Hindawi Publishing Corporation
Abstract and Applied Analysis
Volume 2012, Article ID 149508, 12 pages
doi:10.1155/2012/149508

*Research Article*

# An Extrapolated Iterative Algorithm for Multiple-Set Split Feasibility Problem

**Yazheng Dang[1,2] and Yan Gao[1]**

[1] *School of Management, University of Shanghai for Science and Technology, Shanghai 200093, China*
[2] *School of Mathematics and Information Science, Henan Polytechnic University, Jiaozuo 454000, China*

Correspondence should be addressed to Yazheng Dang, jgdyz@163.com

Received 29 December 2011; Revised 23 February 2012; Accepted 23 February 2012

Academic Editor: Khalida Inayat Noor



The multiple-set split feasibility problem (MSSFP), as a generalization of the split feasibility problem, is to find a point in the intersection of a family of closed convex sets in one space such that its image under a linear transformation will be in the intersection of another family of closed convex sets in the image space. Censor et al. (2005) proposed a method for solving the multiple-set split feasibility problem (MSSFP), whose efficiency depends heavily on the step size, a fixed constant related to the Lipschitz constant of $\nabla p(x)$ which may be slow. In this paper, we present an accelerated algorithm by introducing an extrapolated factor to solve the multiple-set split feasibility problem. The framework encompasses the algorithm presented by Censor et al. (2005). The convergence of the method is investigated, and numerical experiments are provided to illustrate the benefits of the extrapolation.

## 1. Introduction

The multiple-set split feasibility problem (MSSFP) is to find a point

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

where $t$ and $r$ are positive integers, $C_i \subset \Re^N, i = 1, \ldots, t$ and $Q_j \subset \Re^M, j = 1, \ldots, r$ are closed convex, $A$ is an $M \times N$ real matrix. When $t = r = 1$, the problem becomes to find a point $x \in C$ and $Ax \in Q$, which is just the two-set split feasibility problem (SFP, for short). SFP was originally introduced in [1] allowing for constraints both in the domain and range of a linear operator. Many methods have been developed for solving the SFP, for example, the basic CQ algorithm proposed by Byrne [2], the relaxed CQ algorithm presented by Yang [3] and the KM-CQ-like algorithm developed by Dang and Gao [4]. The MSSFP, formulated in [5], arises in the field of intensity-modulated radiation therapy when one attempts to describe physical does constraints and equivalent uniform does (EUD) constraints within a single model, see [6]. Censor et al. generalized the CQ algorithm [2] to solve the MSSFP [5] to get the following iterative process:

$$x^{k+1} = x^k - \frac{s}{L} \cdot \left( \sum_{i=1}^{t} \alpha_i \left( x^k - P_{C_i}\left(x^k\right)\right) + \sum_{j=1}^{r} \beta_j A^T \left( Ax^k - P_{Q_j}\left(Ax^k\right)\right)\right), \tag{1.2}$$

where $0 < s < 2$, $L = \sum_{i=1}^{t} \alpha_i + \rho(A^T A) \sum_{j=1}^{r} \beta_j$ and $\rho(A^T A)$ is the spectral radius of $A^T A$, and $\alpha_i > 0, \beta_j > 0$, for all $i$ and $j$ with $\sum_{i=1}^{t} \alpha_i + \sum_{j=1}^{r} \beta_j = 1$. Let $P_S$ denote the projection onto the convex set $S$, that is,

$$P_S x = \arg\min_{y \in S} \|x - y\|. \tag{1.3}$$

There also came out other algorithms for solving MSSFP, such that Xu in [7] and Masad and Reich in [8] introduced strong convergence methods in infinite dimensional Hilbert space, respectively. Censor et al. in [9] presented the perturbed projection and simultaneous subgradient projection algorithm to deal with the limit of accurately computing the orthogonal projection, Censor and Segal proposed string-averaging algorithmic scheme for sparse case in [10] and employed product space formulation to derive and analyze the simultaneous algorithm for MSSFP in [11]. However, the above algorithms use a fixed stepsize related to the largest eigenvalue of the matrix $A^T A$, which sometimes affects the convergence speed of the algorithms.

Extrapolated iterative method was first proposed in [12], it is an accelerated method in optimization since Pierra observed that the extrapolation parameter can be much larger than 1 and that the sequence generated by the extrapolated method converges fast. Subsequently, Heinz et al. in [13], proposed a general parallel block-iterative algorithmic framework by introducing extrapolated overrelaxations to solve the affine-convex feasibility problems, the corresponding numerical results also show the fast convergence.

Motivated by the extrapolated method for solving the affine-convex feasibility problems, in this paper, we present an extrapolated iterative method to solve the MSSFP, which includes the algorithm proposed by Censor et al. in [5]. As will be shown our algorithm extends and includes as a special case of the method in [5].

The paper is organized as follows. Section 2 reviews some preliminaries. Section 3 gives an extrapolated algorithm and shows its convergence. Section 4 provides some numerical experiments.

