An Optimal Algorithm for Stochastic Three-Composite Optimization

Renbo Zhao
ORC, MIT

William B. Haskell
ISEM, NUS

Vincent Y. F. Tan
ECE & Mathematics, NUS

Abstract

We develop an optimal primal-dual first-order algorithm for a class of stochastic three-composite convex minimization problems. The convergence rate of our method not only improves upon the existing methods, but also matches a lower bound derived for all first-order methods that solve this problem. We extend our proposed algorithm to solve a composite stochastic program with any finite number of nonsmooth functions. In addition, we generalize an optimal stochastic alternating direction method of multipliers (SADMM) algorithm proposed for the two-composite case to solve this problem, and establish its connection to our optimal primal-dual algorithm. We perform extensive numerical experiments on a variety of machine learning applications to demonstrate the superiority of our method via-à-vis the state-of-the-art.

1 INTRODUCTION

Consider the three-composite convex minimization problem (TCMP)

$$\min_{\mathbf{x}\in\mathbb{R}^d} \left[P(\mathbf{x}) \triangleq f(\mathbf{x}) + g(\mathbf{x}) + h(\mathbf{A}\mathbf{x}) \right], \tag{1}$$

where $f, g : \mathbb{R}^d \to \overline{\mathbb{R}} \triangleq \mathbb{R} \cup \{+\infty\}$ and $h : \mathbb{R}^m \to \overline{\mathbb{R}}$ are convex, closed, and proper (CCP) functions, and the linear operator $\mathbf{A} : \mathbb{R}^d \to \mathbb{R}^m$ has operator norm $B > 0$. (Throughout this work, both $\mathbb{R}^d$ and $\mathbb{R}^m$ are Euclidean spaces.) In addition, f is continuously differentiable with L-Lipschitz gradient ($L > 0$) on $\mathbb{R}^d$, and g and h have "simple" proximal operators, e.g., those that can be evaluated in closed forms. We denote the solution set of Problem (1) by $\mathcal{X}^*$ and assume that $\mathcal{X}^* \neq \emptyset$. We also assume that Slater's condition holds for Problem (1), i.e., $\mathbf{ri}\,(\mathbf{dom}\, P) \neq \emptyset$.

We focus on the stochastic setting in which $f(\mathbf{x}) \triangleq \mathbb{E}_{\boldsymbol{\xi}\sim\nu}[F(\mathbf{x}, \boldsymbol{\xi})]$, where $\boldsymbol{\xi}$ is a random variable with distribution ν. We assume that there exists a stochastic first-order oracle $\mathsf{SFO}(f, \sigma)$ that upon a query at $\mathbf{x} \in \mathbb{R}^d$, returns an unbiased estimate of $\nabla f(\mathbf{x})$ with variance σ^2, conditioned on past history. (See Assumption 1 for precise statements.)

In statistical learning, Problem (1) represents a *doubly regularized* expected risk minimization problem that includes many important instances, such as (graph-guided) fused lasso [1, 2], matrix completion [3], portfolio optimization [4] and graph-guided sparse logistic regression [5]. Note that if we set the distribution ν to be the empirical distribution defined on the training data, we indeed recover the *empirical risk minimization* problem. In this case, the oracle $\mathsf{SFO}(f, \sigma)$ simply returns the stochastic gradient obtained by random mini-batch sampling. Beyond statistical learning, Problem (1) also arises in many other important areas, such as two-stage stochastic programming [6] and constrained TV-denoising [7]. Such wide applications are due to the flexibility of the (possibly nonsmooth) functions g and h, i.e., they can either be regularizers or encode constraints (e.g., linear (in-)equalities or ellipsoidal constraints).

Due to its *wide applicability*, Problem (1) has recently received considerable attention. When $\boldsymbol{\xi}$ is deterministic, many algorithms have been developed to solve (1). These methods include [8–17]. However, the studies of the stochastic setting are relatively limited. In particular, it is unclear whether there are any existing stochastic methods for solving Problem (1) that are (minimax) optimal. In this work, we show that the answer to this question is indeed negative. We do so by developing an *optimal* algorithm with a superior convergence rate compared to the existing methods.

1.1 Saddle-Point Form

Let $h^* : \mathbb{R}^m \to \overline{\mathbb{R}}$ denote the Fenchel conjugate of h. To use the proximal operator of h (or h^*), we need

Proceedings of the 22nd International Conference on Artificial Intelligence and Statistics (AISTATS) 2019, Naha, Okinawa, Japan. PMLR: Volume 89.

to decouple the function h and the linear operator $\mathbf{A}$. To achieve this, we introduce the saddle-point form of Problem (1), i.e.,

$$\min_{\mathbf{x}\in\mathbb{R}^d}\max_{\mathbf{y}\in\mathbb{R}^m}[S(\mathbf{x},\mathbf{y}) \triangleq f(\mathbf{x}) + g(\mathbf{x}) + \langle \mathbf{A}\mathbf{x},\mathbf{y}\rangle - h^*(\mathbf{y})]. \quad (2)$$

By [18, Theorem 36.6], under Slater's condition, $\mathbf{x}^*$ is an optimal solution of Problem (1) if and only if there exists $\mathbf{y}^* \in \mathbb{R}^m$ such that $(\mathbf{x}^*, \mathbf{y}^*)$ is a saddle point of Problem (2). Thus, to find an optimal solution of Problem (1), it suffices to find a saddle point of Problem (2).

1.2 Related Works

We first review the methods that solve Problem (1). If we disregard the specific structure of Problem (1), then it can be solved by methods involving the stochastic subgradient. Concretely, we can apply the stochastic subgradient method [19–22] to P, by treating it as a general nonsmooth function. As for more sophisticated approaches, we can view $\widehat{f} \triangleq f + h\circ\mathbf{A}$ as a nonsmooth function, and apply stochastic proximal subgradient [23, 24] or regularized dual averaging [25] to $P = \widehat{f} + g$. However, to use these methods, we need to assume that the (stochastic) subgradients of P or $\widehat{f}$ are uniformly bounded. This may fail to hold in general. In addition, subgradient-based methods converge slowly in practice.

When $\mathbf{A} = \mathbf{I}$, i.e., the identity operator, two specialized methods have been recently proposed. Specifically, [26] proposed an accelerated stochastic gradient method with proximal average [27], and [28] developed a stochastic gradient method based on three-operator splitting [29]. However, these two methods fail to handle the general linear operator $\mathbf{A}$. Additionally, the method in [28] can *only* handle strongly convex f.

Next, we turn our attention to the methods that solve Problem (2). Similar to the discussions above, for a general nonsmooth convex-concave function, we can apply the stochastic primal-dual subgradient method [20, 30] to find its saddle point. These methods suffer from the same problems as the subgradient-based methods mentioned above. When $g \equiv 0$ (or the indicator function of a linear subspace), many algorithms [31–33] based on the primal-dual hybrid gradient (PDHG) framework [34, 35] have been proposed to solve Problem (2). However, these methods cannot handle a general nonsmooth function g. This limitation has been recently overcome by [36], wherein a three-composite stochastic PDHG algorithm was proposed.

Finally, we note that by introducing a slack variable $\mathbf{y} = \mathbf{A}\mathbf{x}$, Problem (1) can be rewritten as a linearly constrained composite stochastic program. This program can be solved via SADMM algorithms [37–39], by regarding $\widetilde{f} \triangleq f + g$ as a nonsmooth function and leveraging its stochastic subgradient. To exploit the composite structure of $\widetilde{f}$, [40] develop a new SADMM algorithm that makes use of the proximal operator of g, rather than its subgradient.

1.3 Lower Bound and Optimality

We measure the convergence rate of any stochastic method that solves Problem (2) by the expected primal-dual gap (see (18) for its definition). Let K be the total number of iterations. When $g \equiv 0$, a lower bound on the convergence rates of all the methods that solve Problem (2) under $\mathsf{SFO}(f,\sigma)$ has been derived in [31], i.e.,

$$\Omega\left(L/K^2 + B/K + \sigma/\sqrt{K}\right). \quad (3)$$

This bound clearly holds for Problem (2) when g is a general nonsmooth function.[1] Since we focus on the methods whose number of iterations is proportional to the number of oracle queries, the convergence rate reflects the oracle complexity. All of the existing methods (including the ones in this work) have this property.

However, to the best of our knowledge, none of the existing (stochastic) methods achieves (3). The state-of-the-art method in [36] has convergence rate $O(L/K + B/K + \sigma/\sqrt{K})$, when K is known a priori and used in setting the parameters in the algorithm. However, in many scenarios, the algorithm is terminated by other criteria other than the total number of iterations, so the knowledge of K may be unavailable. If K is unknown, the rate degrades to $O(L/K + B\log K/K + \sigma\log K/\sqrt{K})$.

We remark that while many previous works only focus on achieving the optimal dependence on σ (which dominates asymptotically), obtaining optimal dependence on L (and B) is important as well. This is because in many practical applications, due to ill-conditioned data, the value of L can be significantly larger than B and σ. Hence the term involving L dominates for moderate K, which often appears due to time constraints or low-accuracy requirements. The benefits of achieving optimal dependence on L yielded by our method will be illustrated through the extensive numerical experiments in Section 5.

