Nonlinear Convex Analysis and Optimization
Vol. 3, No. 2, 2024, pp. 63–89
https://ncao.design.blog

A Self-adaptive Super Set-relaxed Projection Method for Multiple-sets Split Feasibility Problem with Multiple Output Sets

Guash Haile Taddele [a, c,1,*], Songpon Sriwongsa [a,b,2], Attapol Kaewkhao[d,3]

[a] Fixed Point Research Laboratory, Fixed Point Theory and Applications Research Group, Center of Excellence in Theoretical and Computational Science (TaCS-CoE), Faculty of Science, King Mongkut's University of Technology Thonburi (KMUTT), 126 Pracha Uthit Rd., Bang Mod, Thung Khru, Bangkok 10140, Thailand
[b] Department of Mathematics, Faculty of Science, King Mongkut's University of Technology Thonburi (KMUTT), 126 Pracha Uthit Rd., Bang Mod, Thung Khru, Bangkok 10140, Thailand
[c] Department of Mathematics, Xiamen University Malaysia, Sepang 43900, Malaysia
[d] Research Group in Mathematics and Applied Mathematics, Department of Mathematics, Faculty of Science, Chiang Mai University, Chiang Mai, 50200, Thailand
[1] guashhaile79@gmail.com; [2] songpon.sri@kmutt.ac.th; [3] attapol.k@cmu.ac.th
[*] Corresponding Author

ABSTRACT

This paper introduces an inertial accelerated super set-relaxed CQ method to solve a multiple-sets split feasibility problem with multiple output sets. The convex subsets involved are assumed to be level subsets of given strongly convex functions. Instead of using the involved sets, we approximate the original convex subsets with a sequence of closed balls. The proposed method is easy to implement as the projection onto the closed ball has a closed form. Additionally, we develop a new self-adaptive step-size that does not require any prior information of the norm. Under suitable assumptions, we establish and prove a strong convergence result for the algorithm. Numerical experiments are provided to demonstrate the performance of the proposed algorithm, which generalizes and improves upon existing literature.

Article History
Received 7 Sep 2024
Revised 26 Oct 2024
Accepted 29 Oct 2024

Keywords:
Split feasibility problem; Self-adaptive technique; Half-space relaxation; Strong convergence

MSC
47H09; 47H10; 65K05; 90C25; 47J25; 97M40

1. Introduction

In 1994, Censor and Elfving [9] initially introduced the split feasibility problem within finite-dimensional Hilbert spaces for modeling various inverse problems, including phase retrievals and in medical image reconstruction [9, 4], IMRT [10, 11], gene regulatory network inference [37].

Please cite this article as: G.H. Taddele et al., A Self-adaptive Super Set-relaxed Projection Method for Multiple-sets Split Feasibility Problem with Multiple Output Sets, Nonlinear Convex Anal. & Optim., Vol. 3, No. 2, 63–89. https://doi.org/10.58715/ncao.2024.3.4

Let H_1 and H_2 be two real Hilbert spaces. Let $T : H_1 \to H_2$ be a bounded linear operator and $T^* : H_2 \to H_1$ be its adjoint. The split feasibility problem (SFP) is to find a point $x^* \in H_1$ such that

$$x^* \in C \text{ and } Tx^* \in Q \tag{1.1}$$

where C and Q are nonempty, closed, and convex subsets of H_1 and H_2, respectively. Denote the set of solutions of the SFP (1.1) by $\Gamma = C \cap T^{-1}(Q) \neq \emptyset$.

Over the past century, numerous iterative methods have been developed and analyzed for solving the SFP (1.1), with a focus on practical applications. Notably, Byrne [4, 5] introduced the widely recognized and highly regarded CQ-algorithm, which remains the first and most prominent method of its kind. The algorithm is as follows: given an initial point $x_0 \in H_1$, iterate

$$x_{k+1} = P_C(x_k - \tau_k T^*(I - P_Q)Tx_k), \tag{1.2}$$

where P_C and P_Q are the metric projections onto C and Q, respectively, and the stepsize $\tau_k \in \left(0, \frac{2}{\|T\|^2}\right)$, where $\|T\|^2$ is the spectral radius of the matrix T^*T.

The CQ-algorithm, proposed by Byrne [4, 5], necessitates the computation of metric projection onto the sets C and Q. However, in certain cases, exact computation of the metric projection may be infeasible or prohibitively expensive. Additionally, determining the stepsize relies on the computation or at least estimation of the operator norm, which is a non-trivial task. In practical applications, the sets C and Q typically correspond to the level sets of convex functions, defined as follows:

$$C := \{x \in H_1 : c(x) \leq 0\} \text{ and } Q := \{y \in H_2 : q(y) \leq 0\}, \tag{1.3}$$

where, $c : H_1 \to \mathbb{R}$ and $q : H_2 \to \mathbb{R}$ are convex and subdifferentiable functions on H_1 and H_2, respectively. Furthermore, the subdifferentials $\partial c(x)$ and $\partial q(y)$ of c and q, respectively, are bounded operators, meaning they are bounded on bounded sets.

In 2004, Yang [39] introduced a generalized version of the CQ-algorithm, known as the relaxed CQ-algorithm. This algorithm requires the computation of the metric projection onto relaxed sets, specifically the half-spaces C_k and Q_k, where

$$C_k := \{x \in H_1 : c(x_k) \leq \langle \xi_k, x_k - x\rangle\}, \tag{1.4}$$

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

$$Q_k := \{y \in H_2 : q(Tx_k) \leq \langle \eta_k,\ Tx_k - y\rangle\}, \tag{1.5}$$

where $\eta_k \in \partial q(Tx_k)$. It is readily apparent that $C \subseteq C_k$ and $Q \subseteq Q_k$ for all $k \geq 1$. Additionally, it is established that the projections onto the half-spaces C_k and Q_k possess closed forms. Subsequently, define a convex and differentiable function denoted as $f_k(.)$, along with its corresponding gradient function $\nabla f_k(.)$ as

$$f_k(x_k) := \frac{1}{2}\|(I - P_{Q_k})Tx_k\|^2, \quad \nabla f_k(x_k) := T^*(I - P_{Q_k})Tx_k, \tag{1.6}$$

where Q_k is given as in (1.5). More precisely, Yang [39] introduced a relaxed CQ-algorithm for solving the SFP (1.1) in a finite-dimensional Hilbert space. The algorithm starts with a given point $x_0 \in C$ and iteratively updates the solution using the following equation:

$$x_{k+1} = P_{C_k}(x_k - \tau_k \nabla f_k(x_k)), \tag{1.7}$$

where $\tau_k \in \left(0, \frac{2}{\|T\|^2}\right)$. Since P_{C_k} and P_{Q_k} can easily be calculated, this method appears to be very practical. However, the computation of the norm of T proves to be intricate and resource-intensive. In order to address this challenge, López et al. [21] proposed a relaxed CQ-algorithm in 2012 for solving the SFP (1.1). This algorithm incorporates a novel adaptive method for determining the stepsize sequence τ_k, which is defined as follows:

$$\tau_k := \frac{\rho_k f_k(x_k)}{\|\nabla f_k(x_k)\|^2}, \tag{1.8}$$

where $\rho_k \in (0,4)$ such that $\liminf_{k\to\infty} \rho_k(4-\rho_k) > 0$, for all $k \geq 1$. It has been demonstrated that the sequence $\{x_k\}$, generated by (1.7) with τ_k defined by (1.8), exhibits weak convergence towards a solution of the SFP (1.1). In other words, their algorithm demonstrates weak convergence within the context of infinite-dimensional Hilbert spaces.

It is an established fact that a multitude of inverse problems manifest in infinite-dimensional spaces. In such scenarios, strong convergence is deemed more favorable than weak convergence for effectively resolving these problems. Notably, several authors have put forth algorithms that yield a sequence $\{x_k\}$, which strongly converges to a point within the solution set of the SFP (1.1). Relevant examples can be found in [21, 19, 40, 16, 17, 13]. In particular, López et al. [21] proposed an iterative scheme for solving the SFP (1.1) in the context of infinite-dimensional Hilbert spaces. The scheme is as follows: given a fixed point $u \in H_1$ and an initial guess $x_0 \in H_1$, the iteration is defined by (1.9):

$$x_{k+1} = \upsilon_k u + (1-\upsilon_k)P_{C_k}\Big(x_k - \tau_k \nabla f_k(x_k)\Big), \forall k \geq 1, \tag{1.9}$$

and in a subsequent work, Cholamjiak et al. [13] updated (1.9) as follows: given an initial guess $x_0 \in H_1$, the iteration is defined by (1.10):

$$x_{k+1} = P_{C_k}\Big((1-\upsilon_k)(x_k - \tau_k \nabla f_k(x_k))\Big), \forall k \geq 1, \tag{1.10}$$

where, $\{\upsilon_k\} \subset (0,1)$ such that $\lim_{k\to\infty} \upsilon_k = 0$ and $\sum_{k=1}^{\infty} \upsilon_k = +\infty$, and C_k, $\nabla f_k(x_k)$, and τ_k are given by (1.4), (1.6), and (1.8), respectively. Under certain standard conditions, it has been demonstrated that the sequence $\{x_k\}$ produced by (1.9) exhibits strong convergence towards the point $x^* = P_\Gamma(u)$. Similarly, the sequence $\{x_k\}$ generated by (1.10) also converges strongly, but towards the point $x^* = P_\Gamma(0)$ of the SFP (1.1). What is particularly noteworthy is that the iterative schemes (1.9)-(1.10) do not require any prior knowledge of the operator norm. Additionally, they are capable of computing the projections onto the half-spaces C_k and Q_k, making them easily implementable.

It is noteworthy that a ball-relaxed CQ-algorithm for solving the SFP (1.1) has been introduced by Yu et al. [41]. This algorithm is applicable under the condition that $c : H_1 \to (-\infty, +\infty]$ and $q : H_2 \to (-\infty, +\infty]$ are ϱ-strongly and σ-strongly convex subdifferentiable functions on H_1 and H_2, respectively. The conditions for $c(x)$ and $q(y)$ are as follows:

$$c(x) \geq c(x_k) + \langle \xi_k, x - x_k\rangle + \frac{\varrho}{2}\|x - x_k\|^2, \text{ where } \xi_k \in \partial c(x_k),$$

$$q(y) \geq q(Tx_k) + \langle \eta_k, y - Tx_k\rangle + \frac{\sigma}{2}\|y - Tx_k\|^2, \text{ where } \eta_k \in \partial q(Tx_k).$$

Replacing the half-spaces C_k ((1.4)) and Q_k ((1.5)), respectively, by the balls $\widetilde{C}_k$ and $\widetilde{Q}_k$, where

$$\widetilde{C}_k = \Big\{x \in H_1 : c(x_k) + \langle \xi_k, x - x_k\rangle + \frac{\varrho}{2}\|x - x_k\|^2 \leq 0\Big\}, \tag{1.11}$$

$$\widetilde{Q}_k = \Big\{y \in H_2 : q(Tx_k) + \langle \eta_k, y - Tx_k\rangle + \frac{\sigma}{2}\|y - Tx_k\|^2 \leq 0\Big\}, \tag{1.12}$$

Yu et al. [41] proposed the ball-relaxed method follows. For any initial guess $x_0 \in H_1$;

$$x_{k+1} = P_{\widetilde{C}_k}\left(x_k - \frac{\rho_k\|(I - P_{\widetilde{Q}_k})Tx_k\|^2}{2\|T^*(I - P_{\widetilde{Q}_k})Tx_k\|^2}T^*(I - P_{\widetilde{Q}_k})Tx_k\right), \tag{1.13}$$

where $\rho_k \in (0, 4)$ with $\liminf\limits_{k\to\infty} \rho_k(4-\rho_k) > 0$. Under certain standard assumptions, it has been demonstrated that the sequence $\{x_k\}$, generated by (1.13), converges weakly to a solution of the SFP (1.1).

In order to enhance the convergence rate, Nesterov [28] introduced a modified heavy ball method based on Polyak's [29] approach. The method is defined as follows:

$$\begin{aligned} y_k &= x_k + \beta_k(x_k - x_{k-1}), \\ x_{k+1} &= y_k - \tau_k\nabla g(y_k), \forall k \geq 1, \end{aligned} \tag{1.14}$$

where, $\{\tau_k\}$ is a positive sequence, g is a smooth convex function, and $\beta_k \in [0, 1)$ is an inertial factor. The term $\beta_k(x_k - x_{k-1})$ represents the inertia. Numerical experiments conducted in various fields of study have demonstrated that this method, along with other related techniques such as those presented in [1, 14, 15, 22, 24, 25], significantly improves the performance of non-inertial algorithms where $\beta_k = 0$. In light of this, several inertial-type methods have been proposed for solving SFPs, including those discussed in [34, 17, 33, 35].

