

Article

# A Novel Forward-Backward Algorithm for Solving Convex Minimization Problem in Hilbert Spaces

Suthep Suantai <sup>1</sup>, Kunrada Kankam <sup>2</sup> and Prasit Cholamjiak <sup>2,\*</sup>

<sup>1</sup> Research Center in Mathematics and Applied Mathematics, Department of Mathematics, Faculty of Science, Chiang Mai University, Chiang Mai 50200, Thailand; suthep.s@cmu.ac.th

<sup>2</sup> School of Science, University of Phayao, Phayao 56000, Thailand; kunradazzz@gmail.com

\* Correspondence: prasitch2008@yahoo.com or prasit.ch@up.ac.th

Received: 12 December 2019; Accepted: 21 December 2019; Published: 1 January 2020



**Abstract:** In this work, we aim to investigate the convex minimization problem of the sum of two objective functions. This optimization problem includes, in particular, image reconstruction and signal recovery. We then propose a new modified forward-backward splitting method without the assumption of the Lipschitz continuity of the gradient of functions by using the line search procedures. It is shown that the sequence generated by the proposed algorithm weakly converges to minimizers of the sum of two convex functions. We also provide some applications of the proposed method to compressed sensing in the frequency domain. The numerical reports show that our method has a better convergence behavior than other methods in terms of the number of iterations and CPU time. Moreover, the numerical results of the comparative analysis are also discussed to show the optimal choice of parameters in the line search.

**Keywords:** convex minimization problem; forward-backward splitting method; Hilbert space; line search

## 1. Introduction

Let  $H$  be a real Hilbert space, and let  $f, g : H \rightarrow \mathbb{R} \cup \{+\infty\}$  be proper, lower-semicontinuous, and convex functions. The convex minimization problem is modeled as follows:

$$\min_{x \in H} (f(x) + g(x)). \quad (1)$$

Throughout this paper, assume that Problem (1) is nonempty, and the solution set is denoted by  $S_*$ . We know that Problem (1) can be described by the fixed point equation, that is,

$$x = \text{prox}_{\alpha g}(x - \alpha \nabla f(x)) \quad (2)$$

where  $\alpha > 0$  and  $\text{prox}_g$  is the proximal operator of  $g$  defined by  $\text{prox}_g = (Id + \partial g)^{-1}$  where  $Id$  denotes the identity operator in  $H$  and  $\partial g$  is the classical convex subdifferential of  $g$ . Using this fixed point equation, one can define the following iteration process:

$$x^{k+1} = \underbrace{\text{prox}_{\alpha_k g}}_{\text{backward step}} \left( \underbrace{x^k - \alpha_k \nabla f(x^k)}_{\text{forward step}} \right), \quad (3)$$

where  $\alpha_k$  is a suitable stepsize. This algorithm is known as the forward-backward algorithm, and it includes, as special cases, the gradient method [1–3] and the proximal algorithm [4–7]. Recently, the construction of algorithms has become a crucial technique for solving some nonlinear and optimization problems (see also [8–15]).

In 2005, Combettes and Wajs [16] proposed the following algorithm, which is based on iteration (3) as follows:

**Algorithm 1** ([16]). Given  $\epsilon \in (0, \min\{1, \frac{1}{\alpha}\})$  and letting  $x^0 \in \mathbb{R}^N$ , for  $k \geq 1$ :

$$\begin{aligned} y^k &= x^k - \alpha_k \nabla f(x^k), \\ x^{k+1} &= x^k + \lambda_k (\text{prox}_{\alpha_k g} y^k - x^k), \end{aligned} \tag{4}$$

where  $\alpha_k \in [\epsilon, \frac{2}{\alpha} - \epsilon]$ ,  $\lambda_k \in [\epsilon, 1]$  and  $\alpha$  is the Lipschitz constant of the gradient of  $f$ .

It was proven that the sequence generated by (4) converges to minimizers of  $f + g$ . However, we note that the convergence of Algorithm 1 depends on the Lipschitz continuity of the gradient of  $f$ , which is not an easy task to find in general.

The Douglas–Rachford algorithm is another method that can be used to solve the problem (1). It is defined in the following manner:

**Algorithm 2** ([16]). Fix  $\epsilon \in (0, 1)$ ,  $\gamma > 0$  and let  $x^0 \in \mathbb{R}^N$ . For  $k \geq 1$ , calculate

$$\begin{aligned} y^k &= \text{prox}_{\gamma g} x^k, \\ x^{k+1} &= x^k + \lambda_k (\text{prox}_{\gamma f} (2y^k - x^k) - y^k), \end{aligned} \tag{5}$$

where  $\lambda_k \in [\epsilon, 2 - \epsilon]$ .

It was shown that the sequence  $(x^k)$  defined by (5) converges to minimizers of  $f + g$ . In this case, we see that the main drawback of Algorithm 2 is that it requires two proximity operators of convex functions  $f$  and  $g$  per iteration. This leads to a slow convergence speed of algorithms based on Algorithm 2. Please see Section 4 for its convergence.

Very recently, Cruz and Nghia [17] introduced the forward-backward algorithm by using the line search technique in the framework of Hilbert spaces. Assume that the following conditions are satisfied:

(A1)  $f, g : H \rightarrow \mathbb{R} \cup \{+\infty\}$  are two proper, l.s.c, convex functions where  $\text{dom}g \subseteq \text{dom}f$  and  $\text{dom}g$  is nonempty, closed, and convex.

(A2)  $f$  is Fréchet differentiable on an open set that contains  $\text{dom}g$ . The gradient  $\nabla f$  is uniformly continuous on bounded subsets of  $\text{dom}g$ . Moreover, it maps any bounded subset of  $\text{dom}g$  to a bounded set in  $H$ .

