TẠP CHÍ KHOA HỌC ĐẠI HỌC TÂN TRÀO
ISSN: 2354 - 1431
http://tckh.daihoctantrao.edu.vn/

ITERATIVE METHODS FOR SOLVING THE MULTIPLE-SETS SPLIT FEASIBILITY PROBLEM

Tran Thi Huong
Thai Nguyen Univesity of Technology,Viet Nam
Email address: tranthihuong@tnut.edu.vn
DOI: https://doi.org/10.51453/2354-1431/2022/741

Article info

Received:12/3/2022
Revised: 15/4/2022
Accepted: 01/6/2022

Keywords:
Multiple-sets split feasibility problem, nonexpansive mapping, fixed point, metric projection, iterative method

Abstract:

Recently, due to the influence of business management, the concepts of management and administration are used arbitrarily. This gives rise to many misunderstandings in management, leadership and administration. In many fields, there is a tendency to abuse the term administration to replace management. This needs to be seriously considered. This article discusses the nature and relationship between management and administration.

Vol8.No2_June 2022

TẠP CHÍ KHOA HỌC ĐẠI HỌC TÂN TRÀO
ISSN: 2354 - 1431
http://tckh.daihoctantrao.edu.vn/

PHƯƠNG PHÁP LẶP GIẢI BÀI TOÁN CHẤP NHẬN TÁCH ĐA TẬP

Trần Thị Hương
Đại học Kỹ thuật Công nghiệp, Việt Nam
Địa chỉ email: tranthihuong@tnut.edu.vn
DOI: https://doi.org/10.51453/2354-1431/2022/741

Thông tin bài viết

Ngày nhận bài: 12/03/2022
Ngày sửa bài: 15/04/2022
Ngày duyệt đăng: 01/06/2022

Từ khóa:

Bài toán chấp nhận tách đa tập, ánh xạ không giãn, điểm bất động, phép chiếu metric, phương pháp lặp.

Tóm tắt

Bài toán chấp nhận tách đa tập (MSSFP) được đưa ra đầu tiên bởi Censor và Elfving để mô hình hoá bài toán ngược trong khôi phục ảnh. Cho đến nay, có rất nhiều công trình liên quan đến phương pháp lặp để giải bài toán MSSFP và hầu hết các công trình đều sử dụng gradient của hàm xấp xỉ, đo khoảng cách từ một điểm đến các tập trong không gian ảnh để xây dựng phương pháp lặp đồng thời, lặp xoay vòng và các cải biên của chúng. Trong bài báo này, chúng tôi giới thiệu phương pháp tổng quát xây dựng thuật toán lặp giải bài toán MSSFP. Chúng tôi đưa ra sơ đồ thuật toán lặp có tham số lặp được chọn một cách thích nghi và đưa ra phiên bản nới lỏng của lược đồ bằng cách sử dụng phép chiếu lên nửa không gian thay vì chiếu lên những tập lồi thông thường. Cuối cùng là các ví dụ số minh họa cho các kết quả của chúng tôi.

1. Introduction

Let E^n and E^m be two real Euclidian spaces, n, m be positive integers, $\{C_i\}_{i\in I}$ and $\{Q_j\}_{j\in J}$ be two families of closed convex subsets in E^n and E^m, respectively, where $I = \{1, 2, ..., N\}$ and $J = \{1, 2, ..., M\}$ with any fixed positive integers N and M. Let A be an $m \times n$-matrix of real numbers. We use the symbols E, $\langle .;. \rangle$ and $\|.\|$ to denote the unit matrix, an inner product and a norm in any Euclidian space.

The MSSFP is to find a point

$$p \in C := \bigcap_{i\in I} C_i \text{ such that } A_p \in Q := \bigcap_{j\in J} Q_j \qquad (1.1)$$

This problem was first introduced by Censor and Elfving in 1994 [5] for modeling inverse problems that arise from phase retrievals and in image reconstruction [3], [4]. Recently, the MSSFP can also be used to model the intensity-modulated radiation therapy [7]-[10] and references therein. Denote by Γ the set of solution for (1.1). Throughout, this paper, we assume that $\Gamma \neq 0$.

For solving the split convex feasibility problem, that is (1.1) with $N = M = 1$, Byrne [3], [4] introduced a well-known iterative method, named CQ-method and defined by

$$x^{k+1} = P_C\left(E - \gamma A^T(E - P_Q)A\right)x^k, k \geq 1, \qquad (1.2)$$

with a fixed real number $\gamma \in \left(0; 2/\|A\|^2\right)$, where P_C and P_Q denote the metric projections on the sets C and Q, respectively, and A^T is the transpose of A.

In the case that $n = m$ and $A = E$ the MSSFP deduces to the convex feasibility problem (CFP), that is to find a point $p \in C$. To solve the CFP, Censor et al. [6] proposed a string-averaged algorithmic scheme in which the end-points of strings of sequential projections onto the constraints are averaged.

