Algorithmic Versatility of SPF-regularization Methods

Lixin Shen* Bruce W. Suter† Erin E. Tripp‡

March 11, 2020

Abstract

Sparsity promoting functions (SPFs) are commonly used in optimization problems to find solutions which are sparse in some basis. For example, the ℓ_1-regularized wavelet model and the Rudin-Osher-Fatemi total variation (ROF-TV) model are some of the most well-known models for signal and image denoising, respectively. However, recent work demonstrates that convexity is not always desirable in sparsity promoting functions. In this paper, we replace convex SPFs with their induced nonconvex SPFs and develop algorithms for the resulting model by exploring the intrinsic structures of the nonconvex SPFs. These functions are defined as the difference of the convex SPF and its Moreau envelope. We also present simulations illustrating the performance of a special SPF and the developed algorithms in image denoising.

Keywords: Nonlinear optimization; sparse optimization; variational models; image denoising.

Mathematics Subject Classification 2000: 68U10, 65T60.

1 Introduction

Sparsity is a crucial assumption in various applications ranging from signal processing to machine learning and statistics. The widespread interest in sparsity can be attributed to the fact that (i) sparsity infers intrinsic structures of data and (ii) sparse data is easier to manipulate and interpret. Informally, data in the form of vector or matrix is sparse if it contains few nonzero entries. The natural mathematical measure of sparsity is the so-called "ℓ_0-norm", which counts the number of nonzero entries in a vector. In the context of optimization, this measure can be viewed as a penalty on non-sparse solutions, and it is in this context that we call the ℓ_0-norm a sparsity promoting function (SPF). However, solving ℓ_0-regularized optimization problems is known to be NP-hard. To overcome this difficulty, the ℓ_1-regularization methods such as least absolute shrinkage and selection operator (LASSO) [23] and Dantzig selectors [4] have been proposed. This relaxation allows application of the many tools of convex analysis, making the problem numerically tractable, but it also introduces bias by heavily penalizing entries with large magnitude. To address this, nonconvex penalties have been proposed to replace the ℓ_1-penalty, including the ℓ_p-norm with $0 < p < 1$ [5, 11], the smoothly clipped absolute deviation penalty [10], the continuous exact ℓ_0 penalty [21], and the minimax concave penalty (MCP) [27]. There is increasing evidence that supports the use of nonconvex penalties in many applications, see, for example [1, 14, 25] and the references therein. Like the ℓ_0-norm, these penalty functions are all widely accepted as SPF, and, as noted in [10], they all share certain essential properties.

*Department of Mathematics, Syracuse University, Syracuse, NY 13244, USA. Email: lshen03@syr.edu

†Air Force Research Laboratory, Rome, NY. Email: bruce.suter.1@us.af.mil.

‡Air Force Research Laboratory, Rome, NY. Email: erin.tripp.4@us.af.mil

Based on these observations, we have attempted to give a formal mathematical definition of SPFs in our recent work [20]. Loosely speaking, a function is an SPF if its subdifferential at the origin contains the origin and at least one other element; that is, an SPF has a corner or cusp at the origin. Viewed another way, the subdifferential of the function at the origin is a set which defines a threshold for "small" entries which are considered noise. The proximity operator of the SPF will send all elements under this threshold to the origin. Fortunately, all of the above penalties fit this definition.

In [20], we introduced a family of SPFs each of which is the difference of a convex SPF with its Moreau envelope. Functions in this family have the desired nonconvexity but enough useful properties to develop efficient algorithms for optimization problems penalized by these SPFs. These functions are non-negative, semiconvex, and a special case of difference of convex functions with one term having a Lipschitz continuous gradient. Due to these properties, we refer to these functions as structured SPFs. As an example, the MCP is a particular instance of this construction. Many other examples and interesting properties of the structured SPFs can be found in [20].

The goal of this paper is to demonstrate the applicability of structured SPFs to a variety of optimization models. To illustrate these ideas, we consider the regularized least squares model:

$$\operatorname{argmin}\left\{\frac{1}{2\lambda}\|x-z\|^2+(\Phi\circ B)(x):x\in C\right\}, \tag{1}$$

where C is a closed convex subset of $\mathbb{R}^d$, λ is a regularization parameter, $z\in\mathbb{R}^d$, $B\in\mathbb{R}^{n\times d}$, and Φ is a sparsity promoting function on $\mathbb{R}^n$. We note that all of the discussion and results below hold true if the quadratic term is replaced by a differentiable strongly convex function. We simply choose this model as our prototype because of its simplicity and its applicability. Problems of interest in the context of image/signal processing at large can be formulated as finding a solution to (1). For example, if z is an image corrupted by Gaussian noise, and $\Phi\circ B$ is a composition of the ℓ_2-norm with the two-dimensional first order difference operator, model (1) reduces to the well known Rudin-Osher-Fatemi total variation (ROF-TV) model. If Φ is the ℓ_1-norm and B is formulated from a tight framelet, then the resulting model (1) was discussed in [19].

We propose replacing convex Φ with the structured SPFs. The flexibility provided by these functions allows us to approach (1) from several perspectives and to make use of algorithmic advances in convex, difference of convex, and nonconvex optimization. In each case, we are able to see how the structures of the SPF plays out in the algorithms and to what effect. More precisely, three different algorithms for model (1) will be proposed by fully employing the various properties of Φ. The first algorithm explores the semiconvexity property of Φ to identify the objective function of (1) in a form which can be optimized by the primal-dual splitting algorithm in [7]. The second algorithm is based on the natural difference of convex form of Φ, by its design, so that the difference of convex algorithm (DCA), e.g., in [13, 14, 22], can be applied directly. As shown in our previous work [20], the proximity operators of many constructed structured SPFs have explicit expressions available, but, not utilizing it in the development of the above two algorithms. The third algorithm makes use of the explicit form of the proximity operator of Φ. The convergence analysis of these algorithms and their applications in image denoising will be provided. Numerical results demonstrate increased accuracy without additional computational time in many instances.

The rest of the paper is organized in the following manner. In the next section we recall some necessary background in optimization, briefly review the definition of structured SPFs, and point out some properties of these functions that will be explored in the development of algorithms suitable for model (1). We also give examples that can be used in the model (1) for image denoising application. In Section 3, the properties of structured SPFs are used to develop efficient algorithms

for model (1). Numerical experiments are conducted in Section 4 to demonstrate the performance of the developed algorithms in image denoising. Our conclusions are drawn in Section 5.

2 Structured Sparsity Promoting Functions

In this section, we define precisely what we mean by sparsity promoting functions (SPFs) as well as the family of structured SPFs introduced in our recent paper [20]. The relevant results to this paper are included here for completeness and two examples of interest are provided in detail.

We begin by introducing our notation. We denote by $\mathbb{R}^d$ the usual d-dimensional Euclidean space equipped with the standard inner product $\langle \cdot, \cdot \rangle$ and the induced Euclidean norm $\| \cdot \|$. A function p defined on $\mathbb{R}^d$ with values in $\mathbb{R} \cup \{+\infty\}$ is proper if its domain $\mathrm{dom}(p) = \{x \in \mathbb{R}^d : p(x) < +\infty\}$ is nonempty, and p is lower semicontinuous if its epigraph is a closed set. The set of proper and lower semicontinuous functions on $\mathbb{R}^d$ to $\mathbb{R} \cup \{+\infty\}$ is denoted by $\Gamma(\mathbb{R}^d)$. The set of proper, convex, and lower semicontinuous functions on $\mathbb{R}^d$ to $\mathbb{R} \cup \{+\infty\}$ is denoted by $\Gamma_0(\mathbb{R}^d)$.

The subdifferential and proximity operator of a lower semicontinuous function are two important concepts in nonlinear optimization. We review some aspects of these concepts that are needed in this paper. Recall that the Fréchet subdifferential of a function $p : \mathbb{R}^d \to \mathbb{R} \cup \{+\infty\}$ at $z \in \mathbb{R}^d$, denoted by $\partial p(z)$, is defined as

$$\partial p(z) := \left\{ t \in \mathbb{R}^d : \liminf_{u \to z} \frac{p(u) - p(z) - \langle t, u - z \rangle}{\|u - z\|} \geq 0 \right\}.$$

The set $\partial p(z)$ is closed and convex. If $\partial p(z) \neq \emptyset$, we say that p is Fréchet subdifferentiable at z. If p is convex, then $\partial p(z) := \left\{ t \in \mathbb{R}^d : p(u) - p(z) \geq \langle t, u - z \rangle,\ u \in \mathbb{R}^d \right\}$. If p is Fréchet differentiable at z with a derivative, then $\partial p(z) = \{\nabla p(z)\}$.

We further review some useful simple calculus results for Fréchet subdifferentials. If a function $p : \mathbb{R}^d \to \mathbb{R} \cup \{+\infty\}$ attains its local minimum at $z \in \mathbb{R}$, then $0 \in \partial p(z)$ and the point z is called a critical point of p. For any $\alpha > 0$, it holds that $\partial(\alpha p)(z) = \alpha \partial p(z)$. For any functions $p_1 : \mathbb{R}^d \to \mathbb{R} \cup \{+\infty\}$ and $p_2 : \mathbb{R}^d \to \mathbb{R} \cup \{+\infty\}$ Fréchet subdifferentiable at z, then $p_1 + p_2$ is Fréchet subdifferential at z and $\partial(p_1 + p_2)(z) \subseteq \partial p_1(z) + \partial p_2(z)$. If one of the above functions is Fréchet differentiable at z, say p_1, then $\partial(p_1 + p_2)(z) = \nabla p_1(z) + \partial p_2(z)$.

The proximity operator was introduced by Moreau in [17, 18]. For a function $p \in \Gamma(\mathbb{R}^d)$, the proximity operator of p at $z \in \mathbb{R}^d$ with index α is defined by

$$\mathrm{prox}_{\alpha p}(z) := \arg\min \left\{ p(w) + \frac{1}{2\alpha} \|w - z\|^2 : w \in \mathbb{R}^d \right\}.$$

The proximity operator of p is a set-valued operator from $\mathbb{R}^d \to 2^{\mathbb{R}^d}$, the power set of $\mathbb{R}^d$. Clearly, for any $w^\star \in \mathrm{prox}_{\alpha p}(z)$, by the calculus of Fréchet subdifferential, we have that

$$\frac{1}{\alpha}(z - w^\star) \in \partial p(w^\star). \tag{2}$$

The Moreau envelope of p at $z \in \mathbb{R}^d$ with index α, denoted by $\mathrm{env}_\alpha p(z)$ is closely related to the proximity operator $\mathrm{prox}_p(z)$. That is,

$$\mathrm{env}_\alpha p(z) := p(w^\star) + \frac{1}{2\alpha} \|w^\star - z\|^2,$$

where $w^\star$ is in $\text{prox}_{\alpha p}(z)$. If p is convex, then the proximity operator of p is a single-valued operator from $\mathbb{R}^d \to \mathbb{R}^d$. Furthermore, equation (2) becomes

$$\frac{1}{\alpha}(\text{Id} - \text{prox}_{\alpha p})(z) \in \partial p(\text{prox}_{\alpha p}(z)).$$

With this preparation, we are now able to give our definition of sparsity promoting functions and describe some of their properties.

