# Adaptive Three Operator Splitting

Fabian Pedregosa [1 2] Gauthier Gidel [3]

## Abstract

We propose and analyze a novel adaptive step-size variant of the Davis-Yin three operator splitting, a method that can solve optimization problems composed of a sum of a smooth term for which we have access to its gradient and an arbitrary number of potentially non-smooth terms for which we have access to their proximal operator. The proposed method leverages local information of the objective function, allowing for larger step-sizes while preserving the convergence properties of the original method. It only requires two extra function evaluations per iteration and does not depend on any step-size hyperparameter besides an initial estimate. We provide a convergence rate analysis of this method, showing sublinear convergence rate for general convex functions and linear convergence under stronger assumptions, matching the best known rates of its non adaptive variant. Finally, an empirical comparison with related methods on 6 different problems illustrates the computational advantage of the adaptive step-size strategy.

## 1 Introduction

Minimizing the sum of a smooth and a non-smooth term is at the core of many optimization problems that arise in machine learning and signal processing (Rudin et al., 1992; Candès et al., 2006; Chambolle & Pock, 2016). In a few but important cases, such as $\ell_1$ or group lasso regularization, the non-smooth term is simple enough so that its proximal operator is available in closed form or at least fast to compute. In this case, highly scalable methods such as proximal gradient descent (Beck & Teboulle, 2009; Nesterov et al., 2013) or proximal coordinate descent (Richtárik & Takáč, 2014) have shown state of the art performance. However, the desire to model increasingly complex phenomena has led to the development of a flurry of penalties with costly to compute proximal operator. Examples are the overlapping group lasso (Jacob et al., 2009), multidimensional total variation (Barbero & Sra, 2014) or trend filtering (Kim et al., 2009), to name a few.

A key observation is that, despite the difficulty in computing its proximal operator, many of these penalties can be decomposed as a sum of terms for which we have access to their proximal operator. Proximal splitting methods like the three operator splitting (Davis & Yin, 2017) offer a principled way to incorporate these penalties into the optimizer. In this work we will describe a method to solve optimization problems of the form

$$\operatorname*{minimize}_{\boldsymbol{x}\in\mathbb{R}^p} f(\boldsymbol{x}) + g(\boldsymbol{x}) + h(\boldsymbol{x}) \;, \qquad \text{(OPT)}$$

where $f$ is convex and $L_f$-smooth (i.e., differentiable with $L_f$-Lipschitz gradient) and $g, h$ are both convex but potentially non-smooth. We further assume $g$ and $h$ are *proximal*, i.e., we have access to the proximal operator.

This formulation allows to express a broad range of problems arising in machine learning and signal processing: the smooth term includes the least squares or logistic loss functions; the two proximal terms can be extended to an arbitrary number via a product space formulation and as we will see in §4.1 include many important penalties such as the group lasso with overlap, total variation, $\ell_1$ trend filtering, etc. Furthermore, the penalties can be extended-valued, thus allowing an intersection for convex constraints through the use of the indicator function.

**The three operator splitting** (TOS) method (Davis & Yin, 2017) is a recently proposed method for problems of the form (OPT). At each iteration, it only requires to evaluate once the gradient of $f$ and the proximal operator of $g$ and $h$. It also relies on one step-size parameter, and while it can be set based on the Lipschitz constant of the gradient of $f$, this is not entirely satisfactory for two reasons. First, this constant is often costly to compute. Second, this constant is a global upper bound on the Lipschitz constant, while locally the Lipschitz constant might be smaller, allowing for larger step-sizes.

[1]University of California at Berkeley, USA [2]Department of Computer Science, ETH Zurich, Switzerland [3]Universit de Montral - DIRO Montral Institute for Learning Algorithms, Canada. Correspondence to: Fabian Pedregosa <f@bian.net>.

*Proceedings of the $35^{th}$ International Conference on Machine Learning*, Stockholm, Sweden, PMLR 80, 2018. Copyright 2018 by the author(s).

*Adaptive step-size* methods, also known as inexact and backtracking line search, instead choose the step-size by verifying a sufficient decrease condition at each iteration. This allows to take larger step-sizes and has proven to be an important ingredient in the practical implementation of first and second-order methods (Nocedal & Wright, 2006).

**Outline and main contributions.** Our main contribution is the development and analysis of an adaptive variant of the TOS algorithm. The proposed algorithm does not depend on any step-size hyperparameter (besides an initial estimate) and enjoys similar convergence guarantees as the non adaptive variant. The paper is organized as follows:

- *Methods*. §2 describes the proposed algorithm, extended in §2.1 to an arbitrary number of proximal terms.
- *Analysis*. §3 provides a convergence analysis based on an interpretation of the algorithm as a saddle-point optimization method. This significantly departs from the analysis of Davis & Yin (2017) for the non adaptive variant and results in improved and more general rates.
- *Applications*. §4 discusses the application to different penalties and presents an empirical comparison on 6 different problems and 5 different penalties.

