Blind Deconvolution with Non-smooth Regularization via Bregman Proximal DCAs

Shota Takahashi\textsuperscript{a}, Mirai Tanaka\textsuperscript{a,b,c}, Shiro Ikeda\textsuperscript{a,b}

\textsuperscript{a} Department of Statistical Science, The Graduate University for Advanced Studies, 10–3 Midori-cho, Tachikawa, 190–8562, Tokyo, Japan
\textsuperscript{b} Department of Statistical Inference and Mathematics, The Institute of Statistical Mathematics 10–3 Midori-cho, Tachikawa, 190–8562, Tokyo, Japan
\textsuperscript{c} Continuous Optimization Team, RIKEN Center for Advanced Intelligence Project, Nihonbashi 1-chome Mitsui Building, 15th floor, 1–4–1 Nihonbashi, Chuo-ku, 103–0027, Tokyo, Japan

Abstract

Blind deconvolution is a technique to recover an original signal without knowing a convolving filter. It is naturally formulated as a minimization of a quartic objective function under some assumption. Because its differentiable part does not have a Lipschitz continuous gradient, existing first-order methods are not theoretically supported. In this paper, we employ the Bregman-based proximal methods, whose convergence is theoretically guaranteed under the $L$-smooth adaptable ($L$-smad) property. We first reformulate the objective function as a difference of convex (DC) functions and apply the Bregman proximal DC algorithm (BPDCA). This DC decomposition satisfies the $L$-smad property. The method is extended to the BPDCA with extrapolation (BPDC Ae) for faster convergence. When our regularizer has a sufficiently simple structure, each iteration is solved in a closed-form expression, and thus our algorithms solve large-scale problems efficiently. We also provide the stability analysis of the equilibriums and demonstrate the proposed methods through numerical experiments on image deblurring. The results show that BPDC Ae successfully recovered the original image and outperformed other existing algorithms.

Keywords: Blind deconvolution, DC optimization, Bregman proximal DC algorithms,
1. Introduction

We consider the convolution of a filter $f \in \mathbb{R}^m$ and a signal $g \in \mathbb{R}^m$, given by

$$\tilde{y} = f \ast g.$$ (1)

Our goal is to recover $g$ from $\tilde{y}$ without knowing $f$. This problem is known as blind deconvolution. Blind deconvolution arises in many fields of science and engineering, such as sensor networks [2], optics [7], astronomy [9, 11], communication engineering [14], and image processing [1, 10, 13, 20].

Without any assumptions, blind deconvolution is ill-posed. A common approach is to assume that $f$ and $g$ belong to known subspaces [1]. Concretely, for known linear operators $\tilde{B} : \mathbb{R}^{d_1} \rightarrow \mathbb{R}^m$ and $\tilde{A} : \mathbb{R}^{d_2} \rightarrow \mathbb{R}^m$, we assume that there exist the true vectors $h^* \in \mathbb{R}^{d_1}$ and $x^* \in \mathbb{R}^{d_2}$ such that $f = \tilde{B}h^*$ and $g = \tilde{A}x^*$, where $d_1, d_2 < m$. Moreover, we consider blind deconvolution in the Fourier domain. Applying the DFT to both sides of (1) and letting $F \in \mathbb{C}^{m \times m}$ be the unitary discrete Fourier transform (DFT) matrix, we obtain

$$\sqrt{m}F\tilde{y} = \sqrt{m}F(f \ast g) = \sqrt{m}Ff \odot \sqrt{m}Fg = mF\tilde{B}h^* \odot F\tilde{A}x^*,$$

where the second equality holds from the convolution theorem [17] Section 4.4.2] and $\odot$ denotes the Hadamard (elementwise) product. Thus, (1) is rewritten in the Fourier domain as $y = Bh^* \odot Ax^*$, where $y := \frac{1}{\sqrt{m}}F\tilde{y}$, $B := FB$, and $A := FA$.

Now, the goal of our problem is to estimate $h^*$ and $x^*$ from $y$. To evaluate the fidelity, we consider the squared error function $F(h, x) = \frac{1}{2}[Bh \odot Ax - y]^2_2$. In addition, in order to incorporate with the image characteristics, we also consider a regularization term $G : \mathbb{R}^{d_1 + d_2} \rightarrow (-\infty, +\infty]$. It may not be differentiable. Commonly used regul-
larization terms include non-differentiable functions, such as the \( \ell_1 \) norm and the total variation. To compute \( h^c \) and \( x^c \), in this paper, we minimize the sum of these two functions and a constraint set \( \text{cl} \, C \), where \( \text{cl} \, C \) denotes the closure of a nonempty open convex set \( C \subset \mathbb{R}^{d_1 + d_2} \). That is, we consider the following optimization problem:

\[
\min_{(h, x) \in \text{cl} \, C} F(h, x) + G(h, x).
\] (2)

Note that \( F \) is a quartic function because it has a quartic term \( h_i h_j x_k x_l \) with respect to \( (h, x) \), and thus it does not have a Lipschitz continuous gradient. Hence, we cannot rely on the convergence analysis of existing first order methods, such as the fast iterative shrinkage-thresholding algorithm (FISTA) \[3\], because their convergence analysis depends on the existence of Lipschitz continuous gradients. Instead, we try to resort to Bregman-based proximal gradient algorithms \[4, 19\]. These algorithms generalize the proximal gradient method by replacing the squared Euclidean distance with the Bregman distance \( D_H \) associated with a kernel generating distance \( H \). The algorithms generate a sequence converging to a stationary point under the \( L \)-smooth adaptable (\( L \)-smad) property of \((F, H)\) defined later. For our problem, however, finding an appropriate \( H \) is difficult because of the bilinear term of \( F \).

