

A Memory-efficient Algorithm for Large-scale Sparsity Regularized Image Reconstruction

Greg Ongie, Naveen Murthy, Laura Balzano, Jeffrey A. Fessler

Abstract—We derive a memory-efficient first-order variable splitting algorithm for convex image reconstruction problems with non-smooth regularization terms. The algorithm is based on a primal-dual approach, where one of the dual variables is updated using a step of the Frank-Wolfe algorithm, rather than the typical proximal point step used in other primal-dual algorithms. We show in certain cases this results in an algorithm with far less memory demand than other first-order methods based on proximal mappings. We demonstrate the algorithm on the problem of sparse-view X-ray computed tomography (CT) reconstruction with non-smooth edge-preserving regularization and show competitive run-time with other state-of-the-art algorithms while using much less memory.

Index Terms—primal-dual algorithm, Frank-Wolfe algorithm, image reconstruction, sparse-view CT

I. INTRODUCTION

Sparsity regularized inverse problems arising in medical imaging result in large-scale non-smooth convex optimization problems that are computationally challenging to solve. General purpose first-order algorithms for nonsmooth convex optimization, such as subgradient descent or smoothing techniques [7], while memory-efficient, converge undesirably slow for this class of problem. Instead, specialized proximal splitting algorithms, such as the primal-dual algorithm of Chambolle-Pock (PDCP) [2], represent the current state-of-the-art for these problems. However, PDCP and other proximal methods require storing and operating on one or more dual variables with dimensions potentially several times larger than the image volume to be reconstructed. Even for realistic problem sizes arising in X-ray CT reconstruction, storing and operating on these additional dual variables may be prohibitive or infeasible, limiting the scope of these algorithms in practice. This is especially true when using 3D regularization with all 26 neighboring voxels [11], or when computing on GPUs that are limited in memory relative to traditional CPUs.

To address this issue, this paper introduces a novel algorithm that has far less memory demand than previous approaches. Specifically, we focus on reconstruction via weighted least squares with a non-smooth edge-preserving regularization term. This includes the total variation semi-norm and related penalties. The proposed algorithm is based on a novel primal-dual approach. Existing first-order primal-dual approaches [2], [3] alternate between updating the primal variable with a

gradient descent step and the dual variable with a projected gradient ascent step (or their proximal equivalents). The main idea of the proposed approach is to replace the projected gradient ascent step in the dual update with a step of the Frank-Wolfe algorithm [4], [6]. We show that this modification allows for substantial memory savings over standard primal-dual approaches. In particular, the algorithm requires storing at most two additional auxiliary variables with dimensions matching the primal variable. We prove convergence of the algorithm under certain assumptions on its step-sizes.

Finally, we demonstrate the proposed algorithm by reconstructing a sparse-view X-ray CT dataset. Empirically, the proposed algorithm shows competitive convergence with state-of-the-art proximal splitting methods for this problem [10], [8], but with much less memory demand.

II. PROBLEM FORMULATION

We consider the following optimization problem:

$$\min_{\mathbf{x}} \frac{1}{2} \|\mathbf{A}\mathbf{x} - \mathbf{b}\|_{\mathbf{W}}^2 + \lambda R(\mathbf{x}). \quad (1)$$