Numerous variations/generalizations of the SFP have been extensively researched by numerous authors. Examples include the multiple-sets split feasibility problem [10], the split feasibility problem with multiple output sets [31], the split common fixed point problem [12, 26, 8, 20], the multiple-operator split common fixed point problem [3], and the split common null point problem [6, 36].

In the year 2020, Reich et al. [31] introduced and studied the following split feasibility problem with multiple output sets in infinite-dimensional Hilbert spaces. Let H, H_j, $j = 1, 2, \dots, r$, be real Hilbert spaces and let $A_j : H \to H_j$, $j = 1, 2, \dots, r$, be bounded linear operators. The split feasibility problem with multiple output sets (SFPMOS, for short) is to find an element x^* such that

$$x^* \in \Pi := C \cap \left(\cap_{j=1}^r A_j^{-1}(Q_j)\right) \neq \emptyset \tag{1.15}$$

where C and Q_j, $j = 1, 2, \dots, r$, are nonempty, closed and convex subsets of H and H_j, $j = 1, 2, \dots, r$, respectively.

Reich et al. [31] have proposed a projection gradient algorithm and a viscosity approximation iterative method for effectively solving the SFPMOS (1.15) in infinite-dimensional Hilbert spaces. These methods necessitate the computation of metric projections onto the sets C and Q_j, as well as the operator norm, which can be challenging to perform. The two iterative methods are as follows: for any given points $x_0, w_0 \in H$, $\{x_k\}$ and $\{w_k\}$ are sequences generated by

$$x_{k+1} := P_C\Big(x_k - \tau_k \sum_{j=1}^{r} A_j^*(I - P_{Q_j})A_j x_k\Big), \tag{1.16}$$

$$w_{k+1} := \upsilon_k f(w_k) + (1-\upsilon_k)P_C\Big(w_k - \tau_k \sum_{j=1}^{r} A_j^*(I - P_{Q_j})A_j w_k\Big), \tag{1.17}$$

where $f : C \to C$ is a strict contraction mapping of H into itself with the contraction constant $\theta \in [0, 1)$, $\{\tau_k\} \subset (0, \infty)$ and $\{\upsilon_k\} \subset (0, 1)$. It was proved that if the sequence $\{\tau_k\}$ satisfies the condition:

$$0 < a \leq \tau_k \leq b < \frac{2}{r \max_{j=1,2,\ldots,r}\{\|A_j\|^2\}}$$

for all $k \geq 1$, then the sequence $\{x_k\}$ generated by (1.16) converges weakly to a solution point $x^* \in \Pi$ of the SFPMOS (1.15). Furthermore, if the sequence $\{\upsilon_k\}$ satisfies the conditions:

$$\lim_{k\to\infty} \upsilon_k = 0 \quad \text{and} \quad \sum_{k=1}^{\infty} \upsilon_k = \infty,$$

then the sequence $\{w_k\}$ generated by (1.17) converges strongly to a solution point $x^* \in \Pi$ of the SFPMOS (1.15), which is a unique solution of the variational inequality

$$\langle (I - f)x^*, x - x^*\rangle \geq 0 \; \forall x \in \Pi.$$

Besides, using the viscosity approximation iterative method, Reich et al. in [32] proposed an optimization approach method which uses a self-adaptive step-size, for solving the SFPMOS (1.15). It In the context of SFPMOS (1.15), it is evident that the analysis solely focuses on scenarios where the first Hilbert space H comprises a single nonempty, closed, and convex subset C.

Motivating by the above works, in this work, we consider the following multiple-sets split feasibility problem with multiple output sets in general Hilbert spaces. In light of the aforementioned literature, our present study focuses on the following problem: multiple-sets split feasibility problem with multiple output sets.

Let H, H_j, $j = 1, 2, \ldots, r$, be real Hilbert spaces and let $T_j : H \to H_j$, $j = 1, 2, \ldots, r$, be bounded linear operators. The multiple-sets split feasibility problem with multiple output sets (MSSFPMOS, for short) is to find an element x^* such that

$$x^* \in \Omega := \Big(\cap_{i=1}^{s} C_i\Big) \cap \Big(\cap_{j=1}^{r} T_j^{-1}\Big(Q_j\Big)\Big) \neq \emptyset \tag{1.18}$$

where C_i, $i = 1, 2, \ldots, s$, and Q_j, $j = 1, 2, \ldots, r$, are nonempty, closed and convex subsets of H and H_j, $j = 1, 2, \ldots, r$, respectively, $s, r \geq 1$ are given integers. That is, $x^* \in C_i$ for each $i = 1, 2, \ldots, s$, and $T_j x^* \in Q_j$ for each $j = 1, 2, \ldots, r$. It is readily seen that, for the case where $s = 1$, the MSSFPMOS (1.18) reduced to the SFPMOS (1.15). If $s = 1 = r$, then MSSFPMOS (1.18) also reduced to the SFP (1.1).

A question at hand is that whether it is possible to extend the iterative method presented in (1.10) to solve the MSSFPMOS (1.18) within the framework of infinite dimensional Hilbert spaces, incorporating acceleration and the concept of the ball-relaxation. We are motivated to explore this question due to the numerous results found in existing literature. Therefore, in this paper, we propose an inertial accelerated self-adaptive super set (ball)-relaxed CQ method for solving the MSSFPMOS (1.18), where the closed convex sets are defined as level sets of strongly convex functions. This algorithm ensures strong convergence within the framework of infinite-dimensional Hilbert spaces.

The structure of this paper is as follows. In Section 2, we recall some definitions and basic results which are needed in the sequel. In Section 3, we present the algorithm we propose along with its convergence analysis. In Section 4, we present some newly driven results to solve another problem. We provide a set of numerical experiments to illustrate the efficacy of our proposed method in Section 5.

2. Preliminaries

In this section, we recall some definitions and basic results which are needed in the sequel.

Throughout this paper, let H, H_1 or H_2 be a real Hilbert space with the inner product $\langle .,. \rangle$, and induced norm $\|.\|$. Let I denote the identity operator on H, H_1 or H_2. Let the symbols " $\rightharpoonup$ " and " $\to$ ", denote the weak and strong convergence, respectively. For any sequence $\{x_k\} \subseteq H$, $\omega_w(x_k) = \{x \in H : \exists \{x_{k_m}\} \subseteq \{x_k\}$ such that $x_{k_m} \rightharpoonup x\}$ denotes the weak ω-limit set of $\{x_k\}$.

Definition 2.1. [2] Let C be a nonempty closed convex subset of H. An operator $T : C \to H$ is called

(1) Lipschitz continuous with constant $\sigma > 0$ on C if
$$\|Tx - Ty\| \leq \sigma \|x - y\|, \ \forall x, y \in C;$$

(2) nonexpansive on C if
$$\|Tx - Ty\| \leq \|x - y\|, \ \forall x, y \in C;$$

(3) firmly nonexpansive on C if
$$\|Tx - Ty\|^2 \leq \|x - y\|^2 - \|(I - T)x - (I - T)y\|^2, \ \forall x, y \in C,$$
which is equivalent to
$$\|Tx - Ty\|^2 \leq \langle Tx - Ty, \ x - y \rangle, \ \forall x, y \in C;$$

(4) σ-inverse strongly monotone ($\sigma - ism$) on C if there is $\sigma > 0$ such that
$$\langle Tx - Ty, \ x - y \rangle \geq \sigma \|Tx - Ty\|^2, \ \forall x, y \in C.$$

Definition 2.2. [2] Let $C \subseteq H$ be a nonempty closed convex set. For every element $x \in H$, there exists a unique nearest point in C, denoted by $P_C(x)$ such that
$$\|x - P_C(x)\| = \min\{\|x - y\| : y \in C\}.$$

The operator $P_C : H \to C$ is called a metric projection of H onto C.

Lemma 2.3. *[2] Let $C \subseteq H$ be a nonempty closed convex set. Then, the following assertions hold for any $x, y \in H$ and $z \in C$:*

(1) $\langle x - P_C(x),\ z - P_C(x)\rangle \leq 0$;

(2) $\|P_C(x) - P_C(y)\| \leq \|x - y\|$;

(3) $\|P_C(x) - P_C(y)\|^2 \leq \langle P_C(x) - P_C(y),\ x - y\rangle$;

(4) $\|P_C(x) - z\|^2 \leq \|x - z\|^2 - \|x - P_C(x)\|^2$.

(5) The mappings P_C and $I - P_C$ are both firmly nonexpansive and nonexpansive.

Lemma 2.4. *For all $x, y \in H$ and for all $\sigma \in \mathbb{R}$, we have*

(1) $\|x + y\|^2 \leq \|x\|^2 + 2\langle y,\ x + y\rangle$;

(2) $\|x + y\|^2 = \|x\|^2 + \|y\|^2 + 2\langle x,\ y\rangle$;

(3) $\langle x, y\rangle = \frac{1}{2}\|x\|^2 + \frac{1}{2}\|y\|^2 - \frac{1}{2}\|x - y\|^2$;

(4) $\|\sigma x + (1 - \sigma)y\|^2 = \sigma\|x\|^2 + (1 - \sigma)\|y\|^2 - \sigma(1 - \sigma)\|x - y\|^2$.

Definition 2.5. [2] A $f : H \to (-\infty, +\infty]$ be a given function. Then,

(1) The function f is proper if

$$\{x \in H : f(x) < +\infty\} \neq \emptyset.$$

(2) A proper function f is convex if for each $\sigma \in (0, 1)$,

$$f(\sigma x + (1 - \sigma)y) \leq \sigma f(x) + (1 - \sigma)f(y), \forall x, y \in H.$$

(3) f is σ-strongly convex, where $\sigma > 0$, if

$$f(\delta x + (1 - \delta)y) + \frac{\sigma}{2}\delta(1 - \delta)\|x - y\|^2 \leq \delta f(x) + (1 - \delta)f(y), \forall \delta \in (0, 1) \text{ and } \forall x, y \in H.$$

Moreover, f is σ-strongly convex if $f(x) - (\sigma/2)\|x\|^2$ is convex.

Definition 2.6. Let $f : H \to (-\infty, +\infty]$ be a proper function.

(1) A vector $\xi \in H$ is a subgradient of f at a point x if

$$f(y) \geq f(x) + \langle \xi,\ y - x\rangle,\ \forall y \in H.$$

(2) The set of all subgradients of f at $x \in H$, denoted by $\partial f(x)$, is called the subdifferential of f, and is defined by

$$\partial f(x) = \{\xi \in H : f(y) \geq f(x) + \langle \xi,\ y - x\rangle,\ \text{for each } y \in H\}.$$

(3) If $\partial f(x) \neq \emptyset$, f is said to be subdifferentiable at x. If the function f is continuously differentiable then $\partial f(x) = \{\nabla f(x)\}$.

Definition 2.7. Let $f : H \to (-\infty, +\infty]$ be a proper function. Then,

(1) f is lower semi-continuous (lsc) at x if $x_k \to x$ implies

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

(2) f is weakly lower semi-continuous (w-lsc) at x if $x_k \rightharpoonup x$ implies

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

(3) f is weakly/lower semi-continuous on H if it is weakly/lower semi-continuous at every point $x \in H$.

Lemma 2.8. *[2] Let $f : H \to (-\infty, +\infty]$ be a proper convex function. Then f is lower semi-continuous if and only if it is weakly lower semi-continuous.*

Lemma 2.9. *[2] Let $f : H \to (-\infty, +\infty]$ be a ϱ-strongly convex function. Then for all $x, y \in H$,*

$$f(y) \geq f(x) + \langle \xi,\ y - x\rangle + \frac{\varrho}{2}\|y - x\|^2,\ \xi \in \partial f(x).$$

Lemma 2.10. *[38] Let C and Q be closed convex subsets of real Hilbert spaces H_1 and H_2, respectively, and $f : H_1 \to (-\infty, +\infty]$ is given by $f(x) = \frac{1}{2}\|(I - P_Q)Tx\|^2$, where $T : H_1 \to H_2$ be a bounded linear operator. Then for $\sigma > 0$ and $x^* \in H_1$, the following statements are equivalent.*

(1) The point x^* solves the SFP (1.1).

(2) The point x^* is the fixed point of the mapping $P_C(I - \sigma\nabla f)$, i.e,

$$x^* = P_C(x^* - \sigma\nabla f(x^*)).$$

(3) The point x^* solves the variational inequality problem with respect to the gradient ∇f of f; i.e., find a point $z \in C$ such that

$$\langle \nabla f(z),\ x - z\rangle \geq 0, \forall x \in C.$$

Lemma 2.11. *[5] Let the function f be given as in Lemma 2.10. Then,*

(1) the function f is convex and weakly lower semi-continuous on H_1;

(2) $\nabla f(x) = T^*(I - P_Q)Tx$, for $x \in H_1$;

(3) ∇f is $\|T\|^2$-Lipschitz, i.e.,

$$\|\nabla f(x) - \nabla f(y)\| \leq \|T\|^2\|x - y\|,\ \forall x, y \in H_1.$$

Lemma 2.12. *[23, 27] Let $\{s_k\}$ and $\{\gamma_k\}$ be sequences of nonnegative real numbers, such that*

$$s_{k+1} \leq (1 - \sigma_k)s_k + \varepsilon_k + \gamma_k,\ \ k \geq 1,$$

where $\{\sigma_k\} \subset (0, 1)$ and $\{\varepsilon_k\}$ is a real sequence. Assume that $\sum_{k=1}^{\infty}\gamma_k < \infty$. Then the following results hold:

(1) If $\varepsilon_k \leq \sigma_k M$ for some $M \geq 0$, then $\{s_k\}$ is a bounded sequence;

(2) If $\sum_{k=1}^{\infty} \sigma_k = \infty$ and $\limsup_{k\to\infty} \frac{\varepsilon_k}{\sigma_k} \leq 0$, then $s_k \to 0$ as $k \to \infty$.

