Numerical Algorithms
https://doi.org/10.1007/s11075-023-01589-8

ORIGINAL PAPER

Alternated inertial algorithms for split feasibility problems

Bing Tan[1,3] **· Xiaolong Qin**[2] **· Xianfu Wang**[3]

Received: 29 July 2022 / Accepted: 23 May 2023

Abstract
We introduce four novel relaxed CQ algorithms with alternating inertial for solving split feasibility problems in real Hilbert spaces. The proposed algorithms employ a new non-monotonic adaptive step size criterion and utilize two different step sizes in each iteration. The weak convergence of the iterative sequences generated by the proposed algorithms is established under some weak conditions. Moreover, the Fejér monotonicity of the even subsequence with respect to the solution set is recovered. Two applications in signal denoising and image deblurring are given to illustrate the computational efficiency of our algorithms.

Keywords Split feasibility problem · CQ method · Projection and contraction method · Alternated inertial method · Signal processing · Image restoration

Mathematics Subject Classification (2010) 47J20 · 47J25 · 47J30 · 65K15

1 Introduction

The purpose of this paper is to present several efficient numerical algorithms to solve the split feasibility problem (SFP) in the framework of real Hilbert spaces. Let $\mathcal{H}_1$ and $\mathcal{H}_2$ be real Hilbert spaces with inner product $\langle \cdot, \cdot \rangle$ and norm $\| \cdot \|$. Let C and Q be nonempty, closed, and convex subsets in $\mathcal{H}_1$ and $\mathcal{H}_2$, respectively. Recall that the

✉ Xianfu Wang
shawn.wang@ubc.ca

Bing Tan
bingtan72@gmail.com

Xiaolong Qin
qxlxajh@163.com

[1] School of Sciences, Southwest Petroleum University, Chengdu 610500, China

[2] Center for Converging Humanities, Kyung Hee University, Seoul, Korea

[3] Department of Mathematics, University of British Columbia, Kelowna, B.C. V1V 1V7, Canada

Published online: 21 June 2023

SFP introduced by Censor and Elfving [1]:

$$\text{find } x^* \in C \text{ such that } Ax^* \in Q, \tag{1.1}$$

where $A : \mathcal{H}_1 \to \mathcal{H}_2$ is a bounded and linear operator. Throughout the paper, we use Γ to denote the solution set of SFP (1.1), that is,

$$\Gamma := \{x^* \in C |\ Ax^* \in Q\}.$$

Notice that the SFP can be transformed into a Constrained Optimization Problem (COP), a Fixed Point Problem (FPP), and a Variational Inequality Problem (VIP); see [2] for more details. That is, finding $x^* \in C$ to solve the SFP is equivalent to locating the solution to the following problems, respectively.

$$\boxed{\begin{array}{l} \text{(1) COP: } \min\limits_{x \in C} f(x) := \dfrac{1}{2} \left\| Ax - P_Q(Ax) \right\|^2 . \\ \text{(2) FPP: } P_C(\mathrm{Id} - \lambda A^* \left(\mathrm{Id} - P_Q\right) A)x^* = x^*, \quad \lambda > 0. \\ \text{(3) VIP: } \langle A^* \left(\mathrm{Id} - P_Q\right) Ax^*, y - x^* \rangle \geq 0, \quad \forall y \in C. \end{array}}$$

Here A^* represents the adjoint of A (transpose in finite-dimensional spaces), Id denotes the identity operator, and P_C and P_Q refer to the projections onto C and Q, respectively.

The SFP provides a unified model for many inverse problems, and it can be applied to phase retrieval problems, signal processing, image reconstruction, intensity-modulated radiation therapy (IMRT), and other fields; see, e.g., [3–8]. In this paper we introduce four novel relaxed CQ algorithms with relaxation effects and alternating inertial extrapolation terms to solve split feasibility problem (1.1) and analyze their convergence in the framework of real Hilbert spaces. Our contribution in this paper is summarized as follows.

(i) The proposed four algorithms incorporate alternating inertial steps allowing them to improve the convergence of the algorithms without inertial steps. Moreover, our four algorithms utilize two different step sizes in each iteration, which performs better than the algorithms using the same step sizes.
(ii) Our iterative algorithms also add relaxation effects, which improve the range of values of inertial parameters for the methods presented in the literature [9–11]. On the other hand, the proposed algorithms employ a non-monotonic step size criterion that allows them to work adaptively and accelerate the convergence of the algorithms.
(iii) The convergence of the iterative sequences generated by the suggested algorithms is established in infinite-dimensional Hilbert spaces, which enhances the recent results of Shehu et al. [9, 10] and Dong et al. [11] on alternating inertial methods in finite-dimensional spaces.
(iv) The performance and advantages of the algorithms proposed in this paper are confirmed by two applications in signal processing and image restoration.

This paper is organized as follows. In Section 2, we first review some of the known algorithms in the literature for solving SFPs and then collect some auxiliary results which will be used in the sequel. The convergence analysis of the proposed four algorithms is discussed in detail in Section 3. In Section 4, we provide two applications in real problems to demonstrate the efficiency and advantages of the suggested methods over previously known schemes. In the final Section 5 we summarize the whole paper with some concluding remarks.

2 Background and preliminaries

2.1 Background

In the past two decades, a large number of numerical algorithms with convergence guarantees were proposed to solve SFPs in finite- and infinite-dimensional spaces (see, e.g., [2–16]). One of the most well-known methods for solving the SFP (1.1) is the *CQ algorithm* introduced by Byrne [4], which can be summarized in the framework of the fixed-point method and the gradient projection method (see, e.g., [2, Section 3] for more details). Let us state the CQ algorithm as follows:

$$\boxed{x_{n+1} = P_C\left(x_n - \lambda A^*\left(\mathrm{Id} - P_Q\right)Ax_n\right), \quad n \geq 1,} \tag{2.1}$$

where A^* represents for the adjoint of A (transpose in finite-dimensional spaces), step size $\lambda \in \left(0, 2/\|A\|^2\right)$, Id denotes the identity operator in Hilbert space $\mathcal{H}$, and P_C and P_Q stand for the projection operator. Byrne [4] proved that the iterates generated via Algorithm (2.1) converge to the solution of problem (1.1) in a finite-dimensional space. It should be mentioned that CQ algorithm (2.1) needs to evaluate the orthogonal projections onto P_C and P_Q in each iteration, which may affect the computational efficiency of the algorithm when the projections are not easy to compute. Indeed, the projection values $P_C(x)$ and $P_Q(x)$ have explicit formulas when C and Q are special closed convex sets (e.g., half-spaces and affine sets); however, the projections onto C and Q are not easy to obtain in general, since computing the projections is equivalent to solving a multidimensional optimization problem with constraints. To overcome this shortcoming, Yang [12] introduced a modified version of CQ algorithm (2.1), now known as the *relaxed CQ algorithm*, which produces a sequence of iterations that converges to the solution of problem (1.1) by the following procedure:

$$\boxed{x_{n+1} = P_{C_n}\left(x_n - \lambda A^*\left(\mathrm{Id} - P_{Q_n}\right)Ax_n\right), \quad n \geq 1,} \tag{2.2}$$

where step size $\lambda \in \left(0, 2/\|A\|^2\right)$ and the half-spaces C_n and Q_n are defined as follows:

$$\begin{aligned} C_n &:= \left\{x \in \mathbb{R}^n \mid c\left(x_n\right) + \langle \eta_n, x - x_n\rangle \leq 0\right\}, \quad \text{where } \eta_n \in \partial c\left(x_n\right), \\ Q_n &:= \left\{y \in \mathbb{R}^m \mid q\left(Ax_n\right) + \langle \zeta_n, y - Ax_n\rangle \leq 0\right\}, \quad \text{where } \zeta_n \in \partial q\left(Ax_n\right). \end{aligned} \tag{2.3}$$

The nonempty sublevel sets C and Q are given by

$$C := \left\{x \in \mathbb{R}^n \mid c(x) \le 0\right\}, \quad Q := \left\{x \in \mathbb{R}^m \mid q(x) \le 0\right\}, \tag{2.4}$$

where $c : \mathbb{R}^n \to \mathbb{R}$ and $q : \mathbb{R}^m \to \mathbb{R}$ are lower semicontinuous convex functions, and ∂c and ∂q denote the subdifferential mappings of c and q, respectively. Note that (2.4) is sufficiently general since it does not assume the differentiability of the functions c and q, as pointed out in [12, Section 2]. It is known that if functions c and q are convex, and bounded on bounded sets defined on finite-dimensional spaces, then their subdifferentials are nonempty and uniformly bounded on bounded subsets (see [3, Corollary 7.9]). It is clear from the definitions in (2.3) and (2.4) that $C \subset C_n$ and $Q \subset Q_n$. The convergence of Algorithm (2.2) is proved in a finite-dimensional space (see [12, Theorem 1]).

It should be mentioned that the projection on the half-spaces C_n and Q_n can be computed exactly (see, e.g., [17, Example 29.20]), and thus Algorithm (2.2) greatly improves the computational efficiency of Algorithm (2.1) especially when the orthogonal projections are not available easily. The weak convergence of CQ algorithm (2.1) and relaxed CQ algorithm (2.2) in infinite-dimensional Hilbert spaces were proved in [2]. Note that fixed-step algorithms (2.1) and (2.2) may be difficult to implement in practice since the step size of the algorithms requires the prior knowledge of the norm of operator A, which is not readily available (see [18] for more details). On the other hand, it is known that the convergence speed of the algorithm with a fixed step size may be slow even if we can easily compute the norm of operator A (see, e.g., [19, Section 4]). To overcome this drawback, many algorithms incorporating adaptive step sizes were proposed to solve the SFP (1.1) in finite- and infinite-dimensional spaces; see, e.g., [13, 15, 19–24] and the references therein. It should be emphasized that the algorithms presented in [13, 20, 21] with Armijo-type step sizes may affect their computational efficiency because multiple search processes may need to be performed in each iteration for finding the appropriate step size. To overcome this shortcoming, some numerical algorithms using adaptive step size criteria, which update the iteration step size through a simple calculation based on some previously known information, were proposed; see, e.g., [15, 22–24].

In recent years, inertial techniques (see [25, 26]) were investigated to accelerate the convergence of algorithms. They were incorporated into a large number of numerical algorithms for solving various optimization problems in finite- and infinite-dimensional spaces; see, e.g., [9–11, 22–29] and the references therein. Recall that the fundamental feature of inertial methods is that the next iteration is determined by the previous two (or more) iterations, and that this small change can greatly improve the convergence of the non-inertial versions (e.g., see [27] for details of the theory and experiments). The results of numerous computational tests and applications show that inertial methods can significantly improve the convergence of non-inertial methods. However, the monotonicity of the iterative sequence generated by inertial methods is lost, which results in inertial algorithms sometimes converging more slowly than those without inertial. To deal with this situation, Mu and Peng [30] introduced an *alternated inertial method* that recovers the Fejér monotonicity of the even subsequence associated with the solution set of the problem. Recall that the basic idea of the alternated inertial method is to add inertial effects only at odd iteration steps and not at even iteration steps, which is the origin of the word "alternated" in the method. Recently, many scholars proposed some iterative algorithms with alternating inertial techniques to solve split feasibility problems, variational inequalities, and others; see, e.g., [9–11, 31–34]. The advantages of these alternated inertial methods over some classical inertial methods are verified in theory as well as in numerical experiments. Recently, Shehu and Gibali [9] introduced a relaxed CQ algorithm with alternating inertial extrapolation steps for solving the SFP in a finite-dimensional Euclidean space. An Armijo step size criterion embedded in their proposed method allows it to work without the prior information of the norm of operator A. More precisely, their adaptive iterative scheme is stated in Algorithm 2.1.

Algorithm 2.1 Shehu and Gibali's Algorithm 1.

Initialization: Take $\gamma > 0$, $\ell \in (0,1)$, and $\mu \in (0,1)$. Choose $\{\theta_n\}$ with $0 \leq \theta_n \leq \theta < \frac{1-\mu}{1+\mu}$. Set $\nabla f_n := A^*\left(\mathrm{Id} - P_{Q_n}\right)A$. Define the half-spaces C_n and Q_n as follows:

$$\begin{aligned} C_n &:= \left\{x \in \mathbb{R}^n \mid c\left(w_n\right) + \langle \eta_n, x - w_n\rangle \leq 0\right\}, \quad \text{where } \eta_n \in \partial c\left(w_n\right), \\ Q_n &:= \left\{y \in \mathbb{R}^m \mid q\left(Aw_n\right) + \langle \zeta_n, y - Aw_n\rangle \leq 0\right\}, \quad \text{where } \zeta_n \in \partial q\left(Aw_n\right), \end{aligned} \tag{2.5}$$

where c and q are defined in (2.4). Let $x_0, x_1 \in \mathbb{R}^n$. Set $n := 1$.
Iterative Steps: Calculate the next iteration point x_{n+1} as follows.
Step 1. Compute

$$w_n = \begin{cases} x_n, & \text{if } n = \text{even}, \\ x_n + \theta_n\left(x_n - x_{n-1}\right), & \text{if } n = \text{odd}. \end{cases} \tag{2.6}$$

Step 2. Compute $y_n = P_{C_n}\left(w_n - \lambda_n \nabla f_n(w_n)\right)$, where $\lambda_n = \gamma \ell^{m_n}$ and m_n is the smallest non-negative integer m such that

$$\lambda_n \left\|\nabla f_n(w_n) - \nabla f_n(y_n)\right\| \leq \mu \left\|w_n - y_n\right\|.$$

Step 3. Compute $x_{n+1} = P_{C_n}\left(w_n - \lambda_n \nabla f_n(y_n)\right)$.
Set $n := n + 1$ and go to *Step 1.*

They prove that the iterates created by Algorithm 2.1 converge to the solution of SFP (1.1) in Euclidean spaces, see [9, Theorem 3.3]. Notice that Algorithm 2.1 employs an Armijo-like step size criterion (also known as the line search method) for determining the appropriate step size λ_n, which may computationally consume a significant amount of additional time due to the fact that the line search process may require be evaluated several times for the purpose of finding the smallest non-negative integer m_n. To speed up the convergence of Algorithm 2.1 introduced by Shehu and Gibali [9], an alternated inertial relaxed CQ algorithm with an adaptive step size scheme (introduced in [15]) for solving SFPs in finite-dimensional spaces was recently developed by Shehu et al. [10]. Their iterative scheme is illustrated in Algorithm 2.2.

The convergence of Algorithm 2.2 is verified in a finite-dimensional space. On the other hand, motivated by the relaxed CQ algorithm [12], the modified projection contraction algorithm [20], the alternated inertial method [9], and the adaptive step size method [28], Dong et al. [11] proposed an adaptive relaxed CQ algorithm without

Algorithm 2.2 Shehu et al.'s Algorithm 2.

Initialization: Let $\{\chi_n\} \subset (0, 4)$ be non-decreasing. Take the non-negative sequence $\{\theta_n\}$ such that $0 < \alpha \le \chi_n (1 + \theta_n) \le \beta < 4$. Define the half-spaces C_n and Q_n as in (2.5). Let $x_0, x_1 \in \mathbb{R}^n$. Set $n := 1$.

Iterative Steps: Calculate the next iteration point x_{n+1} as follows.

Step 1. Compute w_n by (2.6).

Step 2. Compute $x_{n+1} = w_n - \lambda_n \nabla f_n (w_n)$, where the step size λ_n is generated by

$$\lambda_n := \begin{cases} \dfrac{\chi_n f_n (w_n)}{\|\nabla f_n (w_n)\|^2}, & \|\nabla f_n (w_n)\| \neq 0, \\ 0, & \text{otherwise}, \end{cases}$$

in which

$$\begin{aligned} f_n(x) &:= \frac{1}{2} \left\| \left(\mathrm{Id} - P_{C_n}\right) x \right\|^2 + \frac{1}{2} \left\| \left(\mathrm{Id} - P_{Q_n}\right) Ax \right\|^2, \\ \nabla f_n(x) &= \left(\mathrm{Id} - P_{C_n}\right) x + A^* \left(\mathrm{Id} - P_{Q_n}\right) Ax. \end{aligned}$$

Set $n := n + 1$ and go to *Step 1*.

any line search process to discover the solution of SFP in Euclidean spaces. Indeed, their scheme is described in Algorithm 2.3.

Algorithm 2.3 Dong et al.'s Algorithm 4.1.

Initialization: Take $\lambda_1 > 0$, $\mu \in (0, 1)$, and $\tau \in (0, 2)$. Set $\nabla f_n := A^* \left(\mathrm{Id} - P_{Q_n}\right) A$. Define the half-spaces C_n and Q_n as in (2.5). Let $x_0, x_1 \in \mathbb{R}^n$. Set $n := 1$.