Definition 1. *We say a function $\varphi \in \Gamma(\mathbb{R}^d)$ is sparsity promoting if (i) φ achieves its global minimum of zero at the origin and (ii) there is a nonzero element in $\partial\varphi(0)$. Denote by* $\text{SPF}(\mathbb{R}^d)$ *the set of sparsity promoting functions on $\mathbb{R}^d$.*

The first item of Definition 1 ensures that nonzero entries are penalized. The second item describes the necessary sharpness of SPF, and the set $\partial\varphi(0)$ defines what is considered small and therefore what should be sent to zero. For example, if $\varphi \in \Gamma_0(\mathbb{R}^d)$, then (2) becomes

$$\text{prox}_{\alpha\varphi}(x) = 0 \iff x \in \alpha\partial\varphi(0).$$

We note that all of the penalties discussed above satisfy this definition, as does any norm on $\mathbb{R}^d$.

Now for any convex $\varphi \in \text{SPF}(\mathbb{R}^n)$ and any $\alpha > 0$, we define

$$\varphi_\alpha = \varphi - \text{env}_\alpha \varphi. \tag{3}$$

By construction, φ_α is a nonnegative difference of convex functions. We summarize relevant properties of φ_α below. Based on these properties, we refer to these functions as structured SPF's.

Lemma 1. *Given a convex function $\varphi \in \text{SPF}(\mathbb{R}^d)$ and $\alpha > 0$, the function φ_α defined by* (3) *has the following properties:*

(i) $\varphi_\alpha \in \text{SPF}(\mathbb{R}^d)$ with $\partial\varphi_\alpha(0) = \partial\varphi(0)$;

(ii) φ_α is $\frac{1}{\alpha}$-semiconvex, i.e. $\varphi_\alpha + \frac{1}{2\alpha}\|\cdot\|^2$ is convex;

(iii) given $B \in \mathbb{R}^{n\times d}$, $\varphi_\alpha \circ B$ is $\frac{\|B\|^2}{\alpha}$-semiconvex.

Proof. The proofs of items (i) and (ii) can be found in [20]. We now turn to prove item (iii). Define $\psi = \varphi_\alpha + \frac{1}{2\alpha}\|\cdot\|^2$. Then, for any $x \in \mathbb{R}^n$

$$\varphi_\alpha \circ B(x) + \frac{\|B\|^2}{2\alpha}\|x\|^2 = \psi(Bx) + \frac{\|B\|^2}{2\alpha}\|x\|^2 - \frac{1}{2\alpha}\|Bx\|^2. \tag{4}$$

By item (ii), $\psi \circ B$ is convex. Note that $\|B\|^2\|x\|^2 - \|Bx\|^2 = x^\top(\|B\|^2\,\text{Id} - B^\top B)x \geq 0$ for all $x \in \mathbb{R}^d$, so it is convex. Hence, $\varphi_\alpha \circ B + \frac{\|B\|^2}{2\alpha}\|\cdot\|^2$ is convex, which implies that item (iii) holds. □

One benefit of Definition 1 is that it is sufficiently general to encompass many examples. In practice, we often require more of φ than convexity and can therefore specify further properties of φ_α. Properties such as separability or block-separability are assumed to control the fineness of sparsity enforcement, and convergence analysis may rely on the function being continuous or subanalytic. In each of these cases, φ_α inherits the given properties. This is evident in the following examples.

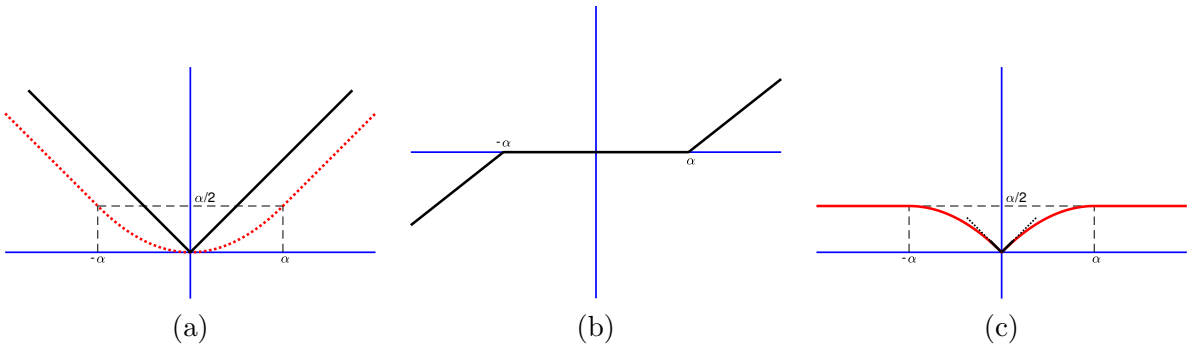
2.1 Example 1: φ is the absolute value function

Relying on the separability of the ℓ_1-norm, we simply let φ be the absolute value function, that is, $\varphi(x) = |x|$ on $\mathbb{R}$. Clearly, because φ achieves its minimum at the origin and $\partial\varphi(0) = [-1, 1]$, the absolute value function is sparsity promoting. The proximity operator and the Moreau envelope of $|\cdot|$ with parameter $\alpha > 0$ are

$$\mathrm{prox}_{\alpha|\cdot|}(x) = \mathrm{sgn}(x)\max\{0, |x| - \alpha\} \quad \text{and} \quad \mathrm{env}_{\alpha}|\cdot|(x) = \begin{cases} \frac{1}{2\alpha}x^2, & \text{if } |x| \le \alpha; \\ |x| - \frac{1}{2}\alpha, & \text{otherwise,} \end{cases}$$

respectively. It is well known that $\mathrm{prox}_{\alpha|\cdot|}$ is called the soft thresholding operator in wavelet literature [9] and $\mathrm{env}_{\alpha}|\cdot|$ is Huber's function in robust statistics [12]. We note that for $x \in \mathbb{R}^d$, $\mathrm{prox}_{\alpha\|\cdot\|_1}(x) = \mathrm{prox}_{\alpha|\cdot|}(x_1) \times \cdots \times \mathrm{prox}_{\alpha|\cdot|}(x_d)$ and $\mathrm{env}_{\alpha}\|\cdot\|_1(x) = \sum_{i=1}^{d} \mathrm{env}_{\alpha}|\cdot|(x_i)$.

Figure 1(a) depicts the graphs of $|\cdot|$ (solid line) and its Moreau envelope (dotted line) while Figure 1(b) shows the graph of its the proximity operator.



(a) (b) (c)

Figure 1: Let $\varphi = |\cdot|$ be the absolute value function. (a) The graphs of φ (solid), $\mathrm{env}_{\alpha}\varphi$ (dotted); (b) The typical shape of $\mathrm{prox}_{\alpha\varphi}$; and (c) the graph of $\varphi_\alpha = \varphi(x) - \mathrm{env}_{\alpha}\varphi(x)$. Near the origin φ_α retains the structure of φ, which is emphasized in black (solid-dotted).

As defined in (3), for the absolute value function φ,

$$\varphi_\alpha(x) := |x| - \mathrm{env}_{\alpha}|\cdot|(x) = \begin{cases} |x| - \frac{1}{2\alpha}x^2, & \text{if } |x| \le \alpha; \\ \frac{1}{2}\alpha, & \text{otherwise.} \end{cases}$$

This function φ_α (see Figure 1(c)) is identical to the minimax convex penalty (MCP) function given in [27], but motivated from a statistical perspective. The expression of $\mathrm{prox}_{\beta\varphi_\alpha}$ depends on the relative values of α and β, and takes the form as follows (see [20]):

$$\mathrm{prox}_{\beta\varphi_\alpha}(x) = \begin{cases} \frac{\alpha}{\alpha-\beta}(|x| - \beta)\cdot\mathrm{sgn}(x)\cdot\max\{|x| - \beta, 0\}\cdot\chi_{\{|x|\le\alpha\}} + \{x\}\cdot\chi_{\{|x|>\alpha\}}, & \text{if } \beta < \alpha; \\ \{0\}\cdot\chi_{\{|x|<\alpha\}} + \mathrm{sgn}(x)\cdot[0, \alpha]\cdot\chi_{\{|x|=\alpha\}} + \{x\}\cdot\chi_{\{|x|>\alpha\}}, & \text{if } \beta = \alpha; \\ \{0\}\cdot\chi_{\{|x|<\alpha\}} + \mathrm{sgn}(x)\cdot\{0, \sqrt{\alpha\beta}\}\cdot\chi_{\{|x|=\sqrt{\alpha\beta}\}} + \{x\}\cdot\chi_{\{|x|>\sqrt{\alpha\beta}\}}, & \text{if } \beta > \alpha. \end{cases} \tag{5}$$

Here χ_S has value 1 at points of the set S, and 0 at points of $\mathbb{R} \setminus S$. The graphs of $\mathrm{prox}_{\beta\varphi_\alpha}$ for different values of α and β are plotted in Figure 2. It is straightforward to extend this to $\mathbb{R}^d$. In fact, we have $(\|\cdot\|_1)_\alpha(x) = \sum_{i=1}^{d} \varphi_\alpha(x_i)$.

2.2 Example 2: φ is a compositional norm

The example here is motivated from the total variation that will be defined in Section 4. To define this function, let the disjoint sets ω_j, $j = 1, \ldots, J$ be the partition of the set $\{1, 2, \ldots, d\}$, that is

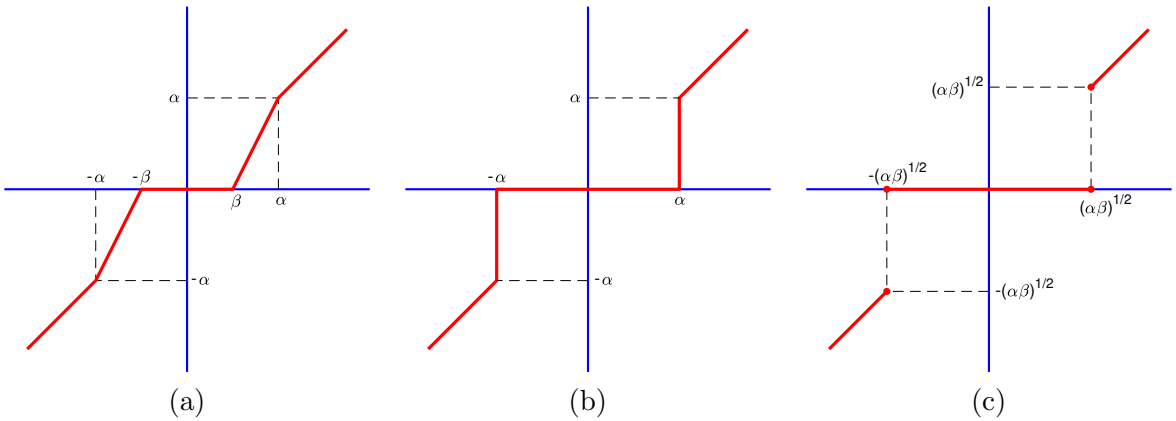


Figure 2: Typical shapes of the proximity operator of $|\cdot|_\alpha$ for (a) $\beta < \alpha$, (b) $\beta = \alpha$, (c) $\beta > \alpha$. The sparsity threshold and the thresholding behavior depend on the relationship between α and β.

