J. Nonlinear Funct. Anal. 2021 (2021), Article ID 39 https://doi.org/10.23952/jnfa.2021.39

MATHRES

Journal of Nonlinear Functional Analysis

Available online at http://jnfa.mathres.org

TWO MODIFIED RELAXED CQ ALGORITHMS FOR THE MULTIPLE-SETS SPLIT FEASIBILITY PROBLEM

YAXUAN ZHANG*, YINGYING LI

College of Science, Civil Aviation University of China, Tianjin 300300, China

Abstract. The multiple-sets split feasibility problem is to find a point $x^* \in \bigcap_{i=1}^{t} C_i$ such that $Ax^* \in \bigcap_{j=1}^{r} Q_j$, where $C_i \subset \mathscr{H}_1$ and $Q_j \subset \mathscr{H}_2$ are nonempty, closed, and convex subsets, $\mathscr{H}_1$ and $\mathscr{H}_2$ are Hilbert spaces, and $A : \mathscr{H}_1 \to \mathscr{H}_2$ is a bounded and linear operator. In this paper, we present two modified relaxed CQ algorithms with the step size determined by the Armijo-line search. Under mild conditions, we establish the weak convergence, and provide numerical experiments to illustrate the effectiveness of the proposed algorithms.

Keywords. Armijo-line search; CQ algorithm; Multiple-sets split feasibility problem.

1. INTRODUCTION

In this paper, we focus on the multiple-sets split feasibility problem (MSSFP), which is formulated as follows.

$$\text{Find a point } x^* \in C = \bigcap_{i=1}^{t} C_i \text{ such that } Ax^* \in Q = \bigcap_{j=1}^{r} Q_j, \tag{1.1}$$

where $A : \mathscr{H}_1 \to \mathscr{H}_2$ is a bounded and linear operator, $C_i \subset \mathscr{H}_1$, $i = 1, \cdots, t$, and $Q_j \subset \mathscr{H}_2$, $j = 1, \cdots, r$ are nonempty, closed, and convex sets, and $\mathscr{H}_1$ and $\mathscr{H}_2$ are Hilbert spaces. If $t = r = 1$, it is easy to see that the MSSFP reduces to the split feasibility problem (SFP) as follows.

$$\text{Find a point } x^* \in C \text{ such that } Ax^* \in Q,$$

where $C \subset \mathscr{H}_1$ and $Q \subset \mathscr{H}_2$ are nonempty, closed, and convex sets. The SFP was first introduced in Euclidean spaces by Censor and Elfving in [1]. They proposed the following iterative algorithm

$$x^{k+1} = A^{-1}(I + AA^T)^{-1}(AP_C(x^k + AA^T P_Q(Ax^k))),$$

*Correspoinding author.

E-mail addresses: bunnyxuan@tju.edu.cn (Y. Zhang), yingyl99@163.com (Y. Li).

Received April 7, 2021; Accepted November 25, 2021.

where T denotes the transpose of the matrix, I denotes the identity matrix, and P_C and P_Q denote the metric projections onto C and Q, respectively.

Note that in each iteration one has to calculate the inverse matrix, which is difficult to compute in high dimensional spaces. To avoid this difficulty, Byrne, in [2, 3], introduced the following CQ algorithm

$$x^{k+1} = P_C(x^k - \alpha_k A^*(I - P_Q)Ax^k), \tag{1.2}$$

where $\alpha_k \in (0, \frac{2}{L})$, and L is the maximum eigenvalue of A^TA. It was proved that the iteration $\{x^k\}$ converges to a solution of the SFP.

Compared with Censor and Elfving's algorithm [1], the CQ algorithm seems more easily to execute since it only deals with metric projections. When P_C and P_Q have explicit expressions, the CQ algorithm is easy to carry out. However, we have to mention that P_C and P_Q have no explicit formulas in general, thus the computation of P_C and P_Q is itself an optimization problem.

To avoid the expensive computation of P_C and P_Q, Yang, in [4], proposed the relaxed CQ algorithm in finite dimensional spaces. The main idea is illustrated below. Let the closed convex subsets C and Q be given as follows

$$C = \{x \in \mathscr{H}_1 \mid c(x) \le 0\} \quad \text{and} \quad Q = \{y \in \mathscr{H}_2 \mid q(y) \le 0\},$$

where $c : \mathscr{H}_1 \to \mathbf{R}$ and $q : \mathscr{H}_2 \to \mathbf{R}$ are weakly lower semicontinuous, convex and proper functions. Define

$$C^k = \{x \in H_1 \mid c(x^k) + \langle \xi^k, x - x^k \rangle \le 0\},$$

and

$$Q^k = \{y \in H_2 \mid q(Ax^k) + \langle \eta^k, y - Ax^k \rangle \le 0\},$$

where $\xi^k \in \partial c(x^k)$, $\eta^k \in \partial q(Ax^k)$, $\partial c(x^k)$, and $\partial q(Ax^k)$ are the subdifferentials of c and q at the point x^k and Ax^k, respectively. Then C^k and Q^k are sequences of closed half spaces containing C and Q, respectively. Thus P_{C^k} and P_{Q^k} have explicit expressions and are easy to calculate. The algorithm is

$$x^{k+1} = P_{C^k}(x^k - \alpha_k A^*(I - P_{Q^k})Ax^k), \tag{1.3}$$

where $\alpha_k \in (0, \frac{2}{L})$, and L is the maximum eigenvalue of A^TA.

In 2010, Xu, in [5], proved that the relaxed CQ algorithm is still applicable in infinite dimensional real Hilbert spaces. To be more precise, he proved the weak convergence of the relaxed CQ algorithm. For the MSSFP (1.1) in finite dimensional spaces, Censor et al., in [6], defined an objective function $p(x)$ to measure the distance from a point to all sets

$$p(x) = \frac{1}{2}\sum_{i=1}^{t} \lambda_i \|x - P_{C_i}(x)\|^2 + \frac{1}{2}\sum_{j=1}^{r} \beta_j \|Ax - P_{Q_j}(Ax)\|^2,$$

where $\lambda_i > 0$, $\beta_j > 0$ for every i and j, and $\sum_{i=1}^t \lambda_i + \sum_{j=1}^r \beta_j = 1$. They proposed the following algorithm

$$x^{k+1} = P_\Omega(x^k - \alpha \nabla p(x^k)), \tag{1.4}$$

where Ω is an auxiliary closed subset, $0 < \alpha < \frac{2}{L}$, $L = \sum_{i=1}^t \lambda_i + \|A\|^2 \sum_{j=1}^r \beta_j$, and they investigated its convergence analysis; see [6] for more details.