Iterative Steps: Calculate the next iteration point x_{n+1} as follows.

Step 1. Compute w_n by (2.6).

Step 2. Compute $y_n = P_{C_n} (w_n - \lambda_n \nabla f_n (w_n))$.

Step 3. Compute $x^I_{n+1} = w_n - \tau \varphi_n d_n$ $\left(\text{or compute } x^{II}_{n+1} = P_{C_n} (w_n - \tau \varphi_n \lambda_n \nabla f_n (y_n))\right)$, where

$$\begin{aligned} d_n &= w_n - y_n - \lambda_n (\nabla f_n (w_n) - \nabla f_n (y_n)), \\ \varphi_n &= \frac{\langle w_n - y_n, d_n \rangle + \lambda_n \left\| \left(\mathrm{Id} - P_{Q_n}\right) Ay_n \right\|^2}{\|d_n\|^2}, \end{aligned}$$

and update the next step size λ_{n+1} by

$$\lambda_{n+1} = \begin{cases} \min \left\{ \dfrac{\mu \|w_n - y_n\|}{\|\nabla f_n (w_n) - \nabla f_n (y_n)\|}, \lambda_n \right\}, & \text{if } \|\nabla f_n (w_n) - \nabla f_n (y_n)\| \neq 0, \\ \lambda_n, & \text{otherwise}. \end{cases}$$

Set $n := n + 1$ and go to *Step 1*.

In the finite-dimensional space, Dong et al. [11] proved that the iterative sequence $\{x^I_{n+1}\}$ (or $\{x^{II}_{n+1}\}$) generated by Algorithm 2.3 converges to the solution of the SFP under the conditions of $\theta_n \in (-1, (2 - \tau)/\tau)$ and $\theta_n \in (-1, 0)$, respectively. The efficiency of the alternated inertial relaxed CQ Algorithms 2.1–2.3 compared to several previously known methods is verified by some numerical examples and applications in signal processing and image restoration (see the numerical experiments in the literature [9–11] for more details).

Therefore, a natural question arises: *How to modify Algorithms 2.1 and 2.3 so that they can use different step sizes in each iteration and improve the computational efficiency of the algorithms*? Inspired and motivated by the results in [9–12], we present in this paper four modified relaxed CQ algorithms for solving the split feasibility problem in real Hilbert spaces.

2.2 Preliminaries

In this subsection, we collect some important definitions and lemmas for further use in the convergence analysis of our main results. Let $\mathcal{H}$ be a Hilbert space and $A : \mathcal{H} \to \mathcal{H}$ be a bounded linear operator. Throughout the paper, we refer to the set of real numbers and the set of positive integers by $\mathbb{R}$ and $\mathbb{N}$, respectively. We use $x_n \rightharpoonup x$ (resp., $x_n \to x$) to denote the weak convergence (resp., strong convergence) of a sequence $\{x_n\}$ to x. It is well known that the following equation

$$\|\alpha x + (1-\alpha)y\|^2 = \alpha\|x\|^2 + (1-\alpha)\|y\|^2 - \alpha(1-\alpha)\|x-y\|^2 \tag{2.7}$$

holds for all $x, y \in \mathcal{H}$ and $\alpha \in \mathbb{R}$. This is frequently used in our convergence analysis.

Definition 2.1 ([17, Definition 4.1]) Recall that a mapping $T : \mathcal{H} \to \mathcal{H}$ is called

- *L-Lipschitz continuous* with $L > 0$ if

$$\|Tx - Ty\| \le L\|x - y\|, \quad \forall x, y \in \mathcal{H}.$$

- *Nonexpansive* if

$$\|Tx - Ty\| \le \|x - y\|, \quad \forall x, y \in \mathcal{H}.$$

- *Firmly nonexpansive* if

$$\|Tx - Ty\|^2 \le \|x - y\|^2 - \|(\mathrm{Id} - T)x - (\mathrm{Id} - T)y\|^2, \quad \forall x, y \in \mathcal{H},$$

  or equivalently (see [17, Proposition 4.4(iv)]),

$$\|Tx - Ty\|^2 \le \langle x - y, Tx - Ty\rangle, \quad \forall x, y \in \mathcal{H}.$$

Definition 2.2 ([17, Definition 3.8]) The *metric projection* of a point x from $\mathcal{H}$ onto $C \subseteq \mathcal{H}$ is defined as

$$P_C(x) := \operatorname{argmin}\{\|x - y\|, y \in C\}.$$

One finds that $P_C(x)$ has a closed-form expressions when C is a special polyhedron. For example, if C is a half-space defined by $C := \{x \in \mathcal{H} \mid \langle u, x\rangle \le v\}$, we can use the following formula to calculate the exact value of the projection of a point x onto C (cf. [17, Example 29.20]).

$$P_C(x) = x - \max\left\{\frac{\langle u, x\rangle - v}{\|u\|^2}, 0\right\} u.$$

For more information about projections onto special polyhedra, the reader can refer to, e.g., [17, Chapter 29] and [35, Chapter 6]. It is known that the projection operator P_C is firmly nonexpansive (cf. [17, Proposition 4.16]) and it satisfies the following two properties which are used in the subsequent convergence analysis, where (2.8) and (2.9) arrive from [17, Proposition 4.16] and [17, Theorem 3.16], respectively.

$$\|P_C(x) - y\|^2 \leq \|x - y\|^2 - \|x - P_C(x)\|^2, \quad \forall x \in \mathcal{H}, y \in C; \tag{2.8}$$

$$\langle x - P_C(x), y - P_C(x) \rangle \leq 0, \quad \forall x \in \mathcal{H}, y \in C. \tag{2.9}$$

Remark 2.1 Let C be a nonempty, closed, and convex subset of $\mathcal{H}$. Then $\mathrm{Id} - P_C$ is firmly nonexpansive; see [17, Corollary 4.18].

Definition 2.3 ([17, Definition 1.21]) Let $f : \mathcal{H} \to (-\infty, +\infty]$ be a proper function. Then f is called

- *Lower semicontinuous* at x if $x_n \to x$ implies $f(x) \leq \liminf_{n\to\infty} f(x_n)$.
- *Weakly lower semicontinuous* at x if $x_n \rightharpoonup x$ implies $f(x) \leq \liminf_{n\to\infty} f(x_n)$.

Remark 2.2 The two definitions described above are equivalent if f is convex (see [17, Theorem 9.1]).

Definition 2.4 ([17, Definition 16.1]) The subdifferential $\partial f(x)$ of a proper function $f : \mathcal{H} \to (-\infty, +\infty]$ at the point x is defined by

$$\partial f(x) := \{s \in \mathcal{H} \mid f(y) - f(x) \geq \langle y - x, s \rangle, \quad \forall y \in \mathcal{H}\}.$$

Definition 2.5 ([17, Definition 5.1]) Let C be a nonempty subset of $\mathcal{H}$, and let $\{x_n\}$ be a sequence in $\mathcal{H}$. Then x_n is Fejér monotone with respect to C if

$$\|x_{n+1} - x\| \leq \|x_n - x\|, \quad \forall x \in C, \ n \in \mathbb{N}.$$

The following lemmas are crucial to the convergence analysis of our algorithms.

Lemma 2.1 ([2, Proposition 3.2]) *Let $\lambda > 0$ and $x^* \in \mathcal{H}$. Then x^* solves SFP (1.1) if and only if x^* solves the fixed point problem*

$$x^* = P_C\left(x^* - \lambda A^*\left(\mathrm{Id} - P_Q\right) A x^*\right).$$

Lemma 2.2 ([17, Lemma 2.47], [36]) *Let C be a nonempty set of $\mathcal{H}$, and $\{x_n\}$ be a sequence in $\mathcal{H}$. If $\lim_{n\to\infty} \|x_n - x\|$ exists for any $x \in C$, and every sequential weak cluster point of $\{x_n\}$ is in C, then $\{x_n\}$ converges weakly to a point in C.*

3 Iterative methods and their convergence analysis

In this section, we introduce four alternated inertial relaxed CQ methods with adaptive non-monotonic step sizes and relaxation effects for finding the solutions to split feasibility problem (1.1) and analyze their convergence in infinite-dimensional Hilbert spaces. Our methods are motivated by the CQ methods [4, 12], the alternated inertial methods [9–11], and the adaptive step size technique [24, 28]. The proposed algorithms can work well without the prior knowledge of the operator norm $\|A\|$ since they employ an adaptive non-monotonic step size criterion that does not involve any line search procedure.

To prove the convergence of the relaxed CQ algorithms proposed in this paper, we suppose that the following two conditions hold:

(A1) The nonempty level sets C and Q in the SFP (1.1) can be presented as follows.

$$C := \{x \in \mathcal{H}_1 \mid c(x) \leq 0\}, \quad Q := \{x \in \mathcal{H}_2 \mid q(x) \leq 0\}, \tag{3.1}$$

where $c : \mathcal{H}_1 \to \mathbb{R}$ and $q : \mathcal{H}_2 \to \mathbb{R}$ are convex and subdifferential functions on Hilbert spaces $\mathcal{H}_1$ and $\mathcal{H}_2$, respectively. Then c and q are also weakly lower semicontinuous (see [17, Theorem 9.1]).

(A2) Let ∂c and ∂q denote the subdifferentials of c and q defined in (3.1), respectively. Assume that at least one subgradient $\eta \in \partial c(x)$ and $\zeta \in \partial q(y)$ can be computed for any $x \in \mathcal{H}_1$ and $y \in \mathcal{H}_2$. Let C_n and Q_n are defined as follows.

$$\begin{aligned} C_n &:= \{x \in \mathcal{H}_1 \mid c(w_n) + \langle \eta_n, x - w_n \rangle \leq 0\}, \quad \text{where } \eta_n \in \partial c(w_n), \\ Q_n &:= \{y \in \mathcal{H}_2 \mid q(Aw_n) + \langle \zeta_n, y - Aw_n \rangle \leq 0\}, \quad \text{where } \zeta_n \in \partial q(Aw_n), \end{aligned} \tag{3.2}$$

where w_n is calculated by (3.3). Let $A : \mathcal{H}_1 \to \mathcal{H}_2$ be a bounded and linear operator. Suppose that the subdifferentials ∂c and ∂q are bounded operators (i.e., bounded on bounded sets).

It is clear from the definitions in (3.1) and (3.2) that $C \subset C_n$ and $Q \subset Q_n$. Obviously, both C_n and Q_n are half-spaces if $\eta_n \neq 0$ and $\zeta_n \neq 0$, so their projections have explicit formulas (see [21, Remark 2.1]). As done in many papers using (relaxed) CQ methods, we define

$$f_n(x) := \frac{1}{2}\left\|\left(\mathrm{Id} - P_{Q_n}\right)Ax\right\|^2 = \frac{1}{2}d^2_{Q_n}(Ax).$$

It is known that f_n is convex and differentiable in $\mathcal{H}_1$. Moreover, the gradient of f_n at x is given by $\nabla f_n(x) = A^*\left(\mathrm{Id} - P_{Q_n}\right)Ax$ and ∇f_n is L-Lipschitz continuous with $L = \|A\|^2$.

3.1 Two modified adaptive relaxed CQ algorithms

In this subsection, based on Algorithm 2.1 proposed by Shehu and Gibali [9], we introduce two relaxed CQ algorithms with adaptive step size criterion and alternating inertial steps for solving SFP (1.1). In order to analyze the convergence of Algorithm 3.1, we assume the following conditions:

(C1) The solution set of SFP (1.1) is nonempty, that is, $\Gamma \neq \emptyset$.

(C2) Let $\lambda_1 > 0$, $\mu \in (0, 1)$, and $\{\xi_n\} \subset [1, \infty)$ such that $\sum_{n=1}^{\infty}(\xi_n - 1) < \infty$, and $\{\rho_n\} \subset [0, \infty)$ such that $\sum_{n=1}^{\infty} \rho_n < \infty$.

(C3) Let $\alpha \in (0, 1]$, $\beta \in (0, 2/(1+\mu))$, and $0 \leq \theta_n \leq \theta < \frac{\beta^*+2(1-\alpha)}{2\alpha}$, where $\beta^* = 2 - \beta - \beta\mu$ when $\beta \in [1, 2/(1+\mu))$, and $\beta^* = \beta - \beta\mu$ when $\beta \in (0, 1)$.

Now, we are in a position to state our Algorithm 3.1.

Algorithm 3.1

Initialization: Take λ_1, μ, α, β, $\{\xi_n\}$, $\{\rho_n\}$, and $\{\theta_n\}$ such that Conditions (C2) and (C3) hold. Choose initial points $x_0, x_1 \in \mathcal{H}$. Set $n := 1$.

Iterative Steps: Calculate the next iteration point x_{n+1} as follows:

Step 1. Compute

$$w_n = \begin{cases} x_n, & \text{if } n = \text{even}, \\ x_n + \theta_n \left(x_n - x_{n-1}\right), & \text{if } n = \text{odd}. \end{cases} \tag{3.3}$$

Step 2. Compute $y_n = P_{C_n}(w_n - \lambda_n \nabla f_n(w_n))$. If $y_n = w_n$ then the iteration stops and y_n is the solution of SFP; otherwise, turn to *Step 3*.

Step 3. Compute $z_n = P_{C_n}(w_n - \beta\lambda_n \nabla f_n(y_n))$.

Step 4. Compute $x_{n+1} = (1-\alpha) w_n + \alpha z_n$, and update the next step size λ_{n+1} by

$$\lambda_{n+1} = \begin{cases} \min\left\{ \dfrac{\mu \|w_n - y_n\|}{\|\nabla f_n(w_n) - \nabla f_n(y_n)\|}, \xi_n\lambda_n + \rho_n \right\}, & \text{if } \|\nabla f_n(w_n) - \nabla f_n(y_n)\| \neq 0, \\ \xi_n\lambda_n + \rho_n, & \text{otherwise.} \end{cases} \tag{3.4}$$

Set $n := n + 1$ and go to *Step 1.*

Remark 3.1 We show that the iterations of Algorithm 3.1 stop when $w_n = y_n$ for some $n \geq 1$. Indeed, if $w_n = y_n$, then we conclude from the definition of y_n that $y_n = P_{C_n}(y_n - \beta\lambda_n \nabla f_n(y_n))$. This implies that $y_n \in C_n$, and $Ay_n \in Q_n$ by means of Lemma 2.1, which together with (3.1) and (3.2) yields that $y_n \in C$ and $Ay_n \in Q$. Therefore, y_n is a solution to (1.1) when $w_n = y_n$.

Remark 3.2 It should be noted that the step size criterion (3.4) is similar to the step size criterions in [11, 28, 29]; however, step size criterion (3.4) is more flexible than step size criterions in [11, 28, 29]. When $\xi_n = 1$ in step size criterion (3.4), it degenerates to the step size rule used in [29, Algorithm 3.1]. If $\xi_n = 1$ and $\rho_n = 0$ in step size criterion (3.4), then it becomes the step size criterion employed in [11, Algorithm 3.1] and [28, Algorithm 1]. In the case of $\xi_n \neq 1$, step size criterion (3.4) can choose a wider range of step sizes than step size criterions in [11, 28, 29]. To the best of our knowledge, the step size criterion (3.4) is the first time adopted in solving the split feasibility problem.

The following lemmas are essential for the convergence analysis of Algorithm 3.1. Let us first establish that the step size created by (3.4) is well defined.