$\cup_{j=1}^J \omega_j = \{1, 2, \ldots, d\}$; and let I_{ω_j} be the $\#\omega_j \times d$ matrix formed by those rows of the $d \times d$ identity matrix with indices in ω_j. Since $I_{\omega_j} x$ for $x \in \mathbb{R}^d$ is the vector whose entries are those of x with indices in ω_j, we call I_{ω_j} an extraction matrix. With these preparation, in the second example, we will consider the following function: for $x \in \mathbb{R}^d$

$$\varphi(x) = \sum_{j=1}^{J} \|I_{\omega_j} x\|. \tag{6}$$

It is not difficult to show that φ in (6) is a norm of $\mathbb{R}^d$ (associated with the given partition). In [3], this φ is referred to as a compositional norm since it is a norm composed of norms over disjoint sets of variables.

For this example, we will show that φ in (6) is a sparsity promoting function and φ_α can be presented in terms of $|\cdot|_\alpha$ from Example 1. Indeed, the following result says that φ in (6) is a sparsity promoting function.

Proposition 1. *Let φ be a compositional norm on $\mathbb{R}^d$, as defined by* (6)*. Then $\varphi \in \mathrm{SPF}(\mathbb{R}^d)$.*

Proof. Clearly, $\varphi(0) = 0$ and φ achieves its global minimum at the origin. Further, we have $\partial\varphi(x) = \sum_{j=1}^J I_{\omega_j}^\top \partial\|\cdot\|(I_{\omega_j} x)$. Since $\partial\|\cdot\|(I_{\omega_j} 0)$ is the unit ball of $\mathbb{R}^{\#\omega_j}$, we know that $\partial\varphi(0)$ contains nonzero elements, so φ is an SPF. □

To compute φ_α and its proximity operator, we need the following lemma, which can be viewed as an extension of the first example.

Lemma 2. *Let f be the ℓ_2-norm on $\mathbb{R}^d$, that is, $f = \|\cdot\|$. Then, it holds that for any nonzero $x \in \mathbb{R}^d$ and two positive parameters α and β*

$$\mathrm{prox}_{\beta f}(x) = \mathrm{prox}_{\beta|\cdot|}(\|x\|)\frac{x}{\|x\|} \quad \textit{and} \quad \mathrm{prox}_{\beta f_\alpha}(x) = \mathrm{prox}_{\beta|\cdot|_\alpha}(\|x\|)\frac{x}{\|x\|}$$

with the convention $\frac{0}{\|0\|} = 0$.

Proof. The derivation of $\mathrm{prox}_{\beta f}$ can be found in [2]. A direct computation (for example, see [16]) gives $\mathrm{env}_\alpha f(x) = \mathrm{env}_\alpha|\cdot|(\|x\|)$. Therefore, $f_\alpha(x) = \|x\| - \mathrm{env}_\alpha|\cdot|(\|x\|) = |\cdot|_\alpha(\|x\|)$. This function

is isotropic, meaning it depends only on the magnitude of its argument and not the direction. Then for any $x \in \mathbb{R}^d$, every element of $\mathrm{prox}_{\beta f_\alpha}(x)$ should be a multiple of $\frac{x}{\|x\|}$. Moreover, we have

$$\begin{aligned}\mathrm{prox}_{\beta f_\alpha}(x) &= \arg\min\left\{f_\alpha(w) + \frac{1}{2\beta}\|w-x\|^2 : w \in \mathbb{R}^d\right\} \\ &= \frac{x}{\|x\|} \cdot \arg\min\left\{|\cdot|_\alpha(\tau) + \frac{1}{2\beta}(\tau - \|x\|)^2 : \tau \in \mathbb{R}\right\} \\ &= \frac{x}{\|x\|} \cdot \mathrm{prox}_{\beta|\cdot|_\alpha}(\|x\|).\end{aligned}$$

This completes the proof. □

The expression of φ_α is given in the following lemma.

Proposition 2. *Let φ be a compositional norm on $\mathbb{R}^d$, as defined by (6). For any $x \in \mathbb{R}^d$ and a positive parameter α, we have that*

$$\varphi_\alpha(x) = \sum_{j=1}^{J} |\cdot|_\alpha(\|I_{\omega_j}x\|). \tag{7}$$

Furthermore, for any positive parameter β, we have that

$$\mathrm{prox}_{\beta\varphi}(x) = \sum_{j=1}^{J} I_{\omega_j}^\top \mathrm{prox}_{\beta|\cdot|}(\|I_{\omega_j}x\|)\frac{I_{\omega_j}x}{\|I_{\omega_j}x\|}. \tag{8}$$

and

$$\mathrm{prox}_{\beta\varphi_\alpha}(x) = \sum_{j=1}^{J} I_{\omega_j}^\top \mathrm{prox}_{\beta|\cdot|_\alpha}(\|I_{\omega_j}x\|)\frac{I_{\omega_j}x}{\|I_{\omega_j}x\|}. \tag{9}$$

Proof. We omit the proof of equation (8) here since its proof is similar to that of equation (9). Because of the block structure of φ given in (6), and using the definition of Moreau envelope, we have that

$$\mathrm{env}_\alpha\varphi(x) = \sum_{j=1}^{J} \min\left\{\frac{1}{2\alpha}\|v - I_{\omega_i}x\|^2 + \|v\| : v \in \mathbb{R}^{\#\omega_i}\right\} = \sum_{j=1}^{J} \mathrm{env}_\alpha|\cdot|(\|I_{\omega_i}x\|).$$

From the above equation, we have

$$\varphi_\alpha(x) = \sum_{j=1}^{J}(\|I_{\omega_i}x\| - \mathrm{env}_\alpha|\cdot|(\|I_{\omega_i}x\|)) = \sum_{j=1}^{J}|\cdot|_\alpha(\|I_{\omega_j}x\|),$$

which is (7).

Next, we compute the proximity operator of φ_α. We have that

$$\begin{aligned}\mathrm{prox}_{\beta\varphi_\alpha}(x) &= \arg\min\left\{\sum_{j=1}^{J}\left(|\cdot|_\alpha(\|I_{\omega_j}w\|) + \frac{1}{2\beta}\|I_{\omega_j}w - I_{\omega_j}x\|^2\right) : w \in \mathbb{R}^d\right\} \\ &= \sum_{j=1}^{J} I_{\omega_j}^\top \arg\min\left\{\left(|\cdot|_\alpha(\|u\|) + \frac{1}{2\beta}\|u - I_{\omega_j}x\|^2\right) : u \in \mathbb{R}^{\#\omega_j}\right\},\end{aligned}$$

which is (9) by Lemma 2. □

3 Problem Formulation and Algorithms

We are now able to state the model under consideration in this paper and to provide some insight into its benefits. We are interested in solving

$$\operatorname{argmin}\left\{ \frac{1}{2\lambda}\|x-z\|^2 + \varphi_\alpha(Bx) \ : x \in C \right\}, \tag{$\mathcal{P}$}$$

where $C \subset \mathbb{R}^d$ is closed and convex, $z \in \mathbb{R}^d$, $B \in \mathbb{R}^{n\times d}$, and φ_α is an SPF as defined by (3).

While in some instances φ_α may be convex (e.g. if φ is sufficiently strongly convex), without further information, we regard φ_α as nonconvex. However, depending on the parameters α and λ as well as the choice of matrix B, we see that ($\mathcal{P}$) may be convex.

Lemma 3. *For any convex function φ on $\mathbb{R}^n$, an $n \times d$ matrix B, positive parameters λ and α, and any fixed $z \in \mathbb{R}^n$, define*

$$W(x) := \frac{1}{2\lambda}\|x-z\|^2 + \varphi_\alpha(Bx).$$

If $\lambda < \frac{\alpha}{\|B\|^2}$, then W is strictly convex on $\mathbb{R}^n$. If $\lambda = \frac{\alpha}{\|B\|^2}$, then W is convex.

Proof. This is a direct consequence of item (iv) in Lemma 1. □

Clearly, the above lemma tells us that for $\lambda \le \frac{\alpha}{\|B\|^2}$, any critical point of ($\mathcal{P}$) is a global minimum, and for $\lambda < \frac{\alpha}{\|B\|^2}$, the minimizer is unique.

The structure of φ_α lends flexibility to this model; ($\mathcal{P}$) can be made to fit a variety of generic models by grouping terms in different ways. For example, the objective function of the model can be viewed as a difference of convex functions $\frac{1}{2\lambda}\|x-z\|^2 + \varphi(Bx)$ and $\operatorname{env}_\alpha \varphi(Bx)$, and therefore suitable for the rich framework of DC algorithms [1]. Based on this structure, we can decompose model ($\mathcal{P}$) in three ways which correspond to different classes of algorithms: convex, difference of convex, and nonconvex. More precisely, we study three algorithms which highlight each of these cases: primal-dual splitting (PD), difference of convex (DC), and primal-dual hybrid gradient (PDHG) method.

3.1 Primal Dual Splitting

By identifying

$$F(x) = \frac{1}{2\lambda}\|x-z\|_2^2 - \operatorname{env}_\alpha \varphi(Bx), \quad G(x) = \iota_C(x), \quad H(x) = \varphi(x), \tag{10}$$

model ($\mathcal{P}$) can be viewed as a special case of the following generic model

$$\arg\min\left\{ F(x) + G(x) + H(Bx) : x \in \mathbb{R}^d \right\}. \tag{11}$$

Under the assumptions that (i) F is convex and differentiable with L-Lipschitz gradient and (ii) G and H are proper, convex, lower semicontinuous, and prox-friendly, several algorithms have been developed for the optimization problem (11), see, for example, [6, 7, 24]. We adopt the primal-dual splitting algorithm in [7] as follows: given initial points $(x^{(0)}, y^{(0)})$ and positive parameters σ, τ, ρ, iterate

$$\tilde{x}^{(k+1)} := \operatorname{prox}_{\tau G}(x^{(k)} - \tau \nabla F(x^{(k)}) - \tau B^\top y^{(k)}) \tag{12}$$

$$\tilde{y}^{(k+1)} := \operatorname{prox}_{\sigma H^*}(y^{(k)} + \sigma B(2\tilde{x}^{(k+1)} - x^{(k)})) \tag{13}$$

$$\begin{bmatrix} x^{(k+1)} \\ y^{(k+1)} \end{bmatrix} := \rho \begin{bmatrix} \tilde{x}^{(k+1)} \\ \tilde{y}^{(k+1)} \end{bmatrix} + (1-\rho) \begin{bmatrix} x^{(k)} \\ y^{(k)} \end{bmatrix}. \tag{14}$$

In the above scheme, H^* is the Fenchel conjugate of H. The convergence analysis of the above iterative scheme given in [7] is stated in the following result.

Proposition 3 (Condat [7]). *Let τ, σ, and ρ be the parameters in (12)–(14). Suppose that the functions F, G, and H in (11) are convex, the gradient of F is L-Lipschitz with $L > 0$, and the following hold: (i) $\frac{1}{\tau} - \sigma\|B\|^2 > \frac{L}{2}$; and (ii) $\rho \in (0, 1]$. Then the sequence $(x^{(k)})_{k\in\mathbb{N}}$ converges to a solution of the problem (11).*

We now verify the assumptions of Proposition 3 through the identifications (10).

