FAST SELF-ADAPTIVE REGULARIZATION ITERATIVE ALGORITHM FOR SOLVING SPLIT FEASIBILITY PROBLEM

YA-ZHENG DANG\textsuperscript{1}, ZHONG-HUI XUE\textsuperscript{2, *}, YAN GAO\textsuperscript{1} AND JUN-XIANG LI\textsuperscript{1}

\textsuperscript{1} School of Management
University of Shanghai for Science and Technology, Shanghai 200093, China
\textsuperscript{2} Shanghai Publication and Printing College, Shanghai 200093, China
(Communicated by Jie Sun)

\textbf{Abstract.} Split feasibility problem (SFP) is to find a point which belongs to one convex set in one space, such that its image under a linear transformation belongs to another convex set in the image space. This paper deals with a unified regularized SFP for the convex case. We first construct a self-adaptive regularization iterative algorithm by using Armijo-like search for the SFP and show it converges at a sublinear rate of $O(1/k)$, where $k$ represents the number of iterations. More interestingly, inspired by the acceleration technique introduced by Nesterov\cite{12}, we present a fast Armijo-like regularization iterative algorithm and show it converges at rate of $O(1/k^2)$. The efficiency of the algorithms is demonstrated by some random data and image deblurring problems.

1. \textbf{Introduction.} Split feasibility problem (SFP) is to find a point

$$x^* \in C \text{ and } Ax^* \in Q,$$

(1.1)

where $C$ and $Q$ are nonempty, closed and convex subsets of $\mathbb{R}^N$ and $\mathbb{R}^M$, respectively; $A : \mathbb{R}^N \to \mathbb{R}^M$ is a given bounded linear operator (here we denote its adjoint operator as $A^T$). This problem was first proposed by Censer and Elfving\cite{4} to model for inverse problems arising from medical image reconstruction and intensity modulated radiation therapy\cite{1, 3}. Various algorithms have been invented to solve it (see \cite{5, 14, 17, 19, 20} and references therein). If the set $Q$ takes the form of $Q := \{b \in \mathbb{R}^M\}$, the SFP collapses to the well studied Linear Inverse Problem (LIP) (see \cite{6, 16} and the references therein). Hence, SFP is a natural generalization of LIP.

CQ algorithm is one of the most popular solvers for SFP which was first introduced by Byrne\cite{2}. Indeed, the CQ algorithm can be regarded as an application of the gradient-projection method to SFP\cite{11}, when we formulate (1.1) as a convex minimization problem. An important notice is that we often face the case where $A$ is ill-posed in many applications such as image reconstruction. To deal with the ill-posed case, ones usually use the popular Tikhonov regularization to promote stability of solutions. Such a regularization technique also has been applied to SFP

\textsuperscript{2010 Mathematics Subject Classification.} 90C25, 47J25.

\textit{Key words and phrases.} Split feasible problem, accelerated iterative algorithm, Armijo-like search, regularization, iteration complexity, image deblurring.

\textsuperscript{*} Corresponding author.
in recent works [18, 10, 9]. However, with the increasing dimensionality of the data, many problems have some more special structures, such as (group) sparsity, low-rankness making the Tikhonov regularization not fitting for such structures. As a consequence, the numerical solutions obtained from the traditional model (1.1) often are not satisfactory, which also encourages us to consider a more general model of SFP embracing a generic regularization term. Mathematically, the regularized SFP takes the form of:

$$\min_x \{ \varphi(x) \mid x \in C, \ Ax \in Q \}, \quad (1.2)$$

where $\varphi(x)$ is a convex function, which serves as the role of regularization. Obviously, when $\varphi(x) := 0$ and $\varphi(x) := \frac{1}{2}\|x\|^2$, model (1.2) reduces to the unregularized and Tikhonov regularized SFP, respectively. If we take $\varphi(x) := \|x\|_1$, it is the model considered in [10, 9] for promoting sparse solutions of the SFP. Moreover, model (1.2) also covers regularized LIPs studied in [15].

Similar to the reformulation, we transfer the constraint $Ax \in Q$ into the objective function of (1.2) and get the following minimization problem:

$$\min_x \{ F(x) := \sigma \varphi(x) + f(x) \mid x \in C \}, \quad (1.3)$$

where $\sigma > 0$ is a regularization parameter and

$$f(x) := \frac{1}{2}\|Ax - P_Q(Ax)\|^2, \quad (1.4)$$

where $P_Q(Ax)$ denotes the orthogonal projection of $Ax$ onto $Q$, that is,

$$P_Q(Ax) = \arg\min_{y \in Q} \|Ax - y\|.$$

In this paper, we only consider the case where the regularize function is convex for the purpose of algorithmic design and we name our algorithms as regularization iterative algorithm since we extend our problem to the general regularization optimization problem. We construct a self-adaptive regularization CQ algorithm by using Armijo-like technique in a new way for (1.3) and show it converges at a subliner rate of $O(1/k)$, where $k$ represents the number of iterations, which, as far as we know, is not discussed in the SFP literature. To further improve its convergence rate, we employ the acceleration technique introduced by Nesterov [12] to the CQ algorithm such that the resulting algorithm is also convergent with rate of $O(1/k^2)$. All theoretical results are clearly demonstrated through preliminary numerical results with applications to random synthetic data and image deblurring problems.