Here $\mathbf{x} \in \mathbb{R}^n$ represents a vectorized discrete image. The first term in (1) measures the data-fit, where $\mathbf{A} \in \mathbb{R}^{m \times n}$ with $m \leq n$ is a matrix representing the linear measurement operator, $\mathbf{b} \in \mathbb{R}^m$ are the (noisy) measurements, and $\|\cdot\|_{\mathbf{W}}$ is the weighted ℓ^2 -norm defined as $\|\mathbf{z}\|_{\mathbf{W}} = \sqrt{\mathbf{z}^T \mathbf{W} \mathbf{z}}$ for a fixed diagonal matrix $\mathbf{W} \in \mathbb{R}^{m \times m}$ with positive diagonal entries. The second term is a regularization penalty, where $\lambda > 0$ is a parameter balancing the tightness of data-fit and regularization. This work focuses on regularizers of the form $R(\mathbf{x}) = \varphi(\mathbf{D}\mathbf{x})$ for some *regularization transform* $\mathbf{D} \in \mathbb{R}^{N \times n}$ and where $\varphi(\mathbf{y}) := \sum_i \phi(\mathbf{y}_i)$ for some convex and possibly non-smooth sparsity promoting *potential function* $\phi : \mathbb{R} \rightarrow \mathbb{R}^+$. We assume the regularization transform \mathbf{D} is a tall matrix ($N \gg n$) having block form $\mathbf{D} = [\mathbf{D}_1^T \ \mathbf{D}_2^T \ \cdots \ \mathbf{D}_\ell^T]^T$ with $\mathbf{D}_i \in \mathbb{R}^{N_i \times n}$. For example, if $\{\mathbf{D}_i\}_{i=1}^\ell$ is a collection of first-order finite-difference operators in each dimension and $\phi(x) = |x|$ then $\varphi(\mathbf{D}\mathbf{x}) = \|\mathbf{D}\mathbf{x}\|_1$ is the discrete (anisotropic) total variation (TV) of \mathbf{x} . Other choices for $\{\mathbf{D}_i\}_{i=1}^\ell$ include oriented higher-order finite differences [5] or a collection of pre-trained sparsifying transforms [12]. Likewise, the proposed method also generalizes to other convex potential functions ϕ , such as the Huber loss or Fair potential.

III. PRIMAL-DUAL FRANK-WOLFE ALGORITHM

For ease of exposition we focus on the case $\varphi(\mathbf{y}) = \|\mathbf{y}\|_1$ in the remainder of this work, *i.e.*, $R(\mathbf{x}) = \|\mathbf{D}\mathbf{x}\|_1$. Because the ℓ^∞ -norm is dual to the ℓ^1 -norm, the primal problem (1)

G. Ongie, N. Murthy, L. Balzano, and J.A. Fessler are with University of Michigan, Department of Electrical Engineering and Computer Science, Ann Arbor, MI, USA. E-mail: {gongie, nnmurthy, girasole, fessler}@umich.edu. This work was supported in part by NIH Grant U01 EB018753, DARPA-16-43-D3M-FP-037, and NSF ECCS-1508943.

with an ℓ^1 regularizer has an equivalent primal-dual saddle point formulation¹ given by

$$\min_{\mathbf{x}} \max_{\|\mathbf{y}\|_{\infty} \leq \lambda} \frac{1}{2} \|\mathbf{Ax} - \mathbf{b}\|_{\mathbf{W}}^2 + \langle \mathbf{Dx}, \mathbf{y} \rangle \quad (2)$$

Partitioning the dual vector as $\mathbf{y}^T = [\mathbf{y}_1^T \cdots \mathbf{y}_\ell^T]$ corresponding to the blocks $\mathbf{D}^T = [\mathbf{D}_1^T \cdots \mathbf{D}_\ell^T]$, we re-express the inner product in (2) as

$$\langle \mathbf{Dx}, \mathbf{y} \rangle = \sum_{i=1}^{\ell} \langle \mathbf{D}_i \mathbf{x}, \mathbf{y}_i \rangle = \left\langle \mathbf{x}, \sum_{i=1}^{\ell} \mathbf{D}_i^T \mathbf{y}_i \right\rangle.$$