Proposition 4. *Let F be defined as in (10). Then the following statements hold:*

1. *F is differentiable. Moreover, its gradient is L-Lipschitz continuous with*

$$L = \begin{cases} \frac{1}{\lambda}, & \text{if } \|B\|^2 \leq \frac{2\alpha}{\lambda}; \\ \sqrt{\frac{1}{\lambda^2} + \frac{\|B\|^2}{\alpha^2}(\|B\|^2 - \frac{2\alpha}{\lambda})}, & \text{otherwise.} \end{cases}$$

2. *F is strictly convex on $\mathbb{R}^n$ if $\lambda < \frac{\alpha}{\|B\|^2}$; convex if $\lambda = \frac{\alpha}{\|B\|^2}$.*

Proof. (i): We know that Moreau envelope of a convex function is differentiable. Hence, F is differentiable and is simply the difference of two differentiable functions. Actually, we have that

$$\nabla F = \frac{1}{\lambda}(\cdot - z) - B^\top \mathrm{prox}_{\alpha^{-1}\varphi^*}(\alpha^{-1}B\cdot).$$

For any x and y in $\mathbb{R}^n$, let us denote $p = \mathrm{prox}_{\alpha^{-1}\varphi^*}(\alpha^{-1}Bx)$ and $q = \mathrm{prox}_{\alpha^{-1}\varphi^*}(\alpha^{-1}By)$. Then, one has

$$\begin{aligned} \|\nabla F(x) - \nabla F(y)\|^2 &= \frac{1}{\lambda^2}\|x-y\|^2 - \frac{2\alpha}{\lambda}\langle\alpha^{-1}B(x-y), p-q\rangle + \|B^\top(p-q)\|^2 \\ &\leq \frac{1}{\lambda^2}\|x-y\|^2 - \frac{2\alpha}{\lambda}\|p-q\|^2 + \|B^\top(p-q)\|^2 \\ &= \frac{1}{\lambda^2}\|x-y\|^2 + (p-q)^\top(BB^\top - \frac{2\alpha}{\lambda}\,\mathrm{Id})(p-q). \end{aligned}$$

Obviously, if $\|B\|^2 \leq \frac{2\alpha}{\lambda}$, then $BB^\top - \frac{2\alpha}{\lambda}\,\mathrm{Id}$ is semi-negative. Thus, $\|\nabla F(x) - \nabla F(y)\| \leq \frac{1}{\lambda^2}\|x-y\|$. If $\|B\|^2 > \frac{2\alpha}{\lambda}$, then, by using the inequality $\|p-q\| \leq \alpha^{-1}\|B\|\|x-y\|$, we have

$$(p-q)^\top(BB^\top - \frac{2\alpha}{\lambda}\,\mathrm{Id})(p-q) \leq \frac{\|B\|^2}{\alpha^2}(\|B\|^2 - \frac{2\alpha}{\lambda})\|x-y\|^2.$$

The result follows immediately.

(ii): By the definition of the Moreau envelope, we have

$$\begin{aligned} F(x) &= \frac{1}{2\lambda}\|x-z\|^2 - \min\left\{\frac{1}{2\alpha}\|u-Bx\|^2 + \varphi(u) : u \in \mathbb{R}^n\right\} \\ &= \frac{1}{2\lambda}\|x-z\|^2 - \frac{1}{2\alpha}\|Bx\|^2 + \frac{1}{2\alpha}\max\left\{2\langle B^\top u, x\rangle - \|u\|^2 - 2\alpha\varphi(u) : u \in \mathbb{R}^n\right\}. \end{aligned}$$

Since

$$\frac{1}{2\lambda}\|x-z\|^2 - \frac{1}{2\alpha}\|Bx\|^2 = x^\top\left(\frac{1}{2\lambda}\,\mathrm{Id} - \frac{1}{2\alpha}B^\top B\right)x + \frac{1}{2\lambda}(\|z\|^2 - 2z^\top x),$$

which is strictly convex if $\lambda < \frac{\alpha}{\|B\|^2}$, and $\max\left\{2\langle B^\top u, x\rangle - \|u\|^2 - 2\alpha\varphi(u) : u \in \mathbb{R}^n\right\}$ is convex as a function of x, we see that F is strictly convex. Finally, if $\lambda = \frac{\alpha}{\|B\|^2}$, it is clear that F is convex. □

Algorithm 1 is the direct application of (12)-(14) to $(\mathcal{P})$ through the identifications (10). We note that for the given F,

$$\nabla F(x) = \frac{1}{\lambda}(x - z) - B^\top \nabla \operatorname{env}_\alpha \varphi(Bx).$$

Applying the Moreau identity, we write $\nabla \operatorname{env}_\alpha \varphi(Bx) = \operatorname{prox}_{\alpha^{-1}\varphi^*}(\alpha^{-1}Bx)$.

Algorithm 1: Primal-Dual Splitting Algorithm for $(\mathcal{P})$

Input: Choose the positive parameters τ, σ, the sequence of positive relaxation parameters $(\rho_n)_{n\in\mathbb{N}}$, and the initial estimates $x^{(0)} \in \mathbb{R}^d$, $y^{(0)} \in \mathbb{R}^n$.

for $n = 0, 1, \ldots$ **do**

$$\begin{aligned}
\tilde{x}^{(k+1)} &\leftarrow \operatorname{proj}_C\left(x^{(k)} - \tau\left(\frac{1}{\lambda}(x^{(k)} - z)\right) + \tau B^\top\left(\operatorname{prox}_{\alpha^{-1}\varphi^*}(\alpha^{-1}Bx^{(k)}) - y^{(k)}\right)\right)\\
\tilde{y}^{(k+1)} &\leftarrow \operatorname{prox}_{\sigma\varphi^*}\left(y^{(k)} + \sigma B(2\tilde{x}^{(k+1)} - x^{(k)})\right)\\
\begin{bmatrix} x^{(k+1)} \\ y^{(k+1)} \end{bmatrix} &\leftarrow \rho \begin{bmatrix} \tilde{x}^{(k+1)} \\ \tilde{y}^{(k+1)} \end{bmatrix} + (1-\rho)\begin{bmatrix} x^{(k)} \\ y^{(k)} \end{bmatrix}
\end{aligned}$$

Theorem 1. *Let λ, α, and z be as in problem $(\mathcal{P})$, and let τ, σ, and ρ be the parameters in Algorithm 1. Suppose that $\lambda < \frac{\alpha}{\|B\|^2}$ and the following hold:*

(i) $\frac{1}{\tau} - \sigma\|B\|^2 > \frac{1}{2\lambda}$;

(ii) $\rho \in (0, 1]$.

Then the sequence $(x^{(k)})_{k\in\mathbb{N}}$ produced by Algorithm 1 converges to a solution of the problem $(\mathcal{P})$.

Proof. By Lemma 3 and Proposition 4, if $\lambda < \frac{\alpha}{\|B\|^2}$, then the objective function of problem $(\mathcal{P})$ is strictly convex, and the gradient of F given in (10) is $\frac{1}{\lambda}$-Lipschitz continuous. Hence, the convergence of the sequence $(x^{(k)})_{k\in\mathbb{N}}$ is the consequence of Proposition 3. □

3.2 Difference of Convex Algorithm

By $\varphi_\alpha = \varphi - \operatorname{env}_\alpha\varphi$ from (3), set

$$Q(x) = \frac{1}{2\lambda}\|x - z\|_2^2 + \iota_C(x) + \varphi(Bx), \quad P(x) = \operatorname{env}_\alpha(Bx), \tag{15}$$

then model $(\mathcal{P})$ can be viewed as a special case of the following generic model

$$\min\{Q(x) - P(x) : x \in \mathbb{R}^d\}, \tag{16}$$

where both P and Q are convex functions. Due the objective function is the difference of convex (DC) functions, model (16) is referred to as DC program.

DCA (DC algorithm) is based on local optimality conditions and duality in DC programming [13]. The main idea of DCA is as follow: at each iteration k, DCA approximates the second DC

component $P(x)$ by the affine approximation $P_k(x) = P(x^{(k)}) + \langle y^{(k)}, x - x^{(k)} \rangle$, with $y^{(k)} \in \partial P(x^{(k)})$, and minimizes the resulting convex function. DCA for (16) is as follows:

$$y^{(k)} \in \partial P(x^{(k)}) \tag{17}$$
$$x^{(k+1)} \in \arg\min\{Q(x) - P_k(x) : x \in \mathbb{R}^d\} \tag{18}$$

As the optimal solution set of (18) is $\partial Q^*(y^{(k)})$, the DCA scheme can be expressed in another form:

$$\text{For } k = 0, 1, \ldots, \text{ set} \quad y^{(k)} \in \partial P(x^{(k)}); \quad x^{(k+1)} \in \partial Q^*(y^{(k)}).$$

We state the local convergence properties of DCA in the following theorem (see [22]).

Theorem 2 ([22],Theorem 3.7)**.** *Suppose that the sequence $\{x^{(k)}\}_{k\in\mathbb{N}}$ is defined by the iterative scheme* (17)-(18) *for problem* (16)*. Then we have*

(i) The objective value sequence $\{Q(x^{(k)}) - P(x^{(k)})\}_{k\in\mathbb{N}}$ is monotonically decreasing.

(ii) If the optimal value of problem (16) *is finite and the sequence $\{x^{(k)}\}_{k\in\mathbb{N}}$ is bounded, then every limit point $x^\diamond$ of $\{x^{(k)}\}_{k\in\mathbb{N}}$ is a critical point of the problem.*

With these properties on DC programming in hands, we turn back to the problem (16) with P and Q given in (15).

Algorithm 2: DCA scheme for (16) with P and Q given in (15)

Input: Choose initial estimate $x^{(0)} \in \mathrm{dom}\partial P$.

for $k = 0, 1, \ldots$ **do**

$$y^{(k)} \leftarrow B^\top \nabla \mathrm{env}_\alpha\, \varphi(Bx^{(k)}) \tag{19}$$
$$x^{(k+1)} \leftarrow \arg\min \left\{ \frac{1}{2\lambda}\|x - z\|^2 + \iota_C(x) + \varphi(Bx) - \langle y^{(k)}, x\rangle : x \in \mathbb{R}^d \right\} \tag{20}$$

Theorem 3. *Suppose that the sequences $\{x^{(k)}\}_{k\in\mathbb{N}}$ and $\{y^{(k)}\}_{k\in\mathbb{N}}$ are generated by Algorithm 2 for problem* (16) *with P and Q given in* (15)*. Then every limit point $x^\diamond$ of $\{x^{(k)}\}_{k\in\mathbb{N}}$ is a critical point of the problem. Moreover, $\lim_{k\to\infty} \|x^{(k+1)} - x^{(k)}\| = 0$.*

Proof. Recall that $Q(x) - P(x) = \frac{1}{2\lambda}\|x - z\|^2 + \iota_C(x) + \varphi_\alpha(Bx)$ which is nonnegative and continuous on its domain. Hence, the optimal value of problem (16) is finite. From item (i) of Theorem 2, we have that

$$\frac{1}{2\lambda}\|x^{(k)} - z\|^2 \le Q(x^{(k)}) - P(x^{(k)}) \le Q(x^{(0)}) - P(x^{(0)}) < \infty.$$

This leads to the boundedness of the sequence $\{x^{(k)}\}_{k\in\mathbb{N}}$. From (19), $\mathrm{prox}_{\alpha\phi}(0) = 0$, and the fact that $\mathrm{Id} - \mathrm{prox}_{\alpha\varphi}$ is nonexpansive operator, we have $\|y^{(k)}\| = \frac{1}{\alpha}\|B^\top(\mathrm{Id} - \mathrm{prox}_{\alpha\varphi})(Bx^{(k)})\| \le \frac{\|B\|^2}{\alpha}\|x^{(k)}\|$, hence the $\{y^{(k)}\}_{k\in\mathbb{N}}$ is bounded. By item (ii) of Theorem 2, we know that every limit point $x^\diamond$ of $\{x^{(k)}\}_{k\in\mathbb{N}}$ is a critical point of the problem.