**Notation.** We denote vectors with boldface lower case letters (i.e., $\boldsymbol{x}$), and matrices and vector-valued functions in boldface upper case (i.e., $\boldsymbol{X}, \boldsymbol{T}(\cdot)$). $\|\cdot\|$ denotes the euclidean vector norm. Given a matrix $\boldsymbol{X} \in \mathbb{R}^{n \times p}$, we denote by $\overline{\boldsymbol{X}}$ the average along rows, that is, $\overline{\boldsymbol{X}} = {}^1\!/_n \sum_{i=1}^n \boldsymbol{X}_i$. We make extensive use of the proximal operator, defined for a convex function $\varphi$ and $\gamma > 0$ as

$$\mathbf{prox}_{\gamma\varphi}(\boldsymbol{x}) \overset{\text{def}}{=} \mathop{\mathbf{arg\,min}}_{\boldsymbol{z} \in \mathbb{R}^p} \left\{ \varphi(\boldsymbol{z}) + \frac{1}{2\gamma}\|\boldsymbol{x} - \boldsymbol{z}\|^2 \right\} . \quad (1)$$

The domain of a function $f : \mathbb{R}^p \to ]-\infty, \infty]$ is $\operatorname{dom} f \overset{\text{def}}{=} \{\boldsymbol{x} \in \mathbb{R}^p | f(\boldsymbol{x}) < \infty\}$. The indicator function is denoted $\imath\{\text{condition}\}$, which is 0 if condition is verified and $+\infty$ otherwise. Basic properties and definitions of convex functions are provided for convenience in Appendix A.

### 1.1 Related work

Proximal splitting methods that can solve problems involving a sum of terms by accessing the proximal operators of their constituents can be traced back to the 1970s in the works of Glowinski & Marroco (1975); Gabay & Mercier (1976); Lions & Mercier (1979). There has been a surge in interest in these methods in the last years due to their applicability in machine learning (Parikh & Boyd, 2013), signal processing (Combettes & Pesquet, 2011) and parallel optimization (Boyd et al., 2011).

Algorithms to solve problems of the form (OPT) with two or more proximal terms and a smooth term accessed via its gradient have recently been proposed. Examples are the generalized forward-backward splitting (Raguet et al., 2013), the three operator splitting (TOS) (Davis & Yin, 2017), the primal-dual hybrid gradient (PDHG) method, proposed in (Condat, 2013b; Vũ, 2013) and analyzed by Chambolle & Pock (2015) and the very recent primal-dual three operator splitting (Yan, 2018). We note that the last two methods can optimize a more general objective function in which $h(\boldsymbol{x})$ is replaced with $h(\boldsymbol{K}\boldsymbol{x})$ for an arbitrary matrix $\boldsymbol{K}$. The original formulation of these methods requires to set the step-size based on criteria such as the Lipschitz constant of the gradient of the smooth term, but variants with adaptive step-size have recently emerged.

An adaptive step-size variant of the PDHG algorithm has recently been proposed by Malitsky & Pock (2018, §5). Compared to the proposed method, it requires one less function evaluation per iteration but since the original algorithm has two step-sizes, it still relies on one step-size hyperparameter. Convergence rates are not derived.

A different adaptive step-size strategy was proposed by Giselsson et al. (2016) as a general scheme for averaged operators. TOS is averaged for step-sizes $< 2/L_f$, and we denote the combination of both methods LSAO-TOS. An $\mathcal{O}(1/\sqrt{t})$ convergence rate in terms of the operator residual norm is derived. Unfortunately, this quantity is difficult to relate to the more common objective function suboptimality used in the other contributions.

Another adaptive step-size variant of TOS was proposed without proof in the technical report Davis & Yin (2015, Algorithm 3). It uses the same sufficient decrease inequality as our method, although the iterates are defined differently. We found the algorithm sometimes non-convergent and did not consider it further.

In contrast, we provide a convergence analysis for our method that achieves a $\mathcal{O}(1/t)$ convergence rate for the ergodic (i.e., averaged) iterate, and linear convergence under stronger assumptions, matching and in some cases even improving the best known rates of the non adaptive variant.

| Method | Adaptive | Sublinear rate | Linear rate |
|---|---|---|---|
| Adaptive TOS<br>(*this work*) | ✓ | ✓ | ✓ |
| TOS<br>(Davis & Yin, 2017) | ✗ | ✓ | ✓ |
| LSAO-TOS<br>(Giselsson et al., 2016) | ✓ | ✓[1] | ✗ |
| PDHG<br>(Condat, 2013b; Vũ, 2013) | ✗ | ✓ | ✗ |
| PDHG-LS<br>(Malitsky & Pock, 2018) | ✓ | ✗ | ✗ |

[1] Convergence rate in terms of operator residuals.