To address the problem, we first reformulate \( F \) into a difference of convex (DC) functions and apply the Bregman proximal DC algorithm (BPDCA) \[18\]. For our DC decomposition \( F = F_1 - F_2 \), we find an \( H \) that makes \((F_1, H)\) \( L \)-smad, and hence BPDCA with corresponding \( D_H \) converges to a stationary point of (2). We also apply a variant, the BPDCA with extrapolation (BPDCAe) \[18\], which converges faster. When \( G \) has a simple structure, each iteration of BPDCA(e) is solved in a closed form. This fact implies that our algorithms solve large-scale problems efficiently. We also show the stability property of the algorithms around the equilibriums and compare them with existing algorithms. Moreover, we show a numerical example of image deblurring.

This paper is organized as follows. Section 2 shows that \( F \) is reformulated into
a DC functions, introduces the Bregman Proximal DC Algorithms, and provides the stability analysis. Section 3 illustrates numerical experiments on image deblurring. Section 4 summarizes our contributions.

**Notation.** Vectors and matrices are shown in boldface. All one vector is \(1 \in \mathbb{R}^m\) and the identity matrix is \(I_d \in \mathbb{R}^{d \times d}\). Let \(|z|\) and \(z^2\) be elementwise absolute and squared vectors for \(z \in \mathbb{C}^d\), respectively. \(\text{Re}(z), \bar{z},\) and \(z^H\) denote its real part, complex conjugate, and complex conjugate transpose, respectively. The inner product of \(z, w \in \mathbb{C}^d\) is defined by \(\langle z, w \rangle = z^H w\).

In what follows, we use the following notations. Let \(\text{int} C\) be the interior of a set \(C \subset \mathbb{R}^d\). For a proper and lower semicontinuous function \(F : \mathbb{R}^d \to (-\infty, +\infty]\), the domain \(\{z \in \mathbb{R}^d \mid F(z) < +\infty\}\) of \(F\) is denoted by \(\text{dom} F\), and the domain \(\{z \in \mathbb{R}^d \mid \partial F(z) \neq \emptyset\}\) of the limiting subdifferential of \(F\) is also denoted by \(\text{dom} \partial F\), where \(\partial F\) denotes the limiting subdifferential of \(F\) defined in [18, Definition 5]. Note that the limiting subdifferential coincides with the (classical) subdifferential of \(F\) under the convexity.

### 2. Proposed Method

#### 2.1. DC Decomposition

We first reformulate \(F\) in (2) into a DC function. Let us define \(F_1\) and \(F_2\) as follows:

\[
\begin{align*}
F_1(h, x) &= \frac{1}{4} \|Bh\|^4_4 + \frac{1}{4} \|Ax\|^4_4 + \frac{1}{2} \left( \|Bh \odot Ax\|^2_2 + \|y \odot Bh\|^2_2 + \|Ax\|^2_2 + \|y\|^2_2 \right), \\
F_2(h, x) &= \frac{1}{4} \|Bh\|^4_4 + \frac{1}{4} \|Ax\|^4_4 + \frac{1}{2} \|y \odot Bh + Ax\|^2_2.
\end{align*}
\]

\(F_2\) is convex because a composite function of a linear transform and a convex function is convex [15 Theorem 3.1.6]. By denoting \(b_j\) and \(a_j\) be the \(j\)-th column vectors of \(B^H\)}
and $A^H$, respectively, we have

$$F_1(h, x) = \frac{1}{4} \sum_{j=1}^{m} \left( |b_j^H|^2 h_j^2 + |a_j^H x|^2 \right)^2 + \frac{1}{2} \left( \| y \odot Bh \|_2^2 + \| Ax \|_2^2 + \| y \|_2^2 \right),$$

which proves the convexity of $F_1$. Since $F = F_1 - F_2$ holds, (2) is equivalent to the following DC optimization problem:

$$\min_{(h, x) \in \mathcal{C}} F_1(h, x) - F_2(h, x) + G(h, x). \quad (3)$$

### 2.2. Bregman Proximal DC Algorithms

Before showing DC algorithms, we define the kernel generating distances, the Bregman distances, and the $L$-smooth adaptable property.

**Definition 1** (Kernel generating distances [4] and Bregman distances [5]). Let $C$ be a nonempty open convex subset of $\mathbb{R}^d$. $H : \mathbb{R}^d \to (-\infty, +\infty]$ is called a kernel generating distance associated with $C$ if it meets the following conditions:

(i) $H$ is proper, lower semicontinuous, and convex, with $\text{dom } H \subset \text{cl } C$ and $\text{dom } \partial H = C$.

(ii) $H$ is $C^1$ on $\text{int } \text{dom } H = C$.

We denote the class of the kernel generating distances associated with $C$ by $\mathcal{G}(C)$. Given $H \in \mathcal{G}(C)$, the Bregman distance $D_H : \text{dom } H \times \text{int } \text{dom } H \to \mathbb{R}_+$ is defined by

$$D_H(z, w) := H(z) - H(w) - \langle \nabla H(w), z - w \rangle.$$

**Definition 2** ($L$-smooth adaptable (L-smad) [4]). Consider a pair of functions $(F, H)$ satisfying the following conditions:

(i) $H \in \mathcal{G}(C)$,

(ii) $F : \mathbb{R}^d \to (-\infty, +\infty]$ is proper, lower semicontinuous, and $C^1$ on $C = \text{int } \text{dom } H$ with $\text{dom } F \supset \text{dom } H$. 

