A difference-of-convex approach for split feasibility with applications to matrix factorizations and outlier detection

Chen Chen · Ting Kei Pong · Lulin Tan

March 3, 2019

Abstract The split feasibility problem is to find an element in the intersection of a closed set $C$ and the linear preimage of another closed set $D$, assuming the projections onto $C$ and $D$ are easy to compute. This class of problems arises naturally in many contemporary applications such as compressed sensing. While the sets $C$ and $D$ are typically assumed to be convex in the literature, in this paper, we allow both sets to be possibly nonconvex. We observe that, in this setting, the split feasibility problem can be formulated as an optimization problem with a difference-of-convex objective so that standard majorization-minimization type algorithms can be applied. Here we focus on the nonmonotone proximal gradient algorithm with majorization studied in [15, Appendix A]. We show that, when this algorithm is applied to a split feasibility problem, the sequence generated clusters at a stationary point of the problem under mild assumptions. We also study local convergence property of the sequence under suitable assumptions on the closed sets involved. Finally, we perform numerical experiments to illustrate the efficiency of our approach on solving split feasibility problems that arise in completely positive matrix factorization, (uniformly) sparse matrix factorization, and outlier detection.

Ting Kei Pong’s research was supported partly by Hong Kong Research Grants Council PolyU155085/16p.
Lulin Tan’s research was supported partly by Natural Science Foundation of China (No. 11601162) and Natural Science Foundation of Guangdong Province, China (No. 2017A030310167).

Chen Chen
School of Mathematical Sciences, South China Normal University, Guangzhou 510631, China
E-mail: chenchen@m.scnu.edu.cn
Ting Kei Pong
Department of Applied Mathematics, The Hong Kong Polytechnic University, Hong Kong
E-mail: tk.pong@polyu.edu.hk
Lulin Tan
School of Mathematical Sciences, South China Normal University, Guangzhou 510631, China
E-mail: lulin_9@hotmail.com
1 Introduction

The split feasibility problem aims at finding an element common to a closed set $C$ and the linear preimage of another closed set $D$, under the assumption that the projections onto $C$ and $D$ can be computed efficiently. This problem was first introduced in [7], and has found various applications, such as compressed sensing, signal processing, image reconstruction and intensity modulated therapy; see, for example, [6, 8, 16, 26] and references therein.

Although the split feasibility problem can be seen as a special case of the classical feasibility problem that finds a point in the intersection of two closed sets, a direct application of algorithms for feasibility problems such as the alternating projection method and the Douglas-Rachford splitting method may not be desirable. This is because, in a split feasibility problem, we only assume that the projections onto $C$ and $D$ are easy to compute; in particular, it can be difficult to project onto the linear preimage of $D$, rendering a direct application of classical methods for feasibility problems inefficient. Specialized algorithms have thus been proposed for solving split feasibility problems, using only projections onto $C$ and $D$ as well as applications of the linear map and its adjoint. However, most existing work on split feasibility problems focuses on the convex settings, i.e., they assume $C$ and $D$ are also convex; see, for example [6–9, 16, 22, 24–28]. This does not cover contemporary applications that involve nonconvex constraints.

In this paper, we consider the split feasibility problem in a possibly nonconvex setting, i.e., we allow the sets $C$ and $D$ to be possibly nonconvex. We propose an algorithm for solving it and analyze its global and local convergence properties. The algorithm we propose can be viewed as a generalization of the classical CQ algorithm, which was proposed in [6] for convex split feasibility problems. Indeed, as we will discuss in more detail in Section 3, the split feasibility problem can be reformulated into a special possibly nonconvex optimization problem with a difference-of-convex objective function so that standard majorization-minimization type algorithms can be employed. Our main algorithm is an adaptation of the majorization-minimization type algorithm proposed in [15, Appendix A] to solve this special optimization problem. When $C$ and $D$ are both convex and a constant stepsize strategy is adopted, our proposed algorithm reduces to the classical CQ algorithm.

Since we are solving the split feasibility problem via solving a nonconvex optimization problem, one cannot expect to obtain a global minimizer in general. Instead, we define a new concept of stationary point for split feasibility problem in Definition 3 below and show that, under mild assumptions, any cluster point of the sequence generated by our algorithm is a stationary point of the split feasibility problem. The whole sequence generated is further shown to be convergent under additional assumptions such as the Kurdyka-Łojasiewicz property [2, 3] and Lipschitz differentiability at the limit point. Furthermore, we also analyze local convergence rate, based on the Kurdyka-Łojasiewicz exponent [2, 3] and a generalization of the concept of linearly regular intersection: the concept of linearly regular intersection was proposed in [12] for studying local convergence rate of the alternating projection method for feasibility problems.

Finally, we perform numerical experiments to illustrate the efficiency of our method for solving split feasibility problems. Specifically, we perform numerical experiments on the completely positive matrix factorization problem, the (uniformly) sparse matrix factorization problem and an outlier detection problem. In
particular, for the completely positive matrix factorization problem, we follow the approach in [11] to reformulate the completely positive matrix factorization problem into a nonconvex split feasibility problem. Our numerical results show that our method always outperforms [11, Algorithm 2] in terms of both CPU time and solution quality.

The rest of the paper is organized as follows. In Section 2, we introduce notation and some preliminary results. The mathematical formulation of the split feasibility problem and our main algorithm for solving it are described in Section 3. In Section 4, we study subsequential convergence of the sequence generated by our algorithm. The global (sequential) convergence and the local convergence rate of the sequence generated by our algorithm are studied in Section 5. Finally, in Sections 6, 7 and 8, we discuss how our algorithm can be applied to solving the completely positive matrix factorization problem, the (uniformly) sparse matrix factorization problem and an outlier detection problem, respectively, and perform numerical experiments to study the performance of our algorithm.

2 Notation and preliminaries

In this paper, we let $\mathbb{R}^n$ denote the $n$-dimensional Euclidean space. For a vector $x \in \mathbb{R}^n$, we denote its Euclidean norm, $\ell_1$ norm and $\ell_\infty$ norm by $\|x\|$, $\|x\|_1$ and $\|x\|_\infty$ respectively. We also let $B(x, r)$ denote the closed ball centered at $x$ with radius $r$, i.e., $B(x, r) = \{u \in \mathbb{R}^n : \|u - x\| \leq r\}$.

An extended real valued function $f : \mathbb{R}^n \to (-\infty, \infty]$ is said to be proper if $\text{dom } f := \{x \in \mathbb{R}^n : f(x) < \infty\} \neq \emptyset$. Such a function is said to be closed if it is lower semicontinuous. For a proper closed function $f$, the regular subdifferential and the (limiting) subdifferential of $f$ at an $\bar{x} \in \text{dom } f$ are respectively defined as [23, Definition 8.3]

$$\hat{\partial}f(\bar{x}) := \left\{v \in \mathbb{R}^n : \liminf_{x \to \bar{x}, x \neq \bar{x}} \frac{f(x) - f(\bar{x}) - \langle v, x - \bar{x} \rangle}{\|x - \bar{x}\|} \geq 0\right\},$$

and

$$\partial f(\bar{x}) := \left\{v \in \mathbb{R}^n : \exists x^t \xrightarrow{f} \bar{x} \text{ and } v^t \to v \text{ with } v^t \in \hat{\partial}f(x^t) \text{ for each } t\right\},$$

where $x^t \xrightarrow{f} \bar{x}$ means both $f(x^t) \to f(\bar{x})$ and $x^t \to \bar{x}$. By convention, we also set $\hat{\partial}f(x) = \emptyset$ if $x \notin \text{dom } f$. We let $\text{dom } \partial f$ denote the domain of subdifferential, which is defined as $\text{dom } \partial f := \{x \in \mathbb{R}^n : \partial f(x) \neq \emptyset\}$. It is known in [23, Exercise 8.8] that if $f$ is continuously differentiable at $x \in \mathbb{R}^n$, then $\partial f(x) = \{\nabla f(x)\}$. In addition, if $f$ is proper convex, then $\partial f$ coincides with the notion of subdifferential in convex analysis; see [23, Proposition 8.12].

For a nonempty closed set $C \subseteq \mathbb{R}^n$, we let $C^\infty$ denote the horizon cone of $C$, which is defined in [23, Definition 3.3] as

$$C^\infty := \{x \in \mathbb{R}^n : \exists x^t \in C \text{ and } \beta_t \downarrow 0 \text{ with } \beta_t x^t \to x\}.$$

It is known that $C$ is compact if and only if $C^\infty = \{0\}$; see [23, Theorem 3.5]. We also let $\delta_C$ denote the indicator function of $C$, which is zero in $C$ and equals infinity.
otherwise. The distance from \(x\) to \(C\) is denoted by \(d(x,C) := \inf_{u \in C} \|x - u\|\), and we use \(\text{Proj}_C(x)\) to denote the set of projections of \(x\) onto \(C\), which is defined as

\[
\text{Proj}_C(x) := \text{Argmin}_{y \in C} \|y - x\|
\]

where Argmin denotes the set of minimizers. The set of projections onto the nonempty closed set \(C\) is always nonempty, and reduces to a singleton set if \(C\) is in addition convex. The regular normal cone and the (limiting) normal cone of a nonempty closed set \(C\) at \(x \in C\) are defined by \(\hat{N}_C(x) := \partial \bar{\delta}_C(x)\) and \(N_C(x) := \partial \delta_C(x)\) respectively. These notions of normal cones are closely related to projections. Indeed, if \(x \in \text{Proj}_C(y)\), then we have from [23, Example 6.16] and [23, Proposition 6.5] that

\[
y - x \in \hat{N}_C(x) \subseteq N_C(x)
\]

(1)

Finally, following [23, Definition 6.4] and [23, Definition 7.25], we say that a nonempty closed set \(C\) is (Clarke) regular at \(x \in C\) if \(N_C(x) = \hat{N}_C(x)\), and a proper closed function \(f\) is regular at an \(x \in \text{dom } f\) if its epigraph \(\text{epi } f := \{(x,t) \in \mathbb{R}^n \times \mathbb{R} : f(x) \leq t\}\) is regular at \((x,f(x))\).

We next recall the Kurdyka-Łojasiewicz (KL) property [3]. This property has been used extensively in recent years for analyzing the rate of convergence of various first-order methods, especially in a nonconvex setting; see, for example, [2-4].

**Definition 1 (KL property)** We say that a proper closed function \(f\) satisfies the KL property at \(\bar{x} \in \text{dom } \partial f\) if there exist a neighborhood \(U\) of \(\bar{x}\), \(s \in (0, \infty]\) and a continuous concave function \(\psi : [0,s) \to \mathbb{R}^+\) with \(\psi(0) = 0\) such that:

(i) \(\psi\) is continuously differentiable on \((0,s)\) with \(\psi' > 0\);
(ii) for all \(x \in U\) with \(f(\bar{x}) < f(x) < f(\bar{x}) + s\), one has

\[
\psi'(f(x) - f(\bar{x}))d(0,\partial f(x)) \geq 1.
\]

A proper closed function \(f\) satisfying the KL property at all points in \(\text{dom } \partial f\) is called a KL function.

Functions satisfying the KL property arise naturally in many applications. In particular, it is known that any proper closed semialgebraic function is a KL function; see [3,5] for more examples. Moreover, for proper closed semialgebraic functions, the \(\psi\) in Definition 1 can actually be chosen as \(\psi(a) = ca^{1-\theta}\) for some \(c > 0\) and \(\theta \in [0,1)\); see [3, Section 4.3] and references therein. This exponent \(\theta\) is important in estimating the rate of convergence of sequences generated by various first-order methods; see, for example, [2,4,13].