Therefore, introducing the auxiliary variable $\mathbf{z} \triangleq \sum_{i=1}^{\ell} \mathbf{D}_i^T \mathbf{y}_i$, (2) is equivalent to the equality constrained problem:

$$\min_{\mathbf{x}} \max_{\substack{\|\mathbf{y}_i\|_{\infty} \leq \lambda, \\ i=1, \dots, \ell}} \frac{1}{2} \|\mathbf{Ax} - \mathbf{b}\|_{\mathbf{W}}^2 + \langle \mathbf{x}, \mathbf{z} \rangle \quad \text{s.t.} \quad \mathbf{z} = \sum_{i=1}^{\ell} \mathbf{D}_i^T \mathbf{y}_i. \quad (3)$$

Below we show that the proposed algorithm only needs to maintain an estimate of the auxiliary variable $\mathbf{z} \in \mathbb{R}^n$, which has dimensions of the image volume, rather than the full dual variable $\mathbf{y} \in \mathbb{R}^N$ that is generally several times larger.

We also dualize the data-fit term by defining $g(\mathbf{t}) := \frac{1}{2} \|\mathbf{t} - \mathbf{b}\|_{\mathbf{W}}^2$ and using the identity

$$g(\mathbf{Ax}) = \max_{\mathbf{t}} \langle \mathbf{t}, \mathbf{Ax} \rangle - g^*(\mathbf{t}),$$

where g^* is the convex conjugate of g . Simple analysis yields $g^*(\mathbf{t}) = \frac{1}{2} \|\mathbf{t} + \mathbf{Wb}\|_{\mathbf{W}^{-1}}^2 - \frac{1}{2} \|\mathbf{b}\|_{\mathbf{W}}^2$. Inserting this into (3) and dropping constant terms yields the equivalent formulation

$$\min_{\mathbf{x}} \max_{\mathbf{t}} \max_{\substack{\|\mathbf{y}_i\|_{\infty} \leq \lambda, \\ i=1, \dots, \ell}} \langle \mathbf{t}, \mathbf{Ax} \rangle - \frac{1}{2} \|\mathbf{t} + \mathbf{Wb}\|_{\mathbf{W}^{-1}}^2 + \langle \mathbf{x}, \mathbf{z} \rangle \quad (4)$$

$$\text{subject to } \mathbf{z} = \sum_{i=1}^{\ell} \mathbf{D}_i^T \mathbf{y}_i.$$

A. Frank-Wolfe dual update

The Frank-Wolfe (FW) algorithm [4], [6], also known as the conditional gradient method, is a projection-free approach to solving constrained problems of the form

$$\max_{\mathbf{y} \in \mathcal{C}} f(\mathbf{y}),$$

where f is a concave function and \mathcal{C} is a closed, convex set. At each iteration, the FW algorithm solves for a search-direction \mathbf{s}^* via

$$\mathbf{s}^* = \arg \max_{\mathbf{s} \in \mathcal{C}} \langle \mathbf{s}, \nabla f(\mathbf{y}^{(k)}) \rangle$$

then updates \mathbf{y} with a convex combination of the previous iterate $\mathbf{y}^{(k)}$ and the search direction

$$\mathbf{y}^{(k+1)} = (1 - \alpha_k) \mathbf{y}^{(k)} + \alpha_k \mathbf{s}^*$$

where α_k is some iteration-dependent step-size.

If we apply one step of the FW algorithm to the dual variable \mathbf{y} in (2) while holding the primal variable \mathbf{x} fixed, then the function to maximize is simply the linear function

¹For a general regularization penalty of the form $\varphi(\mathbf{Dx})$, we can derive a similar saddle-point formulation (2) by writing $\varphi(\mathbf{Dx}) = \max_{\mathbf{y}} \langle \mathbf{Dx}, \mathbf{y} \rangle - \varphi^*(\mathbf{y})$ where φ^* is the convex conjugate of φ .