5
The pair \((F,H)\) is called \(L\)-smooth adaptable (L-smad) on \(C\) if there exists \(L > 0\) such that \(LH - F\) and \(LH + F\) are convex on \(C\).

**Algorithm 1** \(\text{BPDCA} [18]\) for DC optimization problem (3)

**Input:** \(H \in \mathcal{G}(C), z^0 \in C\) and \(0 < \lambda < 1/L\).

for \(k = 0, 1, 2, \ldots, \) do

Compute

\[
z^{k+1} = \arg\min_{z \in \text{cl} C} \{\langle \nabla F_1(z^k) - \nabla F_2(z^k), z \rangle + G(z) + \frac{1}{\lambda} D_H(z, z^k)\}. \tag{4}
\]

**Algorithm 2** \(\text{BPDCAe} [18]\) for DC optimization problem (3)

**Input:** \(H \in \mathcal{G}(C), z^{-1} = z^0 \in C, t_{-1} = t_0 = 1, 0 < \lambda < 1/L, \rho \in [0, 1), \text{ and } N \in \mathbb{N}\).

for \(k = 0, 1, 2, \ldots, \) do

Set \(w^k = z^k + \frac{t_{k-1}}{t_k}(z^k - z^{k-1}), t_{k+1} = \frac{1 + \sqrt{1 + 4t_k^2}}{2}\).

if \(k = k'N\) with \(k' \in \mathbb{N}, w^k \notin C, \text{ or } D_H(z^k, w^k) > \rho D_H(z^{k-1}, z^k)\) then

Set \(t_{k-1} = t_k = 1\) and \(w^k = z^k\).

Compute

\[
z^{k+1} = \arg\min_{z \in \text{cl} C} \{\langle \nabla F_1(w^k) - \nabla F_2(z^k), z \rangle + G(z) + \frac{1}{\lambda} D_H(z, w^k)\}. \tag{5}
\]

To solve (3), we introduce the Bregman proximal DC algorithm (BPDCA) [18] in Algorithm [1] where we denoted \(z = (h, x)\). At each iteration of Algorithm [1] we minimize \(\langle \nabla F_1(z^k) - \nabla F_2(z^k), z \rangle + G(z) + \frac{1}{\lambda} D_H(z, z^k)\). The first and second terms of this objective function are a first-order approximation of the original objective function, that is, \(F_1(z) - F_2(z) + G(z) \simeq F_1(z^k) - F_2(z^k) + \langle \nabla F_1(z^k) - \nabla F_2(z^k), z \rangle + G(z)\). The third term is the Bregman proximity \(\frac{1}{\lambda} D_H(z, z^k)\) between \(z\) and \(z^k \in \text{cl} C\). The parameter \(\lambda\) controls
the balance of these two components. We also introduce an accelerated version of BPDCA with extrapolation (BPDCAe) [18] in Algorithm 2. We solve Subproblem (5), which is similar to Subproblem (4) in Algorithm 1, at each iteration. See [18] for more details. The function $G$ is possibly non-smooth, and when $G$ has a sufficiently simple structure, (4) and (5) are easily solved. For instance, we solve Subproblem (4) to obtain the sparse signal and filter when $G(h, x) = \theta_1 \|h\|_1 + \theta_2 \|x\|_1$ for $\theta_1, \theta_2 \geq 0$.

Let $u = S_{\lambda \theta_1}(\lambda \nabla_h f(z^k) - \nabla_h F(z^k))$ and $v = S_{\lambda \theta_2}(\lambda \nabla_x f(z^k) - \nabla_x H(z^k))$, where $S_\tau$ is the soft-thresholding operator [4] with $\tau > 0$. We can prove from [4, Proposition 5.1] that $z^{k+1} = (t^* u, -t^* v)$ solves Subproblem (4), where $t^*$ is the unique positive real root of the cubic equation $t^3 (\|u\|_2^2 + \|v\|_2^2) + t - 1 = 0$. Note that every cubic equation has a closed-form solution via Cardano’s formula. This fact indicates solution of (4) is expressed in closed form. It is also true for Subproblem (5). When $(F_1, H)$ is $L$-smad, the convergence of the algorithms to a stationary point is guaranteed [18]. In theory and practice, BPDCAe converges faster than BPDCA [18], whereas BPDCAe requires the convexity of $G$.

The following theorem provides appropriate $H$ and $L$ in use of Algorithms [1] and [2].

**Theorem 1.** Let $H$ be defined by

$$H(h, x) = \frac{1}{4} \left( \|h\|_2^2 + \|x\|_2^2 \right)^2 + \frac{1}{2} \left( \|h\|_2^2 + \|x\|_2^2 \right).$$

(6)

Then, for any $L$ satisfying

$$L \geq \sum_{j=1}^{m} \left( 3\|b_j\|_2^4 + 3\|a_j\|_2^4 + \|b_j\|_2^2 \|a_j\|_2^2 + \|y_j\|_2^2 \|b_j\|_2^2 + \|a_j\|_2^2 \right),$$

(7)

the function $LH - F_1$ is convex on $\mathbb{R}^{d_1+d_2}$.