1.4 Main Contributions

Our main contributions are threefold.

[1] Due to the equivalence of Problems (1) and (2), the lower bound (3) also applies to all the methods for solving Problem (1). In this case, the convergence is measured by the expected primal sub-optimality gap.

First, we develop an optimal primal-dual algorithm for solving Problem (2) whose convergence rate in expectation matches the lower bound (3), even when K is unknown a priori. (See Remark 2 for the *innovations and technical difficulties* to achieve this.) This also implies that the lower bound derived for $g \equiv 0$ is indeed tight for the more general Problem (2). Additionally, we also derive large-deviation-type convergence results for the primal-dual gap. Such results complement the convergence in-expectation results and have practical significance. See Remark 8 for details.

Second, we extend our proposed algorithm to handle the sum of any finite number of nonsmooth functions, each coupled with a linear operator. This formulation subsumes many important applications, e.g., (sparse) overlapping group lasso [41, 42]. We also provide convergence analysis for this extension.

Third, we generalize the optimal SADMM algorithm in [39] to solve Problem (1), and establish the connection of this generalized algorithm to our primal-dual method for solving Problem (2).

Notations. Denote the set of nonnegative integers by $\mathbb{Z}^+$ and define $\mathbb{N} \triangleq \mathbb{Z}^+ \setminus \{0\}$. We denote the Euclidean inner product by $\langle \cdot, \cdot \rangle$ and $\| \cdot \|$ the norm induced by $\langle \cdot, \cdot \rangle$. We use bold lowercase letters and bold uppercase letters to denote vectors and matrices, respectively. For a (bounded) linear operator $\mathbf{A}$, we use $\mathbf{A}^T$ to denote its adjoint and $|||\mathbf{A}|||$ its operator norm. For any $n \in \mathbb{N}$, define $[n] \triangleq \{1, \ldots, n\}$. For any CCP function $h : \mathbb{R}^m \to \overline{\mathbb{R}}$, define $\mathbf{dom}\, h \triangleq \{\mathbf{y} \in \mathbb{R}^m \,|\, h(\mathbf{y}) < +\infty\}$ and for any $t > 0$ and $\mathbf{x} \in \mathbb{R}^m$, $\mathbf{prox}_{th}(\mathbf{x}) \triangleq \arg\min_{\mathbf{z} \in \mathbf{dom}\, h} h(\mathbf{z}) + \|\mathbf{x} - \mathbf{z}\|^2/(2t)$. Finally, all the sections and lemmas with indices beginning with 'S' will appear in the supplemental material.

2 ALGORITHMS

The pseudo-code of our algorithm is shown in Algorithm 1. Algorithm 1 includes six sequences of iterates, namely $\{\mathbf{x}^k\}_{k\in\mathbb{Z}^+}$, $\{\widetilde{\mathbf{x}}^k\}_{k\in\mathbb{Z}^+}$, $\{\overline{\mathbf{x}}^k\}_{k\in\mathbb{Z}^+}$, $\{\mathbf{y}^k\}_{k\in\mathbb{Z}^+}$, $\{\overline{\mathbf{y}}^k\}_{k\in\mathbb{Z}^+}$ and $\{\mathbf{z}^k\}_{k\in\mathbb{Z}^+}$. Among them, $\{\mathbf{x}^k\}_{k\in\mathbb{Z}^+}$ and $\{\mathbf{y}^k\}_{k\in\mathbb{Z}^+}$ are the primal and dual iterates respectively, and $\{\overline{\mathbf{x}}^k\}_{k\in\mathbb{Z}^+}$ and $\{\overline{\mathbf{y}}^k\}_{k\in\mathbb{Z}^+}$ are their weighted averages (see (8) and (9)). At each iteration $k \in \mathbb{Z}^+$, we first construct an interpolated point $\widetilde{\mathbf{x}}^k$ between $\mathbf{x}^k$ and $\overline{\mathbf{x}}^k$, and then obtain a stochastic gradient of f at $\widetilde{\mathbf{x}}^k$ from $\mathsf{SFO}(f, \sigma)$, denoted by $\mathbf{v}^k$. After that, we perform dual ascent, primal descent and extrapolation steps in (5), (6) and (7) respectively. Finally, we obtain the weighted averages $\overline{\mathbf{x}}^{k+1}$ and $\overline{\mathbf{y}}^{k+1}$.

We then choose the input sequences $\{\beta_k\}_{k\in\mathbb{Z}^+}$, $\{\alpha_k\}_{k\in\mathbb{Z}^+}$, $\{\tau_k\}_{k\in\mathbb{Z}^+}$ and $\{\theta_k\}_{k\in\mathbb{Z}^+}$ as

$$\theta_k = \frac{k+1}{k+2},\ \beta_k = \frac{(k+1)(k+4)}{2(k+2)},\ \alpha_k = \rho'/B \quad (11)$$

Algorithm 1 Optimal Stochastic Primal-Dual Algorithm for TCMP

Input: Interpolation sequence $\{\beta_k\}_{k\in\mathbb{Z}^+}$, dual stepsizes $\{\alpha_k\}_{k\in\mathbb{Z}^+}$, primal stepsizes $\{\tau_k\}_{k\in\mathbb{Z}^+}$, extrapolation sequence $\{\theta_k\}_{k\in\mathbb{Z}^+}$
Initialize: $\mathbf{x}^0 \in \mathbf{dom}\, g$, $\mathbf{y}^0 \in \mathbf{dom}\, h^*$, $\overline{\mathbf{x}}^0 = \mathbf{x}^0$, $\overline{\mathbf{y}}^0 = \mathbf{y}^0$, $\mathbf{z}^0 = \mathbf{x}^0$, $k = 0$
Repeat (until a convergence criterion is met)

$$\widetilde{\mathbf{x}}^k := \beta_k^{-1}\mathbf{x}^k + (1-\beta_k^{-1})\overline{\mathbf{x}}^k \quad (4)$$

Sample $\boldsymbol{\xi}^k \sim \nu$ and $\mathbf{v}^k \triangleq \nabla_{\mathbf{x}} F(\mathbf{x}, \boldsymbol{\xi}^k)|_{\mathbf{x}=\widetilde{\mathbf{x}}^k}$

$$\mathbf{y}^{k+1} := \mathbf{prox}_{\alpha_k h^*}(\mathbf{y}^k + \alpha_k \mathbf{A}\mathbf{z}^k) \quad (5)$$

$$\mathbf{x}^{k+1} := \mathbf{prox}_{\tau_k g}(\mathbf{x}^k - \tau_k(\mathbf{A}^T\mathbf{y}^{k+1} + \mathbf{v}^k)) \quad (6)$$

$$\mathbf{z}^{k+1} := \mathbf{x}^{k+1} + \theta_{k+1}(\mathbf{x}^{k+1} - \mathbf{x}^k) \quad (7)$$

$$\overline{\mathbf{x}}^{k+1} := \beta_k^{-1}\mathbf{x}^{k+1} + (1-\beta_k^{-1})\overline{\mathbf{x}}^k \quad (8)$$

$$\overline{\mathbf{y}}^{k+1} := \beta_k^{-1}\mathbf{y}^{k+1} + (1-\beta_k^{-1})\overline{\mathbf{y}}^k \quad (9)$$

$$k := k+1 \quad (10)$$

Output: $(\overline{\mathbf{x}}^k, \overline{\mathbf{y}}^k)$

$$\gamma_k = k+1,\ \tau_k^{-1} = \frac{4L}{k+2} + 2\rho' B + \rho\sigma\sqrt{k+2} \quad (12)$$

for any $k \in \mathbb{Z}^+$, where $\rho, \rho' > 0$ are constants (independent of k). Note that the convergence rate of Algorithm 1 matches the lower bound (3) for any values of ρ and ρ'. See Section 3 for details.

Remark 1. We can easily extend Algorithm 1 to the case where $\mathbf{x}$ in Problems (1) or (2) is minimized over a closed convex set $\mathcal{X}$. Indeed, we only need to replace step (6) with

$$\mathbf{x}^{k+1} := \arg\min_{\mathbf{x}\in\mathcal{X}} \|\mathbf{x} - \mathbf{x}^k\|^2/(2\tau_k) + \langle \mathbf{x}, \mathbf{A}^T\mathbf{y}^{k+1} + \mathbf{v}^k\rangle + g(\mathbf{x}). \quad (13)$$

The minimization problem in (13) admits closed-form solutions in many scenarios [43].

