mathematics

MDPI

Article

Parallel Hybrid Algorithms for a Finite Family of *G*-Nonexpansive Mappings and Its Application in a Novel Signal Recovery

Suthep Suantai [1,2], Kunrada Kankam [3], Watcharaporn Cholamjiak [3,*] and Watcharaporn Yajai [3]

1 Research Group in Mathematics and Applied Mathematics, Faculty of Science, Chiang Mai University, Chiang Mai 50200, Thailand; suthep.s@cmu.ac.th
2 Data Science Research Center, Department of Mathematics, Faculty of Science, Chiang Mai University, Chiang Mai 50200, Thailand
3 School of Science, University of Phayao, Phayao 56000, Thailand; kunradazzz@gmail.com (K.K.); watcharaporn.yajai@gmail.com (W.Y.)
* Correspondence: c-wchp007@hotmail.com

Abstract: This article considers a parallel monotone hybrid algorithm for a finite family of G-nonexpansive mapping in Hilbert spaces endowed with graphs and suggests iterative schemes for finding a common fixed point by the two different hybrid projection methods. Moreover, we show the computational performance of our algorithm in comparison to some methods. Strong convergence theorems are proved under suitable conditions. Finally, we give some numerical experiments of our algorithms to show the efficiency and implementation of the LASSO problems in signal recovery with different types of blurred matrices and noise.

Keywords: hybrid projection method; parallel hybrid method; G-nonexpansive mapping; signal recovery

MSC: 47H10; 65K05; 94A12

Citation: Suantai, S.; Kankam, K.; Cholamjiak, W.; Yajai, W. Parallel Hybrid Algorithms for a Finite Family of G-Nonexpansive Mappings and Its Application in a Novel Signal Recovery. *Mathematics* **2022**, *10*, 2140. https://doi.org/10.3390/math10122140

Academic Editor: Christopher Goodrich

Received: 26 May 2022
Accepted: 16 June 2022
Published: 20 June 2022

1. Introduction

Let C be a nonempty subset of a real Banach space X. Let Δ denotes the diagonal of the cartesian product $C \times C$, i.e., $\Delta = \{(s,s) : s \in C\}$. Assume that G is a directed graph such that $V(G)$ is the set of its vertices that coincides with C, and $E(G)$ is the set of its edges. We assume that G has no parallel edge and $\Delta \subseteq E(G)$. A graph of G is defined by $(V(G), E(G))$. A mapping $\Theta : C \to C$ is said to be G-nonexpansive if Θ satisfies the conditions: Θ preserves edges of G, i.e.,

$$(s,t) \in E(G) \Rightarrow (\Theta s, \Theta t) \in E(G),\ \forall (s,t) \in E(G);$$

and Θ non-indecreases the weights of edges of G in the following way:

$$(s,t) \in E(G) \Rightarrow \|\Theta s - \Theta t\| \leq \|s - t\|,\ \forall (s,t) \in E(G).$$

It's easy to see that G-nonexpansive mapping generalizes nonexpansive mapping. Many problems in mathematical sciences have been solved by finding a fixed-point approximation of a nonexpansive mapping in many metric spaces. Iterative sequences have been proposed for finding fixed points and their applications by many mathematicians, see [1–3]. One of the most famous is the S-iteration method introduced by Agarwal et al. [4] for some operators in norm linear spaces. Recently, Suparatulatorn et al. [2] used the S-iteration method for finding a fixed point of three different G-nonexpansive mappings Θ_1, Θ_2, Θ_3 with directed graphs. The weak convergence was proved under some conditions on the parameters in Hilbert spaces endowed with graphs. This modified S-iteration method is defined as Algorithm 1:

Algorithm 1: Choose $s_1 \in C$, and $\{\alpha_n\}$, $\{\beta_n\}$ are real sequences in $[0,1]$.

(STEP 1) Compute
$$u_n = (1-\beta_n)s_n + \beta_n \Theta_1 s_n.$$

(STEP 2) Construct s_{n+1} by
$$s_{n+1} = (1-\alpha_n)\Theta_1 s_n + \alpha_n \Theta_2 u_n, \ \forall n \geq 1.$$

(STEP 3) Set $n := n+1$, and go to (Step 1).

In 2017, Sridarat et al. [5] studied the convergence analysis of SP-iteration in Hilbert spaces endowed with graphs. A weak convergence theorem was proved. The following iteration in Algorithm 2 is known as SP-iteration:

Algorithm 2: Choose $s_1 \in C$, and $\{\alpha_n\}$, $\{\beta_n\}$, $\{\gamma_n\}$ are real sequences in $[0,1]$.

(STEP 1) Compute
$$u_n = (1-\gamma_n)s_n + \gamma_n \Theta_3 s_n.$$

(STEP 2) Compute
$$t_n = (1-\beta_n)u_n + \beta_n \Theta_2 u_n.$$

(STEP 3) Construct s_{n+1} by
$$s_{n+1} = (1-\alpha_n)t_n + \alpha_n \Theta_1 t_n, \ \forall n \geq 1.$$

(STEP 4) Set $n := n+1$, and go to (Step 1).

In studying fixed-point algorithms, the rate of convergence is very important to show the efficiency of an algorithm. Recently, Yambangwai et al. [6] introduced a new modified three-step iteration method for three different G-nonexpansive mappings in Banach spaces with a graph and showed a better rate of convergence compared with Sridarat et al. [5]. The algorithm also gets a weak convergence theorem under some suitable conditions. This algorithm is defined as Algorithm 3:

Algorithm 3: Choose $s_1 \in C$, and $\{\alpha_n\}$, $\{\beta_n\}$, $\{\gamma_n\}$ are real sequences in $[0,1]$.

(STEP 1) Compute
$$u_n = (1-\gamma_n)s_n + \gamma_n \Theta_3 s_n.$$

(STEP 2) Compute
$$t_n = (1-\beta_n)u_n + \beta_n \Theta_2 u_n.$$

(STEP 3) Construct s_{n+1} by
$$s_{n+1} = (1-\alpha_n)\Theta_2 u_n + \alpha_n \Theta_1 t_n, \ \forall n \geq 1.$$

(STEP 4) Set $n := n+1$, and go to (Step 1).

For obtaining strong convergence theorems, Nakajo and Takahashi [7] proposed the following hybrid projection algorithm which is well-known as the CQ projection algorithm for finding a fixed point of a nonexpansive mapping in a real Hilbert space H. For each $t \in H$, $\operatorname{argmax}_{s \in H} \|s-t\| = \{x \in H : \|s-t\| \leq \|x-t\| \text{ for all } s \in H\}$ They investigated the sequence $\{s_n\}$ generated by Algorithm 4 as follows:

Algorithm 4: Choose $s_1 \in C$, $Q_1 = C = C_1$ and $\{\alpha_n\} \subset [0, a]$ for some $a \in [0, 1)$.

(STEP 1) Set
$$t_n = \alpha_n s_n + (1 - \alpha_n)\Theta_n s_n.$$
(STEP 2) Compute
$$\bar{t}_n = \operatorname{argmax}\{\|t_n^i - s_n\| : i = 1, 2, \ldots, N\}.$$
(STEP 3) Compute
$$C_n = \{z \in C : \|\bar{t}_n - z\| \leq \|s_n - z\|\}$$
and
$$Q_n = \{z \in C : \langle z - s_n, s_1 - s_n \rangle \leq 0\}.$$
(STEP 4) Construct s_{n+1} by
$$s_{n+1} = P_{C_n \cap Q_n}(s_1), \ \forall n \geq 1.$$
(STEP 5) Set $n := n + 1$, and go to (Step 1).