**Linesearch 1.** Let  $\sigma > 0$ ,  $\theta \in (0, 1)$  and  $\delta \in (0, \frac{1}{2})$ .

**Input.** Given  $\alpha = \sigma$  and  $J(x, \alpha) := \text{prox}_{\alpha g}(x - \alpha \nabla f(x))$  with  $x \in \text{dom}g$

**While**  $\alpha \|\nabla f(J(x, \alpha)) - \nabla f(x)\| > \delta \|J(x, \alpha) - x\|$

**do**  $\alpha = \theta \alpha$ .

**End While**

**Output.**  $\alpha$ .

It was proven that Line Search 1 is well defined, i.e., this line search stops after finitely many steps. By this fact, Cruz and Nghia [17] also considered the following algorithm:

**Algorithm 3.** Fix  $\sigma > 0$ ,  $\theta \in (0, 1)$  and  $\delta \in (0, \frac{1}{2})$ , and let  $x^0 \in \text{dom}g$ . For  $k \geq 1$ , calculate:

$$x^{k+1} = \text{prox}_{\alpha_k g}(x^k - \alpha_k \nabla f(x^k)),$$

with  $\alpha_k := \text{Line Search 1}(x^k, \sigma, \theta, \delta)$ .

They showed that the sequence  $(x^k)$  generated by Algorithm 3 involving the line search technique that eliminates the Lipschitz assumption on the gradient of  $f$  converges weakly to minimizers of  $f + g$ . It is observed that, to obtain its convergence, one has to find the stepsize  $\alpha$  in each iteration. This can be costly and time consuming in computation.

Recently, there have been many works on modifying the forward-backward method for solving convex optimization problems (see, for example, [18–23]).

In variational theory, Tseng [24] introduced the following method for solving the variational inequality problem (VIP):

$$\begin{aligned} y^k &= P_C(x^k - \lambda Fx^k), \\ x^{k+1} &= y^k + \lambda(Fy^k - Fx^k), \end{aligned} \tag{6}$$

where  $P_C$  is a metric projection from a Hilbert space onto the set  $C$ ,  $F$  is a monotone and  $L$ -Lipschitz continuous mapping, and  $\lambda \in (0, 1/L)$ . Then, the sequence generated by (6) weakly converges to a solution of VIP. This method is often called Tseng’s extragradient method and has received great attention by researchers due to its convergence speed (see, for example, [25–28]). Following this research direction, the main challenge is to design novel algorithms that can speed up the convergence rate compared to Algorithms 1–3.

In this paper, inspired by Cruz and Nghia [17], we suggest a new forward-backward algorithm to solve the convex minimization problem. We then prove weak convergence theorems of the proposed algorithm. Finally, some numerical experiments in signal recovery are given to show its efficiency. Numerical experiments show that our new algorithms have a better convergence behavior than other methods in comparison. The main advantage of this work is that our schemes do not require the computation of the Lipschitz constant as assumed in Algorithm 1.

The content is organized as follows: In Section 2, we recall the useful concepts that will be used in the sequel. In Section 3, we establish the main theorem of our algorithms. In Section 4, we give numerical experiments to validate the convergence theorems, and finally, in Section 5, we give the conclusions of this paper.

## 2. Preliminaries

In this section, we give some definitions and lemmas that play an essential role in our analysis. The strong and weak convergence of  $(x^k)_{k \in \mathbb{N}}$  to  $x$  will be denoted by  $x^k \rightarrow x$  and  $x^k \rightharpoonup x$ , respectively. The subdifferential of  $h$  at  $z$  is defined by:

$$\partial h(z) = \{u \in H : \langle u, x - z \rangle \leq h(x) - h(z), z \in H\}.$$

It is known that  $\partial h$  is maximal monotone [29].

The proximal operator of  $g$  is defined by  $\text{prox}_g : H \rightarrow \text{dom}g$  with  $\text{prox}_g(z) = (Id + \partial g)^{-1}(z), z \in H$ . We know that the  $\text{prox}_g$  is single valued with full domain. Moreover, we have:

$$\frac{z - \text{prox}_{\alpha g}(z)}{\alpha} \in \partial g(\text{prox}_{\alpha g}(z)) \text{ for all } z \in H, \alpha > 0. \tag{7}$$

The following lemma is crucial in convergence analysis.

**Lemma 1** ([29]). *Let  $H$  be a Hilbert space. Let  $S$  be a nonempty, closed, and convex set of  $H$ , and let  $(x^k)$  be a sequence in  $H$  that satisfies:*

- (i)  $\lim_{k \rightarrow \infty} \|x^k - x\|$  exists for each  $x \in S$ ;
- (ii)  $\omega_w(x^k) \subset S$ .

*Then,  $(x^k)$  weakly converges to an element of  $S$ .*

### 3. Main Results

In this section, we suggest a new forward-backward algorithm and prove the weak convergence. Next, we assume that Conditions (A1)–(A2) hold.

**Algorithm 4. Step 0.** Given  $\sigma > 0, \theta \in (0, 1), \gamma \in (0, 2)$ , and  $\delta \in (0, \frac{1}{2})$ . Let  $x^0 \in \text{dom}g$ .

**Step 1.** Calculate:

$$y^k = \text{prox}_{\alpha_k g}(x^k - \alpha_k \nabla f(x^k))$$

where  $\alpha_k = \sigma \theta^{m_k}$  and  $m_k$  is the smallest nonnegative integer such that:

$$\alpha_k \|\nabla f(x^k) - \nabla f(y^k)\| \leq \delta \|x^k - y^k\|. \tag{8}$$

If  $x^k = y^k$ , then stop, and  $y^k$  is a solution of  $S_*$ . Else, go to Step 2.

