Thai Journal of Mathematics
Volume 18 Number 3 (2020)
Pages 1359–1373
http://thaijmath.in.cmu.ac.th

ISSN 1686-0209

Dedicated to Prof. Suthep Suantai on the occasion of his 60^{th} anniversary

A Novel Relaxed Projective Method for Split Feasibility Problems

Teerapon Saelii, Suparat Kesornprom and Prasit Cholamjiak*

School of Science, University of Phayao, Phayao 56000, Thailand
e-mail : teerapon5938@gmail.com (T. Saelii); suparat.ke@gmail.com (S. Kesornprom); prasitch2008@yahoo.com (P. Cholamjiak)

Abstract In this paper, we propose the modified projection algorithms for solving the split feasibility problem in Euclidean a spaces. A new stepsize is introduced by using the self-adaptive technique. Convergence analysis is discussed under suitable conditions. Numerical experiments show that the proposed methods are more efficient than other projective algorithms in comparison.

MSC: 47H09
Keywords: split feasibility problem; projection; self-adaptive technique; convergence theorem

Submission date: 27.03.2020 / Acceptance date: 23.06.2020

1. Introduction

Let C and Q be nonempty closed convex subsets of $\mathbb{R}^N$ and $\mathbb{R}^M$, respectively, let A be an $M \times N$ real matrix. We consider the problem of finding

$$x^* \in C \text{ such that } Ax^* \in Q.$$

This problem is called the split feasibility problem (SFP) which was first studied in Euclidean spaces by Censor and Elfving [1]. It was subsequently studied by Xu [2] in Hilbert spaces.

Byrne [3] introduced the CQ algorithm which takes an initial point x_0 arbitrarily, and defines the iterative step as

$$x_{k+1} = P_C(I - \gamma A^T(I - P_Q)A)x_k, \tag{1.1}$$

where $0 < \gamma < 2/\rho(A^TA)$ and $\rho(A^TA)$ is the spectral radius of A^TA. Recently, He et al. [4] developed a self-adaptive method for solving a variational problem. Subsequently, a number of self-adaptive projection methods were presented to solve SFP, see also [5–13]. Preliminary numerical results show that they are generally promising. The implementation of these algorithms, however, involves the computation of the projections P_C and P_Q

*Corresponding author.

Published by The Mathematical Association of Thailand.

and therefore causes additional difficulty in the case P_C and P_Q do not have closed-form expressions.

In 2004, Yang [5] introduced the relaxed CQ algorithm, by replacing P_C and P_Q by the projection of half-spaces C_k and Q_k which are given by

$$C_k = \{x \in \mathbb{R}^{\mathbb{N}} : c(x_k) \leq \langle \xi_k, x - x_k \rangle\}, \tag{1.2}$$

where $\xi_k \in \partial c(x_k)$ and

$$Q_k = \{y \in \mathbb{R}^{\mathbb{M}} : q(Ax_k) \leq \langle \eta_k, y - Ax_k \rangle\}, \tag{1.3}$$

where $\eta_k \in \partial q(Ax_k)$.

In 2012, López et al. [14] introduced a new way to select the stepsize and also practiced this way of selecting stepsizes for variants of the CQ algorithm, including relaxed CQ algorithm. They introduced the following:

Algorithm 1.1. Choose an initial guess $x_0 \in \mathbb{R}^{\mathbb{N}}$ arbitrarily. Assume that $x_k \in C$ has been constructed and $\nabla f_k(x_k) \neq 0$. Then we calculate the $(k+1)$ iterate x_{k+1} via the formula

$$x_{k+1} = P_{C_k}(x_k - \beta_k(A^T(I - P_{Q_k})Ax_k)), \forall k \geq 0, \tag{1.4}$$

where the stepsize β_k is defined by

$$\beta_k = \frac{\rho_k f_k(x_k)}{\|\nabla f_k(x_k)\|^2}\ , \tag{1.5}$$

and $\{\rho_k\}$ is a sequence in $(0, 4)$ such that $\liminf_{k\to\infty} \rho_k(4 - \rho_k) > 0$ and $f_k(x_k) = \frac{1}{2}\|(I - P_{Q_k})Ax_k\|^2, k \geq 1$.

It was proved that the sequence $\{x_k\}$ generated by Algorithm 1.1 converges to a solution of SFP.

In 2005, Qu and Xiu [15] modified the relaxed CQ algorithm by adopting Armijo-line searches in Euclidean spaces. Subsequently, Gibali et al. [16] extended results of Qu and Xiu [15] to Hilbert spaces as follows:

$$x_{k+1} = P_{C_k}(x_k - \beta_k \nabla f_k(y_k)) \tag{1.6}$$

$$y_k = P_{C_k}(x_k - \beta_k \nabla f_k(x_k)), \tag{1.7}$$

where $\beta_k = \gamma l^{m_k}$ and m_k is the smallest nonnegative integer, and $\gamma > 0, l \in (0, 1)$ and $\mu \in (0, 1)$ such that

$$\beta_k \|\nabla f_k(x_k) - \nabla f_k(y_k)\| \leq \mu \|x_k - y_k\|. \tag{1.8}$$

Dang and Gao [17], using the extragradient strategy, proposed two double projection algorithms for SFP, as follows:

For every k, define the function $F_k : \mathbb{R}^{\mathbb{N}} \to \mathbb{R}^{\mathbb{N}}$ as

$$F_k(x) := A^T(I - P_{Q_k})Ax\ .$$

Algorithm 1.2. Step 0. Select a point $x_0 \in C$ arbitrarily. For any $\gamma > 0, l \in (0,1), \lambda > 1, t_k \in (0,2)$. Set $k = 0$.
Step 1. Find $y_k = P_{C_k}(x_k - \beta_k F_k(x_k))$, where $\beta_k = \gamma^{m_k}$ and m_k is smallest nonnegative integer such that

$$\langle F_k(x_k), x_k - y_k\rangle \geq \lambda\langle F_k(x_k) - F_k(y_k), x_k - y_k\rangle. \tag{1.9}$$

Step 2. Compute

$$x_{k+1} = P_{C_k}[x_k - t_k \frac{\langle F_k(y_k), x_k - y_k\rangle}{\|F_k(y_k)\|^2} F_k(y_k)]. \tag{1.10}$$

Set $k = k+1$ and go to step 1.

