



Minimizing Compositions of Functions Using Proximity Algorithms with Application in Image Deblurring

Feishe Chen¹, Lixin Shen^{1*}, Bruce W. Suter² and Yuesheng Xu¹

¹ Department of Mathematics, Syracuse University, Syracuse, NY, USA, ² Air Force Research Laboratory, Rome, NY, USA

OPEN ACCESS

Edited by:

Yuan Yao,
Peking University, China

Reviewed by:

Alfio Borzie,
University of Würzburg, Germany
Xiaoqun Zhang,
Shanghai Jiao Tong University, China
Ming Yan,
Michigan State University, USA
Xueying Zeng,
Ocean University of China, China

*Correspondence:

Lixin Shen
lshen03@syr.edu

Specialty section:

This article was submitted to
Optimization,
a section of the journal
Frontiers in Applied Mathematics and
Statistics

Received: 03 June 2016

Accepted: 22 August 2016

Published: 22 September 2016

Citation:

Chen F, Shen L, Suter BW and Xu Y
(2016) Minimizing Compositions of
Functions Using Proximity Algorithms
with Application in Image Deblurring.
Front. Appl. Math. Stat. 2:12.
doi: 10.3389/fams.2016.00012

We consider minimization of functions that are compositions of functions having closed-form proximity operators with linear transforms. A wide range of image processing problems including image deblurring can be formulated in this way. We develop proximity algorithms based on the fixed point characterization of the solution to the minimization problems. We further refine the proposed algorithms when the outer functions of the composed objective functions are separable. The convergence analysis of the developed algorithms is established. Numerical experiments in comparison with the well-known Chambolle-Pock algorithm and Zhang-Burger-Osher scheme for image deblurring are given to demonstrate that the proposed algorithms are efficient and robust.

Keywords: proximity operator, deblurring, primal-dual algorithm, ADMM, Gauss-Seidel method

1. INTRODUCTION

In this paper, we study minimization problems of the form

$$\min\{f_1(A_1x) + f_2(A_2x) : x \in \mathbb{R}^n\}, \quad (1)$$

where A_i are $m_i \times n$ matrices and the functions $f_i : \mathbb{R}^{m_i} \rightarrow (-\infty, +\infty]$ are proper, lower semi-continuous and convex, for $i = 1$ and 2 . We assume that the proximity operators of f_i , $i = 1, 2$ have closed-form or can be efficiently computed numerically. The formulation (1) admits a wide variety of applications including image deblurring, machine learning, and compressive sensing. A large family of instances of Equation (1) arises in the area of regularized minimization, where $f_1(A_1x)$ serves as a fidelity term while $f_2(A_2x)$ serves as a regularization term. Concrete examples of Equation (1) in the context of image processing will be given later. Hence, it is practically important to develop efficient numerical algorithms for solving model (1).

The formulation (1), which can be noticed after reformulation, is intrinsically a composite minimization problem. Set $m = m_1 + m_2$. Indeed, by defining a mapping $A : \mathbb{R}^n \rightarrow \mathbb{R}^m$ at $x \in \mathbb{R}^n$ by $Ax = (A_1x, A_2x) \in \mathbb{R}^{m_1} \times \mathbb{R}^{m_2}$ and a function $f : \mathbb{R}^m \rightarrow (-\infty, +\infty]$ at $w = (u, v) \in \mathbb{R}^{m_1} \times \mathbb{R}^{m_2}$ by

$$f(w) = f_1(u) + f_2(v), \quad (2)$$

we are able to rewrite the formulation (1) as the following composite minimization problem

$$\min\{f(Ax) : x \in \mathbb{R}^n\}. \quad (3)$$

We say a function $f: \mathbb{R}^m \rightarrow (-\infty, +\infty]$ is block separable if this function f takes a form of Equation (2).

An efficient scheme proposed in Zhang et al. [1] can be applied to solve Equation (3) by introducing an auxiliary variable and converting it into a linearly constrained minimization problem. Introducing variable $w = Ax$ yields an equivalent problem of Equation (3):

$$\min\{f(w) : Ax - w = 0, x \in \mathbb{R}^n, w \in \mathbb{R}^m\}. \quad (4)$$

In general, the aforementioned scheme in Zhang et al. [1] solves

$$\min\{f(w) + g(x) : Ax - w = 0, x \in \mathbb{R}^n, w \in \mathbb{R}^m\}. \quad (5)$$

via

$$\begin{cases} w^{k+1} = \operatorname{argmin} \left\{ f(w) - \langle \lambda^k, Ax^k - w \rangle + \frac{\beta}{2} \|Ax^k - w\|_2^2 \right. \\ \left. + \frac{1}{2} \|w - w^k\|_{Q_1}^2 : w \in \mathbb{R}^m \right\} \\ x^{k+1} = \operatorname{argmin} \left\{ g(x) - \langle \lambda^k, Ax - w^{k+1} \rangle + \frac{\beta}{2} \|Ax \right. \\ \left. - w^{k+1}\|_{\frac{1}{2}Q_2}^2 \right\} \\ \lambda^{k+1} = \lambda^k - \gamma(Ax^{k+1} - w^{k+1}) \end{cases} \quad (6)$$

where Q_1 is positive definite, Q_2 is semi-positive definite and $\beta, \gamma > 0$. The scheme (6) has explicit form if the Q_1, Q_2 are chosen appropriately and the proximity operators of f, g have closed form. This feature of scheme (6) makes it efficiently implemented. Recently, some similar algorithms are reported in Chen et al. [2] and Deng and Yin [3].

However, applying Equation (6) directly to solve Equation (4) and therefore solve Equation (1) still have drawbacks. As a matter of fact, the variable w embeds two parts, say $w = (u; v)$, with $A_1x - u = 0, A_2x - v = 0$. Under the choice of Q_1 such that w^{k+1} in the first step of Equation (6) has explicit form, it yields two mutually independent, parallel steps in the computation of u^{k+1} and v^{k+1} . In another word, the new results of u^{k+1} has not been used in the computation of v^{k+1} if u^{k+1} is computed ahead of v^{k+1} . A relatively strict condition is imposed on the parameters brought into in the matrices Q_1, Q_2 . These drawbacks prevents the scheme (6) from converging fast.

The composition of the objective function of Equation (3) is decoupled in its dual formulation. Indeed, the dual formulation of model (3) has a form of

$$\min\{f^*(w) : A^T w = 0, w \in \mathbb{R}^m\}. \quad (7)$$

where f^* represents the Fenchel conjugate of f whose definition will be given in the next section. A solution of Equation (3) and a solution of Equation (7) can be derived from a stationary point of the Lagrangian function of Equation (7). Augmented Lagrangian methods [4–8] are commonly adopted for searching stationary points of the Lagrangian function of Equation (7). The convergence of an augmented Lagrangian method is guaranteed as long as the subproblem, which is usually involved in the augmented Lagrangian method, is solved to an increasingly highly accuracy at every iteration [8]. Therefore, solving the subproblem may become costly.

Correspondingly, the dual formulation of model (1) has a form of

$$\min\{f_1^*(u) + f_2^*(v) : A_1^T u + A_2^T v = 0, u \in \mathbb{R}^{m_1}, v \in \mathbb{R}^{m_2}\}, \quad (8)$$

From the notation point of view, the formulation (8) generalizes the formulation (5) without considering the conjugate, matrix transpose. To find a stationary point of the Lagrangian function of Equation (8) and therefore find solutions of Equations (1) and (8), the well-known ADMM (alternating direction method of multiplier) can be applied. The ADMM allows block-wise Gauss-Seidel acceleration among the variables u, v in solving the subproblems involved in the iterations. This illustrates some advantage of ADMM over the scheme (6) since no Gauss-Seidel acceleration occurs among the variable u, v when scheme (6) is applied to solve Equation (1). But as in Augmented Lagrange Method, solving the subproblems could be costly in ADMM.

Both the advantage and disadvantage of ADMM and scheme (6) motivates us to develop efficient and fast algorithms to solve the problems (1) and (3). Firstly of all, we provides a characterization of solutions of general problem (3) and its dual formulation (7) from the sub-differential point of view and develop proximal type algorithms from the characterization. We shall show the proposed algorithms have explicit form under the assumption that the proximity operators of f has closed form. Further, if the function f exhibits some appealing structures, we are able to derive an accelerated variant algorithm. Indeed, we show that if the function f is separable and problem (3) becomes (1), we are able to relax the parameters introduced in the algorithm and use block-wise Gaussian-Seidel acceleration. We shall show that this variant algorithm is a type of alternating direction method but exhibits some advantage over the classical ADMM (alternating direction method of multiplier).

This paper is organized in the following manner. In Section 2, we provide a characterization of solutions of the primal problem (3) and show that a stationary point of the Lagrangian function of Equation (7) yields a solution of Equation (3). We further develop a proximity operator based algorithm based on the characterization of solutions. In Section 3, we propose an accelerated variant algorithm if the function f is well-structured (i.e., f is separable). A unified convergence analysis of both algorithms is provided in this section. In Section 4, we discuss the connection of the proposed algorithms to CP (Chambolle and Pock's primal-dual) method, Augmented Lagrangian method and Alternating Direction Method of Multipliers. In Section 5, We identify the L2-TV and L1-TV models as special cases of the general problem (3) and demonstrate that the proximity operators of the corresponding functions can be efficiently computed in the proposed algorithms. in section 6, we apply the proposed algorithms to solve L2-TV and L1-TV image deblurring problems. The performance of the proposed algorithms is shown and comparison of proposed algorithms with CP [9] and scheme (6) is fulfilled. The conclusions on the proposed algorithms are given in the last section.

2. DUAL FORMULATION: ALGORITHM

In this section, we shall see that a saddle point of the Lagrangian function of Equation (7) will yield a solution of Equation (3) and

a solution of Equation (7). We identify the saddle point of the Lagrangian function of Equation (7) as a solution of a fixed-point equation in terms of proximity operator and propose an iterative scheme to solve this fixed-point equation. A connection of the resulting iterative scheme with the one given in Zhang et al. [1] is pointed out.

We begin with introducing our notation and recalling some background from convex analysis. For a vector x in the d -dimensional Euclidean space \mathbb{R}^d , we use x_i to denote the i th component of a vector $x \in \mathbb{R}^d$ for $i = 1, 2, \dots, d$. We define $\langle x, y \rangle := \sum_{i=1}^d x_i y_i$, for $x, y \in \mathbb{R}^d$ the standard inner product in \mathbb{R}^d . The ℓ_2 -norm induced by the inner product in \mathbb{R}^d is defined as $\|\cdot\|_2 := \sqrt{\langle \cdot, \cdot \rangle}$. For the Hilbert space \mathbb{R}^d , the class of all lower semicontinuous convex functions $\psi : \mathbb{R}^d \rightarrow \overline{\mathbb{R}} := (-\infty, +\infty]$ such that $\text{dom } \psi := \{x \in \mathbb{R}^d : \psi(x) < +\infty\} \neq \emptyset$ is denoted by $\Gamma_0(\mathbb{R}^d)$. In this paper, we always assume that $f_1 \in \Gamma_0(\mathbb{R}^{m_1})$, $f_2 \in \Gamma_0(\mathbb{R}^{m_2})$, and $f \in \Gamma_0(\mathbb{R}^m)$, where the functions f_1 and f_2 are in Equation (1) and f in Equation (3).