Lemma 2.13. *[18] Let $\{s_k\}$ be a non-negative real sequence, such that*

$$s_{k+1} \leq (1-\sigma_k)s_k + \sigma_k \mu_k, \ \ k \geq 1,$$
$$s_{k+1} \leq s_k - \phi_k + \varphi_k, \ \ k \geq 1,$$

where $\{\sigma_k\} \subset (0,1)$, $\{\phi_k\}$ is a non-negative, real sequence, and $\{\mu_k\}$ and $\{\varphi_k\}$ are real sequences such that

(1) $\sum_{k=1}^{\infty} \sigma_k = \infty$;

(2) $\lim_{k\to\infty} \varphi_k = 0$;

(3) $\lim_{m\to\infty} \phi_{k_m} = 0$ implies $\limsup_{m\to\infty} \mu_{k_m} \leq 0$ for any subsequence $\{k_m\}$ of $\{k\}$.

Then, $\lim_{k\to\infty} s_k = 0$.

3. The Proposed Algorithm with Convergence Analysis

In this section, we hereby present our proposed algorithm in the following manner. For simplicity, hereafter, denote $J_1 := \{1, 2, \dots, s\}$ and $J_2 := \{1, 2, \dots, r\}$.

We examine the MSSFPMOS (1.18), where the sets C_i $(i \in J_1)$ and Q_j $(j \in J_2)$ are defined as follows:

$$C_i = \{x \in H : c_i(x) \leq 0\} \quad \text{and} \quad Q_j = \{y \in H_j : q_j(y) \leq 0\}, \tag{3.1}$$

where, $c_i : H \to (-\infty, +\infty]$ for all $i \in J_1$ and $q_j : H_j \to (-\infty, +\infty]$ for all $j \in J_2$ are ϱ_i-strongly and σ_j-strongly convex functions, respectively. It is important to note that $\varrho_i \geq 0$ and $\sigma_j \geq 0$, allowing for the inclusion of cases where c_i or q_j are only convex.

Assumption 3.1. for the MSSFPMOS (1.18), we require the following standard assumptions.

(1) Both $c_i (i \in J_1)$ and $q_j (j \in J_2)$ are subdifferentiable on H and H_j, respectively.

(2) For any $x \in H$ and for each $i \in J_1$, a subgradient $\xi_i \in \partial c_i(x)$ can be calculated.

(3) For any $y \in H_j$ and for each $j \in J_2$, a subgradient $\eta_j \in \partial q_j(y)$ can be calculated.

(4) Both $\partial c_i (i \in J_1)$ and $\partial q_j (j \in J_2)$ are bounded operators (bounded on bounded sets).

According to Assumption 3.1, it is evident that the functions $c_i (i \in J_1)$ and $q_j (j \in J_2)$ are lower semi-continuous. Also, since $c_i (i \in J_1)$ and $q_j (j \in J_2)$ are convex, it can be inferred from Lemma 2.8 that $c_i (i \in J_1)$ and $q_j (j \in J_2)$ are weakly lower semi-continuous.

In our algorithm, given the k^{th} iterative point x_k, we construct "s" super-sets $\widetilde{C}_{ik}$ $(i \in J_1)$ and "r" super-sets $\widetilde{Q}_{jk}$ $(j \in J_2)$. These super-sets include the original sets C_i $(i \in J_1)$ and Q_j $(j \in J_2)$ respectively. The set $\widetilde{C}_{ik}$ $(i \in J_1)$ is constructed as

$$\widetilde{C}_{ik} = \Big\{x \in H : c_i(x_k) + \langle \xi_{ik}, \ x - x_k \rangle + \frac{\varrho_i}{2}\|x - x_k\|^2 \leq 0\Big\}, \tag{3.2}$$

where $\xi_{ik} \in \partial c_i(x_k)$. If $\varrho_i = 0$, then $\widetilde{C}_{ik}$ above is reduced to the following half-space

$$C_{ik} = \Big\{x \in H : c_i(x_k) + \langle \xi_{ik},\ x - x_k\rangle \le 0\Big\}. \tag{3.3}$$

If $\varrho_i > 0$, then for $i \in J_1$, $\widetilde{C}_{ik}$ can be defined by (see in [41])

$$\widetilde{C}_{ik} = \Big\{x \in H : \Big\|x - \Big(x_k - \frac{1}{\varrho_i}\xi_{ik}\Big)\Big\|^2 \le \frac{1}{\varrho_i^2}\|\xi_{ik}\|^2 - \frac{2}{\varrho_i}c_i(x_k)\Big\}$$

and it follows from the fact that $\widetilde{C}_{ik} \supseteq C_i \neq \emptyset$ $(i \in J_1)$ the set $\widetilde{C}_{ik}$ is nonempty. Furthermore, let $x^* \in C_i$ $(i \in J_1)$. Since each c_i $(i \in J_1)$ is ϱ_i-strongly convex function, it then follows from Lemma 2.9 that

$$c_i(x_k) + \langle \xi_{ik},\ x^* - x_k\rangle + \frac{\varrho_i}{2}\|x^* - x_k\|^2 \le c_i(x^*) \le 0,$$

which implies that for each $i \in J_1$

$$\frac{2}{\varrho_i}c_i(x_k) \le \frac{2}{\varrho_i}\|\xi_{ik}\|\|x_k - x^*\| - \|x_k - x^*\|^2 \le \frac{1}{\varrho_i^2}\|\xi_{ik}\|^2$$

which also yields $\frac{1}{\varrho_i^2}\|\xi_{ik}\|^2 - \frac{2}{\varrho_i}c_i(x_k) \ge 0$. Therefore, each $\widetilde{C}_{ik}$ $(i \in J_1)$ is a nonempty ball of radius $\sqrt{\frac{1}{\varrho_i^2}\|\xi_{ik}\|^2 - \frac{2}{\varrho_i}c_i(x_k)}$ centred at $x_k - \frac{1}{\varrho_i}\xi_{ik}$. The set $\widetilde{Q}_{jk}$ $(j \in J_2)$ is defined as

$$\widetilde{Q}_{jk} = \Big\{y \in H_j : q_j(T_jx_k) + \langle \eta_{jk},\ y - T_jx_k\rangle + \frac{\sigma_j}{2}\|y - T_jx_k\|^2 \le 0\Big\}, \tag{3.4}$$

where $\eta_{jk} \in \partial q_j(T_jx_k)$. If $\sigma_j = 0$, then $\widetilde{Q}_{jk}$ above is reduced to the following half-space

$$Q_{jk} = \Big\{y \in H_j : q_j(T_jx_k) + \langle \eta_{jk},\ y - T_jx_k\rangle \le 0\Big\}. \tag{3.5}$$

If $\sigma_j > 0$, then $\widetilde{Q}_{jk}$ above is nothing but a nonempty closed ball. Indeed, $\widetilde{Q}_{jk}$ is nonempty because $\widetilde{Q}_{jk} \supseteq Q_j \neq \emptyset$ $(j \in J_2)$. Similarly, for all $k \ge 0$ and for each $j \in J_2$, observe that

$$\widetilde{Q}_{jk} = \Big\{y \in H_j : \Big\|y - \Big(T_jx_k - \frac{1}{\sigma_j}\eta_{jk}\Big)\Big\|^2 \le \frac{1}{\sigma_j^2}\|\eta_{jk}\|^2 - \frac{2}{\sigma_j}q_j(T_jx_k)\Big\}.$$

That is, each $\widetilde{Q}_{jk}$ $(j \in J_2)$ is also a nonempty closed ball of radius $\sqrt{\frac{1}{\sigma_j^2}\|\eta_{jk}\|^2 - \frac{2}{\sigma_j}q_j(T_jx_k)}$ centred at $T_jx_k - \frac{1}{\sigma_j}\eta_{jk}$. Therefore, both $\widetilde{C}_{ik}$ and $\widetilde{Q}_{jk}$ are nothing but nonempty closed balls and it is easy to verify that (see [41]) $\widetilde{C}_{ik} \supseteq C_i$ $(i \in J_1)$ and $\widetilde{Q}_{jk} \supseteq Q_j$ $(j \in J_2)$ hold for every $k \ge 0$.

In order to enhance the efficiency and ease of implementation of the algorithm, we utilize metric projections onto the balls $\widetilde{C}_{ik}$ and $\widetilde{Q}_{jk}$ as defined in (3.2) and (3.4), respectively, instead of the given sets C_i and Q_j. Additionally, to circumvent the need for estimating the operator norm (which can be a challenging task), we introduce a self-adaptive approach for updating the step-size in solving the MSSFPMOS (1.18) within the framework of infinite dimensional real Hilbert spaces.

Here, we present our proposed self-adaptive relaxed CQ algorithm for solving the MSSFPMOS (1.18). In building our algorithm, we also require the following assumption.

Assumption 3.2. Let $\{\upsilon_k\}$, $\{\rho_k\}$, and $\{\varepsilon_k\}$ be three sequences satisfy the conditions:

(a1) $\{\upsilon_k\} \subset (0,1)$ such that $\lim\limits_{k\to\infty} \upsilon_k = 0$ and $\sum\limits_{k=0}^{\infty} \upsilon_k = \infty$.

(a2) $\{\rho_k\} \subset (0,2)$ such that $\liminf\limits_{k\to\infty} \rho_k(2-\rho_k) > 0$.

(a3) $\{\varepsilon_k\} \subset [0,\varepsilon)$, where $\varepsilon \in (0,1)$ such that $\varepsilon_k = \mathbf{o}(\upsilon_k)$.

Algorithm 1:

Step 0. Choose three sequences $\{\upsilon_k\}$, $\{\rho_k\}$, and $\{\varepsilon_k\}$ satisfying the conditions in Assumption 3.2 and select $\beta \in [0,1)$. Fix $u \in H$ and let $x_0, x_1 \in H$ be arbitrary initial guesses and set $k := 1$. Take the weights α_i^k $(i \in J_1) > 0$ and the constant parameters δ_j $(j \in J_2) > 0$ such that

$$\sum_{i=1}^{s} \alpha_i^k = 1 \quad \text{and} \quad \inf_{i\in I_k} \alpha_i^k > \alpha > 0, \text{ where } I_k = \{i \in J_1 : \alpha_i^k > 0\}, \quad \text{and} \quad \sum_{j=1}^{r} \delta_j = 1.$$

Step 1. Given the iterates $x_{k-1}, x_k \in H$, then compute y_k via the manner

$$y_k = x_k + \beta_k(x_k - x_{k-1}) \tag{3.6}$$

where β_k is chosen in $[0, \overline{\beta_k})$ and

$$\overline{\beta_k} := \begin{cases} \min\left\{\beta, \dfrac{\varepsilon_k}{\max\left\{\|x_k-x_{k-1}\|^2, \|x_k-x_{k-1}\|\right\}}\right\}, & \text{if } x_k \neq x_{k-1}, \\ \beta, & \text{if } x_k = x_{k-1}. \end{cases} \tag{3.7}$$

Step 2. Compute the next iterate x_{k+1} via the manner

$$x_{k+1} = \sum_{i=1}^{s} \alpha_i^k P_{\widetilde{C}_{ik}}\left(\upsilon_k u + (1-\upsilon_k)\left(y_k - \tau_k \sum_{j=1}^{r} \delta_j T_j^*\left(I - P_{\widetilde{Q}_{jk}}\right) T_j y_k\right)\right)$$

where $\widetilde{C}_{ik}$ and $\widetilde{Q}_{jk}$ are the balls given as in (3.2) and (3.4), respectively and the stepsize τ_k is self-adaptively updated via

$$\tau_k := \frac{\rho_k \sum_{j=1}^{r} \delta_j \left\|\left(I - P_{\widetilde{Q}_{jk}}\right) T_j y_k\right\|^2}{\lambda_k^2} \tag{3.8}$$

where

$$\lambda_k := \max\left\{1, \left\|\sum_{j=1}^{r} \delta_j T_j^*\left(I - P_{\widetilde{Q}_{jk}}\right) T_j y_k\right\|\right\}.$$