## 2. Preliminaries

Under normal circumstances, the MSSFP considers both the feasible and the infeasible cases by the use of a proximity function, that is, if the MSSFP problem is consistent then unconstrained minimization of the proximity function yields the value 0; in the inconsistent case, it finds a point which is least violating the feasibility by being "closest" to all sets, as "measured" by the proximity function. The minimization problem is

$$\min \frac{1}{2}\|x - P_C(x)\|^2 + \frac{1}{2}\|Ax - P_Q(Ax)\|^2. \tag{2.1}$$

We know that the projections of a point onto the sets $C$ and $Q$ are difficult to implement, even if each individual sets $C_i$ and $Q_j$ have simple or special structures such that projection onto each of them is easy to implement. In practical applications, the projections onto individual sets $C_i$ are more easily calculated than the projection onto the intersection $C$. For this purpose, Censor et al. [5] introduced the proximity function $p(x)$, to measure the distance of a point to all sets. We have

$$p(x) := \frac{1}{2}\sum_{i=1}^{t}\alpha_i\|P_{C_i}(x) - x\|^2 + \frac{1}{2}\sum_{j=1}^{r}\beta_j\left\|P_{Q_j}(Ax) - Ax\right\|^2, \tag{2.2}$$

where $\alpha_i > 0$, $\beta_j > 0$, for all $i$ and $j$ with $\sum_{i=1}^{t}\alpha_i + \sum_{j=1}^{r}\beta_j = 1$. Then,

$$\nabla p(x) = \sum_{i=1}^{t}\alpha_i\left(x^k - P_{C_i}\left(x^k\right)\right) + \sum_{j=1}^{r}\beta_j A^T\left(Ax^k - P_{Q_j}\left(Ax^k\right)\right), \tag{2.3}$$

Hence, (1.2) can be rewritten as

$$x^{k+1} = x^k - \frac{s \cdot 1}{L \cdot \nabla p(x^k)}. \tag{2.4}$$

The lemma provides well-known properties of orthogonal projections.

**Lemma 2.1** (see [14]). *Let $S$ be a nonempty closed convex subset of $\Re^N$, for any $x, y \in \Re^N$ and any $z \in S$, the following properties hold:*

(1) $x \in S \Leftrightarrow P_S(x) = x$,

(2) $\langle x - P_S(x), z - P_S(x)\rangle \le 0$,

(3) $\|P_S(x) - z\|^2 \le \|x - z\|^2 - \|P_S(x) - x\|^2$,

(4) $\|P_S(x) - P_S(y)\| \le \|x - y\|$.

## 3. The Extrapolated Projection Algorithm and Its Convergence

The following is our extrapolated projection algorithm.

*Algorithm 3.1.* For an arbitrary initial point $x^0$, $\{x^k\}_{k\ge 0}$ is generated by the iteration

$$x^{k+1} = x^k + s\max\left\{\frac{1}{L}, \lambda_k\right\}\left(\sum_{i=1}^{t}\alpha_i\left(P_{C_i}\left(x^k\right) - x^k\right) + \sum_{j=1}^{r}\beta_j A^T\left(P_{Q_j}\left(Ax^k\right) - Ax^k\right)\right), \tag{3.1}$$

where $s$ is a positive scalar such that $0 < s < 2$, $\alpha_i > 0, \beta_j > 0$ for all $i$ and $j$ with $\sum_{i=1}^{t} \alpha_i + \sum_{j=1}^{r} \beta_j = 1$, $L = \sum_{i=1}^{t} \alpha_i + \rho(A^T A) \sum_{j=1}^{r} \beta_j$ and $\rho(A^T A)$ being the spectral radius of $A^T A$,

$$\lambda_k = \frac{\sum_{i=1}^{t} \alpha_i \left\| (P_{C_i}(x^k) - x^k) \right\|^2 + \sum_{j=1}^{r} \beta_j \left\| \left(P_{Q_j}(Ax^k) - Ax^k\right) \right\|^2}{\left\| \sum_{i=1}^{t} \alpha_i (P_{C_i}(x^k) - x^k) + \sum_{j=1}^{r} \beta_j A^T \left(P_{Q_j}(Ax^k) - Ax^k\right) \right\|^2}. \tag{3.2}$$

Evidently, (3.1) happens to be (1.2), when $1/L > \lambda_k$.

Now we prove the convergence of the Algorithm 3.1.

**Theorem 3.2.** *Assume that the set of the solution of the multiple-sets split feasibility problem (MSSFP) is nonempty. Then, any sequence $\{x^k\}_{k=0}^{\infty}$ generated by Algorithm 3.1 converges to a solution of MSSFP* (1.1).

*Proof.* Let $h_k = \max\{1/\mathrm{L}, \lambda_k\}$ and take a point $z \in C$ with $Az \in Q$.

*Step 1.* First we show that $\|x^k - z\| \leq \|x^{k+1} - z\|$. From (3.1), we have

$$\begin{aligned}
\left\| x^{k+1} - z \right\|^2 &= \Bigg\| x^k + sh_k \left( \sum_{i=1}^{t} \alpha_i \left( P_{C_i}\left(x^k\right) - x^k \right) \right. \\
&\qquad \left. + \sum_{j=1}^{r} \beta_j A^T \left( P_{Q_j}\left(Ax^k\right) - Ax^k \right) - z \right\|^2 \\
&= \left\| x^k - z \right\|^2 + s^2 h_k^2 \Bigg\| \left( \sum_{i=1}^{t} \alpha_i \left( P_{C_i}\left(x^k\right) - x^k \right) \right) \\
&\qquad + \sum_{j=1}^{r} \beta_j A^T \left( P_{Q_j}\left(Ax^k\right) - Ax^k \right) \Bigg\|^2 \\
&\quad + 2sh_k \left\langle x^k - z, \sum_{i=1}^{t} \alpha_i \left( P_{C_i}\left(x^k\right) - x^k \right) \right\rangle \\
&\quad + 2sh_k \left\langle x^k - z, \sum_{j=1}^{r} \beta_j A^T \left( P_{Q_j}\left(Ax^k\right) - Ax^k \right) \right\rangle.
\end{aligned} \tag{3.3}$$