$f(\mathbf{y}) = \langle \mathbf{Dx}, \mathbf{y} \rangle$ with $\nabla f(\mathbf{y}) = \mathbf{Dx}$ subject to $\|\mathbf{y}\|_{\infty} \leq \lambda$. The FW search-direction update in this case is

$$\mathbf{s}^* = \arg \max_{\|\mathbf{s}\|_{\infty} \leq \lambda} \langle \mathbf{s}, \mathbf{Dx} \rangle = \lambda \text{sign}(\mathbf{Dx}).$$

where $\text{sign}(\cdot)$ is applied entrywise and we define $\text{sign}(0) = 0$. Hence, a FW update of \mathbf{y} has the form

$$\mathbf{y}^{(k+1)} = (1 - \alpha_k) \mathbf{y}^{(k)} + \alpha_k \lambda \text{sign}(\mathbf{Dx}).$$

A key to saving memory is that the above update is separable in terms of the \mathbf{y}_i -blocks:

$$\mathbf{y}_i^{(k+1)} = (1 - \alpha_k) \mathbf{y}_i^{(k)} + \alpha_k \lambda \text{sign}(\mathbf{D}_i \mathbf{x}), \quad \text{for all } i = 1, \dots, \ell.$$

Applying \mathbf{D}_i^T to both sides above and summing over i yields

$$\mathbf{z}^{(k+1)} = (1 - \alpha_k) \mathbf{z}^{(k)} + \alpha_k \lambda \sum_{i=1}^{\ell} \mathbf{D}_i^T \text{sign}(\mathbf{D}_i \mathbf{x}) \quad (5)$$

where we define $\mathbf{z}^{(k)} = \sum_{i=1}^{\ell} \mathbf{D}_i^T \mathbf{y}_i^{(k)}$ for all $k \geq 0$. To save memory, we compute $\mathbf{z}^{(k+1)}$ incrementally, first by rescaling the current estimate by $(1 - \alpha_k)$ then by adding $\alpha_k \lambda \mathbf{D}_i^T \text{sign}(\mathbf{D}_i \mathbf{x})$ for all $i = 1, \dots, \ell$ in sequence.

B. Proximal dual update

Similar to other primal-dual approaches [2], [3], to update the dual variable \mathbf{t} we take one step of a proximal point algorithm applied to (4) while fixing the other variables. Specifically, given the current iterates $(\mathbf{x}^{(k)}, \mathbf{t}^{(k)})$, we set

$$\mathbf{t}^{(k+1)} = \arg \max_{\mathbf{t}} \langle \mathbf{t}, \mathbf{Ax}^{(k)} \rangle - \frac{1}{2} \|\mathbf{t} + \mathbf{Wb}\|_{\mathbf{W}^{-1}}^2 - \frac{1}{2\sigma_k} \|\mathbf{t} - \mathbf{t}^{(k)}\|_{\mathbf{W}^{-1}}^2$$

where $\sigma_k > 0$ is a step-size parameter to be specified later. This has the closed form solution

$$\mathbf{t}^{(k+1)} = \frac{1}{1 + \sigma_k} \mathbf{t}^{(k)} + \frac{\sigma_k}{1 + \sigma_k} \mathbf{W}(\mathbf{Ax}^{(k)} - \mathbf{b}). \quad (6)$$

C. Primal update

Finally, we update the primal variable \mathbf{x} via a gradient descent step (or equivalently a proximal-point step) applied to (4) with the dual variables fixed:

$$\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} - \tau_k (\mathbf{z}^{(k+1)} + \mathbf{A}^T \mathbf{t}^{(k+1)}) \quad (7)$$

where $\tau_k > 0$ is a step-size parameter to be specified later. Inspired by [2] we include an optional over-relaxation step:

$$\bar{\mathbf{x}}^{(k+1)} = \mathbf{x}^{(k+1)} + \theta (\mathbf{x}^{(k+1)} - \mathbf{x}^{(k)}), \quad (8)$$

where $\theta \in [0, 1]$, and perform the dual variable updates (5) and (6) with $\bar{\mathbf{x}}^{(k)}$ in place of $\mathbf{x}^{(k)}$.