## 2 Methods

In this section we present our main contribution, a three operator splitting method with adaptive step-size. The method is detailed in Algorithm 1 and requires at each iteration to evaluate once the gradient of $f$ and the proximal operators of $g$ and $h$, and perform two function evaluations of $f$. At iteration $t$ the candidate step-size $\gamma_t$ is chosen as to verify the following *sufficient decrease* condition between the iterates $\boldsymbol{z}_t$ and $\boldsymbol{x}_{t+1}$ (Line 4):

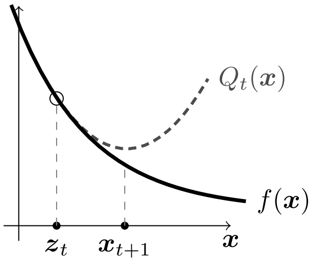
**Algorithm 1:** Adaptive Three Operator Splitting

```
Input: $z_0 \in \mathbb{R}^p$, $u_0 \in \mathbb{R}^p$, $\gamma_0 > 0, \tau \in (0, 1)$
1  for $t = 0, 1, 2, \ldots$ do
2      repeat                                          ▷ step-size search loop
3          $x_{t+1} = \mathbf{prox}_{\gamma_t g}(z_t - \gamma_t u_t - \gamma_t \nabla f(z_t))$
4          if $f(x_{t+1}) \leq Q_t(x_{t+1})$ then
5              break                                   ▷ sufficient decrease verified
6          else
7              $\gamma_t = \tau\gamma_t$                              ▷ decrease step-size
8      $z_{t+1} = \mathbf{prox}_{\gamma_t h}(x_{t+1} + \gamma_t u_t)$
9      $u_{t+1} = u_t + (x_{t+1} - z_{t+1})/\gamma_t$
10             ▷ choose step-size for next iteration, two variants
11     Variant 1
12         $\gamma_{t+1} = \gamma_t$
13     Variant 2                                       ▷ only if $h$ is $\beta_h$-Lipschitz
14         $\delta_t = Q_t(x_{t+1}) - f(x_{t+1})$
15         Choose any $\gamma_{t+1} \in [\gamma_t, \sqrt{\gamma_t^2 + \gamma_t\delta_t(2\beta_h)^{-2}}]$
16 return $x_{t+1}$, $u_{t+1}$
```

$$
f(\boldsymbol{x}_{t+1}) \leq Q_t(\boldsymbol{x}_{t+1})\,, \quad \text{with } Q_t \text{ defined as}
$$

$$
Q_t(\boldsymbol{x}) \overset{\text{def}}{=} f(\boldsymbol{z}_t) + \langle \nabla f(\boldsymbol{z}_t), \boldsymbol{x} - \boldsymbol{z}_t \rangle + \frac{1}{2\gamma_t}\|\boldsymbol{x} - \boldsymbol{z}_t\|^2\,. \tag{2}
$$

This inequality can be interpreted as a quadratic upper bound on the function $f$ at $\boldsymbol{x}_{t+1}$. Effectively, the right-hand side of the above equation is a quadratic $Q_t$ which is tangent to $f$ at $\boldsymbol{z}_t$ and has amplitude $(2\gamma_t)^{-1}$. Then the sufficient decrease condition amounts to verifying that the quadratic function $Q_t$ estimated using local information of $f$ at $\boldsymbol{z}_t$ is still an upper bound at $\boldsymbol{x}_{t+1}$.



By the properties of $L_f$-smooth functions, the sufficient decrease condition is verified for any $\gamma_t \leq 1/L_f$. Hence the line search loop always has a finite termination and the step-size is lower bounded by $\gamma_t \geq \min\{\tau/L_f, \gamma_0\}$. The practical advantage of this strategy is that it allows to consider a step-size potentially larger than $1/L_f$ and verify whether the above is verified at each iteration. If it is, then the algorithm uses the current step-size, and if not, it decreases the step-size by a factor which we denote $\tau$.

**Growing step-size strategies.** We consider two different strategies to initialize next iterate step-size. The first strategy (*Variant 1*) is the simplest and consists in initializing the next step-size with the current one (Line 12). In this variant, the step-size is only allowed to decrease.

The second strategy (*Variant 2*) allows the step-size to increase but in exchange requires the proximal term $h$ to be Lipschitz continuous (note, not smooth as $f$ but only Lipschitz). This is the case of most penalties (i.e., $\ell_1$, group lasso, total variation, etc.) but not of indicator functions and so is less general than the first variant. As we will see in the applications section, the ability to grow the step-size has an important effect on its empirical performance.

**Initial and default values.** The proposed method takes as input 4 parameters, which we briefly discuss, together with a growing step-size heuristic for Variant 2:

- *Initial guess $\boldsymbol{z}_0$ and $\boldsymbol{u}_0$.* $\boldsymbol{z}_0$ is an initial guess of the primal problem (OPT), while $\boldsymbol{u}_0$ is an initial guess for a minimizer of a (yet to be defined) dual function (8). In practice, we initialize both variables to zero.