It was shown that Algorithm 1.2 converges to a solution of SFP.

In this paper, we modify Algorithm 1.2 by replacing the stepsize that satisfies (1.9) by a new stepsize. Moreover, we use only one projection in computation.We then prove that the sequence generated by our algorithm converges to a solution of SFP. Numerical experiments show that the proposed methods are more efficient than other methods in comparison.

2. Preliminaries

In this section, we provide some basic concepts and notation. Let I denotes the identity operator, $Fix(T)$ denotes the set of the fixed points of an operator T i.e.,

$$Fix(T) := \{x : x = Tx\}.$$

Here Γ denotes the solution set of SFP, that is

$$\Gamma = \{y \in C : Ay \in Q\}. \tag{2.1}$$

Definition 2.1. Let $f : \mathbb{R}^{\mathbb{N}} \to \mathbb{R}$ be convex. The subdifferential of f at x is defined as

$$\partial f(x) = \{\xi \in \mathbb{R}^{\mathbb{N}} : f(y) \geq f(x) + \langle \xi, y - x\rangle, \forall\, y \in \mathbb{R}^{\mathbb{N}}\}. \tag{2.2}$$

Lemma 2.2 ([18, 19])**.** *Suppose that $f : \mathbb{R}^{\mathbb{N}} \to \mathbb{R}$ is convex. Then its subdifferential are uniformly bounded on any bounded subsets of $\mathbb{R}^{\mathbb{N}}$.*

Definition 2.3. Let $T : C \to H$. Then T is said to be β-inverse strongly monotone(ism) if there exists $\beta > 0$ such that for any $x, y \in C$,

$$\beta\|x - y\|^2 \leq \langle x - y, Tx - Ty\rangle. \tag{2.3}$$

Definition 2.4. Given $T : \mathbb{R}^{\mathbb{N}} \to \mathbb{R}^{\mathbb{N}}$. Then
a) T is said to be monotone if

$$\langle Tx - Ty, x - y\rangle \;\geq\; 0, \forall x, y \in \mathbb{R}^{\mathbb{N}},$$

b) T is said to be nonexpansive if

$$\|Tx - Ty\| \leq \|x - y\|, \forall x, y \in \mathbb{R}^{\mathbb{N}},$$

c) T is said to be co-coercive on $\mathbb{R}^{\mathbb{N}}$ with modulus $\alpha > 0$ if

$$\langle Tx - Ty, x - y\rangle \geq \alpha\|Tx - Ty\|^2, \forall x, y \in \mathbb{R}^{\mathbb{N}},$$

d) T is said to be Lipschitz continuous on $\mathbb{R}^{\mathbb{N}}$ with constant $L > 0$ if

$$\|Tx - Ty\| \leq L\|x - y\|, \forall x, y \in \mathbb{R}^{\mathbb{N}}.$$

Definition 2.5. A function $f : \mathbb{R}^{\mathbb{N}} \to \mathbb{R}$ is said to be lower semi-continuous (lsc) at x if $x_k \to x$ implies

$$f(x) \leq \liminf_{k\to\infty} f(x_k). \tag{2.4}$$

We know that the orthogonal projection of x onto C is defined as

$$P_C x := \min_{y\in C} \|x - y\|^2, x \in \mathbb{R}^{\mathbb{N}}. \tag{2.5}$$

Lemma 2.6 ([20]). *Let C be a nonempty closed convex subset of $\mathbb{R}^{\mathbb{N}}$. Then for any $x, y \in \mathbb{R}^{\mathbb{N}}$ and $z \in C$,*
(i) $\langle P_C x - x, z - P_C x\rangle \geq 0$,
(ii) $\|P_C x - P_C y\|^2 \leq \langle P_C x - P_C y, x - y\rangle$,
(iii) $\|P_C x - z\|^2 \leq \|x - z\|^2 - \|P_C x - x\|^2$.

From Lemma 2.6, the operator $I - P_C$ is also firmly nonexpansive, *i.e.*, for any $x, y \in \mathbb{R}^{\mathbb{N}}$,

$$\langle (I - P_C)x - (I - P_C)y, x - y\rangle \geq \|(I - P_C)x - (I - P_C)y\|^2. \tag{2.6}$$

3. A Novel Projection Algorithm and Its Convergence

As in [5], the following conditions are supposed to be satisfied:
(H1) The set C is defined as

$$C = \{x \in \mathbb{R}^{\mathbb{N}} : c(x) \leq 0\},$$

where $c : \mathbb{R}^{\mathbb{N}} \to \mathbb{R}$ is convex and C is nonempty.
The set Q is defined as

$$Q = \{y \in \mathbb{R}^{\mathbb{M}} : q(y) \leq 0\},$$

where $q : \mathbb{R}^{\mathbb{M}} \to \mathbb{R}$ is convex and Q is nonempty.
(H2) For any $x \in \mathbb{R}^{\mathbb{N}}$ and $y \in \mathbb{R}^{\mathbb{M}}$, a subgradient $\xi \in \partial c(x)$ and a subgradient $\eta \in \partial q(y)$ can be calculated.