In 2006, Xu, in [7], presented several kinds of algorithms in Hilbert spaces with the objective function below

$$T_i x = P_{C_i}(x - \alpha \sum_{j=1}^{r} \beta_j A^*(I - P_{Q_j})Ax),$$

where $\beta_j > 0$ for all $1 \leq j \leq r$. If $0 < \alpha < \frac{2}{L}$ and $L = \|A\|^2 \sum_{j=1}^{r} \beta_j$, then the sequence generated by any of the following algorithms converges to a solution of the MSSFP (1.1) weakly

(1) successive iteration

$$x^{k+1} = T_t \cdots T_2 T_1 x^k;$$

(2) parallel iteration

$$x^{k+1} = \sum_{i=1}^{t} \lambda_i T_i x^k,$$

where $\lambda_i > 0$ for all $1 \leq i \leq t$ and $\sum_{i=1}^{t} \lambda_i = 1$.

(3) cyclic iteration

$$x^{k+1} = T_{[k]} x^k,$$

where $[n] = n \bmod t$ is the mod function taking values in $\{1, 2, \cdots, t\}$.

It is worth noting that, for fixed step size, one needs to calculate (or at least estimate) the maximum eigenvalue of $A^T A$, which is also an obstacle for the numerical computation of the above algorithms. Thus, many scholars adopted variable (self-adaptive) step size instead of fixed step size.

In 2005, based on the CQ algorithm (1.2), Yang [8] proposed the step size

$$\alpha_k = \frac{\rho_k}{\|\nabla f(x^k)\|},$$

where $\{\rho_k\}$ is a sequence of positive real numbers satisfying $\sum_{n=0}^{\infty} \rho_k = \infty$ and $\sum_{n=0}^{\infty} \rho_k^2 < +\infty$, and $f(x) = \frac{1}{2}\|(I - P_Q)Ax\|^2$. Assuming that Q is bounded and A is a matrix with full column rank, Yang proved the convergence of the underlying algorithm in finite dimensional spaces. In 2012, López et al. [9] introduced another choice of the step size sequence $\{\alpha_k\}$ in the algorithm (1.3) as follows

$$\alpha_k = \frac{\rho_k f_k(x^k)}{\|\nabla f_k(x^k)\|^2},$$

where $0 < \rho_k < 4$ and $f_k(x) = \frac{1}{2}\|(I - P_{Q^k})Ax\|^2$. They proved the weak convergence of the iteration sequence in Hilbert spaces. The advantage of this choice of the step size lies in the fact that neither prior information about the matrix norm of A nor any other conditions on Q and A are required.

Recently, Qu and Xiu [10] modified the CQ algorithm and the relaxed CQ algorithm by using the Armijo-line search in Euclidean spaces. Gibali et al. [11] further extended this result to infinite dimensional Hilbert spaces. Motivated by Qu and Xiu's idea in [10], Zhao and Yang [12] introduced a self-adaptive CQ algorithm by adopting the Armijo-line search to solve the MSSFP (1.1), and proposed a relaxed projection version by using orthogonal projections onto half spaces instead of the metric projections onto the original convex sets. But as the same as in algorithm (1.4), their algorithm involves an auxiliary projection P_Ω. In 2012, Chen et al. [13] proposed a relaxed self-adaptive CQ algorithm for the MSSFP (1.1) without the projection P_Ω in finite dimensional spaces.

In this paper, motivated by Qu and Xiu [10], Gibali et al. [11], and Chen et al. [13], we propose two relaxed CQ algorithms with the Armijo-line search to solve the MSSFP (1.1), and prove their weak convergence in infinite dimensional Hilbert spaces.

The rest of the paper is arranged as follows. In Section 2, definitions and notions which will be useful for our analysis are presented. In Section 3, we present our algorithms and prove their weak convergence in infinite dimensional Hilbert spaces. In Section 4, the last section, we present some numerical simulations to show the validity of the proposed algorithms.

2. PRELIMINARIES

In this section, we give some symbols, and recall some definitions and basic results that will be used in this paper.

Throughout this paper, $\mathscr{H}$ denotes a real Hilbert space endowed with inner product $\langle\cdot,\cdot\rangle$ and its deduced norm $\|\cdot\|$, and I denotes the identity operator on $\mathscr{H}$. We denote by S the solution set of the MSSFP (1.1). Moreover, $x^k \to x$ ($x^k \rightharpoonup x$) represents that the sequence $\{x^k\}$ converges strongly (weakly) to x. We denote by $\omega_\omega(x^k)$ the set of weak cluster points of $\{x^k\}$.

Let $T : \mathscr{H} \to \mathscr{H}$ be an operator. Recall that T is said to be nonexpansive if, for all $x, y \in \mathscr{H}$,

$$\|Tx - Ty\| \leq \|x - y\|;$$

$T : \mathscr{H} \to \mathscr{H}$ is said to be firmly nonexpansive if, for all $x, y \in \mathscr{H}$,

$$||Tx - Ty||^2 \leq ||x - y||^2 - ||(I - T)x - (I - T)y||^2,$$

or equivalently

$$||Tx - Ty||^2 \leq \langle Tx - Ty, x - y\rangle.$$

It is known that T is firmly nonexpansive if and only if $I - T$ is firmly nonexpansive.

Let C be a nonempty, closed and convex subset of $\mathscr{H}$. Then the metric projection P_C from $\mathscr{H}$ onto C is defined as

$$P_C(x) = \underset{y\in C}{\operatorname{argmin}}\, ||x - y||^2,\ x \in \mathscr{H}.$$

The metric projection P_C is a firmly nonexpansive operator.

Recall that a function $f : \mathscr{H} \to \mathbf{R}$ is said to be weakly lower semicontinuous at $\widehat{x}$ if x^k converging weakly to $\widehat{x}$ implies

$$f(\widehat{x}) \leq \liminf_{k\to\infty} f(x^k).$$

If $\varphi : \mathscr{H} \to \mathbf{R}$ is a convex function, then the subdifferential of φ at x is defined as

$$\partial\varphi(x) = \{\xi \in \mathscr{H} \mid \varphi(y) \geq \varphi(x) + \langle\xi, y - x\rangle,\ \forall y \in \mathscr{H}\}.$$

Lemma 2.1. ([14]) *Let C be a nonempty, closed, and convex subset of $\mathscr{H}$. For any $x, y \in \mathscr{H}$, $z \in C$, the following assertions hold:*
(i) $\langle x - P_C x, z - P_C x\rangle \leq 0$;
(ii) $\|P_C x - P_C y\|^2 \leq \langle P_C x - P_C y, x - y\rangle$;
(iii) $\|P_C x - z\|^2 \leq \|x - z\|^2 - \|P_C x - x\|^2$.