D. Algorithm summary and convergence

Algorithm 1 summarizes the proposed primal-dual Frank-Wolfe (PDFW) algorithm. Using similar analysis as in [1] we are able to prove the following convergence result for Algorithm 1 in the special case $\theta = 0$ by showing it is a particular instance of an ϵ -subgradient descent method; we omit the proof for brevity.

Theorem 1. *Let \mathcal{X}^* denote the set of minimizers to (1), and let $\{\mathbf{x}^k\}_{k=1}^{\infty}$ be the iterates generated by Algorithm 1 with $\theta = 0$.*

Algorithm 1 Primal-Dual Frank-Wolfe (PDFW)

Initialize $\mathbf{x}^{(0)} = \bar{\mathbf{x}}^{(0)}, \mathbf{z}^{(0)} = \mathbf{0} \in \mathbb{R}^n$ and $\mathbf{t}^{(0)} = \mathbf{0} \in \mathbb{R}^m$.
 Choose step-sizes $(\tau_k, \sigma_k, \alpha_k), \theta \in [0, 1]$.
for $k = 0, \dots, k_{max}$ **do**
 $\mathbf{t}^{(k+1)} = \frac{1}{1+\sigma_k} \mathbf{t}^{(k)} + \frac{\sigma_k}{1+\sigma_k} \mathbf{W}(\mathbf{A}\bar{\mathbf{x}}^{(k)} - \mathbf{b})$
 $\mathbf{z}^{(k+1)} = (1 - \alpha_k) \mathbf{z}^{(k)} + \alpha_k \lambda \sum_{i=1}^{\ell} \mathbf{D}_i^T \text{sign}(\mathbf{D}_i \bar{\mathbf{x}}^{(k)})$
 $\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} - \tau_k (\mathbf{A}^T \mathbf{t}^{(k+1)} + \mathbf{z}^{(k+1)})$
 $\bar{\mathbf{x}}^{(k+1)} = \mathbf{x}^{(k+1)} + \theta (\mathbf{x}^{(k+1)} - \mathbf{x}^{(k)})$
end for

Suppose the iterates $\{\mathbf{x}^{(k)}\}_{k=1}^{\infty}$ are bounded. If the step-size sequences $\{\alpha_k\} \subset [0, 1]$, $\{\sigma_k\} \subset (0, \infty)$, and $\{\tau_k\} \subset (0, \infty)$ satisfy $\tau_k \rightarrow 0$, $\sum_{k=0}^{\infty} \tau_k = +\infty$, and

$$\sum_{j=1}^k \tau_{j-1} \prod_{i=j}^k (1 - \alpha_i) \rightarrow 0, \quad \sum_{j=1}^k \tau_{j-1} \prod_{i=j}^k \frac{1}{1 + \sigma_k} \rightarrow 0$$

as $k \rightarrow \infty$, then $\text{dist}(\mathbf{x}^k, \mathcal{X}^*) \rightarrow 0$, where dist denotes the Euclidean distance of a point to a set. In particular, if the solution \mathbf{x}^* to (1) is unique then $\mathbf{x}^k \rightarrow \mathbf{x}^*$.

The step-size conditions in Theorem 1 are satisfied, for example, when $\tau_k = O(\frac{1}{k^p})$, $0 < p \leq 1$, α_k and σ_k are constant. There are also valid choices of $\tau_k, \alpha_k, \sigma_k$ for which $\alpha_k \rightarrow 0$ and $\sigma_k \rightarrow \infty$, such as $\tau_k = O(\frac{1}{k^p})$, $\alpha_k = O(\frac{1}{k^q})$, and $\sigma_k = O(\frac{1}{\tau_k})$ with $0 < p < 1$ and $0 < q < p/2$.

Empirically, we observe improved convergence rates using $\theta = 1$ and a constant step-size $\tau_k = \tau$. However, our current proof of Theorem 1 does not extend to the case $\theta \neq 0$ nor to the case of τ_k constant, and we leave its convergence under these conditions as an open problem for future work.

