Three Operator Splitting with Subgradients, Stochastic Gradients, and Adaptive Learning Rates

Anonymous Author(s)
Affiliation
Address
email

Abstract

Three Operator Splitting (TOS) (Davis & Yin, 2017) can minimize the sum of three convex functions effectively when an efficient gradient oracle or proximal operator is available for each term. This requirement often fails in machine learning applications: *(i)* instead of full gradients only stochastic gradients may be available; and *(ii)* instead of proximal operators, using subgradients to handle complex penalty functions may be more efficient and realistic. Motivated by these concerns, we analyze three potentially valuable extensions of TOS. The first two permit using subgradients and stochastic gradients, and are shown to ensure a $\mathcal{O}(1/\sqrt{t})$ convergence rate. The third extension ADAPTOS endows TOS with adaptive step-sizes. For the important setting of optimizing a convex loss over the intersection of convex sets ADAPTOS attains universal convergence rates, i.e., the rate adapts to the *unknown* smoothness degree of the objective. We compare our proposed methods with competing methods on various applications.

1 Introduction

We study convex optimization problems of the form

$$\min_{x\in\mathbb{R}^n} \quad \phi(x) := f(x) + g(x) + h(x), \tag{1}$$

where $f : \mathbb{R}^n \to \mathbb{R}$ and $g, h : \mathbb{R}^n \to \mathbb{R} \cap \{+\infty\}$ are proper, lower semicontinuous and convex functions. Importantly, this template captures constrained problems via indicator functions.

Problem (1) is motivated by a number of applications in machine learning, statistics, and signal processing, where the three functions comprising the objective ϕ model data fitting, structural priors, or decision constraints. Examples include overlapping group lasso (Yuan et al., 2011), isotonic regression (Tibshirani et al., 2011), dispersive sparsity (El Halabi & Cevher, 2015), graph transduction (Shivanna et al., 2015), learning with correlation matrices (Higham & Strabić, 2016), and multidimensional total variation denoising (Barbero & Sra, 2018).

An important technique for addressing composite problems of the form (1) is *operator splitting* (Bauschke et al., 2011). However, a naive use of the basic proximal-(sub)gradient method may be unsuitable, since it requires access to the prox-operator of $g + h$, computing which may be vastly more expensive than individual prox-operators of g and h. An elegant, recent method, Three Operator Splitting method (TOS, Davis & Yin (2017), see Algorithm 1) offers a practical choice for solving Problem (1) when f is L-smooth, and prox-operators of both g and h are available. Importantly, at each iteration, TOS evaluates the gradient of f and the proximal operators of g and h only once.

Unfortunately, TOS is not readily applicable to many optimization problems that arise in machine learning. Most important among those are problems where only access to stochastic gradients is

Submitted to 35th Conference on Neural Information Processing Systems (NeurIPS 2021). Do not distribute.

feasible, e.g., when performing large-scale empirical risk minimization and online learning. Moreover, prox-operators for some complex penalty functions are computationally expensive and it may be more efficient to instead use subgradients. For example, proximal operator for the maximum eigenvalue function that appears in dual-form semidefinite programs (e.g., see Section 6.1 in (Ding et al., 2019)) may require computing a full eigendecomposition (with a cubic arithmetic cost). In contrast, we can form a subgradient by computing only the top eigenvector via power method or Lanczos algorithm.

Contributions. With the above motivation, this paper contributes three key extensions of TOS. We tackle nonsmoothness in Section 3 and stochasticity in Section 4. These two extensions enable us to use subgradients and stochastic gradients in TOS (see Section 2 for a comparison with related work), and satisfy a $\mathcal{O}(1/\sqrt{T})$ convergence rate in function value after T iterations. The third main contribution is AdapTos in Section 5. This extension provides an adaptive step-size rule for TOS in the spirit of AdaGrad (Duchi et al., 2011; Levy, 2017). Notably, for optimizing a convex loss over the intersection of two convex sets, an important subclass of Problem (1), AdapTos ensures universal convergence rates. That is, AdapTos implicitly adapts to the *unknown* smoothness of the problem, and ensures a $\tilde{\mathcal{O}}(1/\sqrt{T})$ rate when the problem is nonsmooth or stochastic, with the rate improving to $\tilde{\mathcal{O}}(1/T)$ if the problem is smooth and a solution lies in the relative interior of the feasible set.

In Section 6, we discuss empirical performance of our methods by comparing them against present established methods on various benchmark problems from COPT Library (Pedregosa et al., 2020) including the overlapping group lasso, total variation deblurring, and sparse and low-rank matrix recovery. We also test our methods on nonconvex optimization to train a neural network model.

Notation. We denote a solution of Problem (1) by $x_\star$ and $\phi_\star = \phi(x_\star)$. The distance between a point $x \in \mathbb{R}^n$ and a set $\mathcal{G} \subseteq \mathbb{R}^n$ is $\operatorname{dist}(x, \mathcal{G}) := \inf_{y\in\mathcal{G}} \|x - y\|$; the projection of x onto $\mathcal{G}$ is given by $\operatorname{proj}_{\mathcal{G}}(x) := \arg\min_{y\in\mathcal{G}} \|x - y\|$. The proximal operator (or prox-operator) of a function $g : \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}$ is defined by $\operatorname{prox}_g(x) := \arg\min_{y\in\mathbb{R}^n}\{g(y) + \frac{1}{2}\|x - y\|^2\}$. The indicator function of a set $\mathcal{G}$ gives 0 for any $x \in \mathcal{G}$ and $+\infty$ otherwise. Recall that the prox-operator for the indicator function is the projection onto the corresponding set.

2 Background and related work

TOS, proposed recently by Davis & Yin (2017), can be seen as a generic extension of various operator splitting schemes, including the forward-backward splitting, Douglas-Rachford splitting, forward-Douglas-Rachford splitting (Briceño-Arias, 2015), and the generalized forward-backward splitting (Raguet et al., 2013). It covers these aforementioned approaches as special instances when the terms f, g and h in Problem (1) are chosen appropriately. Convergence of TOS is well studied when f has Lipschitz continuous gradients. It ensures $\mathcal{O}(1/t)$ convergence rate in this setting, see (Davis & Yin, 2017) and (Pedregosa, 2016) for the technical details.