Lemma 2.2. ([15]) *Let S be a nonempty, closed, and convex subset of $\mathscr{H}$, and let $\{x^k\}$ be a sequence in $\mathscr{H}$ that satisfies the following properties:*
(i) $\lim_{k\to\infty} \|x^k - x\|$ *exists for each* $x \in S$;

(ii) $\omega_\omega(x^k) \subset S$.
Then $\{x^k\}$ *converges weakly to a point in* S.

3. The Algorithms and Their Convergence

In this section, we introduce two modified relaxed CQ algorithms for the MSSFP (1.1), and prove their weak convergence. First, we assume that the following three assumptions hold.

(A1) S, the solution set of the MSSFP (1.1), is nonempty.

(A2) The sets C_i and Q_j can be expressed by

$$C_i = \{x \in \mathscr{H}_1 \mid c_i(x) \leq 0\}$$

and

$$Q_j = \{y \in \mathscr{H}_2 \mid q_j(y) \leq 0\},$$

where $c_i : \mathscr{H}_1 \to \mathbf{R}$ $(i = 1, \cdots, t)$, and $q_j : \mathscr{H}_2 \to \mathbf{R}$ $(j = 1, \cdots, r)$ are weakly lower semicontinuous and convex functions.

(A3) For any $x \in \mathscr{H}_1$ and $y \in \mathscr{H}_2$, at least one subgradient $\xi_i \in \partial c_i(x)$ and $\eta_j \in \partial q_j(y)$ can be calculated. The subdifferential ∂c_i and ∂q_j are bounded on the bounded sets.

Define two sets at point x^k by

$$C_i^k = \{x \in \mathscr{H}_1 \mid c_i(x^k) + \langle \xi_i^k, x - x^k \rangle \leq 0\}$$

and

$$Q_j^k = \{y \in \mathscr{H}_2 \mid q_j(Ax^k) + \langle \eta_j^k, y - Ax^k \rangle \leq 0\},$$

where $\xi_i^k \in \partial c_i(x^k)$ and $\eta_j^k \in \partial q_j(Ax^k)$. Define the function f_k by

$$f_k(x) = \frac{1}{2} \sum_{j=1}^{r} \beta_j \|(I - P_{Q_j^k})Ax\|^2,$$

where $\beta_j > 0$. Then it is easy to verify that the function $f_k(x)$ is convex and differentiable with gradient

$$\nabla f_k(x) = \sum_{j=1}^{r} \beta_j A^*(I - P_{Q_j^k})Ax,$$

and the Lipschitz constant of $\nabla f_k(x)$ is $L = \|A\|^2 \sum_{j=1}^{r} \beta_j$, for every $k \geq 1$.

We see that C_i^k $(i = 1, \cdots, t)$ and Q_j^k $(j = 1, \cdots, r)$ are half spaces, and that $C_i \subset C_i^k$, $Q_j \subset Q_j^k$, for all $k \geq 1$.

Algorithm 3.1. Give $\gamma > 0$, $l \in (0,1)$, and $\mu \in (0,1)$. Let x^0 be arbitrarily chosen. For $k = 1, 2, \cdots$, compute

$$\bar{x}^k = P_{C_{[k]}^k}(x^k - \alpha_k \nabla f_k(x^k)),$$

where $[k] = k \bmod t$ and $\alpha_k = \gamma l^{m_k}$ with m_k being the smallest non-negative integer such that

$$\alpha_k \|\nabla f_k(x^k) - \nabla f_k(\bar{x}^k)\| \leq \mu \|x^k - \bar{x}^k\|. \tag{3.1}$$

Construct the next iterate x^{k+1} by

$$x^{k+1} = P_{C_{[k]}^k}(x^k - \alpha_k \nabla f_k(\bar{x}^k)).$$

For step size α_k, we have the following assertion.

Lemma 3.2. ([10]) *The Armijo-line search terminates after a finite number of steps. In addition,* $\frac{\mu l}{L} < \alpha_k \leq \gamma$, $\forall k \geq 1$, *where* $L = \|A\|^2 \sum_{j=1}^{r} \beta_j$.

Now, we are ready to present our main result, which is the weak convergence of Algorithm 3.1.

Theorem 3.3. *Let* $\{x^k\}$ *be the sequence generated by Algorithm 3.1, and assumptions (A1), (A2), and (A3) hold. Then* $\{x^k\}$ *converges weakly to a solution of the MSSFP (1.1).*

Proof. Let x^* be a solution of the MSSFP. In view of $C \subset C_i \subset C_i^k$, $Q \subset Q_j \subset Q_j^k$, $i = 1, \cdots, t$, $j = 1, \cdots, r$, $k = 1, 2, \cdots$, we have $x^* = P_C(x^*) = P_{C_i}(x^*) = P_{C_i^k}(x^*)$ and $Ax^* = P_Q(Ax^*) = P_{Q_j}(Ax^*) = P_{Q_j^k}(Ax^*)$. Thus, $f_k(x^*) = 0$ and $\nabla f_k(x^*) = 0$.