**Proof.** We obtain the Hessian of $H$ and $F_1$ as follows:

$$\nabla^2 H(z) = \left( \|z\|_2^2 + 1 \right) I_{d_1+d_2} + 2zz^T.$$
\[ \nabla^2 F_1(h, x) = \text{Re} \begin{bmatrix} H_{11} & H_{12} \\ H_{12}^* & H_{22} \end{bmatrix}, \]

where \( z = (h, x) \in \mathbb{R}^{d_1+d_2} \) and

\[
\begin{align*}
H_{11} & := B^H \text{diag} (2|Bh|^2 + |Ax|^2 + |y|^2)B + B^H \text{diag}((Bh)^2)\overline{B}, \\
H_{12} & := B^H \text{diag}(Bh \odot Ax)\overline{A} + B^H \text{diag}(Bh \odot \overline{Ax})A, \\
H_{22} & := A^H \text{diag}(|Bh|^2 + 2|Ax|^2 + 1)A + A^H \text{diag}((Ax)^2)A.
\end{align*}
\]

Since the sum of a complex number and its conjugate is real, \( \nabla^2 F_1 \) is real. To prove the convexity of \( LH - F_1 \), it is sufficient to show that \( \langle w, \nabla^2 F_1(h, x)w \rangle \leq L \langle w, \nabla^2 H(h, x)w \rangle \) for any \( w \in \mathbb{R}^{d_1+d_2} \). Let separate \( w = (u, v) \) using \( u \in \mathbb{R}^{d_1} \) and \( v \in \mathbb{R}^{d_2} \). We obtain

\[
\langle w, \nabla^2 H(z)w \rangle = (||z||_2^2 + 1)||w||_2^2 + 2\langle z, w \rangle^2
\]

and

\[
\langle w, \nabla^2 F_1(h, x)w \rangle = \text{Re} \langle u, H_{11}u \rangle + \text{Re} \langle v, H_{22}v \rangle + 2 \text{Re} \langle u, H_{12}v \rangle.
\]

Each term of \( \langle w, \nabla^2 F_1(h, x)w \rangle \) is bounded as follows:

\[
\begin{align*}
\text{Re} \langle u, H_{11}u \rangle &= (2|Bh|^2 + |Ax|^2 + |y|^2, |Bu|^2) + \text{Re} \langle (Bu)^2, (Bh)^2 \rangle \\
&\leq \langle |Bh|^2 + |Ax|^2 + |y|^2, |Bu|^2 \rangle + 2\langle |Bh|^2, |Bu|^2 \rangle, \\
\text{Re} \langle v, H_{22}v \rangle &= \langle |Bh|^2 + 2|Ax|^2 + 1, |Av|^2 \rangle + \text{Re} \langle (Av)^2, (Ax)^2 \rangle \\
&\leq \langle |Bh|^2 + |Ax|^2 + |Av|^2 \rangle + 2\langle |Ax|^2, |Av|^2 \rangle, \\
\text{Re} \langle u, H_{12}v \rangle &= \text{Re} \langle Bu \odot Av, Bh \odot Ax \rangle + \text{Re} \langle Bu \odot \overline{Av}, Bh \odot \overline{Ax} \rangle \\
&= \text{Re} \langle Bu \odot Av, Bh \odot Ax \rangle + \text{Re} \langle Bu \odot \overline{Av}, Bh \odot \overline{Ax} \rangle \\
&= \text{Re} \sum_{j=1}^{m} \langle b_j, u \rangle \langle a_j, v \rangle \langle b_j, h \rangle \langle a_j, x \rangle + \text{Re} \sum_{j=1}^{m} \langle b_j, u \rangle \langle a_j, v \rangle \langle b_j, h \rangle \langle a_j, x \rangle \\
&\leq 2 \sum_{j=1}^{m} ||b_j||_2^2 ||a_j||_2^2 ||h||_2 ||u||_2 ||x||_2 ||v||_2,
\end{align*}
\]
where all the inequalities hold by $\text{Re}(\cdot) \leq |\cdot|$, and the last inequality holds by the Cauchy-Schwarz inequality. From the above relation, we obtain

\[
\langle Bh^2, Bu\rangle + \langle |Ax|^2, |Av|^2 \rangle + \text{Re}(a, H_{12}v)
\]

\[
\leq \sum_{j=1}^{m} (|b_j|^2||h||_2^2||a||_2^2 + ||a_j||^2||x||_2^2||v||_2^2 + 2|b_j||h||_2||a||_2||x||_2||v||_2)
\]

\[
= \sum_{j=1}^{m} (|b_j|^2||h||_2^2||a||_2^2 + ||a_j||^2||x||_2^2||v||_2^2)^2
\]

\[
\leq \sum_{j=1}^{m} (|b_j|^2 + ||a_j||^2)(||h||_2^2||a||_2^2 + ||x||^2||v||_2^2),
\]

where both inequalities hold by the Cauchy-Schwarz inequality. Thus, we obtain

\[
\langle w, \nabla^2 F_1(h, x)w \rangle
\]

\[
\leq \langle Bh^2 + |Ax|^2 + |y|^2, Bu \rangle + \langle Bh^2 + |Ax|^2 + 1, Av \rangle
\]

\[
+ 2\langle Bh^2, Bu \rangle + 2\langle |Ax|^2, Av \rangle + 2 \text{Re}(a, H_{12}v)
\]