We shall provide necessary and sufficient conditions for a solution to the model (3). To this end, we first recall the definition of sub-differential and definition of Fenchel conjugate. The subdifferential of $\psi \in \Gamma_0(\mathbb{R}^d)$ is defined as the set-valued operator $\partial\psi : x \in \mathbb{R}^d \mapsto \{y \in \mathbb{R}^d : \psi(z) \geq \psi(x) + \langle y, z - x \rangle \text{ for all } z \in \mathbb{R}^d\}$. For a function $\psi : \mathbb{R}^d \rightarrow [-\infty, +\infty]$, the Fenchel conjugate of ψ at $x \in \mathbb{R}^d$ is $\psi^*(x) := \sup\{\langle y, x \rangle - \psi(y) : y \in \mathbb{R}^d\}$.

The characterization of a solution to the model (3) in terms of sub-differential is given in the following.

Proposition 2.1. $\hat{x} \in \mathbb{R}^n$ is a solution to the model (3) iff there exists a $\hat{w} \in \mathbb{R}^m$ such that the following hold

$$A^\top \hat{w} = 0, \quad A\hat{x} \in \partial f^*(\hat{w}). \tag{9}$$

PROOF. Suppose \hat{x} is a solution to Equation (3). By Fermat's rule, $0 \in A^\top \partial f(A\hat{x})$. There exists a $\hat{w} \in \partial f(A\hat{x})$ such that $0 = A^\top \hat{w}$. Since $f \in \Gamma_0(\mathbb{R}^m)$, $\hat{w} \in \partial f(A\hat{x})$ implies $A\hat{x} \in \partial f^*(\hat{w})$ by Proposition 11.3 in [10]. The Equation (9) follows.

The above reasoning is reversible. That is, if there is a $\hat{w} \in \partial f(A\hat{x})$ such that Equation (9) holds, then \hat{x} is a solution to model (3). \square

In the meantime, Equation (9) also characterizes the KKT conditions for the linear constraint minimization problem

$$\min\{f^*(w) : A^\top w = 0, w \in \mathbb{R}^m\}$$

in a way that \hat{x} acts as the Lagrange multipliers of the Lagrangian function $f^*(w) - \langle x, A^\top w \rangle$.

The Equation (9) in Proposition 2.1 provides a necessary and sufficient condition characterization for a solution to model (3). Based on this Proposition, we shall provide a fixed point equation characterization of a solution to model (3) based on proximity operator. The proximity operator of ψ is defined by $\text{prox}_\psi(x) := \text{argmin} \left\{ \frac{1}{2} \|u - x\|^2 + \psi(u) : u \in \mathbb{R}^d \right\}$. The sub-differential and the proximity operator are closely related. Indeed, if $\psi \in \Gamma_0(\mathbb{R}^d)$ and $\lambda > 0$, then

$$y \in \partial\psi(x) \Leftrightarrow x = \text{prox}_{\lambda\psi}(x + \lambda y). \tag{10}$$

Proposition 2.2. $\hat{x} \in \mathbb{R}^n$ is a solution to the model (3) iff for any positive numbers $\alpha > 0$, $\beta > 0$, $\gamma > 0$, there exists an $\hat{w} \in \mathbb{R}^m$ such that the following hold

$$\begin{cases} \hat{w} = \text{prox}_{\alpha f^*}(\hat{w} + \alpha A\hat{x}), \\ \hat{x} = \hat{x} - \gamma A^\top \hat{w} \end{cases}, \tag{11}$$

or equivalently,

$$\begin{cases} \hat{w} = \text{prox}_{\alpha f^*}(\hat{w} + \alpha A(\hat{x} - \beta A^\top \hat{w})), \\ \hat{x} = \hat{x} - \gamma A^\top \hat{w} \end{cases}. \tag{12}$$

PROOF. It follows from proposition 2.1 and Equation (10). \square

Intuitively, the fixed-point Equation (11) yield the following simple iteration scheme

$$\begin{cases} w^{k+1} = \text{prox}_{\alpha f^*}(w^k + \alpha Ax^k), \\ x^{k+1} = x^k - \gamma A^\top w^{k+1}. \end{cases} \tag{13}$$

which is in nature the classical Arrow-Hurwicz algorithm [11] for a stationary point of Lagrange function $L(w, x) = f^*(w) - \langle x, A^\top w \rangle$. Although the Arrow-Hurwicz algorithm exhibits appealing simplicity feature, its convergence is only established under the assumption that f^* is strictly convex [12]. But the assumption on the strict convexity is usually not satisfied.

A modification of the Arrow-Hurwitz algorithm based on the fixed point Equation (12) yields

$$\begin{cases} w^{k+1} = \text{prox}_{\alpha f^*}(w^k + \alpha A(x^k - \beta A^\top w^k)), \\ x^{k+1} = x^k - \gamma A^\top w^{k+1}. \end{cases} \tag{14}$$

We shall show that this modification allow us to achieve convergence for general convex function f^* . Further, we are able to develop Gauss-Seidel acceleration if the function f is separable.

The key to the iterative scheme (14) is to compute the proximity operator of the f^* in the first step. We point it out that $\text{prox}_{\alpha f^*}$ can be easily computed, if necessary, by using the Moreau's decomposition [13, 14] $I = \text{prox}_{\lambda f^*} + \lambda \text{prox}_{\frac{1}{\lambda} f} \circ (\frac{1}{\lambda} I)$ as long as the proximity operator of its Fenchel conjugate of f can be computed easily, and vice-versa.

We adapt the scheme (14) for model (3) to model (1). Specifically, we have

$$f(w) = f_1(u) + f_2(v) \quad \text{and} \quad A = \begin{bmatrix} A_1 \\ A_2 \end{bmatrix}, \tag{15}$$

where $w = (u, v) \in \mathbb{R}^{m_1} \times \mathbb{R}^{m_2}$, $f_1 \in \Gamma_0(\mathbb{R}^{m_1})$, $f_2 \in \Gamma_0(\mathbb{R}^{m_2})$, A_1 is an $m_1 \times n$ matrix, and A_2 is an $m_2 \times n$ matrix. It can be directly verified that for $w = (u, v) \in \mathbb{R}^{m_1} \times \mathbb{R}^{m_2}$

$$f^*(w) = f_1^*(u) + f_2^*(v) \quad \text{and} \quad \text{prox}_{\alpha f^*}(w) = \begin{bmatrix} \text{prox}_{\alpha f_1^*}(u) \\ \text{prox}_{\alpha f_2^*}(v) \end{bmatrix}.$$

Hence, an adaptation of the iterative scheme (14) for model (1) is presented in Algorithm 1.

Algorithm 1 Dual Algorithm for Model (1)

Initialization: $u^0 \in \mathbb{R}^{m_1}$, $v^0 \in \mathbb{R}^{m_2}$, $x^0 \in \mathbb{R}^n$; parameters α , β , γ satisfying inequalities $0 < \gamma \leq 2\beta$ and $0 < \alpha\beta < 1/\left\|\begin{bmatrix} A_1 \\ A_2 \end{bmatrix}\right\|^2$.
repeat($k \geq 0$)

$$\begin{cases} u^{k+1} = \text{prox}_{\alpha f_1^*} \left(u^k + \alpha A_1 (x^k - \beta(A_1^\top u^k + A_2^\top v^k)) \right) \\ v^{k+1} = \text{prox}_{\alpha f_2^*} \left(v^k + \alpha A_2 (x^k - \beta(A_1^\top u^k + A_2^\top v^k)) \right) \\ x^{k+1} = x^k - \gamma(A_1^\top u^{k+1} + A_2^\top v^{k+1}) \end{cases} \quad (16)$$

until a given stopping criteria is met

3. PARAMETERS RELAXATION AND GAUSS-SEIDEL METHOD FOR ALGORITHM 1 AND ITS CONVERGENCE ANALYSIS

In this section, Algorithm 1 is modified from two aspects. First, the parameter α will be relaxed such that it is different for updating u^{k+1} and v^{k+1} . Second, the block Gauss-Seidel technique is applied in the sense that the updated result u^{k+1} will be immediately used in computing v^{k+1} . As a consequence, we have a new algorithm, as a variant of Algorithm 1, which is depicted in Algorithm 2.

Algorithm 2 Gauss-Seidel Method for Model (1)

Initialization: $u^0 \in \mathbb{R}^{m_1}$, $v^0 \in \mathbb{R}^{m_2}$, $x^0 \in \mathbb{R}^n$; $\alpha_1, \alpha_2, \beta, \gamma > 0$.
repeat($k \geq 0$)

$$\begin{cases} u^{k+1} = \text{prox}_{\alpha_1 f_1^*} \left(u^k + \alpha_1 A_1 (x^k - \beta(A_1^\top u^k + A_2^\top v^k)) \right) \\ v^{k+1} = \text{prox}_{\alpha_2 f_2^*} \left(v^k + \alpha_2 A_2 (x^k - \beta(A_1^\top u^{k+1} + A_2^\top v^k)) \right) \\ x^{k+1} = x^k - \gamma(A_1^\top u^{k+1} + A_2^\top v^{k+1}) \end{cases} \quad (17)$$

until a given stopping criteria is met

Next, we describe the general forms of the schemes (14) and (17) based on which the convergence analysis of the two algorithms will be derived. By the definition of proximity operator, the first step of iterative scheme (14) can be equivalently rephrased as

$$w^{k+1} = \text{argmin} \left\{ f^*(w) + \frac{1}{2\alpha} \|w\|^2 - (w^k + \alpha A(x^k - \beta A^\top w^k)) \right\}_2 : w \in \mathbb{R}^m \quad (18)$$

rearranging the objective function in Equation (18) and ignoring some constant, the optimization problem of Equation (18) is equivalent to

$$w^{k+1} = \text{argmin} \left\{ \begin{aligned} & f^*(w) - \langle x^k, A^\top w \rangle + \frac{\beta}{2} \|A^\top w\|_2^2 + \frac{1}{2} \\ & \langle (\frac{1}{\alpha} I - \beta A A^\top)(w - w^k), w - w^k \rangle : \\ & w \in \mathbb{R}^m \end{aligned} \right\} \quad (19)$$

Under the condition $0 < \alpha\beta < \frac{1}{\|A\|^2}$, where $\|A\|$ is the largest singular value of A , the matrix $\frac{1}{\alpha} I - \beta A A^\top$ is symmetric, positive definite. As a result, the iterative scheme (14) can be cast into the following iterative scheme given in Zhang et al. [1]:

$$\begin{cases} w^{k+1} = \text{argmin} \left\{ \begin{aligned} & f^*(w) - \langle x^k, A^\top w \rangle + \frac{\beta}{2} \|A^\top w\|_2^2 \\ & + \frac{1}{2} \|w - w^k\|_Q^2 : w \in \mathbb{R}^m \end{aligned} \right\} \\ x^{k+1} = x^k - \gamma A^\top w^{k+1} \end{cases} \quad (20)$$

where the Q -norm $\|\cdot\|_Q$ is defined as $\sqrt{\langle Q \cdot, \cdot \rangle}$ for a positive definite symmetric matrix Q .