**Definition 2 (KL exponent)** Let \(\theta \in [0,1)\) and \(f\) be a proper closed function. We say that \(f\) satisfies the KL property at \(\bar{x} \in \text{dom } \partial f\) with exponent \(\theta\) if there exist \(c, \epsilon > 0\) and \(s \in (0, +\infty)\) such that

\[
d(0,\partial f(x)) \geq c(f(x) - f(\bar{x}))^\theta
\]

whenever \(\|x - \bar{x}\| \leq \epsilon\) and \(0 < f(x) - f(\bar{x}) < s\). If \(f\) satisfies the KL property with exponent \(\theta\) at every \(x \in \text{dom } \partial f\), then we say that \(f\) is a KL function with exponent \(\theta\).
3 Problem statement and difference-of-convex reformulation

In this section, we give the formal mathematical definition of the split feasibility problem and describe the basic ideas leading to our solution strategy. Precisely, the split feasibility problem \cite{Bauschke2013} is stated as follows: Given a matrix $A \in \mathbb{R}^{m \times n}$, and two nonempty closed sets $C \subseteq \mathbb{R}^n$ and $D \subseteq \mathbb{R}^m$,

\begin{equation}
\text{Find } x \in \mathbb{R}^n \text{ s.t. } Ax \in D \text{ and } x \in C; \tag{2}
\end{equation}

here, we assume that an element of $\text{Proj}_C(x)$ and $\text{Proj}_D(x)$ can be computed efficiently for any given $x$. The above problem arises in various contemporary applications. For instance, the noiseless compressed sensing problem was modeled as an instance of (2) in \cite[Section 6]{Candes2005}, where $D$ is the singleton set containing the noiseless measurement and $C$ is the $\ell_1$ norm ball of suitable radius; notice that $\text{Proj}_C(x)$ and $\text{Proj}_D(x)$ can be computed efficiently for this choice of $C$ and $D$.

We would like to point out that (2) can also be viewed as a special case of the classical feasibility problem, where one attempts to find a point in the intersection of two closed sets: in this case, $A^{-1}D$ and $C$. However, classical algorithms for such a feasibility problem typically involve $\text{Proj}_{A^{-1}D}(x)$, which can be hard to compute even though $\text{Proj}_D(x)$ can be computed efficiently. Thus, specialized algorithms have been designed for solving (2), making use of only projections onto $C$ and $D$ as well as multiplications by the matrix $A$ and its transpose. Almost all algorithms proposed for solving (2) were for the convex setting, i.e., when $C$ and $D$ are both in addition convex. One classical algorithm is the so-called CQ algorithm proposed in \cite[Algorithm 1.1]{Bauschke2011}, which takes the following form: given $x^0 \in \mathbb{R}^n$ and $\gamma \in (0, \frac{2}{\lambda_{\max}(A^T A)})$, update

\begin{equation}
x^{t+1} = \text{Proj}_C \left( x^t - \gamma A^T [Ax^t - \text{Proj}_D(Ax^t)] \right). \tag{3}
\end{equation}

After the proposal of the CQ algorithm, many other algorithms for solving the split feasibility problem (2) in the convex setting have been proposed; we refer the interested readers to \cite{Deutsch2002,Reich2004} for more detail.

In this paper, we consider the split feasibility problem (2) in a possibly nonconvex setting, i.e., we allow the sets $C$ and $D$ to be possibly nonconvex. Our approach is based on a (standard) reformulation of (2) into the following optimization problem:

\begin{equation}
\min_x F(x) := \frac{1}{2} d^2(Ax, D) + \delta_C(x). \tag{4}
\end{equation}

Indeed, it is easy to see that (2) is solved if and only if (4) has an optimal solution with the optimal value being zero. Thus, in order to solve (2), it suffices to solve (4).

In the case when $C$ and $D$ are both convex, problem (4) is a convex optimization problem and the function $x \mapsto \frac{1}{2} d^2(Ax, D)$ is smooth with Lipschitz gradient whose modulus is $\lambda_{\max}(A^T A)$. Thus, one can apply first-order methods such as the proximal gradient algorithm and its variant for solving (4) efficiently; in particular, in each iteration of these algorithms, one only needs to compute the projections onto $D$ (for evaluating the gradient of $\frac{1}{2} d^2(A \cdot D)$) and $C$ as well as multiplications by $A$ and $A^T$, which can be done efficiently. Notice that the classical CQ algorithm (3) is just an application of the standard proximal gradient algorithm to (4) in the convex setting; see, for example, the introduction of \cite{Candes2005}.
However, in the general case when $C$ and $D$ can be both nonconvex, the squared distance function in problem (1) is nonsmooth in general, and the proximal gradient algorithm cannot be applied. Fortunately, it is known that the squared distance function can be written as the difference of two convex functions [1]: specifically, for any $u \in \mathbb{R}^m$, we have

$$\frac{1}{2}d^2(u, D) = \frac{1}{2}\|u\|^2 - \sup_{y \in D} \left\{ \langle u, y \rangle - \frac{1}{2}\|y\|^2 \right\}.$$ 

Now, notice that the function $u \mapsto \sup_{y \in D} \{\langle u, y \rangle - \frac{1}{2}\|y\|^2\}$, as the supremum of affine functions and being finite valued, is convex continuous. Thus, we can write $F$ in (4) as

$$F(x) = \frac{1}{2}\|Ax\|^2 + \delta_C(x) - \sup_{y \in D} \left\{ \langle Ax, y \rangle - \frac{1}{2}\|y\|^2 \right\},$$

(5)

where $h$ is quadratic, $P$ is proper closed and $g$ is convex continuous. Moreover, under a mild additional assumption, we can show in the next proposition that for any $x^0 \in C$, the set $\{x : F(x) \leq F(x^0)\}$ is bounded.

**Lemma 1** Consider the split feasibility problem (2) with $C^\infty \cap A^{-1}D^\infty = \{0\}$. Then for any $x^0 \in C$, the set $\{x : F(x) \leq F(x^0)\}$ is bounded, where $F$ is defined in (4).

**Proof** Suppose to the contrary that the set $\{x : F(x) \leq F(x^0)\}$ is unbounded. Then there exists a sequence $\{x^t\}$ such that $F(x^t) \leq F(x^0)$ for all $t$ and $\lim_{t \to \infty} \|x^t\| = +\infty$. By passing to a subsequence if necessary, we may assume without loss of generality that $\|x^t\| \neq 0$ for all $t$ and that $\frac{x^t}{\|x^t\|} \to y^*$ for some $y^*$. Then it holds that $\|y^*\| = 1$ and we also have $y^* \in C^\infty$ from the definition of the horizon cone.

Next, for each $t$, we have

$$\frac{1}{2}d^2(Ax^t, D) = F(x^t) \leq F(x^0) = \frac{1}{2}d^2(Ax^0, D).$$

Hence there exists $\{d^t\} \subseteq D$ such that for each $t$,

$$\|Ax^t - d^t\| \leq d(Ax^0, D).$$

Since $\|x^t\| \neq 0$ for all $t$, dividing both sides of the above inequality by $\|x^t\|$, we have for all $t$ that

$$\left|\frac{Ax^t}{\|x^t\|} - \frac{d^t}{\|x^t\|}\right| \leq \frac{d(Ax^0, D)}{\|x^t\|}.$$ 

Passing to the limit as $t \to \infty$ in the above inequality and noting that $\frac{x^t}{\|x^t\|} \to y^*$, we deduce further that $\frac{d^t}{\|x^t\|} \to Ay^*$. Thus, we have $Ay^* \in D^\infty$, i.e., $y^* \in A^{-1}D^\infty$, according to the definition of the horizon cone. Since it also holds that $y^* \in C^\infty$ and $\|y^*\| = 1$, we have arrived at a contradiction. This completes the proof. □
Consequently, under the additional assumption that $C^\infty \cap A^{-1}D^\infty = \{0\}$, the function $F = h + P - g$ given in \eqref{F} satisfies all the assumptions necessary for applying the so-called NPG major in \cite{[15]} Appendix A. In addition, since
\begin{equation}
A^T \text{Proj}_D(Ax) \subseteq \partial g(x),
\end{equation}
one can choose in Step 1a) of the NPG major any $\eta^t \in \text{Proj}_D(Ax^t)$ so that $\zeta^t := A^T \eta^t \in \partial g(x^t)$. Moreover, using the definition of $h$ and $P$ in \eqref{hP}, the subproblem of NPG major in \cite{[15]} Eq (45) becomes
\begin{equation}
\begin{aligned}
u & \in \text{Proj}_C \left(x^t - \frac{A^T(Ax^t - \eta^t)}{L_t}\right).
\end{aligned}
\end{equation}

Having these in mind, we are now ready to present our algorithm $\text{SpFeas}_{DC\alpha}$ as Algorithm \textbf{1} below for solving \eqref{ls}, which is basically an application of the NPG major in \cite{[15]} Appendix A to \eqref{ls}.

\begin{algorithm}[H]
\caption{$\text{SpFeas}_{DC\alpha}$ for \eqref{ls}}
\label{alg:SpFeas}
\begin{algorithmic}
\STATE \textbf{Step 0.} Choose $x^0 \in C$, $L_{\text{max}} \geq L_{\text{min}} > 0$, $\tau > 1$, $\epsilon > 0$ and an integer $M \geq 0$. Set $t = 0$.
\STATE \textbf{Step 1.} Choose any $L_t^0 \in [L_{\text{min}}, L_{\text{max}}]$ and set $L_t = L_t^0$.
\STATE \textbf{Step 1a)} Pick any $\eta^t \in \text{Proj}_D(Ax^t)$ and find
\begin{equation}
\begin{aligned}
u & \in \text{Proj}_C \left(x^t - \frac{A^T(Ax^t - \eta^t)}{L_t}\right).
\end{aligned}
\end{equation}
\STATE \textbf{Step 1b)} If
\begin{equation}
\begin{aligned}
\max_{t-M \leq i \leq t} d^2(Au, D) \leq \max_{t-M \leq i \leq t} d^2(Ax^i, D) - \epsilon \|u - x^t\|^2
\end{aligned}
\end{equation}
is satisfied, go to \textbf{Step 2}).
\STATE \textbf{Step 1c)} Set $L_t \leftarrow \tau L_t$ and go to \textbf{Step 1a}).
\STATE \textbf{Step 2.} If a termination criterion is not met, set $L_t = L_t$, $x^{t+1} = u, t = t + 1$. Go to \textbf{Step 1}.
\end{algorithmic}
\end{algorithm}

Notice that each iteration of $\text{SpFeas}_{DC\alpha}$ only involves projections onto $C$ and $D$ as well as multiplications by the matrix $A$ and its transpose, which can be performed efficiently. Moreover, when $C^\infty \cap A^{-1}D^\infty = \{0\}$, it can be shown as in \cite{[15]} Proposition 1] that the linesearch criterion in Step 1b) must be satisfied after finitely many inner iterations (independent of $t$), and as in \cite{[15]} Proposition 2] that successive changes of the sequence $\{x^t\}$ go to zero. In summary, we have the following convergence result as an immediate corollary of \cite{[15]} Proposition 1] and \cite{[15]} Proposition 2].

\begin{lemma}
Consider the split feasibility problem \eqref{ls} with $C^\infty \cap A^{-1}D^\infty = \{0\}$ and let $\{x^t\}$ and $\{L_t\}$ be generated by $\text{SpFeas}_{DC\alpha}$. Then it holds that $\sup_t L_t < \infty$ and $\lim_{t \to \infty} \|x^{t+1} - x^t\| = 0$.
\end{lemma}

Using Lemma \textbf{2} it is routine to show that, when $C^\infty \cap A^{-1}D^\infty = \{0\}$, any accumulation point $\bar{x}$ of $\{x^t\}$ generated by $\text{SpFeas}_{DC\alpha}$ satisfies
\begin{equation}
0 \in \partial h(\bar{x}) + \partial P(\bar{x}) - \partial g(\bar{x}),
\end{equation}
\footnote{The condition $C^\infty \cap A^{-1}D^\infty = \{0\}$ holds in particular when $C$ is bounded, because $\Omega^\infty = \{0\}$ when $\Omega$ is compact; see \cite{[23]} Theorem 3.5.}
where $h$, $P$ and $g$ are given in (3). However, in view of the structure of $g$, it is not trivial to completely characterize the set $\partial g$ so as to relate (3) to the original split feasibility problem (2). In the next section, we will look at another characterization of the set of accumulation points of $\{x^t\}$ that is more closely tied with (2).