Other related methods that can be used for Problem (1) with smooth f are the primal-dual hybrid gradient (PDHG) method (Condat, 2013; Vũ, 2013) and the primal-dual three operator splitting method (Yan, 2018). These methods can handle a slightly more general template where g or h is composed with a linear map, however, they require f to be smooth. The convergence rate of PDHG is analyzed in (Chambolle & Pock, 2016).

Nonsmooth setting. We are unaware of any prior result that permits using subgradients in TOS (or in related methods that can use prox-operator of g and h separately) for Problem (1). The closest approach is the proximal subgradient method which applies if h is missing, and it is covered by our nonsmooth TOS as a special case.

Stochastic setting. There are multiple attempts to devise a stochastic TOS in the literature. Yurtsever et al. (2016) studied Problem (1) under the assumption that f is smooth and strongly convex, and an unbiased gradient estimator with bounded variance is available. Their stochastic TOS has a guaranteed $\mathcal{O}(1/t)$ convergence rate. In (Cevher et al., 2018), they relaxed the strong convexity assumption but assume that the gradient estimator's variance is summable. They show asymptotic convergence with no guarantees on the rate. Very recently, Pedregosa et al. (2019) proposed a stochastic variance-reduced TOS and analyzed its non-asymptotic convergence guarantees. Their method gets $\mathcal{O}(1/t)$ convergence rate for arbitrary smooth convex f. The rate becomes linear if f is smooth and strongly convex and g (or h) is smooth. None of these prior works cover the broad template we consider: f is smooth or Lipschitz continuous and the stochastic first-order oracle has

Algorithm 1 Three Operator Splitting (TOS)

Input: Initial point $y_0 \in \mathbb{R}^n$, step-size sequence $\{\gamma_t\}_{t=0}^T$
for $t = 0, 1, 2, \ldots, T$ **do**
  $z_t = \operatorname{prox}_{\gamma_t g}(y_t)$
  Choose an update direction $u_t \in \mathbb{R}^n$
  $x_t = \operatorname{prox}_{\gamma_t h}(2z_t - y_t - \gamma_t u_t)$
  $y_{t+1} = y_t - z_t + x_t$
end for
Return: Ergodic sequence $\bar{x}_t$ and $\bar{z}_t$ defined in (3).

bounded variance. To our knowledge, our paper gives the first analysis for stochastic TOS without strong convexity assumption or variance reduction.

Other related methods include stochastic PDHG in (Zhao & Cevher, 2018) and a stochastic primal-dual method in (Zhao et al., 2019). The latter can be viewed as an extension of stochastic ADMM (Ouyang et al., 2013; Azadi & Sra, 2014) from the sum of two terms to three terms in the objective.

Adaptive step-sizes. The standard writings of TOS and PDHG require the knowledge of the smoothness constant of f for the step-size. Backtracking line-search strategies are proposed to find a suitable step-size when the smoothness constant is unknown, for PDHG in (Malitsky & Pock, 2018) and for TOS in (Pedregosa & Gidel, 2018). A heuristic line-search strategy was also introduced in the original preprint (Davis & Yin, 2015), but it is reported as nonconvergent in (Pedregosa & Gidel, 2018).

The motivation and goals of these line-search strategies significantly differ from our adaptive learning rate. Importantly, these methods work only when f is smooth; they require extra function evaluations, and are thus not suitable for stochastic optimization. And their goal is to estimate the *smoothness constant*. In contrast, our method does not require function evaluations. It can be used for smooth, nonsmooth, or stochastic settings. Most importantly, it can adapt to the unknown *smoothness degree*.

At the heart of our method lie adaptive online learning algorithms (Duchi et al., 2011; Rakhlin & Sridharan, 2013) together with online to offline conversion techniques (Levy, 2017; Cutkosky, 2019). Similar methods appear in the literature for other problem templates with no constraint or a single constraint in (Levy, 2017; Levy et al., 2018; Kavis et al., 2019; Cutkosky, 2019). Our method extends these results to optimization over the intersection of convex sets. When f is nonsmooth or stochastic, the proposed method ensures a convergence rate of $\tilde{\mathcal{O}}(1/\sqrt{t})$, whereas the rate of the same algorithm improves to $\tilde{\mathcal{O}}(1/t)$ if f is smooth and there is a solution in the relative interior of the feasible set.

3 TOS for Nonsmooth Setting

Algorithm 1 presents the generalized TOS for Problem (1). It recovers the standard method in (Davis & Yin, 2017) if f is smooth and we choose $u_t = \nabla f(z_t)$. For convenience, we define the mapping

$$\texttt{TOS}_\gamma(y, u) := y - \operatorname{prox}_{\gamma g}(y) + \operatorname{prox}_{\gamma h}\big(2 \cdot \operatorname{prox}_{\gamma g}(y) - y - \gamma u\big) \tag{2}$$

which represents one iteration of Algorithm 1.

The first step of the analysis is the fixed-point characterization of TOS. The following lemma is a straightforward extension of Lemma 2.2 in (Davis & Yin, 2017) to permit subgradients. The proof is a similar to (Davis & Yin, 2017), we present it in the supplementary material for completeness.

Assumption 1. *To avoid pathological examples, we assume the relative interiors of* $\operatorname{dom}(f)$, $\operatorname{dom}(g)$ *and* $\operatorname{dom}(h)$ *have a nonempty intersection.*

Lemma 1 (Fixed points of TOS)**.** *Let* $\gamma > 0$. *Then, there exists a subgradient* $u \in \partial f(\operatorname{prox}_{\gamma g}(y))$ *that satisfies* $\texttt{TOS}_\gamma(y, u) = y$ *if and only if* $\operatorname{prox}_{\gamma g}(y)$ *is a solution to Problem (1).*

Ergodic sequence. Convergence of operator slitting methods are often given in terms of ergodic (averaged) sequences. This strategy requires maintaining the running averages of z_t and x_t:

$$\bar{x}_t = \frac{1}{t+1}\sum_{\tau=0}^{t} x_\tau \qquad \text{and} \qquad \bar{z}_t = \frac{1}{t+1}\sum_{\tau=0}^{t} z_\tau. \tag{3}$$

Clearly, we do not need to store the history of x_t and z_t to maintain these sequences. In practice, the last iterate often converges faster than the ergodic sequence. We can evaluate the objective function at both points and return the one with the smaller value.