Similarly, the iterative scheme (17) in Algorithm 2 can be cast as a special case of the following scheme

$$\begin{cases} u^{k+1} = \text{argmin} \left\{ \begin{aligned} & f_1^*(u) - \langle x^k, A_1^\top u + A_2^\top v^k \rangle + \frac{\beta}{2} \|A_1^\top u \\ & + A_2^\top v^k\|_2^2 + \frac{1}{2} \|u - u^k\|_{Q_1}^2 : u \in \mathbb{R}^{m_1} \end{aligned} \right\} \\ v^{k+1} = \text{argmin} \left\{ \begin{aligned} & f_2^*(v) - \langle x^k, A_1^\top u^{k+1} + A_2^\top v \rangle \\ & + \frac{\beta}{2} \|A_1^\top u^{k+1} + A_2^\top v\|_2^2 + \frac{1}{2} \|v - v^k\|_{Q_2}^2 : v \in \mathbb{R}^{m_2} \end{aligned} \right\} \\ x^{k+1} = x^k - \gamma(A_1^\top u^{k+1} + A_2^\top v^{k+1}) \end{cases} \quad (21)$$

where both Q_1 and Q_2 are symmetric and positive definite matrices. Indeed, by choosing $Q_1 = \frac{1}{\alpha_1} I - \beta A_1 A_1^\top$, $Q_2 = \frac{1}{\alpha_2} I - \beta A_2 A_2^\top$ with $0 < \alpha_1 \beta < \frac{1}{\|A_1\|^2}$ and $0 < \alpha_2 \beta < \frac{1}{\|A_2\|^2}$, the matrices Q_1, Q_2 are symmetric, positive definite and scheme (21) reduces to (17) in Algorithm 2. Recall that the parameters α, β in Algorithm 1 need satisfy $0 < \alpha\beta < \frac{1}{\|A\|^2}$, where A is chosen as Equation (15). It can be noticed that $\max\{\|A_1\|^2, \|A_2\|^2\} \leq \|A\|^2$, which implies $\min\{\frac{1}{\|A_1\|^2}, \frac{1}{\|A_2\|^2}\} \geq \frac{1}{\|A\|^2}$. Hence, more flexibility exhibits for the choice of $\alpha_1, \alpha_2, \beta$ in Algorithm 2 than for the choice of α, β in Algorithm 1.

We remark that the iterative scheme (21) generalizes the iterative scheme (22)

$$\begin{cases} u^{k+1} = \text{argmin} \left\{ \begin{aligned} & f_1^*(u) - \langle x^k, A_1^\top u - v^k \rangle + \frac{\beta}{2} \|A_1^\top u - v^k\|_2^2 \\ & + \frac{1}{2} \|u - u^k\|_{Q_1}^2 : u \in \mathbb{R}^{m_1} \end{aligned} \right\} \\ v^{k+1} = \text{argmin} \left\{ \begin{aligned} & f_2^*(v) - \langle x^k, A_1^\top u^{k+1} \\ & - v \rangle + \frac{\beta}{2} \|A_1^\top u^{k+1} - v\|_2^2 + \frac{1}{2} \|v - v^k\|_{Q_2}^2 : v \in \mathbb{R}^{m_2} \end{aligned} \right\} \\ x^{k+1} = x^k - \gamma(A_1^\top u^{k+1} - v^{k+1}) \end{cases} \quad (22)$$

whose form is equivalent to Equation (6) by specifying the variables, functions and matrices appropriately. Indeed, when A_2 is specified as $-I$, the iterative scheme (21) reduces to (22). Unlike [1], the iterative scheme (21) is derived from the dual formulation instead of primal problems. In addition, the generality of matrix A_2 in iterative scheme (21) generalizes the matrix $-I$ in scheme (6).

The rest of this section is devoted to a unified convergence analysis of the two iterative schemes (20) and (21), from which Algorithm 1 and Algorithm 2 can be derived. The convergence of these two schemes is analyzed in the following manner: we

first prove the convergence of schemes (21) and obtain the convergence of scheme (20) as an immediate result.

For convenient exposition, Equation (9) for the special case Equation (15) is rewritten as

$$A_1^\top \hat{u} + A_2^\top \hat{v} = 0, \quad A_1 \hat{x} \in \partial f_1^*(\hat{u}), \quad \text{and} \quad A_2 \hat{x} \in \partial f_2^*(\hat{v}). \quad (23)$$

We will show that for any initial seed $(u^0, v^0, x^0) \in \mathbb{R}^{m_1} \times \mathbb{R}^{m_2} \times \mathbb{R}^n$ and a positive parameter γ the sequence $\{(u^k, v^k, x^k) : k \in \mathbb{N}\}$ converges to a triple $(\hat{u}, \hat{v}, \hat{x}) \in \mathbb{R}^{m_1} \times \mathbb{R}^{m_2} \times \mathbb{R}^n$ satisfying Equation (23).

First of all, we look at the characterization of u^{k+1}, v^{k+1} involved in the two subproblems in Equation (21). Using Fermat's rule for the two subproblems, we are able to get

$$\begin{cases} A_1 x^k - \beta A_1 (A_1^\top u^{k+1} + A_2^\top v^k) - Q_1 (u^{k+1} - u^k) \in \partial f_1^*(u^{k+1}), \\ A_1 x^k - \beta A_1 (A_1^\top u^{k+1} + A_2^\top v^k) - Q_2 (u^{k+1} - u^k) \in \partial f_2^*(v^{k+1}). \end{cases}$$

Let

$$\begin{cases} r^{k+1} := A_1 x^k - \beta A_1 (A_1^\top u^{k+1} + A_2^\top v^k) - Q_1 (u^{k+1} - u^k), \\ t^{k+1} := A_2 x^k - \beta A_2 (A_1^\top u^{k+1} + A_2^\top v^{k+1}) - Q_2 (v^{k+1} - v^k). \end{cases} \quad (24)$$

Then $r^{k+1} \in f_1^*(u^{k+1}), t^{k+1} \in f_2^*(v^{k+1})$. If the sequence $\{(u^k, v^k, x^k) : k \in \mathbb{N}\}$ converges to a triple $(\hat{u}, \hat{v}, \hat{x}) \in \mathbb{R}^{m_1} \times \mathbb{R}^{m_2} \times \mathbb{R}^n$ satisfying Equation (23), we observe $r^{k+1} \rightarrow A_1 \hat{x}, t^{k+1} \rightarrow A_2 \hat{x}$ as $k \rightarrow \infty$. For convenience, we also introduce the following notations

$$\begin{aligned} \hat{\rho} &:= \begin{bmatrix} \hat{u} \\ \hat{v} \\ \hat{x} \end{bmatrix}, \quad \rho^k := \begin{bmatrix} u^k \\ v^k \\ x^k \end{bmatrix}, \quad P := \begin{bmatrix} Q_1 & & \\ & Q_2 & \\ & & \frac{1}{\gamma} I \end{bmatrix}, \quad r := A_1 \hat{x}, \\ t &:= A_2 \hat{x} \\ u_e^k &:= u^k - \hat{u}, \quad v_e^k := v^k - \hat{v}, \quad x_e^k := x^k - \hat{x}, \\ r_e^{k+1} &:= r^{k+1} - r, \quad t_e^{k+1} := t^{k+1} - t, \quad \rho_e^k := \rho^k - \hat{\rho}. \end{aligned}$$

Next, we shall establish a relationship between a triple $(\hat{u}, \hat{v}, \hat{x})$ and the sequence $\{(u^k, v^k, x^k) : k \in \mathbb{N}\}$ generated by the iterative scheme (21).

Lemma 3.1. *Let Q_1 and Q_2 be two positive definite symmetric matrices, let the triple $(\hat{u}, \hat{v}, \hat{x}) \in \mathbb{R}^{m_1} \times \mathbb{R}^{m_2} \times \mathbb{R}^n$ satisfy Equation (23), and let $\{(u^k, v^k, x^k) : k \in \mathbb{N}\}$ be the sequence generated by scheme (21).*

Then the following equation holds:

$$(\| \rho_e^{k+1} \|_P^2 + \beta \| A_2^\top v_e^{k+1} \|^2) - (\| \rho_e^k \|_P^2 + \beta \| A_2^\top v_e^k \|^2) = y^k, \quad (25)$$

where

$$\begin{aligned} y^k &= - \| u^{k+1} - u^k \|_{Q_1}^2 - 2 \langle r_e^{k+1}, u_e^{k+1} \rangle - \| v^{k+1} - v^k \|_{Q_2}^2 \\ &\quad - 2 \langle t_e^{k+1}, v_e^{k+1} \rangle - (\beta - \gamma) \| A_1^\top u^{k+1} + A_2^\top v^{k+1} \|^2 \\ &\quad - \beta \| A_1^\top u^{k+1} + A_2^\top v^k \|^2. \end{aligned}$$

PROOF. It follows from the definitions of $r, r^k, t,$ and t^k , the iteration scheme (21) and the characterization of saddle points Equation (23) that

$$\begin{cases} r_e^{k+1} - A_1 x_e^k + \beta A_1 (A_1^\top u_e^{k+1} + A_2^\top v_e^k) \\ \quad + Q_1 (u^{k+1} - u^k) = 0, \\ t_e^{k+1} - A_2 x_e^k + \beta A_2 (A_1^\top u_e^{k+1} + A_2^\top v_e^{k+1}) \\ \quad + Q_2 (v^{k+1} - v^k) = 0, \\ x_e^{k+1} = x_e^k - \gamma (A_1^\top u_e^{k+1} + A_2^\top v_e^{k+1}). \end{cases} \quad (26)$$

By taking the inner product with $2u_e^{k+1}$ on the both sides of the first equality of Equation (26) and rearranging the terms and using the identity $\| u_e^k \|_{Q_1}^2 = \| u^{k+1} - u^k \|_{Q_1}^2 + \| u_e^{k+1} \|_{Q_1}^2 - 2 \langle Q_1 (u^{k+1} - u^k), u_e^{k+1} \rangle$, we obtain

$$\begin{aligned} (\| u_e^{k+1} \|_{Q_1}^2 - \| u_e^k \|_{Q_1}^2) &= - \| u^{k+1} - u^k \|_{Q_1}^2 - 2 \langle r_e^{k+1}, u_e^{k+1} \rangle \\ &\quad - 2\beta \langle A_1^\top u_e^{k+1}, A_1^\top u_e^{k+1} + A_2^\top v_e^k \rangle \\ &\quad + 2 \langle A_1^\top u_e^{k+1}, x_e^k \rangle, \end{aligned}$$