\[
\leq \sum_{j=1}^{m} (|b_j|^2||u||_2^2(|b_j||h||_2^2 + ||a_j||^2||x||_2^2 + |y_j|^2)
\]

\[
+ ||a_j||^2||v||_2^2(|b_j||h||_2^2 + ||a_j||^2||x||_2^2 + 1)
\]

\[
+ 2(|b_j||h||_2^2 + ||a_j||^2)(||h||_2^2||a||_2^2 + ||x||_2^2||v||_2^2))
\]

\[
\leq \sum_{j=1}^{m} (3|b_j|^2 + 3||a_j||^2 + |b_j||h||_2^2||a||_2^2 + |y_j|^2||b_j||_2^2 + ||a_j||^2)(||h||_2^2 + 1)||w||_2^2
\]

\[
\leq L\langle w, \nabla^2 H(h, x)w \rangle,
\]

which proves $LH - F_1$ is convex. \qed

From Theorem 1, we obtain the following.

**Corollary 1.** Let $H_C \in \mathcal{G}(C)$ be defined by $H_C(z) = \frac{1}{4}||z||_2^2 + \frac{1}{4}||z||_2^2 + \delta_C(z)$, where $\delta_C(z)$ is the indicator function $\delta_C(z) = 0$ for $z \in C$ and $\delta_C(z) = \infty$ otherwise. For any $L$ satisfying (7), the pair $(F_1, H_C)$ is $L$-smad on $C$. 

9
Proof. Because $C$ is an open set, $H_C$ and $\nabla^2 H_C$ is the same to these of $H$ given by (6) on $C$. From the convexity of $C$ and Theorem 1, the pair $(F_1, H_C)$ is $L$-sma on $C$. □

From Corollary 1 we can immediately prove the following corollary by using Theorems 2 and 7.

Corollary 2. Let $\{z^k\}$ be a sequence generated by BPDCA(e) with $0 < \lambda L < 1$ for (3). For BPDCAe, assume $G$ is convex. Then, $\{z^k\}$ converges to a stationary point of (3).

Note that a stationary point of (3) is defined as a point $z \in C$ satisfying $-\nabla F_1(z) + \nabla F_2(z) \in \partial G(z)$. Note that under the convexity of $G$, it is theoretically guaranteed that a sequence generated by FISTA would converge to the stationary point if $F$ had a Lipschitz continuous gradient. In our problem, the convergence of FISTA is not theoretically guaranteed because $F$ does not have it. Although FISTA is not applicable, the convergent points of BPDCA(e) and FISTA share the same stationary points in theory.

Remark 1. An appropriate value of $\lambda$ can be evaluated by backtracking, i.e., decrease $\lambda$ until $F_1(z^{k+1}) - F_1(z^k) - \langle \nabla F_1(z^k), z^{k+1} - z^k \rangle \leq \frac{1}{\lambda}D_H(z^{k+1}, z^k)$ is satisfied, because, if this inequality holds, the theoretical convergence is guaranteed. For $n_k$ backtracking procedures at the $k$th iteration, we need one calculation of $\nabla F_1(z^k)$ and $n_k$ calculations of $F_1(z^{k+1})$. There calculations are sometimes expensive. Thus, we did not use the backtracking.

2.3. Stability Analysis
We show the stability analysis of BPDCA(e), FISTA, and the alternating minimization (AM) [6,12,16] around the equilibrium points, which are the fixed points of the update formula of each algorithm. Here, we define $\Phi(z) = G(z) + \langle \nabla F(z^k), z \rangle + \frac{1}{\lambda}D_H(z, z^k)$. Then, the first-order condition $0 \in \partial \Phi(z^{k+1}) = \partial G(z^{k+1}) + \nabla F(z^k)$ +
\( \frac{1}{\lambda} \left( \nabla H(z^{k+1}) - \nabla H(z^k) \right) \) is approximated as follows:

\[
0 \approx \xi^{k+1} + \nabla F(z^k) + \frac{1}{\lambda} \nabla^2 H(z^k)(z^{k+1} - z^k),
\]

where \( \xi^{k+1} \in \partial G(z^{k+1}) \). Note that \( \nabla^2 H(z^k) \) is regular. In fact, its inverse is explicitly written as

\[
\nabla^2 H(z) \approx \frac{1}{\|z\|^2 + 1} \left( I_{d_1+d_2} - \frac{2zz^T}{3\|z\|^2 + 1} \right).
\]

By multiplying it, we obtain

\[
z^{k+1} - z^k \approx -\lambda \nabla^2 H(z^k)^{-1}(\xi^{k+1} + \nabla F(z^k)),
\]

which indicates that \( z^{k+1} - z^k \) is greatly affected by \( \nabla^2 H(z^k)^{-1} \). Thus, \( H \) is important for the performance of BPDCA. For BPDCAe, this fact is also true by substituting \( w_k \) for \( z_k \).

To simplify the stability analysis of FISTA, we consider ISTA, which is FISTA without extrapolation. Setting \( H(z) = \frac{1}{2}\|z\|^2 =: H_1(z), i.e., \nabla^2 H_1(z) = I_{d_1+d_2} \), we obtain

\[
z^{k+1} - z^k \approx -\lambda(\xi^{k+1} + \nabla F(z^k)).
\]

Since \( F \) does not have a Lipschitz continuous gradient, \( \lambda \) is close to 0, i.e., \( z^k \approx z^{k+1} \). This implies that convergence of (F)ISTA is slow.