4 Subsequential convergence of $\text{SpFeas}_{\text{DC}}$

In this section, we characterize the set of accumulation points of the sequence $\{x^t\}$ generated by $\text{SpFeas}_{\text{DC}}$ under the assumption $C^\infty \cap A^{-1}D^\infty = \{0\}$. We start with the following proposition, which concerns the subdifferential of $F$ in (4).

Proposition 1 Consider the split feasibility problem (2) and let $F$ be defined in (4). Then for every $x \in C$, we have

$$\partial F(x) \subseteq A^TAx - A^T\text{Proj}_D(Ax) + N_C(x).$$

(9)

If in addition $C$ is regular at some $\bar{x} \in C$ and the function $y \mapsto d_D(y)$ is regular at $A\bar{x} \in \mathbb{R}^m$, then we have

$$\partial F(\bar{x}) = A^TA\bar{x} - A^T\text{Proj}_D(A\bar{x}) + N_C(\bar{x}).$$

Remark 1 In view of [23, Example 8.53] and [23, Corollary 8.11], we see that the function $y \mapsto d_D(y)$ is regular at $A\bar{x}$ under one of the following conditions:

(i) $A\bar{x} \notin D$ and $\text{Proj}_D(A\bar{x})$ is a singleton set;
(ii) $A\bar{x} \in D$ and $D$ is regular at $A\bar{x}$.

Proof Notice from [23, Example 8.53] and [18, Theorem 1.110(ii)] that for any $y \in \mathbb{R}^m$, we have

$$\partial \left(\frac{1}{2} d^2(\cdot, D)\right)(y) = d(y, D)\partial(d(\cdot, D))(y) = y - \text{Proj}_D(y).$$

(10)

This together with [23, Corollary 10.9] and [23, Theorem 10.6] gives (9).

Now, assume in addition that $C$ is regular at some $\bar{x} \in C$ and $y \mapsto d_D(y)$ is regular at $A\bar{x} \in \mathbb{R}^m$. Then at $\bar{y} := A\bar{x}$, we have

$$\partial \left(\frac{1}{2} d^2(\cdot, D)\right)(\bar{y}) \overset{(a)}{=} d(\bar{y}, D)\partial(d(\cdot, D))(\bar{y}) \overset{(b)}{=} d(\bar{y}, D)\partial(d(\cdot, D))(\bar{y}) \overset{(c)}{\subseteq} \partial \left(\frac{1}{2} d^2(\cdot, D)\right)(\bar{y}) \subseteq \partial \left(\frac{1}{2} d^2(\cdot, D)\right)(\bar{y}).$$

where (a) follows from [18, Theorem 1.110(ii)], (b) follows from the assumption that $y \mapsto d_D(y)$ is regular at $\bar{y}$ and [23, Corollary 8.11] and (c) can be verified directly from the definition. Thus,

$$\partial \left(\frac{1}{2} d^2(\cdot, D)\right)(\bar{y}) = \partial \left(\frac{1}{2} d^2(\cdot, D)\right)(\bar{y}).$$
This together with [23] Corollary 8.11 implies that the function \( y \mapsto \frac{1}{2} d^2(y, D) \) is regular at \( \tilde{y} \). Using this together with [23] Theorem 10.6], we deduce further that the function \( x \mapsto \frac{1}{2} d^2(Ax, D) \) is regular at \( \tilde{x} \) and 
\[
\partial \left( \frac{1}{2} d^2(A \cdot, D) \right)(\tilde{x}) = A^T \partial \left( \frac{1}{2} d^2(A \cdot, D) \right)(A\tilde{x}) = A^T (A\tilde{x} - \text{Proj}_D(A\tilde{x})) ,
\]
where the last equality follows from [10]. Finally, since \( C \) is regular at \( \tilde{x} \), we have from [23] Example 7.28] that \( \delta_C \) is regular at \( \tilde{x} \). The desired conclusion now follows from [23] Corollary 10.9].

\[ \square \]

Notice that if \( x^* \) solves (2), then it also solves (4). According to [23] Theorem 10.1], we must then have \( 0 \in \partial F(x^*) \). Motivated by this observation and Proposition 1, we make the following definition.

**Definition 3 (Stationary points of (2))** For the split feasibility problem (2), we say that \( x^* \) is a stationary point of this problem if
\[
0 \in A^T Ax^* - A^T \text{Proj}_D(Ax^*) + N_C(x^*).
\]

Based on [23] Theorem 10.1] and Proposition [1] we see that if \( x^* \) solves (2) (and hence (4), then it is a stationary point of (2). Moreover, if \( C \) is regular at \( x^* \) and \( y \mapsto d_D(y) \) is regular at \( Ax^* \in \mathbb{R}^m \), then \( x^* \) being stationary for problem (2) is the same as \( x^* \) being a stationary point of the function \( F \) defined in (4), in the sense that \( 0 \in \partial F(x^*) \). Finally, in view of (1), we deduce that if \( x^* \) is a stationary point of (2) in the sense of Definition 3, then it satisfies (3) in place of \( x \). This shows that the notion of stationarity of (2) defined in Definition 3 is at least as strong as the condition (3).

We next show that the sequence generated by \( \text{SpFeas}_{\text{DC}^0} \) clusters at a stationary point of (2).

**Theorem 1 (Subsequential convergence of \( \text{SpFeas}_{\text{DC}^0} \))** Consider the split feasibility problem (2) with \( C^\infty \cap A^{-1}D^\infty = \{0\} \) and let \( \{x^j\} \) be the sequence generated by \( \text{SpFeas}_{\text{DC}^0} \). Then the following statements hold:

(i) The sequence \( \{x^j\} \) is bounded.

(ii) Any accumulation point of \( \{x^j\} \) is a stationary point of (2).

**Proof** The boundedness of \( \{x^j\} \) follows from Lemma 1 and 7.

Next, let \( x^* \) be an accumulation point of \( \{x^j\} \), which exists because the sequence is bounded. Then there exists a convergent subsequence \( \{x^{j_i}\} \) such that \( \lim_{j_i \to \infty} x^{j_i} = x^* \). Clearly, \( x^* \in C \) because \( C \) is closed. Now, using (1) and the definition of \( x^{j_i+1} \) as a projection of \( x^j - \tilde{L}_t [A^T(Ax^j - \eta^j)] \) onto \( C \), we have for each \( j \) that
\[
0 \in \tilde{L}_t (x^{j_i+1} - x^{j_i}) + A^T Ax^{j_i} - A^T \eta^{j_i} + N_C(x^{j_i+1}) ; \tag{11}
\]

moreover, \( \{\tilde{L}_t\} \) is bounded thanks to Lemma 2.

On the other hand, notice from the definition of \( \eta^{j_i} \) as projection that \( F(x^{j_i}) = \frac{1}{2} ||Ax^{j_i} - \eta^{j_i}||^2 \) for all \( j \), where \( F \) is defined in (4). Since \( F(x^{j_i}) \leq F(x^0) \) for all \( j \) according to (4) and the sequence \( \{x^j\} \) is bounded according to (i), we deduce further that \( \{\eta^{j_i}\} \) is bounded. By passing to a further subsequence if necessary,
we may assume without loss of generality that \( \eta^j \to \eta^* \) for some \( \eta^* \). Since \( \eta^j \in \text{Proj}_D(Ax^j) \) and \( D \) is closed, we have \( \eta^* \in D \). Also, we have

\[
\frac{1}{2} d^2(Ax^*, D) = \lim_{j \to \infty} \frac{1}{2} d^2(Ax^j, D) = \lim_{j \to \infty} \frac{1}{2} ||Ax^j - \eta^j||^2 = \frac{1}{2} ||Ax^* - \eta^*||^2.
\]

This shows that \( \eta^* \in \text{Proj}_D(Ax^*) \). Now, passing to the limit as \( j \to \infty \) in (11) and invoking Lemma 2 and the closedness of \( x \Rightarrow N_C(x) \) at \( x^* \in C \), we obtain

\[
0 \in A^T Ax^* - A^T \eta^* + N_C(x^*),
\]

showing that \( x^* \) is a stationary point of (2). This completes the proof. \( \square \)

The algorithm \( \text{SpFeas}_{DC} \) involves a linesearch subroutine in each iteration. We next discuss a variant that does not require any linesearch procedure, which is presented in Algorithm 2 below as \( \text{SpFeas}_{DC} \), where

\[
r_C := \begin{cases} 
2 & \text{if } C \text{ is convex}, \\
1 & \text{otherwise}. 
\end{cases}
\]

(12)

We will argue that \( \text{SpFeas}_{DC} \) is a special instance of \( \text{SpFeas}_{DC, \lambda} \).

**Algorithm 2 SpFeas\textsubscript{DC} for (2).**

**Step 0.** Choose \( x^0 \in C \) and \( L > \lambda_{\text{max}}(A^T A) \).

**Step 1.** For each \( t = 0, 1, 2, \ldots \), pick any \( \eta^t \in \text{Proj}_D(Ax^t) \) and set

\[
x^{t+1} \in \text{Proj}_C \left( x^t - \frac{A^T (Ax - \eta)}{L} \right).
\]

**Step 2.** If a termination criterion is not met, go to Step 1.

To this end, we first prove the following proposition.

**Proposition 2** Consider the split feasibility problem (2) and let \( F \) be defined as in (4). Let \( L > 0 \), \( x \in C \), \( \eta \in \text{Proj}_D(Ax) \) and set

\[
u \in \text{Proj}_C \left( x - \frac{A^T (Ax - \eta)}{L} \right),
\]

(13)

Then

\[
F(u) \leq F(x) - \frac{r_C (A^T A - \lambda_{\text{max}}(A^T A))}{2} \| u - x \|^2,
\]

(14)

where \( r_C \) is defined as in (12).

**Proof** Using the decomposition of \( F \) in (12), we have for any \( u \in C \) that

\[
F(u) = h(u) + P(u) - g(u) = \frac{1}{2} \| Au \|^2 + \delta_C(u) - g(u)
\]

\[
\leq \frac{1}{2} \| Ax \|^2 + (Ax)^T (Au - Ax) + \frac{\lambda_{\text{max}}(A^T A)}{2} \| u - x \|^2 - \eta^T (Au - Ax) - g(x)
\]

\[
= F(x) + (Ax - \eta)^T (Au - Ax) + \frac{\lambda_{\text{max}}(A^T A)}{2} \| u - x \|^2,
\]

(15)
where the inequality holds because of the Taylor’s inequality applied to \( w \mapsto \frac{1}{2} \|Aw\|^2 \) and the fact that \( A^T \eta \in \partial g(x) \) (see \((\text{11})\)).

Now, suppose that \( C \) is not convex. Then we see from \((\text{15})\) that

\[
\left\| u - x + \frac{A^T(Ax - \eta)}{L} \right\|^2 \leq \left\| \frac{A^T(Ax - \eta)}{L} \right\|^2,
\]

because \( x \in C \). Thus, \( \frac{1}{2} \| u - x \|^2 + (u - x)^T A^T (Ax - \eta) \leq 0 \). This together with \((\text{15})\) shows that \((\text{14})\) holds (with \( r_C = 2 \)) when \( C \) is nonconvex.

