Yokohama Publishers
ISSN 1349-8169 ONLINE JOURNAL
Pacific Journal of Optimization
Vol. 21, No. 2, pp. 269–283, 2025

EXTENDED CQ ALGORITHM INTEGRATED WITH SELECTION TECHNIQUE FOR MULTIPLE-SETS SPLIT FEASIBILITY PROBLEM

Huan Gao, Zhibao Li* and Guangming Zhou

Abstract: In this paper, we propose an extended CQ algorithm integrated with selection technique to address the multiple-sets split feasibility problem (MSFP). At each iteration, the selection technique is employed to formulate a split feasibility subproblem for MSFP, subsequently it is resolved by means of the CQ algorithm. Under mild conditions, we establish the global convergence results for the extended CQ algorithm. Furthermore, we provide empirical evidence in the form of numerical results, which conclusively affirm the effectiveness and competitiveness of our proposed algorithm.

Key words: *multiple-sets split feasibility problem, extended CQ algorithm, selection technique*

Mathematics Subject Classification: *47J20, 47J25, 47N10, 65J15*

---

1 Introduction

We consider the multiple-sets split feasibility problem (MSFP) of finding a point x such that

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

where $C_i \subset \mathbb{R}^n$, $i = 1, 2, \ldots, s$, and $Q_j \subset \mathbb{R}^m$, $j = 1, 2, \ldots, t$, are nonempty closed convex sets, and $A \in \mathbb{R}^{m\times n}$ is a given matrix. MSFP was first investigated by Censor and Elfving [3], and it is a kind of inverse problem that arises from phase retrievals in image reconstruction [1], intensity-modulated radiation therapy [4, 5, 6, 7]. MSFP reduces to a split feasibility problem (SFP) when $s = t = 1$, which is to find a point x such that

$$x \in C \quad \text{such that} \quad Ax \in Q, \tag{1.2}$$

where $C \subset \mathbb{R}^n$ and $Q \subset \mathbb{R}^m$ are two closed convex sets.

*Corresponding author. This work is supported by the National Natural Science Foundation of China Grants (12271526, 12471295, 12171145 and 12071399), the Natural Science Foundation of Hunan Province (2022JJ30675), the Training Program for Excellent Young Innovators of Changsha (kq2209021) and the Natural Science Foundation of Changsha (kq2202068). The authors also grateful for technical support from the High Performance Computing Center of Central South University, and take this opportunity to thank the anonymous referees for their patient and valuable comments, which improved the quality of this paper greatly.

 DOI: https:// doi.org/10.61208/pjo-2024-003

Over the past few decades, there has been a prolific development of numerical methods aimed at solving the SFP and its multiple-sets variant (MSFP). Byrne firstly developed the CQ algorithm to solve the SFP, successfully applying it within signal processing and image reconstruction [1, 2]. Subsequently, a group of enhanced and relaxed CQ algorithms were developed for SFP [24, 17, 11, 21, 10], as well as the comprehensive analyses of their convergence [8, 16].

Concurrently, building upon the fundamental work of Censor *et al.* in solving MSFP [4], several projection methods have been designed to address this kind of problem [29, 30, 19], along with the development of self-adaptive projection techniques for MSFP [26, 28, 22]. Additionally, a lot of effective and efficient methods have been investigated to solve MSFPs in the last two decades. For instance, the variable Krasnosel'skiĭ-Mann algorithm by Xu [23], the efficient simultaneous for the constrained MSFP by Zhang *et al.* [27], the self-adaptive CQ algorithm by He *et al.* [14], the successive projection algorithm by Qu and Chang [18], the relaxed CQ algorithm involving the inertial technique by Suantai *et al.* [20], and so on. Expanding beyond MSFP, self-adaptive projection methods have also been developed to solve the nonlinear MSFP [15, 13].

Throughout this paper, we assume that the solution set of MSFP (1.1) is nonempty. By defining a merit function

$$q(x) = \frac{1}{2}\sum_{i=1}^{s}\alpha_i \left\|x - P_{C_i}(x)\right\|^2 + \frac{1}{2}\sum_{j=1}^{t}\beta_j \left\|Ax - P_{Q_j}(Ax)\right\|^2, \tag{1.3}$$

with

$$\sum_{i=1}^{s}\alpha_i + \sum_{j=1}^{t}\beta_j = 1, \quad \alpha_i > 0,\ \beta_j > 0,$$

we can see that the MSFP (1.1) is equivalent to the minimization problem

$$\min_{x\in\Omega}\ q(x), \tag{1.4}$$

where Ω is an auxiliary closed convex set. It is known that the merit function $q(x)$ is continuously differentiable, and its gradient is given by

$$\nabla q(x) = \sum_{i=1}^{s}\alpha_i\left(x - P_{C_i}(x)\right) + \sum_{j=1}^{t}\beta_j A^T\left(Ax - P_{Q_j}(Ax)\right).$$

Consequently, a projection algorithm can be established to solve the optimization problem (1.4) with the following procedure, *i.e.*,

$$\begin{aligned} x_{k+1} &= P_\Omega\left(x_k - \gamma\nabla q(x)\right) \\ &= P_\Omega\left(x_k - \gamma\left(\sum_{i=1}^{s}\alpha_i\left(x - P_{C_i}(x_k)\right) + \sum_{j=1}^{t}\beta_j A^T\left(Ax_k - P_{Q_j}(Ax_k)\right)\right)\right) \end{aligned} \tag{1.5}$$

for proper selection of $\gamma > 0$. It has been proved that the convergence of algorithm (1.5) is attainable, provided that the condition $0 < \gamma < 2/L_0$ is met, where L_0 represents the Lipschitz constant of ∇q. In other words,

$$L_0 = \sum_{i=1}^{s}\alpha_i + L\sum_{j=1}^{t}\beta_j,$$

where $L = \rho(A^T A)$ denotes the spectral radius of $A^T A$. This is essentially the projection method proposed by Censor *et al.* [4] for solving the MSFP.