We define the following halfspaces at point x_k, respectively,

$$C_k = \{x \in \mathbb{R}^{\mathbb{N}} : c(x_k) + \langle \xi_k, x - x_k\rangle \leq 0\},$$

where $\xi_k \in \partial c(x_k)$, and

$$Q_k = \{y \in \mathbb{R}^{\mathbb{M}} : q(Ax_k) + \langle \eta_k, y - Ax_k\rangle \leq 0\},$$

where $\eta_k \in \partial q(Ax_k)$.

Obviously, by the definition of subgradient, we know that the orthogonal projections onto C_k and Q_k may be computed directly by reason of the specific forms of C_k and Q_k, see [21].

In the following, for every k, we define the function $F_k : \mathbb{R}^{\mathbb{N}} \to \mathbb{R}^{\mathbb{N}}$ as

$$F_k(x) := A^T(I - P_{Q_k})Ax$$

and

$$f_k(x) = \frac{1}{2}\|(I - P_{Q_k})Ax\|^2, k \geq 1.$$

We next define our method as follows:

Algorithm 3.1. Step 0. Select a point $x_0 \in C$ arbitrarily, choose parameters $0 < t_k < 2, 0 < \rho_k < 4$. Set $k = 0$.

Step 1. Find $y_k = x_k - \beta_k F_k(x_k)$, where $\beta_k = \dfrac{\rho_k f_k(x_k)}{\|F_k(x_k)\|^2}$.

Step 2. Compute

$$x_{k+1} = P_{C_k}\left[y_k - t_k\left(\frac{\langle F_k(x_k) - F_k(y_k), y_k - x_k\rangle}{\|F_k(y_k)\|^2}\right)F_k(y_k)\right]. \tag{3.1}$$

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

We are in position to prove our convergence theorem of Algorithm 3.1.

Theorem 3.2. *Let $\{x_k\}$ be a sequence generated by Algorithm 3.1 and $\Gamma \neq \emptyset$. If $\liminf_{k\to\infty} \rho_k(4 - \rho_k) > 0$ and $\liminf_{k\to\infty} t_k(2 - t_k) > 0$, then $\{x_k\}$ converges to a solution of SFP.*

Proof. Let $z \in \Gamma$ and set $\alpha_k = \frac{\langle F_k(x_k) - F_k(y_k),\, y_k - x_k\rangle}{\|F_k(y_k)\|^2}$. Since $z \in \Gamma$ and $C \subset C_k$, $Q \subset Q_k$, it follows that $z = P_C(z) = P_{C_k}(z)$ and $Az = P_Q(Az) = P_{Q_k}(Az)$. Hence $z \in C_k$ and $F_k(z) = 0$ for all $k = 0, 1, 2, \ldots..$ From Lemma 2.6 (iii), we obtain

$$\begin{aligned}
\|x_{k+1} - z\|^2 &= \|P_{C_k}(y_k - t_n\alpha_k F_k(y_k)) - z\|^2 \\
&\leq \|y_k - t_n\alpha_k F_k(y_k) - z\|^2 - \|x_{k+1} - y_k + t_k\alpha_k F_k(y_k)\|^2 \\
&= \|y_k - z\|^2 + t_k^2\alpha_k^2\|F_k(y_k)\|^2 - 2t_k\alpha_k\langle F_k(y_k), y_k - z\rangle \\
&\quad -\|x_{k+1} - y_k + t_k\alpha_k F_k(y_k)\|^2.
\end{aligned} \tag{3.2}$$

From (2.6) and $F_k(z) = 0$, we have

$$\begin{aligned}
\langle F_k(y_k), y_k - z\rangle &= \langle F_k(y_k) - F_k(z), y_k - z\rangle \\
&= \langle A^T(I - P_{Q_k})Ay_k - A^T(I - P_{Q_k})Az, y_k - z\rangle \\
&= \langle (I - P_{Q_k})Ay_k - (I - P_{Q_k})Az, Ay_k - Az\rangle \\
&\geq \|(I - P_{Q_k})Ay_k\|^2 \\
&= 2f_k(y_k).
\end{aligned}$$

It also follows that

$$\langle F_k(x_k), x_k - z\rangle \geq 2f_k(x_k). \tag{3.3}$$

By (3.3), we see that

$$\begin{aligned}
\|y_k - z\|^2 &= \|x_k - \beta_k F_k(x_k) - z\|^2 \\
&= \|x_k - z\|^2 + \beta_k^2\|F_k(x_k)\|^2 - 2\beta_k\langle F_k(x_k), x_k - z\rangle \\
&\leq \|x_k - z\|^2 + \beta_k^2\|F_k(x_k)\|^2 - 4\beta_k f_k(x_k).
\end{aligned} \tag{3.4}$$

Since $F_k(z)=0$ and F_k is monotone, we obtain

$$\begin{aligned}\langle F_k(y_k), y_k - z\rangle &= \langle F_k(y_k), y_k - x_k\rangle + \langle F_k(y_k), x_k - z\rangle \\ &= \langle F_k(y_k) - F_k(x_k), y_k - x_k\rangle + \langle F_k(x_k), y_k - x_k\rangle + \langle F_k(y_k), x_k - z\rangle \\ &\geq \langle F_k(x_k), y_k - x_k\rangle + \langle F_k(y_k), x_k - z\rangle \\ &= \langle F_k(x_k), y_k - x_k\rangle + \langle F_k(y_k), x_k - y_k\rangle + \langle F_k(y_k) - F_k(z), y_k - z\rangle \\ &\geq \langle F_k(x_k) - F_k(y_k), y_k - x_k\rangle. \qquad (3.5)\end{aligned}$$