In 2005, Anh and Hieu [8,9] introduced a projection method which is called the parallel monotone hybrid algorithm for solving common fixed point problems of a finite family of quasi ϕ-nonexpansive mappings $\{\Theta_i\}_{i=1}^N$ in a Banach space. This algorithm is presented in a real Hilbert space as Algorithm 5:

Algorithm 5: Choose $s_1 \in C$, $C_1 = C$ and $\{\alpha_n\}$ is a sequence in $[0, 1]$.

(STEP 1) Set
$$t_n^i = \alpha_n s_n + (1 - \alpha_n)\Theta_i s_n, i = 1, 2, \ldots, N.$$
(STEP 2) Compute
$$\bar{t}_n = \operatorname{argmax}\{\|t_n^i - s_n\| : i = 1, 2, \ldots, N\}.$$
(STEP 3) Compute
$$C_{n+1} = \{v \in C_n : \|v - \bar{t}_n\| \leq \|v - s_n\|\}.$$
(STEP 4) Construct s_{n+1} by
$$s_{n+1} = P_{C_{n+1}}(s_1), n \geq 1.$$
(STEP 5) Set $n := n + 1$, and go to (Step 1).

A strong convergence theorem has been proved under a condition on the parameter α_n with $\limsup_{n \to \infty} \alpha_n < 1$. Recently, there have been some works involving the parallel method for solving the fixed point problem (see [2,10,11]).

In this work, we wish to study the parallel monotone hybrid algorithm for a finite family of G-nonexpansive mappings in Hilbert spaces and introduce algorithms, based on the hybrid projection method. We then prove the strong convergence theorems of the proposed methods using the parallel monotone hybrid algorithm. Finally, some numerical experiments in signal recovery are provided to show the efficiency and implementation of our algorithms.

2. Main Results

In this section, we prove strong convergence theorems for a finite family of G-nonexpansive mappings and propose a projection Algorithm 6 for finding a common fixed point.

Assume that six below conditions hold.

(1) The mappings $\Theta_i : C \to C$ are G-nonexpansive for all $i = 1, 2, \ldots, N$.
(2) The solution set $F := \bigcap_{i=1}^{N} F(\Theta_i) \neq \emptyset$.
(3) F is closed and $F(\Theta_i) \times F(\Theta_i) \subseteq E(G)$ for all $i = 1, 2, \ldots, N$.
(4) $\{s_n\}$ dominates p for all $p \in F$ and if there exists a subsequence $\{s_{n_k}\}$ of $\{s_n\}$ such that $s_{n_k} \rightharpoonup w \in C$, then $(s_{n_k}, w) \in E(G)$.
(5) The sequence $\{\alpha_n^i\} \subset [0,1]$ and $\liminf_{n\to\infty} \alpha_n^i(1-\alpha_n^i) > 0$ for all $i = 1, 2, \ldots, N$.
(6) $\liminf_{n\to\infty} \alpha_n^0 \alpha_n^i > 0$ for all $i = 1, 2, \ldots, N$.

Algorithm 6: Choose $s_1 \in C$, $Q_1 = C$ and $\{\alpha_n^i\}$ is a sequence in $[0,1]$.

(STEP 1) Set
$$t_n^i = (1-\alpha_n^i)s_n + \alpha_n^i \Theta_i s_n, i = 1, 2, \ldots, N.$$

(STEP 2) Compute
$$\bar{t}_n = \operatorname{argmax}\{\|t_n^i - s_n\| : i = 1, 2, \ldots, N\}.$$

(STEP 3) Compute
$$C_n = \{v \in C : \|\bar{t}_n - v\|^2 \leq \|s_n - v\|^2\}$$

and
$$Q_n = \{v \in Q_{n-1} : \langle s_1 - s_n, s_n - v\rangle \geq 0\}.$$

(STEP 4) Construct s_{n+1} by
$$s_{n+1} = P_{C_n \cap Q_n}(s_1).$$

(STEP 5) Set $n := n+1$, and go to (Step 1).

Theorem 1. *Let C be a nonempty closed and convex subset of a real Hilbert space H. Let $G = (V(G), E(G))$ be a directed graph such that $V(G) = C$ and $E(G)$ be convex. Assume that $\{s_n\}$ is generated by Algorithm 6. Then under conditions (1)–(5), $\{s_n\}$ strongly converges to $w = P_F(s_1)$.*

Proof. We divide the proof in five steps.

Claim 1: $P_{C_n \cap Q_n}$ is well-defined for every $s_1 \in H$. By Theorem 3.2 of Tiammee et al. [12], we obtain that F is closed and convex, and $F(\Theta_i)$ is convex for all $i = 1, 2, \ldots, N$. From definitions of C_n and Q_n, and from Lemma 2.2 in [13] that $C_n \cap Q_n$ is closed and convex. Let $p \in F$. Since $\{s_n\}$ dominates p and Θ_i is edge-preserving, we have $(\Theta_i s_n, p) \in E(G)$ for all $i = 1, 2, \ldots, N$. This implies that $(t_n^i, p) = ((1-\alpha_n^i)s_n + \alpha_n^i \Theta_i s_n, p) \in E(G)$ ans as $E(G)$ is convex, we have

$$\begin{aligned} \|t_n^i - p\| &= \|(1-\alpha_n^i)s_n + \alpha_n^i \Theta_i s_n - p\| \\ &\leq (1-\alpha_n^i)\|s_n - p\| + \alpha_n^i \|\Theta_i s_n - p\| \\ &\leq \|s_n - p\|. \end{aligned} \tag{1}$$

This implies that $\|\bar{t}_n - p\| \leq \|s_n - p\|$. Thus, we have $p \in C_n$. Therefore $F \subset C_n \cap Q_n$. This implies that $P_{C_n \cap Q_n}(s_1)$ is well-defined.

Claim 2: $\lim_{n\to\infty} \|s_n - s_1\|$ exists. By the definition of the metric projection P_F, and since F is a nonempty, closed and convex subset of H, we know that there exists a unique $v \in F$ such that $v = P_F(s_1)$. From $s_{n+1} = P_{C_n \cap Q_n}(s_1)$, we get

$$\|s_{n+1} - s_1\| \leq \|z - s_1\|, \quad \text{for all } z \in C_n \cap Q_n \text{ and } n \in \mathbb{N}. \tag{2}$$

On the other hand, as $F \subset C_n \cap Q_n$, we obtain

$$\|s_{n+1} - s_1\| \leq \|v - s_1\|, \quad \text{for all } n \in \mathbb{N}. \tag{3}$$

Thus, $\{s_n\}$ is bounded. Since $s_n = P_{Q_n}(s_1)$ and $s_{n+1} \in Q_n$, we get

$$\|s_n - s_1\| \leq \|s_{n+1} - s_1\|, \quad \text{for all } n \in \mathbb{N}.$$

It follows that the sequence $\{\|s_n - s_1\|\}$ is bounded and non-decreasing. Therefore $\lim_{n\to\infty} \|s_n - s_1\|$ exists.