Observe that

$$\begin{aligned}
\left\langle x^k - z, \sum_{i=1}^{t} \alpha_i \left( P_{C_i}\left(x^k\right) - x^k \right) \right\rangle &= \sum_{i=1}^{t} \alpha_i \left\langle x^k - z, P_{C_i}\left(x^k\right) - x^k \right\rangle \\
&= \sum_{i=1}^{t} \alpha_i \left\langle x^k - P_{C_i}\left(x^k\right), P_{C_i}\left(x^k\right) - x^k \right\rangle \\
&\quad + \sum_{i=1}^{t} \alpha_i \left\langle P_{C_i}\left(x^k\right) - z, P_{C_i}\left(x^k\right) - x^k \right\rangle.
\end{aligned} \tag{3.4}$$

By the property (2) in Lemma 2.1, we get

$$\sum_{i=1}^{t}\alpha_i\left\langle P_{C_i}\left(x^k\right)-z,P_{C_i}\left(x^k\right)-x^k\right\rangle\leq 0. \tag{3.5}$$

Therefore,

$$\left\langle x^k-z,\sum_{i=1}^{t}\alpha_i\left(P_{C_i}\left(x^k\right)-x^k\right)\right\rangle\leq-\sum_{i=1}^{t}\alpha_i\left\|P_{C_i}\left(x^k\right)-x^k\right\|^2. \tag{3.6}$$

Similarly, we have

$$\begin{aligned}\left\langle x^k-z,\sum_{j=1}^{r}\beta_jA^T\left(P_{Q_j}\left(Ax^k\right)-Ax^k\right)\right\rangle&=\sum_{j=1}^{r}\beta_j\left\langle x^k-z,A^T\left(P_{Q_j}\left(Ax^k\right)-Ax^k\right)\right\rangle\\&=\sum_{j=1}^{r}\beta_j\left\langle Ax^k-Az,\left(P_{Q_j}\left(Ax^k\right)-Ax^k\right)\right\rangle.\end{aligned} \tag{3.7}$$

Since $Az\in Q$, using again property (2) in Lemma 2.1, we obtain that

$$\left\langle x^k-z,\sum_{j=1}^{r}\beta_jA^T\left(P_{Q_j}\left(Ax^k\right)-Ax^k\right)\right\rangle\leq-\sum_{j=1}^{r}\beta_j\left\|P_{Q_j}\left(Ax^k\right)-Ax^k\right\|^2. \tag{3.8}$$

Substituting (3.6) and (3.8) into (3.3), we get the following:

$$\begin{aligned}\left\|x^{k+1}-z\right\|^2\leq\left\|x^k-z\right\|^2&+s^2h_k^2\left\|\left(\sum_{i=1}^{t}\alpha_i\left(P_{C_i}\left(x^k\right)-x^k\right)\right)+\sum_{j=1}^{r}\beta_jA^T\left(P_{Q_j}\left(Ax^k\right)-Ax^k\right)\right\|^2\\&-2sh_k\sum_{i=1}^{t}\alpha_i\left\|P_{C_i}\left(x^k\right)-x^k\right\|^2-2sh_k\sum_{j=1}^{r}\beta_j\left\|P_{Q_j}\left(Ax^k\right)-Ax^k\right\|^2.\end{aligned} \tag{3.9}$$

Assume at $k$ th step, $1/L>\lambda_k$, then algorithms (3.1) and (2.4) coincide. Since $\nabla p(x)$ has a Lipschitz constant $L$, and $\nabla p(x)$ is $1/L$-ism (inverse-strongly monotone), that is,

$$\begin{aligned}\|\nabla p(x)-\nabla p(y)\|&\leq L\|x-y\|,\\\langle\nabla p(x)-\nabla p(y),x-y\rangle&\geq\frac{1}{L}\|\nabla p(x)-\nabla p(y)\|^2,\end{aligned} \tag{3.10}$$

see [5]; then, from the proof of Theorem 2.1 in [2], we get that the sequence $\{x^k\}$ generated by (1.2) satisfies

$$\left\|x^{k+1}-z\right\|^2 \leq \left\|x^k-z\right\|^2-s(2-s)\frac{\left\|\sum_{i=1}^{t}\alpha_i\left(P_{C_i}\left(x^k\right)-x^k\right)+\sum_{j=1}^{r}\beta_j A^T\left(P_{Q_j}\left(Ax^k\right)-Ax^k\right)\right\|^2}{L^2}. \tag{3.11}$$

Similarly, assume at $k$th step, $1/L \leq \lambda_k$, that is $h_k = \lambda_k$, then replacing $1/L$ with $\lambda_k$ in (3.9), we get the following:

$$\left\|x^{k+1}-z\right\|^2 \leq \left\|x^k-z\right\|^2-s(2-s)\frac{\left[\sum_{i=1}^{t}\alpha_i\left\|\left(P_{C_i}\left(x^k\right)-x^k\right)\right\|^2+\sum_{j=1}^{r}\beta_j\left\|\left(P_{Q_j}\left(Ax^k\right)-Ax^k\right)\right\|^2\right]^2}{\left\|\sum_{i=1}^{t}\alpha_i\left(P_{C_i}\left(x^k\right)-x^k\right)+\sum_{j=1}^{r}\beta_j A^T\left(P_{Q_j}\left(Ax^k\right)-Ax^k\right)\right\|^2}. \tag{3.12}$$