Combining (3.2)-(3.5), we obtain

$$\begin{aligned}\|x_{k+1} - z\|^2 &\leq \|x_k - z\|^2 + \beta_k^2\|F_k(x_k)\|^2 - 4\beta_k f_k(x_k) + t_k^2\alpha_k^2\|F_k(y_k)\|^2 \\ &\quad -2t_k\alpha_k\langle F_k(x_k) - F_k(y_k), y_k - x_k\rangle - \|x_{k+1} - y_k + t_k\alpha_k F_k(y_k)\|^2 \\ &= \|x_k - z\|^2 + \frac{\rho_k^2 f_k^2(x_k)}{\|F_k(x_k)\|^4}\|F_k(x_k)\|^2 - 4\frac{\rho_k f_k(x_k)}{\|F_k(y_k)\|^2} f_k(x_k) \\ &\quad + t_k^2\frac{\langle F_k(x_k) - F_k(y_k), y_k - x_k\rangle^2}{\|F_k(y_k)\|^4}\|F_k(y_k)\|^2 \\ &\quad -2t_k\frac{\langle F_k(x_k) - F_k(y_k), y_k - x_k\rangle^2}{\|F_k(y_k)\|^2} - \|x_{k+1} - y_k + t_k\alpha_k F_k(y_k)\|^2 \\ &= \|x_k - z\|^2 - \rho_k(4-\rho_k)\frac{f_k^2(x_k)}{\|F_k(x_k)\|^2} \\ &\quad -t_k(2-t_k)\frac{\langle F_k(x_k) - F_k(y_k), y_k - x_k\rangle^2}{\|F_k(y_k)\|^2} \\ &\quad -\|x_{k+1} - y_k + t_k\alpha_k F_k(y_k)\|^2. \qquad (3.6)\end{aligned}$$

Since $0<\rho_k<4$ and $0<t_k<2$, it implies that

$$\|x_{k+1} - z\| \leq \|x_k - z\|. \qquad (3.7)$$

Thus $\lim_{k\to\infty}\|x_k - z\|$ exists and $\{x_k\}$ is bounded. Again, from (3.6), it follows that

$$\liminf_{k\to\infty} \rho_k(4-\rho_k)\frac{f_k^2(x_k)}{\|F_k(x_k)\|^2} = 0, \qquad (3.8)$$

and

$$\liminf_{k\to\infty} t_k(2-t_k)\frac{\langle F_k(x_k) - F_k(y_k), y_k - x_k\rangle^2}{\|F_k(y_k)\|^2} = 0 \qquad (3.9)$$

which implies by our assumptions that

$$\lim_{k\to\infty}\frac{f_k^2(x_k)}{\|F_k(x_k)\|^2} = 0 \qquad (3.10)$$

and

$$\lim_{k\to\infty}\frac{\langle F_k(x_k) - F_k(y_k), y_k - x_k\rangle^2}{\|F_k(y_k)\|^2} = 0\ . \qquad (3.11)$$

On the other hand, we can check that $\{\|F_k(x_k)\|\}$ is bounded. So we get $\lim_{k\to\infty} f_k(x_k) = 0$. This means $\lim_{k\to\infty} \|(I - P_{Q_k})Ax_k\| = 0$. We note that

$$\beta_k\|F_k(x_k)\| = \frac{\rho_k f_k(x_k)}{\|F_k(x_k)\|^2}\|F_k(x_k)\| \to 0, \text{ as } k \to \infty. \tag{3.12}$$

Furthermore, by (3.6), we see that

$$\lim_{k\to\infty} \|x_{k+1} - y_k + t_k\alpha_k F_k(y_k)\| = 0. \tag{3.13}$$

So, we get

$$t_k\alpha_k\|F_k(y_k)\| = t_k\frac{\langle F_k(x_k) - F_k(y_k), y_k - x_k\rangle}{\|F_k(y_k)\|^2}\|F_k(y_k)\| \to 0, \text{ as } k \to \infty. \tag{3.14}$$

By (3.13) and (3.14), we obtain $\lim_{k\to\infty} \|x_{k+1} - y_k\| = 0$. On the other hand, from (3.12) and $y_k = x_k - \beta_k F_k(x_k)$, we get $\lim_{k\to\infty} \|y_k - x_k\| = 0$. Hence $\lim_{k\to\infty} \|x_{k+1} - x_k\| = 0$. Since $\{x_k\}$ is bounded, there exists a subsequence $\{x_{k_i}\}$ of $\{x_k\}$ such that $x_{k_i} \to \bar{x} \in \mathbb{R}^{\mathbb{N}}$.

We will show that $\bar{x} \in \Gamma$. Since $x_{k_i+1} \in C_{k_i}$, by the definition of C_{k_i}, we have

$$c(x_{k_i}) + \langle \xi_{k_i}, x_{k_i+1} - x_{k_i}\rangle \le 0, \forall i = 1, 2, ..., \tag{3.15}$$

where $\xi_{k_i} \in \partial c(x_{k_i})$. From (3.15), it follows that

$$c(x_{k_i}) \le \|\xi_{k_i}\|\|x_{k_i+1} - x_{k_i}\|. \tag{3.16}$$

By the lsc of c, $x_{k_i} \to \bar{x}$, the boundedness of ∂c and (3.16), we conclude that

$$c(\bar{x}) \le \liminf_{i\to\infty} c(x_{k_i}) \le 0. \tag{3.17}$$

Thus $\bar{x} \in C$.

Next, we show that $A\bar{x} \in Q$. Since $P_{Q_{k_i}}(Ax_{k_i}) \in Q_{k_i}$, we have

$$q(Ax_{k_i}) + \langle \eta_{k_i}, P_{Q_{k_i}}(Ax_{k_i}) - Ax_{k_i}\rangle \le 0, \forall i = 1, 2, ..., \tag{3.18}$$

where $\eta_{k_i} \in \partial q(Ax_{k_i})$. Then we get

$$q(Ax_{k_i}) \le \|\eta_{k_i}\|\|P_{Q_{k_i}}(Ax_{k_i}) - Ax_{k_i}\|. \tag{3.19}$$

By the lsc of q, $Ax_{k_i} \to A\bar{x}$, the boundedness of ∂q and (3.19), we obtain

$$q(A\bar{x}) \le \liminf_{i\to\infty} q(Ax_{k_i}) \le 0. \tag{3.20}$$

Thus $A\bar{x} \in Q$. Hence $\bar{x}$ is a solution of the SFP. This completes the proof. ■

4. Numerical Experiments

In this section, we will test two numerical examples to show that our proposed method converges faster than that of Dang and Gao [17].