Considering the computational aspects involved in the projection step outlined in (1.5), it's noteworthy that $s+t$ projection operations are executed to compute the gradient ∇q at each iteration. As a consequence of this, the iterative sequences $\{x_k\}$ and $\{Ax_k\}$ follow a trajectory converging towards a solution pair (x^*, Ax^*) of (1.1) along what we can refer to as the "straight path", as depicted by the red star points in Figure 1. However, as the sequences approach the solution, the degree of improvement becomes limited. This phenomenon can likely be attributed to the cumulative influence of all the terms in the objective function $q(x)$, which involve $\|x - P_{C_i}(x)\|, i = 1, 2, \ldots, s$, and $\|Ax - P_{Q_j}(Ax)\|, j = 1, 2, \ldots, t$.

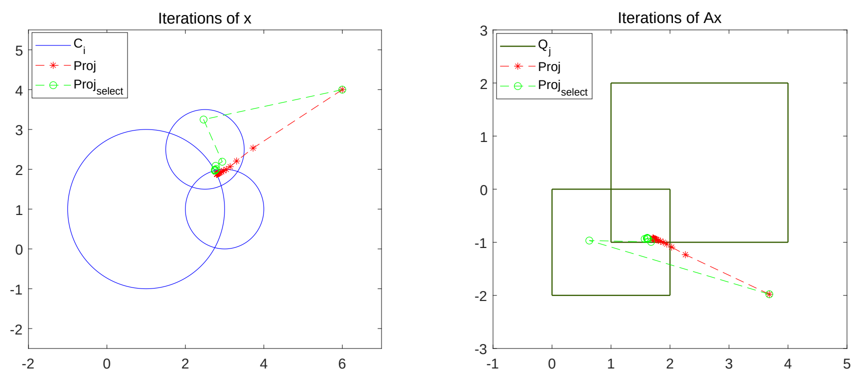

Figure 1: Convergent trajectories of iterative sequences by projection method and projection method integrated with selection technique for MSFP.

Recently, based on the insight captured from the fact that x solves (1.1) if and only if it is the zero point of

$$\bar{q}(x) = \frac{1}{2}\left\|x - P_{C_{i_{\max}}}(x)\right\|^2 + \frac{1}{2}\left\|Ax_k - P_{Q_{j_{\max}}}(Ax_k)\right\|^2, \tag{1.6}$$

where

$$i_{\max} \in \left\{ i \,\middle|\, \max_{1\le i\le s} \|x - P_{C_i}(x)\| \right\}, \quad j_{\max} \in \left\{ j \,\middle|\, \max_{1\le j\le t} \left\|Ax - P_{Q_j}(Ax)\right\| \right\}, \tag{1.7}$$

Yao *et al.* [25] introduced a novel selection technique aimed at enhancing the projection method's efficiency for solving the MSFP (1.1). At each iteration, the selection technique identifies two sets, namely, $C_{i_{\max}}$ and $Q_{j_{\max}}$, which have the greatest distances from the current points x_k and Ax_k, respectively. Subsequently, it formulates a simplified optimization problem with the objective function defined in (1.6). The new candidate point, denoted as x_{k+1}, can be obtained by solving this reduced optimization problem. The iteration sequences, depicted as green circles in Figure 1, converge more quickly, which implies that the selection technique results in a faster rate of convergence compared to the standard projection method. Whereas it's important to note that this method still encounters challenges in achieving rapid convergence as the sequences $\{x_k\}$ and $\{Ax_k\}$ approach the solution points x^* and Ax^*.

It's worth reiterating that the selection technique represents a constructive strategy for transforming the MSFP into a set of SFPs, a transformation that undeniably simplifies the original optimization challenge. Furthermore, existing empirical evidence in the literature underscores the effectiveness and efficiency of the CQ algorithm when employed to address SFPs, as illustrated in the example shown in Figure 2. Consequently, the utilization of the CQ algorithm to solve the SFP subproblem at each iteration is a valid and justified approach.

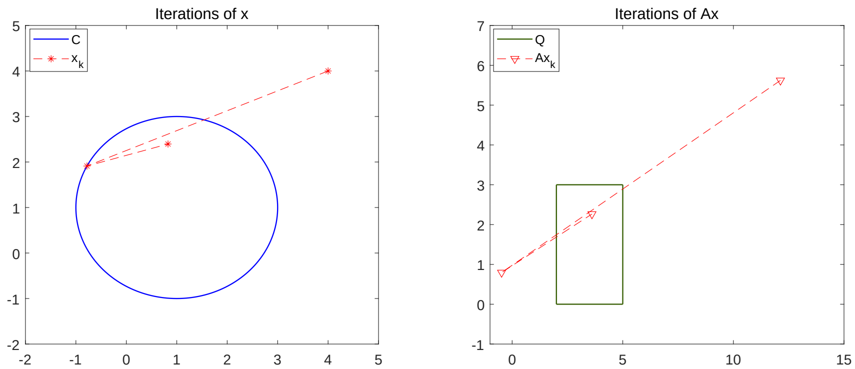

Figure 2: Convergent trajectory of iterative sequence by CQ algorithm for SFP.