Claim 3: $\lim_{n\to\infty} s_n = w \in C$. For $m > n$, by the definition of C_n, since $s_m = P_{C_m}(s_1) \in C_m \subset C_n$, it follows from the property of the metric projection P_{C_m} that

$$\|s_m - s_n\|^2 \leq \|s_m - s_1\|^2 - \|s_n - s_1\|^2.$$

Since $\|s_n - s_1\|$ exists, therefore by Step 2, $s_m \to s_n$ as $n \to \infty$. From the completeness of a Hilbert space, $\{s_n\}$ is a Cauchy sequence. Hence, there exists $w \in C$ such that $s_n \to w$ as $n \to \infty$. In particular, we get

$$\lim_{n\to\infty} \|s_{n+1} - s_n\| = 0. \tag{4}$$

Claim 4: $w \in F$. Since $s_{n+1} \in C_n$, therefore from (4), we have

$$\|\bar{t}_n - s_{n+1}\| \leq \|s_n - s_{n+1}\| \to 0 \text{ as } n \to \infty.$$

Now, we obtain

$$\|\bar{t}_n - s_n\| \leq \|\bar{t}_n - s_{n+1}\| + \|s_{n+1} - s_n\|.$$

That is

$$\lim_{n\to\infty} \|\bar{t}_n - s_n\| = 0. \tag{5}$$

We know that $\|t_n^i - s_n\| \leq \|\bar{t}_n - s_n\|$. By (5), we obtain

$$\lim_{n\to\infty} \|t_n^i - s_n\| = 0, \tag{6}$$

for all $i = 1, 2, \ldots, N$. It follows from the properties in a real Hilbert space, that

$$\begin{aligned}
\|t_n^i - p\|^2 &= \|(1 - \alpha_n^i)s_n + \alpha_n^i \Theta_i s_n - p\|^2 \\
&= (1 - \alpha_n^i)\|s_n - p\|^2 + \alpha_n^i \|\Theta_i s_n - p\|^2 - \alpha_n^i(1 - \alpha_n^i)\|\Theta_i s_n - s_n\|^2 \\
&\leq (1 - \alpha_n^i)\|s_n - p\|^2 + \alpha_n^i \|s_n - p\|^2 - \alpha_n^i(1 - \alpha_n^i)\|\Theta_i s_n - s_n\|^2 \\
&= \|s_n - p\|^2 - \alpha_n^i(1 - \alpha_n^i)\|\Theta_i s_n - s_n\|^2.
\end{aligned} \tag{7}$$

By (7), we obtain

$$\alpha_n^i(1 - \alpha_n^i)\|\Theta_i s_n - s_n\|^2 \leq \|s_n - p\|^2 - \|t_n^i - p\|^2. \tag{8}$$

By our Assumption (5) and (8), we obtain

$$\lim_{n\to\infty} \|\Theta_i s_n - s_n\| = 0.$$

From $s_n \to w$ as $n \to \infty$, the assumption (1) and Lemma 6 in [2], we have $w \in F$.
Claim 5: $w = P_F(s_1)$. Since $s_{n+1} = P_{C_n \cap Q_n}(s_1)$ and $w \in F \subset C_n \cap Q_n$, we have

$$\langle s_1 - s_n, s_n - p \rangle \geq 0, \ \forall p \in C_n \cap Q_n. \tag{9}$$

By taking the limit in (9), we obtain

$$\langle s_1 - w, w - p \rangle \geq 0, \ \forall p \in C_n \cap Q_n.$$

Since $F \subset C_n \cap Q_n$, so we have $\langle s_1 - w, w - p \rangle \geq 0$, for each $p \in F$, which gives $w = P_F(s_1)$. This completes the proof. □

We know that if Θ is G-nonexpansive, that Θ is nonexpansive. From direct consequences of Theorem 1, we have the following corollary.

Corollary 1. *Assume that $\{s_n\}$ is a sequence generated by Algorithm 6. Let C be a nonempty closed and convex subset of a real Hilbert space H, and let $\Theta_i : C \to C$ be a nonexpansive mapping for all $i = 1, 2, \ldots, N$ such that $F := \bigcap_{i=1}^{N} F(\Theta_i) \neq \emptyset$. Then under conditions (3)–(5), $\{s_n\}$ strongly convergence to $w = P_F(s_1)$.*

Next, we propose the following Algorithm 7:

Algorithm 7: Choose $s_1 \in C$, $Q_1 = C$ and $\{\alpha_n^i\}$ is a sequence in $[0,1]$ for all $i = 0, 1, \ldots, N$ such that $\sum_{i=0}^{N} \alpha_n^i = 1$ for all $n \geq 1$.

(STEP 1) Set

$$t_n = \alpha_n^0 s_n + \sum_{i=1}^{N} \alpha_n^i \Theta_i s_n,$$

(STEP 2) Compute

$$C_n = \{v \in C_n : \|v - t_n\|^2 \leq \|v - s_n\|^2\}$$

and

$$Q_n = \{v \in Q_{n-1} : \langle s_1 - s_n, s_n - v \rangle \geq 0\}.$$

(STEP 3) Compute

$$s_{n+1} = P_{C_n \cap Q_n}(s_1).$$

(STEP 4) Set $n := n + 1$, and go to (Step 1).

Theorem 2. *Let C be a nonempty closed and convex subset of a real Hilbert space H. Let $G = (V(G), E(G))$ be a directed graph such that $V(G) = C$ and $E(G)$ be convex. Assume that $\{s_n\}$ is generated by Algorithm 7. Then under conditions (1)–(4) and (6), $\{s_n\}$ strongly converges to $w = P_F(s_1)$.*

Proof. We shell show that $P_{C_n \cap Q_n}$ is well-defined and $F \subseteq C_n \cap Q_n, \forall n \geq 0$. Similar to Step 1 in Theorem 1, we can show that $C_n \cap Q_n$ is closed and convex, $\forall n \geq 0$. Also, we can show that

$$\begin{aligned} \|t_n - p\| &= \|\alpha_n^0 s_n + \sum_{i=1}^{N} \alpha_n^i \Theta_i s_n - p\| \\ &\leq \alpha_n^0 \|s_n - p\| + \sum_{i=1}^{N} \alpha_n^i \|\Theta_i s_n - p\| \\ &\leq \|s_n - p\|. \end{aligned} \tag{10}$$

Thus, we have $p \in C_n$. Therefore $F \subset C_n \cap Q_n$. This implies that $P_{C_n \cap Q_n}(s_1)$ is well-defined. By the same proof of Step 2–3 in Theorem 1, we obtain $\lim_{n\to\infty} \|s_n - s_1\|$ exists. Hence, there exists $w \in C$ such that $s_n \to w$ as $n \to \infty$. In particular, we have

$$\lim_{n\to\infty} \|s_{n+1} - s_n\| = 0. \tag{11}$$

Next, we show that $w \in F$. By $s_{n+1} \in C_n$, it follows from (11) that

$$\|t_n - s_n\| \le \|t_n - s_{n+1}\| + \|s_{n+1} - s_n\| \le 2\|s_{n+1} - s_n\| \to 0 \tag{12}$$

as $n \to \infty$. For $p \in \Omega$, it follows from Lemma 2.1 in [14] and since $\{s_n\}$ dominates p that

$$\begin{aligned}
\|t_n - p\|^2 &= \|\alpha_n^0 s_n + \sum_{i=1}^{N} \alpha_n^i \Theta_i s_n - p\|^2 \\
&\le \alpha_n^0 \|s_n - p\|^2 + \sum_{i=1}^{N} \alpha_n^i \|\Theta_i s_n - p\|^2 - \sum_{i=1}^{N} \alpha_n^0 \alpha_n^i \|\Theta_i s_n - s_n\|^2 \\
&\le \|s_n - p\|^2 - \sum_{i=1}^{N} \alpha_n^0 \alpha_n^i \|\Theta_i s_n - s_n\|^2.
\end{aligned}$$