Combing (3.11) with (3.12)

$$\begin{aligned}\left\|x^{k+1}-z\right\|^2 \leq{} & \left\|x^k-z\right\|^2 \\ & -s(2-s)\max\left\{\frac{\left\|\sum_{i=1}^{t}\alpha_i\left(P_{C_i}\left(x^k\right)-x^k\right)+\sum_{j=1}^{r}\beta_j A^T\left(P_{Q_j}\left(Ax^k\right)-Ax^k\right)\right\|^2}{L^2},\right. \\ & \left.\frac{\left[\sum_{i=1}^{t}\alpha_i\left\|\left(P_{C_i}\left(x^k\right)-x^k\right)\right\|^2+\sum_{j=1}^{r}\beta_j\left\|\left(P_{Q_j}\left(Ax^k\right)-Ax^k\right)\right\|^2\right]^2}{\left\|\sum_{i=1}^{t}\alpha_i\left(P_{C_i}\left(x^k\right)-x^k\right)+\sum_{j=1}^{r}\beta_j A^T\left(P_{Q_j}\left(Ax^k\right)-Ax^k\right)\right\|^2}\right\}.\end{aligned} \tag{3.13}$$

Since $s \in (0,2)$, we have

$$\left\|x^{k+1}-z\right\| \leq \left\|x^k-z\right\| \tag{3.14}$$

for all $z \in C$ such that $Az \in Q$. Evidently, both $\{x^k\}$ and $\{\|x^k-z\|\}$ are bounded.

*Step 2.* Secondly we show that $\lim_{k\to\infty}x^k = x^*$ with $x^* \in C$ and $Ax^* \in Q$.

As shown in Step 1, the sequence $\{\|x^k-z\|\}$ is monotonically decreasing and bounded, and there exists the limit

$$\lim_{k\to\infty}\left\|x^k-z\right\| = d\ . \tag{3.15}$$

Since the case $h_k = 1/L$ is already treated in [5], we only need to consider the subsequence $\{x^{k_p}\}_{\lambda_{k_p}>1/L}$. Hence, we need to show that the subsequence $\{x^{k_p}\}$ converges to $x^*$ with $x^* \in C$ and $Ax^* \in Q$. From (3.12) and (3.15), and replacing $x^k$ by $x^{k_p}$, we have

$$\lim_{p\to\infty}\frac{\left[\sum_{i=1}^{t}\alpha_i\left\|\left(P_{C_i}\left(x^{k_p}\right)-x^{k_p}\right)\right\|^2+\sum_{j=1}^{r}\beta_j\left\|\left(P_{Q_j}\left(Ax^{k_p}\right)-Ax^{k_p}\right)\right\|^2\right]^2}{\left\|\sum_{i=1}^{t}\alpha_i\left(P_{C_i}\left(x^{k_p}\right)-x^{k_p}\right)+\sum_{j=1}^{r}\beta_jA^T\left(P_{Q_j}\left(Ax^{k_p}\right)-Ax^{k_p}\right)\right\|^2}=0 \tag{3.16}$$

From (3) in Lemma 2.1, we know that $\|P_{C_i}(x^{k_p})-x^{k_p})\| \leq \|x^{k_p}-z\|$ and $\|P_{Q_j}(Ax^{k_p})-Ax^{k_p})\| \leq \|Ax^{k_p}-Az\|$, then, we may assume that there exists a constant $M$ such that

$$\left\|\sum_{i=1}^{t}\alpha_i\left(P_{C_i}\left(x^{k_p}\right)-x^{k_p}\right)+\sum_{j=1}^{r}\beta_jA^T\left(P_{Q_j}\left(Ax^{k_p}\right)-Ax^{k_p}\right)\right\|^2 \leq M. \tag{3.17}$$

Therefore,

$$\begin{aligned}&\frac{\left[\sum_{i=1}^{t}\alpha_i\left\|\left(P_{C_i}\left(x^{k_p}\right)-x^{k_p}\right)\right\|^2+\sum_{j=1}^{r}\beta_j\left\|\left(P_{Q_j}\left(Ax^{k_p}\right)-Ax^{k_p}\right)\right\|^2\right]^2}{\left\|\sum_{i=1}^{t}\alpha_i\left(P_{C_i}\left(x^{k_p}\right)-x^{k_p}\right)+\sum_{j=1}^{r}\beta_j\left(P_{Q_j}\left(Ax^{k_p}\right)-Ax^{k_p}\right)\right\|^2}\\&\quad\geq\frac{1}{M}\left(\sum_{i=1}^{t}\alpha_i\left\|\left(P_{C_i}\left(x^{k_p}\right)-x^{k_p}\right)\right\|^2+\sum_{j=1}^{r}\beta_j\left\|\left(P_{Q_j}\left(Ax^{k_p}\right)-Ax^{k_p}\right)\right\|^2\right)^2\geq 0.\end{aligned} \tag{3.18}$$

Taking limits as $p \to \infty$ in (3.18) and considering (3.16) lead to

$$\lim_{p\to\infty}\sum_{i=1}^{t}\alpha_i\left\|P_{C_i}\left(x^{k_p}\right)-x^{k_p}\right\|^2+\sum_{j=1}^{r}\beta_j\left\|P_{Q_j}\left(Ax^{k_p}\right)-Ax^{k_p}\right\|^2=0, \tag{3.19}$$

this implies that

$$\begin{aligned}&\lim_{p\to\infty}\left\|P_{C_i}\left(x^{k_p}\right)-x^{k_p}\right\|=0,\quad i=1,\ldots,t\ ,\\&\lim_{p\to\infty}\left\|P_{Q_j}\left(Ax^{k_p}\right)-Ax^{k_p}\right\|=0,\quad j=1,\ldots,r.\end{aligned} \tag{3.20}$$

Since the sequence $\{x^{k_p}\}$ is bounded, there exists a subsequence $\{x_l^{k_p}\}$ of $\{x^{k_p}\}$ which converges to a point $b$, and a corresponding subsequence $\{Ax_l^{k_p}\}$ of $\{Ax^{k_p}\}$ which converges to a point $Ab$. Therefore, from (3.20), it is easy to get that $b \in C$ and $Ab \in Q$.

