Yazheng Dang; Yan Gao

A new simultaneous subgradient projection algorithm for solving a multiple-sets split feasibility problem

Applications of Mathematics, Vol. 59 (2014), No. 1, 37–51

Persistent URL: http://dml.cz/dmlcz/143597

A NEW SIMULTANEOUS SUBGRADIENT PROJECTION ALGORITHM FOR SOLVING A MULTIPLE-SETS SPLIT FEASIBILITY PROBLEM

Yazheng Dang, Yan Gao, Shanghai

(Received September 26, 2011)

Abstract. In this paper, we present a simultaneous subgradient algorithm for solving the multiple-sets split feasibility problem. The algorithm employs two extrapolated factors in each iteration, which not only improves feasibility by eliminating the need to compute the Lipschitz constant, but also enhances flexibility due to applying variable step size. The convergence of the algorithm is proved under suitable conditions. Numerical results illustrate that the new algorithm has better convergence than the existing one.

Keywords: multiple-sets split feasibility problem; subgradient; extrapolated technique

MSC 2010: 90C30, 90C33

1. Introduction

The multiple-sets split feasibility problem (MSSFP), first introduced by Censor et al. in [4], 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 be in the intersection of another family of closed convex sets in the image space. In other words, the problem

This work was supported by the National Science Foundation of China under Grant 11171221, Shanghai Leading Academic Discipline Project under Grant XTKX2012, Basic and Frontier Research Program of Science and Technology Department of Henan Province under Grant 112300410277, Innovation Program of Shanghai Municipal Education Commission under Grant 14YZ094, Doctoral Program Foundation of Institutions of Higher Education of China under Grant 20123120110004, Doctoral Starting Projection of the University of Shanghai for Science and Technology under Grant ID-10-303-002, and Young Teacher Training Projection Program of Shanghai for Science and Technology.

is to find a point x such that

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

where t and r are positive integers, C_i, $i = 1, \dots, t$, and Q_j, $j = 1, \dots, r$, are closed convex subsets of spaces $\mathbb{R}^N$ and $\mathbb{R}^M$, and A is a linear bounded operator from $\mathbb{R}^N$ to $\mathbb{R}^M$. The MSSFP has broad applications such as image reconstruction [5], signal processing [3] and so on. When $t = 1$, $r = 1$, the problem is called the two-sets split feasibility problem (abbreviated as SFP). For solving the SFP or the MSSFP, many methods have been developed, for example, the CQ algorithm [2], the relaxed CQ algorithm [16] and the KM-CQ-like algorithm [11] for the SFP, the strong convergence methods in infinite dimensional Hilbert space [12], [15], the perturbed projections and simultaneous subgradient algorithm [6], the string-averaging algorithmic scheme [7] and the simultaneous algorithm [8] for the MSSFP. Now, we recall the outline of the primary algorithm proposed by Censor et al. [5] for solving the MSSFP. Define the proximity function $p(x)$ which measures the distance of a point to all sets

$$p(x) := \frac{1}{2}\sum_{i=1}^{t} \alpha_i \|P_{C_i}(x) - x\|^2 + \frac{1}{2}\sum_{j=1}^{r} \beta_j \|P_{Q_j}(Ax) - Ax\|^2, \tag{1.2}$$

where $\alpha_i > 0$, $\beta_j > 0$ for all i and j, $\sum\limits_{i=1}^{t} \alpha_i + \sum\limits_{j=1}^{r} \beta_j = 1$, and $P_S(x)$ denotes the projection of x onto the convex set S, that is,

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

With help of the proximity function (1.2), the MSSFP is converted to the constrained optimization problem

$$\min\{p(x) \mid x \in \Omega\}, \tag{1.4}$$

where $\Omega \subset \mathbb{R}^N$ is an auxiliary simple set. Censor et al. in [5] developed the following iterative formula by using the projection gradient method for solving the MSSFP:

$$x^{k+1} = P_\Omega(x^k - s\nabla p(x^k)), \tag{1.5}$$

where

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

$0 < s < 2/L$, L is the Lipschitz constant of $\nabla p(x)$ with $L = \sum\limits_{i=1}^{t} \alpha_i + \varrho(A^{\mathrm{T}}A) \sum\limits_{j=1}^{r} \beta_j$, $\varrho(A^{\mathrm{T}}A)$ is the spectral radius of $A^{\mathrm{T}}A$, $\sum\limits_{i=1}^{t} \alpha_i + \sum\limits_{j=1}^{r} \beta_j = 1$ with $\alpha_i > 0$, $\beta_j > 0$.

However, the above projection method (1.5) lacks feasibility and flexibility. On one hand, the choice of the step size s in (1.5) depends greatly on the Lipschitz constant of $\nabla p(x)$, while the Lipschitz constant may be difficult to estimate in many cases. On the other hand, if we know the Lipschitz constant, a method with fixed step size may be very slow. Extrapolation, as an accelerated technique, has been widely used to solve the convex feasibility problem [10], [12], [14], [13], [9]. It has been shown by Pierra in [14], [13] that the step-size contained extrapolated factor is variable and that the extrapolation parameter can be much larger than 1; this just gives an explanation for the acceleration. Patrick et al. in [9] proposed a parallel subgradient projection algorithm by introducing extrapolated over-relaxations to solve a convex set theoretic image recovery problem (a convex feasibility problem), which also showed fast convergence. Motivated by the extrapolated method for solving the convex feasibility problems, in this paper we propose a simultaneous subgradient projection algorithm to solve the MSSFP by using two extrapolated factors in one iterative step. (The "extrapolated factor" can guarantee the next iteration x^{k+1} is an intersection of a certain half line with a certain support hyperplane in the process of constructing a sequence $\{x^k\}_{k=0}^{\infty}$ in $\mathbb{R}^n$ with an algorithm which contains only extrapolated factors without other relaxed parameter, see [13, Lemma 1.2] for more details). Hence, to a certain extent, our algorithm improves the convergence and flexibility, and numerical results manifest the benefit of employing the extrapolated technique.