This implies that

$$\sum_{i=1}^{N} \alpha_n^0 \alpha_n^i \|\Theta_i s_n - s_n\|^2 \le \|s_n - p\|^2 - \|t_n - p\|^2.$$

By our assumption (3) and (12), we obtain

$$\lim_{n\to\infty} \|\Theta_i s_n - s_n\| = 0$$

for all $i = 1, 2, \dots, N$. From the fact that $s_n \to w$ as $n \to \infty$, the assumption (1) and Lemma 6 in [2], we have $w \in F$. By the same proof of Step 5 in Theorem 1, we obtain $w \in P_F(s_1)$. This completes the proof. □

Corollary 2. *Assume that $\{s_n\}$ is a sequence generated by Algorithm 7. Let C be a nonempty closed and convex subset of a real Hilbert space H. Let $\Theta_i : C \to C$ be a nonexpansive mapping for all $i = 1, 2, \dots, N$ such that $F := \bigcap_{i=1}^{N} F(\Theta_i) \neq \emptyset$. Then under conditions (3), (4) and (6), $\{s_n\}$ convergence strongly to $w = P_F(s_1)$.*

3. Numerical Experiments

In this section, we give numerical results to support our main theorem. We now give an example in a Euclidean space $\mathbb{R}^3$ with a numerical experiment to support our main results.

Example 1. *Let $H = \mathbb{R}^3$ and $C = [0, \infty) \times [-10, 5] \times [0, \infty)$. Assume that $(s, t) \in E(G)$ if and only if $1 \le s_1, t_1$, $-9 \le s_2, t_2 \le 1.5$ and $0 \le s_3, t_3 \le 1.25$ or $s = t$ for all $s = (s_1, s_2, s_3)$, $t = (t_1, t_2, t_3) \in C$. Define mappings $\Theta_1, \Theta_2, \Theta_3 : C \to C$ by*

$$\begin{aligned}
\Theta_1 s &= (\log \frac{s_1}{2} + 2, \frac{\arctan s_2}{4}, 1); \\
\Theta_2 s &= (2, 0, \frac{\tan(s_3 - 1)}{4} + 1); \\
\Theta_3 s &= (2, \frac{e^{s_2} - 1}{2}, 1)
\end{aligned}$$

for all $s = (s_1, s_2, s_3) \in C$. It is easy to check that Θ_1 and Θ_2 are G-nonexpansive such that $F(\Theta_1) \cap F(\Theta_2) = \{(2, 0, 1)\}$. On the other hand, Θ_1 is not nonexpansive since for $s = (0.31, 1, 7)$ and $t = (0.22, 1, 7)$, this implies that $\|\Theta_1 s - \Theta_1 t\| > 0.1 > \|s - t\|$. Θ_2 is not nonexpansive since for $s = (5, -0.5, 2.11)$ and $t = (5, -0.5, 2.28)$, we have $\|\Theta_2 s - \Theta_2 t\| > 0.3 > \|s - t\|$. Moreover, Θ_3 is not nonexpansive since for $s = (1, 1.19, 0.2)$ and $t = (1, 1.02, 0.2)$, we have $\|\Theta_3 s - \Theta_3 t\| > 0.2 > \|s - t\|$. In this section, CPU and Iter are denoted by the time of CPU and the number of iterations, respectively. All numerical experiments presented were obtained from MATLAB R2019b running on the same laptop computer. In our experiment, we give three cases as follows:

Case 1: Algorithm 6 with $\alpha_n = 0.1$ of the initial point (1.01, −0.02, 1.26) .
Case 2: Algorithm 6 with $\alpha_n = 0.5$ of the initial point (1.01, −0.02, 1.26).
Case 3: Algorithm 6 with $\alpha_n = 0.9$ of the initial point (1.01, −0.02, 1.26).
The numerical results are reported as follows:

From Table 1 and Figure 1, we see that in the case of two or more inputting $\Theta_i (i \geq 2)$ of the proposed parallel monotone hybrid Algorithm 6 achieves fewer iterations than the inputted one. In the case of three inputting, a little more CPU time is required than in some cases of one or two inputting.

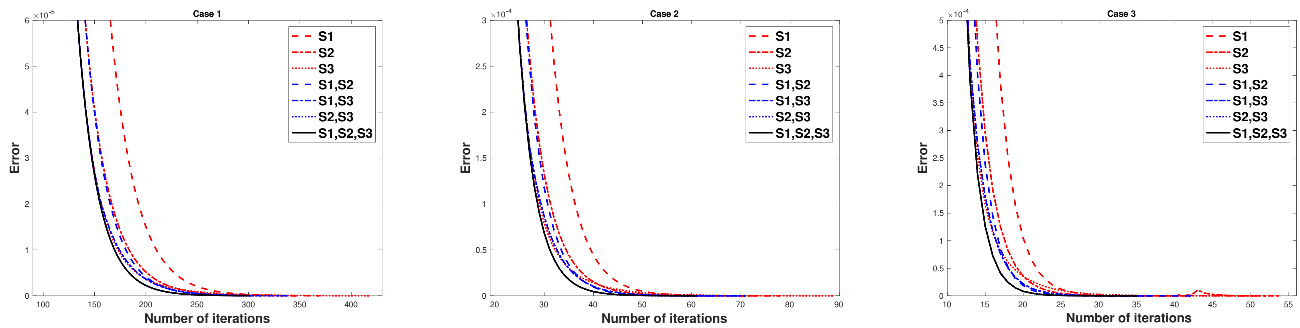

Figure 1. Graph of number of the iterations versus error.

Table 1. The convergence behavior of inputting Θ_i, $i = 1, 2, 3$, stop condition (Cauchy error) $< 10^{-9}$.