The rest of the paper is organized as follows. Some basic definitions and lemmas are given in Section 2. Section 3 presents the self-adaptive regularization CQ algorithm (new version construction of CQ algorithm) and show its convergence in a new way. In Section 4, an accelerated self-adaptive regularization CQ algorithm is proposed and the convergence is proved under some suitable conditions. Section 5 is devoted to numerical applications to general split feasibility problem (random synthetic data) with no regularization and image deblurring problems. The conclusion appears in Section 6.

2. Preliminaries. In this section, we give some preliminaries which will be used in the sequel. Let $\mathbb{R}^N$ be an $N$-dimensional Euclidean space. For any two vectors $x, y \in \mathbb{R}^N$, $\langle x, y \rangle$ denotes the standard inner product. Furthermore, we denote
\[ f(x) = \| x \|_1, \| x \|_2 \] and \[ \| x \|_\infty \] the standard \( l_1 \)-norm, \( l_2 \)-norm (Euclidean) norm, and \( l_\infty \)-norm, respectively.

**Definition 2.1** [7]. Let \( f : \mathbb{R}^N \to \mathbb{R} \) be convex. The subdifferential of \( f \) at \( x \) is defined as

\[
\partial f(x) = \{ \xi \in \mathbb{R}^N \mid f(y) \geq f(x) + \langle \xi, y-x \rangle, \forall y \in \mathbb{R}^N \}.
\]

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

**Definition 2.2.** Let \( f(\cdot) : \mathbb{R}^N \to \mathbb{R} \) be a convex function, its gradient \( \nabla f(\cdot) \) is said to be Lipschitz continuous on \( \mathbb{R}^N \) with constant \( L_f > 0 \), if

\[
\| \nabla f(x) - \nabla f(y) \| \leq L_f \| x - y \|, \forall x, y \in \mathbb{R}^N.
\]

We know the following results.

**Lemma 2.1** [Lemma 8.1, [1]]. Let \( f : \mathbb{R}^N \to \mathbb{R} \) be given by \( f(x) = \frac{1}{2} \| (I - P_Q)Ax \|^2 \). Then,

1. \( f \) is convex and differentiable;
2. \( \nabla f(x) = A^T(I - P_Q)Ax, x \in \mathbb{R}^N; \)
3. \( \nabla f(\cdot) \) is Lipschitz continuous with a constant \( \| A \|^2 \), that is, \( \| \nabla f(x) - \nabla f(y) \| \leq \| A \|^2 \cdot \| x - y \| \) for all \( x, y \in \mathbb{R}^N \), where \( \| A \| \) stands for the matrix \( l_2 \)-norm of the matrix \( A \).

**Lemma 2.2.** Let \( \{ a_k \} \) and \( \{ b_k \} \) be two positive sequences of reals satisfying

\[
a_k - a_{k+1} \geq b_{k+1} - b_k, \forall k \geq 1, \text{ with } a_1 + b_1 \leq c, \ c > 0.
\]

Then, \( a_k \leq c \) for every \( k \geq 1 \).

3. **Self-adaptive regularization CQ algorithm and its convergence.** Define the following minimization problem:

\[
(P) \quad \min_x \{ F(x) := \sigma \varphi(x) + f(x) \mid x \in C \},
\]

where \( f(x) := \frac{1}{2} \| Ax - P_Q(Ax) \|^2 \), \( \varphi(x) \) is defined as in (1.2), \( \sigma > 0 \) is the regularization parameter. Note that \( f(x) \) defined as above is a convex and differentiable function. For any \( L > 0 \), consider the following quadratic approximation of \( F(x) \) at a given point \( y \):

\[
Q_L(x, y) := [f(y) + \langle x - y, \nabla f(y) \rangle + \frac{L}{2} \| x - y \|^2] + \sigma \varphi(x).
\]

Suppose \( p_L(y) \) is the minimizer of the approximation function \( Q_L(x, y) \) at point \( y \) on set \( C \), that is,

\[
p_L(y) := \arg \min_{x \in C} \{ Q_L(x, y) \}.
\]

By ignoring constant terms in \( y \), we can further simplify the above scheme and easily obtain

\[
p_L(y) = \arg \min_{x \in C} \{ \sigma \varphi(x) + \frac{L}{2} \| x - (y - \frac{1}{L} \nabla f(y)) \|^2 \}.
\]

Clearly, from (3.3), we know the basic step of regularization CQ (projection gradient) method for the problem (P) is

\[
x^k = p_L(x^{k-1}),
\]

where \( \frac{1}{L} \) plays the role of a stepsize.
If we consider the standard SFP without regularization term, that is, $\varphi(x) = 0$, the previous algorithm reduces to the standard CQ algorithm introduced in [1], that is

$$x^k = p_L(x^{k-1}) = P_C[x^{k-1} - \frac{1}{L} A^T(I - P_Q)Ax^{k-1}],$$

(3.4)

where $L \geq \|A\|^2$.