Remark 2 (**Innovations and Technical Difficulties**). Although steps (5), (6) and (7) also appear in [36, Algorithm 1], in this work, we add in three important steps, i.e., the interpolation step (4) and the primal and dual averaging steps (8) and (9). Although these steps seem natural and simple, they require *highly nontrivial* choices of the *input sequences*, including $\{\beta_k\}_{k\in\mathbb{Z}^+}$, $\{\alpha_k\}_{k\in\mathbb{Z}^+}$, $\{\tau_k\}_{k\in\mathbb{Z}^+}$ and $\{\theta_k\}_{k\in\mathbb{Z}^+}$, in (11) and (12). Indeed, it is these judicious choices of sequences that allow the convergence rate of Algorithm 1 to match the lower bound in (3). In fact, one can observe *significant differences* between these choices and those in [36, Section 2.3]. In addition, the sequences in (11) and (12) require *no prior knowledge* of the total number of iterations K—this is again in stark contrast to those in [36, Section 2.3] and being much

Algorithm 2 Optimal Stochastic Primal-Dual Algorithm for MCMP

Input: Interpolation sequence $\{\beta_k\}_{k\in\mathbb{Z}^+}$, dual stepsizes $\{\alpha_k\}_{k\in\mathbb{Z}^+}$, primal stepsizes $\{\tau_k\}_{k\in\mathbb{Z}^+}$, extrapolation sequence $\{\theta_k\}_{k\in\mathbb{Z}^+}$
Initialize: $\mathbf{x}^0 \in \mathbf{dom}\, g$, $\overline{\mathbf{x}}^0 = \mathbf{x}^0$, $\mathbf{z}^0 = \mathbf{x}^0$, $(\mathbf{y}_1^0, \ldots, \mathbf{y}_p^0) \in \prod_{i=1}^p \mathbf{dom}\, h_i^*$, $(\overline{\mathbf{y}}_1^0, \ldots, \overline{\mathbf{y}}_p^0) = (\mathbf{y}_1^0, \ldots, \mathbf{y}_p^0)$, $k = 0$
Repeat (until a convergence criterion is met)

$\mathbf{y}_i^{k+1} := \mathbf{prox}_{\alpha_k h_i^*}(\mathbf{y}_i^k + \alpha_k \mathbf{A}_i \mathbf{z}^k),\ \forall\, i \in [p]$

$\overline{\mathbf{y}}_i^{k+1} := \beta_k^{-1}\mathbf{y}_i^{k+1} + (1-\beta_k^{-1})\overline{\mathbf{y}}_i^k,\ \forall\, i \in [p]$

$\widetilde{\mathbf{x}}^k := \beta_k^{-1}\mathbf{x}^k + (1-\beta_k^{-1})\overline{\mathbf{x}}^k$

Sample $\boldsymbol{\xi}^k \sim \nu$ and $\mathbf{v}^k \triangleq \nabla_{\mathbf{x}} F(\mathbf{x}, \boldsymbol{\xi}^k)|_{\mathbf{x}=\widetilde{\mathbf{x}}^k}$

$\mathbf{x}^{k+1} := \mathbf{prox}_{\tau_k g}(\mathbf{x}^k - \tau_k(\sum_{i=1}^p \mathbf{A}_i^T \mathbf{y}_i^{k+1} + \mathbf{v}^k))$

$\mathbf{z}^{k+1} := \mathbf{x}^{k+1} + \theta_{k+1}(\mathbf{x}^{k+1} - \mathbf{x}^k)$

$\overline{\mathbf{x}}^{k+1} := \beta_k^{-1}\mathbf{x}^{k+1} + (1-\beta_k^{-1})\overline{\mathbf{x}}^k,\ k := k+1$

Output: $(\overline{\mathbf{x}}^k, \overline{\mathbf{y}}_1^k, \ldots, \overline{\mathbf{y}}_p^k)$

more practical. Consequently, these algorithmic innovations require much more *novel* and *technical* analysis techniques (detailed in Section 3).

2.1 Extension to Multiple Nonsmooth Terms

Algorithm 1 can be extended to handle the multi-composite convex minimization problem (MCMP)

$$\min_{\mathbf{x}\in\mathbb{R}^d}\ f(\mathbf{x}) + g(\mathbf{x}) + \textstyle\sum_{i=1}^p h_i(\mathbf{A}_i\mathbf{x}), \tag{14}$$

where $p \in \mathbb{N}$ and for each $i \in [p]$, the linear operator $\mathbf{A}_i : \mathbb{R}^d \to \mathbb{R}^{m_i}$ has operator norm $B_i > 0$ and $h_i : \mathbb{R}^{m_i} \to \mathbb{R}$ is CCP with a "simple" proximal operator. Our approach is to use the product-space technique (see e.g., [9, Section 5]). Specifically, define $\mathbb{R}^{\mathbf{m}} \triangleq \prod_{i=1}^p \mathbb{R}^{m_i}$, $\widehat{\mathbf{A}} : \mathbf{x} \mapsto (\mathbf{A}_1\mathbf{x}, \ldots, \mathbf{A}_p\mathbf{x})$ and $H : \widehat{\mathbf{y}} \mapsto \sum_{i=1}^p h_i(\mathbf{y}_i)$, where $\widehat{\mathbf{y}} \triangleq (\mathbf{y}_1, \ldots, \mathbf{y}_p) \in \mathbb{R}^{\mathbf{m}}$. Then (14) can be rewritten in the three-composite form as

$$\min_{\mathbf{x}\in\mathbb{R}^d} f(\mathbf{x}) + g(\mathbf{x}) + H(\widehat{\mathbf{A}}\mathbf{x}). \tag{15}$$

By noting that $H^*(\widehat{\mathbf{y}}) = \sum_{i=1}^p h_i^*(\mathbf{y}_i)$, the saddle-point form of (15) can be written as

$$\min_{\mathbf{x}\in\mathbb{R}^d} \max_{(\mathbf{y}_1,\ldots,\mathbf{y}_p)\in\mathbb{R}^{\mathbf{m}}} [\widehat{S}(\mathbf{x}, \mathbf{y}_1, \ldots, \mathbf{y}_p) \triangleq f(\mathbf{x}) + g(\mathbf{x}) + \textstyle\sum_{i=1}^p \langle \mathbf{A}_i\mathbf{x}, \mathbf{y}_i\rangle - \sum_{i=1}^p h_i^*(\mathbf{y}_i)]. \tag{16}$$

Based on (16) and that $\widehat{\mathbf{A}}^T\widehat{\mathbf{y}} = \sum_{i=1}^p \mathbf{A}_i^T\mathbf{y}_i$ and $\mathbf{prox}_{\alpha H}(\widehat{\mathbf{y}}) = (\mathbf{prox}_{\alpha h_1}(\mathbf{y}_1), \ldots, \mathbf{prox}_{\alpha h_p}(\mathbf{y}_p))$ for any $\alpha > 0$, we can derive a parallelizable algorithm for (14) based on Algorithm 1. The pseudo-code is shown in Algorithm 2. Note that the choices of the input sequences in Algorithm 2 directly follow (11) and (12), except that in this case, we replace B with $|||\widehat{\mathbf{A}}||| = (\sum_{i=1}^p B_i^2)^{1/2}$.

Remark 3. We can obtain $\mathbf{prox}_{\alpha_k h^*}$ in (5) from $\mathbf{prox}_{h/\alpha_k}$ via Moreau's identity, i.e.,

$$\mathbf{prox}_{th^*}(\mathbf{x}) = \mathbf{x} - t\mathbf{prox}_{h/t}(\mathbf{x}/t),\ \forall\, t > 0. \tag{17}$$

Remark 4. Based on the techniques in Section 2.1, one may intend to rewrite the nonsmooth function g as $g(\mathbf{x}) = \sup_{\mathbf{y}_0\in\mathbb{R}^d}\langle \mathbf{y}_0, \mathbf{x}\rangle - g^*(\mathbf{y}_0)$ and apply Algorithm 2 to the new problem. However, this will introduce an additional variable $\mathbf{y}_0$ (with the same dimension as $\mathbf{x}$) and hence *increase the memory requirement* of Algorithm 2. When $\mathbf{x}$ is high-dimensional, e.g., a positive semi-definite matrix, this memory increase is *significant*. Thus, when g is not coupled with a linear operator, we prefer to perform the proximal step on the primal side (see e.g., [13, 14]).