		Θ_1	Θ_2	Θ_3	Θ_1, Θ_2	Θ_1, Θ_3	Θ_2, Θ_3	$\Theta_1, \Theta_2, \Theta_3$
Case 1	Iter	369	348	418	313	339	311	301
	CPU	0.0192	0.0130	0.0113	0.0161	0.0192	0.0192	0.0166
Case 2	Iter	78	72	89	65	71	64	61
	CPU	0.0194	0.0122	0.0101	0.0166	0.0166	0.0164	0.0209
Case 3	Iter	52	54	50	43	37	36	35
	CPU	0.0223	0.0121	0.0109	0.0127	0.0181	0.0191	0.0199

Next, we give an experiment of Algorithm 7. We give 9 cases for parameter α as follows:
Case 1: $\alpha_0 = 0.1$, $\alpha_1 = \frac{0.9}{3}$, $\alpha_2 = \frac{0.9}{3}$ and $\alpha_3 = \frac{0.9}{3}$.
Case 2: $\alpha_0 = 0.5$, $\alpha_1 = \frac{0.5}{3}$, $\alpha_2 = \frac{0.5}{3}$ and $\alpha_3 = \frac{0.5}{3}$.
Case 3: $\alpha_0 = 0.9$, $\alpha_1 = \frac{0.1}{3}$, $\alpha_2 = \frac{0.1}{3}$ and $\alpha_3 = \frac{0.1}{3}$.
Case 4: $\alpha_0 = 0.1$, $\alpha_1 = \frac{0.9}{6}$, $\alpha_2 = 2(\frac{0.9}{6})$ and $\alpha_3 = 3(\frac{0.9}{6})$.
Case 5: $\alpha_0 = 0.1$, $\alpha_1 = \frac{0.9}{6}$, $\alpha_2 = 3(\frac{0.9}{6})$ and $\alpha_3 = 2(\frac{0.9}{6})$.
Case 6: $\alpha_0 = 0.1$, $\alpha_1 = 2(\frac{0.9}{6})$, $\alpha_2 = \frac{0.9}{6}$ and $\alpha_3 = 3(\frac{0.9}{6})$.
Case 7: $\alpha_0 = 0.1$, $\alpha_1 = 2(\frac{0.9}{6})$, $\alpha_2 = 3(\frac{0.9}{6})$ and $\alpha_3 = \frac{0.9}{6}$.
Case 8: $\alpha_0 = 0.1$, $\alpha_1 = 3(\frac{0.9}{6})$, $\alpha_2 = \frac{0.9}{6}$ and $\alpha_3 = 2(\frac{0.9}{6})$.
Case 9: $\alpha_0 = 0.1$, $\alpha_1 = 3(\frac{0.9}{6})$, $\alpha_2 = 2(\frac{0.9}{6})$ and $\alpha_3 = \frac{0.9}{6}$.
For all cases, we choose the initial point (1.48, −0.06, 2.72).

From Table 2, we see that the CPU time and the number of iterations of Algorithm 7 decrease when the parameter α_0 *approaches 0 and the rate of* α_n*, by* $3:2:1$ *this has an effect on the CPU time and the number of iterations for input many mappings* Θ_i*.*

Table 2. The convergence behavior of inputting Θ_i, $i = 1,2,3$, stop condition (Cauchy error) $< 10^{-9}$.

		Θ_1	Θ_2	Θ_3	Θ_1,Θ_2	Θ_1,Θ_3	Θ_2,Θ_3	$\Theta_1,\Theta_2,\Theta_3$
Case 1	Iter	43	42	60	37	44	44	37
	CPU	0.01127	0.0081	0.0066	0.0061	0.0097	0.0087	0.0075
Case 2	Iter	73	81	103	68	104	70	68
	CPU	0.0118	0.0063	0.0053	0.0075	0.0099	0.0073	0.0076
Case 3	Iter	357	419	448	335	365	340	324
	CPU	0.0125	0.0085	0.0053	0.0075	0.0099	0.0073	0.0076
Case 4	Iter	43	42	60	35	51	53	40
	CPU	0.0123	0.0061	0.0048	0.0084	0.0109	0.0060	0.0109
Case 5	Iter	43	42	60	35	51	37	36
	CPU	0.0120	0.0062	0.0061	0.0061	0.0086	0.0087	0.0102
Case 6	Iter	43	42	60	40	47	38	41
	CPU	0.0143	0.0089	0.0056	0.0070	0.0082	0.0078	0.0098
Case 7	Iter	43	42	60	35	47	37	49
	CPU	0.0122	0.0067	0.0051	0.0053	0.0105	0.0085	0.0094
Case 8	Iter	43	42	60	40	43	38	41
	CPU	0.0124	0.0071	0.0056	0.0062	0.0079	0.0088	0.0103
Case 9	Iter	43	42	60	35	43	53	34
	CPU	0.00126	0.0057	0.0048	0.0065	0.0085	0.0075	0.0142

Next, we present some numerical examples of signal recovery. We provide a comparison between Algorithms 2, 3, 6 and 7. In this case, we set $\Theta(s_n) = \mathrm{prox}_{\lambda g}(s_n - \lambda \nabla f(s_n))$. It is known that Θ is a nonexpansive mapping when $\lambda \in (0, 2/L)$ and L is the Lipschitz constant of ∇f. Compressed sensing can be modeled as the following underdeterminated linear equation system:

$$t = As + \epsilon, \tag{13}$$

where $s \in \mathbb{R}^N$ is a original signal vector, $t \in \mathbb{R}^M$ is the observed signal which disturbed by filer operator $A : \mathbb{R}^N \to \mathbb{R}^M (M < N)$ and noisy ϵ. It is known that the solution of (13) can be seen as solving the LASSO problem:

$$\min_{s \in \mathbb{R}^N} \frac{1}{2}\|t - As\|_2^2 + \lambda\|s\|_1, \tag{14}$$

where $\lambda > 0$. So we can apply our method for solving (14) in the case that $f(s) = \frac{1}{2}\|t - As\|_2^2$ and $g(s) = \lambda\|s\|_1$. It is noted that $\nabla f(s) = A^T(As - t)$.

The goal of this paper is to remove noise without knowing the type of it. Thus, we focus on the following problem:

$$\begin{aligned} \min_{s \in \mathbb{R}^N} \frac{1}{2}\|A_1 s - t_1\|_2^2 &\quad + \quad \lambda_1\|s\|_1, \\ \min_{s \in \mathbb{R}^N} \frac{1}{2}\|A_2 s - t_2\|_2^2 &\quad + \quad \lambda_2\|s\|_1, \\ &\quad \vdots \\ \min_{s \in \mathbb{R}^N} \frac{1}{2}\|A_N s - t_N\|_2^2 &\quad + \quad \lambda_N\|s\|_1, \end{aligned} \tag{15}$$

where s is the original signal, A_i is a bounded linear operator and t_i is observed signal with noisy for all $i = 1, 2, \ldots, N$. We can apply the Algorithms 6 and 7 to solve the problem (15) by setting $\Theta_i s = \text{prox}_{\lambda_i g_i}(s_n - \lambda_i \nabla f_i(s_n))$.

In our experiment, the observations t_1, t_2, t_3 are generated by different Gaussian noise white signal-to-noise ratio SNR and normal distribution with zero mean and one invariance matrix $A_1, A_2, A_3 \in \mathbb{R}^{M \times N}$, respectively. The initial point s_1 is picked randomly. We use the mean squared error (MSE) for showing the restoration accuracy. This MSE is defined by

$$\text{MSE} = \frac{1}{N}\|s_n - s^*\|_2^2 < 10^{-5},$$

where s^* is an estimated signal of s.

In what follows, let $\alpha_n^i = 0.5$ for all $i = 1, 2, 3$ and let the step sizes $\lambda_1 = \frac{1}{\|A_1\|^2}$, $\lambda_2 = \frac{1}{\|A_2\|^2}$ and $\lambda_3 = \frac{1}{\|A_3\|^2}$. The numerical results are shown as follows:

The performance of the studied proposed Algorithm 6 with the following original signal (Figure 2) is tested.