To obtain the result that the sequence $\{x^{k_p}\}$ itself is convergent to a point $b \in C$ with $Ab \in Q$, it is now sufficient to show that the subsequence of $\{x^{k_p}\}$ converges to the same point $b$ and the corresponding subsequence of $\{Ax^{k_p}\}$ is convergent to $Ab$. Let us suppose that there exists a subsequence $\{x_{l'}^{k_p}\}$ of $\{x^{k_p}\}$ that is convergent to point $b'$, as above, $b' \in C$ and $Ab' \in Q$. For $l \in Z^+$, we obtain

$$\left\|x_l^{k_p} - b'\right\|^2 - \left\|x_l^{k_p} - b\right\|^2 = \left\langle x_l^{k_p} - b + b - b', x_l^{k_p} - b + b - b'\right\rangle - \left\|x_l^{k_p} - b\right\|^2, \tag{3.21}$$

which, after calculating the inner product, leads to

$$\left\|x_l^{k_p} - b'\right\|^2 - \left\|x_l^{k_p} - b\right\|^2 = 2\left\langle x_l^{k_p} - b, b - b'\right\rangle + \|b' - b\|^2. \tag{3.22}$$

Similarly, for $l' \in Z^+$, it is easily to obtain that

$$\left\|x_{l'}^{k_p} - b\right\|^2 - \left\|x_{l'}^{k_p} - b'\right\|^2 = 2\left\langle x_{l'}^{k_p} - b', b' - b\right\rangle + \|b - b'\|^2. \tag{3.23}$$

As remarked, the sequences $\{\|x^{k_p} - b\|\}_{p=1}^{+\infty}$ and $\{\|x^{k_p} - b'\|\}_{p=1}^{+\infty}$ are convergent to $d(b)$ and $d(b')$. In particular, we get the following:

$$\lim_{p \to +\infty}\left(\left\|x^{k_p} - b\right\| - \left\|x^{k_p} - b'\right\|\right) = d(b) - d(b'). \tag{3.24}$$

Taking the limits in (3.22) and (3.23), for $l \to +\infty$ and for $l' \to +\infty$, we deduce that

$$\begin{aligned} d(b')^2 - d(b)^2 &= 0 + \|b - b'\|^2, \\ d(b)^2 - d(b')^2 &= 0 + \|b' - b\|^2, \end{aligned} \tag{3.25}$$

from above we conclude that $b = b'$. Similarly, $Ab = Ab'$. Hence, $\lim_{p\to\infty} x^{k_p} = b$ with $b \in C$ and $Ab \in Q$, that is, $\lim_{p\to\infty}\|x^{k_p} - b\| = 0$ with $b \in C$ and $Ab \in Q$. Replace $b$ with $x^*$, it can be written as $\lim_{p\to\infty}\|x^{k_p} - x^*\| = 0$ with $x^* \in C$ and $Ax^* \in Q$. And by reason of the monotonicity and boundness of the sequence $\{\|x^k - x^*\|\}$, we get the result.

Here we shortly explain the rational for the choice of the parameter $\lambda_k$ in Algorithm 3.1. In fact, if $s = 1$, (3.9) can be rewritten as

$$\begin{aligned} \left\|x^k - z\right\|^2 - \left\|x^{k+1} - z\right\|^2 \geq{}& 2h_k\sum_{i=1}^{t}\alpha_i\left\|P_{C_i}(x^k) - x^k\right\|^2 + 2h_k\sum_{j=1}^{r}\beta_j\left\|P_{Q_j}(Ax^k) - Ax^k\right\|^2 \\ &- h_k^2\left\|\left(\sum_{i=1}^{t}\alpha_i\left(P_{C_i}\left(x^k\right) - x^k\right)\right) - \sum_{j=1}^{r}\beta_j A^T\left(P_{Q_j}\left(Ax^k\right) - Ax^k\right)\right\|^2. \end{aligned} \tag{3.26}$$

Evidently, when

$$h_k = \frac{\sum_{i=1}^{t} \alpha_i \left\| P_{C_i}(x^k) - x^k \right\|^2 + \sum_{j=1}^{r} \beta_j \left\| P_{Q_j}(Ax^k) - Ax^k \right\|^2}{\left\| \left( \sum_{i=1}^{t} \alpha_i (P_{C_i}(x^k) - x^k) - \sum_{j=1}^{r} \beta_j A^T \left( P_{Q_j}(Ax^k) - Ax^k \right) \right) \right\|^2}, \tag{3.27}$$

the maximal value of the right hand side expression of (3.26) is obtained. Hence, if $s = 1$, for the case $\lambda_k > 1/L$, the factor $\lambda_k$ can be considered as the "best" possible value which assures that $x^{k+1}$ as the "closest" point to the set of solution of MSSFP along the direction $\sum_{i=1}^{t} \alpha_i (P_{C_i}(x^k) - x^k) - \sum_{j=1}^{r} \beta_j A^T (P_{Q_j}(Ax^k) - Ax^k)$. Therefore, to some extent, the extrapolated factor $\lambda_k$ plays an important role for the accelerated convergence for Algorithm 3.1.