Likewise, by taking the inner product with $2v_e^{k+1}$ and $2x_e^k$ on the both sides of the second and third Equations of (26), we obtain

$$\begin{aligned} (\| v_e^{k+1} \|_{Q_2}^2 - \| v_e^k \|_{Q_2}^2) &= - \| v^{k+1} - v^k \|_{Q_2}^2 - 2 \langle t_e^{k+1}, v_e^{k+1} \rangle \\ &\quad - 2\beta \langle A_2^\top v_e^{k+1}, A_1^\top u_e^{k+1} + A_2^\top v_e^{k+1} \rangle \\ &\quad + 2 \langle A_2^\top v_e^{k+1}, x_e^k \rangle, \end{aligned}$$

and

$$\begin{aligned} \frac{1}{\gamma} (\| x_e^{k+1} \|_2^2 - \| x_e^k \|_2^2) &= \gamma \| A_1^\top u_e^{k+1} + A_2^\top v_e^{k+1} \|^2 \\ &\quad - 2 \langle x_e^k, A_1^\top u_e^{k+1} + A_2^\top v_e^{k+1} \rangle, \end{aligned}$$

respectively. By adding up the above three equations and using the identity $\| \rho^k - \hat{\rho} \|_P^2 = \| u^k - \hat{u} \|_{Q_1}^2 + \| v^k - \hat{v} \|_{Q_2}^2 + \frac{1}{\gamma} \| x^k - \hat{x} \|_2^2$ and the fact $A_1^\top \hat{u} + A_2^\top \hat{v} = 0$, we get Equation (25). This completes the proof of the result. \square

To show that the convergence of sequence $\{(u^k, v^k, x^k) : k \in \mathbb{N}\}$ generated by the iterative scheme (21), we need two properties of the subdifferential. The first one is the monotonicity of the subdifferential. The subdifferential a function $\psi \in \Gamma_0(\mathbb{R}^d)$ as a set-valued function is monotone (see [15]) in the sense that for any u and v in the domain of ψ

$$\langle \tilde{u} - \tilde{v}, u - v \rangle \geq 0, \quad \text{for all } \tilde{u} \in \partial \psi(u), \tilde{v} \in \partial \psi(v).$$

Another useful property of the sub-differential is presented in the following lemma.

Lemma 3.2. *Let ψ be in $\Gamma_0(\mathbb{R}^d)$ and let $\{(x^k, y^k) : k \in \mathbb{N}\}$ be a sequence with $y^k \in \partial \psi(x^k)$. Suppose that $x^k \rightarrow x$ and $y^k \rightarrow y$ as $k \rightarrow \infty$. Then $y \in \partial \psi(x)$.*

PROOF. By the definition of sub-differential, the inequality $\psi(z) \geq \psi(x^k) + \langle y^k, z - x^k \rangle$ holds all $z \in \mathbb{R}^d$ and $k \in \mathbb{N}$. By taking limit inferior to both sides of the above inequality, it

follows that $\psi(z) \geq \liminf_{k \rightarrow \infty} (\psi(x^k) + \langle y^k, z - x^k \rangle)$. By virtue of the lower semi-continuity of ψ , i.e., $\liminf_{k \rightarrow \infty} \psi(x^k) \geq \psi(x)$, and the fact of $\langle y^k, z - x^k \rangle \rightarrow \langle y, z - x \rangle$, we obtain that $\psi(z) \geq \psi(x) + \langle y, z - x \rangle$, for all $z \in \mathbb{R}^d$. That is, $y \in \partial\psi(x)$. \square

With these preparations, we are ready to prove the convergence of the sequence $\{(u^k, v^k, x^k) : k \in \mathbb{N}\}$ generated by the iterative scheme (21).

Theorem 3.3. *Let Q_1 and Q_2 be two positive definite symmetric matrices and let $\{(u^k, v^k, x^k) : k \in \mathbb{N}\}$ be the sequence generated by scheme (21). If $0 < \gamma \leq \beta$, then the sequence $\{(u^k, v^k, x^k) : k \in \mathbb{N}\}$ converges to a triple $(\hat{u}, \hat{v}, \hat{x})$ satisfying Equation (23).*

PROOF. To show $\{(u^k, v^k, x^k) : k \in \mathbb{N}\}$ converges to a triple $(\hat{u}, \hat{v}, \hat{x})$ satisfying Equation (23), we first show $\{(u^k, v^k, x^k) : k \in \mathbb{N}\}$ is bounded and therefore has a convergent subsequence, then show that this convergent subsequence converges to a triple $(\hat{u}, \hat{v}, \hat{x})$ satisfying Equation (23) and finally show the entire sequence $\{(u^k, v^k, x^k) : k \in \mathbb{N}\}$ converges to this triple $(\hat{u}, \hat{v}, \hat{x})$.

Let the symbols $\rho, \rho^k, \rho_e^k, P, r, r^k, r_e^k, t, t^k, t_e^k$, and y^k be the same as before. Equations (23) and (24) and the monotonicity of subdifferential yield

$$\langle r_e^{k+1}, u_e^{k+1} \rangle \geq 0 \quad \text{and} \quad \langle t_e^{k+1}, v_e^{k+1} \rangle \geq 0. \quad (27)$$

Therefore, when $0 < \gamma \leq \beta$, the values of y^k are non-positive. Thus, from Equation (25) we know that the sequence $\{\|\rho_e^k\|_P^2 + \beta\|A_2^\top v_e^k\|_2^2 : k \in \mathbb{N}\}$ is decreasing and convergent. This implies the boundedness of the sequence $\{\|\rho^k - \rho\|_P : k \in \mathbb{N}\}$ which further yield the boundedness of the sequence $\{(u^k, v^k, x^k) : k \in \mathbb{N}\}$. Therefore, there exists a convergent subsequence $\{(u^{k_i}, v^{k_i}, x^{k_i}) : i \in \mathbb{N}\}$ such that for some vector $(\tilde{u}, \tilde{v}, \tilde{x}) \in \mathbb{R}^{m_1} \times \mathbb{R}^{m_2} \times \mathbb{R}^n$

$$(u^{k_i}, v^{k_i}, x^{k_i}) \rightarrow (\tilde{u}, \tilde{v}, \tilde{x}) \quad (28)$$

as i goes to infinity.

We shall show that $(\tilde{u}, \tilde{v}, \tilde{x})$ satisfies Equation (23), that is, $A_1^\top \tilde{u} + A_2^\top \tilde{v} = 0$, $A_1 \tilde{x} \in \partial f_1^*(\tilde{u})$, and $A_2 \tilde{x} \in \partial f_2^*(\tilde{v})$. Summing Equation (25) for k from 1 to infinity, we conclude that

$$\sum_{k=1}^{\infty} \left(\|u^{k+1} - u^k\|_{Q_1}^2 + \|v^{k+1} - v^k\|_{Q_2}^2 + (\beta - \gamma)\|A_1^\top u^{k+1} + A_2^\top v^{k+1}\|_2^2 \right) \leq \|\rho_e^1\|_P^2 + \beta\|A_2^\top v_e^1\|_2^2. \quad (29)$$

The convergence of three series in inequality Equation (29) yields that as k goes to infinity

$$u^{k+1} - u^k \rightarrow 0, \quad v^{k+1} - v^k \rightarrow 0, \quad A_1^\top u^{k+1} + A_2^\top v^{k+1} \rightarrow 0,$$

which particularly indicate

$$u^{k_i+1} - u^{k_i} \rightarrow 0, \quad v^{k_i+1} - v^{k_i} \rightarrow 0, \quad A_1^\top u^{k_i+1} + A_2^\top v^{k_i+1} \rightarrow 0, \quad (30)$$

as i goes to infinity. By Equations (28) and (30), we have that

$$\lim_{i \rightarrow \infty} u^{k_i+1} = \tilde{u}, \quad \lim_{i \rightarrow \infty} v^{k_i+1} = \tilde{v}, \quad (31)$$

and

$$A_1^\top \tilde{u} + A_2^\top \tilde{v} = \lim_{i \rightarrow \infty} (A_1^\top u^{k_i+1} + A_2^\top v^{k_i+1}) = 0. \quad (32)$$

Equations (28), (30) and (32) together with the definitions of r^k and t^k yield

$$\lim_{i \rightarrow \infty} r^{k_i+1} = A_1 \tilde{x} \quad \text{and} \quad \lim_{i \rightarrow \infty} t^{k_i+1} = A_2 \tilde{x}. \quad (33)$$

Recall that $r^{k_i+1} \in \partial f_1^*(u^{k_i+1})$, $t^{k_i+1} \in \partial f_2^*(v^{k_i+1})$ and Equations (31) and (33). We obtain from Lemma 3.2 that $A_1 \tilde{x} \in \partial f_1^*(\tilde{u})$ and $A_2 \tilde{x} \in \partial f_2^*(\tilde{v})$. Hence, the vector $(\tilde{u}, \tilde{v}, \tilde{x})$ satisfies Equation (23).

Now, let us take $(\hat{u}, \hat{v}, \hat{x}) = (\tilde{u}, \tilde{v}, \tilde{x})$. Then from Equation (28) we have that

$$\lim_{i \rightarrow \infty} \left(\|\rho^{k_i} - \hat{\rho}\|_P^2 + \beta\|A_2^\top (v^{k_i} - \hat{v})\|_2^2 \right) = 0.$$

The monotonicity and convergence of the sequence $\{\|\rho^k - \hat{\rho}\|_P^2 + \beta\|A_2^\top (v^k - \hat{v})\|_2^2 : k \in \mathbb{N}\}$ imply that

$$\lim_{k \rightarrow \infty} \left(\|\rho^k - \hat{\rho}\|_P^2 + \beta\|A_2^\top (v^k - \hat{v})\|_2^2 \right) = 0.$$

Thus, the sequence $\{(u^k, v^k, x^k) : k \in \mathbb{N}\}$ converges to $(\tilde{u}, \tilde{v}, \tilde{x})$ satisfying Equation (23). This completes the proof of this theorem. \square

Next, we will show that we can specify scheme (20) as a special case of scheme (21) and therefore the convergence of scheme (20) follows automatically. To this end, we consider the two schemes (20), (21) are mutually independent and functions or matrices in the two schemes are not related hereafter.