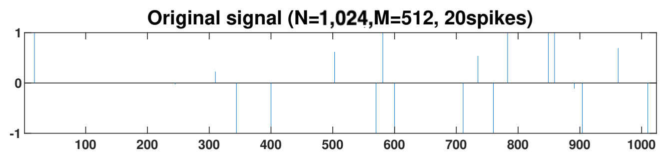

Figure 2. The original signal $N = 1024$, $M = 512$, and 20 spikes.

The different types of blurred matrices A_1, A_2 and A_3 are shown in Figure 3.

Figure 3. Measured values with SNR = 40, 50, 60, respectively.

The results of the Algorithm 6 with $(N = 1)$ by inputting $A_i, i = 1, 2, 3$ for the following three cases:

Case 1.1: Inputting A_1, SNR = 40 on the proposed algorithm;

Case 1.2: Inputting A_2, SNR = 50 on the proposed algorithm;

Case 1.3: Inputting A_3, SNR = 60 on the proposed algorithm;

Are shown in Figure 4 which are composed of the recovered signal.

Next, we present finding the common solutions to signal recovery problem (15) with $(N \geq 2)$ by using the Algorithm 6. So, we can consider the results algorithm in the following three cases:

Case 2.1: Inputting A_1, SNR = 40 and A_2, SNR = 50;

Case 2.2: Inputting A_1, SNR = 40 and A_3, SNR = 60;

Case 2.3: Inputting A_2, SNR = 50 and A_3, SNR = 60;

Case 3.1: Inputting A_1, SNR = 40, A_2, SNR = 50 and A_3, SNR = 60;
Are shown in Figure 5 which are composed of the recovered signal.

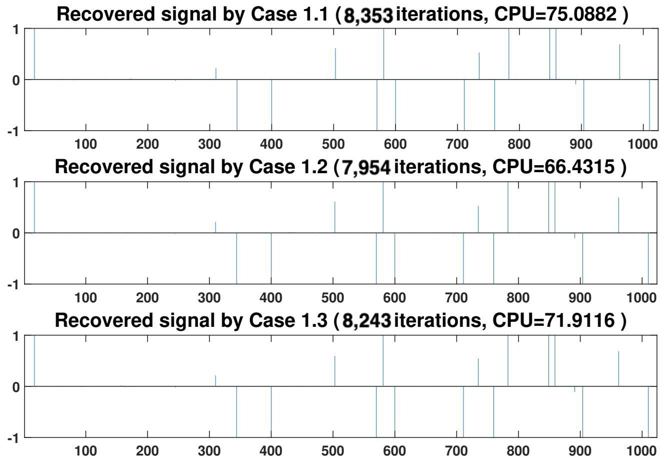

Figure 4. Recovered signal by Case 1.1, Case 1.2 and Case 1.3, respectively.

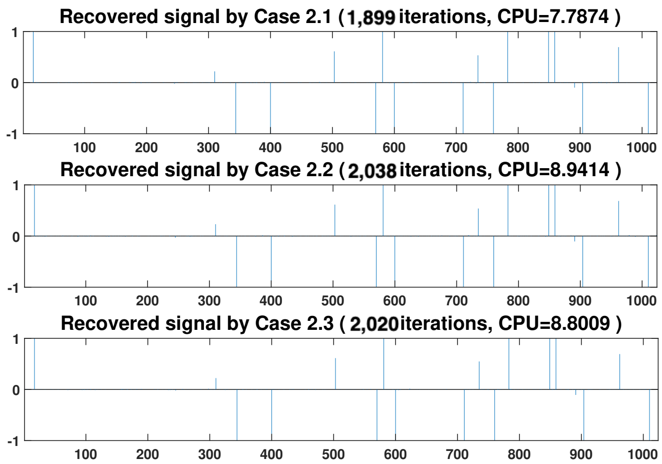

Figure 5. Recovered signal by Case 2.1, Case 2.2 and Case 2.3, respectively.

From Figures 4 and 5, we see that Case 2.1–2.3 has less number of iterations and CPU time than Case 1.1–1.3 in all of the cases.

Finally, we present the common solution the signal recovery problem (15) with $(N = 3)$ of Case 3.1 and a comparison among Algorithms 2 and 3 shown in Figure 6.

From Figures 5 and 6, we see that Case 3.1 has a lower number of iterations and CPU time than Case 2.1–2.3 all of the cases. This means that the efficiency of the proposed Algorithm 6 is better when the number of subproblems is increasing. Moreover, we see that our proposed Algorithm 6 gets less CPU time and fewer iterations than the other two algorithms. The following Figure 7 shows the efficiency of our parallel Algorithm 6 in all cases and a comparison with the other two algorithms by MSE versus the number of iterations.

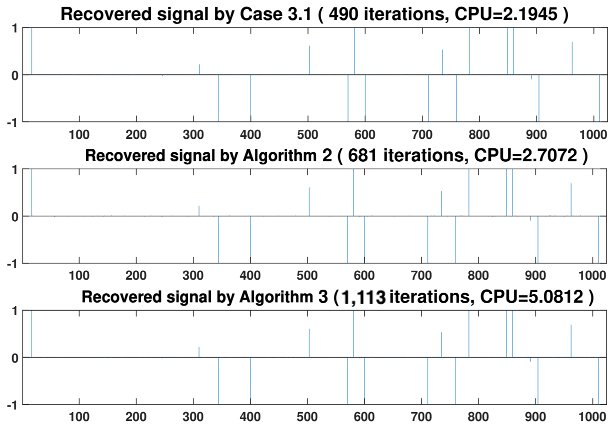

Figure 6. Recovered signal by Algorithms 2, 3 and 6 inputting A_1, A_2 and A_3, respectively.

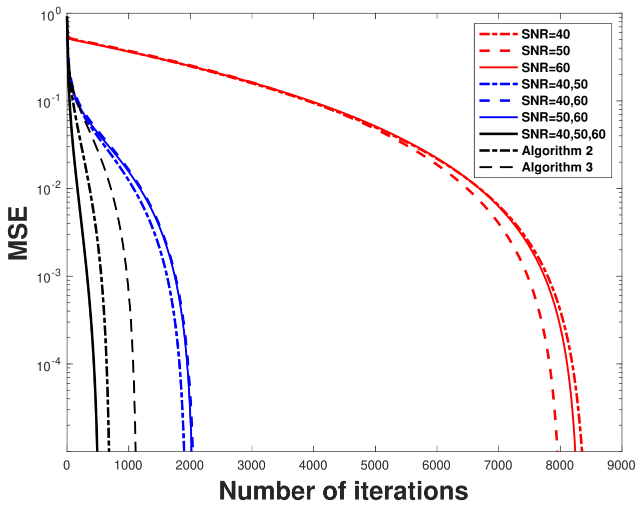

Figure 7. MSE versus number of iterations in case N = 1024 and M = 512.

Next, we analyze the convergence and the effects of $\{\alpha_n\}$ in Algorithm 6 in each case. From Table 3, we observe that the CPU time and the number of iterations of Algorithm 6 small reduction when the parameter $\{\alpha_n\}$ approaches 1. The following Figures 8 and 9 show numerical results for each $\{\alpha_n\}$ in Table 3.

Table 3. The convergence of Algorithm 6 with each $\{\alpha_n\}$.