On the other hand, if \( C \) is convex, then the function \( \rho(y) := (A^T Ax - A^T \eta)^T (y - x) + \frac{1}{2} \| y - x \|^2 + \delta_C(y) \) is a strongly convex function with modulus \( L \). Moreover, we see from \((\text{15})\) that \( u \) is the unique minimizer of \( \rho \). Thus, we have \( \rho(x) - \rho(u) \geq \frac{L}{2} \| x - u \|^2 \), which gives

\[
L \| u - x \|^2 + (u - x)^T A^T (Ax - \eta) \leq 0.
\]

This together with \((\text{15})\) shows that \((\text{14})\) holds (with \( r_C = 2 \)) when \( C \) is convex. This completes the proof. \( \Box \)

We can now argue that \( \text{SpFeas}_{DC} \) is a special instance of \( \text{SpFeas}_{DCa} \). To this end, pick an \( x^0 \in C \) and suppose that an \( L > \frac{\lambda_{\text{max}}(A^T A)}{r_C} \) is chosen. Let \( c > 0 \) be such that

\[
L = \frac{c + \lambda_{\text{max}}(A^T A)}{r_C} > 0.
\]

If we use this \( c \) in \( \text{SpFeas}_{DCa} \), set \( L_{\text{max}} = L_{\text{min}} = L \) and fix any nonnegative integer \( M \) and any \( \tau > 1 \), then, according to Proposition\([2]\) the linesearch condition in \((\text{4})\) is always satisfied with \( L_t = L_t^0 = L \). Hence, \( \text{SpFeas}_{DC} \) initialized at \( x^0 \in C \) with the chosen \( L > \frac{\lambda_{\text{max}}(A^T A)}{r_C} \) generates the same sequence as \( \text{SpFeas}_{DCa} \) initialized at \( x^0 \in C \) with \( c \) chosen as in \((\text{16})\) and \( L_{\text{max}} = L_{\text{min}} = L \). We thus have the following immediate corollary concerning the convergence of \( \text{SpFeas}_{DC} \).

**Corollary 1 (Subsequential convergence of \( \text{SpFeas}_{DC} \))** Consider the split feasibility problem \((\text{2})\) with \( C^\infty \cap A^{-1} D^\infty = \{0\} \) and let \( \{x^t\} \) be the sequence generated by \( \text{SpFeas}_{DC} \). Then the following statements hold:

(i) The sequence \( \{x^t\} \) is bounded.

(ii) Any accumulation point of \( \{x^t\} \) is a stationary point of \((\text{2})\).

On passing, we note that \( \text{SpFeas}_{DC} \) reduces to the classical CQ algorithm \((\text{3})\) when \( C \) and \( D \) are both convex. Thus, our algorithm \( \text{SpFeas}_{DCa} \) is a generalization of the classical CQ algorithm for solving \((\text{2})\) in the general nonconvex setting.

## 5 Sequential convergence based on KL property

### 5.1 Global convergence

In this section, we establish the convergence of the whole sequence generated by \( \text{SpFeas}_{DCa} \) with \( M = 0 \) under the KL property and some mild assumptions. The KL property has been used extensively in recent years for establishing global convergence of the sequence generated by various first-order methods; see, for example, \([2, 3]\). Our proof for the next theorem follows closely the arguments in \([4]\) and is routine. We include its proof in the appendix for the ease of readers.
Theorem 2 (Global convergence of \( \text{SpFeas}_{\text{DC}_n} \) with \( M = 0 \)) Consider the split feasibility problem \( \| \) with \( C^\infty \cap A^{-1}D^\infty = \{0\} \) and let the function \( F \) in \( \| \) be a KL function. Let \( \{x^t\} \) be the sequence generated by \( \text{SpFeas}_{\text{DC}_n} \) with \( M = 0 \). Then the sequence \( \{x^t\} \) is bounded. Let \( x^* \) be an accumulation point of \( \{x^t\} \) and suppose that the function \( y \mapsto \frac{1}{2}d^2(y, D) \) is continuously differentiable at \( Ax^* \) with locally Lipschitz gradient. Then the whole sequence \( \{x^t\} \) is convergent.

Theorem 2 requires the function \( F \) in \( \| \) to have the KL property. This particular condition is not very restrictive. Indeed, according to [3, Section 4.3], the function \( F \) satisfies the KL property when \( C \) and \( D \) are both in addition semialgebraic. On the other hand, since \( D \) is possibly nonconvex, the function \( y \mapsto \frac{1}{2}d^2(y, D) \) may not be differentiable in general at \( Ax^* \), where \( x^* \) is an accumulation point of \( \{x^t\} \), as required by Theorem 2. In the next two propositions, we give simple sufficient conditions for \( y \mapsto \frac{1}{2}d^2(y, D) \) to be continuously differentiable at \( Ax^* \) with locally Lipschitz gradient. The first proposition concerns prox-regularity and is an immediate consequence of [12, Proposition 8.1]. Recall from [21, Theorem 1.3] that a closed set \( \Omega \) is said to be prox-regular at an \( \bar{x} \in \Omega \) if there exists \( \epsilon > 0 \) such that \( \text{Proj}_\Omega(x) \) is single-valued for every \( x \in B(\bar{x}, \epsilon) \).

Proposition 3 Consider the split feasibility problem \( \| \). If \( x^* \in C \cap A^{-1}D \) and \( D \) is prox-regular at \( Ax^* \), then \( y \mapsto \frac{1}{2}d^2(y, D) \) is continuously differentiable at \( Ax^* \) with locally Lipschitz gradient.

Proof The desired result follows directly from [12, Proposition 8.1]. \( \square \)

Proposition 4 Consider the split feasibility problem \( \| \) and suppose that \( D = \bigcup_{i=1}^m D_i \), where each \( D_i \), \( i = 1, \ldots, m \), is closed and convex. If \( I(Ax^*) := \{i : d(Ax^*, D_i) = d(Ax^*, D_i)\} \) is a singleton set, then \( y \mapsto \frac{1}{2}d^2(y, D) \) is continuously differentiable at \( Ax^* \) with locally Lipschitz gradient.

Proof It is easy to see that \( \frac{1}{2}d^2(y, D) = \min_{1 \leq i \leq m} \frac{1}{2}d^2(y, D_i) \) for all \( y \in \mathbb{R}^m \). Moreover, for each \( i \), the function \( y \mapsto \frac{1}{2}d^2(y, D_i) \) is continuously differentiable with Lipschitz gradient because \( D_i \) is closed and convex. Write \( y^* = Ax^* \) for notational simplicity and note that \( I(y^*) = \{i_0\} \) for some \( i_0 \in \{1, \ldots, m\} \) by assumption.

From the definition of \( I(y^*) \) we have \( \min_{i \notin I(y^*)} \frac{1}{2}d^2(y^*, D_i) > \frac{1}{2}d^2(y^*, D) \). By continuity, it then holds that for all \( y \) sufficiently close to \( y^* \), we have

\[
\min_{i \notin I(y)} \frac{1}{2}d^2(y, D_i) > \frac{1}{2}d^2(y, D).
\]

Hence \( I(y) = \{i_0\} \) for all \( y \) sufficiently close to \( y^* \). Thus, it holds that \( \frac{1}{2}d^2(y, D) = \frac{1}{2}d^2(y, D_{i_0}) \) locally around \( y^* \). Consequently, the function \( y \mapsto \frac{1}{2}d^2(y, D) \) is continuously differentiable at \( y^* \) with locally Lipschitz gradient. This completes the proof. \( \square \)

5.2 Local convergence behavior

In this section, we study the local convergence rate of the sequence \( \{x^t\} \) generated by \( \text{SpFeas}_{\text{DC}_n} \) with \( M = 0 \). Our discussion is based on the assumption that the \( F \) defined in \( \| \) is a KL function with exponent \( \theta \in [0, 1) \). The relationship between
the exponent \( \theta \) and the convergence rate of various first-order methods has been widely studied in the literature; see, for example, \([2, 3, 11]\). In particular, using a similar line of arguments as in \([4, \text{ Theorem 2}]\), one can prove the following theorem concerning local convergence rate of the sequence \( \{x^t\} \) generated by \( \text{SpFeas}_{DC} \) with \( M = 0 \). The proof is standard and is thus omitted for brevity.

**Theorem 3 (Local convergence rate)** Consider the split feasibility problem \( \{2\} \) with \( C^\infty \cap A^{-1}D^\infty = \{0\} \) and let the function \( F \) in \( \{4\} \) be a KL function with exponent \( \theta \in [0, 1) \). Let \( \{x^t\} \) be the sequence generated by \( \text{SpFeas}_{DC} \) with \( M = 0 \) and let \( x^* \) be its accumulation point. If \( y \mapsto \frac{1}{2}d^2(y, D) \) is continuously differentiable at \( Ax^* \) with locally Lipschitz gradient, then the following statements hold:

1. If \( \theta = 0 \), then there exists \( t^* \geq 0 \) such that \( x^t = x^* \) whenever \( t \geq t^* \).
2. If \( \theta \in (0, \frac{1}{2}] \), then there exist \( t^* \geq 0, \delta > 0 \) and \( \sigma \in (0, 1) \) such that \( \|x^t - x^*\| \leq \delta \sigma^t \) for all \( t \geq t^* \).
3. If \( \theta \in (\frac{1}{2}, 1) \), then there exist \( t^* \geq 0, \delta > 0 \) such that \( \|x^t - x^*\| \leq \delta t^\theta \) for all \( t \geq t^* \).

From Theorem 3, we know that if the function \( F \) in \( \{4\} \) satisfies the KL property with exponent \( \frac{1}{2} \) and a certain differentiability assumption holds at an accumulation point of \( \{x^t\} \), then the sequence \( \{x^t\} \) generated is locally linearly convergent. We next give sufficient conditions on \( C \) and \( D \) in \( \{2\} \) so that the \( F \) in \( \{4\} \) satisfies the KL property with exponent \( \frac{1}{2} \). Our first result concerns polyhedrality.

**Theorem 4** Consider the split feasibility problem \( \{2\} \) and let \( F \) be defined in \( \{4\} \). If \( C \) and \( D \) are both unions of polyhedral sets, then \( F \) is a KL function with exponent \( \frac{1}{2} \).

**Proof** Let \( D = \bigcup_{i=1}^k D_i \) and \( C = \bigcup_{j=1}^{\ell} C_j \), where \( D_1, \ldots, D_k \) and \( C_1, \ldots, C_\ell \) are all polyhedral sets. Then one can see that

\[
F(x) = \min_{1 \leq i \leq k} \min_{1 \leq j \leq \ell} \frac{1}{2} \min_{F_{i,j}(x)} d^2(Ax, D_i) + \delta_{C_j}(x).
\] (17)

Since \( D_i \) and \( C_j \) are polyhedral for each \( i \) and \( j \), we conclude from \([23, \text{ Exercise 10.22}]\) and \([23, \text{ Example 12.31}]\) that \( F_{i,j} \) is piecewise linear-quadratic for each \( i \) and \( j \). Using the definition of piecewise linear-quadratic function, we can further rewrite \( F_{i,j} \) as follows:

\[
F_{i,j}(x) = \min_{1 \leq \nu \leq l_{i,j}} \left\{ \frac{1}{2} x^T G_{i,j,\nu} x + \alpha_{i,j,\nu} x + \beta_{i,j,\nu} + \delta_{P_{i,j,\nu}}(x) \right\},
\] (18)

where \( G_{i,j,\nu} \in \mathbb{R}^{n \times n} \) is symmetric, \( \alpha_{i,j,\nu} \in \mathbb{R}^n \), \( \beta_{i,j,\nu} \in \mathbb{R} \), and \( P_{i,j,\nu} \) is polyhedral for each \( \nu \in \{1, \ldots, l_{i,j}\} \). The desired conclusion now follows from \([17, 18]\) and \([13]\) Corollary 5.2. This completes the proof.