- *Initial step-size $\gamma_0$.* To estimate a starting value for the step-size, we start with $\varepsilon = 10^{-3}$, $\widetilde{\boldsymbol{z}} = \boldsymbol{z}_0 - \varepsilon\nabla f(\boldsymbol{z}_0)$ and divide $\varepsilon$ by 10 until $f(\widetilde{\boldsymbol{z}}) \leq f(\boldsymbol{z}_0)$. Then we solve $f(\widetilde{\boldsymbol{z}}) = Q_0(\widetilde{\boldsymbol{z}})$ for $\gamma_0$ and double that estimate, giving
$$
\gamma_0 = 4(f(\boldsymbol{z}_0) - f(\widetilde{\boldsymbol{z}}))\|\nabla f(\boldsymbol{z}_0)\|^{-2}\,. \tag{3}
$$

- The *line search decrease parameter* $\tau$ regulates the factor by which the step-size is decreased each time the line search condition is unsuccessful. This is a parameter that is common to all line search methods and can be set to any value $\tau \in (0, 1)$. Following (Malitsky & Pock, 2018) we set it to $\tau = 0.7$.

- *step-size growth.* Variant 2 allows the step-size to grow by an amount that depends on $\beta_h^{-2}$. This quantity can be arbitrarily large (e.g., vanishing regularization), and so choosing the largest admissible step-size might result in too many decrease corrections. This can be avoided e.g. by limiting its growth to double every 20 iterations. Line 15 then becomes:
$$
\gamma_{t+1} = \min\{\gamma_t 2^{0.05}, \sqrt{\gamma_t^2 + \gamma_t\delta_t(2\beta_h)^{-2}}\}\,. \tag{4}
$$

Upon termination, the algorithm returns two vectors. The first vector is an approximate solution to (OPT), while the second vector is an approximate minimizer of a dual objective which we will detail in §3.

**Special cases and related methods.** We mention two notable special cases of this algorithm. First, for any step-size $\gamma_t \leq 1/L_f$, the line search condition will always succeed by the properties of $L_f$-smooth functions and so the step-size in Variant 1 is constant. Defining $\boldsymbol{y}_t = \boldsymbol{x}_t + \gamma_t \boldsymbol{u}_{t-1}$, it is easy to verify that Algorithm 1 (Variant 1) can be written with a constant step-size $\gamma = \gamma_t$ as an iteration of the form

$$\begin{aligned} \boldsymbol{z}_t &= \mathbf{prox}_{\gamma h}(\boldsymbol{y}_t) \\ \boldsymbol{x}_{t+1} &= \mathbf{prox}_{\gamma g}(2\boldsymbol{z}_t - \boldsymbol{y}_t - \gamma \nabla f(\boldsymbol{z}_t)) \\ \boldsymbol{y}_{t+1} &= \boldsymbol{y}_t - \boldsymbol{z}_t + \boldsymbol{x}_{t+1} \ , \end{aligned} \tag{5}$$

which is the standard (non-overrelaxed) form of the three operator splitting (Davis & Yin, 2017, Algorithm 1). The adaptive variant requires two extra function evaluations $f(\boldsymbol{z}_t)$ and $f(\boldsymbol{x}_{t+1})$ for the line search condition in Line 4, but as we will see in the experimental section, most often the ability to take larger step outweighs this extra cost.

Second, for $h = 0$, we have from lines 8 and 9 that $\boldsymbol{u}_t = 0$ and in this case (ignoring growing step-size strategies), this algorithm simplifies to the proximal gradient descent with line search of (Beck & Teboulle, 2009).

Algorithm 1 can be written equivalently in a way that highlights similarities and differences with the PDHG method. Using Moreau's decomposition $\mathbf{prox}_{\gamma h}(\boldsymbol{x}) = \boldsymbol{x} - \gamma\, \mathbf{prox}_{\gamma h^\star}(\boldsymbol{x}/\gamma)$ yields the following recurrence

$$\begin{aligned} \boldsymbol{u}_{t+1} &= \mathbf{prox}_{h^\star/\gamma}(\boldsymbol{u}_t + \boldsymbol{x}_t/\gamma) \ , \\ \boldsymbol{x}_{t+2} &= \mathbf{prox}_{\gamma g}(\boldsymbol{x}_{t+1} - \gamma(\nabla f(\boldsymbol{z}_{t+1}) + 2\boldsymbol{u}_{t+1} - \boldsymbol{u}_t)) \ . \end{aligned} \tag{6}$$

This form is almost identical to Algorithm 3.2 in (Condat, 2013b), but with a different step-size and the gradient evaluated at the extrapolated $\boldsymbol{z}_{t+1} = \boldsymbol{x}_{t+1} - \gamma(\boldsymbol{u}_{t+1} - \boldsymbol{u}_t)$ instead of the previous iterate $\boldsymbol{x}_{t+1}$ in PDHG.