E. Connections to Chambolle-Pock primal dual algorithm

Algorithm 1 is closely related to the primal-dual algorithm of Chambolle-Pock (PDCP) [2]. If we introduce an auxiliary variable $\mathbf{s}^{(k)} \in \mathbb{R}^N$ and replace the $\mathbf{z}^{(k+1)}$ update in Algorithm 1 with the alternative update

$$\mathbf{z}^{(k+1)} = \mathbf{D}^T \mathbf{s}^{(k+1)} := \mathbf{D}^T \text{proj}_{\|\cdot\|_{\infty} \leq \lambda} (\mathbf{s}^{(k)} + \sigma_k \mathbf{D} \bar{\mathbf{x}}^{(k)})$$

where $\text{proj}_{\|\cdot\|_{\infty} \leq \lambda}$ denotes Euclidean projection onto the set $\{\mathbf{s} : \|\mathbf{s}\|_{\infty} \leq \lambda\}$, then this modified version of Algorithm 1 coincides with PDCP applied to (4). In [2] it is shown that PDCP converges when $\sigma_k = \sigma$ and $\tau_k = \tau$ are constant and $\tau \sigma L^2 < 1$ and $\theta = 1$, where L is the operator norm of the concatenated matrix $[\mathbf{A}^T, \mathbf{D}^T]^T$.

F. Memory benefits

Table 1 summarizes the memory requirements of different first-order proximal methods for solving (2). An important feature of Algorithm 1 is that it only requires storing at most three arrays having the size of the image volume to be reconstructed. In contrast, the linearized augmented Lagrangian method (LALM) of [8] would need to store several arrays have the same size as the image plus two additional arrays of size N , the output dimension of the regularization transform. Similarly, the PDCP algorithm [2] implemented as

	number of variables of size			total memory 3D CT example (in GB)
	$\mathbf{x} \in \mathbb{R}^n$ (image)	$\mathbf{D}\mathbf{x} \in \mathbb{R}^N$ (reg. transform)	$\mathbf{b} \in \mathbb{R}^m$ (data)	
LALM [8]	4	2	2	3.02
PDCP [2], [10]	2	1	2	1.60
PDFW, $\theta = 1$	3	0	2	0.47
PDFW, $\theta = 0$	2	0	2	0.38

TABLE I
MEMORY DEMANDS OF FIRST-ORDER METHODS FOR SOLVING (1).

in [10] needs to store at least one array of size N . The last column of Table 1 we illustrates the memory demand of these algorithms for the iterative reconstruction of a 3D axial CT scan as specified in the next section. The proposed PDFW algorithm requires an order of magnitude less memory for this example because it avoids having to store large auxiliary variables associated with the regularization transformed image.

IV. EXPERIMENTS

Here we demonstrate the proposed PDFW algorithm's potential for sparse-view X-ray CT reconstruction. We simulate an axial CT scan of the XCAT phantom [9] of size $1024 \times 1024 \times 154$ voxels to obtain a sinogram of size $m = 888 \times 64 \times 120$ (channels \times rows \times views) and reconstruct on a coarser grid of size $n = 512 \times 512 \times 90$. Our reconstruction is obtained by solving (1) with regularizer $R(\mathbf{x}) = \|\mathbf{D}\mathbf{x}\|_1$ where \mathbf{D} computes all finite-differences with thirteen nearest-neighbors of each voxel. We set the statistical weighting matrix $\mathbf{W} = \mathbf{I}$, and set the regularization parameter $\lambda = 4096$. We compare against two state-of-the-art first-order algorithms for solving (1): PDCP as adapted to CT reconstruction in [10], and the linearized augmented Lagrangian method² (LALM) of [8]. For the proposed PDFW algorithm we test two sets of step-sizes. The first set (S1) is designed to satisfy the conditions of Theorem 1:

$$\tau_k = \frac{2}{2+k}, \quad \sigma_k = \frac{1}{L^2 \tau_k}, \quad \alpha_k = \left(\frac{2}{2+k}\right)^{0.49}, \quad \theta = 0. \quad (\text{S1})$$