**Step 2.** Calculate:

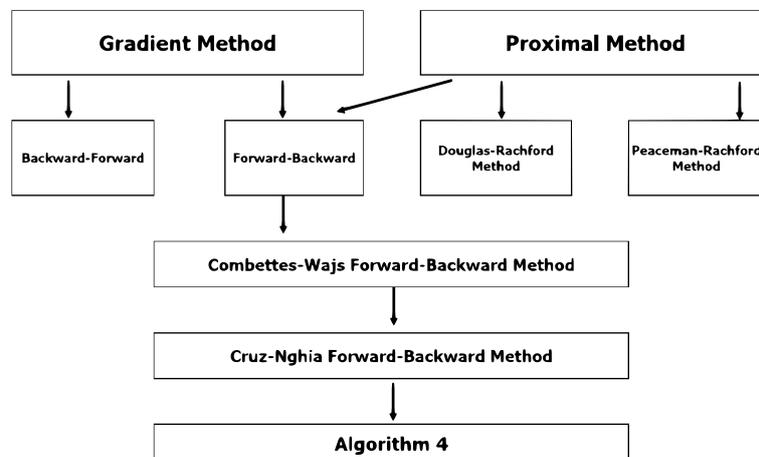
$$x^{k+1} = x^k - \gamma \eta_k d_k$$

where:

$$d_k = x^k - y^k - \alpha_k (\nabla f(x^k) - \nabla f(y^k)) \quad \text{and} \quad \eta_k = (1 - \delta) \frac{\|x^k - y^k\|^2}{\|d_k\|^2}.$$

Set  $k := k + 1$ , and go to **Step 1**.

We next summarize the methods for solving the convex minimization problem (CMP) in Figure 1.



**Figure 1.** The flowchart of the method for the convex minimization problem (CMP).

**Theorem 1.** Let  $(x^k)_{k \in \mathbb{N}}$  and  $(\alpha_k)_{k \in \mathbb{N}}$  be generated by Algorithm 4. If there is  $\alpha > 0$  such that  $\alpha_k \geq \alpha > 0$  for all  $k \in \mathbb{N}$ , then  $(x^k)_{k \in \mathbb{N}}$  weakly converges to an element of  $S_*$ .

**Proof.** Let  $x_*$  be a solution in  $S_*$ . Then, we obtain:

$$\begin{aligned} \|x^{k+1} - x_*\|^2 &= \|x^k - \gamma \eta_k d_k - x_*\|^2 \\ &= \|x^k - x_*\|^2 - 2\gamma \eta_k \langle x^k - x_*, d_k \rangle + \gamma^2 \eta_k^2 \|d_k\|^2. \end{aligned} \tag{9}$$

Using the definition of  $d_k$  and the line search (8), we have:

$$\begin{aligned}
 & \langle x^k - x_*, d_k \rangle \\
 = & \langle x^k - y^k, d_k \rangle + \langle y^k - x_*, d_k \rangle \\
 = & \langle x^k - y^k, x^k - y^k - \alpha_k(\nabla f(x^k) - \nabla f(y^k)) \rangle + \langle y^k - x_*, x^k - y^k - \alpha_k(\nabla f(x^k) - \nabla f(y^k)) \rangle \quad (10) \\
 = & \|x^k - y^k\|^2 - \langle x^k - y^k, \alpha_k(\nabla f(x^k) - \nabla f(y^k)) \rangle + \langle y^k - x_*, x^k - y^k - \alpha_k(\nabla f(x^k) - \nabla f(y^k)) \rangle \\
 \geq & \|x^k - y^k\|^2 - \alpha_k \|x^k - y^k\| \|\nabla f(x^k) - \nabla f(y^k)\| + \langle y^k - x_*, x^k - y^k - \alpha_k(\nabla f(x^k) - \nabla f(y^k)) \rangle \\
 \geq & \|x^k - y^k\|^2 - \delta \|x^k - y^k\|^2 + \langle y^k - x_*, x^k - y^k - \alpha_k(\nabla f(x^k) - \nabla f(y^k)) \rangle.
 \end{aligned}$$

Since  $y^k = \text{prox}_{\alpha_k g}(x^k - \alpha_k \nabla f(x^k))$ , it follows that  $(I - \alpha_k \nabla f)x^k \in (I - \alpha_k \partial g)y^k$ . Moreover, since  $\partial g$  is maximal monotone, there is  $v^k \in \partial g(y^k)$  such that:

$$(I - \alpha_k \nabla f)x^k = y^k - \alpha_k v^k.$$

This shows that:

$$v^k = \frac{1}{\alpha_k}(x^k - y^k - \alpha_k \nabla f(x^k)). \quad (11)$$

On the other hand,  $0 \in \nabla f(x_*) + \partial g(x_*) \subseteq \partial(f + g)(x_*)$  and  $\nabla f(y^k) + v^k \in \partial(f + g)y^k$ . Therefore, we see that:

$$\langle \nabla f(y^k) + v^k, y^k - x_* \rangle \geq 0. \quad (12)$$

Substituting (11) into (12), we get that:

$$\frac{1}{\alpha_k} \langle x^k - y^k - \alpha_k \nabla f(x^k) + \alpha_k \nabla f(y^k), y^k - x_* \rangle \geq 0. \quad (13)$$

This gives:

$$\langle x^k - y^k - \alpha_k \nabla f(x^k) + \alpha_k \nabla f(y^k), y^k - x_* \rangle \geq 0. \quad (14)$$

Combining (10) and (14), we obtain:

$$\langle x^k - x_*, d_k \rangle \geq (1 - \delta) \|x^k - y^k\|^2. \quad (15)$$

Since  $\eta_k = (1 - \delta) \frac{\|x^k - y^k\|^2}{\|d_k\|^2}$ , we have  $\eta_k \|d_k\|^2 = (1 - \delta) \|x^k - y^k\|^2$ . This shows that:

$$-2\gamma\eta_k(1 - \delta) \|x^k - y^k\|^2 = -2\gamma\eta_k^2 \|d_k\|^2. \quad (16)$$

By (15) and (16), we see that:

$$\begin{aligned}
 -2\gamma\eta_k \langle x^k - x_*, d_k \rangle & \leq -2\gamma\eta_k(1 - \delta) \|x^k - y^k\|^2 \\
 & = -2\gamma\eta_k^2 \|d_k\|^2.
 \end{aligned}$$

Therefore, from (9) and the above, we have:

$$\begin{aligned}
 \|x^{k+1} - x_*\|^2 & \leq \|x^k - x_*\|^2 - 2\gamma\eta_k^2 \|d_k\|^2 + \gamma^2 \eta_k^2 \|d_k\|^2 \\
 & = \|x^k - x_*\|^2 - \frac{2}{\gamma} \|\gamma\eta_k d_k\|^2 + \|\gamma\eta_k d_k\|^2 \\
 & = \|x^k - x_*\|^2 - \frac{2 - \gamma}{\gamma} \|\gamma\eta_k d_k\|^2. \quad (17)
 \end{aligned}$$