3 CONVERGENCE ANALYSIS

Preliminaries. Let the sequence of random vectors $\{\boldsymbol{\xi}^k\}_{k\in\mathbb{Z}^+}$ be given in Algorithm 1 and denote the probability space on which it is defined by $(\Omega, \mathcal{G}, \Pr)$. For any $k \in \mathbb{N}$, define $\Xi_k \triangleq \{\boldsymbol{\xi}^i\}_{i=0}^{k-1}$. Accordingly, define a filtration $\{\mathcal{F}_k\}_{k\in\mathbb{Z}^+}$ such that $\mathcal{F}_0 \triangleq \{\emptyset, \Omega\}$ and $\mathcal{F}_k$ is the σ-field generated by Ξ_k. Define $D_g \triangleq \sup_{\mathbf{x},\mathbf{x}'\in\mathbf{dom}\, g} \|\mathbf{x} - \mathbf{x}'\|$ and D_{h^*} in a similar way. Based on $\{\theta_k\}_{k\in\mathbb{Z}^+}$, we define an auxiliary sequence $\{\gamma_k\}_{k\in\mathbb{Z}^+}$ such that $\gamma_k = \prod_{i=0}^k \theta_i^{-1}$ for any $k \in \mathbb{Z}^+$. Finally, define the *primal-dual gap*

$$G(\mathbf{x}, \mathbf{y}) \triangleq \sup_{\mathbf{y}'\in\mathbf{dom}\, h^*} S(\mathbf{x}, \mathbf{y}') - \inf_{\mathbf{x}'\in\mathbf{dom}\, g} S(\mathbf{x}', \mathbf{y}). \tag{18}$$

Assumption 1. *For any $k \in \mathbb{Z}^+$ and $\varsigma \in \mathbb{R}$, the stochastic noise $\boldsymbol{\varepsilon}^k \triangleq \mathbf{v}^k - \nabla f(\widetilde{\mathbf{x}}^k)$ satisfies*

(A1) $\mathbb{E}_{\boldsymbol{\xi}^k}\left[\boldsymbol{\varepsilon}^k \,|\, \mathcal{F}_k\right] = 0$ *almost surely (a.s.)*

(A2) $\mathbb{E}_{\boldsymbol{\xi}^k}\left[\|\boldsymbol{\varepsilon}^k\|^2 \,\big|\, \mathcal{F}_k\right] \le \sigma^2$ *a.s.*

(A3) $\mathbb{E}_{\boldsymbol{\xi}^k}\left[\exp\left\{\varsigma\|\boldsymbol{\varepsilon}^k\|^2/\sigma^2\right\} \,\big|\, \mathcal{F}_k\right] \le \exp\{\varsigma^2 + \varsigma\}$ *a.s.*

Remark 5. Two remarks are in order. First, by Jensen's inequality, (A3) implies (A2). These two assumptions will be used in proving different convergence results below. Specifically, for convergence in expectation, (A2) is sufficient. To show large-deviation-type convergence results, we need (A3) instead, which indicates that the random variable $(\|\boldsymbol{\varepsilon}^k\|^2/\sigma^2 - 1)$ is sub-Gaussian conditioned on $\mathcal{F}_k$. Second, by setting $\varsigma = 1$ in (A3), we (essentially) recover the classical assumption in the literature, e.g., [22, Assumption A2]. We impose this stronger assumption to obtain $O(\sqrt{\log(1/\delta)})$ dependence on the probability of failure δ (see Remark 7).

Main Results. We establish convergence results for both Algorithms 1 and 2 when the domains of all the

nonsmooth component functions are bounded. This follows the convention of most works in the literature on primal-dual methods. However, note that in the unbounded case, Algorithms 1 and 2 still perform well numerically (see Section 5). The proofs of all the results in this section are deferred to Sections S-2 and S-3 in the supplemental material.

To start with, we first establish convergence results for all the input sequences (including $\{\beta_k\}_{k\in\mathbb{Z}^+}$, $\{\alpha_k\}_{k\in\mathbb{Z}^+}$, $\{\tau_k\}_{k\in\mathbb{Z}^+}$ and $\{\theta_k\}_{k\in\mathbb{Z}^+}$) that satisfy certain conditions.

Proposition 1. *Let* $\mathbf{dom}\, g$ *be compact and* $\mathbf{dom}\, h^*$ *be bounded. In Algorithm 1, let* $\beta_0 = 1$,

$$\beta_{k-1}\theta_k + 1 = \beta_k, \forall\, k \in \mathbb{Z}^+, \tag{19}$$

$$0 < \theta_k \leq \min\{\tau_{k-1}/\tau_k, \alpha_{k-1}/\alpha_k\}, \forall\, k \in \mathbb{N}, \tag{20}$$

$$B^2\alpha_{k-1} + L/\beta_{k-1} \leq (1-\zeta)/\tau_{k-1}, \forall\, k \in \mathbb{N}, \tag{21}$$

for some $\zeta \in (0,1)$. *Define* $\Gamma_K \triangleq \sum_{k=0}^{K-1}\gamma_k\tau_k$ *and* $\Gamma'_K \triangleq (\sum_{k=0}^{K-1}\gamma_k^2)^{1/2}$. *If (A1) and (A2) hold, then*

$$\mathbb{E}_{\Xi_K}\left[G(\overline{\mathbf{x}}^K, \overline{\mathbf{y}}^K)\right] \leq \frac{D_g^2}{\beta_{K-1}\tau_{K-1}} + \frac{D_{h^*}^2}{2\beta_{K-1}\alpha_{K-1}} + \frac{(1+\zeta)\Gamma_K}{2\zeta\beta_{K-1}\gamma_{K-1}}\sigma^2, \ \forall\, K \in \mathbb{N}. \tag{22}$$

Also, if (A1) and (A3) hold, then for any $\delta \in (0,1)$,

$$G(\overline{\mathbf{x}}^K, \overline{\mathbf{y}}^K) \leq \frac{1}{\beta_{K-1}}\left\{\frac{4\sqrt{\log(2/\delta)}D_g}{\gamma_{K-1}}\Gamma'_K\sigma + \frac{D_g^2}{\tau_{K-1}} + \frac{D_{h^*}^2}{2\alpha_{K-1}} + \frac{1+2\sqrt{\log(2/\delta)}}{2\zeta\gamma_{K-1}(1+\zeta)^{-1}}\Gamma_K\sigma^2\right\} \tag{23}$$

with probability (w.p.) at least $1-\delta$.

Remark 6. Note that the large-deviation-type result in (23) cannot be obtained by a straightforward application of Markov's inequality to (22), otherwise the dependence of (23) on δ will be $O(1/\delta)$, instead of the much improved $O(\sqrt{\log(1/\delta)})$. Rather, it requires a finer analysis involving Azuma-type martingale concentration results [44] and Assumption (A3). To be specific, in our analysis (see Section S-4), the martingale difference sequence that we work with is $\{\gamma_k\langle\boldsymbol{\varepsilon}^k, \mathbf{x}_\natural^k - \mathbf{x}^k\rangle\}_{k\in\mathbb{Z}^+}$, where $\{\mathbf{x}_\natural^k\}_{k\in\mathbb{Z}^+}$ is an auxiliary sequence defined recursively as $\mathbf{x}_\natural^0 \triangleq \mathbf{x}^0$ and for any $k\in\mathbb{Z}^+$, $\mathbf{x}_\natural^{k+1} \triangleq \Pi_{\mathbf{dom}\, g}[\mathbf{x}_\natural^k + \tau_k\boldsymbol{\varepsilon}^k]$. (Note that $\Pi_{\mathbf{dom}\, g}$ denotes the Euclidean projection onto $\mathbf{dom}\, g$.) The purpose of defining $\{\mathbf{x}_\natural^k\}_{k\in\mathbb{Z}^+}$ in this manner is explained in Lemma S-1.

Remark 7. Note that all the previous large-deviation-type results for the stochastic subgradient methods, e.g., [22, Corollary 1], have $O(\log(1/\delta))$ dependence on δ. In contrast, our result in (23) has an improved dependence on δ, i.e., $O(\sqrt{\log(1/\delta)})$. This is partially due to the slightly strengthened Assumption (A3). See Remark 5 for details.

Next, we verify that the choices of the input sequences in (11) and (12) indeed satisfy the conditions (19) to (21) in Proposition 1. Moreover, these choices lead to the optimal convergence rate of $O(L/K^2 + B/K + \sigma/\sqrt{K})$.

Theorem 1. *Let* $\mathbf{dom}\, g$ *be compact and* $\mathbf{dom}\, h^*$ *be bounded. In Algorithm 1, choose* $\{\beta_k\}_{k\in\mathbb{Z}^+}$, $\{\alpha_k\}_{k\in\mathbb{Z}^+}$, $\{\tau_k\}_{k\in\mathbb{Z}^+}$ *and* $\{\theta_k\}_{k\in\mathbb{Z}^+}$ *as in (11) and (12). As a result, they satisfy conditions (19) to (21). Consequently, if (A1) and (A2) hold, then for any* $K \in \mathbb{N}$,

$$\mathbb{E}_{\Xi_K}\left[G(\overline{\mathbf{x}}^K, \overline{\mathbf{y}}^K)\right] \leq \frac{8L}{K(K+3)}D_g^2 + \frac{4B}{K}\left(\rho' D_g^2 + \frac{D_{h^*}^2}{4\rho'}\right) + \frac{4\sigma}{\sqrt{K+3}}\left(\rho D_g^2 + \frac{2}{\rho}\right). \tag{24}$$

Also, if (A1) and (A3) hold, then for any $\delta \in (0,1)$,

$$G(\overline{\mathbf{x}}^K, \overline{\mathbf{y}}^K) \leq \frac{8L}{K(K+3)}D_g^2 + \frac{4B}{K}\left(\rho' D_g^2 + \frac{D_{h^*}^2}{4\rho'}\right) + \frac{16\sigma}{\sqrt{K+3}}\left(D_g + \frac{2}{\rho}\right)\sqrt{\log(2/\delta)} \tag{25}$$

w.p. at least $1-\delta$.