To cast scheme (20) as scheme (21), we let

$$u = w, \quad f_1^* = f^*, \quad f_2^* = 0, \quad A_1 = A, \quad A_2 = 0, \quad Q_1 = Q \quad (34)$$

in scheme (21). For such the choice of those quantities, we are able to rewrite scheme (21) as

$$\begin{cases} w^{k+1} = \operatorname{argmin}\{f^*(w) - \langle x^k, A^\top w \rangle + \frac{\beta}{2}\|A^\top w\|_2^2 \\ \quad + \frac{1}{2}\|u - w\|_Q^2\} \\ v^{k+1} = \operatorname{argmin}\{-\langle x^k, A^\top w^{k+1} \rangle + \frac{\beta}{2}\|A^\top w^{k+1}\|_2^2 \\ \quad + \frac{1}{2}\|v - v^k\|_{Q_2}^2\} \\ x^{k+1} = x^k - \gamma(A^\top w^{k+1}) \end{cases} \quad (35)$$

from which one can notice that sequence $\{v^k : k \in \mathbb{N}\}$ is a constant vector sequence. Ignoring the trivial step involving v^{k+1} , scheme (35) becomes scheme (20).

Lemma 3.4. *Let Q be a positive definite symmetric matrix, let the pair $(\hat{w}, \hat{x}) \in \mathbb{R}^m \times \mathbb{R}^n$ satisfy Equation (9), and let $\{(w^k, x^k) : k \in \mathbb{N}\}$ be the sequence generated by scheme (20). Set*

$$s := A\hat{x}, \quad s^{k+1} := Ax^k - \beta AA^\top w^{k+1} - Q(w^{k+1} - w^k), \\ \hat{\rho} := \begin{bmatrix} \hat{w} \\ \hat{x} \end{bmatrix}, \quad \rho^k := \begin{bmatrix} w^k \\ x^k \end{bmatrix}, \quad \text{and} \quad P := \begin{bmatrix} Q & \\ & \frac{1}{\gamma}I \end{bmatrix}.$$

Then

$$\begin{aligned} \|\rho^{k+1} - \hat{\rho}\|_p^2 - \|\rho^k - \hat{\rho}\|_p^2 &= -\|w^{k+1} - w^k\|_Q \\ &\quad - 2\langle s^{k+1} - s, w^{k+1} - \hat{w} \rangle \\ &\quad - (2\beta - \gamma)\|A^\top(w^{k+1} - \hat{w})\|_2^2. \end{aligned} \tag{36}$$

PROOF. This is an immediate result of Lemma 3.1 by specifying corresponding quantities as in Equation (34) and noticing that $v^{k+1} = v^k$ for such the choice of those quantities. \square

With Lemma 3.4, we can prove our result on the convergence of the sequence $\{(w^k, x^k) : k \in \mathbb{N}\}$ generated by scheme (20).

Theorem 3.5. *Let Q be a positive definite symmetric matrix and let $\{(w^k, x^k) : k \in \mathbb{N}\}$ be the sequence generated by scheme (20). If $0 < \gamma \leq 2\beta$ then the sequence $\{(w^k, x^k) : k \in \mathbb{N}\}$ converges to a pair (\hat{w}, \hat{x}) satisfying Equation (9).*

PROOF. It follows the proof of Theorem 3.3 by specifying corresponding quantities in scheme (21) as in (34) and using Lemma 3.4. \square

We remark that authors in Zhang et al. [1] only shown convergent subsequence of $\{(w^k, x^k) : k \in \mathbb{N}\}$ converges to a pair (\hat{w}, \hat{x}) satisfying Equation (9). But as shown by Theorem 3.5, the entire sequence $\{(w^k, x^k) : k \in \mathbb{N}\}$ converges to the same point, which strengthens the result in Zhang et al. [1].

4. CONNECTIONS TO EXISTING ALGORITHMS

In this section, we point out the connections of our proposed algorithms to several well-known methods. Specifically, we would explore the connection of the proposed algorithms to Chambolle and Pock’s Primal-Dual method, Augmented Lagrangian Method, Alternating Direction Method of Multipliers, and Generalized ADM.

4.1. Connection to Chambolle and Pock’s Algorithm

First of all, let us review Chambolle and Pock’s (CP) method [9] for solving the following optimization problem

$$\min\{f(Ax) + g(x) : x \in \mathbb{R}^n\}, \tag{37}$$

where $f \in \Gamma_0(\mathbb{R}^m)$, $g \in \Gamma_0(\mathbb{R}^n)$, and A is a matrix of size $m \times n$. We assume that model (37) has a minimizer. The CP method proposed in Chambolle and Pock [9] for model (37) can be written as

$$\begin{cases} w^{k+1} = \text{prox}_{\sigma f^*}(w^k + \sigma A\bar{x}^k), \\ x^{k+1} = \text{prox}_{\tau g}(x^k - \tau A^\top w^{k+1}), \\ \bar{x}^{k+1} = 2x^{k+1} - x^k. \end{cases} \tag{38}$$

For any initial guess $(x^0, \bar{x}^0, w^0) \in \mathbb{R}^n \times \mathbb{R}^n \times \mathbb{R}^m$, the sequence $\{(x^k, w^k) : k \in \mathbb{N}\}$ converges as long as $0 < \sigma\tau < \|A\|^{-2}$.

In particular, when we set $g = 0$, a direct computation shows that $\text{prox}_{\tau g}$ is the identity operator for any $\tau > 0$. Set $\alpha = \sigma$ and $\beta = 2\tau$. Accordingly, the general CP method in Equation (38) becomes

$$\begin{cases} w^{k+1} = \text{prox}_{\alpha f^*}\left(w^k + \alpha A\left(x^{k-1} - \beta A^\top w^k\right)\right), \\ x^{k+1} = x^k - \frac{\beta}{2} A^\top w^{k+1}. \end{cases} \tag{39}$$

On the other hand, when we set $g = 0$, model (37) reduces to model (3). Our algorithm for model (3) is presented in scheme (14).

Therefore, by comparing the CP method and the scheme (14) for model (3), we can see that the CP method uses x^{k-1} while the scheme (14) uses x^k in the computation of w^{k+1} . Further, the step size of the CP method for updating x^{k+1} is fixed as $\frac{\beta}{2}$ while it can be any number in $(0, 2\beta]$ for the scheme (14). Although, the relation $0 < \alpha\beta < 2\|A\|^{-2}$ is required for the CP method while the relation $0 < \alpha\beta < \|A\|^{-2}$ is needed for the scheme (14), for a fixed α , we can choose the step size for the scheme (14) twice bigger than that for the CP method.

4.2. Connection to Augmented Lagrangian Methods

As we already know that the scheme (14) model (3) is derived from the scheme (20) with a proper chosen Q . The scheme (20) is used to solve the constrained dual optimization problem (7).

In the literature of nonlinear programming [16], augmented Lagrangian methods (ALMs) are often used to convert a constrained optimization problem to an unconstrained one by adding the objective function a penalty term associated with the constraints. For model (7), the augmented Lagrangian method is as follows:

$$\begin{cases} w^{k+1} = \text{argmin}\{f^*(w) - \langle x^k, A^\top w \rangle + \frac{\beta}{2}\|A^\top w\|_2^2 : w \in \mathbb{R}^m\}, \\ x^{k+1} = x^k - \beta A^\top w^{k+1}. \end{cases} \tag{40}$$

We can see that the scheme (20) reduces to the ALM Equation (40) if we choose $Q = 0$ and $\gamma = \beta$ in Equation (20). Even though we can assume that the proximity operator of f has a closed form, there is lack of an effective way to update w^{k+1} in Equation (40) when A is not the identity matrix. However, the vector w^{k+1} in the scheme (20) can be effectively updated once a proper Q is chosen. This essentially illustrates that Algorithm 1 is superior to the ALM from the numerical implementation point of view.

4.3. Connection to Alternating Direction Method of Multipliers

By specializing the dual formulation (7) of model (7) to model (1), we have that

$$\min\{f_1^*(u) + f_2^*(v) : A_1^\top u + A_2^\top v = 0, u \in \mathbb{R}^{m_1}, v \in \mathbb{R}^{m_2}\}. \tag{41}$$

The alternating direction method of multipliers (ADMM) for dual problem (41) reads as

$$\begin{cases} u^{k+1} = \operatorname{argmin}\{f_1^*(u) + f_2^*(v^k) - \langle x^k, A_1^\top u + A_2^\top v^k \rangle \\ \quad + \frac{\beta}{2} \|A_1^\top u + A_2^\top v^k\|_2^2 : u \in \mathbb{R}^{m_1}\} \\ v^{k+1} = \operatorname{argmin}\{f_1^*(u^{k+1}) + f_2^*(v) - \langle x^k, A_1^\top u^{k+1} + A_2^\top v \rangle \\ \quad + \frac{\beta}{2} \|A_1^\top u^{k+1} + A_2^\top v\|_2^2 : v \in \mathbb{R}^{m_2}\} \\ x^{k+1} = x^k - \beta(A_1^\top u^{k+1} + A_2^\top v^{k+1}) \end{cases} \quad (42)$$

We can see that the scheme (21) reduces to the above ADMM method if we set $Q_1 = 0$, $Q_2 = 0$, and $\gamma = \beta$ in Equation (21). Similar to what we have observed for the ALM method, solving the two optimization problems in Equation (42) is still challenging in general when both A_1 and A_2 are not the identity matrix. However, the vectors u^{k+1} and v^{k+1} in the scheme (21) can be effectively updated once Q_1 and Q_2 are properly chosen. Hence our Algorithm 2 is superior to the ADMM from the numerical implementation point of view.

4.4. Connection to Generalized Alternating Direction Method

Finally, we comment that the generalized ADM proposed in Deng and Yin [3] can be applied for solving the optimization problem (41) and the resulting algorithm is exactly the same as Algorithm 2. However, the motivations behind [3] and our current paper are completely different. The generalized ADM in Deng and Yin [3] was developed based on the augmented Lagrangian function of the objective function of Equation (41) and the work in there focused on linear convergence rate of the corresponding algorithm. In our current work, we formulated the optimization problem (8) as a constrained optimization problem (8) and recognized the solution of Equation (8) as a solution of a fixed-point equation (see Proposition 2.2) from which Algorithm 2 was naturally derived.

TABLE 1 | Numerical results for the L2-TV model for images blurred by the (21, 10)-GBM.