Step 3. If $x_{k+1} = y_k$, then stop; otherwise, set $k := k+1$ and return to **Step 1.**

Remark 3.3. We note that the positive sequence $\{\varepsilon_k\}$ satisfies the conditions:

$$\lim_{k\to\infty} \frac{\varepsilon_k}{\upsilon_k} = 0 \quad \text{and} \quad \sum_{k=1}^{\infty} \varepsilon_k < \infty.$$

Remark 3.4. From (3.7), we have that $\beta_k\|x_k - x_{k-1}\|^2$, $\beta_k\|x_k - x_{k-1}\| \le \varepsilon_k$, $\forall k \ge 1$. This together with Remark 3.3 give

$$\lim_{k\to\infty} \frac{\beta_k}{\upsilon_k}\|x_k - x_{k-1}\|^2 \le \lim_{k\to\infty} \frac{\varepsilon_k}{\upsilon_k} = 0 \quad \text{and} \quad \lim_{k\to\infty} \frac{\beta_k}{\upsilon_k}\|x_k - x_{k-1}\| \le \lim_{k\to\infty} \frac{\varepsilon_k}{\upsilon_k} = 0.$$

Thus, there exist two positive constant M_1 and M_2 such that

$$\frac{\beta_k}{\upsilon_k}\|x_k - x_{k-1}\|^2 \le M_1 \quad \text{and} \quad \frac{\beta_k}{\upsilon_k}\|x_k - x_{k-1}\| \le M_2.$$

Furthermore, by assuming (3.7), we have that

$$\sum_{k=1}^{\infty} \beta_k\|x_k - x_{k-1}\|^2 < \infty \quad \text{and} \quad \sum_{k=1}^{\infty} \beta_k\|x_k - x_{k-1}\| < \infty, \tag{3.9}$$

which implies

$$\lim_{k\to\infty} \beta_k\|x_k - x_{k-1}\|^2 = 0 \quad \text{and} \quad \lim_{k\to\infty} \beta_k\|x_k - x_{k-1}\| = 0. \tag{3.10}$$

Lemma 3.5. *Assume that the sequences $\{\upsilon_k\}$, $\{\rho_k\}$, and $\{\varepsilon_k\}$ satisfy all the conditions in Assumption 3.2. Then the sequence $\{x_k\}$ generated by Algorithm 1 is bounded.*

Proof. Let $x^* \in \Omega$, then we have the following estimation

$$\begin{aligned}
&\left\| y_k - \tau_k \sum_{j=1}^{r} \delta_j T_j^* \left(I - P_{\widetilde{Q}_{jk}}\right) T_j y_k - x^* \right\|^2 \\
= &\left\| (y_k - x^*) - \tau_k \sum_{j=1}^{r} \delta_j T_j^* \left(I - P_{\widetilde{Q}_{jk}}\right) T_j y_k \right\|^2 \\
= &\|y_k - x^*\|^2 + \tau_k^2 \left\| \sum_{j=1}^{r} \delta_j T_j^* \left(I - P_{\widetilde{Q}_{jk}}\right) T_j y_k \right\|^2 \\
&-2\tau_k \left\langle \sum_{j=1}^{r} \delta_j T_j^* \left(I - P_{\widetilde{Q}_{jk}}\right) T_j y_k, y_k - x^* \right\rangle.
\end{aligned} \tag{3.11}$$

Note that for each $j \in J_2$, $I - P_{\widetilde{Q}_{jk}}$ is firmly nonexpansive and $\sum_{j=1}^{r} \delta_j T_j^* \left(I - P_{\widetilde{Q}_{jk}}\right) T_j x^* = 0$. Hence, it follows from lemma 2.3 that

$$\left\langle \tau_k \sum_{j=1}^{r} \delta_j T_j^* \left(I - P_{\widetilde{Q}_{jk}}\right) T_j y_k, y_k - x^* \right\rangle = \tau_k \sum_{j=1}^{r} \delta_j \left\langle T_j^* \left(I - P_{\widetilde{Q}_{jk}}\right) T_j y_k, y_k - x^* \right\rangle$$

$$
\begin{aligned}
&= \tau_k \sum_{j=1}^{r} \delta_j \Big\langle \Big(I - P_{\widetilde{Q}_{jk}}\Big) T_j y_k, T_j y_k - T_j x^* \Big\rangle \\
&\geq \tau_k \sum_{j=1}^{r} \delta_j \Big\| \Big(I - P_{\widetilde{Q}_{jk}}\Big) T_j y_k \Big\|^2, \qquad (3.12)
\end{aligned}
$$

Substituting (3.12) in to (3.11), we get

$$
\begin{aligned}
&\Big\| y_k - \tau_k \sum_{j=1}^{r} \delta_j T_j^* \Big(I - P_{\widetilde{Q}_{jk}}\Big) T_j y_k - x^* \Big\|^2 \\
\leq\ & \|y_k - x^*\|^2 + \tau_k^2 \Big\| \sum_{j=1}^{r} \delta_j T_j^* \Big(I - P_{\widetilde{Q}_{jk}}\Big) T_j y_k \Big\|^2 - 2\tau_k \sum_{j=1}^{r} \delta_j \Big\| \Big(I - P_{\widetilde{Q}_{jk}}\Big) T_j y_k \Big\|^2.
\end{aligned} \qquad (3.13)
$$

(3.13) together with (3.8) and $\Big\| \sum_{j=1}^{r} \delta_j T_j^* \Big(I - P_{\widetilde{Q}_{j,n}}\Big) T_j y_k \Big\| \leq \lambda_k$ implies that

$$
\begin{aligned}
&\Big\| y_k - \tau_k \sum_{j=1}^{r} \delta_j T_j^* \Big(I - P_{\widetilde{Q}_{jk}}\Big) T_j y_k - x^* \Big\|^2 \\
\leq\ & \|y_k - x^*\|^2 + \tau_k^2 \lambda_k^2 - 2\tau_k \sum_{j=1}^{r} \delta_j \Big\| \Big(I - P_{\widetilde{Q}_{jk}}\Big) T_j y_k \Big\|^2 \\
=\ & \|y_k - x^*\|^2 - \rho_k (2 - \rho_k) \frac{\Big(\sum_{j=1}^{r} \delta_j \Big\| \Big(I - P_{\widetilde{Q}_{jk}}\Big) T_j y_k \Big\|^2 \Big)^2}{\lambda_k^2} \\
\leq\ & \|y_k - x^*\|^2. \qquad (3.14)
\end{aligned}
$$

By Lemma 2.3, we also obtain the following estimation.

$$
\begin{aligned}
\|x_{k+1} - x^*\| &= \Big\| \sum_{i=1}^{s} \alpha_i^k P_{\widetilde{C}_{ik}} \Big(\upsilon_k u + (1 - \upsilon_k) \Big(y_k - \tau_k \sum_{j=1}^{r} \delta_j T_j^* \Big(I - P_{\widetilde{Q}_{jk}}\Big) T_j y_k \Big) \Big) - x^* \Big\| \\
&= \Big\| \sum_{i=1}^{s} \alpha_i^k P_{\widetilde{C}_{ik}} \Big(\upsilon_k u + (1 - \upsilon_k) \Big(y_k - \tau_k \sum_{j=1}^{r} \delta_j T_j^* \Big(I - P_{\widetilde{Q}_{jk}}\Big) T_j y_k \Big) \Big) - \sum_{i=1}^{s} \alpha_i^k P_{\widetilde{C}_{ik}} x^* \Big\| \\
&\leq \Big\| \upsilon_k u + (1 - \upsilon_k) \Big(y_k - \tau_k \sum_{j=1}^{r} \delta_j T_j^* \Big(I - P_{\widetilde{Q}_{jk}}\Big) T_j y_k \Big) - x^* \Big\| \\
&= \Big\| \upsilon_k (u - x^*) + (1 - \upsilon_k) \Big[\Big(y_k - \tau_k \sum_{j=1}^{r} \delta_j T_j^* \Big(I - P_{\widetilde{Q}_{jk}}\Big) T_j y_k \Big) - x^* \Big] \Big\| \\
&\leq (1 - \upsilon_k) \Big\| y_k - \tau_k \sum_{j=1}^{r} \delta_j T_j^* \Big(I - P_{\widetilde{Q}_{jk}}\Big) T_j y_k - x^* \Big\| + \upsilon_k \|u - x^*\|. \qquad (3.15)
\end{aligned}
$$

From (3.6), we have that

$$
\|y_k - x^*\| = \|x_k + \beta_k (x_k - x_{k-1}) - x^*\| \leq \|x_k - x^*\| + \beta_k \|x_k - x_{k-1}\|. \qquad (3.16)
$$

Combining (3.14)-(3.16), we obtain

$$\begin{aligned}\|x_{k+1}-x^*\| &\leq (1-\upsilon_k)\|y_k-x^*\|+\upsilon_k\|u-x^*\| \\ &\leq (1-\upsilon_k)[\|x_k-x^*\|+\beta_k\|x_k-x_{k-1}\|]+\upsilon_k\|u-x^*\| \\ &\leq (1-\upsilon_k)\|x_k-x^*\|+\upsilon_k[\Sigma_k+\|u-x^*\|]. \end{aligned} \tag{3.17}$$

where $\Sigma_k=(1-\upsilon_k)\frac{\beta_k}{\upsilon_k}\|x_k-x_{k-1}\|$ and by Remark 3.4, we have

$$\lim_{k\to\infty}\Sigma_k=\lim_{k\to\infty}(1-\upsilon_k)\frac{\beta_k}{\upsilon_k}\|x_k-x_{k-1}\|=0. \tag{3.18}$$

This implies that the sequence $\{\Sigma_k\}$ is bounded. Setting

$$M=\max\left\{\sup_{k\in\mathbb{N}}\Sigma_k,\ \|u-x^*\|\right\}, \tag{3.19}$$

as well as using Lemma 2.12, we conclude that the sequence $\{\|x_k-x^*\|\}$ is bounded. This shows that the sequence $\{x_k\}$ is bounded and so are $\{y_k\}$ and $\{T_jy_k\}$ for each $j\in J_2$. This completes the proof. ∎

Theorem 3.6. *Assume that the sequences $\{\upsilon_k\}$, $\{\rho_k\}$, and $\{\varepsilon_k\}$ satisfy the conditions (a1)-(a3) of Assumption 3.2. Then the sequence $\{x_k\}$ generated by Algorithm 1 converges strongly to the point $x^*\in\Omega$ of the MSSFPOMS (1.18), where $x^*=P_\Omega u$.*

Proof. Let $x^*\in\Omega$. Since $\{\|x_k-x^*\|\}$ is bounded, assume that there exists a constant $K_1>0$ such that $\|x_k-x^*\|\leq K_1$. Hence, it follows from (3.16)

$$\begin{aligned}\|y_k-x^*\|^2 &\leq \|x_k-x^*\|^2+\beta_k^2\|x_k-x_{k-1}\|^2+2\beta_k\|x_k-x_{k-1}\|.\|x_k-x^*\| \\ &\leq \|x_k-x^*\|^2+\beta_k\|x_k-x_{k-1}\|^2+2K_1.\beta_k\|x_k-x_{k-1}\|. \end{aligned} \tag{3.20}$$

This together with (3.14) further gives

$$\begin{aligned}\|x_{k+1}-x^*\|^2 &= \left\|\sum_{i=1}^{s}\alpha_i^kP_{\widetilde{C}_{ik}}\left(\upsilon_ku+(1-\upsilon_k)\left(y_k-\tau_k\sum_{j=1}^{r}\delta_jT_j^*\left(I-P_{\widetilde{Q}_{jk}}\right)T_jy_k\right)\right)-x^*\right\|^2 \\
&= \left\|\sum_{i=1}^{s}\alpha_i^kP_{\widetilde{C}_{ik}}\left(\upsilon_ku+(1-\upsilon_k)\left(y_k-\tau_k\sum_{j=1}^{r}\delta_jT_j^*\left(I-P_{\widetilde{Q}_{jk}}\right)T_jy_k\right)\right)-\sum_{i=1}^{s}\alpha_i^kP_{\widetilde{C}_{ik}}x^*\right\|^2 \\
&\leq \left\|\upsilon_ku+(1-\upsilon_k)\left(y_k-\tau_k\sum_{j=1}^{r}\delta_jT_j^*\left(I-P_{\widetilde{Q}_{jk}}\right)T_jy_k\right)-x^*\right\|^2 \\
&= \left\|\upsilon_k(u-x^*)+(1-\upsilon_k)\left[\left(y_k-\tau_k\sum_{j=1}^{r}\delta_jT_j^*\left(I-P_{\widetilde{Q}_{jk}}\right)T_jy_k\right)-x^*\right]\right\|^2 \\
&\leq \left\|(1-\upsilon_k)\left[y_k-\tau_k\sum_{j=1}^{r}\delta_jT_j^*\left(I-P_{\widetilde{Q}_{jk}}\right)T_jy_k-x^*\right]\right\|^2+2\upsilon_k\langle u-x^*,x_{k+1}-x^*\rangle \\
&\leq (1-\upsilon_k)\left[\|y_k-x^*\|^2-\rho_k(2-\rho_k)\frac{\left(\sum_{j=1}^{r}\delta_j\left\|\left(I-P_{\widetilde{Q}_{jk}}\right)T_jy_k\right\|^2\right)^2}{\lambda_k^2}\right]\end{aligned}$$

$$
\begin{aligned}
&+2\upsilon_k\langle u-x^*, x_{k+1}-x^*\rangle \\
&\le (1-\upsilon_k)\Bigg[\|x_k-x^*\|^2+\beta_k\|x_k-x_{k-1}\|^2+2K_1.\beta_k\|x_k-x_{k-1}\| \\
&-\rho_k(2-\rho_k)\frac{\left(\sum_{j=1}^r \delta_j \left\|\left(I-P_{\widetilde{Q}_{jk}}\right)T_j y_k\right\|^2\right)^2}{\lambda_k^2}\Bigg]+2\upsilon_k\langle u-x^*, x_{k+1}-x^*\rangle. \qquad (3.21)
\end{aligned}
$$