We are ready to present convergence guarantees of TOS for the nonsmooth setting.

Assumption 2. *There exists $G_f > 0$ such that*

$$\|u\| \leq G_f, \qquad \forall u \in \partial f(x), \;\; \forall x \in \mathrm{dom}(g). \tag{4}$$

This is a standard assumption in nonsmooth optimization, equivalent to assuming f is G_f-Lipschitz continuous on $\mathrm{dom}(g)$.

Theorem 1. *Consider Problem (1) and let Assumption 2 holds. Employ TOS (Algorithm 1) and choose the update direction and step-size as*

$$u_t \in \partial f(z_t) \quad \textit{and} \quad \gamma_t = \frac{\gamma_0}{\sqrt{T+1}} \;\; \textit{for some } \gamma_0 > 0, \quad \textit{for } t = 0, 1, \ldots, T. \tag{5}$$

Then, the following guarantees hold:

$$f(\bar{z}_T) + g(\bar{z}_T) + h(\bar{x}_T) - \phi_\star \leq \frac{1}{2\sqrt{T+1}} \left(\frac{D^2}{\gamma_0} + \gamma_0 G_f^2 \right) \tag{6}$$

$$\textit{and} \quad \|\bar{x}_T - \bar{z}_T\| \leq \frac{2}{T+1} \left(D + \gamma_0 G_f \right), \quad \textit{where} \quad D = \|y_0 - x_\star\|. \tag{7}$$

If we can estimate D and G_f, we can optimize the factors in Theorem 3 by choosing $\gamma_0 = D/G_f$. This gives $f(\bar{z}_T) + g(\bar{z}_T) + h(\bar{x}_T) - \phi_\star \leq \mathcal{O}(DG_f/\sqrt{T+1})$ and $\|\bar{x}_T - \bar{z}_T\| \leq \mathcal{O}(D/(T+1))$.

Corollary 1. *We are interested in two particular cases of Theorem 3:*

(i). *Suppose h is G_h-Lipschitz continuous. Then,*

$$\phi(\bar{z}_T) - \phi_\star \leq \frac{1}{2\sqrt{T+1}} \left(\frac{D^2}{\gamma_0} + \gamma_0 G_f^2 \right) + \frac{2G_h}{T+1} \left(D + \gamma_0 G_f \right). \tag{8}$$

(ii). *Suppose h is the indicator function of a convex set $\mathcal{H} \subseteq \mathbb{R}^n$. Then,*

$$f(\bar{z}_T) + g(\bar{z}_T) - \phi_\star \leq \frac{1}{2\sqrt{T+1}} \left(\frac{D^2}{\gamma_0} + \gamma_0 G_f^2 \right) \tag{9}$$

$$\textit{and} \quad \mathrm{dist}(\bar{z}_T, \mathcal{H}) \leq \frac{2}{T+1} \left(D + \gamma_0 G_f \right). \tag{10}$$

Proof. (i). Since h is G_h-Lipschitz, we know $\phi(\bar{z}_T) \leq f(\bar{z}_T) + g(\bar{z}_T) + h(\bar{x}_T) + G_h \|\bar{x}_T - \bar{z}_T\|$.
(ii). $h(\bar{x}_T) = 0$ since $\bar{x}_T \in \mathcal{H}$. Moreover, $\mathrm{dist}(\bar{z}_T, \mathcal{H}) := \inf_{x \in \mathcal{H}} \|\bar{z}_T - x\| \leq \|\bar{z}_T - \bar{x}_T\|$. □

Remark 1. *We fix the horizon T for the ease of analysis and presentation. Similar guarantees (with additional logarithmic factors) hold for all iterations if we use $\gamma_t = \gamma_0/\sqrt{t+1}$. We omit the details.*

Theorem 1 automatically covers the case in which g is the indicator of a convex set $\mathcal{G} \subseteq \mathbb{R}^n$. By definition, $\bar{z}_T \in \mathcal{G}$ and $x_\star \in \mathcal{G}$, hence $g(\bar{z}_T) = g(x_\star) = 0$. If both g and h are indicator functions, TOS gives an an approximately feasible solution in $\mathcal{G}$ and close to $\mathcal{H}$. We could also consider a stronger notion of approximate feasibility, measured by $\mathrm{dist}(\bar{z}_T, \mathcal{G} \cap \mathcal{H})$. However, this requires additional regularity assumptions on $\mathcal{G}$ and $\mathcal{H}$ to avoid pathological examples, see Lemma 1 in (Hoffmann, 1992) and Definition 2 in (Kundu et al., 2018).

Problem (1) covers the unconstrained minimization problem as a special case when $g = h = 0$. Hence, the convergence rate in (1) is optimal in the sense it matches the information theoretical lower bounds for the first-order black-box methods, see Section 3.2.1 in (Nesterov, 2003). Remark that the subgradient method can achieve a faster $\mathcal{O}(1/t)$ convergence when f is strongly convex. We leave the convergence rate analysis of TOS for strongly convex f as an open problem.

4 TOS for Stochastic Setting

In this section, we focus on the three-composite stochastic optimization template:

$$\min_{x\in\mathbb{R}^n} \quad \phi(x) := f(x) + g(x) + h(x) \quad \text{where} \quad f(x) := \mathbb{E}_\xi \tilde{f}(x,\xi) \tag{11}$$

and ξ is a random variable. We suppose the following standard assumption holds.

Assumption 3 (Bounded variance)**.** *There exists a stochastic first-order oracle that, upon query at $x \in \mathbb{R}^n$, returns an unbiased estimate $u \in \mathbb{R}^n$ of the (sub)gradient such that*

$$\mathbb{E}_\xi[u|x] = \nabla f(x) \quad \textit{and} \quad \mathbb{E}_\xi[\|u - \nabla f(x)\|^2] \le \sigma^2 \ \textit{for some } \sigma > 0. \tag{12}$$

The following theorem characterizes the convergence rate of Algorithm 1 for Problem (11).