First, we prove that $\{x^k\}$ is bounded. Following Lemma 2.1 (iii), we have

$$
\begin{aligned}
\|x^{k+1} - x^*\|^2 &= \|P_{C^k_{[k]}}(x^k - \alpha_k \nabla f_k(\bar{x}^k)) - x^*\|^2 \\
&\leq \|x^k - \alpha_k \nabla f_k(\bar{x}^k) - x^*\|^2 - \|x^{k+1} - x^k + \alpha_k \nabla f_k(\bar{x}^k)\|^2 \\
&= \|x^k - x^*\|^2 - \|x^{k+1} - x^k\|^2 - 2\alpha_k \langle \nabla f_k(\bar{x}^k), x^k - x^* \rangle \\
&\quad - 2\alpha_k \langle \nabla f_k(\bar{x}^k), x^{k+1} - x^k \rangle \\
&= \|x^k - x^*\|^2 - \|x^{k+1} - x^k\|^2 - 2\alpha_k \langle \nabla f_k(\bar{x}^k), x^{k+1} - x^* \rangle \\
&= \|x^k - x^*\|^2 - \|x^{k+1} - \bar{x}^k\|^2 - \|\bar{x}^k - x^k\|^2 - 2\langle x^{k+1} - \bar{x}^k, \bar{x}^k - x^k \rangle \\
&\quad - 2\alpha_k \langle \nabla f_k(\bar{x}^k), x^{k+1} - x^* \rangle \\
&= \|x^k - x^*\|^2 - \|x^{k+1} - \bar{x}^k\|^2 - \|\bar{x}^k - x^k\|^2 - 2\langle x^{k+1} - \bar{x}^k, \bar{x}^k - x^k \rangle \\
&\quad - 2\alpha_k \langle \nabla f_k(\bar{x}^k), x^{k+1} - \bar{x}^k \rangle - 2\alpha_k \langle \nabla f_k(\bar{x}^k), \bar{x}^k - x^* \rangle \\
&= \|x^k - x^*\|^2 - \|x^{k+1} - \bar{x}^k\|^2 - \|\bar{x}^k - x^k\|^2 - 2\alpha_k \langle \nabla f_k(\bar{x}^k), \bar{x}^k - x^* \rangle \\
&\quad - 2\langle \bar{x}^k - x^k + \alpha_k \nabla f_k(\bar{x}^k), x^{k+1} - \bar{x}^k \rangle.
\end{aligned} \tag{3.2}
$$

Due to the firmly nonexpensiveness of $I - P_{Q_j^k}$, $\nabla f_k(x^*) = 0$ and Lemma 3.2, we obtain that

$$
\begin{aligned}
2\alpha_k \langle \nabla f_k(\bar{x}^k), \bar{x}^k - x^* \rangle &= 2\alpha_k \langle \sum_{j=1}^{r} \beta_j A^*(I - P_{Q_j^k}) A\bar{x}^k - \sum_{j=1}^{r} \beta_j A^*(I - P_{Q_j^k}) Ax^*, \bar{x}^k - x^* \rangle \\
&= 2\alpha_k \sum_{j=1}^{r} \beta_j \langle (I - P_{Q_j^k}) A\bar{x}^k - (I - P_{Q_j^k}) Ax^*, A\bar{x}^k - Ax^* \rangle \\
&\geq 2\frac{\mu l}{L} \sum_{j=1}^{r} \beta_j \|(I - P_{Q_j^k}) A\bar{x}^k\|^2,
\end{aligned} \tag{3.3}
$$

Based on the definition of $\bar{x}^k$ and Lemma 2.1 (i), we know that

$$
\langle \bar{x}^k - x^k + \alpha_k \nabla f_k(x^k), x^{k+1} - \bar{x}^k \rangle \geq 0.
$$

This together with (3.1) obtains that

$$\begin{aligned}
&-\|x^{k+1}-\bar{x}^k\|^2-\|\bar{x}^k-x^k\|^2+2\langle x^k-\bar{x}^k-\alpha_k\nabla f_k(\bar{x}^k),x^{k+1}-\bar{x}^k\rangle\\
&\quad\le -\|x^{k+1}-\bar{x}^k\|^2-\|\bar{x}^k-x^k\|^2+2\langle x^k-\bar{x}^k-\alpha_k\nabla f_k(\bar{x}^k),x^{k+1}-\bar{x}^k\rangle\\
&\qquad +2\langle \bar{x}^k-x^k+\alpha_k\nabla f_k(x^k),x^{k+1}-\bar{x}^k\rangle\\
&\quad= -\|x^{k+1}-\bar{x}^k\|^2-\|\bar{x}^k-x^k\|^2+2\alpha_k\langle \nabla f_k(x^k)-\nabla f_k(\bar{x}^k),x^{k+1}-\bar{x}^k\rangle\\
&\quad\le -\|x^{k+1}-\bar{x}^k\|^2-\|\bar{x}^k-x^k\|^2+2\alpha_k\|\nabla f_k(x^k)-\nabla f_k(\bar{x}^k)\|\|x^{k+1}-\bar{x}^k\|\\
&\quad\le -\|x^{k+1}-\bar{x}^k\|^2-\|\bar{x}^k-x^k\|^2+\alpha_k^2\|\nabla f_k(x^k)-\nabla f_k(\bar{x}^k)\|^2+\|x^{k+1}-\bar{x}^k\|^2\\
&\quad\le -\|\bar{x}^k-x^k\|^2+\mu^2\|\bar{x}^k-x^k\|^2\\
&\quad= -(1-\mu^2)\|\bar{x}^k-x^k\|^2.
\end{aligned}\tag{3.4}$$

Combining (3.2), (3.3), and (3.4), we obtain

$$\|x^{k+1}-x^*\|^2\le \|x^k-x^*\|^2-(1-\mu^2)\|\bar{x}^k-x^k\|^2-2\frac{\mu l}{L}\sum_{j=1}^{r}\beta_j\|(I-P_{Q_j^k})A\bar{x}^k\|^2,\tag{3.5}$$

which implies that

$$\|x^{k+1}-x^*\|\le\|x^k-x^*\|.$$

Thus $\lim_{k\to\infty}\|x^{k+1}-x^*\|$ exists and thus $\{x^k\}$ is bounded. In (3.5), letting $k\to\infty$, we derive that

$$\lim_{k\to\infty}\|\bar{x}^k-x^k\|=0\tag{3.6}$$

and

$$\lim_{k\to\infty}\sum_{j=1}^{r}\beta_j\|(I-P_{Q_j^k})A\bar{x}^k\|^2=0.$$

Hence, for every $j=1,2,\cdots,r$, we have

$$\lim_{k\to\infty}\|(I-P_{Q_j^k})A\bar{x}^k\|=0.\tag{3.7}$$

Next we prove $\lim_{k\to\infty}\|x^{k+1}-x^k\|=0$. In fact, due to the nonexpansiveness of $P_{C_j^k}$ and (3.1), it holds that

$$\begin{aligned}
\|x^{k+1}-x^k\|&\le\|x^{k+1}-\bar{x}^k\|+\|\bar{x}^k-x^k\|\\
&=\|P_{C_{[k]}^k}(x^k-\alpha_k\nabla f_k(\bar{x}^k))-P_{C_{[k]}^k}(x^k-\alpha_k\nabla f_k(x^k))\|+\|\bar{x}^k-x^k\|\\
&\le\|x^k-\alpha_k\nabla f_k(\bar{x}^k)-x^k+\alpha_k\nabla f_k(x^k)\|+\|\bar{x}^k-x^k\|\\
&=\alpha_k\|\nabla f_k(\bar{x}^k)-\nabla f_k(x^k)\|+\|\bar{x}^k-x^k\|\\
&\le(1+\mu)\|\bar{x}^k-x^k\|.
\end{aligned}$$

By taking $k\to\infty$ in the above inequality and using (3.6), we obtain

$$\lim_{k\to\infty}\|x^{k+1}-x^k\|=0.\tag{3.8}$$

Since $\{x^k\}$ is bounded, we have that $\omega_\omega(x^k)$ is nonempty. Let $\widehat{x}\in\omega_\omega(x^k)$. Then, there exists a subsequence $\{x^{k_n}\}$ of $\{x^k\}$ such that $x^{k_n}\rightharpoonup\widehat{x}$.