Now, we can see from (3.21) that

$$
\begin{aligned}
\|x_{k+1}-x^*\|^2 &\le (1-\upsilon_k)\|x_k-x^*\|^2+\upsilon_k\Big[\tfrac{\beta_k}{\upsilon_k}\|x_k-x_{k-1}\|^2+2K_1.\tfrac{\beta_k}{\upsilon_k}\|x_k-x_{k-1}\| \\
&\quad +2\langle u-x^*, x_{k+1}-x^*\rangle\Big], \\
\|x_{k+1}-x^*\|^2 &\le \|x_k-x^*\|^2-\rho_k(2-\rho_k)\frac{\left(\sum_{j=1}^r \delta_j \left\|\left(I-P_{\widetilde{Q}_{jk}}\right)T_j y_k\right\|^2\right)^2}{\lambda_k^2} \\
&\quad +\beta_k\|x_k-x_{k-1}\|^2+2K_1.\beta_k\|x_k-x_{k-1}\|+2\upsilon_k\langle u-x^*, x_{k+1}-x^*\rangle. \qquad (3.22)
\end{aligned}
$$

Then (3.21) can be transformed to the inequalities

$$
\begin{aligned}
s_{k+1} &\le (1-\upsilon_k)s_k+\upsilon_k\mu_k,\ \ k\ge 1, \\
s_{k+1} &\le s_k-\phi_k+\varphi_k,\ \ k\ge 1,
\end{aligned} \qquad (3.23)
$$

where

$$
\begin{aligned}
s_k &= \|x_{k+1}-x^*\|^2; \\
\mu_k &= \tfrac{\beta_k}{\upsilon_k}\|x_k-x_{k-1}\|^2+2K_1.\tfrac{\beta_k}{\upsilon_k}\|x_k-x_{k-1}\|+2\langle u-x^*, x_{k+1}-x^*\rangle; \\
\phi_k &= \rho_k(2-\rho_k)\frac{\left(\sum_{j=1}^r \delta_j \left\|\left(I-P_{\widetilde{Q}_{jk}}\right)T_j y_k\right\|^2\right)^2}{\lambda_k^2}; \\
\varphi_k &= \beta_k\|x_k-x_{k-1}\|^2+2K_1.\beta_k\|x_k-x_{k-1}\|+2\upsilon_k\langle u-x^*, x_{k+1}-x^*\rangle.
\end{aligned} \qquad (3.24)
$$

Moreover, by Assumption 3.2, we have that

$$
\sum_{k=0}^{\infty}\upsilon_k=\infty, \qquad (3.25)
$$

$$
\lim_{k\to\infty}\varphi_k=\lim_{k\to\infty}\left[\beta_k\|x_k-x_{k-1}\|^2+2K_1.\beta_k\|x_k-x_{k-1}\|+2\upsilon_k\langle u-x^*, x_{k+1}-x^*\rangle\right]=0. \qquad (3.26)
$$

Let $\{k_m\}$ be a subsequence of $\{k\}$ and suppose

$$
\limsup_{m\to\infty}\phi_{k_m}\le 0, \qquad (3.27)
$$

which further yields

$$
\lim_{m\to\infty}\left[\frac{\left(\sum_{j=1}^r \delta_j \left\|\left(I-P_{\widetilde{Q}_{jk_m}}\right)T_j y_{k_m}\right\|^2\right)^2}{\lambda_{k_m}^2}\right]=0 \qquad (3.28)
$$

which also implies that

$$\lim_{m\to\infty}\left[\frac{\sum_{j=1}^{r}\delta_j\left\|\left(I-P_{\widetilde{Q}_{jk_m}}\right)T_j y_{k_m}\right\|^2}{\lambda_{k_m}}\right]=0. \tag{3.29}$$

Since the sequence $\{y_{k_m}\}$ is bounded and by the Lipschitz continuity of the $\left(I-P_{\widetilde{Q}_{jk_m}}\right)T_j y_{k_m}$ for each $j\in J_2$ and for all $m\in\mathbb{N}$, the sequence $\left\{\left\|\sum_{j=1}^{r}\delta_j T_j^*\left(I-P_{\widetilde{Q}_{jk_m}}\right)T_j y_{k_m}\right\|\right\}$ is bound. And hence the sequence $\{\lambda_{k_m}\}$ is bounded as well. Therefore, we get from (3.29) that

$$\lim_{m\to\infty}\sum_{j=1}^{r}\delta_j\left\|\left(I-P_{\widetilde{Q}_{jk_m}}\right)T_j y_{k_m}\right\|^2=0, \tag{3.30}$$

which implies for each $j\in J_2$ that

$$\lim_{m\to\infty}\left\|\left(I-P_{\widetilde{Q}_{jk_m}}\right)T_j y_{k_m}\right\|=0. \tag{3.31}$$

Next, we prove that $\omega_w(x_k)\subset\Omega$. Since $\{x_k\}$ is bounded, $\omega_w(x_k)\neq\emptyset$. Let $\bar{x}\in\omega_w(x_k)$; then we may assume that there exists a subsequence $\{x_{k_m}\}$ of $\{x_k\}$ such that $x_{k_m}\rightharpoonup\bar{x}$. Furthermore,

$$\|y_k-x_k\|=\|x_k+\beta_k(x_k-x_{k-1})-x_k\|=\beta_k\|x_k-x_{k-1}\|\to 0, \tag{3.32}$$

and hence $y_{k_m}\rightharpoonup\bar{x}$, and since each T_j for $j\in J_2$ is linear and bounded, $T_j y_{k_m}\rightharpoonup T_j\bar{x}$. Since ∂q_j for each $j\in J_2$ is bounded on bounded set, we may assume that there is a constant $\eta>0$ such that $\|\eta_{jk_m}\|\le\eta$, where $\eta_{jk_m}\in\partial q_j(T_j y_{k_m})$ for each $j\in J_2$. That is the sequence $\{\eta_{jk_m}\}$ is bounded. Note that $P_{\widetilde{Q}_{jk_m}}(T_j y_{k_m})\in\widetilde{Q}_{jk_m}$ for each $j\in J_2$. Now, it follows from (3.4) and (3.31) for all $j\in J_2$ and as $m\to\infty$ that

$$\begin{aligned} q_j\left(T_j y_{k_m}\right) &\le \left\langle\eta_{jk_m},\ T_j y_{k_m}-P_{\widetilde{Q}_{jk_m}}(T_j y_{k_m})\right\rangle-\frac{\sigma_j}{2}\left\|T_j y_{k_m}-P_{\widetilde{Q}_{jk_m}}(T_j y_{k_m})\right\|^2\\ &\le \left\langle\eta_{jk_m},\ T_j y_{k_m}-P_{\widetilde{Q}_{jk_m}}(T_j y_{k_m})\right\rangle\\ &\le \left\|\eta_{jk_m}\right\|\left\|\left(I-P_{\widetilde{Q}_{jk_m}}\right)T_j y_{k_m}\right\|\\ &\le \eta\left\|\left(I-P_{\widetilde{Q}_{jk_m}}\right)T_j y_{k_m}\right\|\to 0. \end{aligned} \tag{3.33}$$

By the weakly lower semi-continuity of q_j together with (3.33) we get for all $j\in J_2$ that

$$q_j(T_j\bar{x})\le\liminf_{m\to\infty}q_j\left(T_j y_{k_m}\right)\le\lim_{m\to\infty}\eta\left\|\left(I-P_{\widetilde{Q}_{jk_m}}\right)T_j y_{k_m}\right\|=0.$$

It turns out that, $T_j\bar{x}\in Q_j$, $\forall j\in J_2$.

Observe that

$$\|x_{k+1}-y_k\| \le (1-\upsilon_k)\left\|y_k-\tau_k\sum_{j=1}^{r}\delta_j T_j^*\left(I-P_{\widetilde{Q}_{jk}}\right)T_j y_k-y_k\right\|+\upsilon_k\|u-y_k\|$$

$$\begin{aligned} &\leq (1-\upsilon_k)\tau_k\lambda_k + \upsilon_k\|u-y_k\| \\ &= (1-\upsilon_k)\frac{\rho_k \sum_{j=1}^{r}\delta_j \left\|\left(I-P_{\widetilde{Q}_{jk}}\right)T_j y_k\right\|^2}{\lambda_k} + \upsilon_k\|u-y_k\| \to 0. \end{aligned} \tag{3.34}$$

Since ∂c_i for each $i \in J_1$ is bounded on bounded set, we may again assume that for all $k_m \geq 0$, there is a constant $\xi > 0$ such that $\|\xi_{ik_m}\| \leq \xi$, where $\xi_{ik_m} \in \partial c_i(y_{k_m})$ for each $i \in J_1$. That is the sequence $\{\xi_{ik_m}\}$ is bounded. Since $(x_{k_m+1} - \upsilon_{k_m}u) \in \widetilde{C}_{ik_m}$ for all $i \in J_1$ and by (3.2) and (3.34), we have for all $i \in J_1$ as $m \to \infty$ that

$$\begin{aligned} c_i(y_{k_m}) &\leq \left\langle \xi_{ik_m},\ y_{k_m} + \upsilon_{k_m}u - x_{k_m+1}\right\rangle - \frac{\varrho_i}{2}\left\|y_{k_m} + \upsilon_{k_m}u - x_{k_m+1}\right\|^2 \\ &\leq \left\|\xi_{ik_m}\right\|\left\|y_{k_m} + \upsilon_{k_m}u - x_{k_m+1}\right\| \\ &\leq \xi\left\|y_{k_m} + \upsilon_{k_m}u - x_{k_m+1}\right\| \to 0. \end{aligned} \tag{3.35}$$

By the weakly lower semi-continuity of c_i together with (3.35) we get for all $i \in J_1$ that

$$c_i(\bar{x}) \leq \liminf_{m\to\infty} c_i(y_{k_m}) \leq \lim_{m\to\infty} \xi\left\|y_{k_m} + \upsilon_{k_m}u - x_{k_m+1}\right\| = 0,$$

Thus, $\bar{x} \in C_i$, $\forall i \in J_1$, i.e., $\omega_w(x_k) \subset \Omega$.

Next, we show that $\lim_{k\to\infty}\|x_{k+1} - x_k\| = 0$. Indeed, we have

$$\begin{aligned} \|x_{k+1} - x_k\| &\leq (1-\upsilon_k)\left\|y_k - \tau_k\sum_{j=1}^{r}\delta_j T_j^*\left(I-P_{\widetilde{Q}_{jk}}\right)T_j y_k - x_k\right\| + \upsilon_k\|u-x_k\| \\ &\leq (1-\upsilon_k)\left[\|y_k - x_k\| + \tau_k\left\|\sum_{j=1}^{r}\delta_j T_j^*\left(I-P_{\widetilde{Q}_{jk}}\right)T_j y_k\right\|\right] + \upsilon_k\|u-x_k\| \\ &\leq (1-\upsilon_k)\left[\|y_k - x_k\| + \tau_k\lambda_k\right] + \upsilon_k\|u-x_k\| \\ &= (1-\upsilon_k)\left[\|y_k - x_k\| + \frac{\rho_k\sum_{j=1}^{r}\delta_j\left\|\left(I-P_{\widetilde{Q}_{jk}}\right)T_j y_k\right\|^2}{\lambda_k}\right] \\ &+ \upsilon_k\|u-x_k\| \to 0. \end{aligned} \tag{3.36}$$

For $x^* = P_\Omega u$ and $x_{k_m} \rightharpoonup \bar{x} \in \Omega$, it follows from Lemma 2.3 that $\langle u - x^*, \bar{x} - x^*\rangle \leq 0$. So,

$$\limsup_{k\to\infty}\langle u - x^*, x_k - x^*\rangle = \limsup_{m\to\infty}\langle u - x^*, x_{k_m} - x^*\rangle = \langle u - x^*, \bar{x} - x^*\rangle \leq 0. \tag{3.37}$$

Hence,

$$\limsup_{k\to\infty}\langle u - x^*, x_{k+1} - x^*\rangle = \limsup_{k\to\infty}(\langle u - x^*, x_{k+1} - x_k\rangle + \langle u - x^*, x_k - x^*\rangle) \leq 0. \tag{3.38}$$

And then

$$\limsup_{k\to\infty}\mu_k = \limsup_{k\to\infty}\left[\frac{\beta_k}{\upsilon_k}\|x_k - x_{k-1}\|^2 + 2K_1.\frac{\beta_k}{\upsilon_k}\|x_k - x_{k-1}\| + 2\langle u - x^*, x_{k+1} - x^*\rangle\right]$$

$$\leq 0. \tag{3.39}$$

Therefore, applying Lemma 2.13, we conclude that the sequence $\{x_k\}$ converges strongly to the point $x^* \in \Omega$, where $x^* = P_\Omega u$. This completes the proof. ∎

Remark 3.7. For the special case, where $s = 1 = r$, the MSSFPMOS (1.18) becomes the SFP (1.1). Thus, it is worth to mention that, we have corollary 3.8 for solving the SFP (1.1), an immediate consequence of Theorem 3.6.