Since  $x^{k+1} = x^k - \gamma\eta_k d_k$ , it follows that  $\gamma\eta_k d_k = x^k - x^{k+1}$ . This implies:

$$\|x^{k+1} - x_*\|^2 \leq \|x^k - x_*\|^2 - \frac{2-\gamma}{\gamma} \|x^k - x^{k+1}\|^2. \tag{18}$$

Thus,  $\lim_{k \rightarrow \infty} \|x^k - x_*\|$  exists, and hence,  $(x^k)$  is bounded. This yields  $\lim_{k \rightarrow \infty} \|x^k - x^{k+1}\| = 0$ . We note, by (18), that:

$$\frac{2-\gamma}{\gamma} \|x^k - x^{k+1}\|^2 \leq \|x^k - x_*\|^2 - \|x^{k+1} - x_*\|^2.$$

By the monotonicity of  $\nabla f$ , we see that:

$$\begin{aligned} \|d_k\|^2 &= \|x^k - y^k - \alpha_k(\nabla f(x^k) - \nabla f(y^k))\|^2 \\ &= \|x^k - y^k\|^2 + \alpha_k^2 \|\nabla f(x^k) - \nabla f(y^k)\|^2 - 2\alpha_k \langle x^k - y^k, \nabla f(x^k) - \nabla f(y^k) \rangle \\ &\leq (1 + \delta^2) \|x^k - y^k\|^2, \end{aligned}$$

equivalently:

$$\frac{1}{\|d_k\|^2} \geq \frac{1}{(1 + \delta^2) \|x^k - y^k\|^2}.$$

Therefore, we have:

$$\eta_k = (1 - \delta) \frac{\|x^k - y^k\|^2}{\|d_k\|^2} \geq \frac{1 - \delta}{1 + \delta^2}. \tag{19}$$

On the other hand, we have:

$$\eta_k \|d_k\|^2 = (1 - \delta) \|x^k - y^k\|^2. \tag{20}$$

Therefore, it follows that:

$$\begin{aligned} \|x^k - y^k\|^2 &= \frac{1}{1 - \delta} (\eta_k \|d_k\|^2) \\ &= \frac{1}{(1 - \delta)(\gamma^2 \eta_k)} (\|\gamma\eta_k d_k\|^2) \\ &= \frac{1}{(1 - \delta)(\gamma^2 \eta_k)} \|x^{k+1} - x^k\|^2. \end{aligned} \tag{21}$$

Combining (20) and (21), we obtain:

$$\|x^k - y^k\|^2 \leq \frac{1 + \delta^2}{[(1 - \delta)\gamma]^2} \|x^{k+1} - x^k\|^2.$$

Since  $\|x^{k+1} - x^k\| \rightarrow 0$  as  $k \rightarrow \infty$ ,  $\|x^k - y^k\| \rightarrow 0$  as  $k \rightarrow \infty$ . By the boundedness of  $(x^k)_{k \in \mathbb{N}}$ , we know that the set of its weak accumulation points is nonempty. Let  $x^\infty$  be a weak accumulation point of  $(x^k)_{k \in \mathbb{N}}$ . Therefore, there is a subsequence  $(x^{n_k})_{k \in \mathbb{N}}$  of  $(x^k)_{k \in \mathbb{N}}$ . Next, we show that  $x^\infty \in S_*$ . Let  $(v, u) \in \text{Graph}(\nabla(f) + \partial(g))$ , that is  $u - \nabla f(v) \in \partial g(v)$ . Since  $y^{k_n} = (I + \alpha_{k_n} \partial g)^{-1} (I - \alpha_{k_n} \nabla f) x^{k_n}$ , we obtain:

$$(I - \alpha_{k_n} \nabla f) x^{k_n} \in (I + \alpha_{k_n} \partial g) y^{k_n},$$

which yields:

$$\frac{1}{\alpha_{k_n}} (x^{k_n} - y^{k_n} - \alpha_{k_n} \nabla f(x^{k_n})) \in \partial g(y^{k_n}).$$

Since  $\partial g$  is maximal, we have:

$$\langle v - y^{k_n}, u - \nabla f(v) - \frac{1}{\alpha_{k_n}}(x^{k_n} - y^{k_n} - \alpha_{k_n} \nabla f(x^{k_n})) \rangle \geq 0.$$

This shows that

$$\begin{aligned} \langle v - y^{k_n}, u \rangle &\geq \langle v - y^{k_n}, \nabla f(v) + \frac{1}{\alpha_{k_n}}(x^{k_n} - y^{k_n} - \alpha_{k_n} \nabla f(x^{k_n})) \rangle \\ &= \langle v - y^{k_n}, \nabla f(v) - \nabla f(x^{k_n}) \rangle + \langle v - y^{k_n}, \frac{1}{\alpha_{k_n}}(x^{k_n} - y^{k_n}) \rangle \\ &= \langle v - y^{k_n}, \nabla f(v) - \nabla f(y^{k_n}) \rangle + \langle v - y^{k_n}, \nabla f(y^{k_n}) - \nabla f(x^{k_n}) \rangle \\ &\quad + \langle v - y^{k_n}, \frac{1}{\alpha_{k_n}}(x^{k_n} - y^{k_n}) \rangle \\ &\geq \langle v - y^{k_n}, \nabla f(y^{k_n}) - \nabla f(x^{k_n}) \rangle + \langle v - y^{k_n}, \frac{1}{\alpha_{k_n}}(x^{k_n} - y^{k_n}) \rangle. \end{aligned}$$

Since  $\lim_{k \rightarrow \infty} \|x^k - y^k\| = 0$  and by (A2), we have  $\lim_{k \rightarrow \infty} \|\nabla f(x^k) - \nabla f(y^k)\| = 0$ . Hence, we obtain:

$$\langle v - x^\infty, u \rangle = \lim_{n \rightarrow \infty} \langle v - y^{k_n}, u \rangle \geq 0. \tag{22}$$