In this paper, we propose an extended CQ algorithm integrated with a selection technique to solve the MSFP. The core of our approach lies in the construction of a SFP within the framework of the MSFP at each iteration through the application of the selection technique. Subsequently, we utilize the CQ algorithm to solve the resultant SFP, presenting the corresponding convergence results. To substantiate the practical efficiency of our extended CQ algorithm, we create a series of illustrative examples. Our experimental results firmly establish the advantages of the proposed method, evident from both a reduced number of iterations required and decreased computational costs.

The rest of this paper is organized as follows. Section 2 provides an intricate exposition of the extended CQ algorithm, augmented by the selection technique for MSFP. In Section 3, we establish the convergence results for the proposed extended CQ algorithm. To corroborate the performance of our proposed algorithm, Section 4 offers a set of illustrative examples. Finally, we conclude this paper with some concluding remarks in Section 5.

2 The Extended CQ Algorithm

In this section, we begin by revisiting fundamental properties of the projection operator, as previously discussed. Following this, we delve into the presentation of the extended CQ algorithm, integrated with the selection technique for the solving of MSFP.

Let $\mathbb{R}^n$ be an n-dimensional Euclidean space, $\langle \cdot, \cdot \rangle$ be the inner product in $\mathbb{R}^n$, and $\|x\| = \sqrt{\langle x, x \rangle}$ be the ℓ_2-norm in $\mathbb{R}^n$. In the following, we first recall some fundamental properties of the projection operator from $\mathbb{R}^n$ onto a set $\Omega \subset \mathbb{R}^n$ defined by

$$P_\Omega(x) = \arg\min_{z \in \Omega} \|z - x\| \tag{2.1}$$

in summary.

Lemma 2.1. *Let $\Omega \subset \mathbb{R}^n$ be a nonempty closed convex set, for any $x, y \in \mathbb{R}^n$ and $z \in \Omega$, the following properties hold:*

(1) $x \in \Omega \Leftrightarrow P_\Omega(x) = x$,

(2) $\langle x - P_\Omega(x), z - P_\Omega(x)\rangle \leq 0$,

(3) $\|P_\Omega(x) - P_\Omega(y)\|^2 \leq \langle x - y, P_\Omega(x) - P_\Omega(y)\rangle$,

(4) $\|P_\Omega(x) - z\|^2 \leq \|x - z\|^2 - \|P_\Omega(x) - x\|^2$,

(5) $\|P_\Omega(x) - P_\Omega(y)\| \leq \|x - y\|$.

Proof. See proof by Facchinei and Pang in [12]. □

Lemma 2.2. *For any nonempty closed convex set $\Omega \subset \mathbb{R}^n$, the inequality (5) in Lemma 2.1 holds with equality only if the following condition holds, i.e.,*

$$\|P_\Omega(x) - x\| = \|P_\Omega(y) - y\|.$$

Proof. See proof by Cheney and Goldstein in [9]. □

Denote by $C_{i_{\max}}$ and $Q_{j_{\max}}$ the selected sets with greatest distances from x and Ax, respectively, we establish a constrained optimization problem for the MSFP as

$$\min_{x \in C_{i_{\max}}} \; p(x), \tag{2.2}$$

where the merit function is defined as

$$p(x) = \frac{1}{2}\left\|Ax - P_{Q_{j_{\max}}}(Ax)\right\|^2. \tag{2.3}$$

It is noted that the merit function $p(x)$ defined in (2.3) is continuously differentiable [4], and its gradient is given by

$$\nabla p(x) = A^T(I - P_{Q_{j_{\max}}})Ax. \tag{2.4}$$

In addition, the above gradient function $\nabla p(x)$ is Lipschitz continuous with constant L.

Below, we present the extended CQ algorithm for solving MSFP in detail.

Algorithm 1 Extended CQ algorithm for MSFP

Initialization. Given $\varepsilon > 0$, $0 < \gamma < 2/L$, and an initial point x_0, set $k = 0$.

Step 1. Select sets $C_{i_{\max}}$ and $Q_{j_{\max}}$ with greatest distances from x_k and Ax_k, and record the index pair $\{i_k, j_k\}$.

Step 2. Compute

$$\begin{cases} z_k = P_{C_{i_k}}(x_k), \\ y_k = A^T(I - P_{Q_{j_k}})Ax_k. \end{cases} \tag{2.5}$$

Step 3. If

$$\|x_k + y_k - z_k\| < \varepsilon, \tag{2.6}$$

return x_k; otherwise, set

$$x_{k+1} = P_{C_{i_k}}\left(x_k + \gamma A^T(P_{Q_{j_k}} - I)Ax_k\right). \tag{2.7}$$

Step 4. Set $k = k + 1$, and go to **Step 1**.

In the context of **Step 2** within the preceding Algorithm 1, we introduce a temporary iteration point denoted as z_k. This intermediary point assumes a pivotal role in the formulation of the stopping criteria, as detailed in **Step 3**. To elucidate the connection between the solution of the MSFP denoted as (1.1), and the stipulated stopping rule denoted as (2.6) in Algorithm 1, we present the following lemma.