Corollary 3.8. *Let H_1 and H_2 be two real Hilbert spaces and let $T : H_1 \to H_2$ be bounded linear operator. Let C and Q be nonempty, closed and convex subsets of H_1 and H_2, respectively. Assume that $\Gamma = C \cap T^{-1}(Q) \neq \emptyset$. Let $\{x_k\}$ be the sequence generated by Algortihm 2. Suppose that the sequences $\{\upsilon_k\}$, $\{\rho_k\}$, and $\{\varepsilon_k\}$ satisfy all conditions in Assumption 3.2. Then, the sequence $\{x_k\}$ converges strongly to the point $x^* \in \Gamma$ of the SFP (1.1), where $x^* = P_\Gamma u$.*

Algorithm 2:

Step 0. Choose three sequences $\{\upsilon_k\}$, $\{\rho_k\}$, and $\{\varepsilon_k\}$ satisfying the conditions in Assumption 3.2 and select $\beta \in [0, 1)$. Fix $u \in H$ and let $x_0, x_1 \in H_1$ be arbitrary initial guesses and set $k := 1$.

Step 1. Given the iterates $x_{k-1}, x_k \in H_1$, then compute y_k via the manner

$$y_k = x_k + \beta_k(x_k - x_{k-1})$$

where β_k is chosen in $[0, \overline{\beta_k})$ and

$$\overline{\beta_k} := \begin{cases} \min\left\{\beta, \ \dfrac{\varepsilon_k}{\max\left\{\|x_k - x_{k-1}\|^2, \ \|x_k - x_{k-1}\|\right\}}\right\}, & \text{if } x_k \neq x_{k-1}, \\ \beta, & \text{if } x_k = x_{k-1}. \end{cases}$$

Step 2. Compute the next iterate x_{k+1} via the manner

$$x_{k+1} = P_{\widetilde{C}_k}\Big(\upsilon_k u + (1 - \upsilon_k)\Big(y_k - \tau_k T^*(I - P_{\widetilde{Q}_k})Ty_k\Big)\Big)$$

where $\widetilde{C}_k$ and $\widetilde{Q}_k$ are the balls as in (1.11) and (1.12), respectively and τ_k is self-adaptively updated via

$$\tau_k := \frac{\rho_k \left\|(I - P_{\widetilde{Q}_k})Ty_k\right\|^2}{\lambda_k^2}$$

where

$$\lambda_k := \max\left\{1, \ \left\|T^*(I - P_{\widetilde{Q}_k})Ty_k\right\|\right\}.$$

Step 3. If $x_{k+1} = y_k$, then stop; otherwise, set $k := k + 1$ and return to **Step 1.**

---

4. Application to the Generalized Split Feasibility Problem

In this section, we present an application of Theorem 3.6 for solving generalized split feasibility problem (another generalization of the SFP) in Hilbert spaces. To begin, we will provide a brief overview of the generalized split feasibility problem.

Very recently, Reich and Tuyen [30] first introduced and studied the following generalized split feasibility problem. Let H_j, $j = 1, 2, \dots, r$, be real Hilbert spaces and C_j, $j = 1, 2, \dots, r$, be closed and convex subsets of H_j, respectively. Let $B_j : H_j \to H_{j+1}$, $j = 1, 2, \dots, r-1$, be bounded linear operators such that

$$S := C_1 \cap B_1^{-1}(C_2) \cap \cdots \cap B_1^{-1}\left(B_2^{-1} \dots \left(B_{r-1}^{-1}(C_r)\right)\right) \neq \emptyset.$$

The generalized split feasibility problem (GSFP, for short) [30] is to find an element

$$x^* \in S, \tag{4.1}$$

that is $x^* \in C_1$, $B_1 x^* \in C_2, \dots, B_{r-1}B_{r-2}\dots B_1 x^* \in C_r$. In [30], Reich and Tuyen proved a strong convergence theorem for a modification of the CQ-algorithm which solves the GSFP (4.1). For more details on the GSFP (4.1), one can go through the paper [30].

Remark 4.1. It is readily seen that, letting $H = H_1, C = C_1, Q_j = C_{j+1}, 1 \leq j \leq r-1$, $A_1 = B_1, A_2 = B_2B_1, \dots,$ and $A_{r-1} = B_{r-1}B_{r-2}B_{r-3}\dots B_2B_1$, then the SFPMOS (1.15) becomes the GSFP (4.1). Moreover, for the special case where $r = 1$, both the GSFP (4.1) and the SFPMOS (1.15) become the SFP (1.1).

Remark 4.2. For the special case, where $s = 1$, the MSSFPMOS (1.18) becomes the SFP-MOS (1.15). In this case, by Remark 4.1, the MSSFPMOS (1.18) also becomes the GSFP (4.1). Thus, using Theorem 3.6 and Remark 4.1, we obtain Theorem 4.3 for solving the GSFP (4.1).

Theorem 4.3. *Let $H = H_1, C = C_1, Q_j = C_{j+1}, 1 \leq j \leq r-1$, $T_1 = B_1$, $T_2 = B_2B_1, \dots,$ and $T_{r-1} = B_{r-1}B_{r-2}B_{r-3}\dots B_2B_1$. Assume that the GSFP (4.1) is consistent (i.e., $S \neq \emptyset$). Fix $u \in H$ and let $x_0, x_1 \in H$ be arbitrary initial guesses and set $k := 1$. Let $\{x_k\}$ be a sequence generated by the iterative scheme. Given the iterates $x_{k-1}, x_k \in H$, then compute y_k via the manner*

$$y_k = x_k + \beta_k(x_k - x_{k-1})$$

where β_k is chosen in $[0, \overline{\beta_k})$ and

$$\overline{\beta_k} := \begin{cases} \min\left\{\beta, \dfrac{\varepsilon_k}{\max\left\{\|x_k - x_{k-1}\|^2, \|x_k - x_{k-1}\|\right\}}\right\}, & \text{if } x_k \neq x_{k-1}, \\ \beta, & \text{if } x_k = x_{k-1}. \end{cases}$$

Compute the next iterate x_{k+1} via the manner

$$x_{k+1} = P_{\widetilde{C}_{1k}}\left(\upsilon_k u + (1-\upsilon_k)\left(y_k - \tau_k \sum_{j=1}^{r-1} \delta_j T_j^* \left(I - P_{\widetilde{C}_{j+1,k}}\right) T_j y_k\right)\right)$$

where $\widetilde{C}_{1,k}$ and $\widetilde{C}_{j+1,k}$ are balls containing C_1 and C_{j+1}, respectively, the stepsize τ_k is self-adaptively updated via

$$\tau_k := \frac{\rho_k \sum_{j=1}^{r-1} \delta_j \left\| \left(I - P_{\widetilde{C}_{j+1,k}} \right) T_j y_k \right\|^2}{\lambda_k^2}$$

where

$$\lambda_k := max\left\{1, \ \left\| \sum_{j=1}^{r-1} \delta_j T_j^* \left(I - P_{\widetilde{C}_{j+1,k}} \right) T_j y_k \right\| \right\},$$

and the parameters δ_j $(j \in J_2)$ are positive constants such that $\sum_{j=1}^{r} \delta_j = 1$. Suppose the sequences $\{\upsilon_k\}$, $\{\rho_k\}$, and $\{\varepsilon_k\}$ satisfying the conditions in Assumption 3.2. Then, the sequence $\{x_k\}$ converges strongly to the point $x^* \in S$ of the SFP (1.1), where $x^* = P_S u$.

5. Numerical Experiments

In this section, we provide two numerical examples that demonstrate the effectiveness of our proposed scheme. All experiments were conducted on a standard FUJITSUNOTEBOOK laptop equipped with an 11th Gen Intel(R) Core(TM) i7-1165G7 processor running at 2.80GHz with 16GB of memory. The code was implemented using MATLAB R2022a. In Examples 5.1 and 5.2, the terms Iter. (k) and CPU(s) denote the number of iterations and the CPU time in seconds, respectively.

Example 5.1. *Let $H = \mathbb{R}^S$, $H_1 = \mathbb{R}^R$, $H_2 = \mathbb{R}^N$, $H_3 = \mathbb{R}^M$, $H_4 = \mathbb{R}^L$. Consider the sets C_i and Q_j, an ellipsoids in n-dimensional space, defined by*

$$\begin{aligned}
C_1 &= \{x \in \mathbb{R}^S : (x - z_1)^T D_1 (x - z_1) \leq \mathbf{r}_1\}, \\
C_2 &= \{x \in \mathbb{R}^S : (x - z_2)^T D_2 (x - z_2) \leq \mathbf{r}_2\}, \\
C_3 &= \{x \in \mathbb{R}^S : (x - z_3)^T D_3 (x - z_3) \leq \mathbf{r}_3\}, \\
C_4 &= \{x \in \mathbb{R}^S : (x - z_4)^T D_4 (x - z_4) \leq \mathbf{r}_4\}, \\
Q_1 &= \{T_1 x \in \mathbb{R}^R : (T_1 x - w_1)^T P_1 (T_1 x - w_1) \leq \rho_1\}, \\
Q_2 &= \{T_2 x \in \mathbb{R}^N : (T_2 x - w_2)^T P_2 (T_2 x - w_2) \leq \rho_2\}, \\
Q_3 &= \{T_3 x \in \mathbb{R}^M : (T_3 x - w_3)^T P_3 (T_3 x - w_3) \leq \rho_3\}, \\
Q_4 &= \{T_4 x \in \mathbb{R}^L : (T_3 x - w_4)^T P_4 (T_4 x - w_4) \leq \rho_4\},
\end{aligned}$$

where each $D_i \in \mathbb{R}^{S \times S}$, $P_1 \in \mathbb{R}^{R \times R}$, $P_2 \in \mathbb{R}^{N \times N}$, $P_3 \in \mathbb{R}^{M \times M}$, $P_4 \in \mathbb{R}^{L \times L}$ are positive definite matrices, $z_i \in \mathbb{R}^S$, $w_1 \in \mathbb{R}^R$, $w_2 \in \mathbb{R}^N$, $w_3 \in \mathbb{R}^M$, $w_4 \in \mathbb{R}^L$, each $\mathbf{r}_i, \rho_j > 0$, and $T_1 : \mathbb{R}^S \to \mathbb{R}^R$, $T_2 : \mathbb{R}^S \to \mathbb{R}^N$, $T_3 : \mathbb{R}^S \to \mathbb{R}^M$, $T_4 : \mathbb{R}^S \to \mathbb{R}^L$ are bounded linear operators. The goal is to find a point $x^* \in \mathbb{R}^S$ such that

$$x^* \in \Omega := \left(\cap_{i=1}^4 C_i \right) \cap \left(\cap_{j=1}^4 T_j^{-1}\left(Q_j\right) \right) \neq \emptyset. \tag{5.1}$$

It should be noted that an ellipsoid is a closed and convex set that can be mathematically expressed as a sublevel set of a specific convex function, as demonstrated in [7]. Indeed, define $c_i : \mathbb{R}^S \to \mathbb{R}$ and $q_j : \mathbb{R}^{R/N/M/L} \to \mathbb{R}$ by $c_i(x) = \frac{1}{2}\left[(x - z)^T D_i (x - z) - \mathbf{r}_i\right]$ and $q_j(T_j x) = \frac{1}{2}\left[(T_j x - w_j)^T P_j (T_j x - w_j) - \rho_j\right]$. Then, $C_i = \{x \in \mathbb{R}^S : c_i(x) \leq 0\}$ and $Q_j = \{T_j x \in \mathbb{R}^{R/N/M/L} : q_j(T_j x) \leq 0\}$ are level sets of c_i and q_j, respectively. In what follows the subgradients ξ_{ik} and η_{jk} of respectively $c_i(x)$ and $q_j(T_jx)$ can be calculated respectively at the points x and T_jx by $\xi_{ik}(x) = D_i(x-z)$ and $\eta_{jk}(T_jx) = P_j(T_jx - w_j)$. Moreover, since $\nabla c_i(x) = D_i(x-z)$, it can be easily seen that

$$\|\nabla c_i(x) - \nabla c_i(y)\| = \|D_i(x-z) - D_i(y-z)\| = \|D_i(x-y)\| \leq \|D_i\|\|x-y\|, \forall x, y \in \mathbb{R}^S$$

which further implies that ∇c_i is a $\|D_i\|$-Lipschitz continuous mapping.