Lemma 3.1 *Let step size $\{\lambda_n\}$ be a sequence generated by (3.4). Then it is well defined and $\lambda_n \geq \frac{\mu}{\|A\|^2}$ for all $n \in \mathbb{N}$.*

Proof Since the operator ∇f_n is $\|A\|^2$-Lipschitz continuous, one obtains

$$\frac{\mu\|w_n - y_n\|}{\|\nabla f_n(w_n) - \nabla f_n(y_n)\|} \geq \frac{\mu\|w_n - y_n\|}{\|A\|^2\|w_n - y_n\|} = \frac{\mu}{\|A\|^2}.$$

This combining with (3.4) yields $\lambda_{n+1} \geq \min\{\lambda_n, \frac{\mu}{\|A\|^2}\}$. By induction, one finds that $\lambda_n \geq \min\{\lambda_1, \frac{\mu}{\|A\|^2}\}$. On the other hand, it can be seen from (3.4) that $\lambda_{n+1} \leq \xi_n\lambda_n + \rho_n$ for any $n \geq 1$. In view of the condition (C2) and [37, Lemma 1], it can be concluded that $\lim_{n\to\infty}\lambda_n$ exists. Since $\{\lambda_n\}$ has a lower bound $\min\{\lambda_1, \frac{\mu}{\|A\|^2}\}$, we have $\lim_{n\to\infty}\lambda_n := \lambda > 0$. This completes the proof. □

Lemma 3.2 *Suppose that* $\{w_n\}$, $\{y_n\}$, *and* $\{z_n\}$ *are sequences created by Algorithm 3.1. Let* $p \in \Gamma$. *Then,*

$$\begin{aligned}\|z_n - p\|^2 \leq \|w_n - p\|^2 - \frac{2\beta\mu}{\|A\|^2}\left\|\left(\mathrm{Id} - P_{Q_n}\right)Ay_n\right\|^2 \\ - \beta_n\left(\|w_n - y_n\|^2 + \|z_n - y_n\|^2\right),\end{aligned}$$

where

$$\beta_n := \begin{cases} 2 - \beta - \frac{\beta\mu\lambda_n}{\lambda_{n+1}}, & \text{if } \beta \in [1, 2/(1+\mu)), \\ \beta - \frac{\beta\mu\lambda_n}{\lambda_{n+1}}, & \text{if } \beta \in (0, 1). \end{cases}$$

Proof Take a point p in the solution set Γ of SFP (1.1). Combining the definition of z_n, $p \in C_n$, and (2.8), we have

$$\begin{aligned}\|z_n - p\|^2 &= \left\|P_{C_n}(w_n - \beta\lambda_n\nabla f_n(y_n)) - p\right\|^2 \\ &\leq \|(w_n - p) - \beta\lambda_n\nabla f_n(y_n)\|^2 - \|z_n - w_n + \beta\lambda_n\nabla f_n(y_n)\|^2 \\ &= \|w_n - p\|^2 - 2\beta\lambda_n\langle\nabla f_n(y_n), w_n - p\rangle - \|z_n - w_n\|^2 \\ &\quad - 2\beta\lambda_n\langle\nabla f_n(y_n), z_n - w_n\rangle \\ &= \|w_n - p\|^2 - \|z_n - w_n\|^2 - 2\beta\lambda_n\langle\nabla f_n(y_n), z_n - y_n\rangle \\ &\quad - 2\beta\lambda_n\langle\nabla f_n(y_n), y_n - p\rangle.\end{aligned} \tag{3.5}$$

Note that $\nabla f_n(p) = A^*\left(I - P_{Q_n}\right)Ap = 0$ due to $Ap \in Q_n$. From the fact that $\mathrm{Id} - P_{Q_n}$ is firmly nonexpansive and $\nabla f_n(p) = 0$, we obtain

$$\begin{aligned}\langle\nabla f_n(y_n), y_n - p\rangle &= \langle\nabla f_n(y_n) - \nabla f_n(p), y_n - p\rangle \\ &= \left\langle A^*\left(\mathrm{Id} - P_{Q_n}\right)Ay_n - A^*\left(\mathrm{Id} - P_{Q_n}\right)Ap, y_n - p\right\rangle \\ &= \left\langle\left(\mathrm{Id} - P_{Q_n}\right)Ay_n - \left(\mathrm{Id} - P_{Q_n}\right)Ap, Ay_n - Ap\right\rangle \\ &\geq \left\|\left(\mathrm{Id} - P_{Q_n}\right)Ay_n\right\|^2.\end{aligned} \tag{3.6}$$

It follows from Lemma 3.1 that $\lambda_n \geq \frac{\mu}{\|A\|^2}$, which together with (3.6) implies

$$2\beta\lambda_n \left\langle \nabla f_n\left(y_n\right), y_n - p\right\rangle \geq \frac{2\beta\mu}{\|A\|^2} \left\|\left(\mathrm{Id} - P_{Q_n}\right) A y_n\right\|^2. \tag{3.7}$$

Note that

$$-\left\|w_n - z_n\right\|^2 = -\left\|w_n - y_n\right\|^2 - \left\|y_n - z_n\right\|^2 + 2\left\langle w_n - y_n, z_n - y_n\right\rangle, \tag{3.8}$$

and

$$\begin{aligned}
&2\left\langle w_n - y_n, z_n - y_n\right\rangle \\
&= 2\left\langle w_n - \lambda_n \nabla f_n\left(w_n\right) - y_n, z_n - y_n\right\rangle + 2\left\langle \lambda_n \nabla f_n\left(y_n\right), z_n - y_n\right\rangle \\
&\quad + 2\lambda_n \left\langle \nabla f_n\left(w_n\right) - \nabla f_n\left(y_n\right), z_n - y_n\right\rangle.
\end{aligned} \tag{3.9}$$

By the definition of $y_n, z_n \in C_n$, and (2.9), we have

$$\left\langle w_n - \lambda_n \nabla f_n\left(w_n\right) - y_n, z_n - y_n\right\rangle \leq 0. \tag{3.10}$$

Using the definition of λ_{n+1}, one sees that

$$\begin{aligned}
&2\lambda_n \left\langle \nabla f_n\left(w_n\right) - \nabla f_n\left(y_n\right), z_n - y_n\right\rangle \\
&\leq 2\lambda_n \left\|\nabla f_n\left(w_n\right) - \nabla f_n\left(y_n\right)\right\| \left\|z_n - y_n\right\| \\
&\leq 2\mu \frac{\lambda_n}{\lambda_{n+1}} \left\|w_n - y_n\right\| \left\|z_n - y_n\right\| \\
&\leq \mu \frac{\lambda_n}{\lambda_{n+1}} \left(\left\|w_n - y_n\right\|^2 + \left\|z_n - y_n\right\|^2\right).
\end{aligned} \tag{3.11}$$

Substituting (3.9), (3.10), and (3.11) into (3.8), we deduce that

$$\begin{aligned}
-\left\|w_n - z_n\right\|^2 \leq &-\left(1 - \frac{\mu\lambda_n}{\lambda_{n+1}}\right)\left(\left\|w_n - y_n\right\|^2 + \left\|z_n - y_n\right\|^2\right) \\
&+ 2\left\langle \lambda_n \nabla f_n\left(y_n\right), z_n - y_n\right\rangle,
\end{aligned}$$

which yields that

$$\begin{aligned}
-2\beta\left\langle \lambda_n \nabla f_n\left(y_n\right), z_n - y_n\right\rangle \leq &-\beta\left(1 - \frac{\mu\lambda_n}{\lambda_{n+1}}\right)\left(\left\|w_n - y_n\right\|^2 + \left\|z_n - y_n\right\|^2\right) \\
&+ \beta\left\|w_n - z_n\right\|^2, \quad \forall \beta > 0.
\end{aligned} \tag{3.12}$$

Combining (3.5), (3.7), and (3.12), we have

$$\|z_n - p\|^2 \le \|w_n - p\|^2 - \frac{2\beta\mu}{\|A\|^2}\left\|\left(\mathrm{Id} - P_{Q_n}\right)Ay_n\right\|^2 - (1-\beta)\|w_n - z_n\|^2 \\ - \beta\left(1 - \frac{\mu\lambda_n}{\lambda_{n+1}}\right)\left(\|w_n - y_n\|^2 + \|z_n - y_n\|^2\right), \quad \forall \beta > 0. \tag{3.13}$$

By applying the inequality $\|a+b\|^2 \le 2(\|a\|^2 + \|b\|^2)$, one finds that

$$\|w_n - z_n\|^2 \le 2\left(\|w_n - y_n\|^2 + \|z_n - y_n\|^2\right). \tag{3.14}$$

It follows from (3.13) and (3.14) that

$$\|z_n - p\|^2 \le \|w_n - p\|^2 - \frac{2\beta\mu}{\|A\|^2}\left\|\left(\mathrm{Id} - P_{Q_n}\right)Ay_n\right\|^2 \\ - \beta_n\left(\|w_n - y_n\|^2 + \|z_n - y_n\|^2\right),$$

where β_n is defined by

$$\beta_n := \begin{cases} 2 - \beta - \frac{\beta\mu\lambda_n}{\lambda_{n+1}}, & \text{if } \beta \in [1, 2/(1+\mu)), \\ \beta - \frac{\beta\mu\lambda_n}{\lambda_{n+1}}, & \text{if } \beta \in (0,1). \end{cases}$$

This is the desired conclusion. □

Remark 3.3 From $\mu \in (0,1)$, $\beta \in (0, 2/(1+\mu))$, and Lemma 3.1, we have

$$\beta^* := \lim_{n\to\infty} \beta_n = \begin{cases} 2 - \beta - \beta\mu, & \beta \in [1, 2/(1+\mu)), \\ \beta - \beta\mu, & \beta \in (0,1). \end{cases}$$

Thus we obtain that $\lim_{n\to\infty} \beta_n > 0$ for all $\beta \in (0, 2/(1+\mu))$. That is, there exists a positive constant N_0 such that $\beta_n > 0$ holds for all $n \ge N_0$.

Next we show that the even subsequence generated by alternated inertial relaxed CQ algorithm 3.1 has the Fejér monotonicity with respect to the solution Γ of SFP (1.1).

Lemma 3.3 *Let the sequence $\{x_n\}$ be generated by Algorithm 3.1. Then the even subsequence $\{x_{2n}\}$ is bounded and it is Fejér monotone with respect to the solution set Γ. Moreover, for all $p \in \Gamma$, $\lim_{n\to\infty}\|x_{2n} - p\|$ exists, and $\lim_{n\to\infty}\|x_{2n} - y_{2n}\| = 0$.*

Proof It follows from the definition of x_{n+1} and (2.7) that

$$\|x_{n+1} - p\|^2 = (1-\alpha)\|w_n - p\|^2 + \alpha\|z_n - p\|^2 - \alpha(1-\alpha)\|w_n - z_n\|^2. \tag{3.15}$$

Combining (3.14), (3.15), and Lemma 3.2, we obtain

$$\begin{aligned}\|x_{n+1}-p\|^2 &\leq (1-\alpha)\|w_n-p\|^2+\alpha\|w_n-p\|^2-\alpha(1-\alpha)\|w_n-z_n\|^2\\&\quad-\frac{2\alpha\beta\mu}{\|A\|^2}\left\|\left(\mathrm{Id}-P_{Q_n}\right)Ay_n\right\|^2-\alpha\beta_n\left(\|w_n-y_n\|^2+\|z_n-y_n\|^2\right)\\&\leq \|w_n-p\|^2-\alpha\left(\frac{1}{2}\beta_n+(1-\alpha)\right)\|w_n-z_n\|^2\\&\quad-\frac{2\alpha\beta\mu}{\|A\|^2}\left\|\left(\mathrm{Id}-P_{Q_n}\right)Ay_n\right\|^2.\end{aligned} \tag{3.16}$$

Taking $n+1:=2n+2$ in (3.16), one sees that

$$\begin{aligned}\|x_{2n+2}-p\|^2 \leq \|w_{2n+1}-p\|^2-\alpha\left(\frac{1}{2}\beta_{2n+1}+(1-\alpha)\right)\|w_{2n+1}-z_{2n+1}\|^2\\-\frac{2\alpha\beta\mu}{\|A\|^2}\left\|\left(\mathrm{Id}-P_{Q_{2n+1}}\right)Ay_{2n+1}\right\|^2.\end{aligned} \tag{3.17}$$

Letting $n+1:=2n+1$ in (3.16) (noting that $w_{2n}=x_{2n}$), we observe that

$$\begin{aligned}\|x_{2n+1}-p\|^2 \leq \|x_{2n}-p\|^2-\alpha\left(\frac{1}{2}\beta_{2n}+(1-\alpha)\right)\|x_{2n}-z_{2n}\|^2\\-\frac{2\alpha\beta\mu}{\|A\|^2}\left\|\left(\mathrm{Id}-P_{Q_{2n}}\right)Ay_{2n}\right\|^2.\end{aligned} \tag{3.18}$$

It follows from the definition of w_{2n+1} and (2.7) that

$$\begin{aligned}\|w_{2n+1}-p\|^2 &= \|(1+\theta_{2n+1})(x_{2n+1}-p)-\theta_{2n+1}(x_{2n}-p)\|^2\\&=(1+\theta_{2n+1})\|x_{2n+1}-p\|^2-\theta_{2n+1}\|x_{2n}-p\|^2\\&\quad+\theta_{2n+1}(1+\theta_{2n+1})\|x_{2n+1}-x_{2n}\|^2.\end{aligned} \tag{3.19}$$

Using the definition of x_{2n+1} and noting that $w_{2n}=x_{2n}$, one obtains

$$\|x_{2n+1}-x_{2n}\|^2=\alpha^2\|z_{2n}-x_{2n}\|^2. \tag{3.20}$$

Substituting (3.18) and (3.20) into (3.19), we have

$$\begin{aligned}&\|w_{2n+1}-p\|^2\\&\leq(1+\theta_{2n+1})\left[\|x_{2n}-p\|^2-\alpha\left(\frac{1}{2}\beta_{2n}+(1-\alpha)\right)\|x_{2n}-z_{2n}\|^2\right]\\&\quad-\theta_{2n+1}\|x_{2n}-p\|^2+\theta_{2n+1}(1+\theta_{2n+1})\|x_{2n+1}-x_{2n}\|^2\\&\quad-(1+\theta_{2n+1})\frac{2\alpha\beta\mu}{\|A\|^2}\left\|\left(\mathrm{Id}-P_{Q_{2n}}\right)Ay_{2n}\right\|^2\\&=\|x_{2n}-p\|^2-\alpha(1+\theta_{2n+1})\left(\frac{1}{2}\beta_{2n}+(1-\alpha)-\theta_{2n+1}\alpha\right)\|x_{2n}-z_{2n}\|^2\\&\quad-(1+\theta_{2n+1})\frac{2\alpha\beta\mu}{\|A\|^2}\left\|\left(\mathrm{Id}-P_{Q_{2n}}\right)Ay_{2n}\right\|^2.\end{aligned} \tag{3.21}$$

Combining (3.17) and (3.21), we obtain

$$\|x_{2n+2}-p\|^2 \leq \|x_{2n}-p\|^2 - \alpha\left(\frac{1}{2}\beta_{2n+1}+(1-\alpha)\right)\|w_{2n+1}-z_{2n+1}\|^2 \\ -\alpha\left(1+\theta_{2n+1}\right)\left(\frac{1}{2}\beta_{2n}+(1-\alpha)-\theta_{2n+1}\alpha\right)\|x_{2n}-z_{2n}\|^2 \\ -\left(1+\theta_{2n+1}\right)\frac{2\alpha\beta\mu}{\|A\|^2}\left\|\left(\mathrm{Id}-P_{Q_{2n}}\right)Ay_{2n}\right\|^2. \tag{3.22}$$

Since $\alpha \in (0,1]$, $0 \leq \theta_{2n+1} \leq \theta < \frac{\beta^*+2(1-\alpha)}{2\alpha}$, and $\beta_{2n}, \beta_{2n+1} > 0$, $\forall n \geq N_0$, we have

$$\alpha\left(\frac{1}{2}\beta_{2n+1}+(1-\alpha)\right) > 0,\ \forall n \geq N_0$$

and

$$\alpha\left(1+\theta_{2n+1}\right)\left(\frac{1}{2}\beta_{2n}+(1-\alpha)-\theta_{2n+1}\alpha\right) > 0,\ \forall n \geq N_0.$$

Thus it follows from (3.22) that

$$\|x_{2n+2}-p\| \leq \|x_{2n}-p\|,\ \ \forall n \geq N_0.$$

This implies that $\{\|x_{2n}-p\|\}$ and $\{x_{2n}\}$ are bounded. Moreover, $\lim_{n\to\infty}\|x_{2n}-p\|$ exists. Therefore, we conclude from (3.22) that

$$\lim_{n\to\infty}\|x_{2n}-z_{2n}\| = 0 \ \text{ and } \lim_{n\to\infty}\left\|\left(\mathrm{Id}-P_{Q_{2n}}\right)Ay_{2n}\right\| = 0. \tag{3.23}$$

By the fact that $\{x_{2n}\}$ is bounded and (3.23), one obtains that $\{z_{2n}\}$ is also bounded. By virtue of (3.20) and (3.23), one sees that $\lim_{n\to\infty}\|x_{2n+1}-x_{2n}\| = 0$. In view of Lemma 3.2, one can show that

$$\|z_{2n}-p\|^2 \leq \|x_{2n}-p\|^2 - \beta_{2n}\left(\|x_{2n}-y_{2n}\|^2+\|z_{2n}-y_{2n}\|^2\right). \tag{3.24}$$