Example 4.1. Consider the following LASSO problem [22]:

$\min\left\{\frac{1}{2}\|Ax - b\|^2 : x \in \mathbb{R}^5, \|x\|_1 \le 1\right\}$,

where

$$A = \begin{pmatrix} 1 & -3 & 2 & 1 & 0 \\ 5 & -6 & 1 & -1 & 1 \\ 4 & 2 & 3 & 0 & -2 \\ 0 & 2 & -2 & 1 & 9 \\ 0 & -1 & 3 & 0 & 1 \end{pmatrix} \quad \text{and} \quad b = (6, 12, 9, 0, 1).$$

In Algorithm 1.2 and Algorithm 3.1, we take $t_k = 1$ and $\rho_k = 0.14$.

We define $C = \{x \in \mathbb{R}^5 : \|x\|_1 \leq 1\}$ and $Q = \{b\}$. Since the projection onto the closed convex C does not have a closed form solution, we will make use of the subgradient projection. Define a convex function $c(x) = \|x\|_1 - 1$ and denote the level set C_k by:

$$C_k = \{x \in \mathbb{R}^5 : c(x_k) + \langle \xi_k, x - x_k \rangle \leq 0\}, \tag{4.1}$$

where $\xi_k \in \partial c(x_k)$. Then the orthogonal projection onto C_k can be calculated by the following:

$$P_{C_k}(x) = \begin{cases} x, & \text{if } c(x_k) + \langle \xi_k, x - x_k \rangle \leq 0, \\ x - \dfrac{c(x_k) + \langle \xi_k, x - x_k \rangle}{\|\xi_k\|^2} \xi_k, & \text{otherwise.} \end{cases} \tag{4.2}$$

It is worth noting that the subdifferential ∂c at x_k is

$$\partial c(x_k) = \begin{cases} 1, & \text{if } x_k > 0, \\ [-1, 1], & \text{if } x_k = 0, \\ -1, & \text{if } x_k < 0. \end{cases} \tag{4.3}$$

The iteration process is stopped when $E_k = \|x_{k+1} - x_k\|_2 < 10^{-4}$. We denote CPU and Iter by CPU time and number of iteration, respectively.

TABLE 1. The numerical results of Example 4.1

	Initial point	Method	CPU	Iter
Case1	$x_0 = (-1, 0, 2, 0, -1)$	Algorithm 1.2	0.0198	60
		Algorithm 3.1	0.0016	50
Case2	$x_0 = (-2, 1, 2, 1, 9)$	Algorithm 1.2	0.0091	71
		Algorithm 3.1	0.00003	63
Case3	$x_0 = (-2, 1, 4, 0, 2)$	Algorithm 1.2	0.0116	69
		Algorithm 3.1	0.00004	58

From Table 1, we see that our proposed method which is defined by Algorithm 3.1 converges faster than Algorithm 1.2 of Dang and gao [17].

The convergence behavior of the error E_k for each cases is shown in Figures 1-3, respectively.