When \( G(x, h) \) is convex, [2] is convex with respect to \( h \) for fixed \( x \) and vice versa. AM is a method to update \( x \) and \( h \) alternately, which will be easily implemented and converge. The first-order conditions around the equilibrium points are

\[
0 \in \partial_h G(h^{k+1}, x^k) + \nabla_h F(h^{k+1}, x^k).
\]
\[ \partial h G(h^{k+1}, x^k) + \nabla_h F(h^k, x^k) + \nabla^2_{hh} F(h^k, x^k)(h^{k+1} - h^k), \]
\[ 0 \in \partial_x G(h^{k+1}, x^{k+1}) + \nabla_x F(h^{k+1}, x^{k+1}) \]
\[ \simeq \partial_x G(h^{k+1}, x^{k+1}) + \nabla_x F(h^{k+1}, x^k) + \nabla^2_{xx} F(h^k, x^k)(x^{k+1} - x^k), \]

where the last approximation holds from \( h^{k+1} \simeq h^k \) in \( \nabla^2_{xx} F(h^k, x^k) \). Assuming \( \nabla^2_{hh} F(h^k, x^k) \) and \( \nabla^2_{xx} F(h^k, x^k) \) are regular, for \( \zeta^{k+1}_h \in \partial_h G(h^{k+1}, x^k) \) and \( \zeta^{k+1}_x \in \partial_x G(h^{k+1}, x^{k+1}) \), we obtain

\[ h^{k+1} - h^k \simeq -\nabla^2_{hh} F(h^k, x^k) \zeta^k_h + \nabla_h F(h^k, x^k), \]
\[ x^{k+1} - x^k \simeq -\nabla^2_{xx} F(h^k, x^k) \zeta^k_x + \nabla_x F(h^{k+1}, x^k). \]

\( \nabla^2 H(z^k) \) in the approximation of BPDCA is a block matrix that contains the cross derivative terms \( \nabla^2_{hx} H(z^k) \) and \( \nabla^2_{hx} H(z^k) \), while the perturbation around the equilibrium points of AM is approximated only with \( \nabla^2_{xx} F(h^k, x^k) \) and \( \nabla^2_{hh} F(h^k, x^k) \) regardless of the cross derivatives. Thus, an equilibrium point of AM is not necessarily that of BPDCA. This implies that BPDCA is not trapped at the points where AM is stuck.

On the other hand, every equilibrium point of BPDCA is an equilibrium point of AM under the Clarke regularity as we mention below. Here, we assume that \( G \) is Clarke regular at \( z = (h, x) \in \mathbb{R}^{d_1+d_2} \). Then, it holds that \( \partial G(h, x) \subset \partial_h G(h, x) \times \partial_x G(h, x) \) [8, Proposition 2.3.15]. Hence, for an equilibrium point \( z = (h, x) \) of BPDCA, we have

\[ 0 \in \nabla F(z) + \partial G(z) + \nabla H(z) - \nabla H(z) \]
\[ \subset (\nabla_h F(h, x), \nabla_x F(h, x)) + \partial_h G(h, x) \times \partial_x G(h, x). \]

Therefore, an equilibrium point of BPDCA is also an equilibrium point of AM. They are demonstrated in numerical experiments (Figure 4).
3. Numerical Experiments

We demonstrated efficiency of our proposed method by image deblurring via solving problem (2). We set $d_1 = 2304$, $d_2 = 65536$, and $m = 262144$ and appropriately took a ground truth of $(h^*, x^*)$. Using them, we generated a blurring kernel $f = B h^*$ and an original image $g = A x^*$, where $B : \mathbb{R}^{d_1} \to \mathbb{R}^m$ is an operator reshaping $h \in \mathbb{R}^{d_1}$ into a $\sqrt{m} \times \sqrt{m}$ image and $A : \mathbb{R}^{d_2} \to \mathbb{R}^m$ is an inverse discrete Meyer wavelet transform operator. Figure 1 depicts $f$ and $g$ used in our experiments; $f$ corresponds to a diagonal blurring and $g$ approximates a natural image. We also set $C = \{(h, x) \in \mathbb{R}^{d_1} \times \mathbb{R}^{d_2} | h > 0, x > 0\}$ and $G(h, x) = \theta \|h\|_1$ with $\theta = 0.01$ (the non-smooth $\ell_1$ regularizer) because $h$ is supposed to be sparse in practice of image deblurring.

3.1. Comparison of $\ell_1$ and $\ell_2$ Regularization

We solved problem (2) corresponding to the setting above with BPDCA(e), FISTA, and AM. For BPDCA(e), we adjusted $L$ which satisfies (7) and used it as a fixed step size. Step sizes in all iterations of FISTA were obtained by backtracking. Note that the subproblems of BPDCA(e) and FISTA (without backtracking procedures) are solved in closed-form formulae, whose computational cost is almost the same. The maximum number of iterations for BPDCA(e) and FISTA was 30000, and that for AM was 3000 because the subproblems of AM were solved approximately by 10 iterations of FISTA at each iteration. For all methods, the initial points $h^0$ and $x^0$ are set to be the left