Lemma 2.3. *The iteration point x_k generated by Algorithm 1 is a solution of MSFP* (1.1) *if and only if*

$$\|x_k + y_k - z_k\| = 0. \tag{2.8}$$

Proof. First, assume the equality (2.8) holds, for any $z \in C$ be a solution of MSFP (1.1), it follows

$$\begin{aligned} 0 &= \langle x_k + y_k - z_k, x_k - z\rangle \\ &= \langle x_k - P_{C_{i_k}}(x_k), x_k - z\rangle + \langle A^T(I - P_{Q_{j_k}})Ax_k, x_k - z\rangle \\ &= \langle x_k - P_{C_{i_k}}(x_k), x_k - z\rangle + \langle Ax_k - P_{Q_{j_k}}(Ax_k), Ax_k - Az\rangle \\ &= \langle x_k - P_{C_{i_k}}(x_k), x_k - P_{C_{i_k}}(x_k)\rangle + \langle x_k - P_{C_{i_k}}(x_k), P_{C_{i_k}}(x_k) - z\rangle \\ &\quad + \langle Ax_k - P_{Q_{j_k}}(Ax_k), Ax_k - P_{Q_{j_k}}(Ax_k)\rangle + \langle Ax_k - P_{Q_{j_k}}(Ax_k), P_{Q_{j_k}}(Ax_n) - Az\rangle \\ &\geq \|x_k - P_{C_{i_k}}(x_k)\|^2 + \|Ax_k - P_{Q_{j_k}}(Ax_k)\|^2, \end{aligned}$$

the last inequality is obtained by the second property in Lemma 2.1. Then we have

$$\|x_k - P_{C_{i_k}}(x_k)\| = 0 \quad \text{and} \quad \|Ax_k - P_{Q_{j_k}}(Ax_k)\| = 0. \tag{2.9}$$

According to the definitions of i_k and j_k, from (2.9) it follows that

$$\|x_k - P_{C_i}(x_k)\| = 0, \quad 1 \leq i \leq s, \quad \text{and} \quad \|Ax_k - P_{Q_j}(Ax_k)\| = 0, \quad 1 \leq j \leq t.$$

Hence

$$x_k \in \bigcap_{i=1}^{s} C_i, \quad Ax_k \in \bigcap_{j=1}^{t} Q_j, \tag{2.10}$$

which implies that x_k is a solution of MSFP (1.1).

Conversely, let x_k be a solution of MSFP (1.1), then condition (2.10) holds, and

$$P_{C_{i_k}}(x_k) = x_k, \quad P_{Q_{j_k}}(Ax_k) = Ax_k.$$

From (2.5), we get

$$z_k = P_{C_{i_k}}(x_k) = x_k,$$

and

$$y_k = A^T(I - P_{Q_{j_k}})Ax_k = A^T\left(Ax_k - P_{Q_{j_k}}Ax_k\right) = 0.$$

Therefore,

$$x_k + y_k - z_k = 0,$$

which completes the proof.

□

As stipulated by Lemma 2.3, it becomes evident that condition (2.8) can be interpreted as an optimality criterion for the MSFP (1.1). Therefore, adopting rule (2.6) as the termination criteria for the extended CQ algorithm carries significant meaning and relevance. Furthermore, it's noteworthy that the convergence results of the classical CQ algorithm have been well-established under the condition that the relaxation parameter γ falls within the range of $(0, 2/L)$ [1]. In a similar vein, we can confidently establish the convergence result of the proposed extended CQ Algorithm 1 under the same conditions.

3 Convergence Analysis

In this section, we will delve into the convergence analysis of the proposed Algorithm 1. The foundation for this analysis will be built upon the convergence framework established for the CQ algorithm, as expounded in [1].

Denote by

$$S(x) = x + \gamma A^T (P_Q - I)Ax, \tag{3.1}$$

and

$$S_j(x) = x + \gamma A^T (P_{Q_j} - I)Ax, \quad j = 1, 2, \dots, t, \tag{3.2}$$

the iterative scheme (2.7) in Algorithm 1 can be rewritten as

$$x_{k+1} = P_{C_{i_k}}(S_{j_k}(x_k)).$$

Lemma 3.1. *Suppose A has full column rank, then the vector $\hat{c} \in C$ is a fixed point of mapping S, i.e., $S(\hat{c}) = \hat{c}$ if and only if $\hat{c}$ is a minimizer of function $\|P_Q(Ac) - Ac\|$ defined on C, and the minimum value is zero.*

Proof. Assume that $\hat{c}$ minimizes the function $\|P_Q(Ac) - Ac\|$ over $c \in C$ with the minimum value zero, *i.e.,*

$$\|P_Q(A\hat{c}) - A\hat{c}\| = 0,$$

which means

$$P_Q(A\hat{c}) = A\hat{c}.$$

Therefore

$$S(\hat{c}) = \hat{c} + \gamma A^T (P_Q(A\hat{c}) - A\hat{c}) = \hat{c}.$$

Conversely, assume that $S(\hat{c}) = \hat{c}$, it follows

$$\hat{c} + \gamma A^T (P_Q(A\hat{c}) - A\hat{c}) = \hat{c},$$

which implies

$$\gamma A^T (P_Q(A\hat{c}) - A\hat{c}) = 0.$$

Due to $\gamma > 0$ and A has full column rank, we have

$$P_Q(A\hat{c}) - A\hat{c} = 0, \tag{3.3}$$

i.e.,

$$\|P_Q(A\hat{c}) - A\hat{c}\| = 0.$$

This completes the proof. □

From the above Lemma 3.1, we have the following convergence results.

Corollary 3.2. *Let $\mathcal{E}$ be the set of all $c \in C$ at which the function $\|P_Q(Ac) - Ac\|$ attains its minimum value zero over C. Then every element in $\mathcal{E}$ is the solution of MSFP* (1.1).