However, the standard CQ algorithm employs a fixed stepsize related to $\|A\|^2$ which adds amount of calculation, and the next iterate generated by the algorithm relies only on the current one, without previously computed iterates, which leads to slow convergence. Hence, in this paper, we will present a self-adaptive regularization iterative algorithm by adopting Armijo-like technique, which needs not to compute $\|A\|^2$ and makes a sufficient decrease of the objective function at each iterate, see the following algorithm 3.1. To further improve the convergence, we will purpose an accelerated self-adaptive iterative algorithm based on Algorithm 3.1, which computes the next iterate based not only on the previous one, but also on two or more previously computed iterates, more details can be found in Section 4.

Before proceeding with proposing our algorithms we recall the following lemma, which is the well-known fundamental property for a smooth function in the class $C^{1,1}$ [Lemma 3.2. [12], [13]]. It will be crucial for the convergence analyses of self-adaptive regularization CQ algorithm and the new accelerated self-adaptive regularization iterative method.

**Lemma 3.1.** Let $f : \mathbb{R}^N \to \mathbb{R}$ be a continuously differentiable function with Lipschitz continuous gradient and Lipschitz constant $L_f$. Then, for any $L \geq L_f$,

$$f(x) \leq f(y) + \langle x - y, \nabla f(y) \rangle + \frac{L}{2} \|x - y\|^2, \quad \forall x, y \in \mathbb{R}^N. \quad (3.5)$$

Note that from Lemma 3.1, if $L \geq L_f$, then for any $y \in \mathbb{R}^N$,

$$F(p_L(y)) \leq Q(p_L(y), y). \quad (3.6)$$

Now, we construct the self-adaptive regularization CQ Algorithm for model (3.1) as follows.

**Algorithm 3.1 Self-adaptive regularization CQ algorithm**

1. Choose $\beta > 0$, some $\eta > 1$, $\gamma \in (0, 1)$ and an initial point $x^0 \in C$.
2. For $k = 1, 2, \cdots$, update the next iterate $x^k$ via

$$x^k := \arg\min_{x \in C} \{\sigma \varphi(x) + \frac{L_k}{2}\|x - (x^{k-1} - \frac{1}{L_k} \nabla f(x^{k-1}))\|^2\}, \quad (3.7)$$

where $L_k = \eta m_k \beta$ and $m_k$ is the smallest nonnegative integer $m$, such that

$$f(x^k) \leq f(x^{k-1}) - \gamma \langle x^k - x^{k-1}, \nabla f(x^{k-1}) \rangle \quad (3.8)$$

and

$$0 \leq (1 + \gamma)\langle x^k - x^{k-1}, \nabla f(x^{k-1}) \rangle + \frac{\eta m_k \beta}{2}\|x^k - x^{k-1}\|^2. \quad (3.9)$$

3. End for.

**Remark 3.1.** Clearly, from (3.8) and (3.9), we have the following inequality:
\[ f(x^k) \leq f(x^{k-1}) + (x^k - x^{k-1}, \nabla f(x^{k-1})) + \frac{\eta m \beta}{2} \|x^k - x^{k-1}\|^2, \quad (3.10) \]

then, from Lemma 3.1, we know that \( \eta m \beta \geq L_f = \|A\|^2 \) and
\[ F(p_{\eta m \beta}(x^{k-1})) \leq Q(p_{\eta m \beta}(x^{k-1}), x^{k-1}) \quad (3.11) \]
for the selected \( m \).

**Remark 3.2.** From Remark 3.1, we know that two conditions (3.8) and (3.9) are necessary for (3.11). But from (3.11), we can not obtain (3.8) and (3.9). In fact, (3.8) assures the existence of upper boundary and lower boundary of the parameter \( L_k \), see details from Lemma 3.2.

**Remark 3.3.** If there is no (3.9) in Algorithm 3.1 and \( \varphi(x) = 0 \), it recovers the standard self-adaptive CQ algorithm introduced in [5], that is

**General self-adaptive CQ algorithm** [5]

1: Choose \( \beta > 0 \), some \( \eta > 1 \), \( \gamma \in (0, 1) \) and an initial point \( x^0 \in C \).
2: For \( k = 1, 2, \ldots \), update the next iterate \( x^k \) via
\[ x^k = P_C[x^{k-1} - \alpha_k A^T(I - P_Q)Ax^{k-1}], \quad (3.12) \]
where \( \gamma \in (0, 1) \), \( \alpha_k = \frac{1}{L_k} = \frac{1}{\eta m \gamma} \) and \( m_k \) is the smallest nonnegative integer \( m \) such that
\[ f(x^k) \leq f(x^{k-1}) - \gamma \langle \nabla f(x^{k-1}), x^{k-1} - x^k \rangle. \quad (3.13) \]
3. End for.

Therefore, Algorithm 3.1 is an extension of the classical self-adaptive CQ algorithm.