Recently, Nguyen Buong [1], [2] used properties of metric projections instead of the proximity function to construct a general scheme,

$$x^{k+1} = P_1\left(E - \gamma A^T (E - P_2)A\right)x^k,\ k \geq 1, \quad (1.3)$$

where the mappings P_1 and P_2 are defined by one of the following cases:

(i) $P_1 = \sum_{i=1}^{N} \beta_i P_{C_i}$ and $P_2 = \sum_{j=1}^{M} \eta_j P_{Q_j}$;

(ii) $P_1 = P_{C_1} ... P_{C_N}$ and $P_2 = \sum_{j=1}^{M} \eta_j P_{Q_j}$;

(iii) $P_1 = P_{C_1} ... P_{C_N}$ and $P_2 = P_{Q_1} ... P_{Q_M}$;

(iv) $P_1 = \sum_{i=1}^{N} \beta_i P_{C_i}$ and $P_2 = P_{Q_1} ... P_{Q_M}$.

with positive real numbers β_i and η_j such that $\sum_{i=1}^{N} \beta_i = \sum_{j=1}^{M} \eta_j = 1$.

In the present article, we propose a iterative algorithmic scheme which is given with a self adaptive step-size. We also give a relaxed variant of this scheme by using projections onto half-spaces instead of those onto the original convex sets.

2. Preliminaries

In this section, we introduce some definitions and lemmas which can be used in the proof of our main result.

Definitions 1.1. A mapping *T* from a subset *K* of E^n into E^m is called:

(i) nonexpansive, if

$$\|T_x - T_y\| \leq \|x - y\| \text{ for all } x, y \in K;$$

(ii) γ inverse strongly monotone if

$$\gamma \|T_x - T_y\|^2 \leq \langle T_x - T_y, x - y\rangle \text{ for all } x, y \in K,$$

where γ is a positive number, and firmly nonexpansive if, in addition, $\gamma = 1$;

(iii) averaged, if $T = (1-\alpha)E + \alpha U$ for some fixed $\alpha \in (0;1)$ and a nonexpansive mapping *U*, and we say *T* is α- averaged.

For a closed convex subset *K* of E^n, there exists a mapping P_K from E^n onto *K* such that $\|P_K x - x\| \leq \inf_{y\in K} \|y - x\|$ for each $x \in E^n$. The mapping P_K is called the metric projection on *K*. We know that P_K is firmly nonexpansive [10] (hence, nonexpansive) and 1/2-averaged [5]. Moreover,

$$\|x - P_K x\|^2 + \|P_K x - z\|^2 \leq \|x - z\|^2, x \in E^n, z \in K.$$

We denote by $\mathrm{Fix}(T) = \{x \in K : Tx = x\}$ the set of fixed points for a mapping *T*.

Lemma 2.1. [9] Let E^n be any real Euclidean space, T_i be an α_i-averaged mapping with $\alpha_i > 0$ for each $i \in I$ and let $\omega = (\omega_1, \omega_2, ..., \omega_N)$ be a positive real vector such that $\sum_{i=1}^{N} \omega_i = 1$. Set $T = \sum_{i=1}^{N} \omega_i T_i$ and $\alpha = \sum_{i=1}^{N} \omega_i \alpha_i$. Then, *T* is α-averaged. Moreover, the mapping $\tilde{T} = T_N T_{N-1} ... T_1$ is $\tilde{\alpha}$-averaged with $\tilde{\alpha} = 1/\left(1 + 1/\sum_{i=1}^{N} \alpha_i/(1-\alpha_i)\right)$ and

$$\mathrm{Fix}(T) = \mathrm{Fix}(\tilde{T}) = \bigcap_{i=1}^{N} \mathrm{Fix}(T_i).$$

Lemma 2.2. [13] Assume E^n and E^m are real Euclidean spaces. Let $A : E^n \to E^m$ be an $m \times n$-matrix of real numbers such that $A \neq 0$ and let $\bar{T} : E^m \to E^m$ be a nonexpansive mapping. Then, for every fixed $\gamma \in \left(0; 1/\|A\|^2\right)$, $E - \gamma A^T (E - \bar{T})A$ is $\gamma\|A\|^2$-averaged.

3. Main result

Let the *string* $I_t = \left(i_1^t, i_2^t, ..., i_{\gamma(I_t)}^t\right)$ be a finite nonempty subset of *I*, for every t = 1, 2, …, S_1, where the length of the string I_t denoted by $\gamma(I_t)$, is the number of elements in I_t. Put $T_t^1 = P_{i^t_{\gamma(I_t)}} ... P_{i_2^t}.P_{i_1^t}$, where $P_{i_l^t} = P_{C_{i_l^t}}$, for $l = 1, 2, ..., \gamma(I_t)$, t = 1, 2, …, S_1. Given a positive weight vector $\beta = (\beta_1, \beta_2, ..., \beta_{S_1})$ with $\sum_{t=1}^{S_1} \beta_t = 1$, we define the algorithmic mapping $P_1 = \sum_{t=1}^{S_1} \beta_t T_t^1$. We suppose that every element of *I* appears in at least one of the string I_t. Analogously, for the family $\{Q_j\}_{j\in J}$, we can construct the mapping $P_2 = \sum_{h=1}^{S_2} \eta_h T_h^2$ where $T_h^2 = P_{j^h_{\gamma(J_h)}} ... P_{j_2^h}.P_{j_1^h}$,