By $y^{(k)} \in \partial P(x^{(k)})$, we have $P(x^{(k+1)}) \ge P(x^{(k)}) + \langle y^{(k)}, x^{(k+1)} - x^{(k)}\rangle$. Since Q is strongly convex and $x^{(k+1)}$ minimizes $Q(x) - \langle y^{(k)}, x\rangle$, we get

$$Q(x^{(k+1)}) - \langle y^{(k)}, x^{(k+1)}\rangle \le Q(x^{(k)}) - \langle y^{(k)}, x^{(k)}\rangle - \frac{1}{2\lambda}\|x^{(k+1)} - x^{(k)}\|^2.$$

Therefore, it follows that

$$\begin{aligned} & Q(x^{(k+1)}) - P(x^{(k+1)}) \\ \leq\ & Q(x^{(k+1)}) - \Big(P(x^{(k)}) + \langle y^{(k)}, x^{(k+1)} - x^{(k)}\rangle\Big) \\ =\ & Q(x^{(k+1)}) - \langle y^{(k)}, x^{(k+1)}\rangle - \Big(P(x^{(k)}) - \langle y^{(k)}, x^{(k)}\rangle\Big) \\ \leq\ & Q(x^{(k)}) - \langle y^{(k)}, x^{(k)}\rangle - \frac{1}{2\lambda}\|x^{(k+1)} - x^{(k)}\|^2 - \Big(P(x^{(k)}) - \langle y^{(k)}, x^{(k)}\rangle\Big) \\ =\ & Q(x^{(k)}) - P(x^{(k)}) - \frac{1}{2\lambda}\|x^{(k+1)} - x^{(k)}\|^2. \end{aligned}$$

From this, we get

$$\frac{1}{2\lambda}\|x^{(k+1)} - x^{(k)}\|^2 \leq (Q(x^{(k)}) - P(x^{(k)})) - (Q(x^{(k+1)}) - P(x^{(k+1)})).$$

Summing the above inequality for all k from 0 to infinity yields

$$\frac{1}{2\lambda}\sum_{k=0}^{\infty}\|x^{(k+1)} - x^{(k)}\|^2 \leq Q(x^{(0)}) - P(x^{(0)}),$$

which implies $\lim_{k\to\infty}\|x^{(k+1)} - x^{(k)}\| = 0$. □

3.3 Primal-Dual Hybrid Gradient Methods

Set

$$Q(x) = \frac{1}{2\lambda}\|x - z\|_2^2 + \iota_C(x), \quad P(x) = \varphi_\alpha(x), \tag{21}$$

then model $(\mathcal{P})$ can be viewed as a special case of the following generic model

$$\min\{Q(x) + P(Bx) : x \in \mathbb{R}^d\}, \tag{22}$$

where P is semiconvex and Q is convex. In this setting, a primal-dual hybrid gradient (PDHG) method was proposed for model (22) in [15] as follows: Given a pair $(x^{(0)}, \theta^{(0)}) \in \mathbb{R}^d \times \mathbb{R}^n$ and for $\bar{x}^{(0)} = x^{(0)}$, $\sigma > 0$, $\tau > 0$, and $\rho \in [0, 1]$, iterate for all $k \geq 0$

$$u^{(k+1)} = \operatorname{argmin}\left\{\frac{\sigma}{2}\|u - B\bar{x}^{(k)}\|^2 - \langle u, \theta^{(k)}\rangle + P(u) : u \in \mathbb{R}^d\right\} \tag{23}$$

$$\theta^{(k+1)} = \theta^{(k)} + \sigma(B\bar{x}^{(k)} - u^{(k+1)}) \tag{24}$$

$$x^{(k+1)} = \operatorname{argmin}\left\{\frac{1}{2\tau}\|x - x^{(k)}\|^2 + \langle Bx, \theta^{(k+1)}\rangle + Q(x) : x \in \mathbb{R}^n\right\} \tag{25}$$

$$\bar{x}^{(k+1)} = x^{(k+1)} + \rho(x^{(k+1)} - x^{(k)}) \tag{26}$$

We note that both $u^{(k+1)}$ in (23) and $x^{(k+1)}$ in (25) can be explicitly expressed in terms of proximity operator. Below we give our PDHG-based algorithm for solving the optimization problem (22).

Theorem 4. *For optimization model* (22) *with* P *and* Q *given in* (21)*, if* $\alpha \geq \lambda\|B\|^2$*, then Algorithm 3 converges to the unique solution* $x^\star$ *of model* (22) *for* $\sigma\alpha = 2$*,* $\tau\sigma\|B\|^2 \leq 1$*, and any* $\rho \in [0, 1]$*, with rate* $\|x^{(k)} - x^\star\|^2 \leq \widetilde{C}/k$ *for some constant* $\widetilde{C}$*.*

Proof. As we know, P is $\frac{1}{\alpha}$-semiconvex and Q is $\frac{1}{\lambda}$-strongly convex. By Theorem 2.8 in [15], the conclusion of this theorem holds for the given parameters σ, τ. □

Algorithm 3: PDHG scheme for problem (22)

Input: Choose $\lambda > 0$, $\alpha \geq \lambda\|B\|^2$, $\sigma = 2\alpha^{-1}$, $\tau \leq \sigma^{-1}\|B\|^{-2}$, and $\rho \in [0,1]$. Initialize $x^{(0)} = z$, $\theta^{(0)} = 0$, and $\bar{x}^{(0)} = x^{(0)}$.

for $k = 0, 1, \ldots$ **do**

1) $u^{(k+1)} \leftarrow \text{prox}_{\sigma^{-1}\varphi_\alpha}\left(B\bar{x}^{(k)} + \frac{1}{\sigma}\theta^{(k)}\right)$

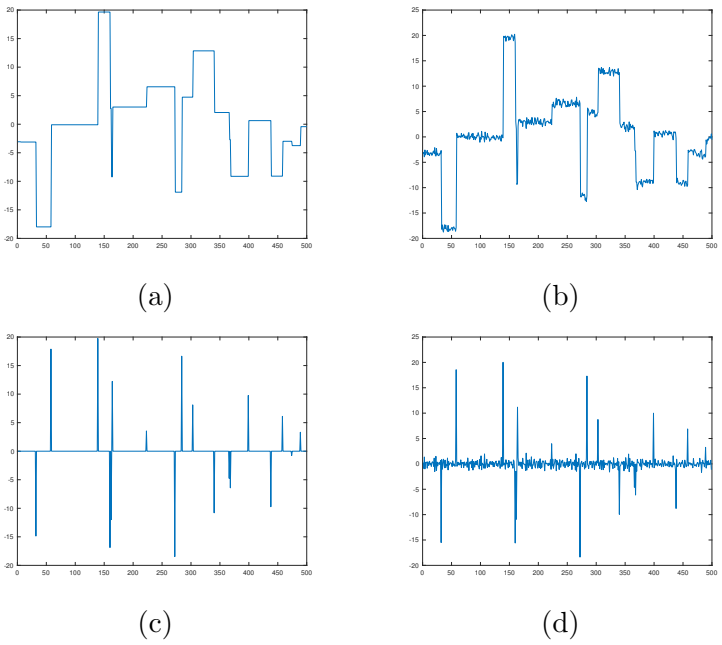
2) $\theta^{(k+1)} \leftarrow \theta^{(k)} + \sigma(B\bar{x}^{(k)} - u^{(k+1)})$

2) $x^{(k+1)} \leftarrow \text{proj}_C\left(\frac{\lambda}{\tau+\lambda}x^{(k)} + \frac{\tau}{\tau+\lambda}z - \frac{\tau\lambda}{\tau+\lambda}B^\top\theta^{(k+1)}\right)$

3) $\bar{x}^{(k+1)} \leftarrow x^{(k+1)} + \rho(x^{(k+1)} - x^{(k)})$

3.4 Discussion

As noted above, one of the main motivations for using nonconvex penalties is to avoid biased solutions. We now provide some discussion to show how this is accomplished in practice in each of the above algorithms. To illustrate these ideas, we look at the example of piecewise constant signals in $\mathbb{R}^d$. To be precise, we set $\varphi = \|\cdot\|_1$, $C = \mathbb{R}^d$, and let B be the one dimensional difference matrix. The vector $z \in \mathbb{R}^d$ is the noisy observation from which we hope to recover the true signal. Piecewise constant signals are sparse under the transformation B; in other words, all of the information about these signals is contained in the amplitude changes. When noise is added, the signal becomes nonsparse, though we assume that the noise is small compared to the signal. An example of such a signal and the noisy observation are given in Figure 3.



(a) (b)

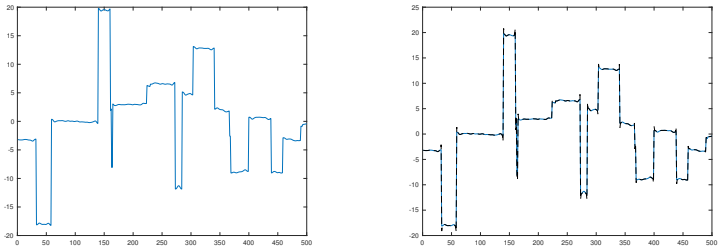
(c) (d)

Figure 3: (a) A piecewise constant signal x, (b) the signal with additive Gaussian noise z, (c) the sparse representation Bx, and (d) the nonsparse Bz.

In Algorithm 1, the primal update is

$$\tilde{x}^{(k+1)} = \operatorname{proj}_C \left(x^{(k)} - \frac{\tau}{\lambda}(x^{(k)} - z) + \tau B^\top \left(\nabla \operatorname{env}_\alpha \varphi(Bx^{(k)}) - y^{(k)} \right) \right),$$

where $\nabla(\operatorname{env}_\alpha \varphi \circ B)(x^{(k)}) = B^\top \operatorname{prox}_{\alpha^{-1}\varphi^*}(\alpha^{-1} B x^{(k)})$, as written in Section 3. When $\varphi = \|\cdot\|_1$, this term is projection of the differences of the current iterate onto the ℓ_∞ unit ball $\{x \in \mathbb{R}^d : \|x\|_\infty \leq 1\}$. This moves $x^{(k)}$ away from the set $\operatorname{argmin} \operatorname{env}_\alpha \varphi \circ B = \operatorname{argmin} \varphi \circ B$, which keeps relevant data from being pulled to zero. As shown in Figure 4, the addition of this term boosts the features of the current iterate in proportion to their magnitude, balancing the shrinkage enforced by the dual update.