Method	Cameraman			Lena			Peppers			Goldhill		
	Iters	CPU(s)	PSNR	Iters	CPU(s)	PSNR	Iters	CPU(s)	PSNR	Iters	CPU(s)	PSNR
CP	177	18.82	23.09	157	107.57	26.45	205	21.10	25.38	166	16.64	24.80
ZBO	177	19.90	23.09	157	109.51	26.45	205	22.96	25.38	166	18.43	24.80
Algorithm 1	151	15.98	23.35	140	93.92	26.66	171	17.17	25.61	152	15.48	24.87
Algorithm 2 with Equation (53)	93	11.34	23.45	84	45.60	26.79	98	10.73	25.75	91	10.01	25.13
Algorithm 2 with Equation (55)	86	9.73	23.68	79	38.68	26.94	90	9.95	25.92	83	9.12	25.13

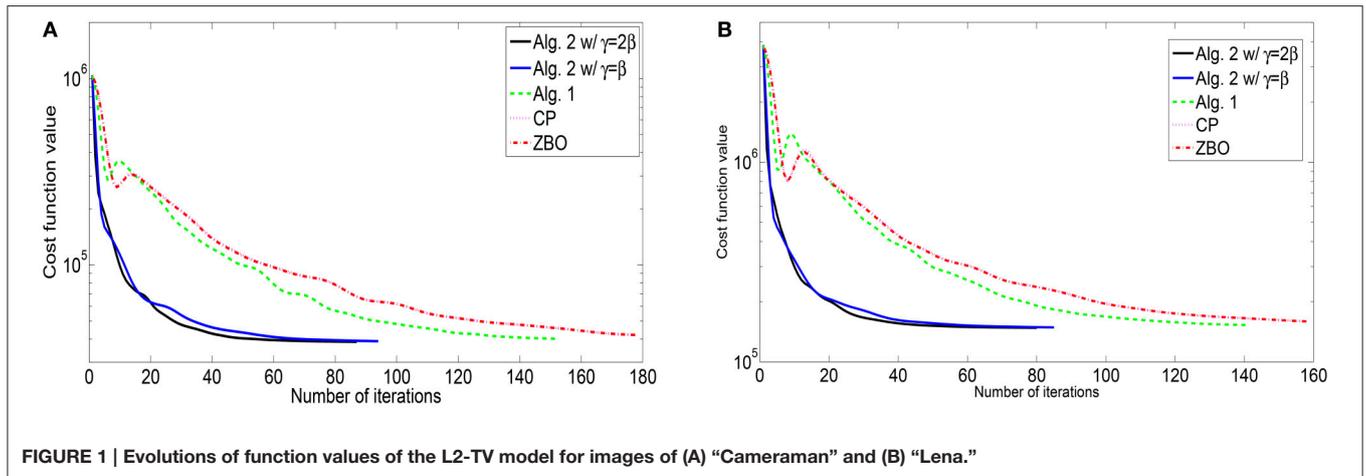


FIGURE 1 | Evolutions of function values of the L2-TV model for images of (A) "Cameraman" and (B) "Lena."

TABLE 2 | Numerical results for the L2-TV model for images blurred by the (15, 10)-GBM.

Method	Cameraman			Lena			Peppers			Goldhill		
	Iters	CPU(s)	PSNR	Iters	CPU(s)	PSNR	Iters	CPU(s)	PSNR	Iters	CPU(s)	PSNR
CP	88	8.00	22.88	77	51.82	26.12	89	8.64	23.98	76	7.42	24.27
ZBO	84	8.95	22.88	77	54.50	26.12	89	9.42	23.98	76	8.06	24.27
Algorithm 1	81	7.76	23.11	76	50.59	26.17	86	8.45	24.19	75	7.53	24.26
Algorithm 2 with Equation (53)	50	5.18	23.00	46	23.75	26.12	50	5.15	24.13	44	4.68	24.29
Algorithm 2 with Equation (55)	49	5.09	23.14	46	25.71	26.10	48	4.96	24.21	44	4.85	24.24

5. APPLICATIONS TO IMAGE DEBLURRING

In this section, we first identify two well-known image deblurring models, namely L2-TV and L1-TV, as special cases of the general model (1). We then give details on how Algorithms 1 and 2 are applied. In particular, we present the explicit expressions of the proximity operators of f_1^* and f_2^* . Since the total variation (TV) is involved in the both image deblurring models, we begin with presenting the discrete setting for total variation.

For convenience of exposition, we assume that an image considered in this paper has a size of $\sqrt{n} \times \sqrt{n}$. The image is

treated as a vector in \mathbb{R}^n in such a way that the ij -th pixel of the image corresponds to the $(i + (j - 1)\sqrt{n})$ -th component of the vector in \mathbb{R}^n . The total variation of the image x can be expressed as the composite function of a convex function $\psi : \mathbb{R}^{2n} \rightarrow \mathbb{R}$ and a $2n \times n$ matrix B . To define the matrix B , we need a $\sqrt{n} \times \sqrt{n}$ difference matrix D as follows:

$$D := \begin{bmatrix} 0 & & & & \\ -1 & 1 & & & \\ & & \ddots & & \\ & & & \ddots & \\ & & & & -1 & 1 \end{bmatrix}$$

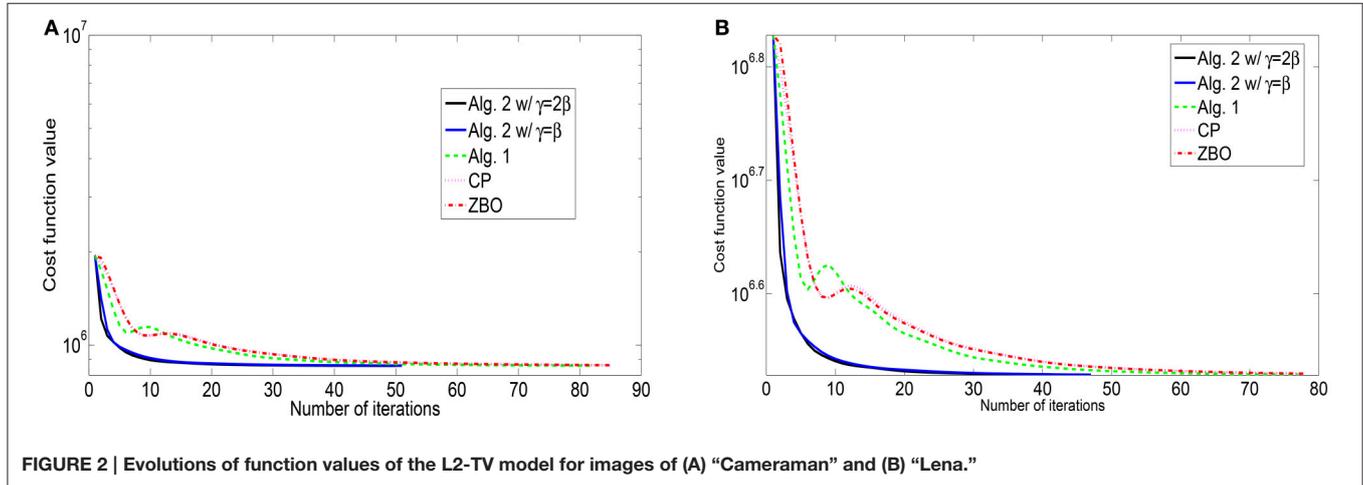


FIGURE 2 | Evolutions of function values of the L2-TV model for images of (A) "Cameraman" and (B) "Lena."

TABLE 3 | Numerical results for the L1-TV model for images blurred by the (21, 10)-GBM.

Method	Cameraman			Lena			Peppers			Goldhill		
	Itrs	CPU(s)	PSNR	Itrs	CPU(s)	PSNR	Itrs	CPU(s)	PSNR	Itrs	CPU(s)	PSNR
CP	367	44.39	23.47	354	241.09	26.53	414	45.57	25.73	381	42.07	25.05
ZBO	368	45.09	23.48	355	251.43	26.53	415	47.70	25.74	382	44.28	25.04
Algorithm 1	275	29.92	23.57	272	194.29	26.69	315	34.68	25.82	312	34.56	25.16
Algorithm 2 with Equation (53)	192	21.34	24.22	189	136.42	27.38	207	22.79	26.66	208	23.09	25.74
Algorithm 2 with Equation (55)	154	17.46	24.33	153	109.64	27.52	162	18.04	26.78	168	18.87	25.78

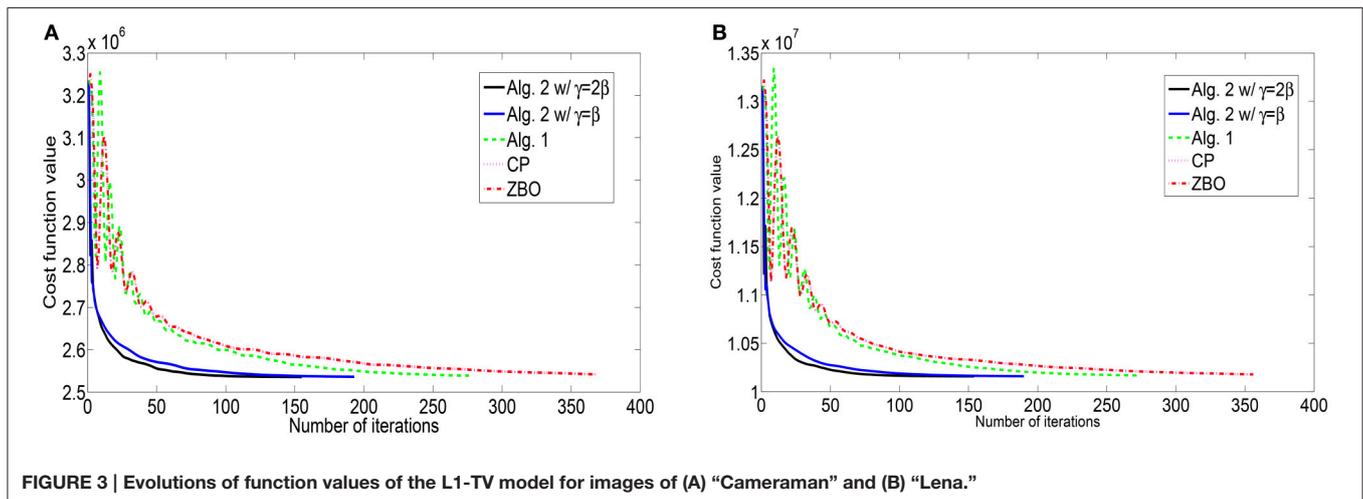


FIGURE 3 | Evolutions of function values of the L1-TV model for images of (A) "Cameraman" and (B) "Lena."

The matrix D will be used to “differentiate” a row or a column of an image matrix. Through the matrix Kronecker product \otimes , we define the $2n \times n$ matrix B by

$$B := \begin{bmatrix} I \otimes D \\ D \otimes I \end{bmatrix}, \tag{43}$$

where I is the $\sqrt{n} \times \sqrt{n}$ identity matrix. The matrix B will be used to “differentiate” the entire image matrix. The norm of B is $\|B\|^2 = 8 \sin^2 \frac{(\sqrt{n}-1)\pi}{2\sqrt{n}}$ (see [17]).