$P_{j_g^h} = P_{Q_{j_g^h}}$, h = 1, 2, …, S_2, $g = 1,2,...,\gamma(J_h)$ and $\eta = (\eta_1, \eta_2, ..., \eta_{S_2})$ is also a positive weight vector such that $\sum_{h=1}^{S_2} \eta_h = 1.$

Algorithmic scheme 1:

Step 0: Let x^1 and ε_1 be any point in E^n and any positive real number, respectively, and set k:=1;

Step 1: Assume that the k^{th} iterate x^k has been constructed. If

$$(E - P_1)x^k = (E - P_2)Ax^k = 0$$

then stop and x^k is a solution of (1.1). Otherwise, compute

$$x^{k+1} = P_1(E - \gamma_k A^*(E - P_2)A)x^k \tag{3.1}$$

where $\gamma_k = \rho_k q(x^k) / \left\| A^*(E - P_2)Ax^k \right\|^2$

if $(E - P_2)Ax^k \neq 0$ and

$$\gamma_k = \frac{\rho_k \tilde{q}(x^k)}{\left(\left\| A^*(E - P_2)Ax^k \right\| + \varepsilon_k\right)^2},$$

$$\tilde{q}(x) = \frac{1}{2}\sum_{h=1}^{S_2} \eta_h \left\|(E - T_h^2)Ax\right\|^2, \tag{3.2}$$

if $(E - P_2)Ax^k = 0$.

Step 2: Set k: = k + 1 and go to Step 1.

where, the parameter ρ_k and ε_k, for all $k \geq 1$, satisfy, respectively, the conditions (ρ): $0 < \underline{\rho} \leq \rho_k \leq \overline{\rho} < 2$ and (ε): $\{\varepsilon_k\}$ is a bounded sequence of positive real numbers such that $\liminf_{k\to\infty} \varepsilon_k > 0$.

For the sake of simplicity in programming, the next iterate x^{k+1} can be calculated by (3.1) and (3.2) without verifying the zero value for $(E - P_2)Ax^k$.

First, we have the following lemmas.

Lemma 3.1. $z \in \Gamma$ *if and only if* $(E - P_1)z = A^T(E - P_2)Az = 0$. *Moreover, the last equality holds if and only if* $(E - P_2)Az = 0$.

Lemma 3.2. There holds the following inequality

$$\frac{1}{2R}\sum_{h=1}^{\gamma(J_h)} \left\|\tilde{U}^{j_g^h} y - \tilde{U}^{j_{g-1}^h} y\right\|^2 \leq \left\|(E - T_h^2)y\right\|,$$

for some positive constant R and any $y \in E^m$, where $\tilde{U}^{j_i^t} = P_{j_i^t} ... j_2^t . j_1^t$ and $\tilde{U}^{j_0^t} = E.$

We have the following main results.

Theorem 3.1. *Let E^n and E^m be two real Euclidean spaces, A be an $m \times n$-matrix of real numbers such that $A \neq 0$. Let $\Gamma \neq \emptyset$, C_i and Q_j, for each $i \in I$ and $j \in J$ be closed convex subsets in E^n and E^m, respectively. Assume that there hold conditions (ρ) and (ε). Then, the sequence $\{x^k\}$, defined by algorithmic scheme 1, converges to a solution of* (1.1) *as* $k \to \infty$.

Proof. We consider only the case when the algorithm does not terminate in a finite number of iterations. First, we prove that $\{x^k\}$ is bounded. Take a point $p \in \Gamma$. Then, since P_1 is nonexpansive and $E - T_h^2$ is 1/2-inverse strongly monotone [17], we have that

$$\| x^{k+1} - p\|^2 = \| P_1 (E - \gamma_k A^T(E - P_2)A)x^k - P_1 p\|^2$$

$$\leq \| x^k - p - \gamma_k A^T(E - P_2)Ax^k \|^2$$

$$= \| x^k - p\|^2 - 2\gamma_k \langle (E - P_2)Ax^k - (E - P_2)Ap, Ax^k - Ap \rangle + \gamma_k^2 \| A^T(E - P_2)Ax^k \|^2 \tag{3.3}$$

$$\overset{(3.2)}{=} \| x^k - p\|^2 - 2\gamma_k \sum_{h=1}^{S_2} \eta_h \left\langle (E - T_h^2)Ax^k - (E - T_h^2)Ap, Ax^k - Ap \right\rangle$$

$$+ \gamma_k^2 \| A^T(E - P_2)Ax^k \|^2$$

$$\leq \| x^k - p\|^2 - 2\gamma_k \sum_{h=1}^{S_2} \eta_h \frac{1}{2} \left\|(E - T_h^2)Ax^k\right\|^2$$

$$+ \gamma_k^2 (\| A^T(E - P_2)Ax^k \| + \varepsilon_k)^2$$

$$= \| x^k - p\|^2 - \rho_k(2 - \rho_k)\tilde{q}^2(x^k) / (\| A^T(E - P_2)Ax^k\| + \varepsilon_k)^2,$$

from which and condition (ρ) it implies that $\| x^{k+1} - p\| \leq \| x^k - p\|$. Consequently, $\{x^k\}$ is bounded and there exists $\liminf_{k\to\infty} \|x_k - p\| > 0$. Therefore, by virtue of (3.3) with conditions (ρ) and (ε), we get that $\liminf_{k\to\infty} \tilde{q}(x_k) = 0$. From this and