The paper is organized as follows. Section 2 reviews some preliminaries. In Section 3, the new simultaneous subgradient projection algorithm is proposed and its convergence is also shown. Section 4 gives some numerical experiments. Some conclusions are drawn in Section 5.

2. Preliminaries

It is well known that the projection operator P_S onto the convex S, for any $x \in \mathbb{R}^N$, is characterized by the following two inequalities:

(1) $\langle x - P_S(x), z - P_S(x)\rangle \leqslant 0$, $z \in S$;

(2) $\|P_S(x) - z\|^2 \leqslant \|x - z\|^2 - \|P_S(x) - x\|^2$, $z \in S$.

Now we recall some concepts, lemmas and basic results.

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

$$\partial f(x) = \{\xi \in \mathbb{R}^N \mid f(y) \geqslant f(x) + \langle \xi, y - x\rangle,\ \forall y \in \mathbb{R}^N\}. \tag{2.1}$$

An element of $\partial f(x)$ is said to be a subgradient.

Lemma 2.1 ([1])**.** *Let $f_i\colon \mathbb{R}^N \to \mathbb{R}$ be convex, $x^n \in \mathbb{R}^N$. Suppose $C_i = \{x \in \mathbb{R}^N \mid f_i(x) \leqslant 0\}$ is nonempty. For any $\xi^{i,n} \in \partial f_i(x^n)$, define the halfspace S_i by*

$$S_i := \{x \in \mathbb{R}^N \mid f_i(x^n) + \langle \xi^{i,n}, x - x^{i,n}\rangle \leqslant 0\}. \tag{2.2}$$

Then

(a) $C_i \subset S_i$;
(b) *if* $\xi^{i,n} \neq 0$, *then* S_i *is a halfspace; otherwise,* $S_i = \mathbb{R}^N$;
(c) $P_{S_i}(x^n) = x^n - (f_i^+(x^n)/\|\xi^{i,n}\|^2)\xi^{i,n}$ *where* $f_i^+(x^n) = \max\{f_i(x^n), 0\}$;
(d) $d(x^n, S_i) = f_i^+(x^n)/\|\xi^{i,n}\|$.

The importance of the halfspace defined in Lemma 2.1 is explained by the following. If we want to find a point in C_i, then if $f_i(x^n) \leqslant 0$, then x^n is such a point. Otherwise $f_i(x^n) > 0$, and it is usually "hard" to solve $f_i(x) = 0$. Hence, we consider a first-order approximation of f_i, i.e.,

$$f_i(x) \approx \tilde{f}_i(x) := f_i(x^n) + \langle \xi^{i,n}, x - x^n\rangle \quad \text{for some } \xi^{i,n} \in \partial f_i(x^n),$$

solve $\tilde{f}_i(x) = 0$, and we get a solution as follows:

$$P_{S_i}(x^n) = x^n - \frac{f_i(x^n)}{\|\xi^{i,n}\|^2}\xi^{i,n}.$$

In [6], Censor proposed the following simultaneous subgradient projection algorithm for MSSFP.

Algorithm 2.1. Initialization: Let x^0 be arbitrary.
Iterative step: Suppose x^k is the current iterative point and let

$$x^{k+1} = x^k + \frac{s}{L}\biggl(\sum_{i=1}^{t} \alpha_i (P_{C_{i,k}}(x^k) - x^k) + \sum_{j=1}^{r} \beta_j A^{\mathrm{T}}(P_{Q_{j,k}}(Ax^k) - Ax^k)\biggr).$$

Here $s \in (0,2)$, $L = \sum\limits_{i=1}^{t} \alpha_i + \varrho(A^{\mathrm{T}}A)\sum\limits_{j=1}^{r} \beta_j$, where $\varrho(A^{\mathrm{T}}A)$ is the spectral radius of $A^{\mathrm{T}}A$, $\alpha_i > 0$, $\beta_j > 0$ with $\sum\limits_{i=1}^{t} \alpha_i + \sum\limits_{j=1}^{r} \beta_j = 1$ and

$$C_{i,k} = \{x \in \mathbb{R}^N \mid c_i(x^k) + \langle \gamma^{i,k}, x - x^k\rangle \leqslant 0\}, \tag{2.3}$$

where $c_i\colon \mathbb{R}^N \to \mathbb{R}$ are convex for $i = 1, \ldots, t$, $\gamma^{i,k}$ is a subgradient of c_i at the point x^k, i.e., $\gamma^{i,k} \in \partial c_i(x^k)$.

$$Q_{j,k} = \{h \in \mathbb{R}^M \mid q_j(Ax^k) + \langle \eta^{j,k}, h - Ax^k\rangle \leqslant 0\}, \tag{2.4}$$

where $q_j\colon \mathbb{R}^M \to \mathbb{R}$ are convex for $j = 1, \ldots, r$, $\eta^{j,k} \in \partial q_j(Ax^k)$.

By the definition of the subgradient, it is clear that the halfspace $C_{i,k}$ contains C_i and the halfspace $Q_{j,k}$ contains Q_j. Due to the specific form of $C_{i,k}$ and $Q_{j,k}$, from Lemma 2.1 we know that the orthogonal projections onto $C_{i,k}$ and $Q_{j,k}$ may be computed directly.