Our next result concerns a certain kind of regularity condition, defined as follows.

**Definition 4 (Linearly regular intersection with respect to \( A \))** Consider the split feasibility problem \( \{2\} \). We say that the pair of sets \( \{C, A^{-1}D\} \) has a linearly
regular intersection with respect to \( A \) at a point \( x^* \in C \cap A^{-1}D \) if the following implication holds:

\[
A^T w^* + v^* = 0 \text{ for some } w^* \in N_D(Ax^*) \text{ and } v^* \in N_C(x^*),
\]

\[
\implies w^* = 0 \text{ and } v^* = 0.
\]

(19)

The concept of linearly regular intersection defined above for split feasibility problems is a generalization of the corresponding property for classical feasibility problems. Recall from [12, Section 2] that the pair of nonempty closed sets \( \{C_1, C_2\} \) has linearly regular intersection at a point \( x^* \in C_1 \cap C_2 \) if the following implication holds:

\[
v_1 + v_2 = 0 \text{ for some } v_1 \in N_{C_1}(x^*) \text{ and } v_2 \in N_{C_2}(x^*),
\]

\[
\implies v_1 = v_2 = 0.
\]

It was proved in [12, Theorem 5.16] that if \( \{C_1, C_2\} \) has linearly regular intersection at an \( x^* \in C_1 \cap C_2 \) and at least one of these two sets is super-regular at \( x^* \) (see [12, Definition 4.3]), then the sequence generated by the alternating projection algorithm for finding a point in \( C_1 \cap C_2 \) is locally linear convergent as long as the algorithm was initialized sufficiently close to \( x^* \). Here, we will show in Theorem 5 below that (19) has a similar implication on split feasibility problem (2), under (19), the function \( F \) defined in (1) has the KL property with exponent \( \frac{1}{2} \) at the point \( x^* \in C \cap A^{-1}D \).

We start with an auxiliary lemma.

**Lemma 3** Consider the split feasibility problem (2). Let \( x^* \in C \cap A^{-1}D \) and suppose that the pair of sets \( \{C, A^{-1}D\} \) has a linearly regular intersection with respect to \( A \) at \( x^* \), i.e., (19) holds. Then the following statements hold:

(i) There exist \( \gamma_1 > 0 \) and \( \epsilon_1 > 0 \) such that

\[
\|A^T(Ax - \eta)\| \geq \gamma_1 \|Ax - \eta\|
\]

whenever \( x \in B(x^*, \epsilon_1) \) and \( \eta \in \text{Proj}_D(AX) \).

(ii) There exist \( \gamma_2 \in [0, 1] \) and \( \epsilon_2 > 0 \) such that

\[
v^T A^T(Ax - \eta) \geq -\gamma_2 \|A^T(Ax - \eta)\| \|v\|
\]

whenever \( x \in B(x^*, \epsilon_2) \cap C, \eta \in \text{Proj}_D(AX) \) and \( v \in N_C(x) \).

**Proof** We first prove (i). Suppose to the contrary that (i) does not hold. Then there exist \( \{x^t\} \) and \( \{\eta^t\} \) satisfying \( x^t \to x^* \), \( \eta^t \in \text{Proj}_D(AX^t) \) and

\[
\|A^T(Ax^t - \eta^t)\| < \frac{1}{t} \|Ax^t - \eta^t\|
\]

for all \( t \geq 1 \). In particular, we have \( Ax^t - \eta^t \neq 0 \) for all \( t \geq 1 \). Moreover, by passing to a subsequence if necessary, we may assume without loss of generality that

\[
\frac{Ax^t - \eta^t}{\|Ax^t - \eta^t\|} \to q^*
\]

for some \( q^* \) with \( \|q^*\| = 1 \).

Now, since \( \eta^t \in \text{Proj}_D(AX^t) \) and \( Ax^t \in D \), we have for all \( t \geq 1 \) that

\[
\|\eta^t - Ax^t\| \leq \|Ax^t - Ax^*\|.
\]
This together with \( x^t \to x^* \) implies that \( \eta^t \to Ax^* \). In addition, the relation \( \eta^t \in \text{Proj}_D(Ax^t) \) together with (1) implies that for all \( t \geq 1 \),

\[
Ax^t - \eta^t \in N_D(\eta^t).
\]

Combining this with (23), the fact that \( \eta^t \to Ax^* \) and the closedness of the normal cone mapping yields \( q^t \in N_D(Ax^*) \). Next, divide both sides of (22) by \( \|Ax^t - \eta^t\| \) and pass to the limit, we obtain \( A^T q^* = 0 \). This together with \( q^* \in N_D(Ax^*) \) and (22) gives \( q^* = 0 \), which is a contradiction. This proves (i).

We now prove (ii). Suppose to the contrary that (ii) does not hold. Then there exist \( \{x^t\} \subseteq C, \{\gamma^t_2\}, \{\eta^t\} \) and \( \{\psi^t\} \) satisfying \( x^t \to x^* \), \( \gamma^t_2 \uparrow 1 \), \( \eta^t \in \text{Proj}_D(Ax^t) \), \( \psi^t \in N_C(x^t) \) and

\[
\psi^T A^T (Ax^t - \eta^t) < -\gamma^t_2 \|A^T (Ax^t - \eta^t)\| \|\psi^t\| \tag{24}
\]

for all \( t \geq 1 \). This implies in particular that \( A^T (Ax^t - \eta^t) \neq 0 \) and \( \psi^t \neq 0 \) for all \( t \geq 1 \). Also, observe from (i) that for all sufficiently large \( t \), we have

\[
\|A^T (Ax^t - \eta^t)\| \geq \gamma^t_1 \|Ax^t - \eta^t\|.
\]

Thus, by passing to subsequences if necessary, we may assume without loss of generality that

\[
\frac{Ax^t - \eta^t}{\|A^T (Ax^t - \eta^t)\|} \to p^* \text{ and } \frac{\psi^t}{\|\psi^t\|} \to q^* \tag{25}
\]

for some \( p^* \) and \( q^* \) so that \( \|A^T p^*\| = \|q^*\| = 1 \).

Next, observe from \( \eta^t \in \text{Proj}_D(Ax^t) \), \( x^t \to x^* \) and \( Ax^* \in D \) that

\[
\limsup_{t \to \infty} \|\eta^t - Ax^t\| \leq \limsup_{t \to \infty} \|Ax^t - Ax^*\| = 0.
\]

Hence, \( \eta^t \to Ax^* \). In addition, the relation \( \eta^t \in \text{Proj}_D(Ax^t) \) together with (1) shows that for all \( t \geq 1 \),

\[
Ax^t - \eta^t \in N_D(\eta^t).
\]

This together with (25), the fact that \( \eta^t \to Ax^* \) and the closedness of the normal cone mapping gives \( p^* \in N_D(Ax^*) \). Similarly, the relation \( \psi^t \in N_C(x^t) \), (24), the fact that \( x^t \to x^* \) and the closedness of normal cone mapping imply \( q^t \in N_C(x^*) \).

Now, divide both sides of (24) by \( \|A^T (Ax^t - \eta^t)\| \|\psi^t\| \) and pass to the limit, we see that \( (A^T p^*)^T q^* \leq -1 \). Hence

\[
\|A^T p^* + q^*\|^2 = \|A^T p^*\|^2 + \|q^*\|^2 + 2(A^T p^*)^T q^* = 2 + 2(A^T p^*)^T q^* \leq 0,
\]

where the second equality holds because of (25). The above display shows that \( A^T p^* + q^* = 0 \). This together with \( p^* \in N_D(Ax^*) \), \( q^* \in N_C(x^*) \) and (24) gives \( A^T p^* = q^* = 0 \), which is a contradiction. This completes the proof. \( \square \)

We are now ready to show that under (13), the function \( F \) defined in (4) has the alleged KL property.

**Theorem 5 (KL exponent under linear regularity)** Consider the split feasibility problem (4). Let \( x^* \in C \cap A^{-1}D \) and suppose that the pair of sets \( \{C, A^{-1}D\} \) has a linearly regular intersection with respect to \( A \) at \( x^* \), i.e., (13) holds. Then the function \( F \) defined in (4) has the KL property with exponent \( \frac{1}{2} \) at \( x^* \).
where $C$ was reformulated as a feasibility problem. Precisely, give $n$ a completely positive nonnegative, i.e.,

Recall that a symmetric matrix $G$ then rewrite the factorization as

In §11, the authors considered two algorithms for solving (27):

The classical alternating projection method, which can be inefficient because

In general difficult to compute;
2. the modified alternating projection algorithm (see [11 Algorithm 2]), which
only requires computing projections onto $C$ and the nonnegative orthant $\mathbb{R}_+^{n \times r}$, as well as multiplications by $B$ and its Moore-Penrose inverse $B^\dagger$. This algorithm is described in Algorithm 3 below.

It was discussed in [11, Section 5] that the modified alternating projection algorithm is more efficient empirically than the classical alternating projection algorithm for solving (27).

Algorithm 3 The modified alternating projection algorithm in [11]

Step 0. Choose $r \geq n$ and $B \in \mathbb{R}^{n \times r}$ so that $G = BB^T$. Then select a $Q^0 \in C$. Set $t = 0$.

Step 1. Compute $W^t = \text{Proj}_{\mathbb{R}^n \times r^+}(BQ^t)$ and find $Q^{t+1} \in \text{Proj}_C[B^\dagger(W^t + (I - B^\dagger B)Q^t)].$

Step 2. If a termination criterion is not met, set $t = t + 1$. Go to Step 1.

Here, we consider an alternative approach for solving (27). Indeed, one can observe immediately that (27) can be reformulated as the following split feasibility problem:

$$\text{Find } Q \in \mathbb{R}^{r \times r} \text{ s.t. } BQ \in D \text{ and } Q \in C,$$

where $D = \mathbb{R}_+^{r \times r}$ and $C$ is the set of $r \times r$ orthogonal matrices. Note that the projections onto $C$ and $D$ have closed form solutions; see, for example, [11, Lemma 4.1] for the closed form formula of $\text{Proj}_C$. Moreover, we have $C_\infty = \{0\}$ because the set of $r \times r$ orthogonal matrices is bounded. Thus, we can apply $\text{SpFeas}_{DC}$ to solving (28), and any accumulation point of the sequence generated is a stationary point of the split feasibility problem (28) according to Theorem 1.

6.1 Numerical experiments

In this section, we compare $\text{SpFeas}_{DCn}$ and the modified alternating projection algorithm (i.e., Algorithm 3) for solving (28) (or equivalently, (27)). All codes are written in Matlab, and the experiments are performed in Matlab 2015b on a 64-bit PC with an Intel(R) Core(TM) i7-4790 CPU (3.60GHz) and 32GB of RAM.