$$\|(E - P_2)Ax^k\|^2 = \left\|\left(E - \sum_{h=1}^{S_2} \eta_h T_h^2\right)Ax^k\right\|^2$$

$$= \left\|\sum_{h=1}^{S_2} \eta_h (E - T_h^2)Ax^k\right\|^2 \leq \sum_{h=1}^{S_2} \eta_h \left\|(E - T_h^2)Ax^k\right\|^2$$

$$= 2\tilde{q}(x^k)$$

it follows that

$$\lim_{k\to\infty} \|(E - P_2)Ax^k\| = 0. \qquad (3.4)$$

Let $\{x^{k_l}\}$ be a subsequence of $\{x^k\}$ such that $x^{k_l} \to x' \in E^n$ as $l \to \infty$. As the mapping $(E - P_1)A$ is continuous, from (3.4) we get that $(E - P_2)Ax' = 0$. In order to prove that x' is a solution of (1.1), by Lemma 3.1, we have to show that $x' = P_1 A x'$. Indeed, from (3.1) we can write that

$$x^{k+1} = P_1(x^k + y^k)$$

where $y^k = -\gamma_k A^T(E - P_2)Ax^k \to 0$ as $k \to \infty$, that is followed from (3.4) again, (3.2) and the property of ε_k. So, $x' = P_1 x'$, and hence, $x' \in \Gamma$. Then,

$$\lim_{k\to\infty}\left\|x^k - x'\right\| = \lim_{l\to\infty}\left\|x^{k_l} - x'\right\| = 0,$$

i.e., all the sequence $\{x^k\}$ converges to x' as $k \to \infty$. The proof is completed.

Remark 1

In the case that $S_2 = M$ and $\gamma(I_t) = 1$ for $t = 1, 2, \ldots, M$, since $E - P_{Q_j}$ is firmly nonexpansive,

$$\langle (E - P_2)Ax^k, Ax^k - Ap \rangle =$$

$$\sum_{j=1}^{M} \eta_j \left\langle (E - P_{Q_j})Ax^k - (E - P_{Q_j})Ap, Ax^k - Ap \right\rangle$$

$$\geq \sum_{j=1}^{M} \eta_j \left\|(E - P_{Q_j})Ax^k\right\|^2 := 2q(x^k), \qquad (3.5)$$

that is the proximity function, introduced by Xu [15]. By taking

$$\gamma_k = \rho_k q(x^k) / (\| A^T(E - P_2)Ax^k\| + \varepsilon_k)^2,$$

we obtain that the upper bound for γ_k equal to 4.

In algorithmic schemes 1, we assume that all the projections P_{C_i} and P_{Q_j} can be easily calculated, but in practice they are sometime difficult to compute or even impossible. In this case, one can turn to relaxed method, proposed by Yang [16] and studied in [11], [14] with the proximity function $q(x)$ defined in the previous section.

Now, we give a relaxed variant for algorithmic scheme 1. First, we assume that the convex subsets C_i and Q_j in this part satisfy the following assumptions:

(a1) The subset C_i for all $i \in I$ is given by $C_i = \{x \in E^n : c_i(x) \leq 0\}$, where $c_i : E^n \to (-\infty, +\infty)$ is a convex function. The subset Q_j for all $j \in J$ is given by

$$Q_j = \{y \in E^m : q_j(y) \leq 0\},$$

where $q_j : E^m \to (-\infty, +\infty)$ is a convex function.

(a2) For any $x \in E^n$ and $y \in E^m$, at least one of subdifferetial $\xi_i \in \partial c_i(x)$ and $\theta_j \in \partial q_j(y)$ can be computed, where $\partial c_i(x)$ and $\partial q_j(y)$ are the subdifferentials of $c_i(x)$ and $q_j(y)$ at the points x and y, respectively,

$$\partial c_i(x) = \left\{ \begin{array}{r} \xi_i \in E^n : c_i(x') \geq c_i(x) + \langle \xi_i, x' - x \rangle, \\ \forall x' \in E^n \end{array} \right\},$$

$$\partial q_j(y) = \left\{ \begin{array}{r} \theta_j \in E^m : q_j(y') \geq q_j(y) + \langle \theta_j, y' - y \rangle, \\ \forall y' \in E^m \end{array} \right\}.$$

We define the following half-spaces:

$$C_i^k = \left\{ x \in E^n : c_i(x^k) + \left\langle \xi_i^k, x^k - x \right\rangle \leq 0 \right\}, \quad \xi_i^k \in \partial c_i(x^k), \quad i \in I,$$

and

$$Q_j^k = \left\{ y \in E^m : q_j(y^k) + \left\langle \theta_j^k, y^k - xy \right\rangle \leq 0 \right\}, \quad \theta_j^k \in \partial q_j(y^k), \quad j \in J,$$

Put $T_t^{1,k} = P^k_{i^t_{\gamma(I_t)}} \ldots P^k_{i^t_2} . P^k_{i^t_1}$, where $P^k_{i^t_l} = P_{C^k_{i^t_l}}$, for all $l = 1, 2, \ldots, \gamma(I_t)$ and $t = 1, 2, \ldots, S_1$. We define the algorithmic mapping $P_1^k = \sum_{t=1}^{S_1} \beta_t T_t^{1,k}$ with the positive weight vector β_t as in the previous section. We suppose also that every element of I appears in at least one of the string I_t. Let $P_2^k := \sum_{t=1}^{S2} \eta_t T_t^{2,k}$ where $P^k_{j^t_l} = P_{Q^k_{j^t_l}}$.