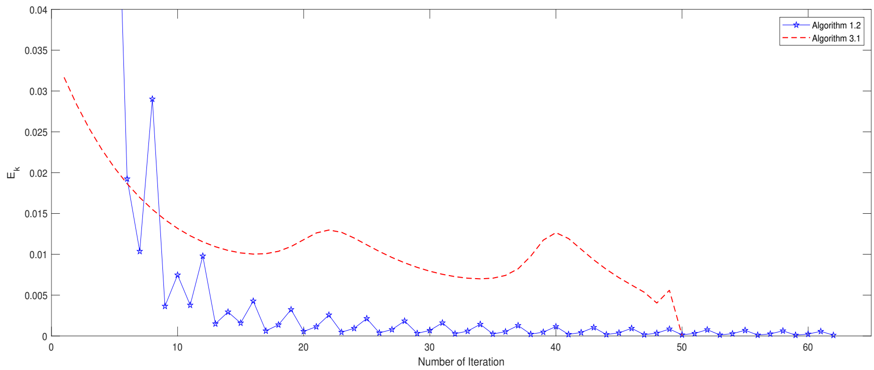

FIGURE 1. Error plotting E_k for Case 1

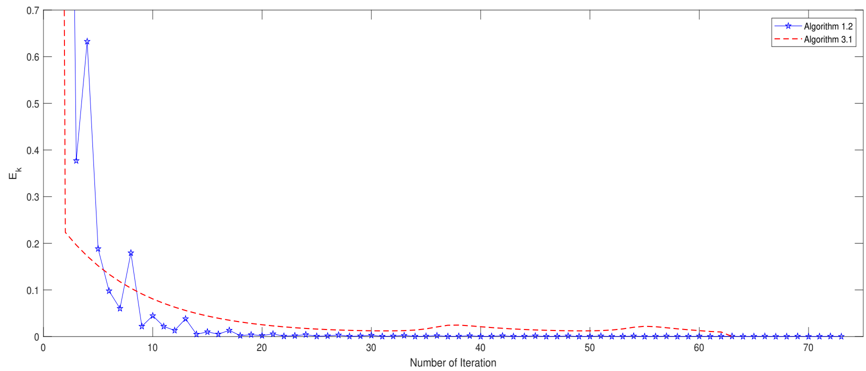

FIGURE 2. Error plotting E_k for Case 2

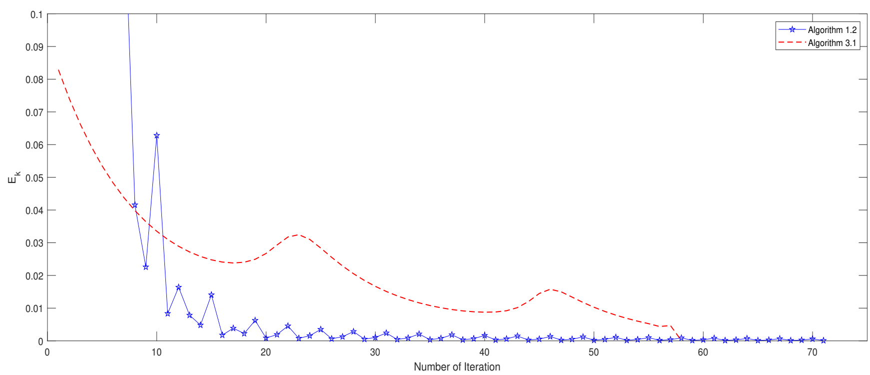

FIGURE 3. Error plotting E_k for Case 3

Example 4.2. In Algorithm 1.2 and Algorithm 3.1, we take $t_k = 1$ and $\rho_k = 0.14$.

The real $M \times N$ matrix A, $b \in \mathbb{R}^M$ and $x_0 \in \mathbb{R}^N$ are chosen randomly.

TABLE 2. The numerical results of Example 4.2

		Method	CPU	Iter
Case1	$M = 100, N = 100$	Algorithm 1.2	0.0426	95
		Algorithm 3.1	0.0076	60
Case2	$M = 200, N = 200$	Algorithm 1.2	0.043	107
		Algorithm 3.1	0.0026	61
Case3	$M = 300, N = 300$	Algorithm 1.2	0.0771	100
		Algorithm 3.1	0.0064	64

From Table 2, we see that our proposed method which is defined by Algorithm 3.1 converges faster than Algorithm 1.2 of Dang and gao [17].

The convergence behavior of the error E_k for each cases is shown in Figures 4-6, respectively.

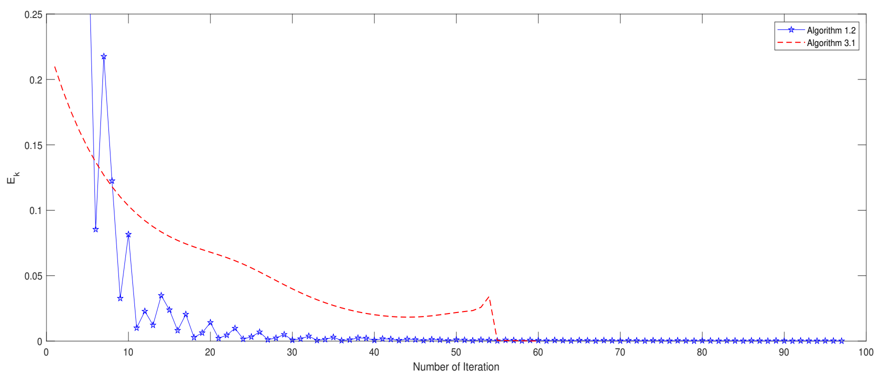

FIGURE 4. Error plotting E_k for Case 1.

Next, we provide some numerical experiments to the sparse signal recovery in compressed sensing. In signal processing, compressed sensing can be modeled as the following under determinated linear equation system:

$$y = Ax + \epsilon, \tag{4.4}$$

where $x \in \mathbb{R}^{\mathbb{N}}$ is a vector with m nonzero components to be recovered, $y \in \mathbb{R}^{\mathbb{M}}$ is the observed or measured data with noisy ϵ, and $A : \mathbb{R}^{\mathbb{N}} \to \mathbb{R}^{\mathbb{M}}$ $(M < N)$ is a bounded linear observation operator. A is sparse, and the range of it is not closed in most inverse problems; thus, A is often ill-condition and the problem is also ill-posed. When x is a

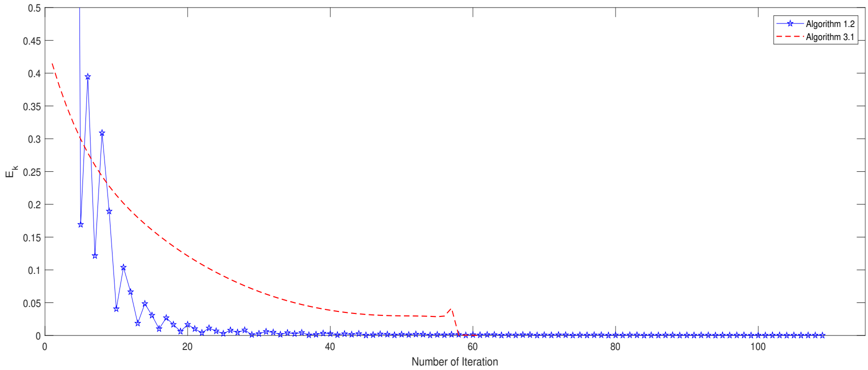

FIGURE 5. Error plotting E_k for Case 2.