□

## 4. Numerical Experiments

In the numerical results listed in the following table CPU time in seconds. We denote that $e_0 = (0, 0, \ldots, 0) \in \Re^N$ and $e_1 = (1, 1, \ldots, 1) \in \Re^N$. "Algorithm (1.2)" in the tables denotes the projection algorithm developed by Censor et al., in [5] as (1.2). "Algorithm 3.1" in the tables denotes Algorithm 3.1.

Now we give the following examples to test the efficiency of the above algorithm.

*Example 4.1.* In this example, we considered the multiples-set split feasibility problem, where

$$\begin{gathered} A = \begin{bmatrix} 2 & -1 & 3 & 2 & 3 \\ 1 & 2 & 5 & 2 & 1 \\ 2 & 0 & 2 & 1 & -2 \\ 2 & -1 & 0 & -3 & 5 \end{bmatrix}, \\ C_1 = \left\{ x \in \Re^5 \mid x_1 + x_2 \le 0.25 \right\}, \\ C_2 = \left\{ x \in \Re^5 \mid x_2 + x_3 \le 0.25 \right\}, \\ C_3 = \left\{ x \in \Re^5 \mid x_3 + x_4 \le 0.25 \right\}, \\ C_4 = \left\{ x \in \Re^5 \mid x_4 + x_5 \le 0.25 \right\}, \\ C_5 = \left\{ x \in \Re^5 \mid x_1 + x_5 \le 0.25 \right\}, \end{gathered} \tag{4.1}$$

and $Q = \{x \in \Re^4 \mid x \le d\}$ with $d = (1, 1, 1, 1)$. Consider the following three cases:

*Case I.* $x^0 = (1, -1, 1, -1, 1)$,

*Case II.* $x^0 = (1, 1, 1, 1, 1)$,

*Case III.* $x^0 = (10, 0, 10, 0, 10)$.

The number of iterative step needed for Algorithm (1.2) and Algorithm 3.1, and the corresponding solutions of this example are shown in Table 1.

**Table 1:** The numerical results of Example 4.1.

| Case | Algorithm (1.2) $s = 1$ | Algorithm 3.1 $s = 1$ | Algorithm (1.2) $s = 0.6$ | Algorithm 3.1 $s = 0.6$ | Algorithm (1.2) $s = 1.6$ | Algorithm 3.1 $s = 1.6$ |
|---|---|---|---|---|---|---|
| I | Iter. = 85 | Iter. = 3 | Iter. = 143 | Iter. = 9 | Iter. = 52 | Iter. = 2 |
| | Sec. = 0.123 | Sec. = 0.002 | Sec. = 0.210 | Sec. = 0.006 | Sec. = 0.099 | Sec. = 0.001 |
| | $x^* = (0.0781$ | $x^* = (0.1149$ | $x^* = (0.0765$ | $x^* = (0.0863$ | $x^* = (0.0809$ | $x^* = (-0.2996$ |
| | −0.6930 | −0.7321 | −0.6912 | −0.7045 | −0.6958 | −0.6310 |
| | 0.4143 | 0.3215 | 0.4175 | 0.3868 | 0.4095 | −0.0882 |
| | −0.6005 | −0.6893 | −0.5997 | −0.6248 | −0.6051 | −0.6830 |
| | −0.3276) | −0.4082) | −0.3249) | −0.3483) | −0.3321) | −0.9525) |
| II | Iter. = 658 | Iter. = 4 | Iter. = 1096 | Iter. = 8 | Iter. = 411 | Iter. = 2 |
| | Sec. = 0.890 | Sec. = 0.003 | Sec. = 0.996 | Sec. = 0.019 | Sec. = 0.633 | Sec. = 0.002 |
| | $x^* = (-0.0289$ | $x^* = (-0.1147$ | $x^* = (-0.0267$ | $x^* = (-0.0607$ | $x^* = (-0.0324$ | $x^* = (-0.3398$ |
| | 0.3333 | 0.3647 | 0.3314 | 0.3399 | 0.3367 | 0.3019 |
| | −0.3736 | −0.5197 | −0.3715 | −0.4232 | −0.3771 | −1.1325 |
| | 0.2065 | 0.2310 | 0.2059 | 0.2142 | 0.2077 | 0.0975 |
| | 0.0682) | 0.0115) | 0.0688) | 0.0467) | 0.0670) | −0.1164) |
| III | Iter. = 774 | Iter. = 5 | Iter. = 1288 | Iter. = 11 | Iter. = 484 | Iter. = 1 |
| | Sec. = 0.910 | Sec. = 0.011 | Sec. = 1.231 | Sec. = 0.022 | Sec. = 0.810 | Sec. = 0.004 |
| | $x^* = (0.5447$ | $x^* = (0.7013$ | $x^* = (0.5376$ | $x^* = (0.5206$ | $x^* = (0.5572$ | $x^* = (-1.2386$ |
| | −0.2349 | −0.4513 | −0.2277 | −0.2120 | −0.2474 | 0.0067 |
| | −0.7627 | −1.4225 | −0.7437 | −1.0221 | −0.7953 | −6.9419 |
| | −0.9891 | −1.4560 | −0.9679 | −1.1625 | −1.0249 | −3.1678 |
| | −0.7520) | −1.3338) | −0.7343) | −0.8719) | −0.7822) | −6.8881) |