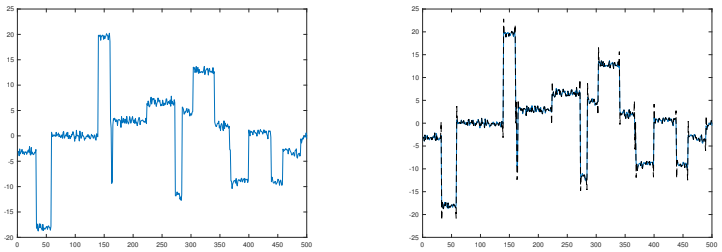
(a) (b)

Figure 4: Algorithm 1. (a) One iterate $x^{(k)}$ and (b) $x^{(k)} + B^\top \nabla \operatorname{env}_\alpha \varphi(Bx^{(k)})$ (dashed black) over $x^{(k)}$ (solid blue). The scaling factor τ is omitted for visibility.

Algorithm 2 requires solving a convex optimization problem in each iteration. Note that

$$\begin{aligned} &\operatorname{argmin}\{\varphi(Bx) + \frac{1}{2\lambda}\|x - z\|^2 - \langle \nabla(\operatorname{env}_\alpha \varphi \circ B)(x^{(k)}), x\rangle : x \in \mathbb{R}^d\} \\ =&\operatorname{argmin}\{\varphi(Bx) + \frac{1}{2\lambda}\|x - (z + \lambda \nabla(\operatorname{env}_\alpha \varphi \circ B)(x^{(k)})\|^2 : x \in \mathbb{R}^d\}. \end{aligned}$$

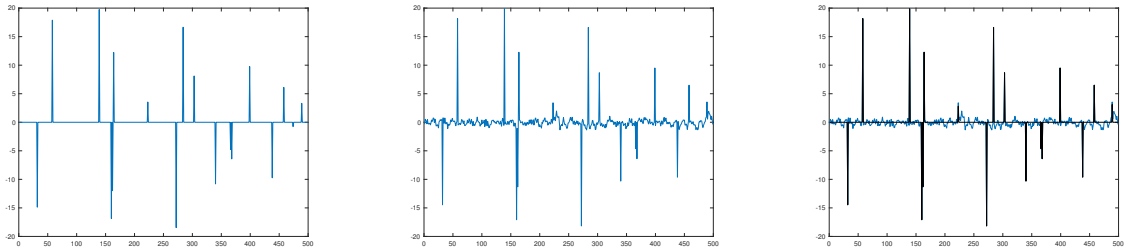
That is, this algorithm modifies the noisy signal at each iteration using the most recent update. This is very similar to Bregman iterations for solving the TV denoising problem with the subgradient of $\|B \cdot\|_1$ replaced by the gradient of the envelope (see [26]). As before, this boosts the relevant features of the signal, as illustrated in Figure 5.



(a) (b)

Figure 5: Algorithm 2. The noisy signal (a) and the L1 initialization points $z + \lambda B^\top \operatorname{env}_\alpha \varphi(Bx^{(k)})$ for (b) $k = 4$.

Algorithm 3 uses the proximity operator of φ_α directly, splitting the problem into a sparsity update and a fidelity update. As sparsity promoting functions, the proximity operators of both φ and φ_α send small entries to zero. However, the nonconvexity of φ_α gives us a greater tolerance for large entries. For instance, when $\varphi = \|\cdot\|_1$, prox_φ shrinks all entries towards zero, while $\text{prox}_{\varphi_\alpha}$ is the identity on entries beyond a certain threshold. These large entries correspond to true signal information. This is illustrated in Figure 6.



(a) (b) (c)

Figure 6: Algorithm 3. (a) The true sparse representation of the signal, (b) $Bx^{(k)} + \frac{1}{\sigma}\theta^{(k)}$ for $k = 10$, and (c) the update $\text{prox}_{\sigma^{-1}\varphi_\alpha}(Bx^{(k)} + \frac{1}{\sigma}\theta^{(k)})$.

In summary, each algorithm reduces bias differently: Algorithm 1 emphasizes the signal features of each primal iterate, Algorithm 2 consists of Bregman-like iterations which incorporate the boosting term into the noisy signal, and Algorithm 3 uses the form of $\text{prox}_{\varphi_\alpha}$ directly. However, in each case we see that the inclusion of the envelope works to preserve signal features, either directly (as in the first two algorithms) or implicitly (as in the last algorithm).

4 Numerical Experiments

In this section, we specify the matrix B, the function φ, and the set C in model $(\mathcal{P})$ so that the resulting model is suitable for image denoising.

We choose the matrix B of size $2N^2 \times N^2$ through an $N \times N$ matrix D as follows:

$$B := \begin{bmatrix} \text{Id}_N \otimes D \\ D \otimes \text{Id}_N \end{bmatrix} \quad \text{with} \quad D := \begin{bmatrix} 0 & & & \\ -1 & 1 & & \\ & \ddots & \ddots & \\ & & -1 & 1 \end{bmatrix},$$

where Id_N is the $N \times N$ identity matrix and the notation $P \otimes Q$ denotes the Kronecker product of matrices P and Q. We know that $\|B\|^2 = 8\sin^2\frac{(N-1)\pi}{2N} < 1$ (see, e.g., [16]).

Let u be a vector in $\mathbb{R}^{2N^2}$. We choose $\varphi : \mathbb{R}^{2N^2} \to \mathbb{R}$ as a compositional norm given in (6) with $\omega_j = \{j, N^2 + j\}$, that is,

$$\varphi(u) := \sum_{j=1}^{N^2} \left\| \begin{bmatrix} u_j \\ u_{N^2+j} \end{bmatrix} \right\|, \quad u \in \mathbb{R}^{2N^2}.$$

With B and φ given in the above, $\varphi(Bx)$ is called the total variation of the image x in $\mathbb{R}^{N^2}$, and the pair of $\varphi(Bx)$ with indices in ω_j is essentially the discrete gradient of the image at the j-th pixel. Here, x is the vectorization of an image formed by stacking the columns of this image

into a single column vector. For easier reading without causing ambiguity, an image is treated as a two-dimensional array and a one-dimensional vector interchangeably. Finally, since all pixel values of a gray-scale image are in $[0, 255]$, we choose $C := [0, 255]^{N^2}$ for images in $\mathbb{R}^{N^2}$.

Prior to applying Algorithm 1 (PD), Algorithm 2 (DCA), and Algorithm 3 (PDHG) for model $(\mathcal{P})$, we also need to know the proximity operators of the functions φ and ι_C. The proximity operator of φ_α is given in (9). From the Moreau identity and (8), we know that for any $u \in \mathbb{R}^{2N^2}$

$$\operatorname{prox}_{\sigma\varphi^*}(u) = u - \sigma \operatorname{prox}_{\sigma^{-1}\varphi}(\sigma^{-1}u) = \sum_{j=1}^{N^2} I_{w_j}^\top \cdot \operatorname{proj}_{[0,1]}(\|I_{\omega_j} u\|) \cdot \frac{I_{\omega_j} u}{\|I_{\omega_j} u\|},$$

which does not depend on σ. This formula says that for each pair of u with indices ω_j, its projection onto the unit ball centered at the origin is the pair of $\operatorname{prox}_{\sigma\varphi^*}(u)$ with the same indices. For the indicator function ι_C, $\operatorname{prox}_{\iota_C} = \operatorname{proj}_C$ which will send the values in a vector larger than 255 or lower than 0 to 255 and 0, respectively.

For comparison, we include the ROF-TV model which is a special case of model (11) with $F = \frac{1}{2\lambda}\|\cdot - z\|^2$, $G = \iota_C$, and $H = \varphi$. This model is solved by the iterative scheme given in (12)-(14). The corresponding algorithm is referred to as ROF-TV algorithm.

In the rest of this section, we present all parameters used in Algorithms ROF-TV, PD, DCA, and PDHG, and compare their numerical performance for image denoising.

4.1 Parameters and Stopping Criterion

We first talk about the parameters related to the underlying models, then discuss the parameters associated with each algorithm, and finally describe the stopping criterion for all algorithms.

Model $(\mathcal{P})$ involves two parameters λ and α. It is well known that the regularization parameter λ varies according to the noise level of the noisy image to be denoised. From Proposition 4, we know that model $(\mathcal{P})$ is strictly convex, hence, has a unique solution when $\alpha > \lambda\|B\|^2$. Therefore, in our experiments, we always choose $\alpha = 1.5\lambda\|B\|^2$ for each given λ.

Methods of ROF-TV and PD exploit the iterative scheme (12)-(14) for which the proper values of the parameters σ, τ, and ρ are to be assigned. We use the model for ROF-TV algorithm as an example to show how to set these parameters. We reformulate the associated model (11) with $F = \frac{1}{2\lambda}\|\cdot - z\|^2$, $G = \iota_C$, and $H = \varphi$ without changing its minimizer, to the one with $F = \frac{1}{2}\|\cdot - z\|^2$, $G = \iota_C$, and $H = \lambda\varphi$. In our simulations, we choose $\sigma = 0.1$, $\tau = 0.99/(0.5 + \sigma\|B\|^2)$, and $\rho = 1$. With these chosen parameters, the sequences of $\{x^{(k)}\}_{k\in\mathbb{N}}$, generated by ROF-TV, PD, and DCA, converge to the solutions of the corresponding optimization models, respectively.

For PDHG, we choose $\sigma = 2/\alpha$, $\tau = 0.99/(\sigma\|B\|^2)$, and $\rho = 1$. Then, the convergence of the sequence of $\{x^{(k)}\}_{k\in\mathbb{N}}$, generated by PDHG, is the consequence of Theorem 4.

Iterations in the algorithms of ROF-TV, PD, DCA, and PDHG are terminated whenever one of the following two conditions occurs: the maximum number of iterations has been exceeded or

$$\|x^{(k+1)} - x^{(k)}\|/\|x^{(k)}\| \leq \texttt{tol},$$

where `tol` denotes a prescribed tolerance value. In our experiments, we set $\texttt{tol} = 10^{-4}$. For Algorithms ROF-TV, PD, and PDHG, the maximum number of iterations is set to be 300. For Algorithm DCA, there are basically two levels of looping: outer loop and inner loop. The outer loop refers to the procedure of generating $y^{(k)}$ and $x^{(k+1)}$ via (19) and (20), respectively. The inner loop is used to find $x^{(k+1)}$ via an iterative scheme. We set the maximum number of iterations for the outer loop to be 10, and 100 for the inner loop.