We define $\psi : \mathbb{R}^{2n} \rightarrow \mathbb{R}$ is at $y \in \mathbb{R}^{2n}$ as

$$\psi(y) := \sum_{i=1}^n \left\| [y_i, y_{n+i}]^T \right\|_2, \tag{44}$$

Based on the definition of the $2n \times n$ matrix B and the convex function ψ , the total variation of an image x can be denoted by $\psi(Bx)$ for (isotropic) total variation, i.e.,

$$\|x\|_{TV} := \psi(Bx). \tag{45}$$

5.1. The L2-TV Image Deblurring Model

The L2-TV image deblurring model is usually used for the recovery of an unknown image $x \in \mathbb{R}^n$ from an observable data $b \in \mathbb{R}^n$ modeled by

$$b = Kx + \text{white noise} \tag{46}$$

where K is a blurring matrix of size $n \times n$. The L2-TV image deblurring model has the form of

$$\min \left\{ \frac{1}{2} \|Kx - b\|_2^2 + \mu \|x\|_{TV} : x \in \mathbb{R}^n \right\}, \tag{47}$$

where μ is a regularization parameter.

Now, let us set

$$\begin{aligned} m_1 &= n, & m_2 &= 2n, & f_1 &:= \frac{1}{2} \|\cdot - b\|_2^2, & f_2 &:= \mu \psi, \\ A_1 &:= K, & \text{and} & & A_2 &:= B, \end{aligned}$$

where K and b are from Equation (46), ψ is given by Equation (44), and B is defined by Equation (43). Then the L2-TV image deblurring mode (47) can be viewed as a special case of model (1). Therefore, both Algorithms 1 and 2 can be applied for the L2-TV model. Further, we give the explicit forms of the proximity operators $\text{prox}_{\alpha f_1^*}$ and $\text{prox}_{\alpha f_2^*}$ for any positive number α . Actually, using Moreau decomposition and the definition of the proximity operator, we have that for $u \in \mathbb{R}^{m_1}$

$$\text{prox}_{\alpha f_1^*}(u) = \frac{1}{1 + \alpha} u - \frac{\alpha}{1 + \alpha} b.$$

For $v \in \mathbb{R}^{m_2}$, we write $z = \text{prox}_{\alpha f_2^*}(v)$. Then for $i = 1, 2, \dots, n$, we have that

$$[z_i, z_{n+i}] = \min \left\{ \left\| [v_i, v_{n+i}] \right\|_2, \alpha \right\} \frac{[v_i, v_{n+i}]}{\left\| [v_i, v_{n+i}] \right\|_2}. \tag{48}$$



FIGURE 4 | Recovered images of “Cameraman” and “Lena” (from top row to bottom row) with the L1-TV model for images blurred by the (21, 10)-GBM and corrupted by impulsive noise of level $p = 0.3$. Row 1: the CP; Row 2: ZBO; Row 3: Algorithm 1; Row 4: Algorithm 2 with Equation (53); Row 5: Algorithm 2 with Equation (55).

5.2. The L1-TV Image Deblurring Model

The L1-TV image deblurring model is usually used for the recovery of an unknown image $x \in \mathbb{R}^n$ from an impulsive noise corrupted observable data $b \in \mathbb{R}^n$ modeled by

$$b_i = \begin{cases} 0, & \text{with a probability } p/2; \\ 255, & \text{with a probability } p/2; \\ (Kx)_i, & \text{with a probability } 1 - p. \end{cases} \quad (49)$$

where K is a blurring matrix of size $n \times n$, p is the noise level, and the index i runs from 1 to n . The L1-TV image deblurring model has the form of

$$\min\{\|Kx - b\|_1 + \mu\|x\|_{TV} : x \in \mathbb{R}^n\}, \quad (50)$$

where μ is again the regularization parameter.

Now, let us set

$$\begin{aligned} m_1 &= n, & m_2 &= 2n, & f_1 &:= \|\cdot - b\|_1, & f_2 &:= \mu\psi, \\ A_1 &:= K, & \text{and } A_2 &:= B, \end{aligned}$$

where K and b are from Equation (49), ψ is given by Equation (44), and B is defined by Equation (43). Then the L1-TV image deblurring mode Equation (50) can be viewed as a special case of model (1). Therefore, both Algorithms 1 and 2 can be applied for the L1-TV model. Further, the proximity operator $\text{prox}_{\alpha f_2^*}$ has been given via Equation (48). We just need to present the proximity operator $\text{prox}_{\alpha f_1^*}$. Actually, we have that for $u \in \mathbb{R}^{m_1}$

$$(\text{prox}_{\alpha f_1^*}(u))_i = \begin{cases} \text{sign}(u_i - \alpha b_i), & \text{if } |u_i - \alpha b_i| \geq 1; \\ u_i - \alpha b_i, & \text{otherwise,} \end{cases}$$

where $i = 1, 2, \dots, n$.

In summary, for both the L2-TV and L1-TV image deblurring models, the associated proximity operators $\text{prox}_{\alpha f_1^*}$ and $\text{prox}_{\alpha f_2^*}$ have closed forms. As a consequence, the sequence $\{(u^k, v^k, x^k) : k \in \mathbb{N}\}$ generated by Algorithms 1 and 2 can be efficiently computed.

6. NUMERICAL EXPERIMENTS

In this section, numerical experiments for image deblurring are carried out to demonstrate the performance of our proposed Algorithms 1 and 2 for the 256×256 test images ‘‘Cameraman,’’ ‘‘Peppers,’’ ‘‘Goldhill,’’ and 512×512 test image ‘‘Lena.’’ The Chambolle-Pock (CP) algorithm and scheme (6) are compared to our algorithms for the L2-TV and L1-TV image deblurring models. In the following, we quote scheme (6) as ZBO algorithm. Each algorithm is carried out until the stopping criterion $\|x^{k+1} - x^k\|^2 / \|x^k\|^2 \leq Tol$ is satisfied, where Tol representing the tolerance, is chosen to be 10^{-6} . The quality of the recovered images from each algorithm is evaluated by the peak-signal-to-noise ratio (PSNR), which is defined as $\text{PSNR} := 20 \log_{10}(\frac{255n}{\|\tilde{x} - x\|_2})$, where $x \in \mathbb{R}^n$ is the original image and \tilde{x} represents the recovered image. The evolution curve of the function values with respect to iteration will be also adopted to evaluate the performance of algorithms.

In our simulations, blurring matrices K in models (46) and (49) are generated by a rotationally symmetric Gaussian lowpass filter of size ‘‘hsize’’ with standard deviation ‘‘sigma’’ from the MATLAB script `fspecial('gaussian', hsize, sigma)`. Such matrix K is referred to as the $(hsize, sigma)$ -GBM. We remark that the norm of K is always 1, i.e.,

$$\|K\| = 1. \quad (51)$$

The (15, 10)-GBM and (21, 10)-GBM are used to generate blurred images in our experiments. All experiments are performed in Matlab 7.11 on Dell Desktop Inspiron 620 with Intel Core i3 CPU @3.30G, 4GB RAM on Windows 7 Home Premium operating system.

6.1. Parameter Settings

Prior to applying Algorithms 1 and 2, and the CP method to the L2-TV model and the L1-TV model for image deblurring problems blurred by $(hsize, sigma)$ -GBMs, the parameters arising from these algorithms need to be determined. Convergence

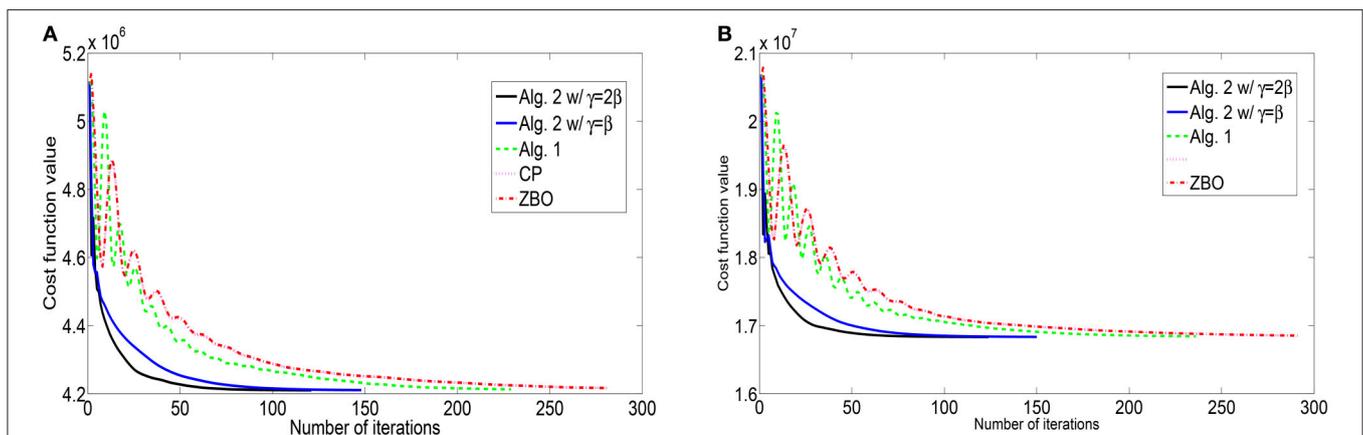


FIGURE 5 | Evolutions of function values of the L1-TV model for images of (A) ‘‘Cameraman’’ and (B) ‘‘Lena.’’

analysis of the algorithms specifies the relation between these parameters. Therefore, once one parameter is fixed, the others can be described by this fixed one. To this end, we fix the value of the parameter β in each above algorithm and then figure out the values of the others.

Let K be a rotationally symmetric Gaussian lowpass filter generated blurring matrix in the L2-TV model and the L1-TV model and let B be the difference matrix defined by Equation (43). We know that $\|K\| = 1$ by Equation (51) and $\|B\| < \sqrt{8}$. To compute the pixel values under the operation of K and B near the boundary of images, we choose to use “symmetric” type for the boundary extension. Correspondingly, we can compute that

$$\left\| \begin{bmatrix} K \\ B \end{bmatrix} \right\|^2 < 8.$$

For Algorithm 1, we set the parameters α and γ as follows:

$$\alpha = \frac{1}{8\beta} \quad \text{and} \quad \gamma = 2\beta. \quad (52)$$

For Algorithm 2, we set the parameters α_1 , α_2 , and γ as follows:

$$\alpha_1 = \frac{0.999}{\beta}, \quad \alpha_2 = \frac{1}{8\beta}, \quad \text{and} \quad \gamma = \beta. \quad (53)$$

For the CP method (see Equation 39), we set

$$\alpha = \frac{1}{4\beta}. \quad (54)$$

With the choices of the parameters given for the algorithms, the convergence of Algorithm 1, Algorithm 2, and the CP method are guaranteed by Theorem 3.5, Theorem 3.3, and a result from Chambolle and Pock [9], respectively. The parameter β in each algorithm is chosen in a way that it would produce better recovered images in terms of PSNR value with the given stopping criterion.