Since $\{\|x_{2n}-p\|\}$ and $\{\|z_{2n}-p\|\}$ are bounded, we deduce from (3.23) that

$$\begin{aligned}
&\|x_{2n}-p\|^2 - \|z_{2n}-p\|^2 \\
&= \left(\|x_{2n}-p\|+\|z_{2n}-p\|\right)\left(\|x_{2n}-p\|-\|z_{2n}-p\|\right) \\
&\leq \left(\|x_{2n}-p\|+\|z_{2n}-p\|\right)\|x_{2n}-z_{2n}\| \to 0, \quad \text{as } n\to\infty.
\end{aligned}$$

This together with (3.24) implies that

$$\lim_{n\to\infty}\left(\|x_{2n}-y_{2n}\|^2+\|z_{2n}-y_{2n}\|^2\right) = 0.$$

That is,

$$\lim_{n\to\infty}\|x_{2n}-y_{2n}\| = 0 \ \text{ and } \lim_{n\to\infty}\|z_{2n}-y_{2n}\| = 0. \tag{3.25}$$

The proof is completed. □

Lemma 3.4 *Suppose that the sequence $\{x_n\}$ is generated by Algorithm 3.1. Let $x^* \in \mathcal{H}$ denote the weak limit of the subsequence $\left\{x_{2n_j}\right\}$ of $\{x_{2n}\}$. Then $x^* \in \Gamma$.*

Proof Since $\mathrm{Id} - P_{Q_{2n}}$ is nonexpansive, one obtains

$$\begin{aligned}\left\|\left(\mathrm{Id} - P_{Q_{2n}}\right) Ax_{2n}\right\| &\leq \left\|\left(\mathrm{Id} - P_{Q_{2n}}\right) Ax_{2n} - \left(\mathrm{Id} - P_{Q_{2n}}\right) Ay_{2n}\right\| \\ &\quad + \left\|\left(\mathrm{Id} - P_{Q_{2n}}\right) Ay_{2n}\right\| \\ &\leq \|Ax_{2n} - Ay_{2n}\| + \left\|\left(\mathrm{Id} - P_{Q_{2n}}\right) Ay_{2n}\right\| \\ &\leq \|A\| \, \|x_{2n} - y_{2n}\| + \left\|\left(\mathrm{Id} - P_{Q_{2n}}\right) Ay_{2n}\right\| .\end{aligned} \tag{3.26}$$

In view of (3.23), (3.25), and the fact that A is a bounded operator, we infer from (3.26) that

$$\lim_{n\to\infty} \left\|\left(\mathrm{Id} - P_{Q_{2n}}\right) Ax_{2n}\right\| = 0. \tag{3.27}$$

From the assumption that ∂q is bounded on bounded sets, there exists a positive constant δ such that $\|\zeta_{2n}\| \leq \delta$. Combining the definition of Q_{2n}, $P_{Q_{2n}} Ax_{2n} \in Q_{2n}$, and (3.27), we obtain

$$\begin{aligned} q\left(Aw_{2n}\right) = q\left(Ax_{2n}\right) &\leq \left\langle \zeta_{2n}, Ax_{2n} - P_{Q_{2n}} Ax_{2n}\right\rangle \\ &\leq \delta \left\|\left(\mathrm{Id} - P_{Q_{2n}}\right) Ax_{2n}\right\| \to 0, \text{ as } n \to \infty. \end{aligned} \tag{3.28}$$

According to the fact that $\{x_{2n}\}$ is bounded, there exists a subsequence $\left\{x_{2n_j}\right\}$ of $\{x_{2n}\}$ such that $x_{2n_j} \rightharpoonup x^*$. Using the weakly lower semicontinuity of q and (3.28), one sees that

$$q\left(Ax^*\right) \leq \liminf_{n\to\infty} q\left(Ax_{2n_j}\right) \leq 0,$$

which implies that $Ax^* \in Q$. On the other hand, there exists a positive constant ν such that $\|\eta_{2n_j}\| \leq \nu$ since ∂c is bounded on bounded sets. By the definition of C_{2n_j}, $y_{2n_j} \in C_{2n_j}$, and (3.23), we have

$$\begin{aligned} c\left(w_{2n_j}\right) = c\left(x_{2n_j}\right) &\leq \left\langle \eta_{2n_j}, x_{2n_j} - y_{2n_j}\right\rangle \\ &\leq \nu \left\|x_{2n_j} - y_{2n_j}\right\| \to 0, \text{ as } j \to \infty. \end{aligned} \tag{3.29}$$

Combining the weakly lower semicontinuity of c, $x_{2n_j} \rightharpoonup x^*$, and (3.29), we infer that

$$c\left(x^*\right) \leq \liminf_{j\to\infty} c\left(x_{2n_j}\right) \leq 0,$$

which implies that $x^* \in C$. Therefore we conclude that $x^* \in \Gamma = \{x^* \in C : Ax^* \in Q\}$. This is the desired result. □

Now we can prove the weak convergence theorem of Algorithm 3.1.

Theorem 3.1 *Let Assumptions (A1)–(A2) and (C1)–(C3) hold and the sequence $\{x_n\}$ be generated by Algorithm 3.1. Then $\{x_n\}$ converges weakly to a point in Γ.*

Proof By Lemma 3.3, one sees that sequence $\{x_{2n}\}$ is bounded, which implies $\{x_{2n}\}$ has weakly convergent subsequences. Let $p \in \mathcal{H}$ be the weak limit of such a subsequence $\left\{x_{2n_k}\right\}$ of $\{x_{2n}\}$. By means of Lemma 3.3 and Lemma 3.4, we obtain that $\lim_{n\to\infty} \|x_{2n} - p\|$ exists and $p \in \Gamma$, respectively. In view of Lemma 2.2, we have that $\{x_{2n}\}$ converges weakly to a point in Γ. The rest of the proof is required to show that sequence $\{x_{2n+1}\}$ also converges weakly to p. Taking $z \in \mathcal{H}$, we have

$$\begin{aligned}|\langle x_{2n+1} - p, z\rangle| &\leq |\langle x_{2n} - p, z\rangle| + |\langle x_{2n+1} - x_{2n}, z\rangle| \\ &\leq |\langle x_{2n} - p, z\rangle| + \|x_{2n+1} - x_{2n}\| \, \|z\| \to 0.\end{aligned}$$

By virtue of $\lim_{n\to\infty} \langle x_{2n} - p, z\rangle = 0$ and $\lim_{n\to\infty} \|x_{2n+1} - x_{2n}\| = 0$ in Lemma 3.3, we deduce that $\lim_{n\to\infty} |\langle x_{2n+1} - p, z\rangle| = 0$. This together with the arbitrariness of z further implies that $\{x_{2n+1}\}$ converges weakly to p. Consequently the whole sequence $\{x_n\}$ converges weakly to a point $p \in \Gamma$. This completes the proof of Theorem 3.1. □

Next, we provide a modified version of Algorithm 3.1. This new version is computed differently from Algorithm 3.1 in Steps 2 and 3. We shall need the following condition for Algorithm 3.2.

(C4) Let $\alpha \in (0, 1]$, $\beta \in (1/(2 - \mu), 1/\mu)$, and $0 \leq \theta_n \leq \theta < \frac{\beta^\dagger + 2(1-\alpha)}{2\alpha}$, where $\beta^\dagger = 2 - \frac{1}{\beta} - \mu$ when $\beta \in (1/(2-\mu), 1]$ and $\beta^\dagger = \frac{1}{\beta} - \mu$ when $\beta \in (1, 1/\mu)$.

Now we are ready to present the proposed Algorithm 3.2 as shown below.

Algorithm 3.2

Initialization: Take λ_1, μ, α, β, $\{\xi_n\}$, $\{\rho_n\}$, and $\{\theta_n\}$ satisfies Conditions (C2) and (C4). Choose initial points $x_0, x_1 \in \mathcal{H}$. Set $n := 1$.

Iterative Steps: Calculate the next iteration point x_{n+1} as follows:

Step 1. Compute w_n by (3.3).

Step 2. Compute $y_n = P_{C_n}(w_n - \beta\lambda_n \nabla f_n(w_n))$. If $y_n = w_n$ then the iteration stops and y_n is the solution of SFP; otherwise, turn to *Step 3*.

Step 3. Compute $z_n = P_{C_n}(w_n - \lambda_n \nabla f_n(y_n))$.

Step 4. Compute $x_{n+1} = (1-\alpha) w_n + \alpha z_n$, and update the next step size λ_{n+1} by (3.4).
Set $n := n + 1$ and go to *Step 1*.

We first prove the following Lemma 3.5 for analyzing the convergence of Algorithm 3.2.

Lemma 3.5 *Suppose that $\{w_n\}$, $\{y_n\}$, and $\{z_n\}$ are three sequences generated by Algorithm 3.2. Let $p \in \Gamma$. Then*

$$\begin{aligned}\|z_n - p\|^2 \leq \|w_n - p\|^2 &- \frac{2\mu}{\|A\|^2} \left\|\left(\mathrm{Id} - P_{Q_n}\right) A y_n\right\|^2 \\ &- \beta_n^\dagger \left(\|w_n - y_n\|^2 + \|z_n - y_n\|^2\right),\end{aligned}$$

where

$$\beta_n^\dagger := \begin{cases} 2 - \frac{1}{\beta} - \frac{\mu\lambda_n}{\lambda_{n+1}}, & \text{if } \beta \in (1/(2-\mu), 1], \\ \frac{1}{\beta} - \frac{\mu\lambda_n}{\lambda_{n+1}}, & \text{if } \beta \in (1, 1/\mu). \end{cases}$$

Proof Similar to (3.5), we have

$$\begin{aligned} \|z_n - p\|^2 \leq \|w_n - p\|^2 - \|z_n - w_n\|^2 - 2\lambda_n \langle \nabla f_n(y_n), z_n - y_n \rangle \\ - 2\lambda_n \langle \nabla f_n(y_n), y_n - p \rangle . \end{aligned} \tag{3.30}$$

From (3.6) and $\lambda_n \geq \frac{\mu}{\|A\|^2}$, one obtains

$$2\lambda_n \langle \nabla f_n(y_n), y_n - p \rangle \geq \frac{2\mu}{\|A\|^2} \left\| \left(\mathrm{Id} - P_{Q_n}\right) A y_n \right\|^2 . \tag{3.31}$$

Note that

$$-\|w_n - z_n\|^2 = -\|w_n - y_n\|^2 - \|y_n - z_n\|^2 + 2\langle w_n - y_n, z_n - y_n \rangle , \tag{3.32}$$

and

$$\begin{aligned} & 2\langle w_n - y_n, z_n - y_n \rangle \\ &= 2\langle w_n - \beta\lambda_n \nabla f_n(w_n) - y_n, z_n - y_n \rangle + 2\langle \beta\lambda_n \nabla f_n(y_n), z_n - y_n \rangle \\ &\quad + 2\beta\lambda_n \langle \nabla f_n(w_n) - \nabla f_n(y_n), z_n - y_n \rangle . \end{aligned} \tag{3.33}$$

By the definition of y_n, (2.9), and the fact that $z_n \in C_n$, we have

$$\langle w_n - \beta\lambda_n \nabla f_n(w_n) - y_n, z_n - y_n \rangle \leq 0. \tag{3.34}$$

Using the definition of λ_{n+1} in (3.4), we obtain

$$2\beta\lambda_n \langle \nabla f_n(w_n) - \nabla f_n(y_n), z_n - y_n \rangle \leq \frac{\beta\mu\lambda_n}{\lambda_{n+1}} \left(\|w_n - y_n\|^2 + \|z_n - y_n\|^2 \right). \tag{3.35}$$

Substituting (3.33), (3.34), and (3.35) into (3.32), we have

$$\begin{aligned} -\|w_n - z_n\|^2 \leq -\left(1 - \frac{\beta\mu\lambda_n}{\lambda_{n+1}}\right)\left(\|w_n - y_n\|^2 + \|z_n - y_n\|^2\right) \\ + 2\beta \langle \lambda_n \nabla f_n(y_n), z_n - y_n \rangle , \end{aligned}$$

which yields

$$\begin{aligned} -2\langle \lambda_n \nabla f_n(y_n), z_n - y_n \rangle \leq -\left(\frac{1}{\beta} - \frac{\mu\lambda_n}{\lambda_{n+1}}\right)\left(\|w_n - y_n\|^2 + \|z_n - y_n\|^2\right) \\ + \frac{1}{\beta}\|w_n - z_n\|^2 , \quad \forall \beta > 0. \end{aligned} \tag{3.36}$$

Combining (3.30), (3.31), and (3.36), we obtain

$$\|z_n - p\|^2 \le \|w_n - p\|^2 - \frac{2\mu}{\|A\|^2} \left\| \left(\mathrm{Id} - P_{Q_n}\right) Ay_n \right\|^2 - \left(1 - \frac{1}{\beta}\right) \|w_n - z_n\|^2 \\ - \left(\frac{1}{\beta} - \frac{\mu\lambda_n}{\lambda_{n+1}}\right) \left(\|w_n - y_n\|^2 + \|z_n - y_n\|^2\right), \quad \forall \beta > 0.$$

This together with (3.14) implies that

$$\|z_n - p\|^2 \le \|w_n - p\|^2 - \frac{2\mu}{\|A\|^2} \left\| \left(\mathrm{Id} - P_{Q_n}\right) Ay_n \right\|^2 \\ - \beta_n^\dagger \left(\|w_n - y_n\|^2 + \|z_n - y_n\|^2\right),$$

where $\beta_n^\dagger$ is defined by

$$\beta_n^\dagger := \begin{cases} 2 - \frac{1}{\beta} - \frac{\mu\lambda_n}{\lambda_{n+1}}, & \text{if } \beta \in (1/(2-\mu), 1], \\ \frac{1}{\beta} - \frac{\mu\lambda_n}{\lambda_{n+1}}, & \text{if } \beta \in (1, 1/\mu). \end{cases}$$

This completes the proof. □

Remark 3.4 Since $\lim_{n\to\infty} \lambda_n$ exists by means of Lemma 3.1, we have

$$\beta^\dagger := \lim_{n\to\infty} \beta_n^\dagger = \begin{cases} 2 - \frac{1}{\beta} - \mu, & \text{if } \beta \in (1/(2-\mu), 1], \\ \frac{1}{\beta} - \mu, & \text{if } \beta \in (1, 1/\mu). \end{cases}$$

Hence, $\lim_{n\to\infty} \beta_n^\dagger > 0$ for all $\beta \in (1/(2-\mu), 1/\mu)$. There exists a positive constant N_1 such that $\beta_n^\dagger > 0$ holds for all $n \ge N_1$.

Theorem 3.2 *Let Assumptions (A1), (A2), (C1), (C2), and (C4) hold and the sequence* $\{x_n\}$ *be created by Algorithm 3.2. Then* $\{x_n\}$ *converges weakly to a point in* Γ.

Proof Fix $p \in \Gamma$. With the help of the proof of Lemma 3.3, we can easily obtain

$$\|x_{2n+2} - p\|^2 \le \|x_{2n} - p\|^2 - \alpha \left(\frac{1}{2}\beta_{2n+1}^\dagger + (1-\alpha)\right) \|w_{2n+1} - z_{2n+1}\|^2 \\ - \alpha\left(1 + \theta_{2n+1}\right) \left(\frac{1}{2}\beta_{2n}^\dagger + (1-\alpha) - \theta_{2n+1}\alpha\right) \|x_{2n} - z_{2n}\|^2 \\ - \left(1 + \theta_{2n+1}\right) \frac{2\alpha\mu}{\|A\|^2} \left\| \left(\mathrm{Id} - P_{Q_{2n}}\right) Ay_{2n} \right\|^2. \tag{3.37}$$

Since $\alpha \in (0, 1]$, $0 \le \theta_{2n+1} \le \theta < \frac{\beta^\dagger + 2(1-\alpha)}{2\alpha}$, and $\beta_{2n}^\dagger, \beta_{2n+1}^\dagger > 0, \forall n \ge N_1$, we conclude from (3.37) that

$$\|x_{2n+2} - p\| \le \|x_{2n} - p\|, \quad \forall n \ge N_1.$$

This implies that the sequence $\{x_{2n}\}$ is Fejér monotone with respect to Γ. Furthermore, we obtain that $\{x_{2n}\}$ is bounded and $\lim_{n\to\infty}\|x_{2n}-p\|$ exists. Thus one can show from (3.37) that

$$\lim_{n\to\infty}\|x_{2n}-z_{2n}\|=0 \text{ and } \lim_{n\to\infty}\left\|\left(\mathrm{Id}-P_{Q_{2n}}\right)Ay_{2n}\right\|=0.$$

In the light of Lemma 3.3, we have

$$\lim_{n\to\infty}\|x_{2n+1}-x_{2n}\|=0 \text{ and } \lim_{n\to\infty}\|x_{2n}-y_{2n}\|=0.$$

According to Lemma 3.4, we can also obtain that the weak limit of the subsequence $\{x_{2n_j}\}$ of $\{x_{2n}\}$ is in Γ, which is the solution set of SFP (1.1). The remaining part of the proof is similar to Theorem 3.1 and thus we omit it. The proof is completed. □

3.2 Two modified adaptive projection and contraction algorithms

In this subsection, we propose two alternated inertial projection and contraction algorithms with non-monotonic step size and relaxation effects to find the solution of the split feasibility problem in real Hilbert spaces. To begin with, we assume the following condition for Algorithm 3.3.