**Remark 3.4.** Note that the sequence of function values \( \{ F(x^k) \} \) produced by Algorithm 3.1 is nonincreasing. Indeed, for every \( k \geq 1 \),
\[ F(x^k) \leq Q_{L_k}(x^k, x^{k-1}) \leq Q_{L_k}(x^{k-1}, x^{k-1}) = F(x^{k-1}). \]

**Lemma 3.2.** Suppose parameters \( \beta > 0 \), \( \eta > 1 \) defined as in Algorithm 3.1, then
\[ \beta \leq L_k \leq \eta \|A\|^2, \quad \forall k \geq 1. \]

**Proof.** Obviously, from the search rule (3.8), \( \beta \leq L_k \) for every \( k \geq 1 \). Since inequality (3.8) and (3.9) are satisfied for some \( \beta \eta^m \geq \|A\|^2 \), we know that \( L_k/\eta \) must violate it, i.e., \( L_k \leq \eta \|A\|^2 \) for every \( k \geq 1 \). Hence, \( \beta \leq L_k \leq \eta \|A\|^2 \), for every \( k \geq 1 \). This completes the proof.

Before narrating the next lemma, we first exhibit the first-order optimality condition of (3.7):
\[ \langle x - x^k, \sigma \xi^k + \nabla f(x^k) + L_k(x^k - x^{k-1}) \rangle \geq 0, \quad \forall x \in C, \]
or equivalently,
\[ \langle x - x^k, \sigma \xi^k + \nabla f(x^k) \rangle \geq L_k \langle x - x^k, (x^k - x^{k-1}) \rangle, \quad \forall x \in C, \quad (3.14) \]
where \( \xi^k \in \partial \varphi(x^k) \). It is a variational inequality problem.

Accordingly, we can obtain the following result.
Lemma 3.3. For any $x \in C$, we have

$$F(x) - F(x^k) \geq \frac{L_k}{2} \|x^k - x^{k-1}\|^2 + L_k(x - x^{k-1}, x^{k-1} - x^k), \forall k \geq 1. \quad (3.15)$$

Proof. From (3.11), we know

$$F(x) - F(x^k) \geq F(x) - Q_{L_k}(x^k, x^{k-1}). \quad (3.16)$$

Since $f$ and $\varphi$ are convex, we have

$$f(x) \geq f(x^{k-1}) + \langle x - x^{k-1}, \nabla f(x^{k-1}) \rangle,$$

$$\sigma \varphi(x) \geq \sigma \varphi(x^k) + \sigma(x - x^k, \xi^k),$$

where $\xi^k \in \partial \varphi(x^k)$. Adding the above two inequalities yields

$$F(x) \geq f(x^{k-1}) + \langle x - x^{k-1}, \nabla f(x^{k-1}) + \sigma \varphi(x) \rangle. \quad (3.17)$$

On the other hand, by setting $x := x^k$, $y := x^{k-1}$ in (3.2), we get

$$Q_{L_k}(x^k, x^{k-1}) := [f(x^{k-1}) + \langle x^k - x^{k-1}, \nabla f(x^{k-1}) \rangle + \frac{L_k}{2} \|x^k - x^{k-1}\|^2] + \sigma \varphi(x). \quad (3.18)$$

Therefore, substituting (3.17) and (3.18) into (3.16), we obtain

$$F(x) - F(x^k) \geq F(x) - Q_{L_k}(x^k, x^{k-1})$$

$$\geq \langle x - x^k, \nabla f(x^{k-1}) + \sigma \xi^k \rangle - \frac{L_k}{2} \|x^k - x^{k-1}\|^2.$$

From (3.14), we arrive at

$$F(x) - F(x^k) \geq L_k\langle x - x^k, x^{k-1} - x^k \rangle - \frac{L_k}{2} \|x^k - x^{k-1}\|^2$$

$$= L_k\langle x - x^{k-1}, (x^{k-1} - x^k) \rangle + \frac{L_k}{2} \|x^k - x^{k-1}\|^2.$$

Hence, we obtain the results. \qed

Theorem 3.1. Suppose the solution set of (3.1) is nonempty. Let $z$ be an arbitrary solution of (3.1), and let $\{x^k\}$ be a sequence generated by Algorithm 3.1. Then, for any $k \geq 1$,

$$F(x^k) - F(z) \leq \frac{L \|x^0 - z\|^2}{2k}.$$

Proof. Let $x = z$, $k = h$ in Lemma 3.3, we obtain

$$\frac{2}{L_h}(F(z) - F(x^h)) \geq \|x^h - x^{h-1}\|^2 + 2\|z - x^{h-1}, x^{h-1} - x^h\|$$

$$\geq \|z - x^h\|^2 - \|z - x^{h-1}\|^2.$$

From Lemma 3.2 and the fact that $F(z) - F(x^k) \leq 0$, we have

$$\frac{2}{\eta \|A\|^2}(F(z) - F(x^h)) \geq \|z - x^h\|^2 - \|z - x^{h-1}\|^2.$$