Given: Random Initial Point, Stop Condition (Cauchy Error) $< 10^{-9}$.					
$\{\alpha_n\}$	**SNR**	**N = 512, M = 256, m = 20**		**N = 1024, M = 512, m = 20**	
		Iter	**CPU**	**Iter**	**CPU**
$\frac{4n^2+12}{20n^2+10}$	40	21,189	206.1261	32,906	1001.1
	50	22,454	232.5960	34,356	1084.8
	40, 50	5529	16.6651	8145	48,778
$\frac{25n^5+15}{50n^5+10}$	40	7360	27.2077	14,285	201.2996
	50	7560	28.3236	13,531	182.0590
	40, 50	2210	3.4878	3610	20.8226
$\frac{81n^9+19}{90n^9+10}$	40	5190	14.0915	8534	77.5054
	50	4316	9.8401	7622	64.8060
	40, 50	1301	1.5145	1660	6.7987

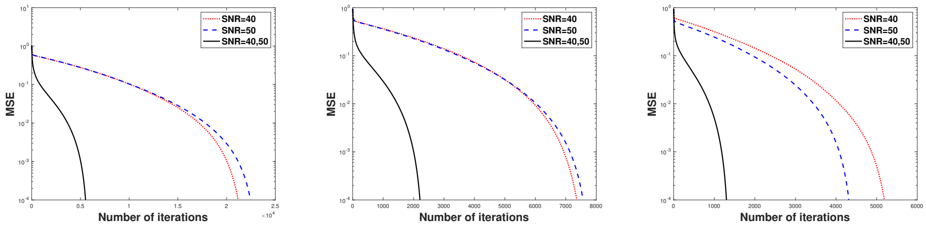

Figure 8. Graph of number of iterations versus Error where $N = 512$ and $M = 265$.

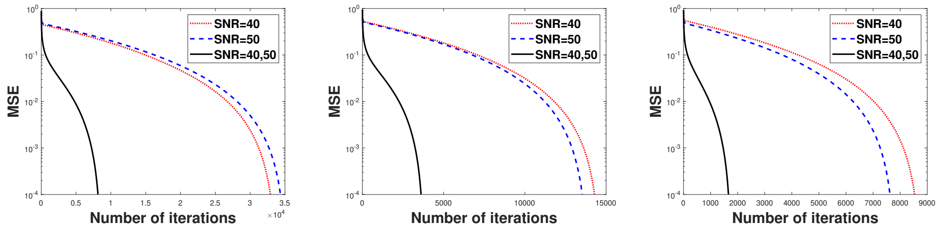

Figure 9. Graph of the number of iterations versus MSE where $N = 1024$ and $M = 512$.

Next, we provide a comparison between Algorithms 2, 3 and 7. For convenience, we set all conditions as in the previous example.

The performance of the studied proposed Algorithm 7 with the following original signal (Figure 10) is tested.

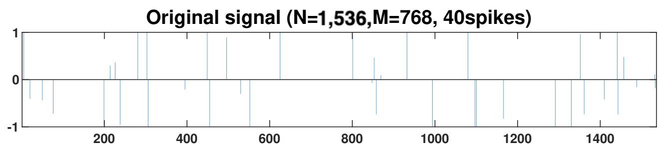

Figure 10. The original signal $N = 1024$, $M = 512$ and 20 spikes.

The different types of blurred matrices A_1, A_2 and A_3 are shown in Figure 11.

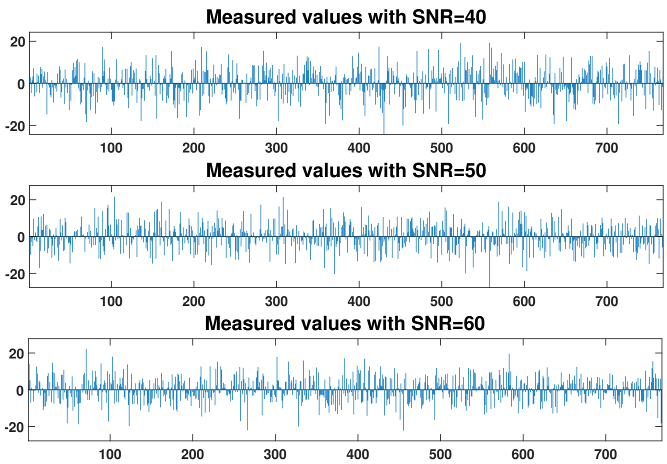

Figure 11. Measured values with SNR = 40, 50, 60, respectively.

The results of the Algorithm 7 with $(N = 1)$ by inputting $A_i, i = 1, 2, 3$ for the following three cases:

Case 1.1: Inputting A_1, SNR = 40 on the proposed algorithm;

Case 1.2: Inputting A_2, SNR = 50 on the proposed algorithm;

Case 1.3: Inputting A_3, SNR = 60 on the proposed algorithm.

Figure 12 which are composed of the recovered signal.

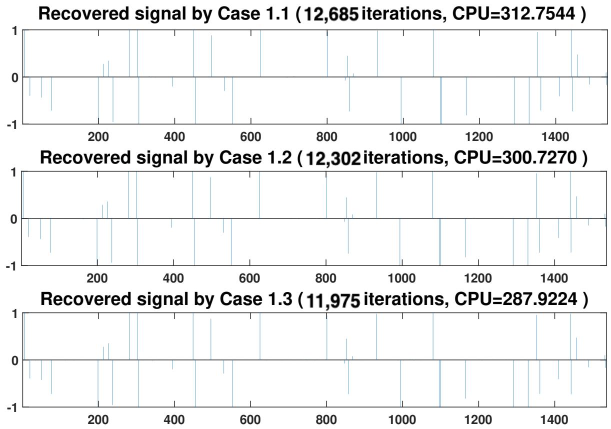

Figure 12. Recovered signal by Case 1.1, Case 1.2 and Case 1.3, respectively.

Next, we present the finding of the common solutions to the signal recovery problem (15) with $(N \geq 2)$ by using the Algorithm 7. So, we can consider the results algorithm in the following three cases:

Case 2.1: Inputting A_1, SNR = 40 and A_2, SNR = 50;
Case 2.2: Inputting A_1, SNR = 40 and A_3, SNR = 60;
Case 2.3: Inputting A_2, SNR = 50 and A_3, SNR = 60;
Case 3.1: Inputting A_1, SNR = 40, A_2, SNR = 50 and A_3, SNR = 60;
Are shown in Figure 13 which are composed of the recovered signal.

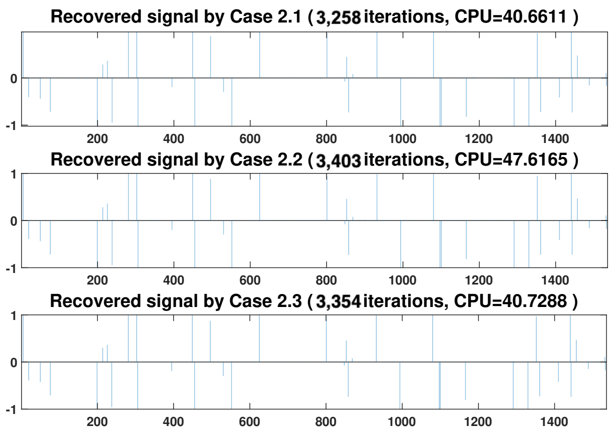

Figure 13. Recovered signal by Case 2.1, Case 2.2 and Case 2.3, respectively.