Remark 8. The large-deviation-type convergence result in (25) complements the in-expectation result in (24), in the sense that it indicates the behavior of *a single realization* of the random iterates $\{(\overline{\mathbf{x}}^k, \overline{\mathbf{y}}^k)\}_{k\in\mathbb{N}}$, rather than the ensemble average. Specifically, (25) shows that the convergence rate of any realization is rather insensitive to the error probability δ (indeed, $O(\sqrt{\log(1/\delta)})$ dependence). This result is important when Algorithm 1 is only run for few times, which often happens in practice.

Remark 9. If D_g and D_{h^*} are known or can be estimated reasonably well, then we can optimize the right-hand sides of (24) and (25) by choosing $\rho' = D_{h^*}/(2D_g)$ and $\rho = 2/D_g$. As a result,

$$\mathbb{E}_{\Xi_K}\left[G(\overline{\mathbf{x}}^K, \overline{\mathbf{y}}^K)\right] \leq \frac{8L}{K(K+3)}D_g^2 + \frac{4B}{K}D_g D_{h^*} + \frac{12\sigma}{\sqrt{K+3}}D_g \tag{26}$$

and for any $\delta \in (0,1)$, w.p. at least $1-\delta$,

$$G(\overline{\mathbf{x}}^K, \overline{\mathbf{y}}^K) \leq \frac{8L}{K(K+3)}D_g^2 + \frac{4B}{K}D_g D_{h^*} + \frac{32\sigma}{\sqrt{K+3}}\sqrt{\log(2/\delta)}D_g. \tag{27}$$

Remark 10. Using similar arguments, we can also prove the convergence results of Algorithm 2. We refer readers to Section S-5 for details.

Algorithm 3 Optimal SADMM for TCMP

Input: Interpolation sequence $\{r_k\}_{k\in\mathbb{Z}^+}$, stepsizes $\{\eta_k\}_{k\in\mathbb{Z}^+}$ and penalty parameter $\varrho>0$

Initialize: $\mathbf{u}^0 \in \mathbf{dom}\, g$, $\boldsymbol{\omega}^0 \in \mathbf{dom}\, h$, $\boldsymbol{\lambda}^0 \in \mathbb{R}^m$, $\overline{\mathbf{u}}^0 = \mathbf{u}^0$, $\overline{\boldsymbol{\omega}}^0 = \boldsymbol{\omega}^0$, $\overline{\boldsymbol{\lambda}}^0 = \boldsymbol{\lambda}^0$, $k = 0$

Repeat (until some convergence criterion is met)

$$\widetilde{\mathbf{u}}^k := r_k\mathbf{u}^k + (1-r_k)\overline{\mathbf{u}}^k \tag{29}$$

Sample $\tilde{\boldsymbol{\xi}}^k \sim \nu$ and define $\widetilde{\mathbf{v}}^k \triangleq \nabla_{\mathbf{u}} F(\mathbf{u}, \tilde{\boldsymbol{\xi}}^k)|_{\mathbf{u}=\widetilde{\mathbf{u}}^k}$

$$\boldsymbol{\omega}^{k+1} := \operatorname*{arg\,min}_{\boldsymbol{\omega}\in\mathbf{dom}\, h} L_k^{\varrho}(\mathbf{u}^k, \boldsymbol{\omega}, \boldsymbol{\lambda}^k) \tag{30}$$

$$\mathbf{u}^{k+1} := \operatorname*{arg\,min}_{\mathbf{u}\in\mathbf{dom}\, g} L_k^{\varrho}(\mathbf{u}, \boldsymbol{\omega}^{k+1}, \boldsymbol{\lambda}^k) \tag{31}$$

$$\boldsymbol{\lambda}^{k+1} := \boldsymbol{\lambda}^k - \varrho(\mathbf{A}\mathbf{u}^{k+1} - \boldsymbol{\omega}^{k+1}) \tag{32}$$

$$\overline{\boldsymbol{\omega}}^{k+1} := r_k\boldsymbol{\omega}^{k+1} + (1-r_k)\overline{\boldsymbol{\omega}}^k \tag{33}$$

$$\overline{\mathbf{u}}^{k+1} := r_k\mathbf{u}^{k+1} + (1-r_k)\overline{\mathbf{u}}^k \tag{34}$$

$$\overline{\boldsymbol{\lambda}}^{k+1} := r_k\boldsymbol{\lambda}^{k+1} + (1-r_k)\overline{\boldsymbol{\lambda}}^k \tag{35}$$

$$k := k+1 \tag{36}$$

Output: $(\overline{\mathbf{u}}^k, \overline{\boldsymbol{\omega}}^k, \overline{\boldsymbol{\lambda}}^k)$

4 CONNECTION TO SADMM

Note that (1) can be equivalently written as a linearly constrained problem

$$\min_{\mathbf{u}\in\mathbb{R}^d, \boldsymbol{\omega}\in\mathbb{R}^m} f(\mathbf{u}) + g(\mathbf{u}) + h(\boldsymbol{\omega}) \quad \text{s.t.} \quad \mathbf{A}\mathbf{u} = \boldsymbol{\omega}. \tag{28}$$

This equivalence naturally motivates us to use SADMM-type algorithms (e.g., [37, 38]) to solve Problem (1). Since most of the existing algorithms are only developed for the case where $g \equiv 0$, we first propose a *new* SADMM algorithm for solving (28). This method recovers a pre-conditioned variant of the optimal SADMM algorithm in [39] when $g \equiv 0$, thus being a useful generalization. Next, we establish the connection of this new algorithm to Algorithm 1. We conclude that this algorithm is a variant of Algorithm 1 with unit extrapolation parameter, i.e., $\theta_k = 1$, for any $k \in \mathbb{Z}^+$.

4.1 A New SADMM Algorithm

The pseudo-code of our new SADMM algorithm is shown in Algorithm 3. Similar to Algorithm 1, at iteration k, we first construct an interpolated point $\widetilde{\mathbf{u}}^k$ at which we obtain the stochastic gradient $\widetilde{\mathbf{v}}^k$ of f. Then, we update the primal iterates $(\boldsymbol{\omega}^k, \mathbf{u}^k)$ in a Gauss-Seidel manner in steps (30) and (31), where the augmented Lagrangian function

$$\begin{aligned} L_k^{\varrho}(\mathbf{u}, \boldsymbol{\omega}, \boldsymbol{\lambda}) \triangleq\ & f(\mathbf{u}^k) + \langle \mathbf{v}^k, \mathbf{u} - \mathbf{u}^k\rangle + g(\mathbf{u}) + h(\boldsymbol{\omega}) \\ & + \frac{r_k}{2\eta_k}\langle \mathbf{u} - \mathbf{u}^k, \mathbf{W}^k(\mathbf{u} - \mathbf{u}^k)\rangle \\ & - \langle \boldsymbol{\lambda}, \mathbf{A}\mathbf{u} - \boldsymbol{\omega}\rangle + (\varrho/2)\|\mathbf{A}\mathbf{u} - \boldsymbol{\omega}\|^2. \end{aligned} \tag{37}$$

In (37), $\mathbf{W}^k \triangleq a\mathbf{I} - (\eta_k/r_k)\varrho\mathbf{A}^T\mathbf{A}$. To ensure $\mathbf{W}^k \succeq 0$, we choose $a \geq \sup_{k\in\mathbb{N}}(\eta_k/r_k)\varrho B^2$. Then we update the dual iterate $\boldsymbol{\lambda}^k$ and finally, obtain the weighted average $(\overline{\boldsymbol{\omega}}^{k+1}, \overline{\mathbf{u}}^{k+1}, \overline{\boldsymbol{\lambda}}^{k+1})$.

For the choices of input parameters, we can choose any $\varrho > 0$. For any $k \in \mathbb{Z}^+$, we choose $r_k = 1/(k+1)$ and

$$\eta_k^{-1} = L + 2\sigma(k+1)^{3/2} + c\varrho B^2(k+1), \tag{38}$$

where $c > 0$ is a (tunable) constant. As a result, we can choose $a = \rho B^2/(3L^{1/3}\sigma^{2/3} + c\varrho B^2)$.

Remark 11. In the definition of L_k^{ϱ} in (37), we use pre-conditioning, i.e., replacing $\|\mathbf{u} - \mathbf{u}^k\|^2$ with the quadratic form $\langle \mathbf{u} - \mathbf{u}^k, \mathbf{W}^k(\mathbf{u} - \mathbf{u}^k)\rangle$, to ensure that the steps (30) and (31) in Algorithm 3 have closed-form solutions. See [34] for more details.

Remark 12. The update order of the primal iterates $(\boldsymbol{\omega}^k, \mathbf{u}^k)$ in Algorithm 3 is slightly different from that in most of the (stochastic) ADMM algorithms, wherein $\mathbf{u}^k$ is updated before $\boldsymbol{\omega}^k$ [45]. However, this difference of update order does not affect the convergence of Algorithm 3. For details, see [46].