The second set (S2) uses a constant step-size $\tau_k = \tau$, violating the conditions of Theorem 1, but matches the settings proposed for the PDCP algorithm in [10] (except for the choice of α_k):

$$\tau_k = 1/L, \quad \sigma_k = 1/L, \quad \alpha_k = \frac{2}{2+k}, \quad \theta = 1. \quad (\text{S2})$$

Figure 1 shows cropped images from the central transaxial plane initial filtered back projection reconstruction, reference solution, and the reconstruction obtained from the proposed PDFW algorithm with settings (S2) after 500 iterations. We obtained a reference solution to the optimization problem by running several thousands of iterations of the LALM algorithm, which reached the smallest cost among the competing algorithms. Observe that there is almost no visual difference between the reference solution and the PDFW solution after 500 iterations.

Figure 2 compares the performance of the algorithms with respect to two convergence metrics: (1) the *normalized cost*

²An ordered subsets variant of LALM is also presented in [8]. The proposed PDFW algorithm could also be modified to include ordered subsets updates, but is outside the scope of this work. For fair comparison, we compare against LALM without ordered subsets.



Fig. 1. Reconstructions of central transaxial plane of XCAT phantom from sparse-view measurements (left to right): filtered back projection (FBP) reconstruction, a reference solution, and a reconstruction obtained after running 500 iterations of the proposed PDFW algorithm with settings (S2). Images displayed in HU (modified so that air is 0) clipped to range $[800, 1200]$ and cropped to the region of interest.

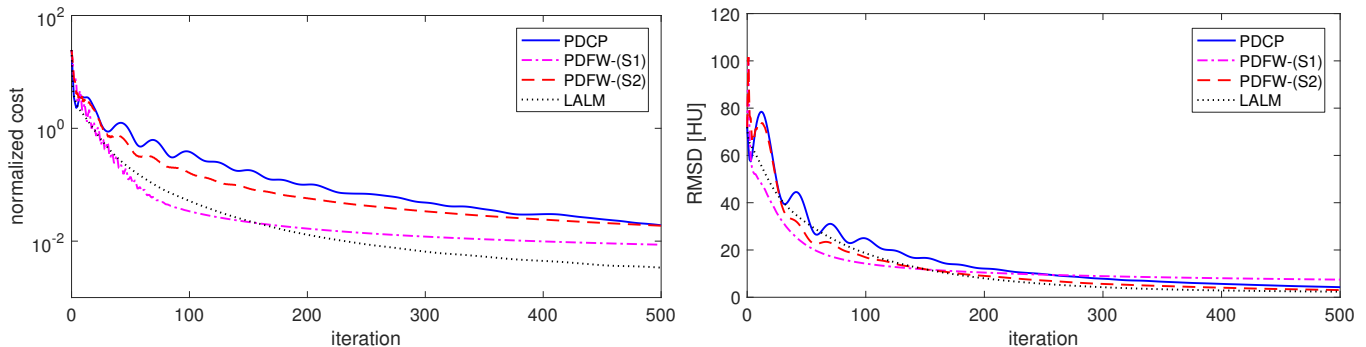


Fig. 2. Plots of convergence metrics using the proposed PDFW algorithm with settings (S1) and (S2), the PDCP algorithm [10], and LALM algorithm [8].