Thus, according to (3.2) and (3.4), the balls $\widetilde{C}_{ik}$ $(i = 1, 2, 3, 4)$ and $\widetilde{Q}_{jk}$ $(j = 1, 2, 3, 4)$, respectively of the sets C_i and Q_j can be easily determined at a point y_k and T_jy_k, respectively and the metric projections onto the balls $\widetilde{C}_{ik}$ $(i = 1, 2, 3, 4)$ and $\widetilde{Q}_{jk}$ $(j = 1, 2, 3, 4)$, can be easily calculated.

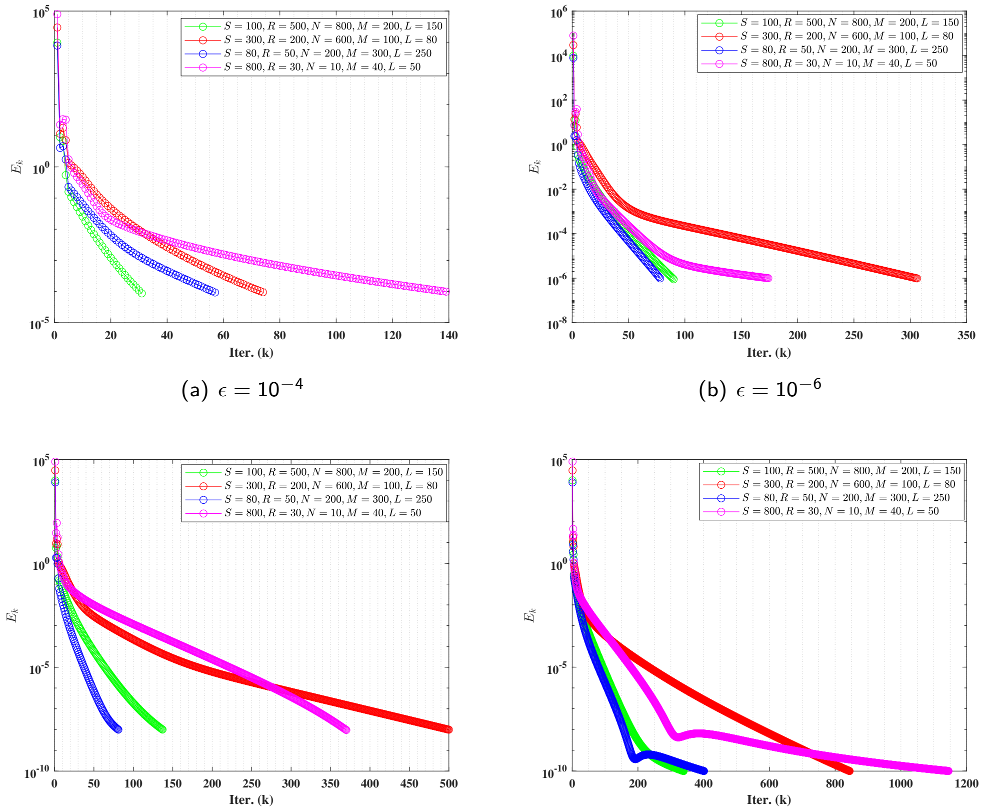
For simplicity, we denote $e_0 = (1, , 1, \dots, 1)^T \in \mathbb{R}^S$, $e_1 = (1, , 1, \dots, 1)^T \in \mathbb{R}^R$, $e_2 = (1, , 1, \dots, 1)^T \in \mathbb{R}^N$, $e_3 = (1, , 1, \dots, 1)^T \in \mathbb{R}^M$, $e_4 = (1, , 1, \dots, 1)^T \in \mathbb{R}^L$. We fix $u = e_0$ and we choose the starting points $x_0 = 100e_0$ and $x_1 = -10e_0$.

The coordinates of the points z_i $(i = 1, 2, 3, 4)$ and w_j $(j = 1, 2, 3, 4)$ are randomly generated in the closed interval $[-1, 1]$. The radii $\mathbf{r}_i$ $(i = 1, 2, 3, 4)$ are randomly generated in the closed interval $[S, 2S]$. Whereas the radii ρ_1, ρ_2, ρ_3, ρ_4 are randomly generated in the closed intervals $[R, 2R]$, $[N, 2N]$, $[M, 2M]$, and $[L, 2L]$, respectively. The positive definite matrices are chosen as $D_i = diag(ie_0)$ $(i = 1, 2, 3, 4)$, and $P_j = diag(e_j)$ $(j = 1, 2, 3, 4)$. Furthermore, the elements of the representing matrices T_j are randomly generated in the closed interval $[-5, 5]$. We also fix the parameter sequences as $\beta = 0.3$, $\epsilon_k = \frac{1}{(k+1)^3}$, $\rho_k = \frac{k}{4k+1}$, $\upsilon_k = \frac{1}{100k+5}$, $\alpha_i^k = \frac{i}{10}$, for $i = 1, 2, 3, 4$, $\varrho_i = 0 = \sigma_j$, $\delta_j = \frac{j}{10}$, for $j = 1, 2, \dots 4$.

We use $E_k = \|x_{k+1} - x_k\|^2 < \epsilon$ for sufficiently small $\epsilon > 0$ as a stopping criteria. In Table 1 and Figure 1, we report the results of Algorithm 1 with different choices of the dimensions S, R, N, M, L and different values of ϵ.

Table 1. Experimental results of Algorithm 1 for different choices of S, R, N, M, L and different values of ϵ.

		$\epsilon = 10^{-4}$	$\epsilon = 10^{-6}$	$\epsilon = 10^{-8}$	$\epsilon = 10^{-10}$
$S = 40, R = 60, N = 90, M = 100, L = 120$	Iter. (k)	34	67	71	230
	CPU(s)	0.023679	0.029949	0.032250	0.046630
$S = 100, R = 500, N = 800, M = 200, L = 150$	Iter. (k)	31	90	137	339
	CPU(s)	0.111460	0.260385	0.376923	0.896085
$S = 300, R = 200, N = 600, M = 100, L = 80$	Iter. (k)	74	306	500	844
	CPU(s)	0.215468	0.774572	1.223163	2.120658
$S = 80, R = 50, N = 200, M = 300, L = 250$	Iter. (k)	57	78	81	401
	CPU(s)	0.046596	0.058950	0.074887	0.201119
$S = 800, R = 30, N = 10, M = 40, L = 50$	Iter. (k)	139	174	370	1145
	CPU(s)	0.839825	0.989611	2.094726	6.272305



(a) $\epsilon = 10^{-4}$ (b) $\epsilon = 10^{-6}$

(c) $\epsilon = 10^{-8}$ (d) $\epsilon = 10^{-10}$

Fig. 1. Iter. (k) vs Error, experimental results of Algorithm 1 for different choices of S, R, N, M, L and different values of ϵ.

Example 5.2. *Let $H = \mathbb{R}^3$, $H_1 = \mathbb{R}^6$, $H_2 = \mathbb{R}^9$, $H_3 = \mathbb{R}^{12}$, $H_4 = \mathbb{R}^{15}$. Find a point $x^* \in \mathbb{R}^3$ such that*

$$x^* \in \Omega := C_1 \cap \Big(\cap_{j=1}^4 T_j^{-1}(Q_j) \Big) \neq \emptyset, \tag{5.2}$$

where

$$\begin{aligned} C_1 &= \{x \in \mathbb{R}^3 : \|x - o_1\|^2 \leq r_1^2\}, \\ Q_1 &= \{T_1x \in \mathbb{R}^6 : \|T_1x - w_1\|^2 \leq \rho_1^2\}, \\ Q_2 &= \{T_2x \in \mathbb{R}^9 : \|T_2x - w_2\|^2 \leq \rho_2^2\}, \\ Q_3 &= \{T_3x \in \mathbb{R}^{12} : \|T_3x - w_3\|^2 \leq \rho_3^2\}, \\ Q_4 &= \{T_4x \in \mathbb{R}^{15} : \|T_4x - w_4\|^2 \leq \rho_4^2\}, \end{aligned}$$

where o_1, $w_1 \in \mathbb{R}^6$, $w_2 \in \mathbb{R}^9$, $w_3 \in \mathbb{R}^{12}$, $w_4 \in \mathbb{R}^{15}$, $r_1, \rho_1, \rho_2, \rho_3, \rho_4 \in \mathbb{R}$, and $T_1 : \mathbb{R}^3 \to \mathbb{R}^6$, $T_2 : \mathbb{R}^3 \to \mathbb{R}^9$, $T_3 : \mathbb{R}^3 \to \mathbb{R}^{12}$, $T_4 : \mathbb{R}^3 \to \mathbb{R}^{15}$. For any $x \in \mathbb{R}^3$, we have $c_1(x) = \|x - o_1\|^2 - r_1^2$ and $q_j(T_jx) = \|T_jx - w_j\|^2 - \rho_j^2$ for $j = 1, 2, 3, 4$. In what follows the subgradients ξ_{1k} and η_{jk} of respectively $c_1(y_k)$ and $q_j(T_jy_k)$ can be calculated respectively at the points y_k and $T_j y_k$ by $\xi_{1k}(y_k) = 2(y_k - o_1)$ and $\eta_{jk}(T_j y_k) = 2(T_j y_k - w_j)$. Thus, according to (3.2) and (3.4), the balls $\widetilde{C}_{1k}$ and $\widetilde{Q}_{jk}$ $(j = 1, 2, 3, 4)$, respectively of the sets C_1 and Q_j can be easily determined at a point y_k and $T_j y_k$, respectively and the metric projections onto the balls $\widetilde{C}_{1k}$ and $\widetilde{Q}_{jk}$ $(j = 1, 2, 3, 4)$, can be easily calculated. Now, we take the radii $r_1 = 4$, $\rho_1 = 8$, $\rho_2 = 15$, $\rho_3 = 22$, $\rho_4 = 18$, the matrices

$$T_1 = \begin{pmatrix} -3.0 & 0.9 & -1.0 \\ -2.0 & -3.0 & -4.0 \\ -1.0 & 3.0 & -2.0 \\ -2.0 & -3.0 & -1.0 \\ 4.0 & 0.0 & -1.0 \\ -4.0 & 4.0 & -2.0 \end{pmatrix}, \quad T_2 = \begin{pmatrix} 4.0 & 4.0 & 3.0 \\ 1.0 & 2.0 & -2.0 \\ 4.0 & 3.0 & -0.0 \\ -0.6 & 0.7 & 3.0 \\ 4.0 & 2.0 & 1.0 \\ -4.0 & -1.0 & 4.0 \\ 1.1 & -2.0 & -2.0 \\ 3.0 & -1.8 & 3.2 \\ -2.6 & 0.8 & 1.7 \end{pmatrix},$$

$$T_3 = \begin{pmatrix} -2.5 & 2.4 & 0.0 \\ -0.2 & 2.5 & 0.2 \\ -1.0 & -1.1 & -4.1 \\ 0.9 & -0.7 & 4.0 \\ 3.0 & 4.5 & 3.8 \\ -3.9 & 0.7 & -0.6 \\ 3.2 & 3.5 & 2.8 \\ 3.4 & -2.2 & -3.5 \\ -1.4 & 1.2 & 1.2 \\ -0.7 & 0.8 & -2.3 \\ 0.7 & 4.6 & -0.5 \\ 2.0 & -4.1 & 3.4 \end{pmatrix}, \quad T_4 = \begin{pmatrix} -3.0 & 1.3 & 1.5 \\ -1.9 & 4.8 & -3.9 \\ -0.1 & 0.5 & -4.6 \\ -1.6 & 4.3 & 1.1 \\ 2.9 & 2.2 & 0.6 \\ 4.8 & -0.1 & 4.6 \\ -3.4 & 1.3 & 2.4 \\ -2.6 & 3.8 & 1.6 \\ 2.0 & -3.0 & 0.2 \\ -1.2 & -1.0 & -2.4 \\ 4.7 & 4.9 & 4.6 \\ 4.7 & -0.9 & 0.4 \\ 1.4 & 1.5 & -4.7 \\ 3.6 & 4.0 & 1.9 \\ -0.9 & 4.9 & 0.2 \end{pmatrix},$$