Proof. The proof can be obtained directly from the definitions of C and Q. □

Theorem 3.3. *Suppose that A has full column rank and $\mathcal{E} \neq \emptyset$, the sequence $\{x^k\}$ generated by the iterative scheme* (2.7) *from Algorithm 1 converges to a member of $\mathcal{E}$.*

Proof. Let $\hat{c} \in \mathcal{E}$, then

$$\hat{c} = S(\hat{c}),$$

and

$$\hat{c} \in C \subset C_{i_k}.$$

By Lemma 2.1 and Lemma 3.1, it follows

$$\begin{aligned} \|\hat{c} - x_{k+1}\| &= \|P_{C_{i_k}}(\hat{c}) - P_{C_{i_k}}(S_{j_k}(x_k))\| \\ &= \|P_{C_{i_k}}(S_{j_k}\hat{c}) - P_{C_{i_k}}(S_{j_k}(x_k))\| \\ &\leq \|S_{j_k}\hat{c} - S_{j_k}(x_k)\|. \end{aligned} \tag{3.4}$$

From the definition of $S_{j_k}(\cdot)$, we have

$$\|S_{j_k}(\hat{c}) - S_{j_k}(x_k)\|^2 = \|\hat{c} - x_k + \gamma A^T(P_{Q_{j_k}} - I)A\hat{c} - \gamma A^T(P_{Q_{j_k}} - I)Ax_k\|^2.$$

By expanding and rearranging the right-hand side of the above equation, we get

$$\begin{aligned} \|S_{j_k}(\hat{c}) - S_{j_k}(x_k)\|^2 &= \|\hat{c} - x_k\|^2 + 2\gamma\langle A\hat{c} - Ax_k, P_{Q_{j_k}}(A\hat{c}) - P_{Q_{j_k}}(Ax_k) + Ax_k - A\hat{c}\rangle \\ &\quad + \gamma^2\|A^T(P_{Q_{j_k}} - I)A\hat{c} - A^T(P_{Q_{j_k}} - I)Ax_k\|^2 \\ &\leq \|\hat{c} - x_k\|^2 - 2\gamma\|A\hat{c} - Ax_k\|^2 + 2\gamma\langle A\hat{c} - Ax_k, P_{Q_{j_k}}(A\hat{c}) - P_{Q_{j_k}}(Ax_n)\rangle \\ &\quad + \gamma^2 L\|(P_{Q_{j_k}} - I)A\hat{c} - (P_{Q_{j_k}} - I)Ax_k\|^2. \end{aligned}$$

Notice that

$$\begin{aligned} & \|(P_{Q_{j_k}} - I)A\hat{c} - (P_{Q_{j_k}} - I)Ax_k\|^2 \\ = \ & \|P_{Q_{j_k}}(A\hat{c}) - P_{Q_{j_k}}(Ax_k)\|^2 - 2\langle A\hat{c} - Ax_k, P_{Q_{j_k}}(A\hat{c}) - P_{Q_{j_k}}(Ax_k)\rangle + \|A\hat{c} - Ax_k\|^2, \end{aligned}$$

it follows that

$$\begin{aligned} & \|S_{j_k}(\hat{c}) - S_{j_k}(x_k)\|^2 \\ \leq \ & \|\hat{c} - x_k\|^2 - (2\gamma - \gamma^2 L)(\|A\hat{c} - Ax_k\|^2) \\ & +\gamma^2 L(\|P_{Q_{j_k}}(A\hat{c}) - P_{Q_{j_k}}(Ax_k)\|^2 - \gamma^2 L\langle A\hat{c} - Ax_k, P_{Q_{j_k}}(A\hat{c}) \\ & -P_{Q_{j_k}}(Ax_k\rangle)) + (2\gamma - \gamma^2 L)\langle A\hat{c} - Ax_k, P_{Q_{j_k}}(A\hat{c}) - P_{Q_{j_k}}(Ax_k)\rangle \\ = \ & \|\hat{c} - x_k\|^2 - (2\gamma - \gamma^2 L)(\|A\hat{c} - Ax_k\|^2 - \langle A\hat{c} - Ax_k, P_{Q_{j_k}}(A\hat{c}) - P_{Q_{j_k}}(Ax_k)\rangle) \\ & -\gamma^2 L(\langle A\hat{c} - Ax_k, P_{Q_{j_k}}(A\hat{c}) - P_{Q_{j_k}}(Ax_k)\rangle - \|P_{Q_{j_k}}(A\hat{c}) - P_{Q_{j_k}}(Ax_k)\|^2). \end{aligned} \tag{3.5}$$

From the third inequality in Lemma 2.1, we have

$$\|P_{Q_{j_k}}(A\hat{c}) - P_{Q_{j_k}}(Ax_k)\|^2 - \langle A\hat{c} - Ax_k, P_{Q_{j_k}}(A\hat{c}) - P_{Q_{j_k}}(Ax_k)\rangle \leq 0, \tag{3.6}$$

by Cauchy's inequality and the non-expansiveness of projection operator, so that

$$\langle A\hat{c} - Ax_k, P_{Q_{j_k}}(A\hat{c}) - P_{Q_{j_k}}(Ax_k)\rangle \leq \|A\hat{c} - Ax_k\|^2. \tag{3.7}$$

Since $0 < \gamma < 2/L$, combined with (3.5) - (3.7), it follows

$$2\gamma - \gamma^2 L > 0,$$

and

$$\|S_{j_k}(\hat{c}) - S_{j_k}(x_k)\|^2 \leq \|\hat{c} - x_k\|^2.$$

In addition, from (3.4) to (3.5) we find that

$$\begin{aligned}\|\hat{c} - x_k\|^2 - \|\hat{c} - x_{k+1}\|^2 \geq{}& \gamma^2 L\langle A\hat{c} - Ax_k, P_{Q_{j_k}}(A\hat{c}) - P_{Q_{j_k}}(Ax_k)\rangle \\ &+ (2\gamma - \gamma^2 L)(\|A\hat{c} - Ax_k\|^2 \\ &- \langle A\hat{c} - Ax_k, P_{Q_{j_k}}(A\hat{c}) - P_{Q_{j_k}}(Ax_k)\rangle).\end{aligned} \tag{3.8}$$

Therefore, the sequence $\{\|\hat{c} - x_k\|^2\}$ is decreasing (so the sequence $\{x_k\}$ is bounded), and

$$\langle A\hat{c} - Ax_k, P_{Q_{j_k}}(A\hat{c}) - P_{Q_{j_k}}(Ax_k)\rangle - \|P_{Q_{j_k}}(A\hat{c}) - P_{Q_{j_k}}(Ax_k)\|^2 \to 0, \tag{3.9}$$

and

$$\|A\hat{c} - Ax_k\|^2 - \langle A\hat{c} - Ax_k, P_{Q_{j_k}}(A\hat{c}) - P_{Q_{j_k}}(Ax_k)\rangle \to 0, \tag{3.10}$$

since both sequences are non-negative.