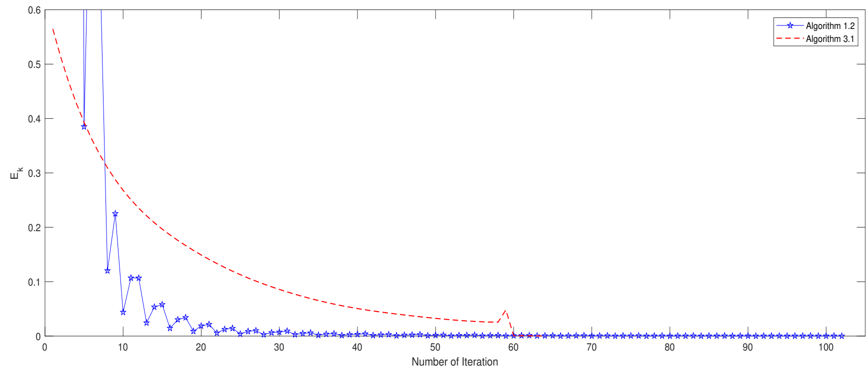

FIGURE 6. Error plotting E_k for Case 3.

sparse expansion, finding the solutions of (4.4) can be seen as solving the LASSO problem which is the following:

$$\min_{x\in\mathbb{R}^{\mathbb{N}}} \frac{1}{2}\|y - Ax\|^2 \text{ subjectto } \|x\|_1 \le t, \tag{4.5}$$

where $t > 0$ is a given constant. In particular, if $C = \{x \in \mathbb{R}^{\mathbb{N}} : \|x\|_1 \le t\}$ and $Q = \{y\}$, then the LASSO problem can be considered as the SFP (1). From this point of view, we can apply the CQ algorithm to solve (4.5).

Example 4.3. In our experiment, we test two cases as follows:
Case 1 : $N = 512, M = 256$ and $m = 20$.
Case 2 : $N = 1024, M = 512$ and $m = 30$.

In what follows, let $\gamma = 0.5, l = 0.8, \lambda = 1.1$ and $t_k = 0.5$ in that of Algorithm 1.2 and let with $t_k = 0.5$ and $\rho_k = 0.5$ in Algorithm 3.1.

The sparse vector $x \in \mathbb{R}^{\mathbb{N}}$ is generated from uniform distribution in the interval $[-2, 2]$ with m nonzero elements. The matrix $A \in \mathbb{R}^{\mathbb{M}\times\mathbb{N}}$ is generated from a normal distribution with mean zero and variance one. The observation y is generated by white Gaussian noise with signal-to-noise ratio SNR=40.The process is started with $t = m$ and initial point $x_1 = 0$.

We next give some numerical results by using the relaxed CQ algorithms defined by Algorithm 1.2 and Algorithm 3.1.

The restoration accuracy is measured by the mean squared error as follows:

$$MSE = \frac{1}{N}\|x_k - x^*\|^2 < 10^{-5}, \tag{4.6}$$

where x_k is the recovered signal signal and x^* is the original signal.

TABLE 3. The numerical results of Example 4.3

		Method	CPU
Case1	$N = 512, M = 256, m = 20$	Algorithm 1.2	2.7195
		Algorithm 3.1	0.0069
Case2	$N = 1024, M = 512, m = 30$	Algorithm 1.2	7.0966
		Algorithm 3.1	0.0741

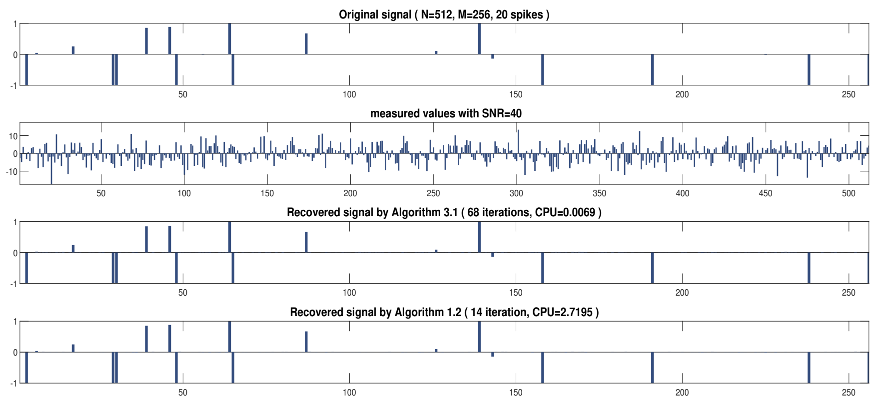

FIGURE 7. From top to bottom: original signal, observation data, recovered signal by Algorithm 3.1, recovered signal by Algorithm 1.2 in Case 1, respectively.

5. Conclusions

We have proposed a new spliting algorithm for solving the split feasibility problem by using the self-adaptive technique. The convergence theorem was proved under suitable conditions. The numerical experiments were performed to show the efficiency of our algorithms. The reports revealed that our method outperforms other methods in CPU time and number of iterations.

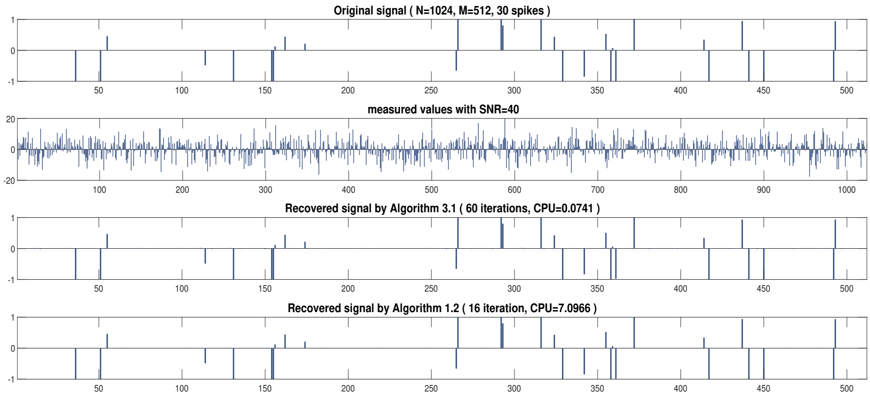

FIGURE 8. From top to bottom: original signal, observation data, recovered signal by Algorithm 3.1, recovered signal by Algorithm 1.2 in Case 2, respectively.