We first discuss the implementation details of the algorithms. In $\text{SpFeas}_{DCn}$, we set $M = 4$, $\tau = 2$, $c = 10^{-4}$, $L_{\max} = 10^8$ and $L_{\min} = 10^{-8}$. Moreover, we set $L_0 = 1$, and when $t \geq 1$:

$$L_t^0 = \begin{cases} \min\{\max\{\frac{\text{tr}((Y^t)^T S^t)}{\|S^t\|_F^2}, L_{\min}\}, L_{\max}\} & \text{if } \text{tr}((Y^t)^T S^t) \geq 10^{-16} \\ \min\{\max\{\frac{\bar{L}_{t-1}}{2}, L_{\min}\}, L_{\max}\} & \text{otherwise} \end{cases}$$

where $S^t = Q^t - Q^{t-1}$ and $Y^t = B^T[BQ^t - \eta^t] - B^T[BQ^{t-1} - \eta^{t-1}]$, with $\eta^t \in \text{Proj}_D(BQ^t)$ chosen in Step 1a) of $\text{SpFeas}_{DCn}$. We terminate it when iter $> 5000$

---

3 We note that in this case $D$ is convex and hence $Q \mapsto \frac{1}{2} d^2(BQ, D)$ is smooth. Our algorithm $\text{SpFeas}_{DCn}$ reduces to the standard gradient projection algorithm with nonmonotone linesearch.
or \( \min \{BQ^T\}_{ij} \geq -10^{-16} \) or \( L_t > 10^{10} \). On the other hand, for the modified alternating projection algorithm (i.e., Algorithm 3), we terminate it when \( \text{iter} > 5000 \) or \( \min \{BQ^T\}_{ij} \geq -10^{-15} \). We will describe their initializations later.

Both algorithms require a choice of \( r \geq n \) and an initial factorization \( G = BB^T \). In our experiments below, we follow the approach in [11, Section 3] to generate the \( B \). Specifically, given a completely positive matrix \( G \in \mathbb{R}^{n \times n} \), we compute the Cholesky decomposition of \( G \) such that \( G = LL^T \) for some lower triangular matrix \( L \), if successful, and set \( \bar{B} = L \). On the other hand, if the Cholesky decomposition fails, we compute the eigenvalue decomposition of \( G \) such that \( G = U\Sigma \Sigma^T U^T \) for some orthogonal matrix \( U \) and diagonal matrix \( \Sigma \), and set \( \bar{B} = U\Sigma^{rac{1}{2}} U^T \). Then we define \( B \) as follows:

\[
B := \left[ \frac{1}{\sqrt{m}} b_j, \ldots, \frac{1}{\sqrt{m}} b_j, \frac{1}{\sqrt{m}} b_{j+1}, \ldots, \frac{1}{\sqrt{m}} b_n \right], \tag{29}
\]

where \( b_j \) is the column of \( \bar{B} \) with the least number of negative entries, and \( m = r - n + 1 \).

We perform two experiments comparing \( \text{SpFeasDCls} \) and Algorithm 3. In our first experiment, we consider randomly generated completely positive matrices as in [11, Section 7.8]. We generate a random \( n \times n \) completely positive matrix \( G \) using the following MATLAB code:

\[ \text{G} = \text{abs} \left( \text{randn}(n, 2n) \right); \quad \text{G} = \text{G} \times \text{G}'; \]

We consider \((n, r) = (n, 1.5n) = (10i, 15i)\) for \( i = 1, 2, 3, 4, 10, 20, 30, 40 \). For each \( i \), we randomly generate 50 completely positive matrices \( G \) as described above. We generate \( B \) as in (29) and initialize both algorithms at \( Q^0 = I \) for solving the corresponding (28). The computational results are presented in Table 1 where we report the largest and smallest function values \( \min \{BQ^T\}, \max \{D\} \) at termination, the average number of iterations among successful instances (iter.) and the average number of iterations among failed instances (iter.). We also report the average CPU time (in seconds) among successful instances (CPU.) and the average CPU time among failed instances (CPUf). The success rate is also listed. We can see from Table 1 that \( \text{SpFeasDCls} \) significantly outperforms Algorithm 3 with \( \text{SpFeasDCls} \) being able to solve all instances and being much faster.

Next, as in [11, Section 7.6], we perform a second experiment to study the performance of the algorithms in factorizing completely positive matrices that are close to the boundary of the completely positive cone. Specifically, as in [11, Example 7.3], we consider

\[
G := \begin{bmatrix}
8 & 5 & 1 & 1 & 5 \\
5 & 8 & 5 & 1 & 1 \\
1 & 5 & 8 & 5 & 1 \\
1 & 1 & 5 & 8 & 5 \\
5 & 1 & 1 & 5 & 8
\end{bmatrix}, \quad P := \begin{bmatrix}
2 & 1 & 1 & 1 & 1 \\
1 & 2 & 1 & 1 & 1 \\
1 & 1 & 2 & 1 & 1 \\
1 & 1 & 2 & 1 & 1 \\
1 & 1 & 1 & 2 & 1
\end{bmatrix},
\]

We say that the instance is solved by the algorithm successfully if the algorithm is terminated with the desired accuracy achieved, i.e., \( \min \{BQ^T\} \geq -10^{-16} \) for \( \text{SpFeasDCls} \), and \( \min \{BQ^T\} \geq -10^{-15} \) for Algorithm 3.
and define
\[ G_\lambda = \lambda G + (1 - \lambda) P. \] (30)

We apply the two algorithms to factorizing \( G_\lambda \) with different values of \( \lambda \). Moreover, for the two algorithms, we consider random initializations: we first generate an \( r \times r \) matrix \( Q \) with i.i.d. standard Gaussian entries and then pick any \( Q^0 \in \text{Proj}(Q) \).

In our experiments below, we consider \( \lambda \) as listed in Table 2 and set \( r = 12 \). This choice of \( r \) was also used in [11, Section 7.6]. For each \( \lambda \), we consider 100 random initializations as described above, and apply the two algorithms to factorizing \( G_\lambda \) from these initial points. Our computational results are presented in Table 2, where we report the largest and smallest function values \( \left( \frac{1}{2} d^2(BQ^t, D) \right) \) at termination, the average number of iterations among successful instances (iter\(_s\)) and the average number of iterations among failed instances (iter\(_f\)). We also report the average CPU time (in seconds) among successful instances (CPU\(_s\)) as well as the average CPU time among failed instances (CPU\(_f\)). We can see from Table 2 that \text{SpFeas}DC\(_a\) again significantly outperforms Algorithm 3. Moreover, the success rates for both algorithms decrease when \( \lambda \) increases.

7 Sparse matrix factorization

Given a matrix \( G \), the sparse matrix factorization problem consists in factorizing \( G \) (approximately) as the product of several sparse matrices. This problem is closely related to deep learning, sparse encoding and dictionary learning; see [19] and references therein. In this section, we consider a special instance of the sparse matrix factorization problem. Specifically, given a positive semidefinite matrix \( G \in \mathbb{S}_+^n \)
Table 2 Comparing SpFeasDCₜₙ and Algorithm 3 on factorizing $G$ in (30).

| $\lambda$ | $r$ | success (%) | $\text{fval}_{\text{max}}$ | $\text{fval}_{\text{min}}$ | $\text{iter}_s$ | $\text{iter}_f$ | CPUₜₙ | CPUₜₜ |
|-----------|-----|-------------|-----------------|-----------------|--------------|--------------|--------|--------|
| 0.00      | 12  | 100         | 0e+00           | 0e+00           | 8            | -            | 0.0011 | -      |
| 0.20      | 12  | 100         | 0e+00           | 0e+00           | 27           | -            | 0.0022 | -      |
| 0.40      | 12  | 100         | 0e+00           | 0e+00           | 88           | -            | 0.0064 | -      |
| 0.60      | 12  | 100         | 0e+00           | 0e+00           | 336          | -            | 0.0248 | -      |
| 0.80      | 12  | 99          | 5e-01           | 0e+00           | 1212         | 5001         | 0.3575 | -      |
| 0.90      | 12  | 82          | 2e-01           | 0e+00           | 2540         | 5001         | 0.1851 | 0.3650 |
| 0.95      | 12  | 22          | 8e-02           | 0e+00           | 3063         | 5001         | 0.2218 | 0.3806 |
| 0.96      | 12  | 7           | 5e-01           | 0e+00           | 3143         | 5001         | 0.2363 | 0.3895 |
| 0.97      | 12  | 5           | 5e-01           | 0e+00           | 3099         | 5001         | 0.2294 | 0.4036 |
| 0.98      | 12  | 5           | 2e-01           | 0e+00           | 3104         | 5001         | 0.2243 | 0.4299 |
| 0.99      | 12  | 0           | 6e-07           | 5e-01           | 3149         | 5001         | 0.4853 | -      |

To solve (31), we mimic the approach described in Section 6 and reformulate it as a split feasibility problem. In detail, starting with an initial factorization $G = BB^T$ for some $B \in \mathbb{R}^{n \times n}$, one can see that (31) can be equivalently reformulated as the following split feasibility problem:

\[
\text{Find } Q \in \mathbb{R}^{n \times n} \text{ s.t. } BQ \in D \text{ and } Q \in C, \quad (32)
\]

where $C$ is the set of $n \times n$ orthogonal matrices and $D = \{ U \in \mathbb{R}^{n \times n} : \|u_i\|_0 \leq s \text{ for each } i = 1, \ldots, n \}$, with $u_i$ being the $i$th column of $U \in \mathbb{R}^{n \times n}$. It is easy to see that if $Q^*$ is a solution of (32), then $BQ^*$ solves (31).

Note that the projections onto $C$ and $D$ have closed form solutions; see [11, Lemma 4.1] and [17, Proposition 3.1] for the closed form formula of $\text{Proj}_C$ and $\text{Proj}_D$, respectively. In addition, the boundedness of $C$ implies that $C^\infty = \{0\}$. Therefore, we can employ SpFeasDCₚ and SpFeasDC to solve (32) according to the discussions in Section 5 and it follows from Theorem 1 and Corollary 1 that any
accumulation point of the sequence generated is a stationary point of the split feasibility problem \([32]\).

### 7.1 Numerical experiments

In this section, we perform numerical experiments to compare the performances of \(\text{SpFeas}_{DC}\) and \(\text{SpFeas}_{DC_{ls}}\) on solving \((32)\). All codes are written in Matlab, and the experiments are performed in Matlab 2015b on a 64-bit PC with an Intel(R) Core(TM) i7-4790 CPU (3.60GHz) and 32GB of RAM.

We first discuss the implementation details of the algorithms. In \(\text{SpFeas}_{DC_{ls}}\), we set \(M = 4\), \(\tau = 2\), \(c = 10^{-4}\), \(L_{\max} = 10^8\), \(L_{\min} = 10^{-8}\). Moreover, we set \(L_0 = 1\), and when \(t \geq 1\):

\[
L_t^0 = \begin{cases} 
\min\{\max\left(\frac{\text{tr}(Y_t^T S_t)}{\|S_t\|_F}, L_{\min}\right), L_{\max}\} & \text{if } \text{tr}(Y_t^T S_t) \geq 10^{-12} \\
\min\{\max\left(\frac{\tilde{L}_t - 1}{2}, L_{\min}\right), L_{\max}\} & \text{otherwise}
\end{cases}
\]

where \(S_t = Q_t - Q_{t-1}\) and \(Y_t = B^T[BQ_t - \eta_t] - B^T[BQ_{t-1} - \eta_{t-1}]\), with \(\eta_t \in \text{Proj}_D(BQ_t)\) chosen in Step 1a) of \(\text{SpFeas}_{DC_{ls}}\). On the other hand, for \(\text{SpFeas}_{DC}\), we set \(L_t = \lambda_{\max}(B^T B) + 10^{-4}\). We initialize both algorithms at the identity matrix. We terminate \(\text{SpFeas}_{DC_{ls}}\) when \(d(BQ_t, D) < 10^{-9}\), or \(\text{iter} > 10000\), or \(L_t > 10^{16}\), while \(\text{SpFeas}_{DC}\) is terminated when \(d(BQ_t, D) < 10^{-9}\) or \(\text{iter} > 10000\).