Theorem 2. *Consider Problem (11) and employ TOS (Algorithm 1) with a fixed step-size $\gamma_t = \gamma = \gamma_0/\sqrt{T+1}$ for some $\gamma_0 > 0$. Let Assumption 2 hold and suppose that we are receiving the update direction u_t from a stochastic first-order oracle that satisfies Assumption 3. Then,*

$$\mathbb{E}[f(\bar{z}_T) + g(\bar{z}_T) + h(\bar{x}_T)] - \phi_\star \le \frac{1}{\sqrt{T+1}}\left(\frac{1}{2\gamma_0}D^2 + \gamma_0(\sigma^2 + G_f^2)\right) \tag{13}$$

$$\textit{and} \quad \mathbb{E}[\|\bar{x}_T - \bar{z}_T\|] \le \frac{2}{T+1}\left(D + \gamma_0\left(G_f + \frac{\sigma}{2}\right)\right), \quad \textit{where} \quad D = \|y_0 - x_\star\|. \tag{14}$$

If we can estimate D, G_f and σ, we can optimize the constants by choosing $\gamma_0 \approx D/\max\{G_f, \sigma\}$, which gives $f(\bar{z}_T) + g(\bar{z}_T) + h(\bar{x}_T) - \phi_\star \le \mathcal{O}(D\max\{G_f,\sigma\}/\sqrt{T+1})$ and $\|\bar{x}_T - \bar{z}_T\| \le \mathcal{O}(D/(T+1))$.

The convergence guarantees for the two specific settings, where h is G_h-Lipschitz or an indicator function, can be derived from (13) and (14), similar to Corollary 1. Similar to the nonsmooth setting, the rates shown in this section are optimal since Problem (11) covers $g(x) = h(x) = 0$ as a special case.

Recall that TOS with stochastic gradients have been already studied under various conditions. In particular, the method is known to get $\mathcal{O}(1/t)$ convergence rate when f is smooth and strongly convex (Yurtsever et al., 2016). We present a detailed discussion on the related works and compare our method with them in Section 2.

Remark 2. *Similar guarantees hold if we replace Assumption 2 with the smoothness of f.*

5 TOS with Adaptive Learning Rates

In this section, we focus on an important subclass of Problem (1) where g and h are indicator functions of some closed convex sets,

$$\min_{x\in\mathbb{R}^n} \quad f(x) \quad \text{subject to} \quad x \in \mathcal{G} \cap \mathcal{H}. \tag{15}$$

TOS is effective for (15) particularly when the projection onto $\mathcal{G}$ and $\mathcal{H}$ are easy, but the projection onto their intersection is challenging. Particular examples include the transportation polytopes, doubly nonnegative matrices, and isotonic regression, among many others.

We propose AdapTos, a new variant of TOS with adaptive step-size in the spirit of adaptive online learning algorithms and online to batch conversion techniques, see (Duchi et al., 2011; Rakhlin & Sridharan, 2013; Levy, 2017; Levy et al., 2018; Cutkosky, 2019; Kavis et al., 2019) and the references therein. AdapTos employs the following step-size rule:

$$\gamma_t = \frac{\gamma_0}{\sqrt{\sum_{\tau=0}^{t}\|u_\tau\|^2}} \quad \text{for some } \gamma_0 > 0. \tag{16}$$

We can employ (16) in Algorithm 1 only for Problem (15) because $z_t = \mathrm{prox}_{\gamma_t g}(y_t) = \mathrm{proj}_{\mathcal{G}}(y_t)$ is the same for any $\gamma_t > 0$. In Section 6, we empirically test AdapTos on generic instances of Problem (1). If g is not an indicator function, we use the step-size from the previous iteration (*i.e.,* γ_{t-1}) in the z_t update.

For AdapTos, we will also use a second ergodic sequence, with weighted averages as

$$\tilde{x}_t = \frac{1}{\sum_{\tau=0}^{t} \gamma_\tau} \sum_{\tau=0}^{t} \gamma_\tau x_\tau \qquad \text{and} \qquad \tilde{z}_t = \frac{1}{\sum_{\tau=0}^{t} \gamma_\tau} \sum_{\tau=0}^{t} \gamma_\tau z_\tau \tag{17}$$

for proving convergence in function value. This sequence was also considered for TOS with line-search in (Pedregosa & Gidel, 2018).

Theorem 3. *Consider Problem (15) and let Assumption 2 holds. Employ TOS (Algorithm 1) with the update direction $u_t \in \partial f(z_t)$ and adaptive step-size (16). Then, the following bounds hold:*

$$f(\tilde{z}_t) - f_\star \leq \frac{G_f}{2\sqrt{t+1}} \left(\frac{D^2}{\gamma_0} + \gamma_0 \log(G_f^2(t+1)) \right) \quad \textit{and} \tag{18}$$

$$\operatorname{dist}(\bar{z}_t, \mathcal{H}) \leq \frac{2D}{t+1} + \frac{\gamma_0 \sqrt{\log(G_f^2(t+1))}}{\sqrt{t+1}}, \quad \textit{where} \quad D = \|y_0 - y_\star\|. \tag{19}$$

If D is known, we recommend choosing $\gamma_0 = D$, which gives $f(\tilde{z}_t) - f_\star \leq \tilde{\mathcal{O}}(DG_f/\sqrt{t+1})$ and $\operatorname{dist}(\bar{z}_t, \mathcal{H}) \leq \tilde{\mathcal{O}}(D/\sqrt{t+1})$.

The next theorem establishes a faster rate for the same algorithm when f is smooth and a solution lies in the interior of the feasible set.

Theorem 4. *Consider Problem (15). Let f be differentiable function with L_f-Lipschitz continuous gradient on $\mathcal{G}$ and assume there exists a solution in the interior of the feasible set. Employ TOS (Algorithm 1) with update direction $u_t = \nabla f(z_t)$ and adaptive step-size (16). Then,*

$$f(\bar{z}_t) - f_\star \leq \frac{2L_f}{t+1} \left(\frac{D^2}{2\gamma_0} + \gamma_0 (1 + \log(G_f \sqrt{t})) \right)^2 \quad \textit{and} \tag{20}$$

$$\operatorname{dist}(\bar{z}_t, \mathcal{H}) \leq \frac{1}{t+1} \left(2D + \gamma_0 \sqrt{\log(G_f^2(t+1))} \right), \quad \textit{where} \quad D = \|y_0 - y_\star\|. \tag{21}$$

If $\gamma_0 = D$ is chosen, this gives $f(\bar{z}_t) - f_\star \leq \tilde{\mathcal{O}}(L_f D^2/(t+1))$ and $\operatorname{dist}(\bar{z}_t, \mathcal{H}) \leq \tilde{\mathcal{O}}(D/(t+1))$.