By Lemma 2.1, if

$$(E - P_1^k)z = A^T(E - P_2^k)Az = 0$$

then we have only that $z \in \cap_{i=1}^{N} C_i^k$ and $Az \in \cap_{j=1}^{M} Q_j^k$. It is difficult to confirm that z is a solution of (1.1). So, we consider the following relaxed algorithmic scheme.

Algorithmic scheme 2

Step 0: Let x^1 and ε_1 be any point in E^n and any positive real number, respectively, and set $k = 1$;

Step 1: The kth iterate x^k is constructed by

$$x^{k+1} = P_1^k (E - \gamma_k A^T(E - P_2^k)A)x^k, \quad (3.6)$$

where $\gamma_k = \rho_k q(x^k) / \| A^T(E - P_2^k)Ax^k\|^2$

if $(E - P_2^k)Ax^k \neq 0$ and

$$\gamma_k = \rho_k q_k(x^k) / (\| A^T(E - P_2^k)Ax^k\| + \varepsilon_k)^2,$$

if $(E - P_2^k)Ax^k = 0$, (3.7)

where $q_k(x) = \frac{1}{2}\sum_{j=1}^{M}\eta_j \left\|\left(E - P_j^k\right)Ax\right\|^2$ and the parameter ρ_k, for all $k \geq 1$, satisfies a new condition $(\rho') : 0 \leq \underline{\rho} \leq \rho_k \leq \bar{\rho} \leq 4$.

The following Lemma is essential in proving convergence.

Lemma 3.3 [12] *Suppose h is a convex function on E^n, then it is subdifferentiable everywhere and its subdifferentials are uniformly bounded subsets of E^n.*

Lemma 3.3 shows that the subdifferentials are bounded on bounded sets.

Theorem 3.2 Let E^n, E^m, A and Γ be as in Theorem 3.1. Let C_i and Q_j, for each $i \in I$ and $j \in J$, be closed convex subsets in E^n and E^m, that be defined by (a1) and (a2). Assume that there hold conditions (ρ') and (ε). Then, the sequence $\{x^k\}$, defined by (3.6)-(3.7), converges to a solution of (1.1) as $k \to \infty$.

Proof. Take a point $p \in \Gamma$. Since $C_i \subseteq C_i^k$, $Q_j \subseteq Q_j^k$, we have $p = P_i p = P_i^k p$ and $Ap = P_j Ap = P_j^k p$ for all $i \in I, j \in J$ and $k \geq 1$. By the similar argument as the above for (3.3), we have that

$$\begin{aligned}
&\| x^{k+1} - p\|^2 \\
&= \| P_1^k (E - \gamma_k A^T(E - P_2^k)A)x^k - P_1^k p\|^2 \\
&\leq \| x^k - p - \gamma_k A^T(E - P_2^k)Ax^k \|^2 \\
&= \| x^k - p\|^2 \\
&- 2\gamma_k \langle (E - P_2^k)Ax^k - (E - P_2^k)Ap, Ax^k - Ap\rangle \\
&\quad + \gamma_k^2 \| A^T(E - P_2^k)Ax^k \|^2 \\
&\leq \| x^k - p\|^2 - 4\gamma_k q_k(x^k) \\
&\quad + \gamma_k^2 (\| A^T(E - P_2^k)Ax^k \| + \varepsilon_k)^2 \\
&= \| x^k - p\|^2 \\
&- \rho_k(4 - \rho_k) q_k^2(x^k) / (\| A^T(E - P_2^k)Ax^k\| + \varepsilon_k)^2.
\end{aligned}$$

Therefore, $\{x^k\}$ is bounded, there exists $\lim_{k\to\infty}\left\|x^k - p\right\|$ and $\lim_{k\to\infty} q_k(x^k)$. Clearly, from the last limit and (3.5) with P_{Q_j} replaced P_j^k, it follows that

$$\lim_{k\to\infty}\left\|(E - P_j^k)Ax^k\right\| = 0, \quad (3.8)$$

for all $j \in J$. Moreover, we have also that $\lim_{k\to\infty}\left\|(E - P_2^k)Ax^k\right\| = 0,$ because

$$\begin{aligned}
\left\|(E - P_2^k)Ax^k\right\|^2 &= \left\|\sum_{j=1}^{M}\eta_j (E - P_j^k)Ax^k\right\|^2 \\
&\leq \sum_{j=1}^{M}\eta_j \left\|(E - P_j^k)Ax^k\right\|^2
\end{aligned}$$

and (3.8). Put $z^k := x^k - \gamma_k A^T(E - P_2^k)Ax^k$. Then, we can write that