Let x^* be an arbitrary cluster point of the sequence $\{x_k\}$, then $x^* \in C$. There exists an index pair (i^*, j^*) corresponding to C_{i^*} and Q_{j^*} with the greatest distances from x^* and Ax^*, respectively. In particular, we have

$$\langle A\hat{c} - Ax^*, P_{Q_{j^*}}(A\hat{c}) - P_{Q_{j^*}}(Ax^*)\rangle = \|P_{Q_{j^*}}(A\hat{c}) - P_{Q_{j^*}}(Ax^*)\|^2, \tag{3.11}$$

and

$$\langle A\hat{c} - Ax^*, P_{Q_{j^*}}(A\hat{c}) - P_{Q_{j^*}}(Ax^*)\rangle = \|A\hat{c} - Ax^*\|^2. \tag{3.12}$$

From the above equations (3.11) and (3.12), we get

$$\|A\hat{c} - Ax^*\| = \|P_{Q_{j^*}}(A\hat{c}) - P_{Q_{j^*}}(Ax^*)\|.$$

By Lemma 2.2, it follows that

$$\|P_{Q_{j^*}}(Ax^*) - Ax^*\| = \|P_{Q_{j^*}}(A\hat{c}) - A\hat{c}\| = 0. \tag{3.13}$$

Therefore, x^* is in the set $\mathcal{E}$. Replacing the generic $\hat{c} \in \mathcal{E}$ with x^*, it implies that the sequence $\{\|x^* - x_k\|\}$ is decreasing, and a subsequence converges to zero, thus the entire sequence converges to zero. This completes the proof. □

4 Numerical Experiments

This section is dedicated to the practical evaluation of the proposed extended CQ algorithm (Algorithm 1) through a series of numerical examples. All the program codes, which are available for reference[1], have been meticulously compiled and executed on the MATLAB R2020b platform, running on a Windows 10-based PC. The hardware configuration includes a CPU processor operating at 2.5GHz, and a memory capacity of 8GB.

[1]The MATLAB codes can be accessed at *http://pan.csu.edu.cn:80/link/6928B4B88951B11178761E4528E9553E.*

4.1 Example 1

In the first, we construct a simple MSFP with a class of convex sets

$$C_i = \{x \in \mathbb{R}^2 \mid \|x - d_i\| \le r_i\}, \quad i = 1, 2, 3, \tag{4.1}$$

where

$$d_1 = (1,\ 1)^T, \quad d_2 = (2.5,\ 2.5)^T, \quad d_3 = (3,\ 1)^T, \quad r_1 = 2, \quad r_2 = 1, \quad r_3 = 1.$$

By taking a point

$$x^* = (2.5,\ 1.6)^T \in C := \bigcap_{i=1}^{3} C_i,$$

and generating a matrix $A \in \mathbb{R}^{2\times 2}$ randomly, we construct the convex sets Q_j based on the vector $y^* = Ax^*$ as

$$Q_j = \{y \in \mathbb{R}^2 | L_j \le y \le U_j\}, \quad j = 1, 2, \tag{4.2}$$

where

$$L_1 = \lfloor y \rfloor - (1,\ 1)^T, \quad U_1 = \lceil y \rceil, \quad L_2 = \lfloor y \rfloor, \quad U_2 = \lceil y \rceil + (2,\ 2)^T.$$

Therefore,

$$y^* \in Q := \bigcap_{j=1}^{2} Q_j,$$

and x^* is a solution of MSFP (1.1), *i.e.*, the solution set of (1.1) is nonempty.

To empirically assess the numerical efficacy of the proposed extended CQ algorithm, denoted as 'ExtendCQ', for solving the created MSFP, we conduct a range of experiments. These experiments involve the application of Algorithm 1 to different initial points x_0. We draw comparisons between our extended CQ approach, the projection method ('Proj') initially introduced by Censor and his collaborators in [4], and the projection method augmented with a selection technique ('$\mathsf{Proj}_{\mathsf{select}}$') proposed by Yao *et al.* in [25]. For consistency, the termination criterion in all the algorithms employed in the following examples is set to $\varepsilon = q(x) = 10^{-4}$.

From the illustration of iterative sequences generated by different methods shown in Figure 3 and Figure 4, we can see that the proposed extended CQ algorithm with selection technique employs fewer iterations to achieve a solution than the other methods. It implies that the proposed method makes improvements to accelerate the convergence of the algorithm to solve MSFP. More comparison results on the number of iterations and CPU times are recorded in Table 1. From the reported results, it demonstrated that the proposed extended CQ algorithm with selection technique has better performance than other methods in terms of number of iterations and CPU costs.