3. Extrapolated subgradient projection algorithm and its convergence

The following is our extrapolated subgradient projection algorithm.

Algorithm 3.1. For an arbitrary initial point x^0, x^k is the current point. Select a parameter s such that $0 < s < 2\min\{\varrho(A^{\mathrm{T}}A)/(1+\varrho(A^{\mathrm{T}}A)),\ 1/(1+\varrho(A^{\mathrm{T}}A))\}$; the next iterative point is generated by

$$\begin{aligned} x^{k+1} = x^k &+ s\lambda_k \sum_{i=1}^{t} \alpha_i (P_{C_{i,k}}(x^k) - x^k) \\ &+ s\frac{1}{\varrho(A^{\mathrm{T}}A)} m_k \sum_{j=1}^{r} \beta_j A^{\mathrm{T}} (P_{Q_{j,k}}(Ax^k) - Ax^k), \end{aligned} \tag{3.1}$$

where

$$\lambda_k = \begin{cases} \dfrac{\sum_{i=1}^{t} \alpha_i \|P_{C_{i,k}}(x^k) - x^k\|^2}{\bigl\|\sum_{i=1}^{t} \alpha_i (P_{C_{i,k}}(x^k) - x^k)\bigr\|^2}, & \text{if } x^k \notin C, \\ 1, & \text{otherwise}; \end{cases} \tag{3.2}$$

$$m_k = \begin{cases} \dfrac{\sum_{j=1}^{r} \beta_j \|(P_{Q_{j,k}}(Ax^k) - Ax^k)\|^2}{\bigl\|\sum_{j=1}^{r} \beta_j (P_{Q_{j,k}}(Ax^k) - Ax^k)\bigr\|^2}, & \text{if } Ax^k \notin Q, \\ 1, & \text{otherwise}; \end{cases} \tag{3.3}$$

and $\alpha_i > 0$, $\beta_j > 0$ with $\sum\limits_{i=1}^{t} \alpha_i + \sum\limits_{j=1}^{r} \beta_j = 1$, $C_{i,k}, i = 1, \ldots, t$, and $Q_{j,k}$, $j = 1, \ldots, r$, are defined by (2.3) and (2.4).

We now discuss the convergence analysis of Algorithm 3.1.

Theorem 3.1. *Assume the set of the solutions of the multiple-sets split feasibility problem is nonempty and the subgradients of c_i, $i = 1, \ldots, r$, and q_j, $j = 1, \ldots, t$, are uniformly bounded. Then the sequence $\{x^k\}_{k=0}^{\infty}$ generated by Algorithm 3.1 converges to a solution of the multiple-sets split feasibility problem.*

P r o o f. Let z be a solution of MSSFP. Since $C_i \subset C_{i,k}$, we have $z = P_{C_i}(z) = P_{C_{i,k}}(z)$. First, we show that $\|x^{k+1} - z\| \leqslant \|x^k - z\|$ for all k. From (3.1) we have

$$\begin{aligned}
\|x^{k+1} - z\|^2 &= \biggl\| x^k + s\lambda_k \Bigl(\sum_{i=1}^{t} \alpha_i (P_{C_{i,k}}(x^k) - x^k) \Bigr) \\
&\quad + \frac{s}{\varrho(A^{\mathrm{T}}A)} m_k \Bigl(\sum_{j=1}^{r} \beta_j A^{\mathrm{T}} (P_{Q_{j,k}}(Ax^k) - Ax^k) \Bigr) - z \biggr\|^2 \\
&\leqslant \|x^k - z\|^2 + s^2 \lambda_k^2 \biggl\| \sum_{i=1}^{t} \alpha_i (P_{C_{i,k}}(x^k) - x^k) \biggr\|^2 \\
&\quad + \frac{s^2}{\varrho(A^{\mathrm{T}}A)} m_k^2 \biggl\| \sum_{j=1}^{r} \beta_j (P_{Q_{j,k}}(Ax^k) - Ax^k) \biggr\|^2 \\
&\quad + 2s\lambda_k \Bigl\langle x^k - z, \sum_{i=1}^{t} \alpha_i (P_{C_{i,k}}(x^k) - x^k) \Bigr\rangle \\
&\quad + \frac{2s}{\varrho(A^{\mathrm{T}}A)} m_k \Bigl\langle x^k - z, \sum_{j=1}^{r} \beta_j A^{\mathrm{T}} (P_{Q_{j,k}}(Ax^k) - Ax^k) \Bigr\rangle \\
&\quad + \frac{2s^2}{\varrho(A^{\mathrm{T}}A)} \Bigl\langle \lambda_k \Bigl(\sum_{i=1}^{t} \alpha_i (P_{C_{i,k}}(x^k) - x^k) \Bigr), \\
&\qquad\qquad m_k \Bigl(\sum_{j=1}^{r} \beta_j A^{\mathrm{T}} (P_{Q_{j,k}}(Ax^k) - Ax^k) \Bigr) \Bigr\rangle.
\end{aligned}$$

Obviously,

$$\begin{aligned}
&\frac{2s^2}{\varrho(A^{\mathrm{T}}A)} \Bigl\langle \lambda_k \Bigl(\sum_{i=1}^{t} \alpha_i (P_{C_{i,k}}(x^k) - x^k) \Bigr), m_k \Bigl(\sum_{j=1}^{r} \beta_j A^{\mathrm{T}} (P_{Q_{j,k}}(Ax^k) - Ax^k) \Bigr) \Bigr\rangle \\
&\qquad \leqslant \frac{s^2}{\varrho(A^{\mathrm{T}}A)} \lambda_k^2 \biggl\| \sum_{i=1}^{t} \alpha_i (P_{C_{i,k}}(x^k) - x^k) \biggr\|^2 \\
&\qquad\quad + s^2 m_k^2 \biggl\| \sum_{j=1}^{r} \beta_j (P_{Q_{j,k}}(Ax^k) - Ax^k) \biggr\|^2.
\end{aligned}$$