Hence, we obtain  $0 \in (\nabla f + \partial g)x^\infty$ , and consequently,  $x^\infty \in S_*$ . This shows that  $(x^k)$  converges weakly to an element of  $S_*$  by applying Lemma 1. We thus complete the proof.  $\square$

#### 4. Numerical Experiments

Next, we apply our result to the signal recovery in compressive sensing. We show the performance of our proposed Algorithm 4, Algorithm 1 of Combettes and Wajs [16], Algorithm 2 of Douglas–Rachford [22], and Algorithm 3 of Cruz and Nghia [17]. This problem can be modeled as:

$$y = Ax + \epsilon, \tag{23}$$

where  $y \in \mathbb{R}^M$  is the observed data,  $\epsilon$  is the noise,  $A : \mathbb{R}^N \rightarrow \mathbb{R}^M (M < N)$  is a bounded and linear operator, and  $x \in \mathbb{R}^N$  is a recovered vector containing  $m$  nonzero components. It is known that (23) can be modeled as the LASSO problem:

$$\min_{x \in \mathbb{R}^N} \left( \frac{1}{2} \|Ax - y\|_2^2 + \lambda \|x\|_1 \right), \tag{24}$$

where  $\lambda > 0$ . Therefore, we can apply the proposed method to solve (1) when  $f(x) = \frac{1}{2} \|y - Ax\|_2^2$  and  $g(x) = \lambda \|x\|_1$ .

In experiment,  $y$  is generated by the Gaussian noise with SNR = 40,  $A$  is generated by the normal distribution with mean zero and variance one, and  $x \in \mathbb{R}^N$  is generated by a uniform distribution in  $[-2, 2]$ . We use the stopping criterion by:

$$\text{MSE} = \frac{1}{N} \|x^k - x^*\|^2 < 10^{-3},$$

where  $x^k$  is an estimated signal of  $x_*$ .

In the following, the initial point  $x^0$  is chosen randomly, and  $\alpha_k$  in Algorithm 1 is  $\frac{1}{\|A\|^2}$  and  $\lambda_k = 0.5$ . In Algorithm 2,  $\lambda_k = 0.02$  and  $\gamma = 0.02$ . Let  $\sigma = 100$ ,  $\theta = 0.1$ , and  $\delta = 0.1$  in Algorithm 3 and Algorithm 4, and let  $\gamma = 1.9$  in Algorithm 4. We denote by CPU the time of CPU and by iter the number of iterations. In Table 1, we test the experiment five times and then calculate the averages of

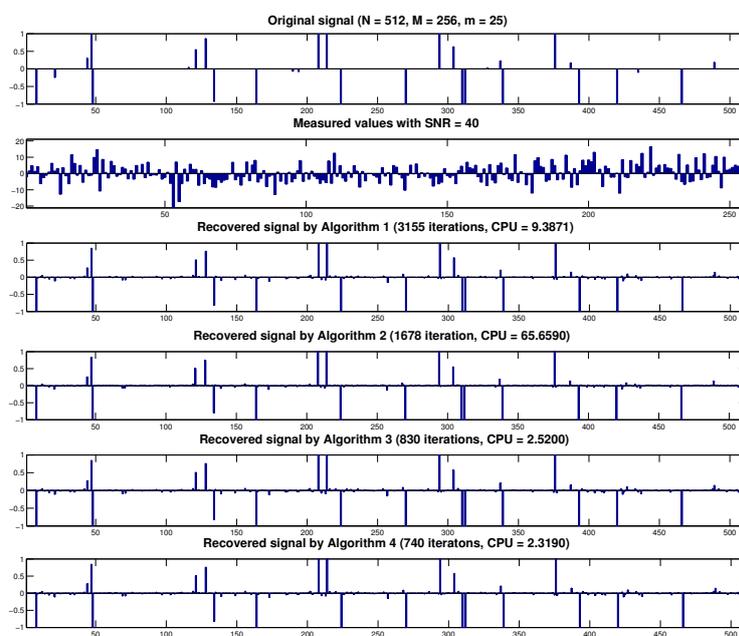
CPU and iter. All numerical experiments presented were obtained from MATLAB R2010a running on the same laptop computer. The numerical results are shown as follows:

**Table 1.** Computational results for compressed sensing.

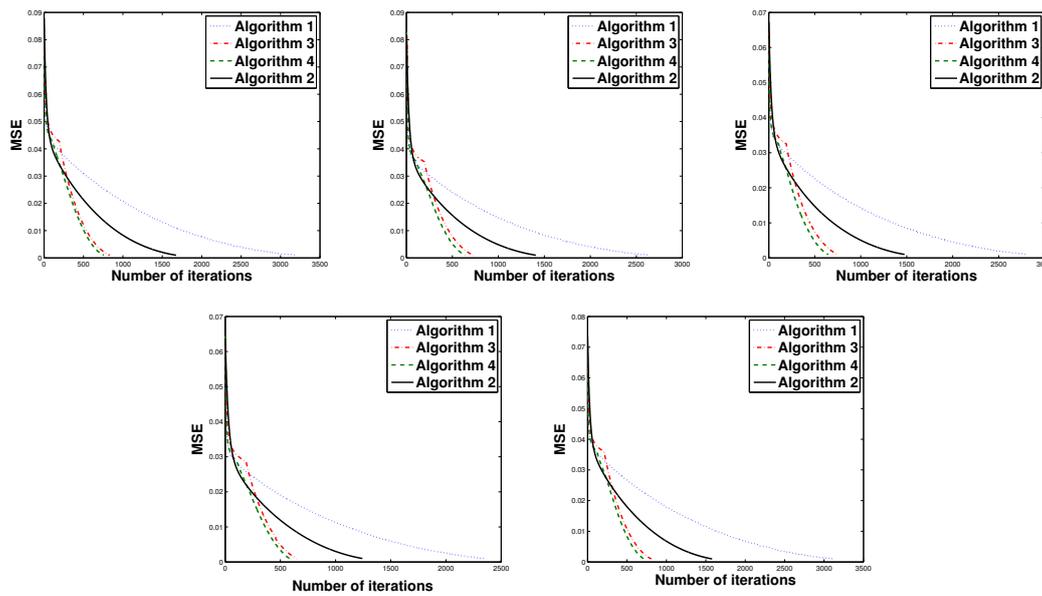
<i>m</i> -Sparse Signal	Method	<i>N</i> = 512, <i>M</i> = 256	
		CPU	Iter
<i>m</i> = 10	Algorithm 1	2.4095	1150
	Algorithm 2	22.1068	608
	Algorithm 3	0.9103	399
	Algorithm 4	0.7775	326
<i>m</i> = 20	Algorithm 1	7.2877	2073
	Algorithm 2	41.1487	1106
	Algorithm 3	1.5127	589
	Algorithm 4	1.3803	516
<i>m</i> = 25	Algorithm 1	10.2686	2463
	Algorithm 2	50.6234	1310
	Algorithm 3	1.9321	706
	Algorithm 4	1.7597	611
<i>m</i> = 30	Algorithm 1	13.4585	2825
	Algorithm 2	57.5277	1477
	Algorithm 3	2.1781	751
	Algorithm 4	2.0341	669