Next, we prove that $\widehat{x}$ is a solution of MSSFP (1.1), which shows that $\omega_\omega(x^k) \subset S$. In fact, since $x^{k_n+1} \in C_{[k_n]}^{k_n}$, by the definition of $C_{[k_n]}^{k_n}$, we have

$$c_{[k_n]}(x^{k_n}) + \langle \xi_{[k_n]}^{k_n}, x^{k_n+1} - x^{k_n} \rangle \le 0, \tag{3.9}$$

where $\xi_{[k_n]}^{k_n} \in \partial c_{[k_n]}(x^{k_n})$. For every $i = 1, 2, \cdots, t$, we choose a subsequence $\{k_{n_s}\} \subset \{k_n\}$ such that $[k_{n_s}] = i$. Then (3.9) is reduced to

$$c_i(x^{k_{n_s}}) + \langle \xi_i^{k_{n_s}}, x^{k_{n_s}+1} - x^{k_{n_s}} \rangle \le 0.$$

Following the assumption (A3) on the boundedness of ∂c_i and (3.8), we see that there exists a constant M_1 such that

$$\begin{aligned} c_i(x^{k_{n_s}}) &\le \langle \xi_i^{k_{n_s}}, x^{k_{n_s}} - x^{k_{n_s}+1} \rangle \\ &\le \|\xi_i^{k_{n_s}}\| \|x^{k_{n_s}} - x^{k_{n_s}+1}\| \\ &\le M_1 \|x^{k_{n_s}} - x^{k_{n_s}+1}\| \to 0, \, s \to \infty. \end{aligned} \tag{3.10}$$

From the weak lower semicontinuity of each convex function $c_i, i = 1, 2, \cdots, t$, we deduce from (3.10) that $c_i(\widehat{x}) \le \liminf_{s\to\infty} c_i(x^{k_{n_s}}) \le 0$, *i.e.*, $\widehat{x} \in C = \bigcap_{i=1}^t C_i$. The fact that $I - P_{Q_j^{k_n}}$ is nonexpansive, together with (3.6), (3.7), and that A is a bounded and linear operator, yields that

$$\begin{aligned} \|(I - P_{Q_j^{k_n}})Ax^{k_n}\| &\le \|(I - P_{Q_j^{k_n}})Ax^{k_n} - (I - P_{Q_j^{k_n}})A\bar{x}^{k_n}\| + \|(I - P_{Q_j^{k_n}})A\bar{x}^{k_n}\| \\ &\le \|Ax^{k_n} - A\bar{x}^{k_n}\| + \|(I - P_{Q_j^{k_n}})A\bar{x}^{k_n}\| \\ &\le \|A\| \|x^{k_n} - \bar{x}^{k_n}\| + \|(I - P_{Q_j^{k_n}})A\bar{x}^{k_n}\| \to 0, \, n \to \infty. \end{aligned} \tag{3.11}$$

Since $P_{Q_j^{k_n}}(Ax^{k_n}) \in Q_j^{k_n}$, we have

$$q_j(Ax^{k_n}) + \langle \eta_j^{k_n}, P_{Q_j^{k_n}}(Ax^{k_n}) - Ax^{k_n} \rangle \le 0, \tag{3.12}$$

where $\eta_j^{k_n} \in \partial q_j(Ax^{k_n})$. From the boundedness assumption (A3), (3.11), and (3.12), there exists a constant M_2 such that

$$\begin{aligned} q_j(Ax^{k_n}) &\le \|\eta_j^{k_n}\| \|P_{Q_j^{k_n}}(Ax^{k_n}) - Ax^{k_n}\| \\ &= M_2 \|(I - P_{Q_j^{k_n}})Ax^{k_n}\| \to 0, \, n \to \infty, \end{aligned}$$

Then $q_j(A\widehat{x}) \le \liminf_{n\to\infty} q_j(Ax^{k_n}) \le 0$. This means that $A\widehat{x} \in Q = \bigcap_{j=1}^r Q_j$, and thus $\widehat{x} \in S$. Using Lemma 2.2, we conclude that the sequence $\{x^k\}$ converges weakly to a solution of MSSFP (1.1). □

Next, we introduce another algorithm to solve MSSFP (1.1) in Hilbert spaces. The definition of C_i, Q_j, C_i^k, and Q_j^k are the same as before. Let us assume without loss of generality that $r = t$. Define

$$g_k(x) = \frac{1}{2}\|(I - P_{Q_{[k]}^k})Ax\|^2,$$

where $[k] = k \bmod r$. Then the function $g_k(x)$ is convex and differentiable with gradient

$$\nabla g_k(x) = A^*(I - P_{Q_{[k]}^k})Ax.$$

Algorithm 3.4. Given constant $\gamma > 0$, $l \in (0,1)$, and $\mu \in (0,1)$, Let x^0 be arbitrarily chosen. For $k = 1, 2, \cdots$, compute

$$\bar{x}^k = P_{C^k_{[k]}}(x^k - \alpha_k \nabla g_k(x^k)),$$

where $\alpha_k = \gamma l^{m_k}$ with m_k being the smallest non-negative integer such that

$$\alpha_k \|\nabla g_k(x^k) - \nabla g_k(\bar{x}^k)\| \le \mu \|x^k - \bar{x}^k\|.$$

Construct the next iterate x^{k+1} by

$$x^{k+1} = P_{C^k_{[k]}}(x^k - \alpha_k \nabla g_k(\bar{x}^k)).$$

Theorem 3.5. *Let $\{x^k\}$ be the sequence generated by Algorithm 3.4, and the assumptions (A1), (A2), and (A3) hold. Then $\{x^k\}$ converges weakly to a solution of the MSSFP (1.1).*