(C5) Let $\alpha\in(0,1]$, $\beta\in(0,1/\mu)$, $\tau\in(0,2)$, and $0\le\theta_n\le\theta<\frac{2}{\tau\alpha}-1$.

Algorithm 3.3 is stated as follows.

Algorithm 3.3

Initialization: Take $\lambda_1,\mu,\alpha,\beta,\tau,\{\xi_n\},\{\rho_n\}$, and $\{\theta_n\}$ satisfies Conditions (C2) and (C5). Choose initial points $x_0,x_1\in\mathcal{H}$. Set $n:=1$.

Iterative Steps: Calculate the next iteration point x_{n+1} as follows:

Step 1. Compute w_n via (3.3).

Step 2. Compute $y_n=P_{C_n}(w_n-\beta\lambda_n\nabla f_n(w_n))$. If $y_n=w_n$ then the iteration stops and y_n is the solution of SFP; otherwise, go to *Step 3*.

Step 3. Compute $z_n=w_n-\tau\varphi_n d_n$, where d_n and φ_n are defined by

$$\begin{aligned} d_n&=w_n-y_n-\beta\lambda_n\left(\nabla f_n(w_n)-\nabla f_n(y_n)\right),\\ \varphi_n&=\frac{\langle w_n-y_n,d_n\rangle+\beta\lambda_n\left\|\left(\mathrm{Id}-P_{Q_n}\right)Ay_n\right\|^2}{\|d_n\|^2}. \end{aligned}\tag{3.38}$$

Step 4. Compute $x_{n+1}=(1-\alpha)w_n+\alpha z_n$, and update the next step size λ_{n+1} by (3.4).
Set $n:=n+1$ and go to *Step 1*.

Remark 3.5 We show that $y_n=w_n$ if and only if $d_n=0$. In this case, the iterations of Algorithm 3.3 stop and y_n is the solution of the SFP. Indeed, it follows from the definition of d_n in (3.38) and (3.4) that

$$
\begin{aligned}
\|d_n\| &\geq \|w_n - y_n\| - \beta\lambda_n \|\nabla f_n(w_n) - \nabla f_n(y_n)\| \\
&\geq \|w_n - y_n\| - \frac{\beta\mu\lambda_n}{\lambda_{n+1}} \|w_n - y_n\| \\
&= \left(1 - \frac{\beta\mu\lambda_n}{\lambda_{n+1}}\right) \|w_n - y_n\| .
\end{aligned}
\tag{3.39}
$$

On the other hand, one can show that

$$
\|d_n\| \leq \left(1 + \frac{\beta\mu\lambda_n}{\lambda_{n+1}}\right) \|w_n - y_n\| . \tag{3.40}
$$

In view of Lemma 3.1 and $\beta \in (0, 1/\mu)$ in Condition (C5), one obtains

$$
\lim_{n\to\infty} \left(1 - \frac{\beta\mu\lambda_n}{\lambda_{n+1}}\right) = 1 - \beta\mu > 0.
$$

Thus we conclude from (3.39) and (3.40) that $w_n = y_n$ if and only if $d_n = 0$. By Remark 3.1, we know that the iterations of Algorithm 3.3 terminate when $w_n = y_n$ or $d_n = 0$.

The following two lemmas are keys to the convergence analysis of Algorithm 3.3.

Lemma 3.6 *Let $\{w_n\}$, $\{y_n\}$, and $\{z_n\}$ be three sequences generated by Algorithm 3.3. Then,*

$$
\|z_n - p\|^2 \leq \|w_n - p\|^2 - \frac{2-\tau}{\tau} \|w_n - z_n\|^2, \quad \forall p \in \Gamma,
$$

and

$$
\|w_n - y_n\|^2 \leq \frac{\left(1 + \frac{\beta\mu\lambda_n}{\lambda_{n+1}}\right)^2}{\tau^2 \left(1 - \frac{\beta\mu\lambda_n}{\lambda_{n+1}}\right)^2} \|w_n - z_n\|^2.
$$

Proof Let $p \in \Gamma$. It follows from the definition of z_n that

$$
\|z_n - p\|^2 = \|w_n - p\|^2 - 2\tau\varphi_n \langle w_n - p, d_n \rangle + \tau^2 \varphi_n^2 \|d_n\|^2. \tag{3.41}
$$

Combining the definition of y_n and the projection property (2.9), we deduce that

$$
\langle w_n - \beta\lambda_n \nabla f_n(w_n) - y_n, y_n - p \rangle \geq 0. \tag{3.42}
$$

Using the definition of d_n and (3.42), one obtains

$$
\begin{aligned}
\langle y_n - p, d_n \rangle &= \langle y_n - p, w_n - \beta\lambda_n \nabla f_n(w_n) - y_n \rangle + \langle y_n - p, \beta\lambda_n \nabla f_n(y_n) \rangle \\
&\geq \langle y_n - p, \beta\lambda_n \nabla f_n(y_n) \rangle .
\end{aligned}
$$

This together with (3.6) yields

$$\begin{aligned}\langle w_n - p, d_n\rangle &= \langle w_n - y_n, d_n\rangle + \langle y_n - p, d_n\rangle \\ &\geq \langle w_n - y_n, d_n\rangle + \beta\lambda_n \left\|\left(\mathrm{Id} - P_{Q_n}\right) Ay_n\right\|^2 \\ &= \varphi_n \left\|d_n\right\|^2.\end{aligned} \tag{3.43}$$

Note that $\|w_n - z_n\| = \tau\varphi_n d_n$. Combining (3.41) and (3.43), we arrive at the first conclusion

$$\begin{aligned}\|z_n - p\|^2 &\leq \|w_n - p\|^2 - 2\tau\varphi_n^2\|d_n\|^2 + \tau^2\varphi_n^2\|d_n\|^2 \\ &= \|w_n - p\|^2 - \frac{2-\tau}{\tau}\|w_n - z_n\|^2.\end{aligned}$$

According to the definition of φ_n and (3.4), one has

$$\begin{aligned}\varphi_n \left\|d_n\right\|^2 &\geq \langle w_n - y_n, d_n\rangle \\ &\geq \|w_n - y_n\|^2 - \beta\lambda_n \left\|\nabla f_n(w_n) - \nabla f_n(y_n)\right\| \|w_n - y_n\| \\ &\geq \left(1 - \frac{\beta\mu\lambda_n}{\lambda_{n+1}}\right)\|w_n - y_n\|^2.\end{aligned} \tag{3.44}$$

Combining (3.40) and (3.44), we obtain

$$\varphi_n \geq \frac{\langle w_n - y_n, d_n\rangle}{\|d_n\|^2} \geq \frac{1 - \frac{\beta\mu\lambda_n}{\lambda_{n+1}}}{\left(1 + \frac{\beta\mu\lambda_n}{\lambda_{n+1}}\right)^2}. \tag{3.45}$$

Using (3.44) and (3.45), we have

$$\varphi_n^2 \left\|d_n\right\|^2 \geq \frac{\left(1 - \frac{\beta\mu\lambda_n}{\lambda_{n+1}}\right)^2}{\left(1 + \frac{\beta\mu\lambda_n}{\lambda_{n+1}}\right)^2}\|w_n - y_n\|^2, \tag{3.46}$$

By the definition of z_n and (3.46), we deduce that

$$\|z_n - w_n\|^2 = \tau^2\varphi_n^2 \left\|d_n\right\|^2 \geq \tau^2\frac{\left(1 - \frac{\beta\mu\lambda_n}{\lambda_{n+1}}\right)^2}{\left(1 + \frac{\beta\mu\lambda_n}{\lambda_{n+1}}\right)^2}\|w_n - y_n\|^2,$$

which is equivalent to the second conclusion as required. □

Lemma 3.7 *Let the sequence $\{x_n\}$ be formed by Algorithm 3.3. Then the even subsequence $\{x_{2n}\}$ is bounded and it is Fejér monotone with respect to the solution set Γ. Let $p \in \Gamma$. Then $\lim_{n\to\infty}\|x_{2n} - p\|$ exists and $\lim_{n\to\infty}\|x_{2n} - y_{2n}\| = 0$.*

Proof Combining (3.15) and Lemma 3.6, we obtain

$$\begin{aligned}\|x_{n+1}-p\|^2 &\leq (1-\alpha)\|w_n-p\|^2+\alpha\|w_n-p\|^2-\alpha(1-\alpha)\|w_n-z_n\|^2\\ &\quad -\alpha\frac{2-\tau}{\tau}\|w_n-z_n\|^2\\ &=\|w_n-p\|^2-\alpha\left(\frac{2}{\tau}-\alpha\right)\|w_n-z_n\|^2 .\end{aligned} \tag{3.47}$$

Letting $n+1:=2n+2$ in (3.47), we have

$$\|x_{2n+2}-p\|^2 \leq \|w_{2n+1}-p\|^2-\alpha\left(\frac{2}{\tau}-\alpha\right)\|w_{2n+1}-z_{2n+1}\|^2 . \tag{3.48}$$

Letting $n+1:=2n+1$ in (3.47) (noting that $w_{2n}=x_{2n}$), we obtain

$$\|x_{2n+1}-p\|^2 \leq \|x_{2n}-p\|^2-\alpha\left(\frac{2}{\tau}-\alpha\right)\|x_{2n}-z_{2n}\|^2 . \tag{3.49}$$

Substituting (3.20) and (3.49) into (3.19), one sees that

$$\begin{aligned}&\|w_{2n+1}-p\|^2\\ &\leq (1+\theta_{2n+1})\left[\|x_{2n}-p\|^2-\alpha\left(\frac{2}{\tau}-\alpha\right)\|x_{2n}-z_{2n}\|^2\right]\\ &\quad -\theta_{2n+1}\|x_{2n}-p\|^2+\theta_{2n+1}(1+\theta_{2n+1})\|x_{2n+1}-x_{2n}\|^2\\ &=\|x_{2n}-p\|^2-\alpha(1+\theta_{2n+1})\left(\frac{2}{\tau}-\alpha-\theta_{2n+1}\alpha\right)\|x_{2n}-z_{2n}\|^2 .\end{aligned} \tag{3.50}$$

Combining (3.48) and (3.50), we have

$$\|x_{2n+2}-p\|^2 \leq \|x_{2n}-p\|^2-\alpha(1+\theta_{2n+1})\left(\frac{2}{\tau}-\alpha-\theta_{2n+1}\alpha\right)\|x_{2n}-z_{2n}\|^2 . \tag{3.51}$$

Since $\alpha\in(0,1]$ and $0\leq\theta_{2n+1}\leq\theta<\frac{2}{\tau\alpha}-1$, we observe that

$$\alpha(1+\theta_{2n+1})\left(\frac{2}{\tau}-\alpha-\theta_{2n+1}\alpha\right)>0.$$

It follows from (3.51) that

$$\|x_{2n+2}-p\| \leq \|x_{2n}-p\|, \quad \forall n\geq 1. \tag{3.52}$$

This shows that $\{\|x_{2n}-p\|\}$ and $\{x_{2n}\}$ are bounded. So $\lim_{n\to\infty}\|x_{2n}-p\|$ exists, which together with (3.51) implies that

$$\lim_{n\to\infty}\|x_{2n}-z_{2n}\|=0. \tag{3.53}$$

This, combined with the definition of z_{2n}, yields

$$\lim_{n\to\infty} \varphi_{2n}^2 \|d_{2n}\|^2 = \lim_{n\to\infty} \|x_{2n} - z_{2n}\| = 0. \tag{3.54}$$

By virtue of (3.20), one sees that $\lim_{n\to\infty} \|x_{2n+1} - x_{2n}\| = 0$. Combining (3.53) and Lemma 3.6, one obtains

$$\lim_{n\to\infty} \|x_{2n} - y_{2n}\| = 0. \tag{3.55}$$

It follows from (3.45) that $\varphi_{2n} > 0$ for some $n \geq N_0$. In view of the definition of φ_{2n} and (3.44), we have

$$\begin{aligned} \varphi_{2n}^2 \|d_{2n}\|^2 &= \varphi_{2n} \left[\langle w_{2n} - y_{2n}, d_{2n} \rangle + \beta\lambda_{2n} \| \left(\mathrm{Id} - P_{Q_{2n}}\right) Ay_{2n}\|^2 \right] \\ &\geq \varphi_{2n} \left[\left(1 - \frac{\beta\mu\lambda_n}{\lambda_{n+1}}\right) \|x_{2n} - y_{2n}\|^2 + \beta\lambda_{2n} \left\| \left(\mathrm{Id} - P_{Q_{2n}}\right) Ay_{2n}\right\|^2 \right]. \end{aligned} \tag{3.56}$$

Using (3.54), (3.55), and (3.56), we arrive at the conclusion

$$\lim_{n\to\infty} \left\| \left(\mathrm{Id} - P_{Q_{2n}}\right) Ay_{2n}\right\| = 0. \tag{3.57}$$

This completes the proof. □

Now we are in a position to prove the weak convergence of Algorithm 3.3.

Theorem 3.3 *Let Assumptions (A1), (A2), (C1), (C2), and (C5) hold and the sequence $\{x_n\}$ be formed by Algorithm 3.3. Then $\{x_n\}$ converges weakly to a point in Γ.*

Proof Combining (3.55) and (3.57) in Lemma 3.7, and Lemma 3.3, one can show that the weak limit point of the subsequence $\{x_{2n_j}\}$ of $\{x_{2n}\}$ is in Γ. It follows from the analysis of Theorem 3.1 that the entire sequence $\{x_n\}$ formed by Algorithm 3.3 converges weakly to a point in Γ. This is the desired conclusion. □

Next, we give another version of an adaptive alternated inertial projection and contraction algorithm to solve the SFP, which is different from the suggested Algorithm 3.3 that computes z_n in Step 3. We shall assume the following condition for Algorithm 3.4.

(C6) Let $\alpha \in (0, 1]$, $\beta \in (\tau/2, 1/\mu)$, $\tau \in (0, 2/\mu)$, and $-1 \leq \theta_n \leq \theta < 0$.

Algorithm 3.4, the last iterative scheme in this paper, is given below.

Algorithm 3.4

Initialization: Take $\lambda_1, \mu, \alpha, \beta, \tau, \{\xi_n\}, \{\rho_n\}$, and $\{\theta_n\}$ satisfies Conditions (C2) and (C6). Choose initial points $x_0, x_1 \in \mathcal{H}$. Set $n := 1$.
Iterative Steps: Calculate the next iteration point x_{n+1} as follows:
Step 1. Compute w_n by (3.3).
Step 2. Compute $y_n = P_{C_n}(w_n - \beta\lambda_n \nabla f_n(w_n))$.
Step 3. Compute $z_n = P_{C_n}(w_n - \tau\varphi_n\lambda_n \nabla f_n(y_n))$, where φ_n is defined in (3.38).
Step 4. Compute $x_{n+1} = (1-\alpha) w_n + \alpha z_n$, and update the next step size λ_{n+1} by (3.4).
Set $n := n + 1$ and go to *Step 1*.

In a similar way as the convergence analysis of Algorithm 3.3, we perform the convergence analysis of Algorithm 3.4 via the following two lemmas.