4.2 Numerical Results for Denoising

In our experiments, we choose the images of "Cameraman" (Figure 8(a)), "House" (Figure 9(a)), and "Peppers" (Figure 10(a)) with size 256×256, as the original images x. The noisy images (for example, see, Figures 8(b), 9(b), and 10(b)) are modeled as

$$z = x + \epsilon$$

with ϵ being the white Gaussian noise of standard derivation η. The noise at level η being 15, 20, and 25 will be added to the test images to evaluate the performance of the proposed model and the corresponding algorithms. The quality of the denoised image $\widetilde{x}$ obtained from a denoising algorithm is measured by the peak-signal-to-noise ratio (PSNR)

$$\text{PSNR} := 20 \log_{10} \left(\frac{255}{256\|x - \widetilde{x}\|} \right).$$

In Table 1, we reported the average PSNR values of the denoised images of "Cameraman" and the CPU time consumed by all tested algorithms for various values of λ over 20 realizations at the same noise level. Note that algorithms PD, DCA, and PDHG are developed to find a solution to model $(\mathcal{P})$. From this table, we observed that PDHG performs always better than PD and DCA in terms of both the PSNR values of the denoised image and the CPU time used. The same conclusion can be drawn for the image of "House" as shown in Table 2. Numerical results for the image of "Peppers" are listed in Table 3. In this case, DCA produced better denoised images than PD and DCA in terms of the PSNR values, however, using much more CPU times. From the PSNR values in these tables, we can see that the quality of the denoised images via the optimization model penalized by the proposed structured promoting functions (solved by PD, DCA, and PDHG) is better than that with the classical ROF-TV model. For noise at level $\eta = 20$, Figure 7(a) illustrates the PSNR values of the denoised "Cameraman" images via all methods over 20 noise realizations while Figure 7(b) presents the used CPU times. We can see that PDHG consistently produces the highest quality images with the least CPU time used.

Figure 8 shows the denoised images when all algorithms apply to the noisy image of "Cameraman" with noise level of 20. For the same noise level, Figure 9 shows the denoised images of "House" while Figure 10 shows the denoised images of "Peppers". Although all denoised images look similar, visually, we can see that the denoised images by PDHG have less artifacts than the others.

Table 1: Numerical results of ROF-TV, PD, DCA, and PDHG methods for the image of "Cameraman". The pair $(\cdot, \cdot)$ is used to report both the PSNR value (the first number) of a denoised image and the CPU time (the second number).

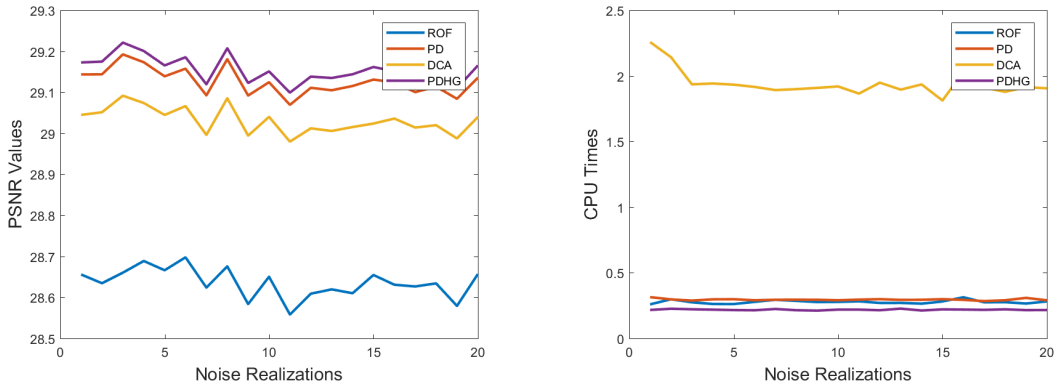
λ	ROF-TV	PD	DCA	PDHG
White Gaussian noise with standard deviation 15				
9	(30.32, 0.19)	(30.20, 0.18)	(30.17, 1.41)	(30.22, 0.18)
10	(30.30, 0.20)	(30.50, 0.21)	(30.44, 1.51)	(30.52, 0.18)
11	(30.18, 0.22)	(30.62, 0.22)	(30.54, 1.52)	(**30.65**, 0.18)
12	(30.01, 0.24)	(30.61, 0.24)	(30.52, 1.59)	(30.63, 0.19)
13	(29.80, 0.26)	(30.50, 0.27)	(30.41, 1.54)	(30.52, 0.20)
White Gaussian noise with standard deviation 20				
14	(28.79, 0.25)	(29.00, 0.26)	(28.92, 1.70)	(29.02, 0.23)
15	(28.73, 0.26)	(29.10, 0.28)	(29.02, 1.80)	(29.13, 0.22)
16	(28.64, 0.28)	(29.13, 0.30)	(29.03, 1.94)	(**29.16**, 0.22)
17	(28.52, 0.30)	(29.09, 0.31)	(28.99, 2.15)	(29.11, 0.22)
18	(28.38, 0.33)	(29.00, 0.34)	(28.91, 1.91)	(29.03, 0.23)
White Gaussian noise with standard deviation 25				
18	(27.67, 0.38)	(27.87, 0.41)	(27.78, 2.55)	(27.89, 0.38)
19	(27.65, 0.38)	(27.97, 0.39)	(27.87, 3.09)	(28.04, 0.27)
20	(27.60, 0.33)	(28.01, 0.36)	(27.90, 2.28)	(**28.04**, 0.26)
21	(27.43, 0.38)	(27.96, 0.39)	(27.85, 2.47)	(27.99, 0.26)
22	(27.33, 0,40)	(27.89, 0.41)	(27.78, 2.34)	(27.91, 0.26)

Table 2: Numerical results of ROF-TV, PD, DCA, and PDHG methods for the image of "House". The pair $(\cdot, \cdot)$ is used to report both the PSNR value (the first number) of a denoised image and the CPU time (the second number).

λ	ROF-TV	PD	DCA	PDHG
White Gaussian noise with standard deviation 15				
9	(32.05, 0.18)	(31.32, 0.18)	(31.30, 1.26)	(31.45, 0.15)
10	(32.30, 0.21)	(31.90, 0.21)	(31.30, 1.40)	(31.93, 0.16)
11	(32.40, 0.20)	(32.26, 0.22)	(32.18, 1.25)	(32.30, 0.16)
12	(32.42, 0.21)	(32.46, 0.24)	(32.35, 1.32)	(32.50, 0.16)
13	(32.37, 0.24)	(32.53, 0.27)	(32.41, 1.41)	(**32.56**, 0.17)
White Gaussian noise with standard deviation 20				
14	(30.94, 0.24)	(30.64, 0.26)	(30.55, 1.62)	(30.67, 0.18)
15	(31.07, 0.25)	(30.95, 0.27)	(30.83, 1.71)	(30.99, 0.17)
16	(31.13, 0.27)	(31.15, 0.29)	(31.02, 1.57)	(31.19, 0.18)
17	(31.14, 0.28)	(31.27, 0.31)	(31.12, 1.67)	(31.31, 0.18)
18	(31.11, 0.29)	(31.31, 0.33)	(31.16, 2.18)	(**31.35**, 0.19)
White Gaussian noise with standard deviation 25				
19	(30.01, 0.36)	(29.91, 0.41)	(29.77, 2.57)	(29.94, 0.26)
20	(30.10, 0.41)	(30.11, 0.48)	(29.95, 2.37)	(30.15, 0.29)
21	(30.14, 0.37)	(30.24, 0.42)	(30.07, 2.28)	(30.29, 0.26)
22	(30.15, 0.36)	(30.33, 0.40)	(30.15, 2.22)	(30.37, 0.24)
23	(30.14, 0.36)	(30.36, 0.43)	(30.18, 2.31)	(**30.41**, 0.24)

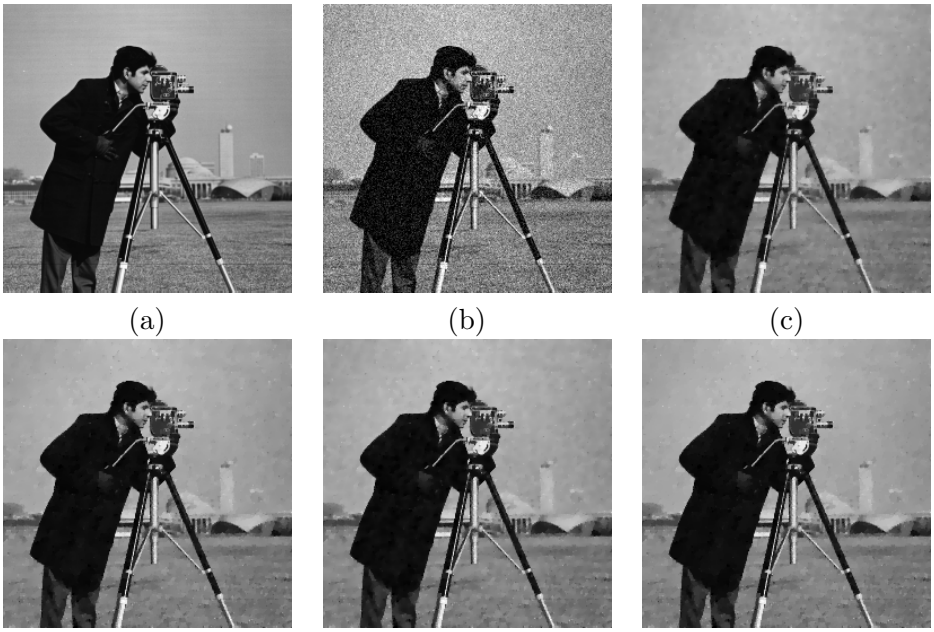
Table 3: Numerical results of ROF-TV, PD, DCA, and PDHG methods for the image of "Peppers". The pair $(\cdot, \cdot)$ is used to report both the PSNR value (the first number) of a denoised image and the CPU time (the second number).

λ	ROF-TV	PD	DCA	PDHG
White Gaussian noise with standard deviation 15				
9	(31.13, 0.21)	(30.48, 0.20)	(30.58, 1.45)	(30.47, 0.19)
10	(31.27, 0.24)	(30.91, 0.23)	(30.01, 1.60)	(30.89, 0.20)
11	(31.31, 0.24)	(31.18, 0.25)	(31.28, 1.45)	(31.15, 0.18)
12	(31.26, 0.27)	(31.29, 0.31)	(31.40, 1.62)	(31.28, 0.21)
13	(31.16, 0.30)	(31.32, 0.31)	(**31.43**, 0.77)	(31.30, 0.19)
White Gaussian noise with standard deviation 20				
14	(29.27, 0.26)	(29.50, 0.27)	(29.58, 1.76)	(29.48, 0.21)
15	(29.81, 0.28)	(29.70, 0.31)	(29.78, 1.94)	(29.69, 0.20)
16	(29.80, 0.36)	(29.82, 0.38)	(29.91, 1.92)	(29.80, 0.24)
17	(29.75, 0.36)	(29.87, 0.40)	(**29.96**, 2.06)	(29.85, 0.24)
18	(29.68, 0.38)	(39.87, 0.47)	(29.96, 2.25)	(29.85, 0.25)
White Gaussian noise with standard deviation 25				
19	(28.66, 0.32)	(28.55, 0.37)	(28.63, 2.23)	(28.54, 0.22)
20	(28.67, 0.36)	(28.67, 0.40)	(28.74, 2.38)	(28.65, 0.23)
21	(28.65, 0.38)	(28.73, 0.42)	(28.80, 2.17)	(28.71, 0.23)
22	(28.61, 0.39)	(28.75, 0.45)	(**28.83**, 2.34)	(28.73, 0.25)
23	(28.55, 0.42)	(28.74, 0.47)	(28.82, 2.42)	(28.72, 0.25)



(a) (b)