Remark 13. Note that Algorithm 3 can be extended to handle the MCMP in (14) in a similar fashion as in Section 2.1.

4.2 Connection

To see the connection between Algorithm 3 and Algorithm 1, for any $k \in \mathbb{Z}^+$, define $\widetilde{\eta}_k \triangleq \eta_k/(ar_k)$,

$$\mathbf{z}_{\diamondsuit}^{k+1} \triangleq 2\mathbf{u}^{k+1} - \mathbf{u}^k, \tag{39}$$

$$\mathbf{y}_{\diamondsuit}^{k+1} \triangleq \mathbf{prox}_{\varrho h^*}(\varrho\mathbf{A}\mathbf{u}^k - \boldsymbol{\lambda}^k). \tag{40}$$

First, from (30), we have

$$\boldsymbol{\omega}^{k+1} = \mathbf{A}\mathbf{u}^k - (\boldsymbol{\lambda}^k + \mathbf{y}_{\diamondsuit}^{k+1})/\varrho. \tag{41}$$

Next, from (31), we have

$$\mathbf{u}^{k+1} = \mathbf{prox}_{\widetilde{\eta}_k g}(\mathbf{u}^k - \widetilde{\eta}_k(\mathbf{v}^k + \mathbf{A}^T\mathbf{y}_{\diamondsuit}^{k+1})). \tag{42}$$

(The detailed derivation steps for both (41) and (42) are deferred to Section S-6.) In addition, by substituting (41) into (32), we have

$$\boldsymbol{\lambda}^k = -\varrho\mathbf{A}(\mathbf{u}^k - \mathbf{u}^{k-1}) - \mathbf{y}_{\diamondsuit}^k. \tag{43}$$

We then substitute (43) into (40) to obtain

$$\mathbf{y}_{\diamondsuit}^{k+1} = \mathbf{prox}_{\varrho h^*}(\mathbf{y}_{\diamondsuit}^k + \varrho\mathbf{A}\mathbf{z}_{\diamondsuit}^k). \tag{44}$$

By letting $\mathbf{x}^k = \mathbf{u}^k$, $\mathbf{y}^k = \mathbf{y}_{\diamondsuit}^k$ and $\mathbf{z}^k = \mathbf{z}_{\diamondsuit}^k$, we observe that steps (44), (42) and (39) above recover steps (5), (6) and (7) in Algorithm 1 respectively. In this case, $\{\widetilde{\eta}_k\}_{k\in\mathbb{Z}^+}$ act as primal stepsizes and ϱ as the

Table 1: Algorithms under Comparison

	Abbrev.	Algorithms
Ours	OTPDHG	Algorithm 1 & Multi-Comp. Ext.
	OSADMM	Algorithm 3 & Multi-Comp. Ext.
Benchmarks	ESADMM	[40, Algorithm 1]
	SADMM	[38, Algorithm 2]
	ASG-PA	[26, Algorithm 1]
	TPDHG	[36, Algorithm 1]
	FOBOS	[23, Section 2]

dual stepsize. Additionally, the extrapolation parameter equals one. Also, by letting $\widetilde{\mathbf{x}}^k = \widetilde{\mathbf{u}}^k$, $\overline{\mathbf{x}}^k = \overline{\mathbf{u}}^k$ and $\overline{\mathbf{y}}^{k+1} = \varrho(\mathbf{A}\overline{\mathbf{u}}^k - \overline{\boldsymbol{\omega}}^{k+1}) - \overline{\boldsymbol{\lambda}}^k$, steps (4), (8) and (9) in Algorithm 1 can be recovered. The sequence $\{r_k\}_{k\in\mathbb{Z}^+}$ now corresponds to $\{\beta_k^{-1}\}_{k\in\mathbb{Z}^+}$ in Algorithm 1.

Finally, we compare the new primal stepsizes $\{\widetilde{\eta}_k\}_{k\in\mathbb{Z}^+}$ with the original ones in Algorithm 1, i.e., $\{\tau_k\}_{k\in\mathbb{Z}^+}$. Indeed, if we choose $\varrho = \rho'/B$, then these two sequences have the same scaling, i.e., $\Theta((L/k+B+\sigma\sqrt{k})^{-1})$. We note that the interpolation sequences $\{r_k\}_{k\in\mathbb{Z}^+}$ and $\{\beta_k^{-1}\}_{k\in\mathbb{Z}^+}$ have the same scaling as well, i.e., $\Theta(1/k)$. This indicates that Algorithm 3 is a variant of Algorithm 1 with unit extrapolation parameter.

5 NUMERICAL EXPERIMENTS

Applications and Datasets. We compared the numerical performance of our algorithms (i.e., Algorithms 1 and 3 and their multi-composite extensions) with *five* benchmark algorithms on three machine learning applications. These applications include graph-guided sparse logistic regression (GLR) [5], graph-guided fused lasso (GFL) [1] and sparse overlapped group lasso (OGL) [42]. For all the applications, the datasets we used were extracted from the LIBSVM [47] repository (and normalized). All the datasets share a common form $\{(\mathbf{a}_i, b_i)\}_{i=1}^n$, where $\{\mathbf{a}_i\}_{i=1}^n \subseteq \mathbb{R}^d$ are feature vectors and $\{b_i\}_{i=1}^n \subseteq \mathbb{R}$ are response variables. For convenience, we term one pass over n data samples as one *epoch*.

Algorithms. Our algorithms and the benchmark algorithms are listed in the Table 1, together with their abbreviations. (All the benchmark methods are described in Section 1.2.) Note that we did not compare our methods to the deterministic and variance-reduced randomized algorithms since these methods cannot solve Problem (1) in general, i.e., when the distribution ν in has infinite support.

Parameter Settings. For all the benchmark algorithms, we used the parameter settings suggested in the original works. For Algorithm 1 (OTPDHG), we set $\rho = 1\times10^{-3}$ and $\rho' = 1\times10^{-5}$ in (11) and (12) throughout all the experiments. For Algorithm 2 (OSADMM), we set the penalty parameter $\varrho = \rho'/B$ (as in Section 4) and $c = 6\times10^{-2}$ in (38).

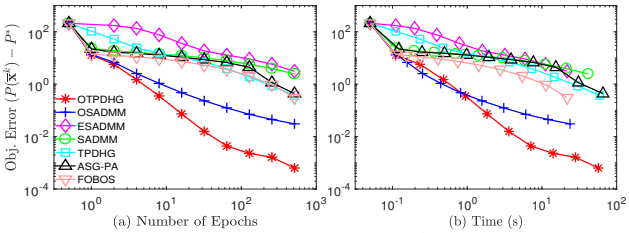

Figure 1: Plot of the obj. error $P_{\mathrm{GLR}}(\overline{\mathbf{x}}^k) - P^*_{\mathrm{GLR}}$ versus (a) number of epochs and (b) time (in seconds) on `a9a`.

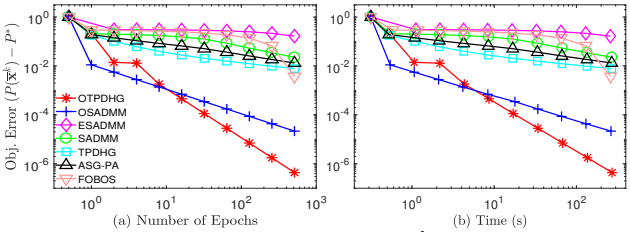

Figure 2: Plot of the obj. error $P_{\mathrm{GLR}}(\overline{\mathbf{x}}^k) - P^*_{\mathrm{GLR}}$ versus (a) number of epochs and (b) time (in seconds) on `covtype`.

Comparison Criterion. Denote $\{\overline{\mathbf{x}}^k\}_{k\in\mathbb{Z}^+}$ as the sequence generated by each algorithm and P^* as the optimal value of Problem (1). We estimated P^* via Algorithm 1 in [14], which is a deterministic algorithm for solving Problem (1). As a fair comparison of the (empirical) convergence rates of all the algorithms, we used the primal sub-optimality gap $P(\overline{\mathbf{x}}^k) - P^*$ as the criterion. We ran each (stochastic) algorithm ten times. Then we plotted the average realization of $P(\overline{\mathbf{x}}^k) - P^*$ versus the number of epochs, which reflects the number of queries to the oracle $\mathsf{SFO}(f, \sigma)$. We also plotted the average realization of $P(\overline{\mathbf{x}}^k) - P^*$ versus the actual running time. (All the algorithms were implemented in Matlab® R2016b on a machine with a 3.9 GHz processor and 8 GB RAM.)