From Table 1, we see that the experiment result of Algorithm 4 was better than those of Algorithms 1 and 2 in terms of CPU time and number of iterations in each cases.

Next, we provide Figure 2 to show signal recovery in compressed sensing for one example and Figure 3 to show the convergence of each algorithm for all cases via the graph of the MSE value and number of iterations when *N* = 512, *M* = 256, and *m* = 20.



**Figure 2.** Original signal, observed data, and recovered signal by Algorithm 1, Algorithm 2, Algorithm 3, and Algorithm 4 when *N* = 512 and *M* = 256.



**Figure 3.** The MSE value and number of iterations for all cases when  $N = 512$  and  $M = 256$ .

From Figures 2 and 3, it is revealed that the convergence speed of Algorithm 4 was better than the other algorithm. To be more precise, Algorithm 2 had the highest CPU time since it required two proximity operators in computation per iteration. Moreover, Algorithm 1 that had the stepsize that was bounded above by the Lipschitz constant had the highest number of iterations. In our experiments, it was observed that the initial guess did not have any significant effect on the convergence behavior.

Next, we analyze the convergence and the effects of the stepsizes, which depended on parameters  $\delta$ ,  $\theta$ ,  $\gamma$ , and  $\sigma$  in Algorithm 4.

We next study the effect of the parameter  $\delta$  in the proposed algorithm for each value of  $\delta$ .

From Table 2, we observe that the CPU time and the number of iterations of Algorithm 4 became larger when the parameter  $\delta$  approached 0.5 when  $N = 512$ ,  $M = 265$  and  $N = 1024$ ,  $M = 512$ . Figure 4 shows the numerical results for each  $\delta$ .

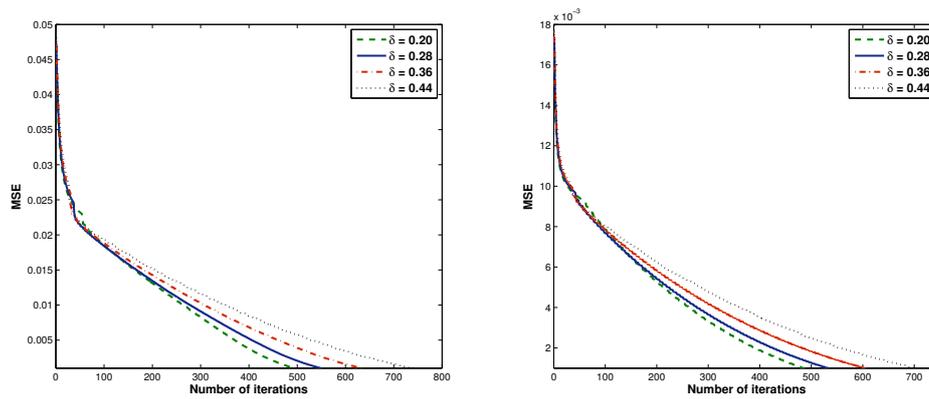
**Table 2.** The convergence of Algorithm 4 with each  $\delta$ .

Given: $\sigma = 100, \theta = 0.1, \gamma = 1.9$				
$\delta$	$N = 512, M = 256$		$N = 1024, M = 512$	
	$m = 20$		$m = 20$	
	CPU	Iter	CPU	Iter
0.20	3.2934	495	12.6411	487
0.28	3.4913	550	14.9438	533
0.36	3.4361	635	16.5947	605
0.44	4.4041	743	20.6721	703

From Figure 4, we see that our algorithm worked effectively when the value of  $\delta$  was taken close to zero.

Next, we investigate the effect of the parameter  $\theta$  in the proposed algorithm. We intend to vary this parameter and study its convergence behavior. The numerical results are shown in Table 3.

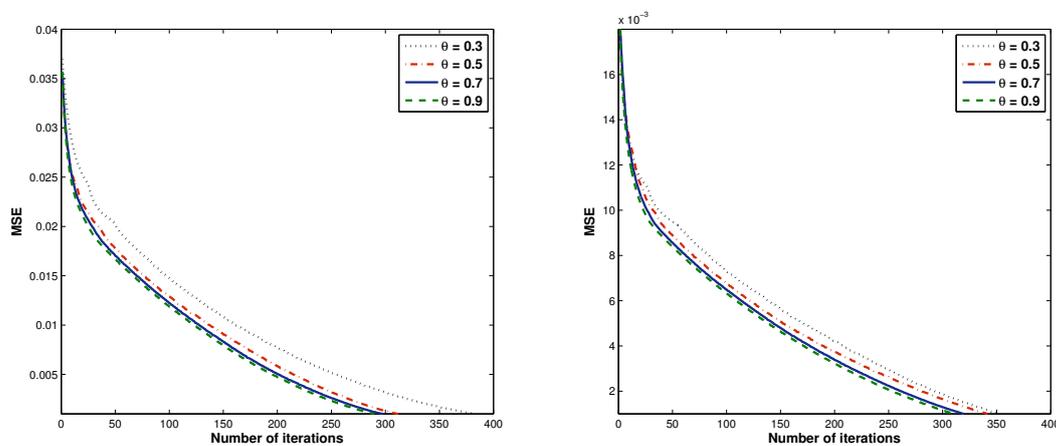
From Table 3, we observe that the CPU time of Algorithm 4 became larger and the number of iterations had a small reduction when the parameter  $\theta$  approached one when  $N = 512$ ,  $M = 265$  and  $N = 1024$ ,  $M = 512$ . Figure 5 shows the numerical results for each  $\theta$ .