4.2 Example 2

In this subsection, we set out to evaluate the efficacy of the proposed extended CQ algorithm. To do this, a spectrum of MSFP of varying problem sizes have been constructed. Our

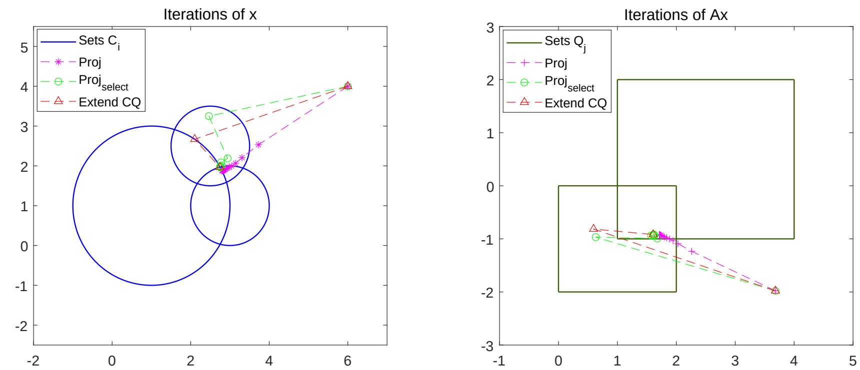

Figure 3: Convergent trajectories of iterative sequences by different methods with initial points $(6,\ 4)^T$.

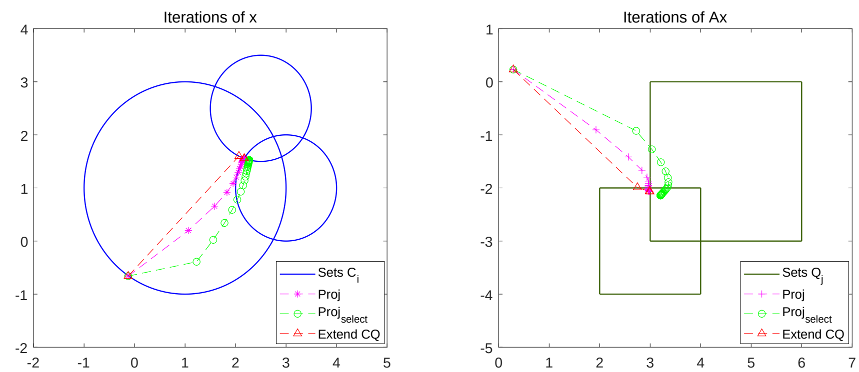

Figure 4: Convergent trajectories of iterative sequences by different methods with randomly generated initial points.

Table 1: Computational results of various methods to solve MSFP in Example 4.1.

Initial point	Iterations			CPU time (s)		
	Proj	$\text{Proj}_{\text{select}}$	Extend CQ	Proj	$\text{Proj}_{\text{select}}$	Extend CQ
$(0,\ 0)^T$	28	7	3	0.0191	0.0160	0.0027
$(1,\ 0)^T$	28	14	5	0.0246	0.0176	0.0080
$(1,\ 1)^T$	27	13	4	0.0191	0.0163	0.0030
$(4,\ 1)^T$	19	6	2	0.0194	0.0152	0.0020
$(6,\ 4)^T$	22	10	2	0.0184	0.0167	0.0021
$(2,\ -1)^T$	24	13	4	0.0187	0.0171	0.0030
$(-2,\ -1)^T$	29	7	1	0.0190	0.0154	0.0017
randn(2,1)	28	7	4	0.0186	0.0159	0.0032

approach involves the selection of a randomly generated point $x^* \in \mathbb{R}^n$ and a matrix $A \in \mathbb{R}^{n\times n}$ to formulate the convex sets C_i and Q_j, following a pattern similar to that outlined in (4.1) and (4.2) with

$$d_i = x^* + 10\xi_i, \quad r_i = \|d_i - x^*\| + 0.1, \quad i = 1, 2, \ldots, s,$$

and

$$L_j = y^* - 10\mu_j, \quad U_j = y^* + 10\nu_j, \quad j = 1, 2, \ldots, t,$$

where $y^* = Ax^*$, and $\xi, \mu, \nu \in \mathbb{R}^n$ are randomly generated vectors. From the construction of the above MSFPs, x^* has always been a solution to them, which implies that the solution sets of the constructed MSFPs are nonempty.

To comprehensively evaluate the performance of the introduced methods in addressing the different kinds of MSFPs, we systematically vary the problem size, setting $n = 2, 3, \ldots, 20$, and $s = t = n$ to establish a diverse set of MSFP instances. In order to provide a holistic assessment, we initialize the algorithms with 100 randomly generated vectors as the initial points x_0. Subsequently, we record and calculate the average results for the number of iterations and CPU times, as summarized in Table 4.2. The symbol " / " in the Table 4.2 indicates instances where the algorithm failed to meet the stopping criteria within 1000-iteration limit, and thus, the corresponding CPU time is not recorded. The results in Table 4.2 underscore that the proposed extended CQ algorithm with the selection technique, consistently outperforms the other methods in terms of both the number of iterations and computational efficiency. This reaffirms its effectiveness and competitiveness.

Table 2: Computational results of various methods to solve MSFP in Example 4.2.

$s = t = n$	Iterations			CPU time (s)		
	Proj	$\text{Proj}_{\text{select}}$	Extend CQ	Proj	$\text{Proj}_{\text{select}}$	Extend CQ
2	94.92	36.97	25.53	0.0016	0.0028	0.0017
3	173.11	100.74	14.15	0.0038	0.0108	0.0011
4	618.49	256.53	45.27	0.0100	0.0131	0.0036
5	425.55	167.45	32.08	0.0089	0.0120	0.0028
6	695.19	483.46	54.13	0.0157	0.0233	0.0041
7	/	/	160.19	/	/	0.0084
8	/	677.61	84.71	/	0.0356	0.0059
9	/	997.62	166.25	/	0.0555	0.0106
10	/	979.69	219.81	/	0.0587	0.0139
15	/	/	162.92	/	/	0.0150
20	/	/	355.97	/	/	0.0394

5 Conclusion

In this paper, we propose an extended CQ algorithm to address the MSFP integrated with a selection technique. We also provide an analysis of the convergence properties of the extended CQ algorithm under relatively lenient conditions. Our empirical findings unequivocally validate that the extended CQ algorithm integrated with the selection technique, yields superior results when applied to solve the MSFP.