Lemma 3.8 *Assume that sequences* $\{w_n\}$, $\{y_n\}$, *and* $\{z_n\}$ *are designed by Algorithm 3.4. Then*

$$\begin{aligned}\|z_n - p\|^2 \leq \|w_n - p\|^2 - \|w_n - z_n - \frac{\tau}{\beta}\varphi_n d_n\|^2 \\ - \frac{\tau}{\beta^2}(2\beta - \tau)\varphi_n^2\|d_n\|^2, \quad p \in \Gamma.\end{aligned}$$

Proof Similar to (3.5), one sees that

$$\begin{aligned}\|z_n - p\|^2 \leq \|w_n - p\|^2 - \|z_n - w_n\|^2 - 2\tau\varphi_n\lambda_n\langle z_n - y_n, \nabla f_n(y_n)\rangle \\ - 2\tau\varphi_n\lambda_n\langle y_n - p, \nabla f_n(y_n)\rangle.\end{aligned} \tag{3.58}$$

By the definition of φ_n and (3.44), one finds $\varphi_n > 0$. From (3.6), one obtains

$$-2\tau\varphi_n\lambda_n\left\langle \nabla f_n(y_n), y_n - p\right\rangle \leq -2\tau\varphi_n\lambda_n\left\|\left(\mathrm{Id} - P_{Q_n}\right)Ay_n\right\|^2. \tag{3.59}$$

Using $z_n \in C_n$ and the definition of y_n, one has

$$\langle w_n - \beta\lambda_n\nabla f_n(w_n) - y_n, z_n - y_n\rangle \leq 0.$$

This shows that

$$\langle w_n - y_n - \beta\lambda_n(\nabla f_n(w_n) - \nabla f_n(y_n)), z_n - y_n\rangle \leq \beta\lambda_n\langle \nabla f_n(y_n), z_n - y_n\rangle. \tag{3.60}$$

It follows from (3.60) that

$$\begin{aligned}-2\tau\varphi_n\lambda_n\langle \nabla f_n(y_n), z_n - y_n\rangle &\leq -2\frac{\tau}{\beta}\varphi_n\langle d_n, z_n - y_n\rangle \\ &= -2\frac{\tau}{\beta}\varphi_n\langle d_n, w_n - y_n\rangle + 2\frac{\tau}{\beta}\varphi_n\langle d_n, w_n - z_n\rangle.\end{aligned} \tag{3.61}$$

According to the formula $2ab = a^2 + b^2 - (a-b)^2$, we have

$$2\frac{\tau}{\beta}\varphi_n\langle d_n, w_n - z_n\rangle = \|w_n - z_n\|^2 + \frac{\tau^2}{\beta^2}\varphi_n^2\|d_n\|^2 - \|w_n - z_n - \frac{\tau}{\beta}\varphi_n d_n\|^2. \tag{3.62}$$

Combining (3.58), (3.59), and (3.61), we obtain

$$\begin{aligned}\|z_n - p\|^2 \leq \|w_n - p\|^2 - \|z_n - w_n\|^2 + 2\frac{\tau}{\beta}\varphi_n\langle d_n, w_n - z_n\rangle \\ - 2\frac{\tau}{\beta}\varphi_n\left(\langle d_n, w_n - y_n\rangle + \beta\lambda_n\left\|\left(\mathrm{Id} - P_{Q_n}\right)Ay_n\right\|^2\right).\end{aligned}$$

This together with (3.62) yields

$$\begin{aligned}\|z_n - p\|^2 &\le \|w_n - p\|^2 + \frac{\tau^2}{\beta^2}\varphi_n^2\|d_n\|^2 - \|w_n - z_n - \frac{\tau}{\beta}\varphi_n d_n\|^2 - 2\frac{\tau}{\beta}\varphi_n^2\|d_n\|^2.\\ &= \|w_n - p\|^2 - \|w_n - z_n - \frac{\tau}{\beta}\varphi_n d_n\|^2 - \frac{\tau}{\beta^2}(2\beta - \tau)\varphi_n^2\|d_n\|^2.\end{aligned}$$

This completes the proof. □

Lemma 3.9 *Let sequence* $\{x_n\}$ *be generated by Algorithm 3.4. Then the even subsequence* $\{x_{2n}\}$ *is bounded and it is Fejér monotone with respect to* Γ. *Let* $p \in \Gamma$. *Then* $\lim_{n\to\infty}\|x_{2n} - p\|$ *exists and* $\lim_{n\to\infty}\|x_{2n} - y_{2n}\| = 0$.

Proof From (3.15) and Lemma 3.8, we obtain

$$\|x_{n+1} - p\|^2 \le \|w_n - p\|^2 - \alpha\|w_n - z_n - \frac{\tau}{\beta}\varphi_n d_n\|^2 - \frac{\alpha\tau}{\beta^2}(2\beta - \tau)\varphi_n^2\|d_n\|^2. \quad (3.63)$$

Letting $n+1 := 2n+1$ in (3.63) (noting that $w_{2n} = x_{2n}$), we have

$$\begin{aligned}\|x_{2n+1} - p\|^2 \le \|x_{2n} - p\|^2 - \alpha\|x_{2n} - z_{2n} - \frac{\tau}{\beta}\varphi_{2n}d_{2n}\|^2\\ - \frac{\alpha\tau}{\beta^2}(2\beta - \tau)\varphi_{2n}^2\|d_{2n}\|^2.\end{aligned} \quad (3.64)$$

Note that $\frac{\alpha\tau}{\beta^2}(2\beta - \tau) > 0$ due to Condition (C6). Combining (3.19), (3.63), and (3.64), we deduce

$$\begin{aligned}&\|x_{2n+2} - p\|^2\\ &\le \|w_{2n+1} - p\|^2\\ &\le \|x_{2n} - p\|^2 + \theta_{2n+1}(1 + \theta_{2n+1})\|x_{2n+1} - x_{2n}\|^2\\ &\quad - \alpha(1 + \theta_{2n+1})\left[\|x_{2n} - z_{2n} - \frac{\tau}{\beta}\varphi_{2n}d_{2n}\|^2 + \frac{\tau}{\beta^2}(2\beta - \tau)\varphi_{2n}^2\|d_{2n}\|^2\right].\end{aligned} \quad (3.65)$$

Since $\alpha \in (0, 1]$, $-1 \le \theta_{2n+1} \le \theta < 0$, and $\beta > \tau/2$, we infer from (3.65) that

$$\|x_{2n+2} - p\| \le \|x_{2n} - p\|, \quad \forall n \ge 1. \quad (3.66)$$

This implies that $\{\|x_{2n} - p\|\}$ and $\{x_{2n}\}$ are bounded. Thus $\lim_{n\to\infty}\|x_{2n} - p\|$ exists, which together with (3.65) yields

$$\lim_{n\to\infty}\|x_{2n+1} - x_{2n}\| = 0,$$

and

$$\lim_{n\to\infty}\varphi_{2n}^2\|d_{2n}\|^2 = 0 \text{ and } \lim_{n\to\infty}\|x_{2n} - z_{2n} - \frac{\tau}{\beta}\varphi_{2n}d_{2n}\| = 0. \quad (3.67)$$

It follows from (3.67) that $\lim_{n\to\infty}\|x_{2n}-z_{2n}\|=0$. Combining (3.46) and (3.67), we obtain $\lim_{n\to\infty}\|x_{2n}-y_{2n}\|=0$, which together with (3.56) and (3.67) yields

$$\lim_{n\to\infty}\left\|\left(\mathrm{Id}-P_{Q_{2n}}\right)Ay_{2n}\right\|=0.$$

This completes the proof of the lemma. □

Theorem 3.4 *Let Assumptions (A1), (A2), (C1), (C2), and (C6) hold and the sequence $\{x_n\}$ be created by Algorithm 3.4. Then $\{x_n\}$ converges weakly to a point in Γ.*

Proof The proof is similar to the analysis of Theorem 3.3 and therefore we omit it. □

Remark 3.6 We have the following comments for Algorithms 3.1–3.4.

- The results obtained in this paper generalize the alternated inertial methods for solving the SFP recently introduced in the literature [9–11] from finite-dimensional Euclidean spaces to infinite-dimensional Hilbert spaces.
- Notice that the proposed Algorithms 3.1 and 3.2 allow the inertial parameter θ_n defined in (3.3) to be greater than or equal to 1, which is not permitted in Algorithm 1 introduced by Shehu and Gibali [9].
- The four convergence theorems obtained in this paper also hold if we replace the step size criterion (3.4) in the suggested algorithms to the Armijo-type step rule used in Algorithm 2.1. The conclusion is easily verified by a similar convergence analysis as in this paper.
- The difference between Algorithm 3.1 and Algorithm 3.2 is that the step sizes used to calculate y_n and z_n are different, which results in a different range of values for the parameter β. Algorithm 3.3 and Algorithm 3.4 are two different projection and contraction type algorithms. They use the same step size and different directions for the calculation of z_n. Preliminary numerical results provided by Cai et al. [38] show that the projection and contraction methods converge twice as fast as the extragradient method [39]. Our numerical experiments in this paper also verify that the projection and contraction type Algorithms 3.3 and 3.4 also converge faster than the extragradient type Algorithms 3.1 and 3.2; see the numerical results in Section 4 for more details.

4 Numerical experiments

In this section, we provide some numerical examples and applications to demonstrate the advantages and efficiency of the proposed four methods compared to the algorithms in [9–11]. All codes were written in MATLAB 2018a and run on a PC with an Intel(R) Core(TM) i5-8250U CPU @ 1.60 GHz 1.80 GHz and 8.00 GB of running memory. It should be emphasized that the MATLAB toolbox[1] for computing projections on special polyhedra written by Beck and Guttman-Beck [40] turned out to be very useful in our experiments. Next, we apply the proposed four algorithms to treat two real problems, one of which is a signal processing problem and the other is an image restoration problem.

[1] Available on the website https://www.tau.ac.il/~becka/home

Example 4.1 The first problem is concerned with recovering a sparse signal, and it is described as follows:

$$\mathbf{b} = \mathbf{A}\mathbf{x} + \mathbf{e}, \tag{4.1}$$

where $\mathbf{A}$ is an observation matrix of size $m \times k$ ($m < k$), $\mathbf{x} \in \mathbb{R}^k$ is the original clean signal, $\mathbf{e} \in \mathbb{R}^m$ is the noise vector, and $\mathbf{b} \in \mathbb{R}^m$ is the captured noise signal. In this example, we want to recover the sparse signal $\mathbf{x}$ with K ($K \ll m$) non-zero elements from problem (4.1). Let us state the well-known basis pursuit denoising model (also known in statistics as the LASSO model) to find the sparse solution of problem (4.1), which is modeled as shown in the following unconstrained optimization problem

$$\min_{\mathbf{x}} \left(\frac{1}{2} \|\mathbf{b} - \mathbf{A}\mathbf{x}\|_2^2 + \iota \|\mathbf{x}\|_1 \right), \tag{4.2}$$

where ι is a regularization parameter used to control the reconstruction fidelity and sparsity. One can find that the problem (4.2) can be converted to the following constrained optimization problem

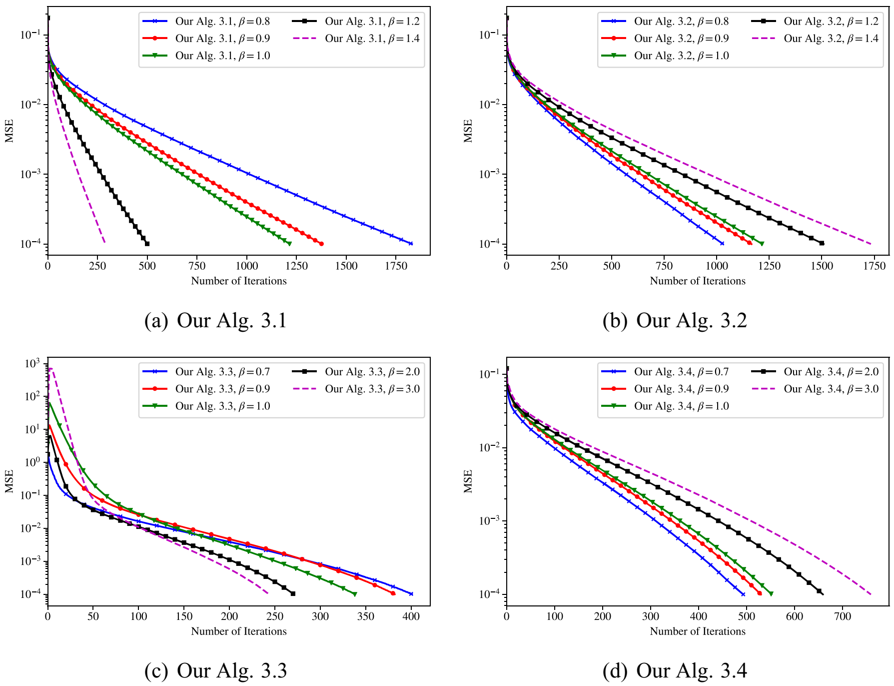
$$\min_{\mathbf{x}} \frac{1}{2} \|\mathbf{b} - \mathbf{A}\mathbf{x}\|_2^2 \quad \text{subject to} \quad \|\mathbf{x}\|_1 \le r, \tag{4.3}$$

which the variable r is related to the regularization parameter ι in unconstrained problem (4.2). Now we can transform the constrained model (4.3) to the split feasibility problem (1.1) by setting $C = \{\mathbf{x} \in \mathbb{R}^k \mid \|\mathbf{x}\|_1 \le r\}$ and $Q = \{\mathbf{b}\}$.

For the experiments, the real clean sparse signal $\mathbf{x} \in \mathbb{R}^k$ has non-zero elements at K random positions whose elements are randomly generated in $\{-1, 1\}$ and maintains the elements at the remaining positions as zero. The matrix $\mathbf{A}$ is a random standard normal distribution of size $m \times k$. A corrupted noise signal $\mathbf{b}$ is known to be generated by $\mathbf{b} = \mathbf{A}\mathbf{x}$ (assuming no noise). Next we use the proposed algorithms as well as some known methods in the literature [9–11] to locate the sparse solution of problem (4.1) due to the l_1 constraint in the set C, and set the following parameters for these algorithms.

- In Algorithms 3.1–3.4, we set $\lambda_1 = 0.3$, $\mu = 0.1$, $\alpha = 1$, $\xi_n = 1 + 10^{-1}/(n+1)^2$, and $\rho_n = 10^{-1}/(n+1)^2$. Set $\theta_n = 0.2$ for Algorithms 3.1–3.3 and select $\theta = -0.2$ for Algorithm 3.4. Choose $\beta = 1.3$ and $\beta = 0.9$ in Algorithm 3.1 and Algorithm 3.2, respectively. Set $\beta = 2.0$ and $\tau = 1.2$ in Algorithm 3.3. Choose $\beta = 0.9$ and $\tau = 1.2$ for Algorithm 3.4.
- In Algorithm 2.1 of Shehu and Gibali [9] (abbreviated as SG Alg. 1), we select $\gamma = 1$, $\ell = 0.5$, $\mu = 0.1$, and $\theta = 0.2$.
- In Algorithm 2.2 of Shehu et al. [10] (abbreviated as SDL Alg. 2), we pick $\theta = 0.2$ and $\chi_n = 2$.
- The Algorithm 2.3 proposed by Dong et al. [11] actually contains two algorithms, and we abbreviate them as DLY Alg. 4.1-I and DLY Alg. 4.1-II, respectively. In DLY Alg. 4.1-I, we choose $\theta = 0.2$, $\mu = 0.1$, $\lambda_1 = 0.3$, and $\tau = 0.2$. In DLY Alg. 4.1-II, we set $\theta = -0.2$, $\mu = 0.1$, $\lambda_1 = 0.3$, and $\tau = 0.2$.

The choice of parameters for the above algorithms is arbitrary and satisfies the prerequisites for their use. Indeed, an in-depth analysis of the parameters should be performed for their optimal performance in practical applications. However, our purpose here is just to demonstrate the numerical performance of these algorithms. Notice that the projection $P_{C_n}(\mathbf{x})$ in these relaxed CQ algorithms can be calculated by a closed formula (see, e.g., [9, Example 4.1] and [21, Section 4]). Now, we can use the proposed algorithms as well as some known methods in the literature [9–11] to solve the signal processing problem (4.1). Without loss of any generality, the Mean Square Error ($\text{MSE} := \|\tilde{\mathbf{x}} - \mathbf{x}\|^2/k$) of the original signal $\mathbf{x} \in \mathbb{R}^k$ and the recovered signal $\tilde{\mathbf{x}} \in \mathbb{R}^k$ less than 10^{-4} is used as their common stopping condition. The iterative process starts with the initial signals $\mathbf{x}_0 = \mathbf{x}_1 = \mathsf{rand}(\mathsf{k}, 1)$ and ends with MSE $< 10^{-4}$. We use the execution time in seconds (denoted by "Time") and the number of iterations (denoted by "Iter.") to evaluate the computational performance of all schemes. Figure 1 shows the numerical behavior of the proposed algorithms with different parameters β. The numerical results for all methods with $m = 256$ and $k = 512$ and with different sparsity K ($K = 10, 20, 30, 40$) are shown in Table 1. Moreover, we draw their MSE trends with the number of iterations under different sparsity K in Fig. 2 (we do not report SDL Alg. 2 because it converges very slowly). In the case of $K = 40$, the original clean signal and the contaminated noise signal are displayed in Fig. 3, and the recovery results of all algorithms are illustrated in Fig. 4.