Hence,

$$\|x^{k+1}-z\|^2 \leqslant \|x^k-z\|^2 + \Big(1+\frac{1}{\varrho(A^{\mathrm T}A)}\Big)s^2\lambda_k^2\Big\|\sum_{i=1}^t \alpha_i(P_{C_{i,k}}(x^k)-x^k)\Big\|^2 + \Big(1+\frac{1}{\varrho(A^{\mathrm T}A)}\Big)s^2m_k^2\varrho(A^{\mathrm T}A)\Big\|\sum_{j=1}^r \beta_j(P_{Q_{j,k}}(Ax^k)-Ax^k)\Big\|^2 + 2s\lambda_k\Big\langle x^k-z,\sum_{i=1}^t \alpha_i(P_{C_{i,k}}(x^k)-x^k)\Big\rangle + \frac{2s}{\varrho(A^{\mathrm T}A)}m_k\Big\langle x^k-z,\sum_{j=1}^r \beta_j A^{\mathrm T}(P_{Q_{j,k}}(Ax^k)-Ax^k)\Big\rangle. \tag{3.4}$$

By the property of the projection (1), we get

$$\sum_{i=1}^t \alpha_i\langle P_{C_{i,k}}(x^k)-z, P_{C_{i,k}}(x^k)-x^k\rangle \leqslant 0.$$

Then

$$\Big\langle x^k-z,\sum_{i=1}^t \alpha_i(P_{C_{i,k}}(x^k)-x^k)\Big\rangle \leqslant -\sum_{i=1}^t \alpha_i\|P_{C_{i,k}}(x^k)-x^k\|^2. \tag{3.5}$$

Noticing that $Az\in Q\subset Q_{j,k}$, $j=1,\dots,r$, similarly to the above argument we obtain

$$\Big\langle x^k-z,\sum_{j=1}^r \beta_j A^{\mathrm T}(P_{Q_{j,k}}(Ax^k)-Ax^k)\Big\rangle \leqslant -\sum_{j=1}^r \beta_j\|P_{Q_{j,k}}(Ax^k)-Ax^k\|^2. \tag{3.6}$$

By (3.5) and (3.6), (3.4) reads

$$\|x^{k+1}-z\|^2 \leqslant \|x^k-z\|^2 + \Big(1+\frac{1}{\varrho(A^{\mathrm T}A)}\Big)s^2\lambda_k^2\Big\|\sum_{i=1}^t \alpha_i(P_{C_{i,k}}(x^k)-x^k)\Big\|^2 + \Big(1+\frac{1}{\varrho(A^{\mathrm T}A)}\Big)s^2m_k^2\Big\|\sum_{j=1}^r \beta_j(P_{Q_{j,k}}(Ax^k)-Ax^k)\Big\|^2 - 2s\lambda_k\sum_{i=1}^t \alpha_i\|P_{C_{i,k}}(x^k)-x^k\|^2 - \frac{2s}{\varrho(A^{\mathrm T}A)}m_k\sum_{j=1}^r \beta_j\|P_{Q_{j,k}}(Ax^k)-Ax^k\|^2. \tag{3.7}$$

When $x^k\notin C$, $Ax^k\notin Q$, we know that

$$\lambda_k=\frac{\sum_{i=1}^t \alpha_i\|P_{C_{i,k}}(x^k)-x^k\|^2}{\big\|\sum_{i=1}^t \alpha_i(P_{C_{i,k}}(x^k)-x^k)\big\|^2},\ m_k=\frac{\sum_{j=1}^r \beta_j\|(P_{Q_{j,k}}(Ax^k)-Ax^k)\|^2}{\big\|\sum_{j=1}^r \beta_j(P_{Q_{j,k}}(Ax^k)-Ax^k)\big\|^2}.$$

Substituting these values into the above inequality (3.7), we conclude that

$$\begin{aligned}(3.8)\quad \|x^{k+1}-z\|^2 &\leqslant \|x^k-z\|^2\\ &\quad - s\Big(2-\Big(1+\frac{1}{\varrho(A^{\mathrm T}A)}\Big)s\Big)\frac{\big(\sum_{i=1}^t \alpha_i\|P_{C_{i,k}}(x^k)-x^k\|^2\big)^2}{\big\|\sum_{i=1}^t \alpha_i(P_{C_{i,k}}(x^k)-x^k)\big\|^2}\\ &\quad - s\Big(\frac{2}{\varrho(A^{\mathrm T}A)}-\Big(1+\frac{1}{\varrho(A^{\mathrm T}A)}\Big)s\Big)\frac{\big(\sum_{j=1}^r \beta_j\|(P_{Q_{j,k}}(Ax^k)-Ax^k)\|^2\big)^2}{\big\|\sum_{j=1}^r \beta_j(P_{Q_{j,k}}(Ax^k)-Ax^k)\big\|^2}.\end{aligned}$$