References

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

[2] C. Byrne, Unified treatment of some algorithms in signal processing and image construction, *Inverse Probl.* 20 (2002) 103–120.

[3] Y. Censor and T. Elfving, A multiprojection algorithm using Bregman projections in a product space, *Numer. Algorithms* 8 (1994) 221–239.

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

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

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

[7] Y. Censor and A. Segal, Iterative projection methods in biomedical inverse problems, *Mathematical Methods in Biomedical Imaging and Intensity-modulated Radiation Therapy (IMRT)* 10 (2008) 65–96.

[8] P.P. Chen, H.J. He, Y.-C. Liou and C.-F. Wen, Convergence rate of the CQ algorithm for split feasibility problems, *J. Nonlinear. Convex A.* 19 (2018) 381–395.

[9] W. Cheney and A.A Goldstein, Proximity maps for convex sets, *in Proceedings of the American Mathematical Society.* 10 (1959) 448–450.

[10] Y.Z. Dang, M. Ang and J. Sun, An inertial triple-projection algorithm for solving the split feasibility problems, *J. Ind. Manag. Optim.* 19 (2023) 1813–1826.

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

[12] F. Facchinei and J.S. Pang, *Finite-Dimensional Variational Inequalities and Complementarity Problems*, Springer, 2003.

[13] A. Gibali, K.H. Küfer and P. Süss, Successive linear programing approach for solving the nonlinear split feasibility problem, *J. Nonlinear Convex A.* 15 (2014) 345–353.

[14] S.N. He, Z.Y. Zhao and B. Luo, A relaxed self-adaptive CQ algorithm for the multiple-sets split feasibility problem, *Optimization* 15 (2015) 1907–1918.

[15] Z.B. Li, D.R. Han and W.X. Zhang, A self-adaptive projection-type method for nonlinear multiple-sets split feasibility problem, *Inverse Probl. Sci. En.* 21 (2013) 155–170.

[16] T.X. Lu, L.L. Zhao and H.J. He, A multi-view on the CQ algorithm for split feasibility problems: from optimization lens, *J. Appl. Numer.* 2 (2020) 387–399.

[17] B. Qu and N.H. Xiu, A note on the CQ algorithm for the split feasibility problem, *Inverse Probl.* 21 (2005) 1655–1665.

[18] B. Qu and H.X. Chang, Remark on the successive projection algorithm for the multiple-sets split feasibility problem, *Numer. Funct. Anal. Optim.* 38 (2017) 1614–1623.

[19] S. Reich and T.M. Tuyen, Projection algorithms for solving the split feasibility problem with multiple output sets, *J. Optimiz. Theor. App.* 190 (2021) 861–878.

[20] S. Suantai, N. Pholasa and P. Cholamjiak, Relaxed CQ algorithms involving the inertial technique for multiple-sets split feasibility problems, *RACSAM Rev. R. Acad A.* 113 (2019) 1081–1099.

[21] G.H. Taddele, P. Kumam and V. Berinde, An extended inertial Halpern-type ball-relaxed CQ algorithm for multiple-sets split feasibility problem, *Ann. Funct. Anal.* 13 (2022) 48.

[22] G.H. Taddele, P. Kumam, P. Sunthrayuth and A.G. Gebrie, A general algorithm for multiple-sets split feasibility problem involving resolvents and Bregman mappings, *Numer. Algorithms* 92 (2023) 1335–1366.

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

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

[25] Y.H. Yao, M. Postolache and Z.C. Zhu, Gradient methods with selection technique for the multiple-sets split feasibility problem, *Optimization* 69 (2019) 269–281.

[26] W.X. Zhang, D.R. Han and Z.B. Li, A self-adaptive projection method for solving the multiple-sets split feasibility problem, *Inverse Probl.* 25 (2009) 115001.

[27] W.X. Zhang, D.R. Han and X.M. Yuan, An efficient simultaneous method for the constrained multiple-sets split feasibility problem, *Comput. Optim. Appl.* 25 (2012) 825–843.

[28] J.L. Zhao and Q.Z. Yang, Self-adaptive projection methods for the multiple-sets split feasibility problem, *Inverse Probl.* 27 (2011): 035009.

[29] J.L. Zhao, Y.J. Zhang and Q.Z. Yang, Modified projection methods for the split feasibility problem and the multiple-sets split feasibility problem, *Appl. Math. Comput.* 219(4) (2012) 1644–1653.

[30] J.L. Zhao and Q.Z. Yang, A simple projection method for solving the multiple-sets split feasibility problem, *Inverse Probl. Sci. En.* 21 (2013) 537–546.

Manuscript received 12 October 2023
revised 18 December 2023
accepted for publication 26 December 2023

Huan Gao
School of Mathematics and Statistics
Hunan First Normal University
Changsha, Hunan 410205, China
E-mail address: huanhuan135213@163.com

Zhibao Li
School of Mathematics and Statistics
Central South University, Changsha
Hunan 410083, China
E-mail address: zblimath@csu.edu.cn

Guangming Zhou
School of Mathematics and Computational Science
Hunan Key Laboratory for Computation and
Simulation in Science and Engineering
Xiangtan University, Xiangtan, Hunan 411105, China
E-mail address: zhougm@xtu.edu.cn