defined as $(f(\mathbf{x}^{(k)}) - f(\mathbf{x}^*)) / f(\mathbf{x}^*)$ where $f(\mathbf{x})$ is the cost function in (1), $\mathbf{x}^{(k)}$ is the k th iteration of a given algorithm, and \mathbf{x}^* is the reference solution; and (2) *the root mean square difference* (RMSD) computed as $\text{RMSD} = \sqrt{\frac{1}{|\Omega|} \sum_{i \in \Omega} |\mathbf{x}_i^{(k)} - \mathbf{x}_i^*|^2}$ where Ω is the index set of voxels in a cylindrical region of interest containing the phantom anatomy. Overall, the LALM algorithm performs best in terms of the convergence metrics, reaching the lowest cost and RMSD after 500 iterations. However, we reiterate that the LALM algorithm has the highest memory demand of the compared methods (see Table I). The proposed PDFW algorithm with step-size scheme (S1) shows a fast initial decrease in the cost and RMSD, but slows in improvement after 100 iterations and has the highest RMSD after 500 iterations, indicating that the (S1) step-size scheme may yield slow asymptotic convergence. The PDFW algorithm with step-size scheme (S2) has better long-run performance in RMSD, yielding nearly the same as LALM after 500 iterations, and its reduction in normalized cost is similar to PDCP.

V. CONCLUSION

We introduce a memory-efficient algorithm for solving large-scale convex image reconstruction problems with transform sparse regularization based on a novel hybrid of proximal methods and the Frank-Wolfe algorithm. Our experiments demonstrate that the algorithm has competitive performance with other first-order algorithms but with substantially less memory demand. In our experiments we use all of the sinogram measurements to update the primal variable in each

iteration, but the proposed algorithm could potentially be modified to incorporate ordered subsets updates similar to [8] for improved computational efficiency and faster convergence.

REFERENCES

- [1] S. Bonettini and V. Ruggiero. On the convergence of primal-dual hybrid gradient algorithms for total variation image restoration. *J. Math. Im. Vision*, 44(3):236–53, Nov. 2012.
- [2] A. Chambolle and T. Pock. A first-order primal-dual algorithm for convex problems with applications to imaging. *J. Math. Im. Vision*, 40(1):120–145, 2011.
- [3] L. Condat. A primal-dual splitting method for convex optimization involving Lipschitzian, proximable and linear composite terms. *J. of Opt. Theory and Applications*, 158(2):460–479, Dec 2012.
- [4] M. Frank and P. Wolfe. An algorithm for quadratic programming. *Naval research logistics quarterly*, 3(1-2):95–110, 1956.
- [5] Y. Hu, G. Ongie, S. Ramani, and M. Jacob. Generalized higher degree total variation regularization. *IEEE TIP*, 23(6):2423–2435, 2014.
- [6] M. Jaggi. Revisiting Frank-Wolfe: Projection-free sparse convex optimization. In *ICML*, pages 427–435, 2013.
- [7] Y. Nesterov. Smooth minimization of non-smooth functions. *Mathematical Programming*, 103(1):127–52, May 2005.
- [8] H. Nien and J. A. Fessler. Fast X-ray CT image reconstruction using a linearized augmented Lagrangian method with ordered subsets. *IEEE TIP*, 34(2):388–99, Feb. 2015.
- [9] W. P. Segars, M. Mahesh, T. J. Beck, E. C. Frey, and B. M. W. Tsui. Realistic CT simulation using the 4D XCAT phantom. *Med. Phys.*, 35(8):3800–8, Aug. 2008.
- [10] E. Y. Sidky, J. H. Jorgensen, and X. Pan. Convex optimization problem prototyping for image reconstruction in computed tomography with the Chambolle-Pock algorithm. *Phys. Med. Biol.*, 57(10):3065–92, 2012.
- [11] J.-B. Thibault, K. Sauer, C. Bouman, and J. Hsieh. A three-dimensional statistical approach to improved image quality for multi-slice helical CT. *Med. Phys.*, 34(11):4526–44, Nov. 2007.
- [12] B. Wen, S. Ravishanker, and Y. Bresler. Structured overcomplete sparsifying transform learning with convergence guarantees and applications. *Int. J. of Computer Vision*, 114(2-3):137–167, 2015.