The limitation on the location of the solution is inherit from the prior work (Levy, 2017; Levy et al., 2018), and it creates a challenge for extending these results for the regularized problems. We believe this limitation of the analysis, and the method can achieve the fast rates when f is smooth regardless of where the solution lies. Numerical experiments support this claim.

Finally, the next theorem shows that AdapTos can successfully handle stochastic (sub)gradients.

Theorem 5. *Consider Problem (15) and let Assumption 2 holds. Employ TOS (Algorithm 1) with adaptive step-size (16). Suppose we are receiving the update direction u_t from a stochastic first-order oracle that satisfies Assumption 3. Then, the following bounds hold:*

$$f(\tilde{z}_t) - f_\star \leq \frac{G_f + \sigma}{2\sqrt{t+1}} \left(\frac{D^2}{\gamma_0} + \gamma_0 \log((G_f^2 + \sigma^2)(t+1)) \right) \quad \textit{and} \tag{22}$$

$$\mathbb{E}[\operatorname{dist}(\bar{z}_t, \mathcal{H})] \leq \frac{2D}{t+1} + \frac{\gamma_0 \sqrt{\log((G_f^2 + \sigma^2)(t+1))}}{\sqrt{t+1}}, \quad \textit{where} \quad D = \|y_0 - y_\star\|. \tag{23}$$

Following the definition in (Nesterov, 2015), we say that an algorithm is universal if it does not require to know whether the objective is smooth or not yet it implicitly adapts to the smoothness of the objective. AdapTos attains universal convergence rates for Problem (15). It converges to a solution with $\tilde{\mathcal{O}}(1/\sqrt{t+1})$ rate (in function value) when f is nonsmooth or stochastic. The rate becomes $\tilde{\mathcal{O}}(1/(t+1))$ if f is smooth and the solution is in the interior of the feasible set.

6 Numerical Experiments

This section demonstrates the empirical performance of the proposed method on a number of smooth and nonsmooth convex optimization templates. We also present an experiment on neural networks. Our experiments are performed in Python 3.7 with Intel Core i9-9820X CPU @ 3.30GHz. The source code is available in the supplemental material.

6.1 Experiments in the Smooth Convex Realm

In this subsection, we compare AdapTos with TOS, PDHG and their line-search variants TOS-LS and PDHG-LS. Our experiments are based on the benchmarks described in (Pedregosa & Gidel, 2018) and the source code available in COPT Library (Pedregosa et al., 2020) under the new BSD License. We implement AdapTos and investigate its performance on three different problems:

▷ Logistic regression with *overlapping group lasso* penalty:

$$\min_{x\in\mathbb{R}^n} \quad \frac{1}{N}\sum_{i=1}^{N}\log(1+\exp(-b_i\langle a_i, x\rangle)) + \lambda\sum_{G\in\mathcal{G}}\sqrt{|G|}\|x_G\| + \lambda\sum_{H\in\mathcal{H}}\sqrt{|H|}\|x_H\|, \tag{24}$$

where $\{(a_1, b_1), \ldots, (a_N, b_N)\}$ is a given set of training examples, $\mathcal{G}$ and $\mathcal{H}$ are the sets of distinct groups and $|\cdot|$ denotes the cardinality. The model we use (from COPT) considers groups of size 10 with 2 overlapping coefficients. In this experiment, we use the benchmarks on synthetic data (dimensions $n = 1002$, $N = 100$) and RCV1 dataset (Lewis et al., 2004) ($n = 677399$, $N = 20242$).

▷ Image recovery with *total variation* penalty:

$$\min_{X\in\mathbb{R}^{m\times n}} \quad \|Y-\mathcal{A}(X)\|_F^2 + \lambda\sum_{i=1}^{m}\sum_{j=1}^{n-1}|X_{i,j+1}-X_{i,j}| + \lambda\sum_{j=1}^{n}\sum_{i=1}^{m-1}|X_{i+1,j}-X_{i,j}|, \tag{25}$$

where Y is a given blurred image and $\mathcal{A}:\mathbb{R}^{m\times n}\to\mathbb{R}^{m\times n}$ is a linear operator (blur kernel). The benchmark in COPT solves this problem for an image of size 153×115 with a provided blur kernel.

▷ Sparse and low-rank matrix recovery via ℓ_1 *and nuclear-norm* regularizations:

$$\min_{X\in\mathbb{R}^{n\times n}} \quad \frac{1}{N}\sum_{i=1}^{N}\texttt{huber}(b_i-\langle A_i, X\rangle) + \lambda\|X\|_* + \lambda\|X\|_1. \tag{26}$$

We use `huber` loss. $\{(A_1, b_1), \ldots, (A_N, b_N)\}$ is a given set of measurements and $\|X\|_1$ is the vector ℓ_1-norm of X. The benchmark in COPT considers a symmetric ground truth matrix $X^\natural \in \mathbb{R}^{20\times 20}$ and noisy synthetic measurements ($N = 100$) where A_i has Gaussian iid entries. $b_i = \langle A_x, X\rangle + \omega_i$, where ω_i is generated from a zero-mean unit variance Gaussian distribution.

At each problem, we consider two different values for the regularization parameter λ. We use all methods with their default parameters in the benchmark. For AdapTos, we tune γ_0 by trying the powers of 10, see the supplement for the values found. Figure 1 shows the results of this experiment. In most cases, the performance of AdapTos lies in between TOS-LS and PDHG-LS. Remark that TOS-LS is using the extra knowledge of the Lipschitz constant of h.