Proof. The proof of the theorem is similar with Theorem 3.3. Here, we only present the differences. First, a simple calculation shows that

$$\begin{aligned} 2\alpha_k \langle \nabla g_k(\bar{x}^k), \bar{x}^k - x^* \rangle &= 2\alpha_k \langle A^*(I - P_{Q^k_{[k]}})A\bar{x}^k - A^*(I - P_{Q^k_{[k]}})Ax^*, \bar{x}^k - x^* \rangle \\ &= 2\alpha_k \langle (I - P_{Q^k_{[k]}})A\bar{x}^k - (I - P_{Q^k_{[k]}})Ax^*, A\bar{x}^k - Ax^* \rangle \\ &\ge 2\frac{\mu l}{L} \|(I - P_{Q^k_{[k]}})A\bar{x}^k\|^2. \end{aligned}$$

Next, similar with the proof of Theorem 3.3, we can obtain that $\{x^k\}$ is bounded and

$$\|x^{k+1} - x^*\|^2 \le \|x^k - x^*\|^2 - (1 - \mu^2)\|\bar{x}^k - x^k\|^2 - 2\frac{\mu l}{L}\|(I - P_{Q^k_{[k]}})A\bar{x}^k\|^2.$$

It follows that

$$\lim_{k\to\infty} \|\bar{x}^k - x^k\| = 0,$$

and

$$\lim_{k\to\infty} \|(I - P_{Q^k_{[k]}})A\bar{x}^k\| = 0.$$

Let $\widehat{x} \in \omega_w(x^k)$. Similar with the proof of Theorem 3.3, we know that $\widehat{x} \in C$. Now we prove $A\widehat{x} \in Q$. For every $j = 1, 2, \cdots, r$, we can choose a subsequence $\{k_n\} \subset \{k\}$ such that $[k_n] = j$ and $x^{k_n} \rightharpoonup \widehat{x}$. The fact that $P_{Q^{k_n}_{[k_n]}}(Ax^{k_n}) = P_{Q^{k_n}_j}(Ax^{k_n}) \in Q^{k_n}_j$ and the assumption (A3) indicate that

$$q_j(Ax^{k_n}) + \langle \eta^{k_n}_j, P_{Q^{k_n}_j}(Ax^{k_n}) - Ax^{k_n} \rangle \le 0, \tag{3.13}$$

where $\eta^{k_n}_j \in \partial q_j(Ax^{k_n})$. From boundedness assumption (A3) and (3.13), there exists a constant M_3 such that

$$\begin{aligned} q_j(Ax^{k_n}) &\le \|\eta^{k_n}_j\| \|P_{Q^{k_n}_j}(Ax^{k_n}) - Ax^{k_n}\| \\ &\le M_3 \|(I - P_{Q^{k_n}_j})Ax^{k_n}\| \to 0, n \to \infty, \end{aligned}$$

Using the weak lower semicontinuity of q_j, we have $q_j(A\widehat{x}) \le \liminf_{n\to\infty} q_j(Ax^{k_n}) \le 0$, $j = 1, 2, \cdots, r$, and thus $A\widehat{x} \in Q = \bigcap_{j=1}^r Q_j$. Hence $\widehat{x} \in S$. Using Lemma 2.2, we conclude that $\{x^k\}$ converges weakly to a solution of MSSFP (1.1). □

Remark 3.6. The assumption $r = t$ here is only for the simplicity of the proof. In fact, if $r \ne t$, the weakly convergence still holds.

4. Numerical Experiments

In this section, we present some numerical simulations to show the validity of Algorithm 3.1 and Algorithm 3.4. We compare our proposed iteration methods with the method proposed by Liu-Tang's Algorithm 2 in [16]. (Algorithm 2 is of the form $x^{k+1} = U_{[k]}(x^k - \alpha_k \sum_{j=1}^{r} \beta_j A^*(I - T_j)Ax)$, $\alpha_k \in (0, \frac{2}{\|A\|^2})$. We take $U_{[k]} = P_{C^k_{[k]}}$ and $T_j = P_{Q^k_j}$). For the sake of convenience, we denote $\mathbf{e}_0 = (0, 0, \cdots, 0)^T$ and $\mathbf{e}_1 = (1, 1, \cdots, 1)^T$, respectively. The codes are written in Matlab 2016a and run on Inter(R) Core(TM) i7-8700 CPU @ 3.20GHz 3.19GHz, RAM 8.00GB.

Example 4.1. [17] Take $\mathscr{H}_1 = \mathscr{H}_2 = \mathbf{R^3}$, $r = t = 2$, $\beta_1 = \beta_2 = \frac{1}{2}$, and $\alpha_k = \gamma l^{m_k}$ for all $k \geq 1$, $\gamma = 1$, $l = \frac{1}{2}$. Define

$$\begin{aligned} C_1 &= \left\{ x = (x_1, x_2, x_3)^T \in \mathbf{R^3} \mid x_1 + x_2^2 + 2x_3 \leq 0 \right\}, \\ C_2 &= \left\{ x = (x_1, x_2, x_3)^T \in \mathbf{R^3} \mid \frac{x_1^2}{16} + \frac{x_2^2}{9} + \frac{x_3^2}{4} - 1 \leq 0 \right\}, \\ Q_1 &= \left\{ x = (x_1, x_2, x_3)^T \in \mathbf{R^3} \mid x_1^2 + x_2 - x_3 \leq 0 \right\}, \\ Q_2 &= \left\{ x = (x_1, x_2, x_3)^T \in \mathbf{R^3} \mid \frac{x_1^2}{4} + \frac{x_2^2}{4} + \frac{x_3^2}{9} - 1 \leq 0 \right\}, \end{aligned}$$

and

$$A = \begin{pmatrix} 2 & -1 & 3 \\ 4 & 2 & 5 \\ 2 & 0 & 2 \end{pmatrix}.$$

The underlying MSSFP is to find an $x^* \in C_1 \bigcap C_2$ such that $Ax^* \in Q_1 \bigcap Q_2$.

Numerical results are shown in the tables for different initial points x^0, in which the k-th step iterate is denoted by $x^k = (x_1^k, x_2^k, x_3^k)^T$. We use $E_k = \|x^{k+1} - x^k\| / \|x^{k+1}\|$ to measure the error of the k-th iterate. If $E_k < 10^{-5}$, then the iteration process stops.