$$\begin{aligned}
&\| x^{k+1} - p\|^2 = \| P_1^k z^k - p\|^2 \\
&\leq \sum_{t=1}^{S_1}\beta_t \left\|T_t^{1,k} z^k - p\right\|^2 \\
&\leq \left\|z^k - p\right\|^2 - \sum_{t=1}^{S_1}\beta_t \sum_{l=1}^{\gamma(I_t)}\left\|U^{i_l^t} z^k - U^{i_{l-1}^t} z^k\right\|^2 \\
&= \| x^k - p\|^2 - 2\gamma_k \langle A^T(E - P_2^k)Ax^k, x^k - p \rangle \\
&\quad + \gamma_k^2 \| A^T(E - P_2^k)Ax^k\|^2 \\
&\quad - \sum_{t=1}^{S_1}\beta_t \sum_{l=1}^{\gamma(I_t)}\left\|U^{i_l^t} z^k - U^{i_{l-1}^t} z^k\right\|^2
\end{aligned}$$

where $U_{i_l^t}^k = P_{i_l^t}^k ... P_{i_2^t}^k . P_{i_1^t}^k$ and $U_{i_0^t}^k = E$. Using the last inequality with the properties of $\{x^k\}$ and $\left\{A^T(E - P_2^k)Ax^k\right\}$, we obtain that $\lim_{k\to\infty}\left\|U_{i_l^t}^k z^k - U_{i_{l-1}^t}^k z^k\right\| = 0,$ this implies that

$$\lim_{k\to\infty}\left\|(E - P_i^k)x^k\right\| = 0, \quad \forall i \in I. \quad (3.9)$$

Next, from the definitions of C_i^k and Q_j^k, it follows that

$$\begin{aligned}
c_i(x^k) &\leq \left\|\xi_i^k\right\|\left\|(E - P_i^k)x^k\right\|, \\
q_j(Ax^k) &\leq \left\|\theta_j^k\right\|\left\|(E - P_j^k)Ax^k\right\|.
\end{aligned} \quad (3.10)$$

Since $\{x^k\}$ is bounded, $\{Ax^k\}$ is bounded in E^m. Therefore, $\left\{\xi_i^k\right\}, \left\{\theta_j^k\right\}$ are bounded and there exists a subsequence $\left\{x^{k_l}\right\}$ of $\{x^k\}$ such that $\left\{x^{k_l}\right\}$ converges to a point $\tilde{x} \in E^n$. Thus, from (3.8)-(3.10) it follows that $c_i(\tilde{x}) \le 0$ and $q_j(A\tilde{x}) \le 0$ for all $i \in I$ and $j \in J$. It means that $\tilde{x} \in \Gamma$. Then, $\lim_{k\to\infty}\|x^k - \tilde{x}\| = \lim_{l\to\infty}\|x^{k_l} - \tilde{x}\| = 0$, i.e., $\{x^k\}$ converges to $\tilde{x} \in \Gamma$. This completes the proof.

Remark 2. Theorem 3.1 has value, when $P_2^k = \sum_{h=1}^{S_2} \eta_h T_h^{2,k}$ with the positive weight vector η as in the previous section, but under condition (ρ) instead of (ρ'). Here, instead of $q_k(x)$, we use the function

$$\tilde{q}_k(x) = \frac{1}{2}\sum_{h=1}^{S_2}\eta_h \left\|(E - T_h^{2,k})Ax\right\|^2.$$

Indeed, as in the proof of Theorem 3.1, we get that $\lim_{k\to\infty}\|(E - T_h^{2,k})Ax^k\| = 0$ for all h = 1, 2, … S_2. Further, by Lemma 3.2, we obtain $\lim_{k\to\infty}\|(E - P_j^k)Ax^k\| = 0$.

4. Numerical examples

In this section, we present some preliminary numerical results, calculated by several methods of algorithmic schemes 1 and 2. The methods, used in computations, are (1.3) and new ones with a self-adaptive step size. In the first example, the sets C_i and Q_j are defined by

$$C_i = \left\{x \in E^2 : \|x - a^i\|^2 \le 1\right\}$$

and

$$Q_j = \left\{y \in E^3 : \|y - a^j\|^2 \le 1\right\}$$

where a^i = (1-0.25i; 0) with N = 4 and a^j =(-1+0.1(j-1); 0; 0) with M = 21. Elements of matrix A has values: $a_{11} = a_{22} = 1$; $a_{21} = a_{12} = 0$ and $a_{31} = a_{32} =$

Table 1. *Method* (3.1) - (3.2) *with* $P_1 = \frac{1}{4}\sum_{i=1}^{4} P_i$ *and* $P_2 = \frac{1}{21}\sum_{j=1}^{21} P_j$.

k	x_1^{k+1}	x_2^{k+1}	k	x_1^{k+1}	x_2^{k+1}
10	0.1322489018	-0.1096046955	100	0.0134375293	-0.0118695275
20	0.0590133866	-0.0531213074	200	0.0081669713	-0.0072623184
30	0.0385431816	-0.333061823	300	0.0062576955	-0.0055807045
40	0.0291680112	-0.0253854829	400	0.0052250300	-0.0046678939
50	0.0237943020	-0.0208028142	500	0.0045615413	-0.0040800764

Table 2. *Method* (3.1) - (3.2) *with* $P_1 = P_4 \ldots P_1$ *and* $P_2 = \frac{1}{21}\sum_{j=1}^{21} P_j$.