(a) Our Alg. 3.1

(b) Our Alg. 3.2

(c) Our Alg. 3.3

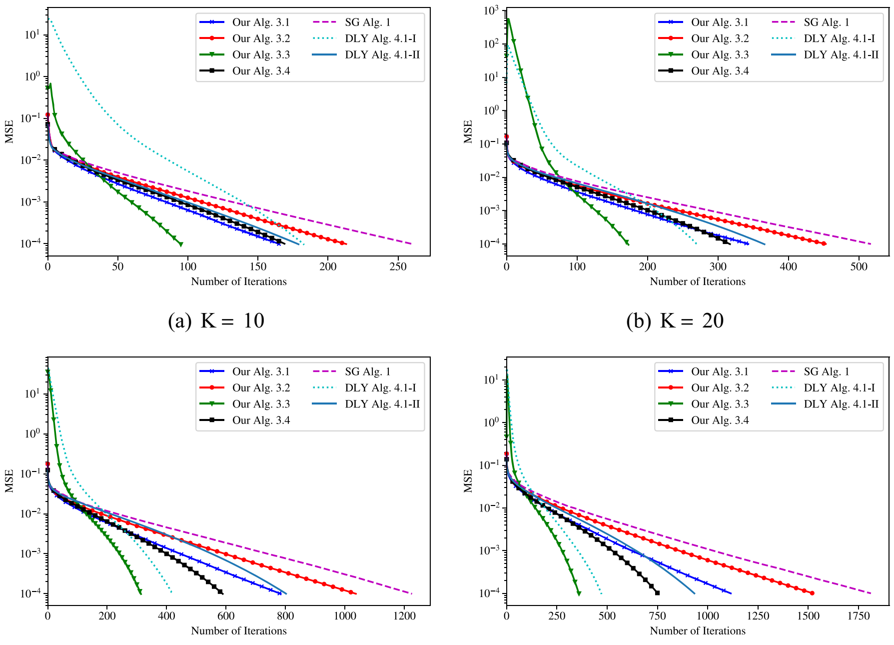
(d) Our Alg. 3.4

Fig. 1 Compare MSE of our Algorithms 3.1–3.4 under different β ($m = 256$, $k = 512$, $K = 30$)

Table 1 Numerical results of all algorithms under different sparsity in Example 4.1

Algorithms	$K = 10$		$K = 20$		$K = 30$		$K = 40$	
	Iter.	Time	Iter.	Time	Iter.	Time	Iter.	Time
Our Alg. 3.1	166	0.11	344	0.22	785	0.52	1116	0.74
Our Alg. 3.2	214	0.14	454	0.30	1038	0.66	1525	1.02
Our Alg. 3.3	96	0.07	174	0.14	315	0.24	361	0.33
Our Alg. 3.4	170	0.13	318	0.25	590	0.48	753	0.61
SG Alg. 1	261	0.59	518	1.25	1227	2.89	1813	4.30
DLY Alg. 4.1-I	184	0.14	271	0.20	421	0.33	473	0.34
DLY Alg. 4.1-II	180	0.15	367	0.29	803	0.63	935	0.71
SDL Alg. 2	7517	6.94	10008	9.06	17853	11.88	29200	25.16

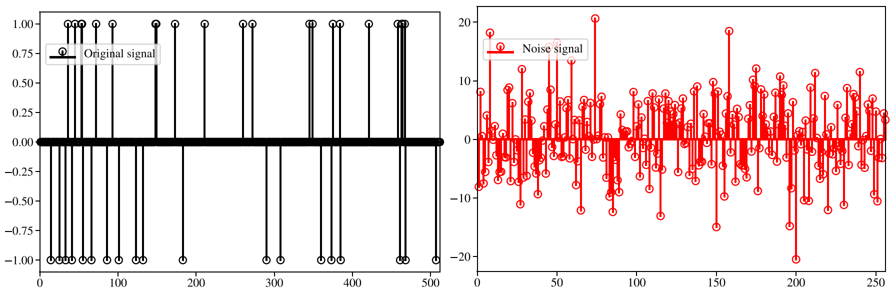
To compare the numerical performance of the algorithms, we use the performance profiles introduced by Dolan and Moré [41]. Let $S = \{s \mid s = 1, 2, 3, \cdots, n_s\}$ denote the set of solvers and $P = \{p \mid p = 1, 2, 3, \cdots, n_p\}$ the set of problems. Assume there is a set of benchmark tests for n_s algorithms solving n_p problems. Recall that the



(a) K = 10 (b) K = 20

(c) K = 30 (d) K = 40

Fig. 2 Compare MSE of all algorithms under different sparsity K in Example 4.1



(a) Original signal (b) Noise signal

Fig. 3 Original signal and noise signal ($K = 40$)

scaled performance profile of the solver s on the problem set P is defined as follows

$$\rho_s(\omega) = \frac{1}{n_p}\text{size}\left\{p \in P \mid \log_2(r_{p,s}) \le \omega\right\},$$

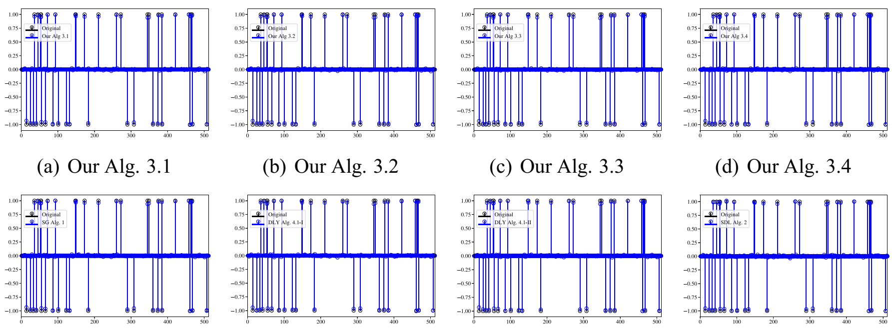
where

$$r_{p,s} = \frac{t_{p,s}}{\min\left\{t_{p,s} \mid s \in S\right\}},$$

and

$$t_{p,s} = \text{computation time (or number of iterations) required by solver } s \\ \in S \text{ to solve problem } p \in P.$$

If algorithm s fails to solve problem p, we define $t_{p,s} = +\infty$. Thus, $\rho_s(\omega)$ is the percentage of problems solved by solver s within a factor of 2^ω of the best solvers. It is easy to know from the definition of $\rho_s(\omega)$ that $\rho_s(0)$ denotes the percentage of problems solved by solver s with maximum efficiency. Furthermore, a larger $\rho_s(\omega)$ when ω is fixed indicates that the solver s can solve more problems, that is, s is "robust"



(a) Our Alg. 3.1 (b) Our Alg. 3.2 (c) Our Alg. 3.3 (d) Our Alg. 3.4

(e) SG Alg. 1 (f) DLY Alg. 4.1-I (g) DLY Alg. 4.1-II (h) SDL Alg. 2

Fig. 4 Comparison of the recovered signal with the original signal for all algorithms ($K = 40$)

if $\rho_s(\omega)$ is large for ω fixed. Next, we fix $m = 512$ and sparsity $K = 30$, and choose $k = \{1024, 1094, 1164, 1234, \cdots, 4034\}$. The performance profiles of the proposed algorithms 3.1–3.4 and the compared methods with respect to computation time and number of iterations are demonstrated in Figs. 5 and 6, respectively.

We have the following observations for the numerical results of Example 4.1.

- It can be visualized from Fig. 1 that the proposed four algorithms have different numerical performances on different parameters β. Specifically, the proposed Algorithms 3.1 and 3.3 have better results when the parameter β is greater than 1, while the proposed Algorithms 3.2 and 3.4 have better performance with the parameter β less than 1.
- The algorithms proposed in this paper for solving the split feasibility problem can tackle the signal denoising problem (as shown in Figs. 3 and 4). On the other hand, the preliminary results presented in Table 1 and Fig. 2 demonstrate the advantages and computational efficiency of the proposed methods over some known schemes. Specifically, the suggested Algorithms 3.1 and 3.2 require fewer iterations and execution time than SG Alg. 1 in reaching the same stopping criterion; the offered Algorithm 3.3 and Algorithm 3.4 perform better than DLY Alg. 4.1-I and DLY Alg. 4.1-II, respectively. Furthermore, it is also noted from Table 1 that SDL Alg. 2 requires more iterations and CPU time than the other algorithms in these experiments when the sparsity K keeps increasing.
- The information in Fig. 5 points out that (1) our Algorithm 3.3 can win 80% in computational time; (2) when $\omega = 1.5$, our Algorithms 3.1–3.4 can solve all problems, DLY Alg. 4.1-I can solve over 95% of problems, DLY Alg. 4.1-II can solve about 85% of problems, while SG Alg. 1 cannot solve any problems; (3) SG Alg. 1 with Armijo line search type step sizes takes more time to reach the stopping criterion than the other algorithms.
- According to Fig. 6, it can be seen that our Algorithm 3.3 can solve all the problems in this test with a minimum number of iterations. Secondly, our Algorithms 3.1 and

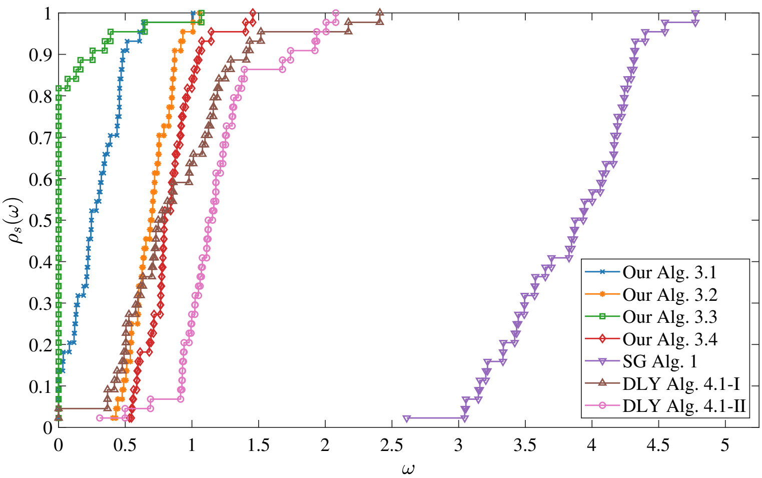


Fig. 5 Performance profiles of all algorithms based on computation time for Example 4.1

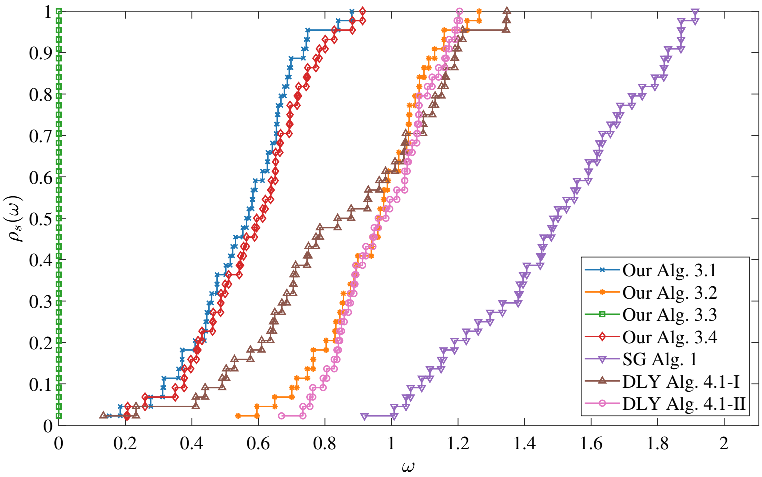


Fig. 6 Performance profiles of all algorithms based on number of iterations for Example 4.1

3.4 perform better than the other ones in terms of the number of iterations (see the value of $\rho_s(\omega)$ when $\omega = 0.8$). Notice that our Algorithm 3.2 and DLY Alg. 4.1-II overtake DLY Alg. 4.1-I in terms of the stability of the number of iterations when $\omega \geq 1.1$. Furthermore, SG Alg. 1 requires more iterations to achieve the required error accuracy.

Example 4.2 In this example we consider the following image restoration problem:

$$\mathbf{A}\mathbf{x} = \mathbf{b} + \mathbf{v}, \tag{4.4}$$

where $\mathbf{A} \in \mathbb{R}^{m\times k}$, $\mathbf{x} \in \mathbb{R}^k$, $\mathbf{b} \in \mathbb{R}^m$, and $\mathbf{v} \in \mathbb{R}^m$ represent the convolution matrix, the original image, the degraded image, and the noise vector, respectively. The problem (4.4) can be converted into the following constrained optimization model

$$\min_{\mathbf{x}\in C} f(\mathbf{x}) := \|\mathbf{A}\mathbf{x} - \mathbf{b}\|^2 .$$

We can put the problem (4.4) into the model of SFP (1.1) by setting C is a box area in $\mathbb{R}^k$ and $Q = \{\mathbf{y} \in \mathbb{R}^m \mid \|\mathbf{y} - (\mathbf{b} + \mathbf{v})\| \leq "\}$ for small enough $" > 0$. In the special case where no noise is added to the degraded image (i.e., $\mathbf{v} = \mathbf{0}$) we set $Q = \{\mathbf{b}\}$ in the SFP (1.1).

In this example we first test two different images[1,2] of size 512×512 whose element values were scaled to between 0 and 1 (i.e., $C := [0, 1]^m$). The tested images are first contaminated by a 9×9 Gaussian random blur[2,3] with a standard deviation of 2 and additionally corrupted by a random Gaussian white noise with zero-mean and standard

[2] Download from the website http://www.imageprocessingplace.com/downloads_V3/root_downloads/image_databases/standard_test_images.zip

[3] Access via the website http://www.imm.dtu.dk/~pcha/HNO/HNO.zip

deviation of 10^{-4}. The two tested clean images and their degraded images are shown in Fig. 7.

We use the proposed four algorithms and some known ones in the literature [9–11] to solve problem (4.4) and keep the parameters of all algorithms the same as those set in Example 4.1. Three metrics, the signal-to-noise ratio (SNR), the peak-signal-to-noise ratio (PSNR), and the structural similarity index measure (SSIM), are used to objectively describe the reconstruction quality of the recovered image $\tilde{\mathbf{x}} \in \mathbb{R}^{m\times k}$ compared to the original image $\mathbf{x} \in \mathbb{R}^{m\times k}$, which are defined as follows.

- The SNR in decibel (dB) is calculated by

$$\mathrm{SNR} := 20 \log_{10} \frac{\|\mathbf{x}\|}{\|\tilde{\mathbf{x}} - \mathbf{x}\|}.$$

  From the definition of SNR, it can be seen that a larger value of SNR indicates a better reconstruction quality.
- The PSNR in decibel (dB) is defined by

$$\mathrm{PSNR} := 10 \log_{10} \frac{\mathrm{MAX}_{\mathbf{x}}^2}{\frac{1}{mk}\sum_{i=0}^{m}\sum_{j=0}^{k}[\mathbf{x}(i,j) - \tilde{\mathbf{x}}(i,j)]^2},$$

  where $\mathrm{MAX}_{\mathbf{x}}$ means the maximum possible pixel value of image $\mathbf{x} \in \mathbb{R}^{m\times k}$. A larger PSNR shows a better quality of recovery.
- The SSIM is a metric proposed by Wang et al. [42] to measure the similarity of two images, and its definition can be found in [42]. The value of SSIM lies between 0 and 1, and its larger value indicates that the two images are more similar.

The iterative procedure of the algorithms presented in this paper and the alternated inertial schemes in the literature [9–11] all begin from the degraded image $\mathbf{b}$ (i.e., initial points $\mathbf{x}_0 = \mathbf{x}_1 = \mathbf{b}$). Without loss of generality, we set the maximum number of iterations 100 as the stopping criterion for all algorithms in this experiment. The recovery results of all algorithms under image "lena" and image "pirate" are shown in Figs. 8 and 9, respectively. In addition, the numerical performance of all algorithms in terms of running time in seconds (denoted by "Time"), SNR, PSNR, and SSIM under two images is shown in Table 2. Furthermore, we plot the PSNR and the SSIM variation curves of all algorithms under two test images in Figs. 10 and 11, respectively.