We compare the above algorithms on randomly generated positive semidefinite matrices that admit uniformly sparse factorizations. We first generate a random matrix \(P_0 \in \mathbb{R}^{n \times n}\) with i.i.d. standard Gaussian entries. We then project \(P_0\) onto \(D\) to obtain \(\tilde{P}_0\) and form a positive semidefinite matrix \(G\) by \(G = \tilde{P}_0 \tilde{P}_0^T\). We then use the above algorithms to solve the corresponding \((32)\) with \(B := G^{1/2}\).

In our experiment below, we set \((n, r) = (100i, 0.1j)\) for \(i = 1, 2, 3, 4\) and \(j = 6, 7, 8, 9\) and set \(s = nr\). For each \(i\) and \(j\), we generate 20 random instances as described above. We present the computational results in Table 3, where we report the value \(d(BQ_t, D)\) at termination, the number of iterations (iter), and the CPU time in seconds (CPU), averaged over the 20 random instances. One can see that \(\text{SpFeas}_{DC_{ls}}\) notably outperforms \(\text{SpFeas}_{DC}\) in terms of both CPU times and the terminating function values. Moreover, both algorithms become faster when \(s\) increases.

### 8 An outlier detection problem

In this section, we look at an outlier detection problem. Specifically, we consider the problem of finding an \(s\)-sparse solution of a linear system \(Ax = b\) where some of the \(b_i\)’s are wrongly recorded. This class of problem arises in applications such as compressed sensing, where signals may be contaminated by the so-called electromyographic noise, resulting in extreme measurements [20].

Here, we approach this problem by considering the following split feasibility problem, which assumes prior knowledge of \(s\) and the number of incorrect \(b_i\)’s:

\[
\text{Find } x \in \mathbb{R}^n \text{ s.t. } x \in C, \ Ax \in D, \tag{33}
\]
where $A \in \mathbb{R}^{m \times n}$, $b \in \mathbb{R}^m$, $C = \{x \in \mathbb{R}^n : \|x\|_0 \leq s, \|x\|_\infty \leq 10^8\}$, and $D = \{y \in \mathbb{R}^m : \|y\|_0 \leq r\} + b$, with $r$ being an upper estimate of the number of incorrect $b_i$’s (outliers).

Note that the projections onto $C$ and $D$ have closed form solutions; see, for example, [17, Proposition 3.1]. Moreover, since $C$ is bounded, we have $C^\infty = \{0\}$. Thus, we can apply $\text{SpFeas}_{\text{DC}_C}$ and $\text{SpFeas}_{\text{DC}}$ to solving (33) according to the discussions in Section 3. Moreover, any accumulation point of the sequence generated is a stationary point of the split feasibility problem (33).

### 8.1 Numerical experiments

In this section, we perform numerical experiments to study the behavior of $\text{SpFeas}_{\text{DC}_C}$ and $\text{SpFeas}_{\text{DC}}$ on the outlier detection problem (33). All codes are written in Matlab, and the experiments are performed in Matlab 2015b on a 64-bit PC with an Intel(R) Core(TM) i7-4790 CPU (3.60GHz) and 32GB of RAM.

We first discuss the implementation details of the algorithms. In $\text{SpFeas}_{\text{DC}_C}$, we set $M = 4$, $\gamma = 2$, $c = 10^{-4}$, $L_{\max} = 10^{6}$ and $L_{\min} = 10^{-8}$. We also set $L_0 = 1$, and when $t \geq 1$,

$$L_t = \begin{cases} \min\{\max\left\{\frac{t^{T}y^t}{||y^t||}, L_{\min}\right\}, L_{\max}\} & \text{if } s^T y^t \geq 10^{-12} \\ \min\{\max\left\{\frac{t-1}{2}, L_{\min}\right\}, L_{\max}\} & \text{otherwise.} \end{cases}$$

where $s^t = x^t - x^{t-1}$, $y^t = A^T [Ax^t - \eta^t] - A^T [Ax^{t-1} - \eta^{t-1}]$, with $\eta^t$ defined in Step 1a) of the algorithm. We initialize $\text{SpFeas}_{\text{DC}_C}$ at $x^0 = 0$ and terminate it when

$$\sqrt{\lambda_{\max}(A^T A)\|A(x^t - x^{t-1})\| + \tilde{L}_{t-1}\|x^t - x^{t-1}\|^2} + \|x^t - x^{t-1}\|^2 \max\{1, ||x^t||\} < 10^{-8},$$

following the discussions in [14, Section 6], this guarantees $d(0, \partial \Xi(x^t, \eta^{t-1})) < 10^{-8}\max\{1, ||x^t||\}$, where $\Xi(x, \eta) := h(x) + P(x) - \eta^T x + g^*(\eta)$, with $h$, $P$ and $g$.

### Table 3 Comparing $\text{SpFeas}_{\text{DC}_C}$ and $\text{SpFeas}_{\text{DC}}$ on solving (32) with $s = nr$.

| n   | $r$  | $\text{SpFeas}_{\text{DC}_C}$ | $\text{SpFeas}_{\text{DC}}$ |
|-----|-----|-----------------------------|-----------------------------|
|     |     | $d(BQ^t, D)$ | Iter | CPU | $d(BQ^t, D)$ | Iter | CPU |
| 100 | 0.6 | 4e-09 | 20.1 | 3e-04 | 100.01 | 24.4 |
| 100 | 0.7 | 1e-09 | 19.1 | 1e-09 | 7270 | 16.8 |
| 100 | 0.8 | 9e-10 | 1.1 | 1e-09 | 2940 | 6.8 |
| 100 | 0.9 | 7e-10 | 0.6 | 1e-09 | 1381 | 3.1 |
| 200 | 0.6 | 1e-09 | 17.8 | 3e-04 | 100.01 | 50.7 |
| 200 | 0.7 | 1e-09 | 13.6 | 1e-09 | 7526 | 64.8 |
| 200 | 0.8 | 9e-10 | 4.0 | 1e-09 | 3122 | 26.7 |
| 200 | 0.9 | 7e-10 | 2.3 | 1e-09 | 1477 | 12.5 |
| 300 | 0.6 | 1e-09 | 17.8 | 4e-04 | 100.01 | 212.7 |
| 300 | 0.7 | 1e-09 | 30.2 | 1e-09 | 7636 | 152.5 |
| 300 | 0.8 | 9e-10 | 10.8 | 1e-09 | 3168 | 63.0 |
| 300 | 0.9 | 7e-10 | 5.8 | 1e-09 | 1509 | 31.5 |
| 400 | 0.6 | 1e-09 | 32.1 | 6e-04 | 100.01 | 409.6 |
| 400 | 0.7 | 1e-09 | 49.8 | 1e-09 | 7711 | 299.6 |
| 400 | 0.8 | 9e-10 | 18.3 | 1e-09 | 3201 | 123.7 |
| 400 | 0.9 | 7e-10 | 10.1 | 1e-09 | 1528 | 58.7 |
given in (3), and \( g^* \) is the convex conjugate of \( g \). On the other hand, for \( \text{SpFeas}_{\text{DC}} \), we set \( L = \lambda_{\text{max}}(A^T A) + 10^{-4} \). We initialize this algorithm at \( x^0 = 0 \) and terminate it when
\[
\sqrt{\lambda_{\text{max}}(A^T A) \|A(x^t - x^{t-1})\|^2 + L\|x^t - x^{t-1}\|^2} < 10^{-8},
\]
or when the number of iterations reaches 3000.
We compare the above algorithms on randomly generated instances. We first generate an \( m \times n \) matrix \( A \) with i.i.d. standard Gaussian entries, and normalize it to have unit column norms. We next generate an \( s \)-sparse vector \( w \in \mathbb{R}^n \) with i.i.d. standard Gaussian entries at uniformly randomly chosen positions. We then set
\[
b_i = \begin{cases} (Aw)_i & \text{if } i \leq m-r+1 \\ (Aw)_i + 10 \cdot \text{sign} \left( \bar{n}_i - m + r \right) & \text{if } i = m - r + 1, \ldots, m. \end{cases}
\]
where \( \bar{n} \in \mathbb{R}^r \) has i.i.d. standard Gaussian entries.

In our experiment below, we set \( (n, m, s, r) \) as listed in Table 1. For each quadruple \( (n, m, s, r) \), we generate 20 random instances as described above. Our computational results are presented in Table 4 where we report the value \( d(Ax^t, D) \) at termination, the number of iterations (iter), and the CPU time in seconds (CPU), averaged over the 20 random instances. One can observe that \( \text{SpFeas}_{\text{DC}} \) significantly outperforms \( \text{SpFeas}_{\text{DCh}} \) in terms of both CPU times and the terminating function values.

### A Proof of Theorem 2

**Proof** The boundedness of \( \{x^t\} \) follows from Theorem 1(i). We now prove convergence of the whole sequence. By assumption, \( x^* \) is an accumulation point of \( \{x^t\} \) so that the function
\[
\kappa(u) := \frac{1}{2} d^2(u, D)
\]

As discussed in [13, Section 6], this termination criterion is motivated by the fact that \( \kappa \) holds if and only if \( 0 \in \partial \Xi(\bar{x}, \bar{\eta}) \) for some \( \bar{\eta} \).
is continuously differentiable at $Ax^*$ with locally Lipschitz gradient. Then we have from [23, Example 8.53], [15, Theorem 1.110(ii)] and the chain rule that
\[
\nabla (\kappa \circ A)(x^*) = A^T (Ax^* - \text{Proj}_D(Ax^*)).
\]
Using this and the fact that $x^*$ is a stationary point of the split feasibility problem (see Theorem 1(ii)), we deduce further that
\[
0 \in A^T (Ax^* - \text{Proj}_D(Ax^*)) + N_C(x^*)
= \nabla (\kappa \circ A)(x^*) + N_C(x^*) = \partial F(x^*),
\]
where the last equality follows from [23, Exercise 8.8(c)]. In particular, it holds that $\tau$ with Lipschitz modulus $\psi$ such that
\[
\nabla (\kappa \circ A)(x^*) + \psi(\partial F(x^*)),
\]
whenever $\|x - x^*\| \leq \epsilon$ and $F(x^*) < F(x) < F(x^*) + \epsilon$. Moreover, by shrinking $\epsilon$ if necessary, we may assume without loss of generality that $\nabla \kappa$ is locally Lipschitz in $\{Ax : x \in B(x^*, \epsilon)\}$ with Lipschitz modulus $\gamma$. Next, observe from (7) with $\lambda = \max(\epsilon, \gamma)$, we have
\[
\|\nabla \kappa(x^*)\| \leq \lambda < \gamma.
\]
Using this and the fact that $\nabla (\kappa \circ A)(x^*)$ is nonincreasing. Since $F$ is also non-negative, we deduce that the limit $\lim_{t \to \infty} F(x^*)$ exists. In addition, notice that $F$ is continuous in its closed domain and $x^*$ is an accumulation point of $\{x^t\}$. Thus, we conclude that
\[
\lim_{t \to \infty} F(x^t) = F(x^*).
\]
Now, if $F(x^{t_0}) = F(x^*)$ for some $t_0 \geq 0$, then we see from (3) with $M = 0$ that $\{F(x^t)\}$ is nonincreasing. Since $F$ is also non-negative, we deduce that the limit $\lim_{t \to \infty} F(x^t)$ exists. In addition, notice that $F$ is continuous in its closed domain and $x^*$ is an accumulation point of $\{x^t\}$. Thus, we conclude that
\[
\lim_{t \to \infty} F(x^t) = F(x^*).
\]
In this case, note from Lemma 2 that there exists $N_0 > 1$ so that $\|x^t - x^t\| \leq 2^{-t}$ whenever $t \geq N_0$. Also, using Lemma 3 the definition of accumulation point and the fact that $\lim_{t \to \infty} F(x^t) = F(x^*)$, there exists $N_1 \geq N_0$ so that
\[
\begin{align*}
(i) \quad & \|x^{N_1} - x^*\| \leq 2^{-N_1} \quad \text{and} \quad F(x^{N_1}) < F(x^*) + \epsilon, \\
(ii) \quad & \|x^{N_1} - x^*\| + \|x^{N_1} - x^{N_1-1}\| + C_1 \psi(F(x^{N_1}) - F(x^*)) \leq \tau,
\end{align*}
\]
where $C_1 := 2(\tau \lambda_{\text{max}}(A^T A) + \beta)$, $c$ is as in (8). $\tau$ is the Lipschitz continuity modulus of $\nabla \kappa$ on $\{Ax : x \in B(x^*, \epsilon)\}$, $\beta = \sup_t L_t$ with $L_t$ defined in Step 2 of $\text{SpFeas}_{DC^*_\kappa}$, and $\beta$ is finite according to Lemma 2.