Summing the above inequality over $h = 1, 2, \cdots, k$ gives

$$\frac{2}{\eta \|A\|^2}(kF(z) - \sum_{h=1}^{k} F(x^h)) \geq \|z - x^k\|^2 - \|z - x^0\|^2. \quad (3.19)$$
We note that Lemma 3.3 holds for any \( x \in C \). Hence, taking \( x := x^h, x^k := x^h \) and \( x^{k-1} := x^{h-1} \) in (3.15) leads to
\[
\frac{2}{L_h}(F(x^{k-1}) - F(x^h)) \geq \|x^h - x^{h-1}\|^2.
\]
Since \( L_h \geq \beta \) and \( F(x^{k-1}) - F(x^h) > 0 \), it follows that
\[
\frac{2}{\beta}(F(x^{k-1}) - F(x^h)) \geq \|x^h - x^{h-1}\|^2.
\]
Multiplying the above inequality by \((h-1)\) and summing over \( h = 1, 2, \cdots, k \), we obtain
\[
\frac{2}{\beta} \sum_{h=1}^{k}((h-1)F(x^{k-1}) - hF(x^h) + F(x^h)) \geq \sum_{h=1}^{k} (h-1)\|x^h - x^{h-1}\|^2,
\]
which implies
\[
\frac{2}{\beta}(-kF(x^k) + \sum_{h=1}^{k} F(x^h)) \geq \sum_{h=1}^{k} (h-1)\|x^h - x^{h-1}\|^2.
\] (3.20)
Adding (3.19) and (3.20) times \( \frac{\beta}{\eta\|A\|^2} \), we get
\[
\frac{2k}{\eta\|A\|^2}(F(z) + F(x^{k-1})) \geq \|z - x^{k-1}\|^2 + \beta \frac{\eta\|A\|^2}{\eta\|A\|^2} \sum_{h=1}^{k} (h-1)\|x^h - x^{h-1}\|^2 - \|z - x^0\|.
\]
which can be simplified as
\[
\frac{2k}{\eta\|A\|^2}(F(z) + F(x^{k-1})) \geq -\|z - x^0\|.
\]
Then,
\[
F(x^{k-1}) - F(z) \geq \frac{\eta\|A\|^2}{2k} \|z - x^0\|^2.
\]
The assertion of this theorem is obtained. \( \square \)

**Remark 3.5.** The above theorem means that obtaining an \( \varepsilon \)-optimal solution, denoted by \( \hat{x} \), requires the number of iterations at most \( \lceil \zeta/\varepsilon \rceil \) such that \( F(\hat{x}) - F(z) \leq \varepsilon \), where \( \zeta := \eta\|A\|^2\|x^0 - x^\ast\|^2 \).

4. Fast regularization CQ algorithm and its convergence. In this section, we will present a fast self-adaptive regularization CQ algorithm and show its convergence, which is simple but with improved convergence.

**Algorithm 4.1 Fast self-adaptive regularization CQ algorithm**

1: Choose \( \beta > 0 \), some \( \eta > 1, \gamma \in (0, 1), \ t_1 = 1 \) and an initial point \( x^0 \in C \).
2: For \( k = 1, 2, \cdots \), update the next iterate \( x^k \) via
\[
x^k := \arg \min_{x \in C} \{\sigma \varphi(x) + \frac{L_k}{2}\|x - (y^k - \frac{1}{L_k}\nabla f(y^k))\|\}, \quad (4.1)
\]
where \( L_k = \eta^m_\ast \beta \) and \( m_\ast \) is the smallest nonnegative integer \( m \), such that
\[
f(x^k) \leq f(y^k) - \gamma\langle x^k - y^k, \nabla f(y^k) \rangle. \quad (4.2)
\]
and
\[ 0 \leq (1 + \gamma)(x^k - y^k, \nabla f(y^k)) + \frac{\eta \alpha^2}{2} \|x^k - y^k\|^2, \]  \hspace{1cm} (4.3)
parameters
\[ t_{k+1} = \frac{1 + \sqrt{1 + 4t_k^2}}{2}, \]  \hspace{1cm} (4.4)
\[ y^{k+1} = x^k + \left( \frac{t_k - 1}{t_{k+1}} \right)(x^k - x^{k-1}) \]  \hspace{1cm} (4.5)

3: End for.

Similarly, the first-order optimality condition of (4.1) is also equivalent to a variational inequality:
\[ (x - x^k, \sigma \xi^k + \nabla f(x^k)) \geq L_k(x - x^k, y^k - x^k), \forall x \in C, \]
where \( \xi^k \in \partial \phi(x^k) \).

Accordingly, we can obtain the following Lemma 4.1.

**Lemma 4.1.** Let \( \{x^k\} \) and \( \{y^k\} \) be two sequences generated by Algorithm 4.1. Then, for any \( x \in C \), we have
\[ F(x) - F(x^k) \geq \frac{L_k}{2} \|x^k - y^k\|^2 + L_k(x - y^k, y^k - x^k), \forall k \geq 1. \]  \hspace{1cm} (4.6)

**Proof.** The process of proof is similar to that of Lemma 3.3 with setting \( x^{k-1} \) instead of \( y^k \). \( \Box \)

**Lemma 4.2.** Let \( z \) be an arbitrary solution of (3.1). Then, the sequences \( \{x^k\} \) and \( \{y^k\} \) generated by Algorithm 4.1 satisfy
\[ \frac{2}{L_k} t_k^2 u^k - t_{k+1}^2 u^{k+1} \geq \|v^{k+1}\|^2 - \|v^k\|^2, \]  \hspace{1cm} (4.7)
where \( u^k := F(x^k) - F(z) \), \( v^k := t_k x^k - (t_k - 1)x^{k-1} - z \).