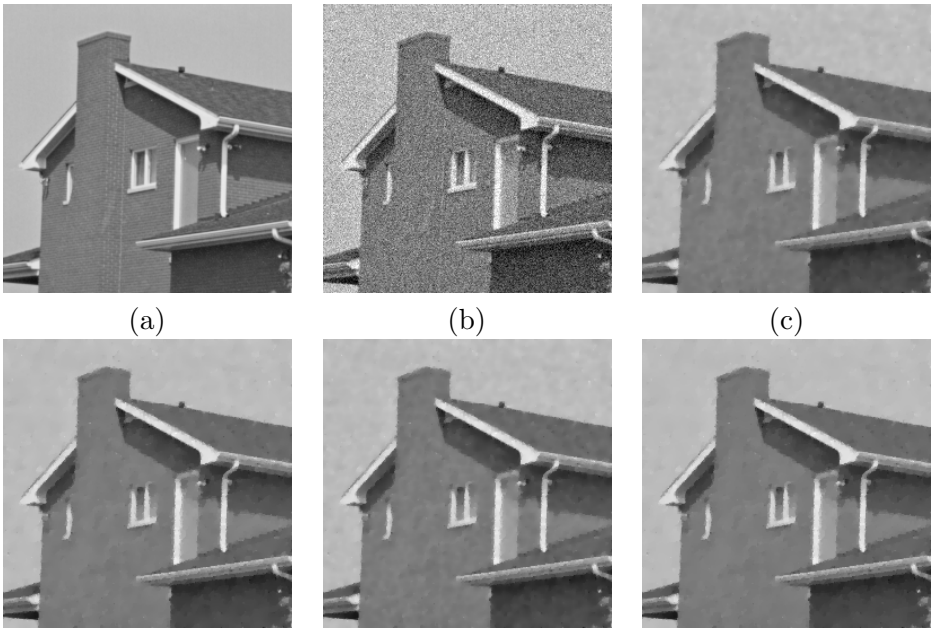
Figure 7: (a) The PSNR value of the denoised image of "Cameraman" for each Gaussian noise realization with standard deviation 20; and (b) the CPU time consumed for various algorithms. The regularization parameter λ is 15 for both ROF-TV model and $(\mathcal{P})$.



(a) (b) (c)

(d) (e) (f)

Figure 8: (a) The image of "Cameraman"; (b) the image of "Cameraman" corrupted by Gaussian noise of standard deviation 20; (c) the denoised image using the ROF-TV denoising model; the denoised images using model ($\mathcal{P}$) by (d) PD; (e) DCA; and (f) PDHG, respectively. The regularization parameter λ for both models is 16.



(a) (b) (c)

(d) (e) (f)

Figure 9: (a) The image of "House"; (b) the image of "House" corrupted by Gaussian noise of standard deviation 20; ((c) the denoised image using the ROF-TV denoising model; the denoised images using model ($\mathcal{P}$) by (d) PD; (e) DCA; and (f) PDHG, respectively. The regularization parameter λ for both models is 18.

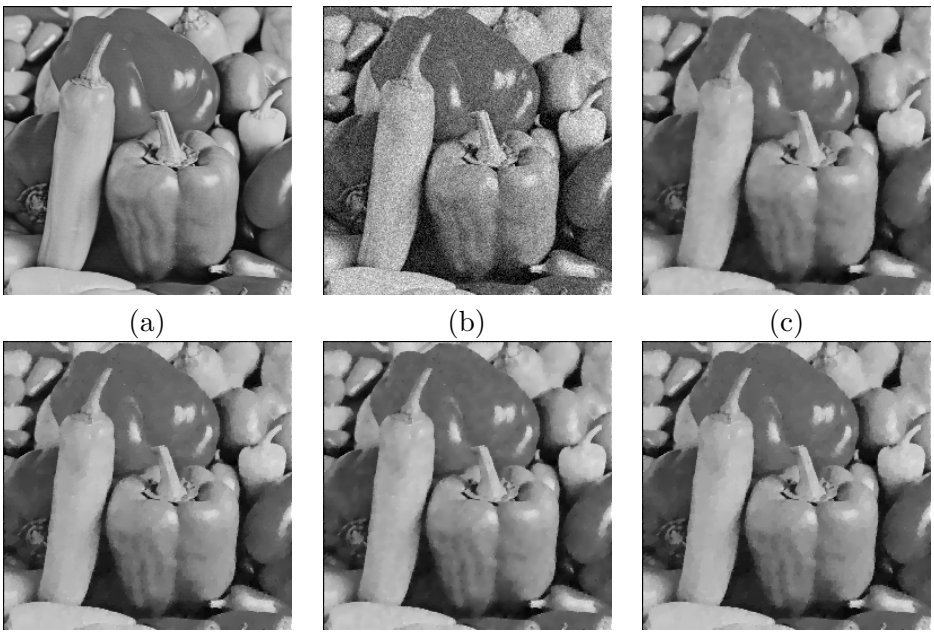


Figure 10: (a) The image of "Peppers"; (b) the image of "Peppers" corrupted by Gaussian noise of standard deviation 20; (c) the denoised image using the ROF-TV denoising model; the denoised images using model ($\mathcal{P}$) by (d) PD; (e) DCA; and (f) PDHG, respectively. The regularization parameter λ for both models is 17.

5 Concluding Remarks

We propose a general denoising model based on structured SPFs, as introduced in [20], and discuss various algorithms for this model. The development of these algorithms is motivated by the intrinsic structure of the model which makes it quite flexible and allows us to easily determine the convergence of the proposed methods. We illustrate the effectiveness of the proposed model by applying the modified ROF-TV model to the problem of image denoising. We see that in comparison to the traditional ROF-TV model, we are able to achieve greater accuracy without increased computation time in most cases.

Future work will feature variations of this denoising model; in particular, we are interested in the addition of a blurring kernel and applications to compressed sensing. Moreover, we believe that the structure of our proposed SPF's can be used to improve convergence results for nonconvex algorithms. Semiconvexity (or, more generally, prox-regularity) has been leveraged in this way here and elsewhere (e.g. [8], [15]), but there are many other properties of these functions which may be useful.

Disclaimer and Acknowledgment of Support

Any opinions, findings and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of AFRL (Air Force Research Laboratory). The work of L. Shen was supported in part by the National Science Foundation under grant DMS-1913039.

References

[1] L. T. H. An and P. D. Tao, *The DC (Difference of Convex Functions) programming and DCA revisited with dc models of real world nonconvex optimization problems*, Annals of Operations Research, 133 (2005), pp. 23–46.

[2] H. L. Bauschke and P. L. Combettes, *Convex Analysis and Monotone Operator Theory in Hilbert Spaces*, AMS Books in Mathematics, Springer, New York, 2011.

[3] D. M. Bradley and J. A. Bagnell, *Convex coding*, in Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence, UAI '09, Arlington, Virginia, United States, 2009, AUAI Press, pp. 83–90.

[4] E. Candes and T. Tao, *Near optimal signal recovery from random projections: Universal encoding strategies?*, IEEE Transactions on Information Theory, 52 (2006), pp. 5406–5425.

[5] F. Chen, L. Shen, and B. W. Suter, *Computing the proximity operator of the ℓ_p norm with $0 < p < 1$*, IET Signal Processing, 10 (2016), pp. 557–565.

[6] P. L. Combettes and J.-C. Pesquet, *Primal-dual splitting algorithm for solving inclusions with mixtures of composite, lipschitzian, and parallel-sum type monotone operators*, Set-Valued and Variational Analysis, 20 (2012), pp. 307–330.

[7] L. Condat, *A primal-dual splitting method for convex optimization involving lipschitzian, proximable and linear composite terms*, Journal of Optimization Theory and Applications, 158 (2013), pp. 460–479.

[8] W. Deng and W. Yin, *On the global and linear convergence of the generalized alternating direction method of multipliers*, Journal of Scientific Computing, 66 (2016), pp. 889–916.

[9] D. Donoho, *Compressed sensing*, IEEE Transactions on Information Theory, 52 (2006), pp. 1289–1306.

[10] J. Fan and R. Li, *Variable selection via nonconcave penalized likelihood and its oracle properties*, Journal of the American Statistical Association, 96 (2001), pp. 1348–1360.

[11] I. E. Frank and J. H. Friedman, *A statistical view of some chemometrics regression tools (with discussion)*, Technometrics, 35 (1993), pp. 109–148.

[12] P. Huber, *Robust Statistics*, John Wiley & Sons Inc., Hoboken, New Jersey, second ed., 2009.

[13] H. A. Le Thi and T. Pham Dinh, *DC programming and DCA: thirty years of developments*, Mathematical Programming, 169 (2018), pp. 5–68.

[14] H. A. Le Thi, T. Pham Dinh, H. Le, and X. Vo, *DC approximation approaches for sparse optimization*, European Journal of Operational Research, 244 (2015), pp. 26 – 46.

[15] T. Möllenhoff, E. Strekalovskiy, M. Moeller, and D. Cremers, *The primal-dual hybrid gradient method for semiconvex splittings*, SIAM Journal on Imaging Sciences, 8 (2015), pp. 827–857.

[16] C. A. Micchelli, L. Shen, and Y. Xu, *Proximity algorithms for image models: Denoising*, Inverse Problems, 27 (2011), p. 045009(30pp).

[17] J.-J. Moreau, *Fonctions convexes duales et points proximaux dans un espace hilbertien*, C.R. Acad. Sci. Paris Sér. A Math., 255 (1962), pp. 1897–2899.

[18] ——, *Proximité et dualité dans un espace hilbertien*, Bull. Soc. Math. France, 93 (1965), pp. 273–299.

[19] L. Shen, I. Kakadiaris, M. Papadakis, I. Konstantinidis, D. Kouri, and D. Hoffman, *Image denoising using a tight frame*, IEEE Transactions on Image Processing, 15 (2006), pp. 1254–1263.

[20] L. Shen, B. W. Suter, and E. E. Tripp, *Structured sparsity promoting functions*, Journal of Optimization Theory and Applications, accepted, (2019).

[21] E. Soubies, L. Blanc-Feraud, and G. Aubert, *A unified view of exact continuous penalties for ℓ_2-ℓ_0 minimization*, SIAM Journal on Optimization, 27 (2017), pp. 2034–2060.

[22] P. Tao and L. An, *A D.C. optimization algorithm for solving the trust-region subproblem*, SIAM Journal on Optimization, 8 (1998), pp. 476–505.

[23] R. Tibshirani, *Regression shrinkage and selection via the LASSO*, Journal of the Royal Statistical Society, Series B, 58 (1996), pp. 267–288.

[24] M. Yan, *A new primal-dual algorithm for minimizing the sum of three functions with a linear operator*, Journal of Scientific Computing, 76 (2018), pp. 1698–1717.

[25] P. Yin, Y. Lou, Q. He, and J. Xin, *Minimization of ℓ_{1-2} for compressed sensing*, SIAM Journal on Scientific Computing, 37 (2015), pp. A536–A563.

[26] W. Yin, S. Osher, D. Goldfarb, and J. Darbon, *Bregman Iterative Algorithms for ℓ_1-Minimization with Applications to Compressed Sensing*, SIAM Journal on Imaging Sciences, 1 (2008), pp. 143–168.

[27] C.-H. Zhang, *Nearly unbiased variable selection under minimax concave penalty*, Annals of Statistics, 38 (2010), pp. 894–942.