5.1 Graph-Guided Sparse Logistic Regression

To formulate the GLR problem, for any $i \in [n]$, we first define the logistic loss function $\ell_i^{\mathrm{LR}} : \mathbb{R}^d \to \mathbb{R}$ as

$$\ell_i^{\mathrm{LR}}(\mathbf{x}) \triangleq \log(1 + \exp(-b_i \mathbf{a}_i^T \mathbf{x})), \ \forall\, \mathbf{x} \in \mathbb{R}^d. \tag{45}$$

Then we define the average loss function

$$\ell^{\mathrm{LR}}(\mathbf{x}) \triangleq (1/n)\textstyle\sum_{i=1}^n \ell_i^{\mathrm{LR}}(\mathbf{x}), \ \forall\, \mathbf{x} \in \mathbb{R}^d. \tag{46}$$

In this case, each entry of the decision vector $\mathbf{x}$ corresponds to a feature. The relations among these features can be represented using a matrix $\mathbf{F}$. (See [37] for details.) Based on $\mathbf{F}$, we then formulate GLR as

$$\min_{\mathbf{x}\in\mathbb{R}^d} \left[P_{\mathrm{GLR}}(\mathbf{x}) \triangleq \ell^{\mathrm{LR}}(\mathbf{x}) + \lambda_1 \left\|\mathbf{x}\right\|_1 + \lambda_2 \left\|\mathbf{F}\mathbf{x}\right\|_1 \right], \tag{47}$$

where $\lambda_1, \lambda_2 > 0$ are regularization parameters.

We set $\lambda_1 = \lambda_2 = 1/\sqrt{n}$ and generated the set of weighted edges $\mathcal{E}$ in a similar fashion as in [37, Sec-

tion 5.2]. At each iteration k, to obtain the stochastic gradient $\mathbf{v}^k$, we first uniformly randomly sampled a subset $\mathcal{B}_k \subseteq [n]$ (without replacement) such that $|\mathcal{B}_k| = \lfloor n/100 \rfloor$. Then we let

$$\mathbf{v}^k = (1/|\mathcal{B}_k|)\textstyle\sum_{i\in\mathcal{B}_k} \nabla \ell_i^{\rm LR}(\mathbf{x}^k). \tag{48}$$

Note that in this case, the noise variance σ^2 in (A2) can be easily estimated from [48, Lemma T-3]. We will also obtain $\mathbf{v}^k$ in the same way in Sections 5.2 and 5.3.

We tested all the algorithms on the `a9a` and `covtype` datasets. The results are shown in Figures 1 and 2 respectively. From both figures, we observe that OTPDHG and OSADMM consistently and significantly outperform the benchmark algorithms, in terms of both the number of epochs and running time. In particular, from Figure 2, we clearly observe that OTPDHG exhibits an $O(1/K^2)$ convergence rate. This indeed corroborates our theoretical results in Theorem 1. Although OSADMM is closely related to OTPDHG (see Section 4.2), the exact parameter settings therein are different from those in OTPDHG. This explains the differences between the numerical performances of OSADMM and OTPDHG.

5.2 Graph-Guided Fused Lasso

The GFL problem can be formulated similarly as GLR in Section 5.1. In this case we replace the logistic loss function $\ell^{\rm LR}$ in (47) by the least-square loss function

$$\ell^{\rm LS}(\mathbf{x}) \triangleq (1/n)\textstyle\sum_{i=1}^n (\mathbf{a}_i^T\mathbf{x} - b_i)^2/2,\ \forall\, \mathbf{x}\in\mathbb{R}^d.$$

As a result, we have

$$\min_{\mathbf{x}\in\mathbb{R}^d} \left[P_{\rm GFL}(\mathbf{x}) \triangleq \ell^{\rm LS}(\mathbf{x}) + \lambda_1 \|\mathbf{x}\|_1 + \lambda_2 \|\mathbf{F}\mathbf{x}\|_1 \right], \tag{49}$$

where the regularization parameters λ_1 and λ_2, and the matrix $\mathbf{F}$ are all the same as in Section 5.1.

We tested all the algorithms on both `cadata` and `YPM` (`YearPredictionMSD`) datasets. The results are shown in Figures 3 and 4 respectively. Our observations from these two figures are indeed consistent with those in Section 5.1. Specifically, our algorithms (OTPDHG and OSADMM) outperform the benchmark methods. Also, the performance of OSADMM is slightly inferior to that of OTPDHG. See Section 5.1 for explanations.

5.3 Sparse Overlapping Group Lasso

We finally consider a multi-composite stochastic optimization problem in the form of (14), where ν has finite support. Given a set of index groups $\{\mathcal{G}_i\}_{i=1}^p$ where each $\mathcal{G}_i \subseteq [d]$ and positive parameters $\{\lambda_i\}_{i=0}^p$, the OGL problem is

$$\min_{\mathbf{x}\in\mathbb{R}^d} \left[P_{\rm OGL}(\mathbf{x}) \triangleq \ell^{\rm LS}(\mathbf{x}) + \lambda_0\|\mathbf{x}\|_1 + \textstyle\sum_{i=1}^p \lambda_i \|\mathbf{x}_{\mathcal{G}_i}\| \right],$$

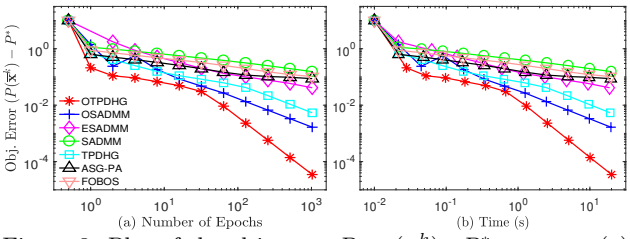

Figure 3: Plot of the obj. error $P_{\rm GFL}(\overline{\mathbf{x}}^k) - P^*_{\rm GFL}$ versus (a) number of epochs and (b) time (in seconds) on `cadata`.

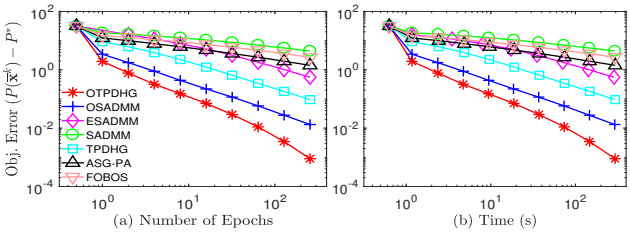

Figure 4: Plot of the obj. error $P_{\rm GFL}(\overline{\mathbf{x}}^k) - P^*_{\rm GFL}$ versus (a) number of epochs and (b) time (in seconds) on `YPM`.

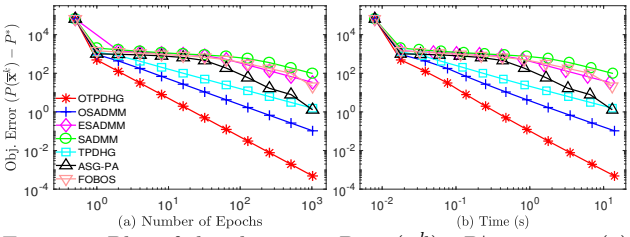

Figure 5: Plot of the obj. error $P_{\rm OGL}(\overline{\mathbf{x}}^k) - P^*_{\rm OGL}$ versus (a) number of epochs and (b) time (in seconds) on `cpusmall`.

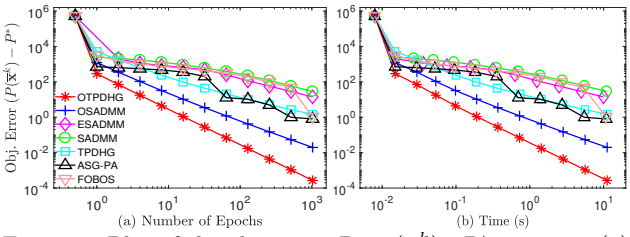

Figure 6: Plot of the obj. error $P_{\rm OGL}(\overline{\mathbf{x}}^k) - P^*_{\rm OGL}$ versus (a) number of epochs and (b) time (in seconds) on `abalone`.

where $\mathbf{x}_{\mathcal{G}_i}$ denotes the subvector of $\mathbf{x}$ indexed by $\mathcal{G}_i$. Since this problem is an instance of Problem (14), we can use the multi-composite extensions of OTPDHG and OSADMM to solve it (see Section 2.1 and Remark 13).

We generated the index groups $\{\mathcal{G}_i\}_{i=1}^p$ and the regularization parameters $\{\lambda_i\}_{i=0}^p$ in the same way as in [36, Section 5.4]. We tested all the algorithms on both `cpusmall` and `abalone` datasets. The results are shown in Figures 5 and 6 respectively. From both figures, we obtain the same conclusions as in Sections 5.1 and 5.2. Specifically, apart from the superior performance of OTPDHG and OSADMM, the $O(1/K^2)$ convergence rates of OTPDHG (for moderate K) are also evident.

References

[1] S. Kim, K.-A. Sohn, and E. P. Xing, "A multivariate regression approach to association analysis of a quantitative trait network," *Bioinform.*, vol. 25, no. 12, pp. 204–212, 2009.

[2] R. Tibshirani, M. Saunders, S. Rosset, J. Zhu, and K. Knight, "Sparsity and smoothness via the fused lasso," *J. R. Stat. Soc. Ser. B*, vol. 67, no. 1, pp. 91–108, 2005.

[3] T. T. Cai and W.-X. Zhou, "Matrix completion via max-norm constrained optimization," *Electron. J. Stat.*, vol. 10, no. 1, pp. 1493–1525, 2016.