**Proof.** It is easy to see from (4.6) that
\[ F(x) - F(x^{k+1}) \geq \frac{L_k}{2} \|x^{k+1} - y^{k+1}\|^2 + L_k(x - y^{k+1}, y^{k+1} - x^{k+1}) \]
holds for arbitrary \( x \in C \). Let \( x := x^k \) and \( x := z \) in the above inequality, respectively, we have
\[ F(x^k) - F(x^{k+1}) \geq \frac{L_k}{2} \|x^{k+1} - y^{k+1}\|^2 + L_k(x - y^{k+1}, y^{k+1} - x^{k+1}) \]
and
\[ F(z) - F(x^{k+1}) \geq \frac{L_k}{2} \|x^{k+1} - y^{k+1}\|^2 + L_k(z - y^{k+1}, y^{k+1} - x^{k+1}). \]

By the definition of \( u^k \), the above two inequalities can be rewritten as
\[ \frac{2}{L_k+1} (u^k - u^{k+1}) \geq \|x^{k+1} - y^{k+1}\|^2 + 2(x^k - y^{k+1}, y^{k+1} - x^{k+1}) \]  \hspace{1cm} (4.8)
and
\[ -\frac{2}{L_k+1} u^{k+1} \geq \|x^{k+1} - y^{k+1}\|^2 + 2(z - y^{k+1}, y^{k+1} - x^{k+1}). \]  \hspace{1cm} (4.9)
Multiplying by \((t_{k+1} - 1)\) in (4.8) and adding it to (4.9) yields
\[
\frac{2}{L_{k+1}}(t_{k+1} - 1)u^k - t_{k+1}u^{k+1} \geq t_{k+1}\|x^{k+1} - y^{k+1}\|^2
+ 2(x^{k+1} - y^{k+1}, t_{k+1}y^{k+1} - (t_{k+1} - 1)x^k - z).
\]
\[
\tag{4.10}
\]
Notice that (4.4) implies that \(t_k^2 = t_{k+1}^2 - t_{k+1}\). Thus, we multiply (4.10) by \(t_{k+1}\) and obtain
\[
\frac{2}{L_{k+1}}(t_k^2u^k - t_{k+1}^2u^{k+1}) \geq t_{k+1}^2\|x^{k+1} - y^{k+1}\|^2
+ 2t_{k+1}(x^{k+1} - y^{k+1}, t_{k+1}y^{k+1} - (t_{k+1} - 1)x^k - z).
\]
\[
\tag{4.11}
\]
Thus, let \(a := t_{k+1}^2 + 1\), \(b := t_{k+1}x^{k+1}\) and \(c := (t_{k+1} - 1)x^k + z\). By the Pythagoras relation
\[
\|a - a\|^2 + 2(b, a - c) = \|b - c\|^2 - \|a - c\|^2,
\]
and (4.11), we get
\[
\frac{2}{L_k}t_k^2u^k - t_{k+1}^2u^{k+1} \geq \|v^{k+1}\|^2 - \|v^k\|^2.
\]
This completes the proof. \(\square\)

**Lemma 4.3.** The positive sequence \(\{t_k\}\) generated by (4.4) with the starting point \(t_1 = 1\) satisfies \(t_k \geq \frac{L_k}{2}\) for all \(k \geq 1\).

Now we prove the improved convergence result.

**Theorem 4.1.** Let \(z\) be an arbitrary solution of (3.1). Then, the sequences \(\{x^k\}\) and \(\{y^k\}\) generated by Algorithm 4.1 satisfy
\[
F(x^k) - F(z) \leq \frac{2\eta\|A\|^2 \cdot \|x^0 - z\|^2}{(k + 1)^2}, \forall k \geq 1.
\]
\[
\tag{4.12}
\]
**Proof.** Let us define the quantities
\[
a_k := \frac{2}{L_k}t_k^2u^k, \ b_k = \|v^k\|^2, \ c := \|y^1 - z\|^2 = \|x^0 - z\|^2,
\]
where \(t_k, u^k\) and \(v^k\) are defined as in Lemma 4.2. Since \(t_1 = 1\), using the definition of \(v^k\), we have
\[
a_1 = \frac{2}{L_1}t_1u_1 = \frac{2}{L_1}u_1, \ b_1 = \|v^1\|^2 = \|x^1 - z\|^2
\]
Applying Lemma 3.3 to the points \(x = z, \ y = y^1\) with \(L = L_1\), we get
\[
F(x) - F(x^1) \geq \|x^1 - y^1\|^2 + L_1(y^1 - z, x^1 - y^1)
= \frac{L_1}{2}\{\|x^1 - z\|^2 - \|y^1 - z\|^2\}.
\]
Consequently,
\[
\frac{2}{L_1}u^1 \leq \|x^1 - z\|^2 - \|y^1 - z\|^2.
\]
That is \(a_1 + b_1 \leq c\).
Using Lemma 4.2 that $u_k := F(x_k) - F(z)$, we have

$$a_k - a_{k+1} \geq b_{k+1} - b_k.$$ 

Since $a_1 + b_1 \leq c$ is true, by Lemma 2.2, we have

$$\frac{2}{L_k} t_k^2 u_k \leq \|x^0 - z\|^2,$$

which combined with $t_k \geq \frac{k+1}{2}$ and using Lemma 3.2 yields

$$u^k \leq \frac{2\eta\|A\|^2 \cdot \|x^0 - z\|^2}{(k+1)^2}.$$ 

The desire result follows.