k	x_1^{k+1}	x_2^{k+1}	k	x_1^{k+1}	x_2^{k+1}
10	0.6665662985	-0.4624660997	100	0.3084168689	-0.3570661143
20	0.5499195599	-0.4292417888	200	0.2383683602	-0.3352044020
30	0.4803304781	-0.4089281741	300	0.2052435826	-0.3247198937
40	0.4333536237	-0.3949881167	400	0.1842047953	-0.3179917898
50	0.3994798577	-0.3848417711	500	0.1695279308	-0.1312747211

Table 3. *Method* (3.1) - (3.2) *with* $P_1 = P_4 \ldots P_1$ *and* $P_2 = P_{21} \ldots P_2 . P_1$

k	x_1^{k+1}	x_2^{k+1}	k	x_1^{k+1}	x_2^{k+1}
10	0.1619563184	-0.3211904319	100	0.0347113740	-0.1986689830
20	0.0930360637	-0.2684585047	200	0.0256552010	-0.1791816768
30	0.0694629971	-0.2456850724	300	0.0219830960	-0.1691445434
40	0.0574979870	-0.2321570358	400	0.0198472169	-0.1624730780
50	0.0501837307	-0.2228438437	500	0.0183962369	-0.1575184514

1/2. Clearly, $\cap_{j=1}^{21} Q_j = \{(0,0,0)\}$. Therefore, p_*=(0, 0) is the unique solution. Put $T^{2,1} = P_7 \ldots P_2.P_1$ and $T^{2,2} = P_{14} \ldots P_9.P_8$ and $T^{2,3} = P_{21} \ldots P_{16}.P_{15}$.

The numerical results computed by several methods, defined by algorithmic schem 2 with $\rho_k = 0.4 + 1/(k+2), \varepsilon_k = 0.1 + 1/(k+2),$ an initial point $x^1 = (3;-2.5)$ and different forms of P_1^k and P_2^k, are given in the following tables.

$C_1 = \{x_1, x_2 \in E^2 : x_1 + x_2^2 \le 0\};$

$C_2 = \{x_1, x_2 \in E^2 : x_1^2 + x_1 - 1 \le 0\};$

$C_3 = \{x_1, x_2 \in E^2 : x_1 + x_2 - 2 \le 0\};$

$C_4 = \{x_1, x_2 \in E^2 : x_1^2 / 4 + x_2^2 / 9 - 3 \le 0\};$

Table 4. *Method* (3.1) - (3.2) *with* $P_1 = \frac{1}{4}\sum_{i=1}^{4} P_i$ *and* $P_2 = P_{21} \ldots P_2.P_1$

k	x_1^{k+1}	x_2^{k+1}	k	x_1^{k+1}	x_2^{k+1}
10	0.1693965225	-0.3615422314	100	0.0358096772	-0.2054187867
20	0.0969110550	-0.2917779458	200	0.0263063929	-0.1829700425
30	0.0722254089	-0.2627693262	300	0.0224535356	-0.1717800444
40	0.0596950361	-0.2458839872	400	0.0202151422	-0.1644844678
50	0.0520319645	-0.2344249979	500	0.0186971106	-0.1591374628

Table 5. *Method* (3.1) - (3.2) *with* $P_1 = \frac{1}{4} p_4 + \frac{3}{4} p_3 p_2 p_1$ *and* $P_2 = \sum_{t=1}^{3} T^{2,t} / 3$.

k	x_1^{k+1}	x_2^{k+1}	k	x_1^{k+1}	x_2^{k+1}
10	0.4814850658	-0.5098712064	100	0.0492697719	-0.3408147049
20	0.3098246489	-0.4476210480	200	0.0237776835	-0.3183714090
30	0.2163652970	-0.4116171161	300	0.0162671169	-0.3069244773
40	0.1602679486	-0.3892344990	400	0.0125256305	-0.2990155796
50	0.1236226400	-0.3742856213	500	0.0102614291	-0.2928868336

Analyzing the numerical results, we see that method (2.2)-(2.3) with P_1 and P_2 defined by convex combinations of P_{C_i} and P_{Q_j} respectively, gives a better result than those with other cases of P_1 and P_2.

In the second example, we consider the sets

$Q_1 = \{y_1, y_2, y_3 \in E^3 : y_1 + y_2^2 + 2y_3 \le 0\};$

$Q_2 = \{y_1, y_2, y_3 \in E^3 : y_1^2 + y_2 + y_3 \le 0\};$

$Q_3 = \left\{y_1, y_2, y_3 \in E^3 : \frac{y_1^2}{4} + \frac{y_2^2}{9} + \frac{y_3^2}{16} - 1 \le 0\right\}.$

Since we do not know the exact solution to (1.1) with C_i and Q_j given above, we use

Table 6. *Method* (3.6) - (3.7) *with* $P_1^k = \frac{1}{4}\sum_{i=1}^{4} P_i^k$ *and* $P_2^k = \frac{1}{3}\sum_{j=1}^{3} P_j^k$.

k	x_1^{k+1}	x_2^{k+1}	e^k
20	-1.8550560864	-1.2529823091	0.0347011051
40	-1.0148022648	-0.9888362074	0.0020806016
60	-1.0022774018	-0.9941922526	0.0001488434
80	-1.0000275029	-0.9953053196	0.0000568477
100	-1.9989834883	-0.9958226816	0.0000311400

The computational results by method (3.6) with the same data as the above and new P_1^k and P_2^k are given in the following numerical table.