We claim that if $t \geq N_1$ and $\|x^t - x^*\| \leq \epsilon/2$, then
\[
\begin{align}
\|x^{t+1} - x^t\| & \leq \|x^t - x^{t-1}\| + C_1 \left[ \psi(F(x^t) - F(x^*)) - \psi(F(x^{t+1}) - F(x^*)) \right].
\end{align}
\]
To this end, note that since $x^t \in B(x^*, \epsilon/2)$ and $t \geq N_1 \geq N_0$, we have $\|x^t - x^{t-1}\| \leq 2^{-t}$ and hence $\|x^{t+1} - x^t\| \leq \epsilon$. Thus, $\kappa$ is continuously differentiable at $Ax^t$ and $Ax^{t-1}$. Moreover, we see from [23, Example 8.53] and [15, Theorem 1.110(ii)] (see also [10]) that $\nabla (\kappa \circ A)(x^{t-1}) = A^T (Ax^{t-1} - \text{Proj}_D(Ax^{t-1}))$. Using this and the definition of $x^t$, we deduce that
\[
x^t \in \text{Proj}_C \left( x^{t-1} - \frac{\nabla (\kappa \circ A)(x^{t-1})}{L_{t-1}} \right).
\]
Thus, according to (3),
\[
\begin{align}
v^t := L_{t-1} (x^{t-1} - x^t) - \nabla (\kappa \circ A)(x^{t-1}) \in N_C(x^t).
\end{align}
\]
Moreover, using the definition of $v^t$, we have
\[
\begin{align}
\|v^t + \nabla (\kappa \circ A)(x^{t-1})\| & \leq \|\nabla (\kappa \circ A)(x^{t-1}) - \nabla (\kappa \circ A)(x^{t-1})\| + L_{t-1} \|x^t - x^{t-1}\|
\leq (\tau \lambda_{\text{max}}(A^T A) + \beta) \|x^t - x^{t-1}\|.
\end{align}
\]
where the second inequality holds because \( \beta = \sup_i l_i \) and \( \nabla \) is globally Lipschitz in \( \{Ax : x \in B(x^*, \epsilon)\} \) with Lipschitz modulus \( \tau \). Since \( v^t + \nabla(\kappa \circ A)(x^t) \in N_C(x^t) + \nabla(\kappa \circ A)(x^t) = \partial F(x^t) \), we obtain from (35) that

\[
d(0, \partial F(x^t)) \leq \left( \tau \lambda_{\max}(A^T A) + \beta \right) \|x^t - x^{t-1}\|.
\]

Making use of this, the concavity of \( \psi \) and (2) with \( M = 0 \), we see further that

\[
\left( \tau \lambda_{\max}(A^T A) + \beta \right) \|x^t - x^{t-1}\| \cdot \left[ \psi(F(x^t) - F(x^*)) - \psi(F(x^{t+1}) - F(x^*)) \right]
\geq d(0, \partial F(x^t)) \cdot \left[ \psi(F(x^t) - F(x^*)) - \psi(F(x^{t+1}) - F(x^*)) \right]
\geq d(0, \partial F(x^t)) \cdot \psi(F(x^t) - F(x^*)) \cdot \left[ F(x^t) - F(x^{t+1}) \right]
\geq \frac{\epsilon}{2} \|x^t - x^{t-1}\|^2,
\]

where the last inequality follows from (1) with \( M = 0, \epsilon/2 \), and the facts that \( \|x^t - x^*\| \leq \epsilon/2 \) and that \( F(x^*) < F(x^t) \leq F(x^{N_1}) < F(x^*) + \epsilon \) (since \( t \geq N_1 \)). Dividing both sides of the above inequality by \( \frac{\epsilon}{2} \), taking square root, using the relation \( \sqrt{ab} \leq \frac{a+b}{2} \) for any nonnegative numbers \( a \) and \( b \) and invoking the definition of \( C_1 \), we obtain further that

\[
\|x^{t+1} - x^t\| \leq \sqrt{\|x^t - x^{t-1}\| \cdot C_1 \left[ \psi(F(x^t) - F(x^*)) - \psi(F(x^{t+1}) - F(x^*)) \right]}
\leq \frac{1}{2} \left( \|x^t - x^{t-1}\| + C_1 \left[ \psi(F(x^t) - F(x^*)) - \psi(F(x^{t+1}) - F(x^*)) \right] \right),
\]

from which (36) follows immediately.

Next, we show by induction that \( x^t \in B(x^*, \epsilon/2) \) whenever \( t \geq N_1 \). The case \( t = N_1 \) follows from construction. Suppose that \( x^t \in B(x^*, \epsilon/2) \) whenever \( t = N_1, \ldots, N_1 + k - 1 \) for some \( k \geq 1 \). Then

\[
\|x^{N_1 + k} - x^*\| \leq \|x^{N_1} - x^*\| + \sum_{t=N_1}^{N_1+k-1} \|x^{t+1} - x^t\|
\leq \|x^{N_1} - x^*\| + \sum_{t=N_1}^{N_1+k-1} \left( \|x^t - x^{t-1}\| - \|x^{t+1} - x^t\| + C_1 \left[ \psi(F(x^t) - F(x^*)) - \psi(F(x^{t+1}) - F(x^*)) \right] \right)
\leq \|x^{N_1} - x^*\| + \|x^{N_1} - x^{N_1-1}\| + C_1 \psi(F(x^{N_1}) - F(x^*)) \leq \frac{\epsilon}{2},
\]

where (a) follows from the induction hypothesis and (36) and (b) follows from the definition of \( N_1 \). Thus, \( x^t \in B(x^*, \epsilon/2) \) whenever \( t \geq N_1 \) by induction.

Since \( x^t \in B(x^*, \epsilon/2) \) whenever \( t \geq N_1 \), we can sum both sides of (36) from \( N_1 \) to \( \infty \) and obtain

\[
\sum_{t=N_1}^{\infty} \|x^{t+1} - x^t\|
\leq \sum_{t=N_1}^{\infty} \left( \|x^t - x^{t-1}\| - \|x^{t+1} - x^t\| + C_1 \left[ \psi(F(x^t) - F(x^*)) - \psi(F(x^{t+1}) - F(x^*)) \right] \right)
\leq \|x^{N_1} - x^{N_1-1}\| + C_1 \psi(F(x^{N_1}) - F(x^*)) < \infty.
\]

Thus, the sequence \( \{x^t\} \) is Cauchy and is hence convergent. \( \Box \)
References

1. E. Asplund. Differentiability of the metric projection in finite-dimensional Euclidean space. *Proceedings of the American Mathematical Society*, 38:218–219, 1973.

2. H. Attouch and J. Bolte. On the convergence of the proximal algorithm for nonsmooth functions involving analytic features. *Mathematical Programming*, 116:5–16, 2009.

3. H. Attouch, J. Bolte, P. Redont, and A. Soubeyran. Proximal alternating minimization and projection methods for nonconvex problems: An approach based on the Kurdyka-Lojasiewicz inequality. *Mathematics of Operations Research*, 35:438–457, 2010.

4. H. Attouch, J. Bolte, and B. F. Svaiter. Convergence of descent methods for semi-algebraic and tame problems: proximal algorithms, forward-backward splitting, and regularized Gauss-Seidel methods. *Mathematical Programming*, 137:91–129, 2013.

5. J. Bolte, A. Daniilidis, and A. Lewis. The Lojasiewicz inequality for nonsmooth subanalytic functions with applications to subgradient dynamical systems. *SIAM Journal on Optimization*, 17:1205–1223, 2007.

6. C. Byrne. Iterative oblique projection onto convex sets and the split feasibility problem. *Inverse Problems*, 18:441–453, 2002.

7. Y. Censor and T. Elfving. A multiprojection algorithm using Bregman projections in a product space. *Numerical Algorithms*, 8:221–239, 1994.

8. Y. Censor, T. Elfving, N. Kopf, and T. Bortfeld. The multiple-sets split feasibility problem and its applications for inverse problems. *Inverse Problems*, 21:2071–2084, 2005.

9. Y. Censor, A. Motova, and A. Segal. Perturbed projections and subgradient projections for the multiple-sets split feasibility problem. *Journal of Mathematical Analysis and Applications*, 327:1244–1256, 2007.

10. P. J. C. Dickinson and L. Gijben. On the computational complexity of membership problems for the completely positive cone and its dual. *Computational Optimization and Applications*, 57:403–415, 2014.

11. P. Groetzner and M. Dür. A factorization method for completely positive matrices. Available at http://www.optimization-online.org/DB_HTML/2018/03/6511.html, 2018.

12. A. S. Lewis, D. R. Luke, and J. Malick. Local linear convergence for alternating and averaged nonconvex projections. *Foundations of Computational Mathematics*, 9:485–513, 2009.

13. G. López, V. Martín-Marquez, F. Wang, and H.-K. Xu. Solving the split feasibility problem without prior knowledge of matrix norms. *Inverse Problems*, 28:085004, 2012.

14. R. A. Poliquin, R. T. Rockafellar, and L. Thibault. Local differentiability of distance functions. *Transactions of the American Mathematical Society*, 352:5231–5249, 2000.

15. B. Qu and N. Xiu. A note on the CQ algorithm for the split feasibility problem. *Inverse Problems*, 21:1655–1665, 2005.

16. B. S. Mordukhovich. *Variational Analysis and Generalized Differentiation, I: Basic Theory*. Springer, 2006.

17. B. Neysabur and R. Panigrahy. Sparse matrix factorization. Available at https://arxiv.org/abs/1311.3315, 2014.

18. L. F. Polania, R. E. Carrillo, M. Blanco-Velasco, and K. E. Barner. Compressive sensing for ECG signals in the presence of electromyographic noise. In *Proceedings of the 38th Annual Northeast Bioengineering Conference*, pages 295–296, 2012.

19. R. T. Rockafellar and R. J.-B. Wets. *Variational Analysis*. Springer, 1998.

20. Y. Shehu and O. S. Iyiola. Strong convergence result for proximal split feasibility problem in Hilbert spaces. *Optimization*, 66:2275–2290, 2017.

21. Z. Wang, Q. Yang, and Y. Yang. The relaxed inexact projection methods for the split feasibility problem. *Applied Mathematics and Computation*, 217:5347–5359, 2011.
26. J. Xu, E. C. Chi, M. Yang, and K. Lange. A majorization-minimization algorithm for split feasibility problems. *Computational Optimization and Applications*, 71:795–828, 2018.

27. Q. Yang. The relaxed CQ algorithm solving the split feasibility problem. *Inverse Problems*, 20:1261–1266, 2004.

28. J. Zhao and Q. Yang. Several solution methods for the split feasibility problem. *Inverse Problems*, 21:1791–1799, 2005.