5. **Numerical results.** In this section, we focus on verifying the efficiency of the proposed fast iterative algorithm. We consider two problems: the first one is a general randomly generated SFP without regularization; and the last one is an $l_1$-regularized wavelet-based image deblurring problem. All codes were written by Matlab 2014b on a laptop with an Intel quad-core i7 2.4GHz CPU with 8GB of RAM running on Vista operating system.

5.1. **Random synthetic date.** We first consider the general SFP(1.1) without regularization for random synthetic data. The problem is constructed randomly as follows:

Let $A = (a_{ij})_{M \times N}$, $a_{ij} \in (0, 1)$ be a random matrix, $M, N$ be two positive integers. The set $C = \{x \in \mathbb{R}^N \mid \sum_{i=1}^N x_i^2 \leq r^2\}$; the set $Q = \{x \in \mathbb{R}^M \mid x \leq b\}$, where the vector $b$ is generated by using the following way: given a random $N$-dimensional negative vector (each component is negative) $z \in C$, $r = \|z\|$, taking $b = Az$. Find $x \in C$ and $Ax \in Q$.

To exhibit the promising performance of the fast CQ algorithm (Algorithm 4.1), we compare it with the CQ algorithm (abbreviated by "CQ") and Algorithm 3.1 (indeed, it is the Armijo-like algorithm introduced in [14]). Throughout, we set $\beta = 4, \eta = 3, \gamma = 0.3$ and take $e_0 = (0, 0, \ldots, 0)$ as the initial point in this example. Set

$$Err := \max\{\|x^k - P_C(x^k)\|,\|Ax^k - P_Q(Ax^k)\|\} \leq 10^{-6}$$

as the termination rule for the algorithms. In our experiments, we test three scenarios with dimensions $(N, M) = (10, 20), (20, 40), (50, 50)$. The numerical results can be seen from Table 1. In this table, "Iter." and "s" denote the number of iterations and cpu time in seconds, respectively.
5.2. Image deblurring. Now, we consider a wavelet-based image deblurring problem, which takes the form of

$$\min_{x} \{ \|Wx\|_1 \mid \|Ax - b\| \leq \delta \},$$

(5.1)

where $x$ represents a vectorized image; $b$ corresponds to a vectorized observed blurred image with Gaussian noise; the matrix $A = BW$ consists of a blur operator $B$ and an inverse of a three stage Haar wavelet transform $W$, and $\delta$ controls the noise level. Clearly, model (5.1) can be formulated as a special case of (1.3), that is,

$$\min_{x} \{ \sigma \|Wx\|_1 + \frac{1}{2} \|Ax - P_{Q}(Ax)\|^2 \},$$

(5.2)

where $C := \mathbb{R}^N$ and $Q := \{ y \in \mathbb{R}^N \mid \|y - b\| \leq \delta \}, \sigma > 0$. Therefore, we can employ Algorithm 3.1 and Algorithm 4.1 to solve model (5.2).

Here, we test three images: “house.png”, “boat.png”, and “pepper.png”. Through, we degrade these images through a Gaussian blur of size $9 \times 9$ and standard deviation 5, which can be done by the MATLAB script

$$[P,center] = psfGause([9,9],5)$$

and

$$T = imfilter(image,P,'symmetric').$$

Additionally, we destroy these blurred images by adding an additive zero-mean white Gaussian noise with standard deviation $10^{-3}$. We adopt the reflective boundary condition and set $\delta = 10^{-3}$, regularization parameter $\sigma = 10^{-4}$. In this experiment, we only make a comparison between Algorithm 3.1 and Algorithm 4.1 and set the parameter $\beta = 4$, $\eta = 3$, $\gamma = 0.3$ for both of the algorithms.

We define the signal-to-noise ratio (SNR) in decibel (dB) to measure the quality of a recovered image, that is

$$SNR := 20 \log_{10} \frac{\|\hat{x}\|}{\|\hat{x} - \tilde{x}\|},$$

where $\hat{x}$ and $\tilde{x}$ represent the clean and restored images, respectively. Obviously, a larger SNR value implies an image with higher quality. Furthermore, we use $\frac{\|x^k - \hat{x}^{k-1}\|}{\|x^k - \hat{x}\|} \leq 10^{-5}$ to be the stopping criterion. Below, we first report the number of iterations ($Iter.$), computing time in seconds ($s$), SNR values ($SNR$) of model (5.2) in Table 2. Clearly, the results in Table 2 also show us that Algorithm 4.1 converges faster than Algorithm 3.1. Then, we plot evolution of the SNR with respect to iterations in Fig.1 and the objective values with respect to iterations in Fig.2, which graphically show that Algorithm 4.1 improves the convergence of Algorithm 3.1.

In Fig.3, the clean images and corrupted images are listed at the top row and the recovered images by Algorithm 3.1 and Algorithm 4.1 are listed at the bottom row in Subfigs (a), (b) and (c), respectively.

From above, we can see that our algorithms perform well on image deblurring. Hence, all results support the effectiveness of the accelerated strategy.