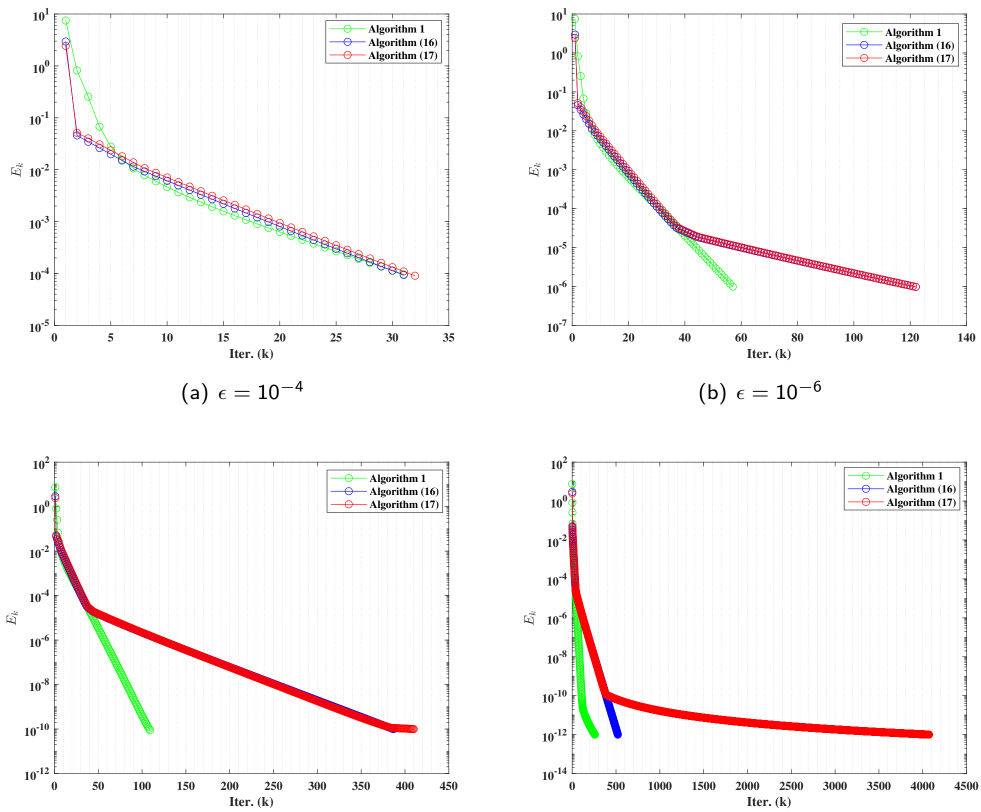
and the centers

$$\begin{aligned} &o_1 = (0.4, 0.6, 0.6)^T, w_1 = (0.1, -0.5, 0.4, -0.5, -0.1, -0.2)^T, \\ &w_2 = (0.1, 1.0, 0.5, 1.0, -0.5, 0.1, -0.9, 0.5, 0.2)^T, \\ &w_3 = (0.7, 1.0, 0.9, -0.2, -1.0, 0.1, -0.6, -0.6, -0.3, -0.9, 0.5, 0.5)^T, \\ &w_4 = (0.1, -0.3, 0.7, 0.1, 0.9, 0.8, -0.3, 0.1, -0.3, 0.26, 0.6, 0.5, -0.7, 0.6, -0.9)^T. \end{aligned}$$

In Example 5.2, we examine the convergence of the sequence $\{x_k\}$ generated by Algorithm 1 compared to the iterative methods given by Algorithm (1.16) and Algorithm (1.17). For this purpose, we consider the values of the parameters appeared in the methods as follows. We take $\beta = 0.3$, $\epsilon_k = \frac{1}{(k+1)^3}$, $\rho_k = \frac{k}{2k+1}$, $\upsilon_k = \frac{1}{10k}$, $\alpha_1^k = 1$, $\varrho_i = 0.95$, $\sigma_j = 0.5$, $\delta_j = \frac{j}{10}$, for $j = 1, 2, \ldots 4$, $u = (0.5, 0.5, 0.1)^T$, $x_0 = (-1, 3, -2)^T$, $x_1 = (4, -2, -3)^T$. Moreover, in Algorithms (1.16) and (1.17), we take $\tau_k = 0.0005$ and $f(x) = 0.975x$ in Algorithm (1.17). We use $E_k = \|x_{k+1} - x_k\|^2 < \epsilon$ for small enough $\epsilon > 0$ as a stopping criteria. In Table 2 and Figure 2, we report the experimental results of the compared methods for different values of ϵ.

Table 2. Numerical results of compared methods for different values of ϵ.

		Algorithm 1	Algorithm (1.16)	Algorithm (1.17)
$\epsilon = 10^{-4}$	Iter. (k)	31	31	32
	CPU(s)	0.022829	72.381332	142.056268
$\epsilon = 10^{-6}$	Iter. (k)	57	122	122
	CPU(s)	0.023292	44.119178	103.183311
$\epsilon = 10^{-8}$	Iter. (k)	83	253	252
	CPU(s)	0.024672	41.280574	80.067036
$\epsilon = 10^{-10}$	Iter. (k)	109	387	410
	CPU(s)	0.022095	38.690317	85.302461
$\epsilon = 10^{-12}$	Iter. (k)	259	521	4072
	CPU(s)	0.024679	43.836248	104.222961



(a) $\epsilon = 10^{-4}$ (b) $\epsilon = 10^{-6}$

(c) $\epsilon = 10^{-10}$ (d) $\epsilon = 10^{-12}$

Fig. 2. Iter. (k) vs Error, experimental results of compared methods for different values of ϵ.

It is readily apparent from Table 2 and Figure 2 that Algorithm 1 exhibits superior performance compared to the other algorithms, as evidenced by its lower number of iterations and shorter runtime in seconds.

Competing Interests

The authors declare that they have no competing interests.

Author's Contributions

All authors contributed equally to the manuscript and read and approved the final manuscript.

Acknowledgments

This research was supported by King Mongkut's University of Technology Thonburi's Post-doctoral Fellowship, Xiamen University Malaysia under grant number XMUMRF/2024-C13/IMAT/0028, Chiang Mai University, and Thailand Science Research and Innovation (TSRI) Basic Research Fund: Fiscal year 2023 under project number FRB660073/0164.

References

[1] F. Alvarez and H. Attouch, An inertial proximal method for maximal monotone operators via discretization of a nonlinear oscillator with damping, Set-Valued Anal., 9 (2001), 3–11.

[2] H.H. Bauschke and P.L. Combettes, Convex analysis and monotone operator theory in Hilbert spaces, Springer, 2011.

[3] M. Brooke, Y. Censor and A. Gibali, Dynamic string-averaging CQ-methods for the split feasibility problem with percentage violation constraints arising in radiation therapy treatment planning, Int. Trans. Oper. Res., 30 (2023), 181–205.

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

[5] C. Byrne, A unified treatment of some iterative algorithms in signal processing and image reconstruction, Inverse Problems, 20 (2003), 103–120.

[6] C. Byrne, Y. Censor, A. Gibali and S. Reich, The split common null point problem, J. Nonlinear Convex Anal., 13 (2012), 759–775.

[7] A. Cegielski, Iterative methods for fixed point problems in Hilbert spaces, Springer, 2012.

[8] A. Cegielski, General method for solving the split common fixed point problem, J. Optim. Theory Appl., 165 (2015), 385–404.

[9] Y. Censor and T. Elfving, A multiprojection algorithm using Bregman projections in a product space, Numer. Algorithms, 8 (1994), 221–39.

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

[11] Y. Censor, T. Bortfeld, B. Martin and A. Trofimov, A unified approach for inversion problems in intensity-modulated radiation therapy, Phys. Med. Biol., 51 (2006), 2353–2365.

[12] Y. Censor and A. Segal, The split common fixed point problem for directed operators, J. Convex Anal., 16 (2009), 587–600.

[13] T.N. Vinh, P. Cholamjiak and S. Suantai, A new CQ algorithm for solving split feasibility problems in Hilbert spaces, Bull. Malays. Math. Sci. Soc., 42 (2019), 2517–2534.

[14] Y. Dang, J. Sun and H. Xu, Inertial accelerated algorithms for solving a split feasibility problem, J. Ind. Manag. Optim., 13 (2017), 1383–1394.

[15] Q. Dong, H. Yuan, Y. Cho and T.M. Rassias, Modified inertial Mann algorithm and inertial CQ-algorithm for nonexpansive mappings, Optim. Lett., 12 (2018), 87–102.

[16] A. Gibali, L.W. Liu and Y.C. Tang, Note on the modified relaxation CQ algorithm for the split feasibility problem, Optim. Lett., 12 (2018), 817–830.

[17] A. Gibali, D.T. Mai, et al., A new relaxed CQ algorithm for solving split feasibility problems in Hilbert spaces and its applications, J. Ind. Manag. Optim., 15 (2019), 963–984.

[18] S. He and C. Yang, Solving the variational inequality problem defined on intersection of finite level sets, Abstr. Appl. Anal., 2013 (2013), Article ID: 942315, 1–8.

[19] S. He and Z. Zhao, Strong convergence of a relaxed CQ algorithm for the split feasibility problem, J. Inequal. Appl., 2013 (2013),Article number: 197, 1–11.

[20] J.K. Kim, T.M. Tuyen and M.T.N. Ha, Two Projection Methods for Solving the Split Common Fixed Point Problem with Multiple Output Sets in Hilbert Spaces, Numer. Funct. Anal. Optim., 42 (2021), 973–988.

[21] G. López, V. Martín-Márquez, F. Wang and H.K. Xu, Solving the split feasibility problem without prior knowledge of matrix norms, Inverse Problems, 28 (2012), 1–18.

[22] D.A. Lorenz and T. Pock, An inertial forward-backward algorithm for monotone inclusions, J. Math. Imaging Vis., 51 (2015), 311–325.

[23] P.E. Maingé, Approximation methods for common fixed points of nonexpansive mappings in Hilbert spaces, J. Math. Anal. Appl., 325 (2007), 469–479.

[24] P.E. Maingé, Inertial iterative process for fixed points of certain quasi-nonexpansive mappings, Set-Valued Anal., 15 (2007), 67–79.

[25] A. Moudafi and M. Oliny, Convergence of a splitting inertial proximal method for monotone operators, J. Comput. Appl. Math., 155 (2003), 447–454.

[26] A. Moudafi, The split common fixed-point problem for demicontractive mappings, Inverse Problems, 26 (2010), 1–5.

[27] A. Moudafi and A. Gibali, $l_1 - l_2$ regularization of split feasibility problems, Numer. Algorithms, 78 (2018), 739–757.

[28] Y.E. Nesterov, A method for solving the convex programming problem with convergence rate $O(1/k^2)$, Dokl. Akad. Nauk SSSR, 269 (1983), 543–547.

[29] B.T. Polyak, Some methods of speeding up the convergence of iteration methods, USSR Comput. Math. Math. Phys., 4 (1964), 1–17.

[30] S. Reich and T.M. Tuyen, Iterative methods for solving the generalized split common null point problem in Hilbert spaces, Optimization., 69 (2020), 1013–1038.

[31] S. Reich, M.T. Truong and T.N.H. Mai, The split feasibility problem with multiple output sets in Hilbert spaces, Optim. Lett., 14 (2020), 2335–2353.

[32] S. Reich, T.M. Tuyen and M.T.N. Ha, An optimization approach to solving the split feasibility problem in Hilbert spaces, J. Global Optim., 79 (2021), 837–852.

[33] D. Sahu, Y.J. Cho, Q.L. Dong, M.R. Kashyap and X.H. Li, Inertial relaxed CQ algorithms for solving a split feasibility problem in Hilbert spaces, Numer. Algorithms, 87 (2021), 1075–1095.

[34] Y. Shehu, P.T. Vuong and P. Cholamjiak, A self-adaptive projection method with an inertial technique for split feasibility problems in Banach spaces with applications to image restoration problems, J. Fixed Point Theory Appl., 21 (2019), 1–24,

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

[36] T.M. Tuyen, N.S. Ha and N.T.T. Thuy, A shrinking projection method for solving the split common null point problem in Banach spaces, Numer. Algorithms, 81 (2019), 813–832.

[37] J. Wang, Y. Hu, C. Li and J.C. Yao, Linear convergence of CQ algorithms and applications in gene regulatory network inference, Inverse Problems, 33 (2017), 1–25.

[38] H.K. Xu, Iterative methods for the split feasibility problem in infinite-dimensional Hilbert spaces, Inverse Problems, 26 (2010), 1–17.

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

[40] Y. Yao, M. Postolache and Y.C. Liou, Strong convergence of a self-adaptive method for the split feasibility problem, Fixed Point Theory Appl., 2013 (2013), Article No. 201, 1–12.

[41] H. Yu, W. Zhan and F. Wang, The ball-relaxed CQ algorithms for the split feasibility problem, Optimization, 67 (2018), 1687–1699.