We choose initial value $x^0 = (0.1, 0.5, 0.4)^T$ for Algorithm 3.1 and initial value $x^0 = (0.2785, 0.547, 0.9575)^T$ for Algorithm 3.4. The convergence results of Algorithm 3.1 and Algorithm 3.4 are shown in Table 1 and Table 2, respectively.

Next, we compare the effectiveness and performance of Algorithm 3.1, 3.4 and Liu-Tang's Algorithm 2 with fixed step size, which is $\alpha_k = 0.01$ or 0.005. Note that these choices of α_k satisfy that $\alpha_k \in (0, \frac{2}{L})$, since we have $2/L \approx 0.0316$. The numerical results are listed in Table 3 under six kinds of choices of the initial value x^0 as follows.

Choice 1: $x^0 = (0, -3, -1)^T$;
Choice 2: $x^0 = (0.3685, 0.6256, 0.7802)^T$;
Choice 3: $x^0 = (0.4, 0.7, 1)^T$;
Choice 4: $x^0 = (1, 0, 1)^T$;
Choice 5: $x^0 = (-2, -5, -3.1)^T$
Choice 6: $x^0 = (0.123, 0.745, 0.789)^T$.

Table 1 and 2 convince that our algorithms are valid for solving the MSSFP (1.1). Table 3 asserts that our algorithms outperform Liu-Tang's Algorithm 2 in which we use fewer iterations and shorter CPU time to find approximate solutions with the same accuracy. Meanwhile, Algorithm 3.4 outperforms Algorithm 3.1 in the CPU time of iterations. This is mainly because

Table 1. $x^0 = (0.1, 0.5, 0.4)^T$ for Algorithm 3.1

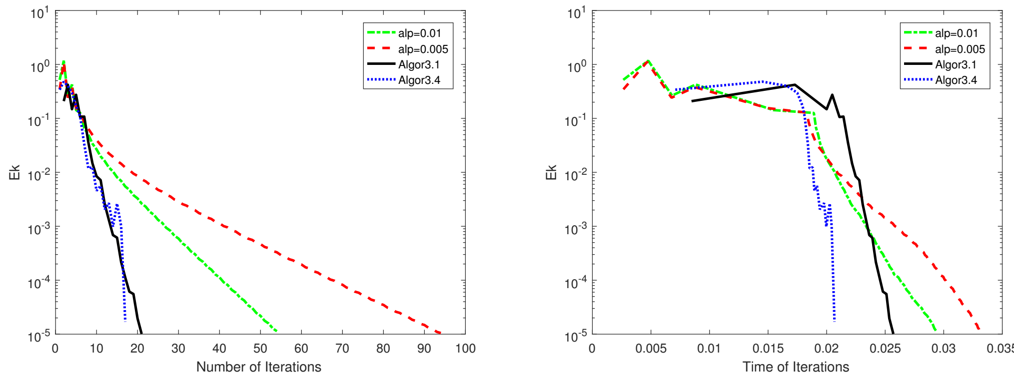
k	x_1^k	x_2^k	x_3^k	E_k
2	-0.0967	0.3559	0.0949	3.10E-01
12	-0.2457	0.1069	-0.1386	9.66E-02
18	-0.2807	0.0146	-0.1911	6.52E-02
24	-0.2838	-0.0325	-0.1972	5.46E-03
30	-0.2812	-0.0376	-0.1939	1.28E-03
37	-0.2810	-0.0383	-0.1938	2.32E-04
38	-0.2810	-0.0384	-0.1937	3.86E-05
40	-0.2810	-0.0384	-0.1937	3.04E-05
42	-0.2810	-0.0384	-0.1937	2.52E-05
43	-0.2809	-0.0384	-0.1936	8.03E-06

Table 2. $x^0 = (0.2785, 0.547, 0.9575)^T$ for Algorithm 3.4

k	x_1^k	x_2^k	x_3^k	E_k
2	-0.8431	-0.3607	-0.4428	5.51E-01
22	-0.3430	-0.1928	-0.0595	4.10E-02
35	-0.3426	-0.1833	-0.0745	6.99E-03
44	-0.3411	-0.1824	-0.0741	1.75E-03
50	-0.3411	-0.1821	-0.0744	7.37E-04
60	-0.3411	-0.1820	-0.0746	1.74E-04
65	-0.3411	-0.1819	-0.0747	9.21E-05
70	-0.3410	-0.1819	-0.0747	4.11E-05
79	-0.3411	-0.1819	-0.0747	1.22E-05
80	-0.3411	-0.1819	-0.0747	9.71E-06

Table 3. Comparison of Algorithm 3.1, 3.4 and the algorithm with fixed step size

		Algorithm 3.1	Algorithm 3.4	$\alpha_k \equiv 0.01$	$\alpha_k \equiv 0.005$
Choice 1	No. of Iter	22	18	55	95
	cpu(time)	0.0257	0.0206	0.0293	0.0331
Choice 2	No. of Iter	39	45	210	398
	cpu(time)	0.0342	0.0256	0.0443	0.0617
Choice 3	No. of Iter	35	67	203	381
	cpu(time)	0.0298	0.0266	0.0437	0.0586
Choice 4	No. of Iter	120	25	330	288
	cpu(time)	0.0432	0.0210	0.0564	0.0536
Choice 5	No. of Iter	23	28	47	62
	cpu(time)	0.0258	0.0216	0.0265	0.0293
Choice 6	No. of Iter	149	101	190	357
	cpu(time)	0.0415	0.0295	0.0422	0.0585



(a) Number of iterations for choice 1

(b) The CPU time for choice 1

FIGURE 1. Comparison Algorithm 3.1, 3.4 and Liu-Tang's Algorithm 2.

the former one is a cyclic algorithm and only have to calculate two projections at each iteration, while the latter one is a parallel algorithm and need to calculate all the projections onto Q_j^k at each iteration.

The asymptotic behavior of the error E_k is also compared for each choice of x^0. We only present the situation of choice 1 (see Figure 1) since the situations of the six initials are similar. From the figure, we see that the iterates generated by our algorithms converge faster.