(a) Original "lena" (b) Degraded "lena" (c) Original "pirate" (d) Degraded "pirate"

Fig. 7 The original and degraded images in Example 4.2

Fig. 8 Recovery results for all algorithms in the image "lena"

It can be seen from Figs. 7, 8, and 9 that the four algorithms proposed in this paper can be used to solve the image restoration problem (4.4). According to Table 2, our Algorithm 3.3 has the largest SNR, PSNR, and SSIM in the recovered results of both images. However, our Algorithms 3.1 and 3.2 do not perform very well. On the other hand, SDL Alg. 2 takes the least computation time to reach the stopping criterion than the other algorithms. Furthermore, the difference in computation time required by our Algorithms 3.1–3.4 and SG Alg. 1, DLY Alg. 4.1-I, and DLY Alg. 4.1-II is

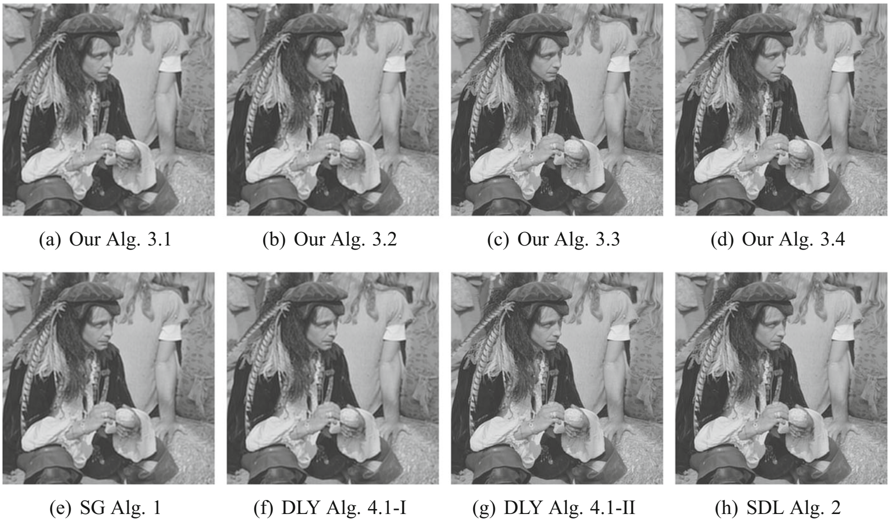


Fig. 9 Recovery results for all algorithms in the image "pirate"

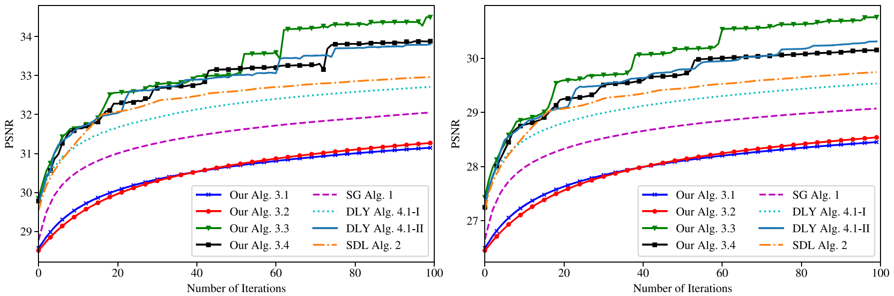
Table 2 Numerical results for all algorithms under different images

Algorithms	Image "lena"				Image "pirate"			
	Time	SNR	PSNR	SSIM	Time	SNR	PSNR	SSIM
Our Alg. 3.1	22.93	25.84	31.15	0.8762	22.68	22.49	28.45	0.8131
Our Alg. 3.2	22.61	25.96	31.27	0.8784	22.40	22.58	28.54	0.8164
Our Alg. 3.3	22.90	29.13	34.50	0.9221	23.03	24.80	30.76	0.8843
Our Alg. 3.4	23.03	28.54	33.88	0.9163	23.08	24.20	30.15	0.8691
SG Alg. 1	22.34	26.72	32.05	0.8910	22.41	23.11	29.07	0.8357
DLY Alg. 4.1-I	22.74	27.34	32.71	0.9006	22.93	23.58	29.53	0.8511
DLY Alg. 4.1-II	23.04	28.46	33.81	0.9157	23.12	24.35	30.31	0.8735
SDL Alg. 2	14.10	27.60	32.97	0.9045	14.13	23.79	29.74	0.8576

not significant. From the results in Figs. 10 and 11, it appears that the convergence of our Algorithms 3.3 and 3.4 is not monotonic, which may be related to the choice of inertial and step size.

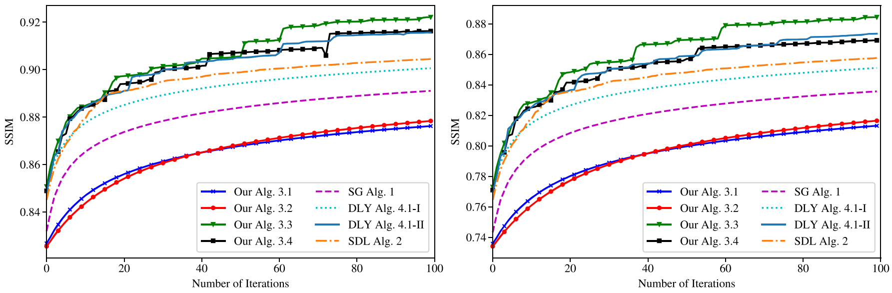
To further test the computational efficiency and stability of our Algorithms 3.1–3.4, we choose 49 standard 512 × 512 greyscale test images from website https://ccia.ugr.es/cvg/CG/base.htm; see Fig. 12. The numerical results for our Algorithms 3.1–3.4 and the comparison methods are shown in Table 3.

The "Num. of wins" in Table 3 represents the number of wins that the algorithm achieved for the best recovery out of the 49 test images. It can be seen from Table 3 that the number of wins for our Algorithms 3.3 and 3.4 is almost equal to the number of wins for DLY Alg. 4.1-I and DLY Alg. 4.1-II. This illustrates the competitive advantage of our Algorithms 3.3 and 3.4 in some tests. Furthermore, our Algorithm 3.4 performs better than our Algorithm 3.3 because Algorithm 3.4 achieves the best recovery more times; our Algorithm 3.4 also outperforms our Algorithm 3.2, which shows that projection and contraction type algorithms perform better than extragradient type algorithms. In conclusion, the four algorithms introduced in this paper provide some viable solutions to image restoration problems and they have some competitive advantages over the algorithms in [9–11] to a certain extent.



(a) Image "lena" (b) Image "pirate"

Fig. 10 The PSNR of all algorithms under different images in Example 4.2



(a) Image “lena” (b) Image “pirate”

Fig. 11 The SSIM of all algorithms under different images in Example 4.2

Fig. 12 Dataset of standard 512 $\times$ 512 grayscale test images

Table 3 The numerical results of all algorithms under 49 standard test images

Algorithms	SNR		PSNR		SSIM	
	Num. of wins	Percent	Num. of wins	Percent	Num. of wins	Percent
Our Alg. 3.1	0	0.00%	0	0.00%	0	0.00%
Our Alg. 3.2	0	0.00%	0	0.00%	0	0.00%
Our Alg. 3.3	7	14.29%	6	12.24%	8	16.33%
Our Alg. 3.4	18	36.73%	17	34.69%	18	36.73%
SG Alg. 1	0	0.00%	0	0.00%	0	0.00%
DLY Alg. 4.1-I	0	0.00%	2	4.08%	0	0.00%
DLY Alg. 4.1-II	24	48.98%	24	48.98%	23	46.94%
SDL Alg. 2	0	0.00%	0	0.00%	0	0.00%

5 Conclusions

In this paper, based on the relaxed CQ method, alternating inertial method, extragradient method, and projection contraction method, we proposed four new adaptive iterative algorithms to discover solutions of split feasibility problems in infinite-dimensional Hilbert spaces. The step sizes of the proposed algorithms are automatically updated by using some previously known information. Thus they can work well without the prior knowledge of the operator norm of the involved operator. Moreover, the proposed algorithms employ two different step sizes in each iteration, which performs better than the algorithms that use the same step size in each iteration as verified by the preliminary example provided in this paper. Under some suitable conditions, we established the weak convergence theorems of the suggested algorithms and obtain the Fejér monotonicity of the even subsequence with respect to the solution set. Finally, the advantages of the proposed algorithms are confirmed by two numerical applications which include signal denoising and image deblurring. The algorithms obtained in this paper improve and generalize many known ones in the literature. It is interesting to extend the results obtained in this paper to Banach spaces or Hadamard manifolds.

Acknowledgements The authors would like to thank the handling editor and the anonymous referee for careful reading and constructive comments, which improved the quality of the manuscript.

Author contribution All authors have equal contributions.

Funding Bing Tan was supported by the China Scholarship Council (CSC No. 202106070094). Xianfu Wang was partially supported by the Natural Sciences and Engineering Research Council of Canada.

Data Availability The datasets generated during and/or analyzed during the current study are available from the corresponding author.

Declarations

Ethical approval Not applicable.

Conflict of interest The authors declare no competing interests.

References

1. Censor, Y., Elfving, T.: A multiprojection algorithm using Bregman projections in a product space. Numer. Algorithms **8**, 221–239 (1994)
2. Xu, H.K.: Iterative methods for the split feasibility problem in infinite-dimensional Hilbert spaces. Inverse Problems **26**, article no. 105018 (2010)
3. Bauschke, H.H., Borwein, J.M.: On projection algorithms for solving convex feasibility problems. SIAM Rev. **38**, 367–426 (1996)
4. Byrne, C.: Iterative oblique projection onto convex sets and the split feasibility problem. Inverse Problems **18**, 441–453 (2002)
5. Byrne, C.: A unified treatment of some iterative algorithms in signal processing and image reconstruction. Inverse Problems **20**, 103–120 (2004)
6. Censor, Y., Bortfeld, T., Martin, B., Trofimov, A.: A unified approach for inversion problems in intensity-modulated radiation therapy. Phys. Med. Biol. **51**, 2353–2365 (2006)
7. Gibali, A., Sabach, S., Voldman, S.: Non-convex split feasibility problems: models, algorithms and theory. Open J. Math. Optim. **1**, article no. 1 (2020)
8. Brooke, M., Censor, Y., Gibali, A.: 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**, 181–205 (2023)
9. Shehu, Y., Gibali, A.: New inertial relaxed method for solving split feasibilities. Optim. Lett. **15**, 2109–2126 (2021)
10. Shehu, Y., Dong, Q.L., Liu, L.L.: Global and linear convergence of alternated inertial methods for split feasibility problems. Rev. R. Acad. Cienc. Exactas Fís. Nat. Ser. A Math. RACSAM **115**, article no. 53 (2021)
11. Dong, Q.L., Liu, L.L., Yao, Y.H.: Self-adaptive projection and contraction methods with alternated inertial terms for solving the split feasibility problem. J. Nonlinear Convex Anal. **23**, 591–605 (2022)
12. Yang, Q.: The relaxed CQ algorithm solving the split feasibility problem. Inverse Problems **20**, 1261–1266 (2004)
13. Qu, B., Xiu, N.: A note on the CQ algorithm for the split feasibility problem. Inverse Problems **21**, 1655–1665 (2005)
14. Xu, H.K.: A variable Krasnosel'skiǐ-Mann algorithm and the multiple-set split feasibility problem. Inverse Problems **22**, 2021–2034 (2006)
15. López, G., Martín-Márquez, V., Wang, F., Xu, H.K.: Solving the split feasibility problem without prior knowledge of matrix norms. Inverse Problems **28**, article no. 085004 (2012)
16. Reich, S., Tuyen, T.M.: A new approach to solving split equality problems in Hilbert spaces. Optimization **71**, 4423–4445 (2022)
17. Bauschke, H.H., Combettes, P.L.: Convex Analysis and Monotone Operator Theory in Hilbert Spaces, 2nd edn. Springer, Berlin (2017)
18. Hendrickx, J.M., Olshevsky, A.: Matrix p-norms are NP-hard to approximate if $p \neq 1, 2, \infty$. SIAM J. Matrix Anal. Appl. **31**, 2802–2812 (2010)
19. Zhang, W.X., Han, D.R., Li Z.: A self-adaptive projection method for solving the multiple-sets split feasibility problem. Inverse problems **25**, article no. 115001 (2009)
20. Dong, Q.L., Tang, Y.C., Cho, Y.J., Rassias, T.M: "Optimal" choice of the step length of the projection and contraction methods for solving the split feasibility problem. J. Global Optim. **71**, 341–360 (2018)
21. Gibali, A., Liu, L.W., Tang, Y.C.: Note on the modified relaxation CQ algorithm for the split feasibility problem. Optim. Lett. **12**, 817–830 (2018)
22. Gibali, A., Mai, D.T., Vinh, N.T.: A new relaxed CQ algorithm for solving split feasibility problems in Hilbert spaces and its applications. J. Ind. Manag. Optim. **15**, 963–984 (2019)
23. Sahu, D.R., Cho, Y.J., Dong, Q.L., Kashyap, M.R., Li, X.H.: Inertial relaxed CQ algorithms for solving a split feasibility problem in Hilbert spaces. Numer. Algorithms **87**, 1075–1095 (2021)
24. Ma, X., Liu, H., Li, X.: Two optimization approaches for solving split variational inclusion problems with applications. J. Sci. Comput. **91**, article no. 58 (2022)
25. Polyak, B.T.: Some methods of speeding up the convergence of iteration methods. U.S.S.R. Comput. Math. Math. Phys. **4**, 1–17 (1964)
26. Nesterov, Y.: A method for solving the convex programming problem with convergence rate $O\left(1/k^2\right)$. Dokl. Akad. Nauk SSSR **269**, 543–547 (1983)

27. Beck, A., Teboulle, M.: A fast iterative shrinkage-thresholding algorithm for linear inverse problems. SIAM J. Imaging Sci. **2**, 183–202 (2009)
28. Yang, J., Liu, H.: Strong convergence result for solving monotone variational inequalities in Hilbert space. Numer. Algorithms **80**, 741–752 (2019)
29. Tan, B., Cho, S.Y.: Inertial extragradient algorithms with non-monotone stepsizes for pseudomonotone variational inequalities and applications. Comput. Appl. Math. **41**, article no. 121 (2022)
30. Mu, Z., Peng, Y.: A note on the inertial proximal point method. Stat. Optim. Inf. Comput. **3**, 241–248 (2015)
31. Iutzeler, F., Malick, J.: On the proximal gradient algorithm with alternated inertia. J. Optim. Theory Appl. **176**, 688–710 (2018)
32. Iutzeler, F., Hendrickx, J.M.: A generic online acceleration scheme for optimization algorithms via relaxation and inertia. Optim. Methods Softw. **34**, 383–405 (2019)
33. Shehu, Y., Iyiola, O.S.: Projection methods with alternating inertial steps for variational inequalities: weak and linear convergence. Appl. Numer. Math. **157**, 315–337 (2020)
34. Shehu, Y., Dong, Q.L., Liu, L.L.: Fast alternated inertial projection algorithms for pseudo-monotone variational inequalities. J. Comput. Appl. Math. **415**, article no. 114517 (2022)
35. Beck, A.: First-Order Methods in Optimization. MOS-SIAM Series on Optimization, vol. **25**. Society for Industrial and Applied Mathematics (SIAM), Philadelphia (2017)
36. Opial, Z.: Weak convergence of the sequence of successive approximations for nonexpansive mappings. Bull. Amer. Math. Soc. **73**, 591–597 (1967)
37. Osilike, M.O., Aniagbosor, S.C.: Weak and strong convergence theorems for fixed points of asymptotically nonexpansive mappings. Math. Comput. Model. **32**, 1181–1191 (2000)
38. Cai, X., Gu, G., He, B.S.: On the $O(1/t)$ convergence rate of the projection and contraction methods for variational inequalities with Lipschitz continuous monotone operators. Comput. Optim. Appl. **57**, 339–363 (2014)
39. Korpelevich, G.M.: The extragradient method for finding saddle points and other problems. Èkonom. i Mat. Metody **12**, 747–756 (1976)
40. Beck, A., Guttmann-Beck, N.: FOM-a MATLAB toolbox of first-order methods for solving convex optimization problems. Optim. Methods Softw. **34**, 172–193 (2019)
41. Dolan, E.D., Moré, J.J.: Benchmarking optimization software with performance profiles. Math. Program. **91**, 201–213 (2002)
42. Wang, Z., Bovik, A.C., Sheikh, H.R., Simoncelli, E.P.: Image quality assessment: from error visibility to structural similarity. IEEE Trans. Image Process. **13**, 600–612 (2004)