**Table 2:** The numerical results of Example 4.2.

| $N$ | $t, r$ | Algorithm (1.2) $s = 1$ | Algorithm 3.1 $s = 1$ | Algorithm (1.2) $s = 0.8$ | Algorithm 3.1 $s = 0.8$ | Algorithm (1.2) $s = 1.95$ | Algorithm 3.1 $s = 1.95$ |
|---|---|---|---|---|---|---|---|
| $N = 20$ | $t = 5$ | Iter. = 1944 | Iter. = 257 | Iter. = 2387 | Iter. = 303 | Iter. = 1076 | Iter. = 165 |
| | $r = 20$ | Sec. = 2.395 | Sec. = 0.313 | Sec. = 2.741 | Sec. = 0.352 | Sec. = 1.305 | Sec. = 0.294 |
| $N = 40$ | $t = 10$ | Iter. = 5494 | Iter. = 267 | Iter. = 6824 | Iter. = 314 | Iter. = 2901 | Iter. = 176 |
| | $r = 40$ | Sec. = 5.403 | Sec. = 0.340 | Sec. = 6.304 | Sec. = 0.396 | Sec. = 2.988 | Sec. = 0.345 |
| $N = 60$ | $t = 10$ | Iter. = 14352 | Iter. = 278 | Iter. = 17895 | Iter. = 326 | Iter. = 7445 | Iter. = 183 |
| | $r = 60$ | Sec. = 20.304 | Sec. = 0.452 | Sec. = 26.401 | Sec. = 0.620 | Sec. = 17.04 | Sec. = 0.370 |

*Example 4.2.* In this example, we consider a multiple-set split feasibility where $A = (a_{ij})_{N\times N} \in \Re^{N\times N}$, and $a_{ij} \in (0, 10)$ are generated randomly

$$\begin{aligned} C_i &= \left\{x \in \Re^N \mid \|x - ie_1\|^2 \le (38 + 2i)^2\right\}, \quad i = 1, \dots, t, \\ Q_j &= \left\{x \in \Re^N \mid (24 \le x_j \le 26\right\}, \quad j = 1, \dots, r(N). \end{aligned} \tag{4.2}$$

Take initial point $x^0 = e_0 \in \Re^N$, we test the algorithms with different values of $s, t, r$, and $N$, respectively, in different dimensional Euclidean space.The number of iterative step needed for Algorithm (1.2) and Algorithm 3.1 is displayed in Table 2.

**Table 3:** The numerical results of Example 4.3.

| Case | Algorithm (1.2) $s = 1$ | Algorithm 3.1 $s = 1$ | Algorithm (1.2) $s = 0.6$ | Algorithm 3.1 $s = 0.6$ | Algorithm (1.2) $s = 1.6$ | Algorithm 3.1 $s = 1.6$ |
|---|---|---|---|---|---|---|
| I | Iter. = 623323 | Iter. = 3 | Iter. = 1038874 | Iter. = 48 | Iter. = 389576 | Iter. = 2 |
| | Sec. = 69.210 | Sec. = 0.013 | Sec. = 200.546 | Sec. = 0.021 | Sec. = 40.087 | Sec. = 0.010 |
| | $x^* = (0.1550$ | $x^* = (0.1250$ | $x^* = (0.1550$ | $x^* = (0.1250$ | $x^* = (0.1550$ | $x^* = (-0.2099$ |
| | −1.1979 | −1.1980 | −1.1979 | −1.1989 | −1.1980 | −1.3168 |
| | 0.8021 | 0.8020 | 0.8021 | 0.8011 | 0.8020 | 0.6832 |
| | −1.1979 | −1.1980 | −1.1979 | −1.1989 | −1.1980 | −1.3168 |
| | 0.1550) | 0.1250) | 0.1550) | 0.1250) | 0.1550) | −0.2099) |
| II | Iter. = 33 | Iter. = 2 | Iter. = 58 | Iter. = 47 | Iter. = 19 | Iter. = 1 |
| | Sec. = 0.069 | Sec. = 0.034 | Sec. = 0.096 | Sec. = 0.081 | Sec. = 0.033 | Sec. = 0.011 |
| | $x^* = (0.0021$ | $x^* = (0.020$ | $x^* = (0.0021$ | $x^* = (0.0020$ | $x^* = (0.0021$ | $x^* = (-0.5968$ |
| | 0.0021 | 0.0020 | 0.0021 | 0.0020 | 0.0021 | −0.5968 |
| | 0.0021 | 0.0020 | 0.0021 | 0.0020 | 0.0021 | −0.5968 |
| | 0.0021 | 0.0020 | 0.0021 | 0.0020 | 0.0021 | −0.5968 |
| | 0.0021) | 0.0020) | 0.0021) | 0.0020) | 0.0021) | −0.5968) |
| III | Iter. = 972361 | Iter. = 4 | Iter. = 1620605 | Iter. = 52 | Iter. = 607724 | Iter. = 2 |
| | Sec. = 165.303 | Sec. = 0.084 | Sec. = 280.564 | Sec. = 0.115 | Sec. = 79.910 | Sec. = 0.071 |
| | $x^* = (0.1550$ | $x^* = (0.1250$ | $x^* = (0.1550$ | $x^* = (0.1250$ | $x^* = (0.1550$ | $x^* = (-0.0419$ |
| | −5.9977 | −6.3782 | −5.9976 | −6.0071 | −5.9977 | −9.5967 |
| | 4.0023 | 3.6218 | 4.0024 | 3.9929 | 4.0023 | 0.4033 |
| | −5.9977 | −6.3782 | −5.9976 | −6.0071 | −5.9977 | −9.5967 |
| | 0.1550) | 0.1250) | 0.1550) | 0.1250) | 0.1550) | −0.0419) |