From Figures 12 and 13, we see that Case 2.1–2.3 has a lower number of iterations and CPU time than Case 1.1–1.3 in all of the cases. Finally, we present the common solution of signal recovery problem (15) with $(N = 3)$ of Case 3.1 and a comparison between Algorithms 2 and 3 which has been shown in Figure 14.

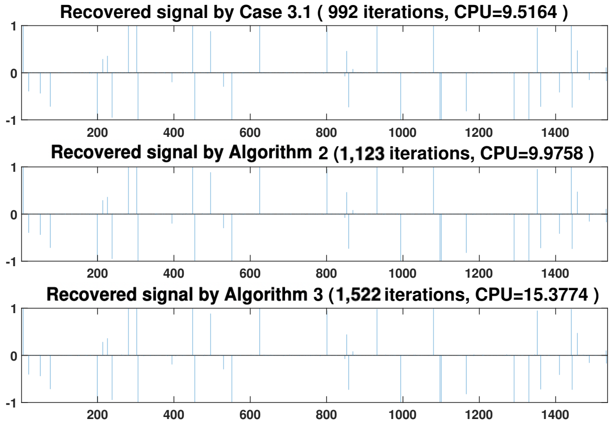

Figure 14. Recovered signal by Algorithms 2, 3 and 7 inputting A_1, A_2 and A_3, respectively.

Figure 15 shows the efficiency of our parallel Algorithm 7 in all cases and a comparison with the other two algorithms by MSE versus the number of iterations.

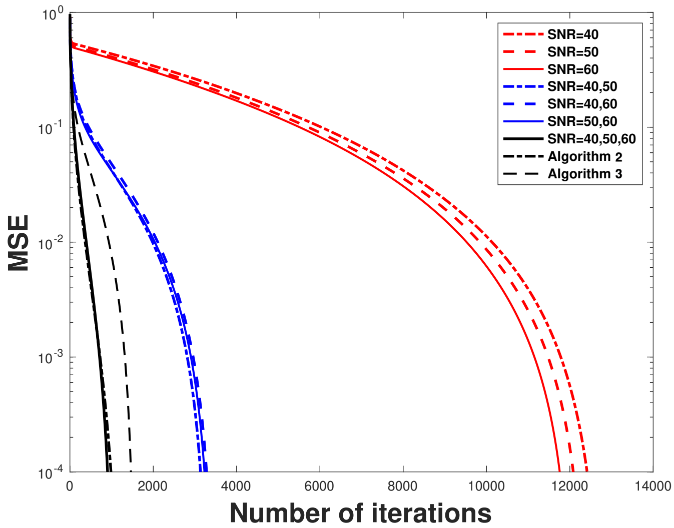

Figure 15. MSE versus number of iterations in case $N = 1024$ and $M = 512$.

From Figures 14 and 15, we see that Case 3.1 has a lower number of iterations and CPU time than Case 2.1–2.3 all of the cases. This means that the efficiency of the proposed Algorithm 7 is also better when the number of subproblems is increasing. Moreover, we see that our proposed Algorithm 7 requires a lower CPU time and number of iterations than the other two algorithms.

4. Conclusions

This paper proposes two parallel hybrid projection algorithms for finding a common fixed point of a finite family of G-nonexpansive mappings in Hilbert spaces with a directed graph. Under some suitable conditions on the update parameters generated in two different algorithms, we obtain strong convergence theorems. We also give examples of numerical experiments to support our main results and compare the rate between the proposed and the existing two methods. It is found that our algorithms have a better convergence behavior than these methods through experiments.

Author Contributions: Funding acquisition and Supervision, S.S.; Writing—review & editing, W.C.; Writing—original draft, W.Y. and Software, K.K. All authors have read and agreed to the published version of the manuscript.

Funding: This research was supported by Chiang Mai University, Thailand.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Acknowledgments: This research was supported by Chiang Mai University, the Thailand Science Research and Innovation Fund, and University of Phayao (Grant No. FF65-RIM072; FF65-UoE002).

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

References

1. Khuangsatung, W.; Kangtunyakarn, A. The Method for Solving Fixed Point Problem of G-Nonexpansive Mapping in Hilbert Spaces Endowed with Graphs and Numerical Example. *Indian J. Pure Appl. Math.* **2020**, *51*, 155–170. [CrossRef]
2. Suparatulatorn, R.; Cholamjiak, W.; Suantai, S. A modified S-iteration process for G-nonexpansive mappings in Banach spaces with graphs. *Numer. Algorithms* **2018**, *77*, 479–490. [CrossRef]
3. Tripak, O. Common fixed points of G-nonexpansive mappings on Banach spaces with a graph. *Fixed Point Theory Appl.* **2016**, *2016*, 1–8. [CrossRef]
4. Agarwal, R.P.; ORegan, D.; Sahu, D.R. Iterative construction of fixed points of nearly asymptotically nonexpansive mappings. *J. Nonlinear Convex Anal.* **2007**, *8*, 61.
5. Sridarat, P.; Suparaturatorn, R.; Suantai, S.; Cho, Y.J. Convergence analysis of SP-iteration for G-nonexpansive mappings with directed graphs. *Bull. Malays. Math. Sci. Soc.* **2019**, *42*, 2361–2380. [CrossRef]
6. Yambangwai, D.; Aunruean, S.; Thianwan, T. A new modified three-step iteration method for G-nonexpansive mappings in Banach spaces with a graph. *Numer. Algorithms* **2020**, *84*, 537–565. [CrossRef]
7. Nakajo, K.; Takahashi, W. Strong convergence theorems for nonexpansive mappings and nonexpansive semigroups. *J. Math. Anal. Appl.* **2003**, *279*, 372–379. [CrossRef]
8. Anh, P.K.; Van Hieu, D. Parallel and sequential hybrid methods for a finite family of asymptotically quasi ϕ-nonexpansive mappings. *J. Appl. Math. Comput.* **2015**, *48*, 241–263. [CrossRef]
9. Anh, P.K.; Van Hieu, D. Parallel hybrid iterative methods for variational inequalities, equilibrium problems, and common fixed point problems. *Vietnam J. Math.* **2016**, *44*, 351–374. [CrossRef]
10. Cholamjiak, P.; Suantai, S.; Sunthrayuth, P. An explicit parallel algorithm for solving variational inclusion problem and fixed point problem in Banach spaces. *Banach J. Math. Anal.* **2020**, *14*, 20–40. [CrossRef]
11. Hieu, D.V. Parallel and cyclic hybrid subgradient extragradient methods for variational inequalities. *Afr. Mat.* **2017**, *28*, 677–692. [CrossRef]
12. Tiammee, J.; Kaewkhao, A.; Suantai, S. On Browder's convergence theorem and Halpern iteration process for G-nonexpansive mappings in Hilbert spaces endowed with graphs. *Fixed Point Theory Appl.* **2015**, *2015*, 1–12. [CrossRef]
13. Martinez-Yanes, C.; Xu, H.K. Strong convergence of the CQ method for fixed point iteration processes. *Nonlinear Anal. Theory Methods Appl.* **2006**, *64*, 2400–2411. [CrossRef]
14. Chidume, C.E.; Ezeora, J.N. Krasnoselskii-type algorithm for family of multi-valued strictly pseudo-contractive mappings. *Fixed Point Theory Appl.* **2014**, *2014*, 1–7. [CrossRef]