![Fig. 1. The ground truth $f$ and $g$, the blurred image $\tilde{y}$, and $B h^K$ and $A x^K$ recovered by BPDCAe.](image)
and right singular vectors corresponding to the leading singular value of \( B^H \text{diag}(y) \tilde{\mathbf{A}} \), respectively, which is proposed in [13]. The difference between the objective value at each iteration and that at the ground truth \( \log_{10}|\Psi(h^k, x^k) - \Psi^*| \) is plotted in Figure 2 in log-scale, where we denoted \( \Psi = \mathcal{F} + \mathcal{G} \) and \( \Psi^* := \Psi(h^*, x^*) \). Figure 3a shows the cosine similarity between \( h^k \) and \( h^* \) defined by \( \text{sim}(h^k, h^*) := \langle h^k, h^* \rangle / (\|h^k\|_2 \|h^*\|_2) \), and Figure 3b shows that \( \text{sim}(x^k, x^*) \). As we can see from Figures 2 and 3, BPDCAe outperformed the other algorithms. Its convergence was the fastest, and \( \Psi(h^K, x^K) \), \( \text{sim}(h^K, h^*) \), and \( \text{sim}(x^K, x^*) \) were also the best, where \( K := 30000 \) (for BPDCA(e) and FISTA) or \( K := 3000 \) (for AM). Figure 4 shows the recovered images. Figure 4c shows that there is almost no difference between \( \tilde{\mathbf{A}}x^e \) and \( \tilde{\mathbf{A}}x^k \), while \( h^k \) was not completely recovered. Figures 4c and 4f show that the sequences generated by AM converged to a different stationary point (see also Subsection 2.3).

We also solved the deblurring problem with the \( \ell_2 \) regularizer \( G(h, x) = \theta \|h\|_2^2 \) with \( \theta = 0.01 \). The comparison between the results from these two regularizers are shown in Figure 5. It shows the superiority of the non-smooth \( \ell_1 \) regularization over the \( \ell_2 \) one, which did not recover the sparse blurring kernel.
3.2. Comparisons under Several Situations

We first demonstrate the efficiency of our proposed methods is independent of the choice of the initial point. To do so, we generated the initial points \( h^0 \) and \( x^0 \) from the uniform distribution on \([0, 0.1]\). From these initial points, we generated \( \{h^k\} \) and \( \{x^k\} \) by each algorithm. Figures 6, 7a, and 7b show \( \{\log_{10}|\Psi(h^k, x^k)| - |\Psi(h^0, x^0)|\} \), \( \{\text{sim}(h^k, h^0)\} \), and \( \{\text{sim}(x^k, x^0)\} \), respectively. Figure 8 shows the recovered images, and Figure 8c shows that there is almost no difference between \( A\hat{x}^\circ \) and \( A\hat{x}^k \) at this case. As we can see from these figures, BPDCAe also outperformed the other algorithms even when \( h^0 \) and \( x^0 \) are generated from the uniform distribution.

We next demonstrate the efficiency of our proposed methods with the noisy data. Here, we consider \( \tilde{y} \) containing Poisson noise, i.e., \( \tilde{y} = f * g + n \), where \( n \in \mathbb{R}^m \) is Poisson noise (see Figure 10a). By changing noise level, i.e., the standard deviation,
we solved the image deblurring problem with each algorithm. Figure 9 shows the objective value $\Psi(h^K, x^K)$, the loss function value $F(h^K, x^K)$, and the cosine similarities $\text{sim}(h^K, h^*)$ and $\text{sim}(x^K, x^*)$, where $h^K$ and $x^K$ are recovered by each algorithm after $K = 30000$ (for BPDCA(e) and FISTA) or $K = 3000$ (for AM) iterations for each noise level. Whereas the cosine similarity $\text{sim}(h^K, h^*)$ recovered by FISTA was the best among Figure 9c, the cosine similarity $\text{sim}(x^K, x^*)$ recovered by BPDCAe was the best among Figure 9d. Figure 10 shows the recovered images when the standard deviation of $n$ is 10.6. $x^K$ recovered by BPDCAe is the best, and the objective value by BPDCAe is the smallest in Figure 9. Thus, BPDCA(e) outperformed the other algorithms with the noisy data.

Finally, we demonstrate that BPDCA(e) is effective with another blur kernel. We executed similar experiments using a Gaussian blur $f$ and another image $g$. Figures 11, 12a, and 12b show $\{\log_{10} |\Psi(h^k, x^k) - \Psi^*|\}$, $\{\text{sim}(h^k, h^*)\}$, and $\{\text{sim}(x^k, x^*)\}$, respectively, and Figure 13 shows the recovered images. The performance in the sense of the objective values and the cosine similarities of BPDCAe is almost the same as that of FISTA. The images recovered by BPDCAe, FISTA, and AM have almost no difference from Figures 13c, 13e, and 13f. Thus, depending on types of the blur kernel $f$ and the image $g$, BPDCA(e) has the same results as the other algorithms. As we saw
Fig. 6. Plots of $\log_{10}|\Psi(h^i, x^i) - \Psi^i|$ when $z^0$ is generated from the uniform distribution.

Fig. 7. Plots of the cosine similarities when $z^0$ is generated from the uniform distribution.

Fig. 8. The upper row shows $h^K$, and the lower row shows $Ax^K$ when $z^0$ is generated from the uniform distribution.
Fig. 9. Plots of the objective value, the loss function value, and the cosine similarities recovered by each algorithm when \( \tilde{y} \) contains Poisson noise.