When $x^k\in C$, $Ax^k\notin Q$, $\lambda_k=1$,

$$m_k=\frac{\sum_{j=1}^r \beta_j\|(P_{Q_{j,k}}(Ax^k)-Ax^k)\|^2}{\big\|\sum_{j=1}^r \beta_j(P_{Q_{j,k}}(Ax^k)-Ax^k)\big\|^2},$$

from (3.7) we get

$$\begin{aligned}(3.9)\quad \|x^{k+1}-z\|^2 &\leqslant \|x^k-z\|^2\\ &\quad - s\Big(\frac{2}{\varrho(A^{\mathrm T}A)}-\Big(1+\frac{1}{\varrho(A^{\mathrm T}A)}\Big)s\Big)\frac{\big(\sum_{j=1}^r \beta_j\|(P_{Q_{j,k}}(Ax^k)-Ax^k)\|^2\big)^2}{\big\|\sum_{j=1}^r \beta_j(P_{Q_{j,k}}(Ax^k)-Ax^k)\big\|^2}.\end{aligned}$$

Similarly, when $x^k\notin C$, $Ax^k\in Q$, we have

$$\begin{aligned}(3.10)\qquad \|x^{k+1}-z\|^2 &\leqslant \|x^k-z\|^2\\ &\quad - s\Big(2-\Big(1+\frac{1}{\varrho(A^{\mathrm T}A)}\Big)s\Big)\frac{\big(\sum_{i=1}^t \alpha_i\|P_{C_{i,k}}(x^k)-x^k\|^2\big)^2}{\big\|\sum_{i=1}^t \alpha_i(P_{C_{i,k}}(x^k)-x^k)\big\|^2}.\end{aligned}$$

From $0<s<2\min\{\varrho(A^{\mathrm T}A)/(1+\varrho(A^{\mathrm T}A)),1/(1+\varrho(A^{\mathrm T}A))\}$ and (3.8)–(3.10), we get

$$\|x^{k+1}-z\|<\|x^k-z\|,\ z\in C,\ Az\in Q.$$

Evidently, both $\{x^k\}$ and $\{\|x^k-z\|\}$ are bounded.

In what follows, we will prove that $\lim\limits_{k\to\infty} x^k=x^*$ with $x^*\in C$ and $Ax^*\in Q$. Since the sequence $\{\|x^k-z\|\}$ is monotonically decreasing and bounded, we may assume that it has a limit, i.e.

$$(3.11)\qquad \lim_{k\to\infty}\|x^k-z\|=d,$$

which combined with (3.8)–(3.10) implies

$$(3.12)\qquad \lim_{k\to\infty}\frac{\big(\sum_{i=1}^t \alpha_i\|P_{C_{i,k}}(x^k)-x^k\|^2\big)^2}{\big\|\sum_{i=1}^t \alpha_i(P_{C_{i,k}}(x^k)-x^k)\big\|^2}=0$$

and

$$\lim_{k\to\infty} \frac{\left(\sum_{j=1}^{r} \beta_j \|(P_{Q_{j,k}}(Ax^k) - Ax^k)\|^2\right)^2}{\left\| \sum_{j=1}^{r} \beta_j (P_{Q_{j,k}}(Ax^k) - Ax^k)\right\|^2} = 0. \tag{3.13}$$

By the property of projection (2), we get $\|P_{C_i}(x^k)-x^k\| \leqslant \|x^k-z\|$ and $\|P_{Q_j}(Ax^k)-Ax^k\| \leqslant \|Ax^k - Az\|$. Hence, we may assume that there exist two constants W, T such that

$$\left\| \sum_{i=1}^{t} \alpha_i (P_{C_i}(x^k) - x^k)\right\|^2 \leqslant W$$

and

$$\left\| \sum_{j=1}^{r} \beta_j A^{\mathrm{T}} (P_{Q_j}(Ax^k) - Ax^k)\right\|^2 \leqslant T.$$

Thus, we obtain

$$\frac{\left(\sum_{i=1}^{t} \alpha_i \|P_{C_{i,k}}(x^k) - x^k\|^2\right)^2}{\left\| \sum_{i=1}^{t} \alpha_i (P_{C_{i,k}}(x^k) - x^k)\right\|^2} \geqslant \frac{\left(\sum_{i=1}^{t} \alpha_i \|P_{C_{i,k}}(x^k) - x^k\|^2\right)^2}{W} \geqslant 0$$

and

$$\frac{\left(\sum_{j=1}^{r} \beta_j \|(P_{Q_{j,k}}(Ax^k) - Ax^k)\|^2\right)^2}{\left\| \sum_{j=1}^{r} \beta_j (P_{Q_{j,k}}(Ax^k) - Ax^k)\right\|^2} \geqslant \frac{\left(\sum_{j=1}^{r} \beta_j \|(P_{Q_{j,k}}(Ax^k) - Ax^k)\|^2\right)^2}{T} \geqslant 0.$$

Taking their limits as $k \to \infty$, due to (3.12) and (3.13) we conclude

$$\lim_{k\to\infty} \sum_{i=1}^{t} \alpha_i \|P_{C_{i,k}}(x^k) - x^k\|^2 = 0$$

and

$$\lim_{k\to\infty} \sum_{j=1}^{r} \beta_j \|P_{Q_{j,k}}(Ax^k) - Ax^k\|^2 = 0,$$

which implies

$$\lim_{k\to\infty} \|P_{C_{i,k}}(x^k) - x^k\| = 0, \quad i = 1, \ldots, t \tag{3.14}$$

and

$$\lim_{k\to\infty} \|P_{Q_{j,k}}(Ax^k) - Ax^k\| = 0, \quad j = 1, \ldots, r. \tag{3.15}$$