### 2.1 Extension to multiple proximal terms

We now consider the problem of minimizing an objective of the form:

$$\underset{\boldsymbol{x} \in \mathbb{R}^p}{\text{minimize}} \ \varphi(\boldsymbol{x}) + \textstyle\sum_{j=1}^k h_j(\boldsymbol{x}) \ , \tag{OPT-$k$}$$

where $\varphi$ is $L_\varphi$-smooth and each $h_j$ is proximal. The adaptive three operator splitting can be used to solve problems of this form by reducing them to a problem of the form (OPT) in an enlarged space. Consider consider the following problem in $\mathbb{R}^{k \times p}$,

$$\underset{\boldsymbol{X} \in \mathbb{R}^{k \times p}}{\text{minimize}} \underbrace{\varphi(\overline{\boldsymbol{X}})}_{=f(\boldsymbol{X})} + \underbrace{\textstyle\sum_{j=1}^k h_j(\boldsymbol{X}_j)}_{=h(\boldsymbol{X})} + \underbrace{\imath\{\boldsymbol{X}_1 = \cdots = \boldsymbol{X}_k\}}_{=g(\boldsymbol{X})} .$$

It is easy to see that this problem shares the same set of solutions as (OPT-$k$) with the correspondence $\boldsymbol{x} = \overline{\boldsymbol{X}}$, as the last term forces all the $\boldsymbol{X}_i$ terms to be equal. In this formulation the first term is smooth, the second term is proximal (variables in $h_i$ are separated) and the proximal operator of the last term is given by $\overline{\boldsymbol{X}}\,\mathbf{1}^T$. Hence Algorithm 1 can be applied to this problem. Deriving the complete algorithm is now merely a matter of replacing $f, g, h$ by its appropriate values in Algorithm 1 and is specified in Appendix B. The resulting adaptive algorithm seems to be new also in this extended formulation.

It is also possible to swap the definitions of $g$ and $h$, which results in a different algorithm that can be seen as an adaptive variant of the Generalized Forward-Backward splitting of Raguet et al. (2013). However, this formulation is less convenient for our purpose, since in this case the $h$ term is always an indicator function and so it would not be possible to apply variant 2 of our algorithm.

## 3 Analysis

In this section we provide a convergence rate analysis of the proposed method. We start by a characterization the set of fixed points of the algorithm, followed by a discussion on the gap function used to measure suboptimality. Finally, we present convergence rates for two different function classes. All proofs can be found in Appendix C.

**Assumption 1: Regularity.** We assume that $f$ is convex and $L_f$-smooth in $\mathbb{R}^p$ and that $g$ and $h$ are proper (i.e., have nonempty domain), lower semicontinuous (i.e., its sublevel sets are closed) convex functions. We note that lower semicontinuity is a weak form of continuity that allows extended-valued functions (such as the indicator function) over a closed domain.

**Assumption 2: Qualification conditions.** We assume the relative interior of $\operatorname{dom} g$ and $\operatorname{dom} h$ have a non-empty intersection. This is a weak and standard assumption that allows to relate the primal and dual optimal objective (see e.g. (Bertsekas, 2015, Proposition 5.3.8)), a property often referred to as strong or total duality.

In this section we will make use of the duality between the objective function in (OPT), which we will now denote $P$, and the following dual function, which we will denote $D$:

$$\begin{aligned} P(\boldsymbol{x}) &\overset{\text{def}}{=} f(\boldsymbol{x}) + g(\boldsymbol{x}) + h(\boldsymbol{x}) && (7) \\ D(\boldsymbol{u}) &\overset{\text{def}}{=} (f+g)^\star(-\boldsymbol{u}) + h^\star(\boldsymbol{u}) \ , && (8) \end{aligned}$$

where $^\star$ denotes the Fenchel conjugate. By strong duality, minimizing the primal and dual objectives can be combined into the search of a saddle point of the Lagrangian, defined as

$$\mathcal{L}(\boldsymbol{x}, \boldsymbol{u}) \overset{\text{def}}{=} f(\boldsymbol{x}) + g(\boldsymbol{x}) + \langle \boldsymbol{x}, \boldsymbol{u} \rangle - h^\star(\boldsymbol{u}) \ . \tag{9}$$

We recall that a saddle point of $\mathcal{L}$ is a pair $(\boldsymbol{x}^*, \boldsymbol{u}^*)$ such that the following is verified for any $(\boldsymbol{x}, \boldsymbol{u})$ in the domain (Hiriart-Urruty & Lemaréchal, 1993, §4.1):

$$\mathcal{L}(\boldsymbol{x}^*, \boldsymbol{u}) \leq \mathcal{L}(\boldsymbol{x}, \boldsymbol{u}^*)\ . \tag{10}$$

A consequence of strong duality is the equivalence between the saddle points of $\mathcal{L}$ and the minimizers of the primal and dual objectives. More precisely, if $(\boldsymbol{x}^*, \boldsymbol{u}^*)$ is a saddle point of $\mathcal{L}$, then $\boldsymbol{x}^*$ is a minimizer of $P$ and $\boldsymbol{u}^*$ is a minimizer of $D$. Likewise, a pair of minimizers of the primal and dual objectives form a saddle point of $\mathcal{L}$.