**Figure 4.** Graph of number of iterations versus MSE when  $N = 512, M = 265$  and  $N = 1024, M = 512$ , respectively.

**Table 3.** The convergence of Algorithm 4 with each  $\theta$ .

Given: $\sigma = 100, \delta = 0.1, \gamma = 1.9$				
$\theta$	$N = 512, M = 256$		$N = 1024, M = 512$	
	$m = 20$		$m = 20$	
	CPU	Iter	CPU	Iter
0.3	4.0185	382	13.4732	354
0.5	4.2358	312	22.2328	342
0.7	7.5081	298	27.8198	320
0.9	21.7847	290	85.3605	311



**Figure 5.** Graph of the number of iterations versus MSE where  $N = 512, M = 265$  and  $N = 1024, M = 512$ , respectively.

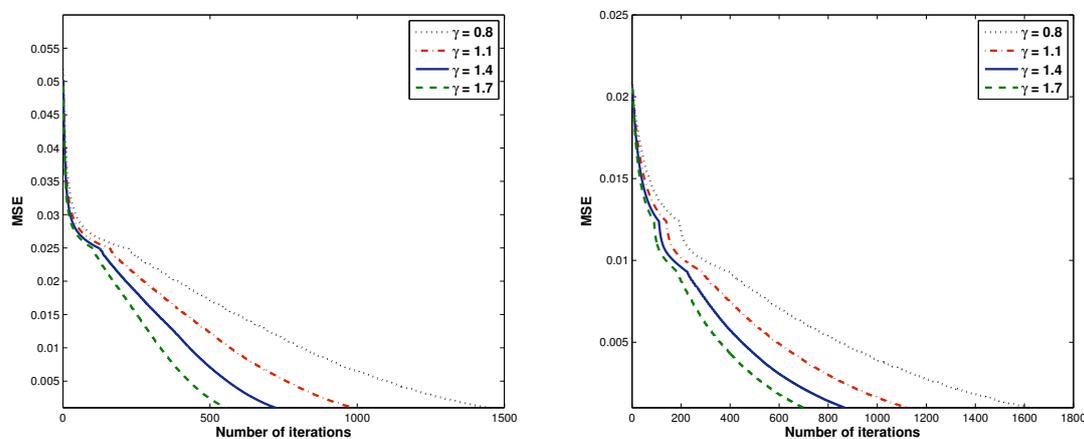
From Figure 5, it is shown that Algorithm 4 worked effectively when the value of  $\theta$  was chosen close to one.

Next, we study the effect of the parameter  $\gamma$  in the proposed algorithm. The numerical results are shown in Table 4.

**Table 4.** The convergence of Algorithm 4 with each  $\gamma$ .

Given: $\sigma = 100, \theta = 0.1, \delta = 0.1$				
$\gamma$	$N = 512, M = 256$ $m = 20$		$N = 1024, M = 512$ $m = 20$	
	CPU	Iter	CPU	Iter
0.8	13.9690	1450	38.2790	1624
1.1	7.8978	988	23.3411	1124
1.4	5.1804	724	17.6275	866
1.7	3.8323	554	13.6241	700

From Table 4, we see that the CPU time and the number of iterations of Algorithm 4 became smaller when the parameter  $\gamma$  approached two when  $N = 512, M = 265$  and  $N = 1024, M = 512$ . We show numerical results for each cases of  $\gamma$  in Figure 6.



**Figure 6.** Graph of the number of iterations versus MSE where  $N = 512, M = 265$  and  $N = 1024, M = 512$ , respectively.

From Figure 6, it is shown that Algorithm 4 worked effectively when the value of  $\gamma$  was chosen close to two.

Next, we study the effect of the parameter  $\sigma$  in the proposed algorithm. The numerical results are given in Table 5.

**Table 5.** The convergence of Algorithm 4 with each  $\sigma$ .

Given: $\delta = 0.1, \theta = 0.1, \gamma = 1.9$				
$\sigma$	$N = 512, M = 256$ $m = 20$		$N = 1024, M = 512$ $m = 20$	
	CPU	Iter	CPU	Iter
5	4.7055	680	11.7794	496
30	6.2280	843	11.3827	453
55	4.4271	634	14.2537	532
80	3.9635	606	16.5507	596

From Table 5, we see that the parameter  $\sigma$  had no effect in terms of the number of iterations and CPU time when  $N = 512, M = 265$  and  $N = 1024, M = 512$ .

## 5. Conclusions

In this work, we studied the modified forward-backward splitting method using line searches to solve convex minimization problems. We proved the weak convergence theorem under some weakened assumptions on the stepsize. It was found that the proposed algorithm had a better convergence behavior than other methods through experiments. Our algorithms did not require the Lipschitz condition on the gradient of functions. We also presented numerical experiments in signal recovery and provided a comparison to other algorithms. Moreover, the effects of all parameters were shown in Section 4. This main advantage was very useful and convenient in practice for solving some optimization problems.

**Author Contributions:** Supervision, S.S.; formal analysis and writing, K.K.; editing and software, P.C. All authors have read and agreed to the published version of the manuscript. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research was funded by Chiang Mai University.