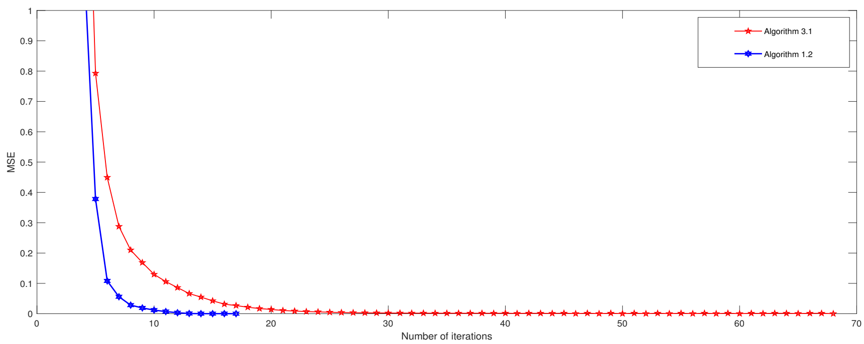

FIGURE 9. MSE versus number of iterations in Case 1.

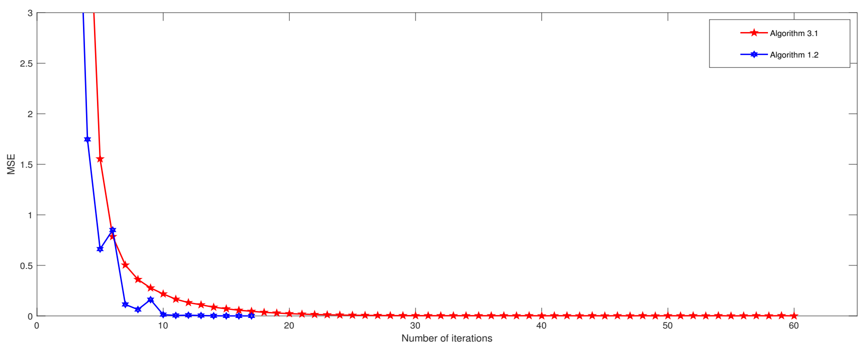

FIGURE 10. MSE versus number of iterations in Case 2.

ACKNOWLEDGMENTS

The authors would like to thank University of Phayao for supporting.

References

[1] Y. Censor, T. Elfving, A multiprojection algorithm using Bregman projections in a product space, Numerical Algorithms 8 (1994) 221–239.

[2] H. Xu, A variate Krasnosel´ ski-Mann algorithm and the multiple-set split feasibility problem, Inverse Problems 22 (2006) 2021–2034.

[3] C. Byrne, Iterative oblique projection onto convex sets and the split feasibility problem, Inverse Problems 18 (2002) 441–453.

[4] B. He, Inexact implicit methods for monotone general variational inequalities, Mathematical Programming 35 (1999) 199–217.

[5] Q. Yang, The relaxed CQ algorithm solving the split feasibility problem, Inverse Problems 20 (2004) 1261–1266.

[6] Y.N. Yang, Q. Yang, S. Zhang, Modified alternating direction methods for the modified multiple-sets split feasibility problems, Journal of Optimization Theory and Applications 163 (2014) 130–147.

[7] W.X. Zhang, D. Han, Z.B. Li, A self-adaptive projection method for solving the multiplesets split feasibility problem, Inverse Problem, 25 (2009) 115001, 16 pages.

[8] J.L. Zhao, Q. Yang, Self-adaptive projection methods for the multiple-sets split feasibility problem, Inverse Problem, 27 (2011) 035009 13 pages.

[9] J. Zhao, Q. Yang, Several solution methods for the split feasibility problem, Inverse Problems 21 (2005) 1791–1799.

[10] S. Kesornprom, N. Pholasa, P. Cholamjiak, On the convergence analysis of the gradient-CQ algorithms for the split feasibility problem, Numer. Algor. 84 (2019) 997–1017.

[11] S. Suantai, N. Pholasa, P. Cholamjiak, The modified inertial relaxed CQ algorithm for solving the split feasibility problems, J. Indust. Manag. Optim. 14 (2018) 1595–1615.

[12] S. Suantai, U. Witthayarat, Y. Shehu, P. Cholamjiak, Iterative methods for the split feasibility problem and the fixed point problem in Banach spaces, Optimization 68 (2019) 955–980.

[13] S. Suantai, Y. Shehu, P. Cholamjiak, O.S. Iyiola, Strong convergence of a self-adaptive method for the split feasibility problem in Banach spaces, J. Fixed Point Theory Appl. (2018) https://doi.org/10.1007/s11784-018-0549-y.

[14] G. López, V. Martín-Márquez, F. Wang, H.K. Xu, Solving the split feasibility problem without prior knowledge of matrix norms, Inverse Probl. 28 (2012) 085004.

[15] B. Qu, N. Xiu, A note on the CQ algorithm for the split feasibility problem, Inverse Problem 21 (2005) 1655–1665.

[16] A. Gibali, L.W. Liu, Y.C. Tang, Note on the modified relaxation CQ algorithm for the split feasibility problem, Optimization Letters. 12 (2017) 1–14.

[17] Y. Dang, Y. Gao, The strong convergence of a KM-CQ-like algorithm for a split feasibility problem, Inverse Prob. 27 (2011) 015007.

[18] Y. Gao, Nonsmooth Optimization (in Chinese), Science Press, Beijing, 2008.

[19] R.T. Rockafellar, Convex Analysis, Princeton University Press, 1971.

[20] E.H. Zarantonello, Projections on Convex Set in Hilbert Space and Spectral Theory, in Contributions to Nonlinear Functional Analysis (ed. E. H. Zarantonello), Academic, New York, 1971.

[21] H. Bauschke, J.M. Borwein, On projection algorithms for solving convex feasibility problems, SIAM Review 38 (1996) 367–426.

[22] R. Tibshirani, Regression shrinkage and selection via the lasso, J.R. Stat. Soc. Ser. B. Stat. Methodol. 58 (1996) 267–288.