6.2 Experiments in the Nonsmooth Convex Setting

We examine the empirical performance of AdapTos for nonsmooth problems on an image impainting and denoising task from (Zeng & So, 2018; Yurtsever et al., 2018). We are given an occluded image (i.e., missing some pixels) of size 517×493, contaminated with salt and pepper noise of 10% density. We use the following template where data fitting is measured in terms of vector ℓ_p-norm:

$$\min_{\boldsymbol{X}\in\mathbb{R}^{m\times n}} \quad \|\mathcal{A}(X)-Y\|_p \quad \text{subject to} \quad \|X\|_* \le \lambda, \quad 0\le X\le 1, \tag{27}$$

where Y is the observed noisy image with missing pixels. This is essentially a matrix completion problem, $\mathcal{A}:\mathbb{R}^{m\times n}\to\mathbb{R}^{m\times n}$ is a linear map that samples the observed pixels in Y. In particular, we consider (27) with $p = 1$ and $p = 2$. The ℓ_2-loss is common in practice for matrix completion (often in the least-squares form) but it is not very robust against the outliers induced by salt and pepper noise. ℓ_1-loss is known to be more reliable for this task.

The subgradients in both cases have a fixed norm at all points (note that the subgradients are binary valued for ℓ_1-loss, and unit-norm for ℓ_2-loss), hence the analytical ($\gamma_t = \gamma_0/\sqrt{t+1}$) and adaptive (16) step-sizes are the same up to a constant factor. We present AdapTos only.

Figure 2 shows the results. The empirical rates for $p = 1$ roughly match our guarantees in Theorem 1. We observe a faster locally linear convergence rate when ℓ_2-loss is used. Interestingly, the ergodic

sequence converges faster than the last iterate on $p = 1$ but significantly slower in $p = 2$. The runtime of the two settings are approximately the same, with 81.4 (for ℓ_1) versus 80.1 (for ℓ_2) msec per iteration on average. Despite the slower rates, we found ℓ_1-loss more practical on this problem. A low-accuracy solution obtained by 1000 iterations (approximately 80 sec) on ℓ_1-loss yields a high quality recovery with PSNR 26.19 dB, whereas the PSNR saturates at 21.15 dB for the ℓ_2-formulation. See the supplements for the visuals.

6.3 An Experiment on Neural Networks

In this section, we train a regularized deep neural network to test our methods on nonconvex optimization. We consider a regularized neural network problem formulation in (Scardapane et al., 2017). This problem involves a fully connected neural network with the standard cross-entropy loss function, a ReLu activation for the hidden layers, and the softmax activation for the output layer. Two regularizers are added to this loss function: The first one is the standard ℓ_1 regularizer, and the second is the group sparse regularizer where the outgoing connections of each neuron is considered as a group. The goal is to force all outgoing connections from the same neurons to be simultaneously zero, so that we can safely remove the neurons from the network. This is shown as an effective way to obtain compact networks (Scardapane et al., 2017), which is crucial for the deployment of the learned parameters on resource-constrained devices such as smartphones (Blalock et al., 2020).

We reuse the open source implementation (built with Lasagne framework based on Theano) published in (Scardapane et al., 2017) under BSD-2 License. We strictly follow their experimental setup and instructions, with the MNIST database (LeCun, 1998) containing 70k grayscale images (28×28) of handwritten digits (split 75/25 into train and test partitions). We train a fully connected neural network with 784 input features, three hidden layers ($400/300/100$) and 10-dimensional output layer. Interested readers can find more details on the implementation in the supplementary material or in (Scardapane et al., 2017).

Scardapane et al. (2017) use SGD and Adam with the subgradient of the overall objective. In contrast, our methods can leverage the prox-operators of the regularizers. Figure 3 compares the performance in two terms: the sparsity of the parameters and the accuracy. On the left side, we see the spectrum of weight and neuron magnitudes. The advantage of using prox-operators is outstanding: More than %93 of the weights are zero and 68% of neurons are inactive when trained with AdapTos. In contrast, subgradient based methods can achieve only approximately sparse solutions.

The third and the fourth subplots present the training and test accuracy. Remarkably, AdapTos performs better than the state-of-the-art (both in train and test). Unfortunately, we could not achieve the same performance gain in preliminary experiments with more complex models like ResNet (He et al., 2016), where SGD with momentum shines. We leave the technical analysis and a comprehensive examination of AdapTos for nonconvex problems to a future work.

7 Conclusions

We studied an extension of TOS that permits subgradients and stochastic gradients instead of the gradient step, and we established the convergence guarantees of this method. Moreover, we proposed an adaptive step-size rule (AdapTos) for the minimization of a convex function over the intersection of two convex sets. AdapTos guarantees a nearly optimal $\tilde{\mathcal{O}}(1/\sqrt{t})$ rate on the baseline setting, and it enjoys the faster $\tilde{\mathcal{O}}(1/t)$ rate when the problem is smooth and the solution is in the relative interior of the domain. We present numerical experiments on various benchmark problems. The empirical performance of the method is promising.

We conclude with a short list of open questions and follow-up directions: (i) In parallel to the subgradient method, we believe TOS can achieve $\mathcal{O}(1/t)$ rate guarantees in the nonsmooth setting if f is strongly convex. The analysis remains open. (ii) The faster rate for AdapTos on smooth f requires an extra assumption on the location of the solution. We believe this assumption can be removed, and leave this as an open problem. (iii) We analyzed AdapTos only for a specific subclass of Problem (1) in which g and h are indicator functions. Extending this result for the whole class is a valuable question for future study.