$e^k = \|x^{k+1} - x^k\| / \|x^k\|$ to measure the error of the kth step iteration. The computational results, by using algorithmic scheme 2 with the same values of ρ_k, ε_k and new x^1 =(-3;-2.5) are presented in the numerical tables, Tables 6 and 7.

Table 7. *Method* (3.6) - (3.7) *with* $P_1^k = \frac{1}{4}P_4^k + \frac{3}{4}P_3^k P_2^k P_1^k$ *and* $P_2^k = \frac{1}{3}P_3^k + \frac{2}{3}P_2^k P_1^k$.

k	x_1^{k+1}	x_2^{k+1}	e^k
20	-1.0013989719	-0.9931580146	0.0004878143
40	-0.9970955125	-0.9952845740	0.0000716891
60	-0.9959607340	-0.9958465143	0.0000298011
80	-0.9954071353	-0.9961214866	0.0000167111
100	-0.9950714116	-0.9962880729	0.0000108605

Clearly, the numerical results in Table 7 show that new method (3.6)-(3.7) with

$$P_1^k = \frac{1}{4}P_4^k + \frac{3}{4}P_3^k P_2^k P_1^k \text{ and } P_2^k = \frac{1}{3}P_3^k + \frac{2}{3}P_2^k P_1^k .$$

is a little faster than the first one, that is usually called the relaxed simultaneous method.

5. Conclusion

In this paper, we proposed a general approach to construct iterative methods for solving the multiple-sets split feasibility problem (MSSFP), that is string-averaged algorithmic schemes.

REFERENCES

[1] Buong. N. (2017), Iterative algorithms for the multiple-sets split feasibility problem in Hilbert spaces, *Numer. Algorithms*, **vol. 76**, pp. 783-789.

[2] Buong. N., Hoai, P. T. T., Binh. K. T. (2020), Iterative regularization methods for the multiple-sets split feasibility problem in Hilbert spaces, *Acta Applicandae Mathematica*, **vol. 165**, pp. 183-197.

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

[4] Byrne. C. (2004), A unified treatment of some iterative methods in signal processing and image reconstruction, *Inverse Problems*, **vol. 20**, pp. 103-120.

[5] Censor. Y., Elfving. T. (1994), A multiprojection algorithm using Bregman projections in a product spaces, *Numer. Algorithms*, v**ol. 8**, pp. 221-239.

[6] Censor. Y., Elfving. T., Herman. G. T. (2001), Averaging strings of sequential iterations for convex feasibility problems. In: Inhenrently parallel algorithms in feasibility and optimization and their applications (Haifa, 2000) *Stud Comput. Math*, **vol. 8**, pp. 101-113, *North-Holland Amstrerdam*.

[7] Censor. Y., Elfving. T., Knop. N., Bortfeld. T. (2005), The multiple-sets split feasibility problem and its applications for inverse problems, *Inverse Problems*, **vol. 21**, pp. 2071-2084.

[8] Censor. Y., Bortfeld. T., Martin. B., Trofimov. A. (2006), A unified approach for inverse problems in intensity-modulated radiation therapy, *Phys. Med. Biol*, **vol. 51**, pp. 2353-2365.

[9] Censor. Y., Motova. A., Segal. A. (2007), Perturbed projections and subgradient projections for the multiple-sets split feasibility problem, *J. Math. Anal. Appl*, **vol. 327**, pp. 1244-1256.

[10] Chen. Y., Guo. Y., Yu. Y., Chen. R. (2012), Self-adaptive and relaxed self-adaptive projection methods for solving the multiple-set split feasibility problems, *Abstract and Applied Analysis*, Article ID 958040, 11pp, doi:10.1155/2012/958040.

[11] Qu. B., Xiu. N. (2008), A new halfspace-relaxation projection method for the split feasibility problem, *Linear algebra and its applications*, **vol. 428**, pp.1218-1229.

[12] Rockafellar. R.T. (1970), *Convex Analysis*, Princeton University Press, NJ.

[13] Takahashi. W., Toyota. M. (2003), Weak convergence theorems for nonexpansive mappings and monotone mappings, *J. Optim. Theory and Appl*, **vol. 118**, pp. 417-428.

[14] Wang. J., Hu. Y., Yu. C. K. W., Zhuang. X. (2019), A family of projection gradient methods for solving the multiple-sets split feasibility problem, *J. Optim. Theory Appl*, **vol. 183**, pp. 520-534.

[15] Xu. H.K. (2006), A variable Krasnosel'skii -Mann algorithm and multiple set split feasibility problem, *Inverse Problems*, **vol. 22**, pp. 2021-2034.

[16] Yang. Q. (2004), The relaxed CQ-algorithm solving the multiple-sets split feasibility problem, *Inverse Problem*, **vol. 20**, pp. 1261-1266.

[17] Zarantonello. E. H. (1971), Projections on convex sets in Hilbert sapce and spectral theory, in: E.H. Zarantonello (Ed.) *Constributions to Nonlinear Functional Analysis, Academic, New York*.