6. Conclusions. We have consider a unified regularized SFP which is an extension of the timely regularized linear inverse problem. To handle this model, we have studied an application of the classical CQ algorithm to the resulting regularized version. Preliminary numerical results indicate that the iterative schemes of Algorithm 3.1...
Table 2. The numerical results for image deblurring

| image | Algorithm 3.1 | Algorithm 4.1 |
|-------|---------------|---------------|
| house | Iter. = 645, s = 35.6395, SNR = 24.0820 | Iter. = 357, s = 13.8825, SNR = 24.2712 |
| boat  | Iter. = 800, s = 133.7814, SNR = 21.2978 | Iter. = 530, s = 42.8210, SNR = 21.4350 |
| pepper| Iter. = 1026, s = 57.6502, SNR = 20.1239 | Iter. = 588, s = 30.8807, SNR = 20.2807 |

Figure 1. Evolutions of SNR with respect to iterations

and Algorithm 4.1 are simple and promising. Algorithm 4.1 converges more quickly than CQ algorithm and Algorithm 3.1. Its potential for analyzing and designing with other types of regularizes (such as non-convex case) and other application, as well as a more thorough computational study, are topics of our further research.

Acknowledgments. This work was supported by Natural Science Foundation of China (61503205,71572113) and Natural Science Foundation of Shanghai (17ZR1419000).

REFERENCES

[1] C. Byrne, A unified treatment of some iterative algorithm algorithms in signal processing and image reconstruction, Inverse Problems, 20 (2004), 103–120.
[2] C. Byrne, Iterative oblique projection onto convex sets and the split feasibility problem, Inverse Problems, 18 (2002), 441–453.
[3] Y. Censor, T. Bortfeld, B. Mortin and A. Trofimov, A unified approach for inversion problems in intensity-modulated radiation, Physics in Medicine and Biology, 51 (2006), 2353–2365.
Figure 2. Evolutions of objective value with respect to iterations
Figure 3. In each subgraph (a or b or c) top (from left to right): clean images and corrupted images, respectively; bottom (from left to right): Recovered images by Algorithm 3.1 and Algorithm 4.1, respectively. house.png (256 × 256); boat.png (256 × 256); pepper.png (256 × 256).
[4] Y. Censor and T. Elfving, A multiprojection algorithm using Bregman projections in a product space, Numerical Algorithms, 8 (1994), 221–239.
[5] Y. Dang and Y. Gao, The strong convergence of a KM-CQ-Like algorithm for split feasibility problem. Inverse Problems, 27 (2011), 015007, 9 pp.
[6] H. Engl, M. Hanke and A. Neubauer, Regularization of Inverse Problems, Spring Netherlands, 2000.
[7] Y. Gao, Nonsmooth Optimization (in Chinese), Science Press, Beijing, 2008.
[8] P.C. Hansen, J. G. Nagy and D. P. Olear, Deblurring Images: Matrices, Spectra and Filtering, SIAM, 2006.
[9] H. J. He, C. Ling and H. Xu, An implementable splitting algorithm for the $l_1$ norm regularized split feasibility problem, Journal of Scientific Computing, 67 (2016), 281–298.
[10] S. N. He and W. L. Zhu, A note on approximating curve with 1-norm regularization method for the split feasibility problem, Journal of Applied Mathematics, 2012 (2012), Article ID 683890, 10 pages.
[11] M. Li, Improved relaxed CQ methods for solving the split feasibility problem, Advanced Modeling and Optimization, 13 (2011), 305–317.
[12] Y. Nesterov, A method for solving a convex programming problems with convergence rate $O(1/k^2)$. Soviet Math. Dokl, 269 (1983), 543–547.
[13] J. M. Ortega and W. C. Rheinboldt, Iterative Solution of Nonlinear Equations in Several Variables, Classics in Applied Mathematics, 30, SIAM, Philadelphia, 2000.
[14] B. Qu and N. Xiu, A note on the CQ algorithm for the split feasibility problem, Inverse Problem, 21 (2005), 1655–1665.
[15] B. Recht, M. Fazel and P. Parrilo, Guaranteed minimum rank solutions of matrix equations via nuclear norm minimization, SIAM Rev, 52 (2010), 471–501.
[16] A. Sabharwal and L. Potter, Convexly constrained linear inverse problems are as and regularization, IEEE Trans. Signal Process, 46 (1998), 2345–2352.
[17] H. Xu, A variate Krasnosel’ski-Mann algorithm and the multiple-set split feasibility problem, Inverse Problems, 22 (2006), 2021–2034.
[18] H. Xu and F. Wang, Approximating curve and strong convergence of the CQ algorithm for the split feasibility problem, Journal of Inequalities and Application, 2010 (2010), Article ID102085, 13 pages.
[19] Q. Yang, The relaxed CQ algorithm solving the split feasibility problem, Inverse Problems, 20 (2004), 1261–1266.
[20] J. Zhao and Q. Yang, Several solution methods for the split feasibility problem, Inverse Problems, 21 (2005), 1791–1799.

Received June 2018; revised September 2018.

E-mail address: jgydz@163.com
E-mail address: hnlgxzh@163.com
E-mail address: gaoyan@usst.edu.cn
E-mail address: lijx@usst.edu.cn