Altogether, Lemma 2.1, (3.14) and (3.15) result in

$$\lim_{k\to\infty} \frac{c_i^+(x^k)}{\|\gamma^{i,k}\|} = 0 \quad \text{for all } i = 1, \dots, t$$

and

$$\lim_{k\to\infty} \frac{q_j^+(Ax^k)}{\|\eta^{j,k}\|} = 0 \quad \text{for all } j = 1, \dots, r,$$

where $\gamma^{i,k} \in \partial c_i(x^k)$ and $\eta^{j,k} \in \partial q_j(Ax^k)$.
By the assumption of uniform boundedness of the subgradients, it is easy to get

$$\lim_{k\to\infty} c_i^+(x^k) = 0 \quad \text{for all } i = 1, \dots, t \tag{3.16}$$

and

$$\lim_{k\to\infty} q_j^+(Ax^k) = 0 \quad \text{for all } j = 1, \dots, r. \tag{3.17}$$

Since the sequence $\{x^k\}$ is bounded, we may assume that x^* is an accumulation point of $\{x^k\}$, Ax^* is an accumulation point of $\{Ax^k\}$. Let $\{x^{k_l}\}$ be a subsequence of $\{x^k\}$ and $\lim\limits_{l\to\infty} x^{k_l} = x^*$, and let $\{Ax^{k_l}\}$ be the corresponding subsequence of $\{Ax^k\}$ and $\lim\limits_{l\to\infty} Ax^{k_l} = Ax^*$. From (3.16) and (3.17) it is easy to see that

$$\lim_{k_l\to\infty} c_i^+(x^{k_l}) = 0 \quad \text{for all } i = 1, \dots, t$$

and

$$\lim_{k_l\to\infty} q_j^+(Ax^{k_l}) = 0 \quad \text{for all } j = 1, \dots, r.$$

By continuity of c_i, $i = 1, \dots, t$, we have $c_i^+(x^*) = 0$, $i = 1, \dots, t$, thus $x^* \in C$. By continuity of q_j, $j = 1, \dots, r$, we have $q_j^+(Ax^*) = 0$, $j = 1, \dots, r$, then $Ax^* \in Q$.

Replacing z by x^* in (3.11) leads to

$$\lim_{k\to\infty} \|x^k - x^*\| = d,$$

furthermore,

$$\lim_{k\to\infty} \|Ax^k - Ax^*\| = Ad.$$

Now

$$\lim_{l\to\infty} \|x^{k_l} - x^*\| = \lim_{l\to\infty} \|Ax^{k_l} - Ax^*\| = 0.$$

Therefore, $\lim\limits_{k\to\infty} \|x^k - x^*\| = \lim\limits_{k\to\infty} \|Ax^k - Ax^*\| = 0$. This completes the proof of the theorem.

4. Numerical experiments

The following Tables 1–2 list the numerical results of Example 4.1, Figures 1–2 show the numerical results of Example 4.2. We denote the number of iterations and the CPU time in seconds by "Iter.", "Sec.", respectively, and denote $e_0 = (0,0,\ldots,0) \in \mathbb{R}^N$ and $e_1 = (1,1,\ldots,1) \in \mathbb{R}^N$. For convenience, we choose $s = \alpha \min\{\varrho(A^{\mathrm{T}}A)/(1+\varrho(A^{\mathrm{T}}A)), 1/(1+\varrho(A^{\mathrm{T}}A))\}$ in Algorithm 3.1, and replace s by α in Algorithm 2.1; obviously, $\alpha \in (0,2)$, and we take the weights to be $1/(r+t)$ for both Algorithm 3.1 and Algorithm 2.1. The stopping criterion is $p(x) < \varepsilon = 10^{-4}$, where $p(x)$ is defined to be (1.2).

Example 4.1. We consider the following multiples-sets split feasibility problem, where

$$A = \begin{bmatrix} 2 & -1 & 3 & 2 & 3 \\ 1 & 2 & 5 & 2 & 1 \\ 2 & 0 & 2 & 1 & -2 \\ 2 & -1 & 0 & -3 & 5 \end{bmatrix}$$

and

$$\begin{aligned} C_1 &= \{x \in \mathbb{R}^5;\ x_1^2 + x_2^2 \leqslant 0.25\}; \\ C_2 &= \{x \in \mathbb{R}^5;\ x_2^2 + x_3^2 \leqslant 0.25\}; \\ C_3 &= \{x \in \mathbb{R}^5;\ x_3^2 + x_4^2 \leqslant 0.25\}; \\ C_4 &= \{x \in \mathbb{R}^5;\ x_4^2 + x_5^2 \leqslant 0.25\}; \\ C_5 &= \{x \in \mathbb{R}^5;\ x_1^2 + x_5^2 \leqslant 0.25\}, \end{aligned}$$

and $Q = \{x \in \mathbb{R}^4;\ x \leqslant d\}$ 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: $x^0 = (5,0,5,0,5)$.

The number of iterative steps needed, the CPU time in seconds, the corresponding solution x^* and the value of $p(x^*)$ for Example 4.1 for Cases I–III by Algorithm 2.1 and Algorithm 3.1 are shown in Tables 1–3, respectively.

From Tables 1–3, we find that for some of the above cases Algorithm 3.1 reaches the solution set in finitely many steps, while Algorithm 2.1 reaches the point needed with the desired error. Hence, Algorithm 3.1 has better convergence than Algorithm 2.1.