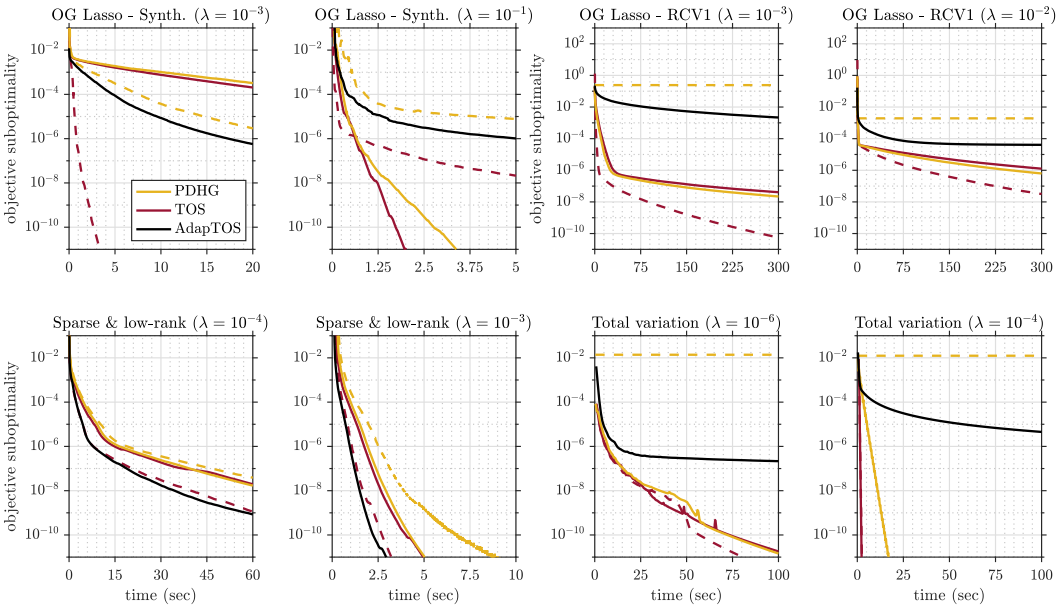

Figure 1: Empirical comparison of 5 algorithms for Problem (1) with smooth f. Dashed lines represent the line-search variants of TOS and PDHG. We observed the performance of AdapTos lies between TOS-LS and PDHG-LS in all cases. TOS and PDHG require the knowledge of the smoothness constant, and TOS-LS uses the Lipschitz constant the regularizer.

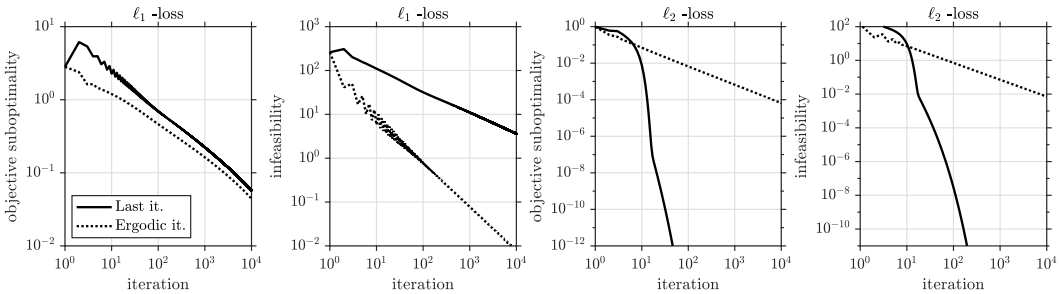

Figure 2: Performance of AdapTos on image impainting and denoising problems with ℓ_1 and ℓ_2-loss functions (mind the difference in the axes limits). The convergence rates on the ℓ_1 norm match the guaranteed rates, $\mathcal{O}(1/\sqrt{t})$ in objective suboptimality and $\mathcal{O}(1/t)$ in infeasibility. The method achieves a locally linear convergence rate on the ℓ_2-loss.

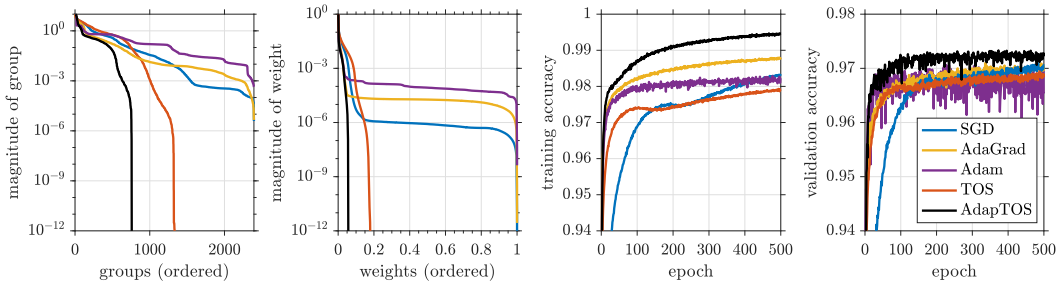

Figure 3: Comparison of methods on training neural networks with group lasso regularization. The outgoing connections of each neuron form a group. The first plot shows the magnitude of weights after 500 epochs. The second plot shows the absolute sum of outgoing weights from each neuron. x-axes are normalized by the total number of weights and neurons in these plots. More than 68% of the neurons are inactive on the network trained by AdapTos. The third and the fourth plots show the training and validation loss.

References

Azadi, S. and Sra, S. Towards an optimal stochastic alternating direction method of multipliers. In *International Conference on Machine Learning*, pp. 620–628. PMLR, 2014.

Barbero, A. and Sra, S. Modular proximal optimization for multidimensional total-variation regularization. *The Journal of Machine Learning Research*, 19(1):2232–2313, 2018.

Bauschke, H. H., Combettes, P. L., et al. *Convex analysis and monotone operator theory in Hilbert spaces*, volume 408. Springer, 2011.

Blalock, D., Ortiz, J. J. G., Frankle, J., and Guttag, J. What is the state of neural network pruning? *arXiv preprint arXiv:2003.03033*, 2020.

Briceño-Arias, L. M. Forward-douglas–rachford splitting and forward-partial inverse method for solving monotone inclusions. *Optimization*, 64(5):1239–1261, 2015.

Cevher, V., Vũ, B. C., and Yurtsever, A. Stochastic forward douglas-rachford splitting method for monotone inclusions. In *Large-Scale and Distributed Optimization*, pp. 149–179. Springer, 2018.

Chambolle, A. and Pock, T. On the ergodic convergence rates of a first-order primal–dual algorithm. *Mathematical Programming*, 159(1):253–287, 2016.

Condat, L. A primal–dual splitting method for convex optimization involving lipschitzian, proximable and linear composite terms. *Journal of optimization theory and applications*, 158(2):460–479, 2013.

Cutkosky, A. Anytime online-to-batch, optimism and acceleration. In *International Conference on Machine Learning*, pp. 1446–1454. PMLR, 2019.

Davis, D. and Yin, W. A three-operator splitting scheme and its optimization applications. *arXiv preprint arXiv:1504.01032*, 2015.

Davis, D. and Yin, W. A three-operator splitting scheme and its optimization applications. *Set-valued and variational analysis*, 25(4):829–858, 2017.

Ding, L., Yurtsever, A., Cevher, V., Tropp, J. A., and Udell, M. An optimal-storage approach to semidefinite programming using approximate complementarity. *arXiv preprint arXiv:1902.03373*, 2019.

Duchi, J., Hazan, E., and Singer, Y. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of machine learning research*, 12(7), 2011.