**Conflicts of Interest:** The authors declare no conflict of interest.

## References

1. Dunn, J.C. Convexity, monotonicity, and gradient processes in Hilbert space. *J. Math. Anal. Appl.* **1976**, *53*, 145–158. [[CrossRef](#)]
2. Wang, C.; Xiu, N. Convergence of the gradient projection method for generalized convex minimization. *Comput. Optim. Appl.* **2000**, *16*, 111–120. [[CrossRef](#)]
3. Xu, H.K. Averaged mappings and the gradient-projection algorithm. *J. Optim. Theory Appl.* **2011**, *150*, 360–378. [[CrossRef](#)]
4. Guler, O. On the convergence of the proximal point algorithm for convex minimization. *SIAM J. Control Optim.* **1991**, *29*, 403–419. [[CrossRef](#)]
5. Martinet, B. Brève communication. Régularisation d'inéquations variationnelles par approximations successives. *Revue française d'informatique et de recherche opérationnelle. Série rouge.* **1970**, *4*, 154–158.
6. Parikh, N.; Boyd, S. Proximal algorithms. *Found. Trends Optim.* **2014**, *1*, 127–239. [[CrossRef](#)]
7. Rockafellar, R.T. Monotone operators and the proximal point algorithm. *SIAM J. Control Optim.* **1976**, *14*, 877–898. [[CrossRef](#)]
8. Zhang, Z.; Hong, W.C. Electric load forecasting by complete ensemble empirical mode decomposition adaptive noise and support vector regression with quantum-based dragonfly algorithm. *Nonlinear Dyn.* **2019**, *98*, 1107–1136. [[CrossRef](#)]
9. Kundra, H.; Sadawarti, H. Hybrid algorithm of cuckoo search and particle swarm optimization for natural terrain feature extraction. *Res. J. Inf. Technol.* **2015**, *7*, 58–69. [[CrossRef](#)]
10. Chalamjiak, W.; Chalamjiak, P.; Suantai, S. Convergence of iterative schemes for solving fixed point problems for multi-valued nonself mappings and equilibrium problems. *J. Nonlinear Sci. Appl.* **2015**, *8*, 1245–1256. [[CrossRef](#)]
11. Chalamjiak, P.; Suantai, S. Viscosity approximation methods for a nonexpansive semigroup in Banach spaces with gauge functions. *Global Optim.* **2012**, *54*, 185–197. [[CrossRef](#)]
12. Chalamjiak, P. The modified proximal point algorithm in CAT (0) spaces. *Optim. Lett.* **2015**, *9*, 1401–1410. [[CrossRef](#)]
13. Chalamjiak, P.; Suantai, S. Weak convergence theorems for a countable family of strict pseudocontractions in Banach spaces. *Fixed Point Theory Appl.* **2010**, *2010*, 632137. [[CrossRef](#)]
14. Shehu, Y.; Chalamjiak, P. Another look at the split common fixed point problem for demicontractive operators. *RACSAM Rev. R. Acad. Cienc. Exactas Fis. Nat. Ser. A Mat.* **2016**, *110*, 201–218. [[CrossRef](#)]
15. Suparatulatorn, R.; Chalamjiak, P.; Suantai, S. On solving the minimization problem and the fixed-point problem for nonexpansive mappings in CAT (0) spaces. *Optim. Methods Softw.* **2017**, *32*, 182–192. [[CrossRef](#)]
16. Combettes, P.L.; Wajs, V.R. Signal recovery by proximal forward-backward splitting. *Multiscale Model. Simul.* **2005**, *4*, 1168–1200. [[CrossRef](#)]

17. Bello Cruz, J. Y.; Nghia, T. T. On the convergence of the forward-backward splitting method with line searches. *Optim. Methods Softw.* **2016**, *31*, 1209–1238. [[CrossRef](#)]
18. Bauschke, H.H.; Borwein, J.M. Dykstra's alternating projection algorithm for two sets. *J. Approx. Theory.* **1994**, *79*, 418–443. [[CrossRef](#)]
19. Chen, G.H.G.; Rockafellar, R.T. Convergence rates in forward-backward splitting. *SIAM J. Optim.* **1997**, *7*, 421–444 [[CrossRef](#)]
20. Cholamjiak, W.; Cholamjiak, P.; Suantai, S. An inertial forward–backward splitting method for solving inclusion problems in Hilbert spaces. *J. Fixed Point Theory Appl.* **2018**, *20*, 42. [[CrossRef](#)]
21. Dong, Q.; Jiang, D.; Cholamjiak, P.; Shehu, Y. A strong convergence result involving an inertial forward–backward algorithm for monotone inclusions. *J. Fixed Point Theory Appl.* **2017**, *19*, 3097–3118. [[CrossRef](#)]
22. Combettes, P.L.; Pesquet, J.C. Proximal Splitting Methods in Signal processing. In *Fixed-Point Algorithms for Inverse Problems in Science and Engineering*; Springer: New York, NY, USA, 2011; pp. 185–212.
23. Cholamjiak, P. A generalized forward-backward splitting method for solving quasi inclusion problems in Banach spaces. *Numer. Algorithms* **2016**, *71*, 915–932. [[CrossRef](#)]
24. Tseng, P. A modified forward–backward splitting method for maximal monotone mappings. *SIAM J. Control. Optim.* **2000**, *38*, 431–446. [[CrossRef](#)]
25. Cholamjiak, P.; Thong, D.V.; Cho, Y.J. A novel inertial projection and contraction method for solving pseudomonotone variational inequality problems. *Acta Appl. Math.* **2019**, 1–29. [[CrossRef](#)]
26. Gibali, A.; Thong, D.V. Tseng type methods for solving inclusion problems and its applications. *Calcolo* **2018**, *55*, 49. [[CrossRef](#)]
27. Thong, D.V.; Cholamjiak, P. Strong convergence of a forward–backward splitting method with a new step size for solving monotone inclusions. *Comput. Appl. Math.* **2019**, *38*, 94. [[CrossRef](#)]
28. Suparatulatorn, R.; Khemphet, A. Tseng type methods for inclusion and fixed point problems with applications. *Mathematics* **2019**, *7*, 1175. [[CrossRef](#)]
29. Bauschke, H.H.; Combettes, P.L. *Convex Analysis and Monotone Operator Theory in Hilbert Spaces*; Springer: Berlin/Heidelberg, Germany, 2011; Volume 408.



© 2020 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).