Example 4.2. We consider a multiple-set split feasibility problem where $A = (a_{ij})_{N\times N} \in \mathbb{R}^{N\times N}$, and $a_{ij} \in (0,1)$ are generated randomly:

$$\begin{aligned} C_i &= \{x \in \mathbb{R}^N;\ \|x - ie_1\|^2 \leqslant (10+2i)^2\}, \quad i = 1,2,\ldots,t; \\ Q_j &= \{x \in \mathbb{R}^N;\ (25-j)e_1 \leqslant x \leqslant (25+j)e_1\}, \quad j = 1,2,\ldots,r, \end{aligned}$$

Algorithm	Algo. 2.1	Algo. 3.1	Algo. 2.1	Algo. 3.1	Algo. 2.1	Algo. 3.1
α	1.0	1.0	0.6	0.6	1.6	1.6
Iter.	1399	47	2354	93	862	21
Sec.	0.864	0.083	1.371	0.101	0.799	0.052
x^*	(0.0973, -0.4201, 0.2962, -0.4521, -0.2355)	(0.0844, -0.4026, 0.2991, -0.4446, -0.2354)	(0.0968, -0.4192, 0.2968, -0.4519, -0.2350)	(0.0882, -0.4168, 0.2921, -0.4488, -0.2381)	(0.0981, -0.4214, 0.2952, -0.4523, -0.2364)	(0.0892, 0, -0.3318, 0.2603, -0.4513, -0.2493)
$p(x^*)$	$9.9912\cdot10^{-5}$	$6.7174\cdot10^{-5}$	$9.9905\cdot10^{-5}$	$7.3366\cdot10^{-5}$	$9.9858\cdot10^{-5}$	$3.6655\cdot10^{-5}$

Table 1. The numerical results of Example 4.1 for Case I.

Algorithm	Algo. 2.1	Algo. 3.1	Algo. 2.1	Algo. 3.1	Algo. 2.1	Algo. 3.1
α	1.0	1.0	0.6	0.6	1.6	1.6
Iter.	769	18	1283	43	480	11
Sec.	0.762	0.045	0.824	0.071	0.331	0.031
x^*	(0.3265, 0.4241, -0.3108, 0.3721 0.3294)	(0.3163, 0.3975, -0.3057, 0.3791 0.3156)	(0.3270, 0.4239, -0.3108, 0.3715 0.3204)	(0.3269, 0.4225, -0.3150, 0.3786 0.3193)	(0.3257, 0.4245, -0.3107, 0.3731 0.3203)	(0.3010, 0.3901, -0.3040, 0.3720 0.3188)
$p(x^*)$	$9.9968\cdot10^{-5}$	$3.3485\cdot10^{-5}$	$9.9848\cdot10^{-5}$	$9.9546\cdot10^{-5}$	$9.9988\cdot10^{-5}$	0

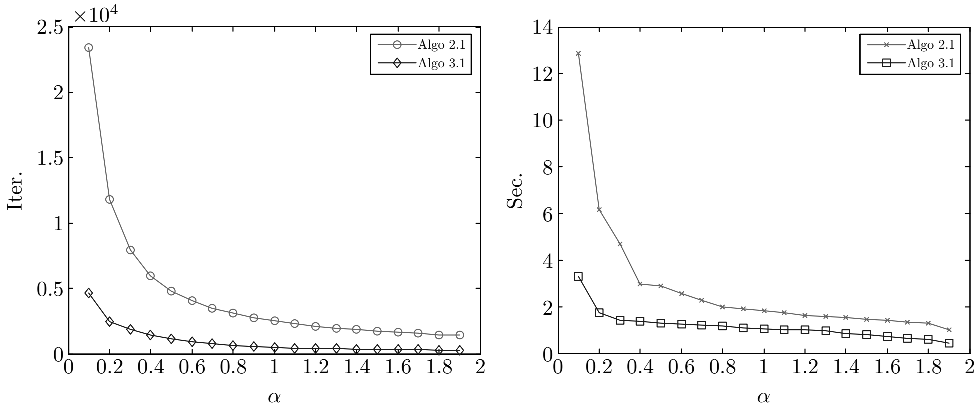
Table 2. The numerical results of Example 4.1 for Case II.

Algorithm	Algo. 2.1	Algo. 3.1	Algo. 2.1	Algo. 3.1	Algo. 2.1	Algo. 3.1
α	1.0	1.0	0.6	0.6	1.6	1.6
Iter.	724	15	1204	37	454	9
Sec.	0.732	0.039	0.811	0.071	0.310	0.025
x^*	(0.4768, 0.2394, -0.2368, -0.4246 -0.2266)	(0.4437, 0.2452, -0.3001, -0.4257 -0.2263)	(0.4770, 0.2401, -0.2329, -0.4215 -0.2248)	(0.4543, 0.2444, -0.3043, -0.4311 -0.2303)	(0.4765, 0.2383, -0.2428, -0.4287 -0.2290)	(0.4040, 0.2001 -0.2300, -0.4011 -0.2019)
$p(x^*)$	$9.9563\cdot10^{-5}$	$2.5864\cdot10^{-5}$	$9.9819\cdot10^{-5}$	$5.8547\cdot10^{-5}$	$9.9264\cdot10^{-5}$	0

Table 3. The numerical results of Example 4.1 for Case III.

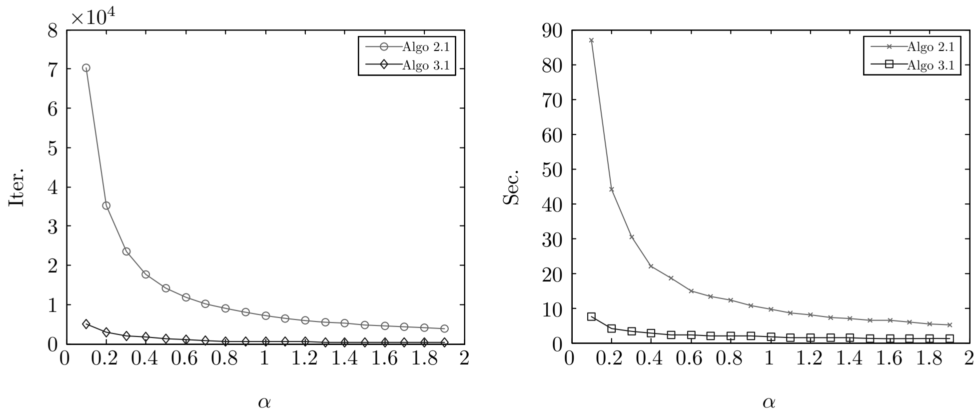
where $ie_1 \in \mathbb{R}^N$ is the center of the ball C_i, $10 + 2i$ is the radius of the ball C_i. We test the algorithms with different t and r in different dimensional Euclidean space. We take e_0 as the initial point in this example.

