Gradient projection for sparse reconstruction: Application to compressed sensing and other inverse problems, *IEEE J. Selected Topics in Signal Process.* 4 (2007), 586-597.
- [35] J.S. Pang, Inexact Newton methods for the nonlinear complementarity problem, *Math. Program.* 36 (1986), 54-71.
- [36] P. Gao, C. He, Y. Liu, An adaptive family of projection methods for constrained monotone nonlinear equations with applications, *Appl. Math. Comput.* 359 (2019), 1-16.
- [37] Z. Wang, A.C. Bovik, H. R. Sheikh, E.P. Simoncelli, Image quality assessment: from error visibility to structural similarity, *IEEE Trans. Image Process.* 13 (2004), 600-612.