### 3.1 Fixed point characterization

A common first step in the analysis of optimization methods is the study of its set of fixed or stationary points. While this does not necessarily imply convergence, knowing which elements will be left invariant by the method improves our understanding and is a stepping stone for further analysis. We will show that the set of fixed points of the algorithm has a particularly simple and elegant structure: the Cartesian product of primal and dual solutions.

For the purpose of analysis it will be useful to express Algorithm 1 as an iteration of the form, $(\boldsymbol{z}_{t+1}, \boldsymbol{u}_{t+1}) = \boldsymbol{T}_{\gamma_t}(\boldsymbol{z}_t, \boldsymbol{u}_t)$, where the operator $\boldsymbol{T}_\gamma$ is defined as

$$\boldsymbol{T}_\gamma(\boldsymbol{z}, \boldsymbol{u}) \overset{\text{def}}{=} (\boldsymbol{z}^+, \boldsymbol{u}^+),\ \text{with} \tag{11}$$
$$\begin{cases} \boldsymbol{z}^+ = \mathbf{prox}_{\gamma h}(\boldsymbol{x}(\boldsymbol{z}, \boldsymbol{u}) + \gamma \boldsymbol{u}) \\ \boldsymbol{u}^+ = \boldsymbol{u} + (\boldsymbol{x}(\boldsymbol{z}, \boldsymbol{u}) - \boldsymbol{z}^+)/\gamma \\ \text{with}\ \ \boldsymbol{x}(\boldsymbol{z}, \boldsymbol{u}) = \mathbf{prox}_{\gamma g}(\boldsymbol{z} - \gamma(\boldsymbol{u} + \nabla f(\boldsymbol{z})))\ . \end{cases}$$

The following theorem characterizes the set of fixed points of this operator, denoted $\mathrm{Fix}(\boldsymbol{T}_\gamma)$.

**Theorem 1.** *Let $\mathcal{P}^*$ denote the set of minimizers of the primal objective (7) and $\mathcal{D}^*$ the set of minimizers of the dual objective (8). Then the set of fixed points of $\boldsymbol{T}_\gamma$ is given by*

$$\mathrm{Fix}(\boldsymbol{T}_\gamma) = \mathcal{P}^* \times \mathcal{D}^*\ . \tag{12}$$

### 3.2 Gap function

The progress of optimization methods is commonly measured in terms of a gap or merit function that is zero at optimum and nonzero otherwise. An appropriate gap function for many first-order methods is the suboptimality of the primal objective, i.e., $P(\boldsymbol{x}_t) - P(\boldsymbol{x}^*)$, where $\boldsymbol{x}^*$ is a minimizer of the primal objective. However, this is not an appropriate suboptimality measure for this algorithm, as $P(\boldsymbol{x}_t)$ might be $+\infty$, for example when $h$ is an indicator function.

Davis & Yin (2015) avoid the issue by either evaluating $h$ at a different point than $g$ (Davis & Yin, 2015, Corollary D.5.1) or assuming Lipschitz continuity of one of the proximal terms (Davis & Yin, 2015, Corollary D.5.2).

In this work we take an alternative approach, and instead use the following *saddle point suboptimality* criterion to measure the progress of our algorithm:

$$\mathcal{L}(\boldsymbol{x}_{t+1}, \boldsymbol{u}) - \mathcal{L}(\boldsymbol{x}, \boldsymbol{u}_{t+1})\ . \tag{13}$$

From the definition of saddle point in Eq. (10), this criterion is non-positive for all $(\boldsymbol{x}, \boldsymbol{u})$ if and only if $(\boldsymbol{x}_{t+1}, \boldsymbol{u}_{t+1})$ is a saddle point, and is so an appropriate suboptimality criterion. Furthermore, contrary to the primal objective function, this is defined for all iterates without further assumptions. Finally, we mention that this criteria has been previously used in the analysis of primal-dual methods, see e.g., Chambolle & Pock (2016; 2015) and Gidel et al. (2017) for a discussion of saddle point gap functions.

This suboptimality criteria can also be related to the primal and dual gap, as minimizing (13) over $\boldsymbol{x}$ and maximizing over $\boldsymbol{u}$ one recovers the primal-dual gap $P(\boldsymbol{x}_t) - D(\boldsymbol{u}_t)$ by definition of Fenchel conjugate.