*Example 4.3.* In this example, we considered the multiples-set split feasibility problem, where

$$A = \begin{bmatrix} 100 & 100 & 100 & 100 & 100 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix},$$
$$C_1 = \left\{x \in \Re^5 \mid x_1 + x_2 \leq 0.25\right\};$$
$$C_2 = \left\{x \in \Re^5 \mid x_2 + x_3 \leq 0.25\right\},$$
$$C_3 = \left\{x \in \Re^5 \mid x_3 + x_4 \leq 0.25\right\}, \quad (4.3)$$
$$C_4 = \left\{x \in \Re^5 \mid x_4 + x_5 \leq 0.25\right\},$$
$$C_5 = \left\{x \in \Re^5 \mid x_1 + x_5 \leq 0.25\right\},$$

and $Q = \{x \in \Re^4 \mid x \leq d\}$ with $d = (1, 1, 1, 1)$. Consider the following three cases:

*Case I.* $x^0 = (1, -1, 1, -1, 1)$,

*Case II.* $x^0 = (1, 1, 1, 1, 1)$,

*Case III.* $x^0 = (10, 0, 10, 0, 10)$.

The number of iterative step needed for Algorithm (1.2) and Algorithm 3.1, and the corresponding solutions of this example are shown in Table 3.

In all numerical experiments, we take the weights as $\alpha_i = \beta_j = 1/(r+t), i = 1, \ldots, t$, $j = 1, \ldots, r$. The stopping criterion is $p(x) < \varepsilon = 10^{-4}$.

From these preliminary numerical results, we can see that the method is efficient, while the computational burden is not too large using the extrapolated technique.

## Acknowledgments

This work was supported by National Science Foundation of China (under Grant 111712210), Shanghai Municipal Committee of Science and Technology (under Grant 10550500800), Shanghai Municipal Government (under Grant S30501), and the Innovation Fund Project for Graduate Student of Shanghai (under Grant JWCXSL1001).

## References

[1] Y. Censor and T. Elfving, "A multiprojection algorithm using Bregman projections in a product space," *Numerical Algorithms*, vol. 8, no. 2–4, pp. 221–239, 1994.

[2] C. Byrne, "Iterative oblique projection onto convex sets and the split feasibility problem," *Inverse Problems*, vol. 18, no. 2, pp. 441–453, 2002.

[3] Q. Yang, "The relaxed CQ algorithm solving the split feasibility problem," *Inverse Problems*, vol. 20, no. 4, pp. 1261–1266, 2004.

[4] Y. Dang and Y. Gao, "The strong convergence of a KM-CQ-like algorithm for a split feasibility problem," *Inverse Problems*, vol. 27, no. 1, Article ID 015007, 9 pages, 2011.

[5] Y. Censor, T. Elfving, N. Kopf, and T. Bortfeld, "The multiple-sets split feasibility problem and its applications for inverse problems," *Inverse Problems*, vol. 21, no. 6, pp. 2071–2084, 2005.

[6] Y. Censor, T. Bortfeld, B. Martin, and A. Trofimov, "A unified approach for inversion problems in intensity-modulated radiation therapy," *Physics in Medicine and Biology*, vol. 51, no. 10, pp. 2353–2365, 2006.

[7] H.-K. Xu, "Krasnosel slii-Mann algorithm and the multiple-set split feasibility problem," *Inverse Problems*, vol. 22, no. 6, pp. 2021–2034, 2006.

[8] E. Masad and S. Reich, "A note on the multiple-set split convex feasibility problem in Hilbert space," *Journal of Nonlinear and Convex Analysis*, vol. 8, no. 3, pp. 367–371, 2007.

[9] Y. Censor, A. Motova, and A. Segal, "Perturbed projections and subgradient projections for the multiple-sets split feasibility problem," *Journal of Mathematical Analysis and Applications*, vol. 327, no. 2, pp. 1244–1256, 2007.

[10] Y. Censor and A. Segal, "Sparse string-averaging and split common fixed points," in *Nonlinear Analysis and Optimization I. Nonlinear Analysis*, vol. 513 of *Contemporary Mathematics Series*, pp. 125–142, American Mathematical Society, Providence, RI, USA, 2010.

[11] Y. Censor and A. Segal, "The split common fixed point problem for directed operators," *Journal of Convex Analysis*, vol. 16, no. 2, pp. 587–600, 2009.

[12] G. Pierra, "Decomposition through formalization in a product space," *Mathematical Programming*, vol. 28, no. 1, pp. 96–115, 1984.

[13] H. H. Bauschke, P. L. Combettes, and S. G. Kruk, "Extrapolation algorithm for affine-convex feasibility problems," *Numerical Algorithms*, vol. 41, no. 3, pp. 239–274, 2006.

[14] F. Facchinei and J.-S. Pang, *Finite-Dimensional Variational Inequalities and Complementarity Problems. Vol. I*, Springer Series in Operations Research, Springer, New York, NY, USA, 2003.