Example 4.2. Take $\mathscr{H}_1 = \mathbf{R}^n$, $\mathscr{H}_2 = \mathbf{R}^m$, $A = (a_{ij})_{m\times n}$ with $a_{ij} \in [0,1]$ generated randomly, $C_i = \{x \in \mathbf{R}^n \mid \|x - d_i\|_2^2 \le r_i^2\}$, $i = 1, 2, \cdots, t$, $Q_j = \{y \in \mathbf{R}^m \mid \|y - l_j\|_1^2 \le h_j^2\}$, $j = 1, 2, \cdots, r$, where $d_i \in [\mathbf{e}_0, 10\mathbf{e}_1]$, $r_i \in [40, 60]$, $l_j \in [\mathbf{e}_0, \mathbf{e}_1]$, $h_j \in [10, 20]$ are all generated randomly. The parameters $\beta_1 = \beta_2 = \cdots = \beta_r = \frac{1}{r}$ and $\alpha_k = \gamma l^{m_k}$ for all $k \ge 1$, $\gamma = 1$, $l = \frac{1}{2}$, $\mu = \frac{1}{2}$.

We now study the number of iterations required and the CPU time of the sequence generated by Algorithm 3.1 and Algorithm 3.4. Meanwhile, we compare our algorithms and Liu-Tang's Algorithm 2 with fixed step size $\alpha_k = 0.1/\|A\|^2$. We randomly choose three initial points $x^0 = 5 * e_1$, $x^0 = 20 * e_1$, and $x^0 = 10 * rand(n,1)$ with m, n, r, t being different values. The asymptotic behavior of the three algorithms is listed in Table 4. The stopping criterion is defined by

$$E_k = \frac{1}{2}\sum_{i=1}^{t} \|x^k - P_{C_i^k} x^k\|^2 + \frac{1}{2}\sum_{j=1}^{r} \|Ax^k - P_{Q_j^k} Ax^k\|^2 < 10^{-5}.$$

From Table 4, we can see that our algorithms are better than the fixed step algorithm both in the number of iterations and in the CPU time. Meanwhile, we can obtain a conclusion similar with Example 4.1. Algorithm 3.1 outperforms Algorithm 3.4 in the number of iterations, and Algorithm 3.4 outperforms Algorithm 3.1 in the CPU time.

Acknowledgements

The authors were supported by the National Natural Science Foundation under Grant No. 61503385 and No. 11705279. The authors would also like to thank the editors and the reviewers for their constructive suggestions and comments, which greatly improved this paper.

Table 4. Comparison of Algorithm 3.1, 3.4 and Liu and Tang's Algorithm 2

Initial point		Algorithm 3.1	Algorithm 3.4	$\alpha_k = 0.1/\|A\|^2$
$r=t=10, m=15, n=20$				
$x^0 = 5*e_1$	No. of Iter	65	284	295
	cpu(time)	0.0881	0.0872	0.0914
$x^0 = 20*e_1$	No. of Iter	93	353	4573
	cpu(time)	0.1185	0.0965	1.3416
$x^0 = 10*rand(n,1)$	No. of Iter	109	370	2294
	cpu(time)	0.1281	0.1145	0.6858
$r=t=10, m=n=30$				
$x^0 = 5*e_1$	No. of Iter	128	520	5622
	cpu(time)	0.1859	0.1645	1.7816
$x^0 = 20*e_1$	No. of Iter	399	2264	10068
	cpu(time)	0.5823	0.6805	3.0494
$x^0 = 10*rand(n,1)$	No. of Iter	443	1071	11796
	cpu(time)	0.6773	0.3926	3.6772
$r=t=20, m=20, n=50$				
$x^0 = 5*e_1$	No. of Iter	652	3904	9694
	cpu(time)	1.7983	1.6779	5.4931
$x^0 = 20*e_1$	No. of Iter	259	680	9503
	cpu(time)	0.6826	0.3107	5.6021
$x^0 = 10*rand(n,1)$	No. of Iter	755	5314	6436
	cpu(time)	2.0892	2.1892	3.9810

REFERENCES

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

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

[3] C. Byrne, A unified treatment of some iterative algorithms in signal processing and image reconstruction, Inverse Probl. 20 (2004), 103-120.

[4] Q.Z. Yang, The relaxed CQ algorithm solving the split feasibility problem, Inverse Probl. 20 (2004), 1261-1266.

[5] H.K. Xu, Iterative methods for the split feasibility problem in infinite-dimensional Hilbert spaces, Inverse Probl. 26 (2010), Article ID 105018.

[6] Y. Censor, T. Elfving, N. Kopf, T. Bortfeld, The multiple-sets spilt feasibility problem and its applications for inverse problems, Inverse Probl. 21 (2005), 2071-2084.

[7] H.K. Xu, A variable Krasnosel'skii-Mann algorithm and the multiple-set split feasibility problem, Inverse Probl. 22(2006), 2021-2034.

[8] Q.Z. Yang, On variable-step relaxed projection algorithm for variational inequalities, J. Math. Anal. Appl. 302 (2005), 166-179.

[9] G. López, V. Martín-Márquez, F.W. Wang, H.K. Xu, Solving the split feasibility problem without prior knowledge of matrix norms, Inverse Probl. 28 (2012), 374-389.

[10] B. Qu, N.H. Xiu, A note on the CQ algorithm for the split feasibility problem, Inverse Probl. 21 (2005), 1655-1665.
[11] A. Gibali, L. W. Liu, Y.C. Tang, Note on the the modified relaxation CQ algorithm for the split feasibility problem, Optim. Lett. 12 (2018), 817-830.
[12] J.L. Zhao, Q.Z. Yang, Self-adaptive projection methods for the multiple-sets split feasibility problem, Inverse Probl. 27(2011), Article ID 035009.
[13] Y. Chen, Y. Guo, Y. Yu, R. Chen, Self-adaptive and relaxed self-adaptive projection methods for solving the multiple-set split feasibility problem, Abst. Appl. Anal. 2012 (2012), 958040.
[14] H.H. Bauschke, P.L. Combettes, Convex Analysis and Monotone Operator Theory in Hilbert Space, Springer, London, 2011.
[15] H.H. Bauschke, P.L. Combettes, A weak-to-strong convergence principle for Fejér-monntone methods in Hilbert spaces, Math. Oper. Res. 26 (2001), 248-264.
[16] L.W. Liu, Y.C. Tang, Several iterative algorithms for solving the split common fixed point problem of directed operators with applications, Optimization 65 (2016), 53-65.
[17] S.N. He, Z.Y. Zhao, B. Luo, A relaxed self-adaptive CQ algorithm for the multiple-sets split feasibility problem, Optimization, 64 (2015), 1907-1918.