### 3.3 Sublinear convergence

The following theorem gives a sublinear convergence rate for Algorithm 1. This convergence will be given in terms of the weighted ergodic (i.e., averaged) sequence. Denoting by $s_t$ the sum of all step-sizes up to iteration $t$, i.e., $s_t \overset{\text{def}}{=} \sum_{i=0}^t \gamma_t$, the ergodic iterates $\overline{\boldsymbol{x}}_t$ and $\overline{\boldsymbol{u}}_t$ are defined as

$$\overline{\boldsymbol{x}}_t \overset{\text{def}}{=} \Big(\sum_{i=0}^{t-1} \gamma_i \boldsymbol{x}_{i+1}\Big)/s_t\ ,\ \ \overline{\boldsymbol{u}}_t \overset{\text{def}}{=} \Big(\sum_{i=0}^{t-1} \gamma_i \boldsymbol{u}_{i+1}\Big)/s_t\ . \tag{14}$$

While results in this subsection will be stated in terms of this ergodic sequence, in practice the last iterate gives most often a better empirical convergence, see e.g., (Chambolle & Pock, 2015, §7.2.1). For a more theoretically-sound algorithm, one can compare the objective at the ergodic and last iterate, and return the one with smallest objective.

**Theorem 2** (sublinear convergence rate)**.** *For every $t \geq 0$ and any $(\boldsymbol{x}, \boldsymbol{u})$ in the domain of $\mathcal{L}$ we have the following convergence rate for Algorithm 1 (both variants):*

$$\mathcal{L}(\overline{\boldsymbol{x}}_{t+1}, \boldsymbol{u}) - \mathcal{L}(\boldsymbol{x}, \overline{\boldsymbol{u}}_{t+1}) \leq \frac{\|\boldsymbol{z}_0 - \boldsymbol{x}\|^2 + \gamma_0^2 \|\boldsymbol{u}_0 - \boldsymbol{u}\|^2}{2 s_t}\ .$$

**Convergence in terms of function value suboptimality.** The previous result gives an $\mathcal{O}(1/t)$ convergence rate for arbitrary convex functions in terms of the saddle point suboptimality. As we have discussed previously, it is not possible to obtain similar rates in terms of the function suboptimality without further assumptions. We will now show that it is sufficient to assume Lipschitz continuity on $h$ to derive from the previous theorem a convergence rate in terms of the primal function suboptimality.

The following Corollary can be obtained by optimizing with respect to $\boldsymbol{u}$ the bound in the previous theorem and using the Lipschitz continuity to bound $\|\boldsymbol{u}_0-\boldsymbol{u}\|^2$. This gives an $\mathcal{O}(1/t)$ convergence rate for the primal function suboptimality, roughly matching that of Davis & Yin (2015, Corollary D.5.2) for the non adaptive variant:

**Corollary 1.** *Let $h$ be $\beta_h$-Lipschitz. Then, we have the following rate for the weighted ergodic iterate*

$$P(\overline{\boldsymbol{x}}_{t+1}) - P(\boldsymbol{x}^*) \leq \frac{\|\boldsymbol{z}_0 - \boldsymbol{x}^*\|^2 + 2\gamma_0^2(\|\boldsymbol{u}_0\|^2 + \beta_h^2)}{2s_t} \,.$$

### 3.4 Linear convergence

In this subsection we assume that $f$ is $\mu_f$-strongly convex and $h$ is $L_h$-smooth (with $0 < \mu_f, 0 < L_h < +\infty$). We denote by $\boldsymbol{x}^*$ the minimizer of the primal loss (unique by strong convexity of $P$) and by $\boldsymbol{u}^*$ the minimizer of the dual loss (also unique by strong convexity of $D$, consequence of the $L_h$-smoothness of $h$).

The convergence rates will be given in terms of the following quantities

$$\begin{aligned} \rho \overset{\text{def}}{=} \mu_f \min\{\gamma_0, \tau/L_f\} \,,\ \sigma \overset{\text{def}}{=} 1/(1+\gamma_0 L_h) \\ \xi \overset{\text{def}}{=} \mu_f/(\mu_f + L_h) \,. \end{aligned} \tag{15}$$

All these belong to the interval $(0, 1)$. Assuming $\gamma_0 \geq \tau/L_f$, then $\rho$ is the inverse of $f$'s condition number, a quantity that appears in the analysis of most gradient-based methods, while $\sigma$ and $\xi$ quantify the smoothness of $h$. Note that by strong convexity, $\gamma_0 < 1/\mu_f$ as otherwise the sufficient decrease condition would not succeed and so $\sigma \geq \xi$.