[4] J. Brodie, I. Daubechiesa, C. D. Mol, D. Giannone, and I. Loris, "Sparse and stable markowitz portfolios," *Proc. Natl. Acad. Sci.*, vol. 106, no. 30, pp. 12267—-12272, 2009.

[5] X. Chen, Q. Lin, S. Kim, J. G. Carbonell, and E. P. Xing, "Smoothing proximal gradient method for general structured sparse regression," *Ann. Appl. Stat.*, vol. 6, no. 2, pp. 719–752, 2012.

[6] A. Shapiro and A. Philpott, "A tutorial on stochastic programming." http://stoprog.org/sites/default/files/SPTutorial/TutorialSP.pdf, 2007.

[7] A. Beck and M. Teboulle, "Fast gradient-based algorithms for constrained total variation image denoising and deblurring problems," *IEEE Trans. Image Process.*, vol. 18, no. 11, pp. 2419–2434, 2009.

[8] 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 Var. Anal.*, vol. 20, no. 2, pp. 307–330, 2012.

[9] L. Condat, "A primal–dual splitting method for convex optimization involving Lipschitzian, proximable and linear composite terms," *J. Optim. Theory Appl.*, vol. 158, pp. 460–479, Aug 2013.

[10] B. C. Vũ, "A splitting algorithm for dual monotone inclusions involving cocoercive operators," *Adv. Comput. Math.*, vol. 38, pp. 667–681, Apr 2013.

[11] R. I. Boţ, E. R. Csetnek, A. Heinrich, and C. Hendrich, "On the convergence rate improvement of a primal-dual splitting algorithm for solving monotone inclusion problems," *Math. Program.*, vol. 150, no. 2, pp. 251–279, 2015.

[12] N. He, A. Juditsky, and A. Nemirovski, "Mirror-Prox algorithm for multi-term composite minimization and semi-separable problems," *Comput. Optim. Appl.*, vol. 61, no. 2, pp. 275–319, 2015.

[13] D. Davis, "Convergence rate analysis of primal-dual splitting schemes," *SIAM J. Optim.*, vol. 25, no. 3, pp. 1912–1943, 2015.

[14] A. Chambolle and T. Pock, "On the ergodic convergence rates of a first-order primal–dual algorithm," *Math. Program.*, vol. 159, no. 1, pp. 253–287, 2016.

[15] Y. He and R. D. C. Monteiro, "An accelerated HPE-type algorithm for a class of composite convex-concave saddle-point problems," *SIAM J. Optim.*, vol. 26, no. 1, pp. 29–56, 2016.

[16] Q. V. Nguyen, O. Fercoq, and V. Cevher, "Smoothing technique for nonsmooth composite minimization with linear operator." arXiv:1706.05837, 2017.

[17] A. Alacaoglu, Q. T. Dinh, O. Fercoq, and V. Cevher, "Smooth primal-dual coordinate descent algorithms for nonsmooth convex optimization," in *Proc. NIPS*, 2017.

[18] R. T. Rockafellar, *Convex analysis*. Princeton University Press, 1970.

[19] C. Hu, W. Pan, and J. T. Kwok, "Accelerated gradient methods for stochastic optimization and online learning," in *Porc. NIPS*, (Vancouver, B.C., Canada), pp. 781–789, 2009.

[20] A. Nemirovski, A. Juditsky, G. Lan, and A. Shapiro, "Robust stochastic approximation approach to stochastic programming," *SIAM J. Optim.*, vol. 19, no. 4, pp. 1574–1609, 2009.

[21] J. Duchi, E. Hazan, and Y. Singer, "Adaptive subgradient methods for online learning and stochastic optimization," *J. Mach. Learn. Res.*, vol. 12, pp. 2121–2159, 2011.

[22] G. Lan, "An optimal method for stochastic composite optimization," *Math. Program.*, vol. 133, no. 1-2, pp. 365–397, 2012.

[23] J. Duchi and Y. Singer, "Efficient online and batch learning using forward backward splitting," *J. Mach. Learn. Res.*, vol. 10, pp. 2899–2934, 2009.

[24] S. Ghadimi and G. Lan, "Optimal stochastic approximation algorithms for strongly convex stochastic composite optimization I: A generic algorithmic framework," *SIAM J. Optim.*, vol. 22, no. 4, pp. 1469–1492, 2012.

[25] L. Xiao, "Dual averaging methods for regularized stochastic learning and online optimization," *J. Mach. Learn. Res.*, vol. 11, pp. 2543–2596, 2010.

[26] W. Zhong and J. Kwok, "Accelerated stochastic gradient method for composite regularization," in *Proc. AISTATS*, (Reykjavik, Iceland), pp. 1086–1094, 2014.

[27] Y. Yu, "Better approximation and faster algorithm using the proximal average," in *Proc. NIPS*, (Lake Tahoe, Nevada), pp. 458–466, 2013.

[28] A. Yurtsever, B. C. Vu, and V. Cevher, "Stochastic three-composite convex minimization," in *Proc. NIPS*, (Barcelona, Spain), pp. 4329–4337, 2016.

[29] D. Davis and W. Yin, "A three-operator splitting scheme and its optimization applications," *Set-Valued Var. Anal.*, 2017.

[30] Y. Nesterov, "Primal-dual subgradient methods for convex problems," *Math. Program.*, vol. 120, no. 1, pp. 221–259, 2009.

[31] Y. Chen, G. Lan, and Y. Ouyang, "Optimal primal-dual methods for a class of saddle point problems," *SIAM J. Optim.*, vol. 24, no. 4, pp. 1779–1814, 2014.

[32] L. Qiao, T. Lin, Y. Jiang, F. Yang, W. Liu, and X. Lu, "On stochastic primal-dual hybrid gradient approach for compositely regularized minimization," in *Proc. ECAI*, (Hague, Netherlands), pp. 167–174, 2016.

[33] L. Rosasco, S. Villa, and B. C. Vu, "A first-order stochastic primal-dual algorithm with correction step," *Numer. Funct. Anal. Optim.*, vol. 38, no. 5, pp. 602–626, 2017.

[34] E. Esser, X. Zhang, and T. F. Chan, "A general framework for a class of first order primal-dual algorithms for convex optimization in imaging science," *SIAM J. Imaging Sci.*, vol. 3, no. 4, pp. 1015–1046, 2010.

[35] A. Chambolle and T. Pock, "A first-order primal-dual algorithm for convex problems with applications to imaging," *J. Math. Imaging Vis.*, vol. 40, no. 1, pp. 120–145, 2011.

[36] R. Zhao and V. Cevher, "Stochastic three-composite convex minimization with a linear operator," in *Proc. AISTATS*, (Lanzarote, Spain), 2018.

[37] H. Ouyang, N. He, L. Tran, and A. Gray, "Stochastic alternating direction method of multipliers," in *Proc. ICML*, (Atlanta, USA), pp. 80–88, 2013.

[38] T. Suzuki, "Dual averaging and proximal gradient descent for online alternating direction multiplier method," in *Proc. ICML*, pp. 392–400, 2013.

[39] S. Azadi and S. Sra, "Towards an optimal stochastic alternating direction method of multipliers," in *Proc. ICML*, (Beijing, China), pp. 620–628, 2014.

[40] T. Lin, L. Qiao, T. Zhang, J. Feng, and B. Zhang, "Stochastic primal-dual proximal extragradient descent for compositely regularized optimization," *Neurocomput.*, vol. 273, pp. 516 – 525, 2018.

[41] L. Jacob, G. Obozinski, and J.-P. Vert, "Group lasso with overlap and graph lasso," in *Proc. ICML*, (Montreal, Quebec, Canada), pp. 433–440, 2009.

[42] N. Simon, J. Friedman, T. Hastie, and R. Tibshirani, "A sparse-group lasso," *J. Comput. Graph. Stat.*, vol. 22, pp. 231–245, 2013.

[43] Y. Yu, "On decomposing the proximal map," in *Proc. NIPS*, (Lake Tahoe, Nevada), pp. 91–99, 2013.

[44] K. Azuma, "Weighted sums of certain dependent random variables," *Tohoku Math. J.*, vol. 19, no. 3, pp. 357–367, 1967.

[45] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Found. Trends Mach. Learn.*, vol. 3, pp. 1–122, Jan. 2011.

[46] M. Yan and W. Yin, *Self Equivalence of the Alternating Direction Method of Multipliers*, pp. 165–194. Springer International Publishing, 2016.

[47] C.-C. Chang and C.-J. Lin, "LIBSVM: A library for support vector machines," *ACM Trans. Intell. Syst. Technol.*, vol. 2, pp. 27:1–27:27, 2011. Software available at http://www.csie.ntu.edu.tw/~cjlin/libsvm.

[48] R. Zhao, W. B. Haskell, and V. Y. F. Tan, "Stochastic L-BFGS: Improved convergence rates and practical acceleration strategies," *IEEE Trans. Signal Process.*, vol. 66, no. 5, pp. 1155–1169, 2018.