Fig. 10. (a) the noisy data; (b–f) the upper row shows \( h^K \), and the lower row shows \( \tilde{A}x^K \) when \( \tilde{y} \) contains Poisson noise.
Fig. 11. Plots of $\log_{10} |\Psi(h^k, x^k) - \Psi'|$ at each iteration when $f$ is a Gaussian blur.

(a) Plots of $\{\text{sim}(h^k, h^\circ)\}$.

(b) Plots of $\{\text{sim}(x^k, x^\circ)\}$.

Fig. 12. Plots of the cosine similarities between the $k$th point and the ground truth when $f$ is a Gaussian blur.

here, the performance of BPDCA(e) is almost the same as or superior to that of the other algorithms.

4. Conclusion

In blind deconvolution, recovering the original signal and the filter is naturally formulated as a minimization of a quartic objective function. While existing first-order methods are not theoretically supported because the differentiable part of the objective function does not have a Lipschitz continuous gradient, our Bregman proximal DC
algorithms efficiently work for this problem. We found an appropriate DC decomposition and kernel generating distance, and we proved the convergence theoretically using them. Our numerical experiments on image deblurring demonstrated that BPDCAe outperformed other existing algorithms and successfully recovered the original image. The numerical successful of image deblurring is also because $g$ is represented sparsely in the wavelet domain.

While BPDCAe outperformed other existing algorithms, we still have several tasks. In theory, although the convergence rate of BPDCA(e) characterized by the Kurdyka-Łojasiewicz (KL) exponent has been established in [18], the actual value of the KL exponent for blind deconvolution has not been revealed. In practice, we have not discussed how to decide on better value of the regularization parameter $\theta$.

**Acknowledgements**

This work was supported by JSPS KAKENHI Grant Number JP19K15247; JSPS KAKENHI Grant Number JP20H01951.

**Declaration of Competing Interest**

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this pa-
References

[1] A. Ahmed, B. Recht, and J. Romberg. Blind deconvolution using convex programming. *IEEE Transactions on Information Theory*, 60(3):1711–1732, 2014.

[2] L. Balzano and R. Nowak. Blind calibration of sensor networks. In *Proceedings of the 6th International Conference on Information Processing in Sensor Networks*, pages 79–88, 2007.

[3] A. Beck and M. Teboulle. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM Journal on Imaging Sciences*, 2(1):183–202, 2009.

[4] J. Bolte, S. Sabach, M. Teboulle, and Y. Vaisbourd. First order methods beyond convexity and Lipschitz gradient continuity with applications to quadratic inverse problems. *SIAM Journal on Optimization*, 28(3):2131–2151, 2018.

[5] L. M. Bregman. The relaxation method of finding the common point of convex sets and its application to the solution of problems in convex programming. *USSR Computational Mathematics and Mathematical Physics*, 7(3):200–217, 1967.

[6] T. F. Chan and C. K. Wong. Convergence of the alternating minimization algorithm for blind deconvolution. *Linear Algebra and its Applications*, 316:259–285, 2000.

[7] J. Chen, R. Lin, H. Wang, J. Meng, H. Zheng, and L. Song. Blind-deconvolution optical-resolution photoacoustic microscopy in vivo. *Optics Express*, 21(6):7316–7327, 2013.

[8] F. H. Clarke. *Optimization and nonsmooth analysis*. SIAM, 1990.
[9] R. J.-L. Fétick, L. M. Mugnier, T. Fusco, and B. Neichel. Blind deconvolution in astronomy with adaptive optics: the parametric marginal approach. Monthly Notices of the Royal Astronomical Society, 496(4):4209–4220, 2020.

[10] C. Jain, A. Kumar, A. Chugh, and N. Charaya. Efficient image deblurring application using combination of blind deconvolution method and blur parameters estimation method. ECS Transactions, 107(1):3695–3704, 2022.

[11] S. M. Jefferies and J. C. Christou. Restoration of astronomical images by iterative blind deconvolution. The Astrophysical Journal, 415:862–874, 1993.

[12] D. Krishnan, T. Tay, and R. Fergus. Blind deconvolution using a normalized sparsity measure. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition 2011, pages 233–240, 2011.

[13] X. Li, S. Ling, T. Strohmer, and K. Wei. Rapid, robust, and reliable blind deconvolution via nonconvex optimization. Applied and Computational Harmonic Analysis, 47(3):893–934, 2019.

[14] J. S. Liu and R. Chen. Blind deconvolution via sequential imputations. Journal of the American Statistical Association, 90(430):567–576, 1995.

[15] Y. Nesterov. Lectures on convex optimization, volume 137. Springer, 2018.

[16] D. Perrone and P. Favaro. Total variation blind deconvolution: The devil is in the details. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition 2014, pages 2909–2916, 2014.

[17] J. G. Proakis and D. G. Manolakis. Digital signal processing: principles algorithms and applications. Pearson, 2006.

[18] S. Takahashi, M. Fukuda, and M. Tanaka. New Bregman proximal type algorithms for solving DC optimization problems. arXiv preprint, arXiv:2105.04873, 2021.
[19] Z. Wu, C. Li, M. Li, and A. Lim. Inertial proximal gradient methods with bregman regularization for a class of nonconvex optimization problems. *Journal of Global Optimization*, 79(3):617–644, 2021.

[20] N. Zhao, Q. Wei, A. Basarab, D. Kouamé, and J.-Y. Tourneret. Blind deconvolution of medical ultrasound images using a parametric model for the point spread function. In *Proceedings of the IEEE International Ultrasonics Symposium 2016*, pages 1–4, 2016.