The number of iterative steps needed and the CPU time in seconds for Example 4.2 with $N = 20$, $t = 5$, $r = 5$ for different α by Algorithm 2.1 and Algorithm 3.1 are shown in Figure 1 (a) and Figure 1 (b), respectively. The number of iterative steps needed and the CPU time in seconds for Example 4.2 with $N = 40$, $t = 10$, $r = 15$



(a) (b)

Figure 1. The numerical results of Example 4.2 ($N = 20, t = 5, r = 5$) by Algorithm 2.1 and Algorithm 3.1.



(a) (b)

Figure 2. The numerical results of Example 4.2 ($N = 40$, $t = 10$, $r = 15$) by Algorithm 2.1 and Algorithm 3.1.

for different α by Algorithm 2.1 and Algorithm 3.1 are shown in Figure 2 (a) and Figure 2 (b), respectively.

From Figures 1–2, we see that the computational burden of Algorithm 3.1 is lighter than that of Algorithm 2.1, especially in higher dimensional space. This shows that using the extrapolated technique to solve the MSSFP in higher dimensions is promissing.

5. Concluding remarks

This paper presents a new simultaneous subgradient projection algorithm for the MSSFP. The advantage of Algorithm 3.1 over Algorithm 2.1 resides in its ability to use larger relaxations which means that Algorithm 3.1 converges in fewer iterations and that the computational cost of each iteration is lower due to not compute Lipschitz constant. On the other hand, Algorithm 3.1 is more flexible than Algorithm 2.1 in that it possesses a variable step size at each iteration.

A c k n o w l e d g m e n t. The authors are grateful to the anonymous referee for his/her valuable suggestions and comments.

References

[1] *H. H. Bauschke, J. M. Borwein*: On projection algorithms for solving convex feasibility problems. SIAM Rev. *38* (1996), 367–426.
[2] *C. Byrne*: Iterative oblique projection onto convex sets and the split feasibility problem. Inverse Probl. *18* (2002), 441–453.
[3] *Y. Censor, T. Bortfeld, B. Martin, A. Trofimov*: A unified approach for inversion problems in intensity-modulated radiation therapy. Physics in Medicine and Biology *51* (2006), 2353–2365.
[4] *Y. Censor, T. Elfving*: A multiprojection algorithm using Bregman projections in a product space. Numer. Algorithms *8* (1994), 221–239.
[5] *Y. Censor, T. Elfving, N. Kopf, T. Bortfeld*: The multiple-sets split feasibility problem and its applications for inverse problems. Inverse Probl. *21* (2005), 2071–2084.
[6] *Y. Censor, A. Motova, A. Segal*: Perturbed projections and subgradient projections for the multiple-sets split feasibility problem. J. Math. Anal. Appl. *327* (2007), 1244–1256.
[7] *Y. Censor, A. Segal*: Sparse string-averaging and split common fixed points. Nonlinear Analysis and Optimization I. Nonlinear Analysis. A conference in celebration of Alex Ioffe's 70th and Simeon Reich's 60th birthdays, Haifa, Israel, June 18–24, 2008 (A. Leizarowitz et al., eds.). Contemporary Mathematics 513, American Mathematical Society, Providence, 2010, pp. 125–142.
[8] *Y. Censor, A. Segal*: The split common fixed point problem for directed operators. J. Convex Anal. *16* (2009), 587–600.
[9] *P. L. Combettes*: Convex set theoretic image reconvery by extrapolated iterations of parallel subgradient projections. IEEE Transactions on Image Processing *6* (1997), 493–506.
[10] *Y. Dang, Y. Gao*: Non-monotonous accelerated parallel subgradient projection algorithm for convex feasibility problem. Optimization (electronic only) (2012).
[11] *Y. Dang, Y. Gao*: The strong convergence of a KM-CQ-like algorithm for a split feasibility problem. Inverse Probl. *27* (2011), Article ID 015007.
[12] *E. Masad, S. Reich*: A note on the multiple-set split convex feasibility problem in Hilbert space. J. Nonlinear Convex Anal. *8* (2007), 367–371.
[13] *G. Pierra*: Decomposition through formalization in a product space. Math. Program. *28* (1984), 96–115.
[14] *G. Pierra*: Parallel constraint decomposition for minimization of a quadratic form. Optimization Techniques. Modeling and Optimization in the Service of Man Part 2. Proceedings, 7th IFIP conference, Nice, September 8-12, 1975 (J. Cea, ed.). Lecture Notes in Computer Science 41, Springer, Berlin, 1976, pp. 200–218. (In French.)

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

[16] *Q. Yang*: The relaxed CQ algorithm solving the split feasibility problem. Inverse Probl. *20* (2004), 1261–1266.

Authors' addresses: *Yazheng Dang*, School of Management, University of Shanghai for Science and Technology, 200093, Shanghai, People's Republic of China, e-mail: `jgdyz@163.com`; *Yan Gao*, School of Management, University of Shanghai for Science and Technology, 200093, Shanghai, People's Republic of China, e-mail: `gaoyan@usst.edu.cn`.