El Halabi, M. and Cevher, V. A totally unimodular view of structured sparsity. In *Artificial Intelligence and Statistics*, pp. 223–231. PMLR, 2015.

He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

Higham, N. J. and Strabić, N. Anderson acceleration of the alternating projections method for computing the nearest correlation matrix. *Numerical Algorithms*, 72(4):1021–1042, 2016.

Hoffmann, A. The distance to the intersection of two convex sets expressed by the distances to each of them. *Mathematische Nachrichten*, 157(1):81–98, 1992.

Kavis, A., Levy, K. Y., Bach, F., and Cevher, V. Unixgrad: A universal, adaptive algorithm with optimal guarantees for constrained optimization. In *Proceedings of the 33rd International Conference on Neural Information Processing Systems*, 2019.

Kundu, A., Bach, F., and Bhattacharya, C. Convex optimization over intersection of simple sets: improved convergence rate guarantees via an exact penalty approach. In *International Conference on Artificial Intelligence and Statistics*, pp. 958–967. PMLR, 2018.

LeCun, Y. The mnist database of handwritten digits. *http://yann. lecun. com/exdb/mnist/*, 1998.

Levy, K. Y. Online to offline conversions, universality and adaptive minibatch sizes. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, 2017.

Levy, K. Y., Yurtsever, A., and Cevher, V. Online adaptive methods, universality and acceleration. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, 2018.

Lewis, D. D., Yang, Y., Russell-Rose, T., and Li, F. Rcv1: A new benchmark collection for text categorization research. *Journal of machine learning research*, 5(Apr):361–397, 2004.

Malitsky, Y. and Pock, T. A first-order primal-dual algorithm with linesearch. *SIAM Journal on Optimization*, 28(1):411–432, 2018.

Nesterov, Y. *Introductory lectures on convex optimization: A basic course*, volume 87. Springer Science & Business Media, 2003.

Nesterov, Y. Universal gradient methods for convex optimization problems. *Mathematical Programming*, 152(1):381–404, 2015.

Ouyang, H., He, N., Tran, L., and Gray, A. Stochastic alternating direction method of multipliers. In *International Conference on Machine Learning*, pp. 80–88. PMLR, 2013.

Pedregosa, F. On the convergence rate of the three operator splitting scheme. *arXiv preprint arXiv:1610.07830*, 2016.

Pedregosa, F. and Gidel, G. Adaptive three operator splitting. In *International Conference on Machine Learning*, pp. 4085–4094, 2018.

Pedregosa, F., Fatras, K., and Casotto, M. Proximal splitting meets variance reduction. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pp. 1–10. PMLR, 2019.

Pedregosa, F., Negiar, G., and Dresdner, G. copt: composite optimization in python. 2020. doi: 10.5281/zenodo.1283339. URL http://openo.pt/copt/.

Raguet, H., Fadili, J., and Peyré, G. A generalized forward-backward splitting. *SIAM Journal on Imaging Sciences*, 6(3):1199–1226, 2013.

Rakhlin, A. and Sridharan, K. Optimization, learning, and games with predictable sequences. In *Proceedings of the 26th International Conference on Neural Information Processing Systems-Volume 2*, pp. 3066–3074, 2013.

Scardapane, S., Comminiello, D., Hussain, A., and Uncini, A. Group sparse regularization for deep neural networks. *Neurocomputing*, 241:81–89, 2017.

Shivanna, R., Chatterjee, B., Sankaran, R., Bhattacharyya, C., and Bach, F. Spectral norm regularization of orthonormal representations for graph transduction. In *Neural Information Processing Systems*, 2015.

Tibshirani, R. J., Hoefling, H., and Tibshirani, R. Nearly-isotonic regression. *Technometrics*, 53(1): 54–61, 2011.

Vũ, B. C. A splitting algorithm for dual monotone inclusions involving cocoercive operators. *Advances in Computational Mathematics*, 38(3):667–681, 2013.

Yan, M. A new primal–dual algorithm for minimizing the sum of three functions with a linear operator. *Journal of Scientific Computing*, 76(3):1698–1717, 2018.

Yuan, L., Liu, J., and Ye, J. Efficient methods for overlapping group lasso. *Advances in neural information processing systems*, 24:352–360, 2011.

Yurtsever, A., Vũ, B. C., and Cevher, V. Stochastic three-composite convex minimization. In *Proceedings of the 30th International Conference on Neural Information Processing Systems*, pp. 4329–4337, 2016.

Yurtsever, A., Fercoq, O., Locatello, F., and Cevher, V. A conditional gradient framework for composite convex minimization with applications to semidefinite programming. In *International Conference on Machine Learning*, pp. 5727–5736. PMLR, 2018.

Zeng, W.-J. and So, H. C. Outlier–robust matrix completion via ℓ_p-minimization. *IEEE Trans. on Sig. Process*, 66(5):1125–1140, 2018.

Zhao, R. and Cevher, V. Stochastic three-composite convex minimization with a linear operator. In *International Conference on Artificial Intelligence and Statistics*, pp. 765–774. PMLR, 2018.

Zhao, R., Haskell, W. B., and Tan, V. Y. An optimal algorithm for stochastic three-composite optimization. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pp. 428–437. PMLR, 2019.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes] We explicitly state limitations of our methods and its analysis in multiple places in the paper.
   (c) Did you discuss any potential negative societal impacts of your work? [No] We present a new convex optimization technique and its technical analysis, we do not expect any direct societal impact.
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [Yes] We listed our assumptions in the beginning of each Theorem.
   (b) Did you include complete proofs of all theoretical results? [Yes] In the supplementary material.
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes] In the supplementary material.
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] Some details are deferred to the supplementary material.
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [No] Most of our experiments are on convex and deterministic problems which do not involve randomness.
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes] In the beginning of Section 5.
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes]
   (b) Did you mention the license of the assets? [Yes]
   (c) Did you include any new assets either in the supplemental material or as a URL? [Yes] We implemented the proposed methods for the openopt/copt Python library.
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [Yes] We used standard benchmark datasets with citation or synthetic data.
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [No] The datasets we use do not contain any personal information to the best of our knowledge.
5. If you used crowdsourcing or conducted research with human subjects...

(a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]

(b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]

(c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]