An additional set of parameters for Algorithm 2 will be used as well. They are

$$\alpha_1 = \frac{0.999}{\beta}, \quad \alpha_2 = \frac{1}{8\beta}, \quad \text{and} \quad \gamma = 2\beta. \quad (55)$$

By comparing these parameters with those in Equation (53), the parameters α_1 and α_2 are kept unchanged, but the parameter γ in Equation (55) is twice bigger than that in Equation (53). By Theorem 3.3, although the convergence analysis for

Algorithm 2 with the parameters given by Equation (55) is missing currently, our numerical experiments presented in the rest of the paper indicate that Algorithm 2 converges and usually produces better recovered images than that with the parameters given by Equation (53) in terms of the CPU times consumed.

6.2. Numerical Results for The L2-TV Image Deblurring

In problems of image deblurring with the L2-TV model, a noisy image is obtained by blurring an ideal image with a ($hsize, sigma$)-GBM followed by adding white Gaussian noise. Two blurring matrices, namely (21, 10)-GBM and (15, 10)-GBM, are used in our experiments.

For the blurring matrix (21, 10)-GBM, the white noise with mean zero and the standard deviation 1 is added to blurred images. We set the regularization parameter $\mu = 0.02$ in the L2-TV model (47). We choose $\beta = 50$ for Algorithm 2, $\beta = 25$ for Algorithm 1, $\beta = 50$ for the CP method, and $\beta = 0.005$ for ZBO method. With these settings, numerical results for four test images “Cameraman,” “Lena,” “Peppers,” and “Goldhill” are reported in **Table 1** in terms of numbers of iterations, the CPU times, and the PSNR values. The evolutions of function values for the images of “Peppers” and “Lena” with the L2-TV model are shown in **Figure 1**. The corresponding curves for the images of “Peppers,” and “Goldhill” are similar to that of “Peppers,” therefore are omitted here. As shown in the Table, Algorithm 2 performs best in terms of computational cost (total iterations and CPU time) and PSNR. Also, as shown in **Figure 1** the sequence of function values generated by Algorithm 2 approaches the minimum value fastest, followed by sequences from Algorithm 1 and then by that from CP and ZBO. For the two versions of Algorithm 2, the one with Equation (55) performs slightly better in terms of computational cost and PSNR. The performance of CP and ZBO methods is quite similar in terms of iterations, CPU time, PSNR and evolution of function values. Indeed, the iterations and PSNR in **Table 1** are consistent for CP and ZBO methods and the evolution curves of function values for CP and ZBO methods overlap together. We comment that comparable PSNRs and function value can be achieved in Algorithm 1, CP method and ZBO method as in Algorithm 2 if more iteration or computational time is allowed.

For the blurring matrix (15, 10)-GBM, the white noise with mean zero and the standard deviation 5 is added to blurred

TABLE 4 | Numerical results for the L1-TV model for images blurred by the (15, 10)-GBM.

Method	Cameraman			Lena			Peppers			Goldhill		
	Itrs	CPU(s)	PSNR	Itrs	CPU(s)	PSNR	Itrs	CPU(s)	PSNR	Itrs	CPU(s)	PSNR
CP	280	29.78	23.77	289	200.71	27.15	300	31.85	25.75	294	32.01	25.15
ZBO	281	31.51	23.77	290	209.85	27.15	300	35.82	25.74	295	32.82	25.15
Algorithm 1	228	24.14	24.20	235	170.57	27.55	250	26.59	26.28	247	26.56	25.52
Algorithm 2 with Equation (53)	147	16.10	24.42	149	108.95	27.89	153	16.37	26.61	151	16.06	25.76
Algorithm 2 with Equation (55)	120	12.79	24.71	123	88.68	28.11	127	14.48	27.01	127	13.78	25.94



images. We set the regularization parameter $\mu = 0.2$ in the L2-TV model (47). We choose $\beta = 10$ for Algorithm 2, $\beta = 5$ for Algorithm 1, $\beta = 10$ for the CP method, and $\beta = 0.025$ for ZBO method. With these settings, numerical results for four test images are reported in **Table 2** in terms of numbers of iterations, the CPU times, and the PSNR values. For each image, the PSNR values from each algorithm are comparable. But Algorithm 2 performs better than Algorithm 1, CP and ZBO in terms of computational cost. The evolutions of function values for the images of “Peppers” and “Lena” are shown in **Figure 2**. The sequence of function values from Algorithm 2 approaches faster to the minimum function value than that from CP and ZBO method. The performance of the two versions of Algorithm 2 is similar. Like in the setting (21, 10)-GBM, the performance of CP and ZBO methods is similar.

6.3. Numerical Results for The L1-TV Image Deblurring

In problems of image deblurring with the L1-TV model, a noisy image is obtained by blurring an ideal image with a ($hsize$, $sigma$)-GBM followed by adding impulsive noise. Two blurring matrices, namely (21, 10)-GBM and (15, 10)-GBM, are used again in our experiments.

For the blurring matrix (21, 10)-GBM, the impulsive noise with noise level $p = 0.3$ is added to blurred images. We set the regularization parameter $\mu = 0.01$ in the L1-TV model (50). we set $\beta = 100$ for Algorithm 2, $\beta = 50$ for Algorithm 1, $\beta = 100$ for the CP method, and $\beta = 0.0025$ for ZBO method. With these settings, numerical results for four test images “Cameraman,” “Lena,” “Peppers,” and “Goldhill” are reported in **Table 3** in terms of numbers of iterations, the CPU times, and the PSNR values. Algorithm 2 yields higher PSNR value but consumes less CPU time than Algorithm 1, CP and ZBO methods. The evolution curves of function values with respect to iteration for the images of “Cameraman” and “Lena” are shown in **Figure 3**. It can be noticed that sequence of function values generated by Algorithm 2 approaches the minimum value fastest. Regarding the two version of Algorithm 2, the one with setting Equation (55) performs better. We point out that the evolution curves for CP and ZBO overlap with each other. Further, visual quality of the deblurred images of “Cameraman” and “Lena” is shown in **Figure 4** for each algorithm. The visual improvement by Algorithm 2 over CP and the ZBO can be seen by the deblurred images.

For the blurring matrix (15, 10)-GBM, the impulsive noise with noise level $p = 0.5$ is added to blurred images. We set the regularization parameter $\mu = 0.02$ in the L1-TV model (50). we set $\beta = 50$ for Algorithm 2, $\beta = 25$ for Algorithm 1, $\beta = 50$ for the CP method, and $\beta = 0.005$ for ZBO method. With these settings, numerical results for four test images “Cameraman,” “Lena,” “Peppers,” and “Goldhill” are reported in **Table 4** in terms of numbers of iterations, the CPU times, and the PSNR values. The evolution curves of function values with respect to iteration for the images of “Cameraman”

and “Lena” are shown in **Figure 5**. Visual quality of the deblurred “Cameraman” and “Lena” images is shown in **Figure 6** for each algorithm. It is the same as above that Algorithm 2 performs the best.

7. CONCLUSION

We propose algorithms to solve a general problem that includes L2-TV and L1-TV image deblurring problems. Algorithm with Block Gauss-Seidel acceleration is also derived for the two term composite minimization Equation (1). The key feature of the proposed algorithms is their ability to yield closed form in each step of iterations. Convergence analysis of the proposed

algorithms can be guaranteed under appropriate conditions. Numerical experiments show that the proposed algorithms has computational advantage over CP method and ZBO algorithm.

AUTHOR CONTRIBUTIONS

All authors listed, have made substantial, direct and intellectual contribution to the work, and approved it for publication.

FUNDING

The research was supported by the US National Science Foundation under grant DMS-1522332.

REFERENCES

- Zhang X, Burger M, Osher S. A unified primal-dual algorithm framework based on Bregman iteration. *J Sci Comput.* (2011) **46**:20–46. doi: 10.1007/s10915-010-9408-8
- Chen P, Huang J, Zhang X. A primal-dual fixed point algorithm for convex separable minimization with applications to image restoration. *Inverse Problems* (2013) **29**:025011. doi: 10.1088/0266-5611/29/2/025011
- Deng W, Yin W. On the global and linear convergence of the generalized alternating direction method of multipliers. *J Sci Comput.* (2016) **66**:889–916. doi: 10.1007/s10915-015-0048-x
- Gabay D, Mercier B. A dual algorithm for the solution of nonlinear variational problems via finite-element approximations. *Comput Math Appl.* (1976) **2**:17–40. doi: 10.1016/0898-1221(76)90003-1
- Glonwinski R, Tallec PL. *Augmented Lagrangians and Operator Splitting Methods in Nonlinear Mechanics*. Philadelphia, PA: SIAM (1989). doi: 10.1137/1.9781611970838
- Hestenes MR. Multiplier and gradient methods. *J Optim Theory Appl.* (1969) **4**:303–20. doi: 10.1007/BF00927673
- Powell MJD. A method for nonlinear constraints in minimization problems. In: Fletcher R, editor, *Optimization*. New York, NY: Academic Press (1969). p. 283–98.
- Rockafellar RT. The multiplier method of Hestenes and Powell applied to convex programming. *J Optim Theory Appl.* (1973) **12**:555–62. doi: 10.1007/BF00934777
- Chambolle A, Pock T. A first-order primal-dual algorithm for convex problems with applications to imaging. *J Mat Imaging Vision* (2011) **40**:120–45. doi: 10.1007/s10851-010-0251-1
- Rockafellar RT, Wets R-JB. *Variational Analysis*. New York, NY: Springer (1998).
- Arrow KJ, Harwicz L, Uzawa H. *Studies in Linear and Non-linear Programming. Stanford Mathematical Studies in the Social Science, II*. Stanford, CA: Stanford University Press (1958).
- Esser E, Zhang X, Chan TF. A general framework for a class of first order primal-dual algorithms for convex optimization in imaging science. *SIAM J Imaging Sci.* (2010) **3**:1015–46. doi: 10.1137/09076934X
- Moreau JJ. Fonctions convexes duales et points proximaux dans un espace hilbertien. *CR Acad Sci Paris Sér A Math.* (1962) **255**:1897–2899.
- Moreau JJ. Proximité et dualité dans un espace hilbertien. *Bull Soc Math France* (1965) **93**:273–99.
- Rockafellar RT. *Convex Analysis*. Princeton, NJ: Princeton University Press (1970).
- Bertsekas D. *Nonlinear Programming*. Belmont, MA: Athena Scientific (2003).
- Micchelli CA, Shen L, Xu Y. Proximity algorithms for image models: denoising. *Inverse Problems* (2011) **27**:30. doi: 10.1088/0266-5611/27/4/045009

Disclaimer: Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of AFRL.

Conflict of Interest Statement: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Copyright © 2016 Chen, Shen, Suter and Xu. This is an open-access article distributed under the terms of the Creative Commons Attribution License (CC BY). The use, distribution or reproduction in other forums is permitted, provided the original author(s) or licensor are credited and that the original publication in this journal is cited, in accordance with accepted academic practice. No use, distribution or reproduction is permitted which does not comply with these terms.