**Theorem 3.** *Let $\boldsymbol{x}_{t+1}, \boldsymbol{u}_{t+1}$ be the iterates produced by Algorithm 1 after t iterations. Then we have the following linear convergence for Variant 1 (V1) and Variant 2 (V2):*

$$V1: \|\boldsymbol{x}_{t+1} - \boldsymbol{x}^*\|^2 \leq \Big(1 - \min\big\{\rho, \sigma\big\}\Big)^{t+1} D_0 \tag{16}$$

$$V2: \|\boldsymbol{x}_{t+1} - \boldsymbol{x}^*\|^2 \leq \Big(1 - \min\big\{\rho, \xi, \tfrac{1}{2}\big\}\Big)^{t+1} E_0 \,, \tag{17}$$

*with $D_0 \overset{def}{=} 6\|\boldsymbol{z}_0 - \boldsymbol{x}^*\|^2 + \frac{6}{1-\sigma}\|\gamma_0(\boldsymbol{u}_0 - \boldsymbol{u}^*)\|^2$ and $E_0 \overset{def}{=} 6\|\boldsymbol{z}_0 - \boldsymbol{x}^*\|^2 + \frac{6}{1-\xi}\|\gamma_0(\boldsymbol{u}_0 - \boldsymbol{u}^*)\|^2$.*

**Discussion.** For $\gamma_t = 1/L_f$, the sufficient decrease condition is always verified and the algorithm can be run with $\tau = 1$. In this case, Variant 1 of Algorithm 1 defaults to TOS, and we can compare the obtained rates with those in (Davis & Yin, 2015).

While the sublinear convergence rate obtained in Corollary 1 roughly matches the rate obtained in (Davis & Yin (2015, Corollary D.5.2, see our Appendix C.5)), the linear convergence rates are instead significantly different. In (Davis & Yin, 2015, Theorem D.6.6), the authors obtain linear convergence rate that, after optimizing for all parameters (see Appendix C.5), yields a rate of $\rho\sigma^2$, which is *strictly worse* than the $\min\{\rho, \sigma\}$ rate that we obtained. This difference can be quite large, e.g., for $\rho = \sigma$ this becomes $\rho$ versus $\rho^3$.

## 4 Applications

### 4.1 Learning with Multiple Penalties

In this subsection we discuss how some penalties with costly to compute proximal operator can be decomposed as a sum of proximal terms and so fall within the current framework. The exact expression of the proximal operators is given in Appendix D.

**Group lasso with overlap.** Jacob et al. (2009) generalized group $\ell_1$ norm by allowing each variable to belong to more than one group, thereby introducing overlaps among groups and allowing for more complex prior knowledge on the structure. For a set of subindices $\mathcal{G}$ which we will call groups, this penalty is defined as $\|\boldsymbol{x}\|_{\mathcal{G}} = \sum_{G \in \mathcal{G}} \|[\boldsymbol{x}]_G\|_2$. If each coefficient is at most in $s$ groups, then $\mathcal{G}$ can be decomposed as $\mathcal{G} = \mathcal{G}_1 \cup \ldots \cup \mathcal{G}_s$, where the $\mathcal{G}_i$ are disjoint. This allows to express the group lasso with overlap as a sum of $s$ non-overlapping group lasso penalties, for which the proximal operator has a closed form expression.

**Multidimensional total variation.** For the task of image restoration and denoising it is common to consider a regularization term in the form of a total variation regularizer. For an image $\boldsymbol{x}$ of size $p \times q$, the 2-dimensional total variation norm $\|\boldsymbol{X}\|_{\text{TV}}$ is defined as

$$\underbrace{\sum_{i=1}^{p}\sum_{j=1}^{q-1}|\boldsymbol{X}_{i,j+1} - \boldsymbol{X}_{i,j}|}_{=g(\boldsymbol{X})} + \underbrace{\sum_{j=1}^{q}\sum_{ji=1}^{p-1}|\boldsymbol{X}_{i+1,j} - \boldsymbol{X}_{i,j}|}_{=h(\boldsymbol{X})} \,.$$

From here we recognize that $g$ and $h$ are fused lasso (also known as 1D-total variation) penalties acting on the columns and rows of $\boldsymbol{X}$ respectively. Efficient methods to evaluate the proximal operator of the fused lasso penalty have been developed by Condat (2013a); Johnson (2013).

**Isotonic and nearly isotonic penalties.** In some applications there exists a natural ordering between variables: $\boldsymbol{x}_1 \leq \boldsymbol{x}_2 \leq \cdots \leq \boldsymbol{x}_p$. This can be enforced through constraints, and the projection onto these is known as isotonic regression (Best & Chakravarti, 1990). The indicator function over the set of constraints can also be split into a sum of two terms as follows

$$\begin{aligned}&\imath\{\boldsymbol{x}_1 \leq \boldsymbol{x}_2 \leq \boldsymbol{x}_3 \leq \boldsymbol{x}_4 \leq \cdots\} \\ &= \underbrace{\imath\{\boldsymbol{x}_1 \leq \boldsymbol{x}_2; \boldsymbol{x}_3 \leq \boldsymbol{x}_4; \cdots\}}_{=g(\boldsymbol{x})} + \underbrace{\imath\{\boldsymbol{x}_2 \leq \boldsymbol{x}_3; \boldsymbol{x}_4 \leq \boldsymbol{x}_5; \cdots\}}_{=h(\boldsymbol{x})},\end{aligned} \tag{18}$$

where each term has a closed form proximal operator (see Appendix D.2).

In cases in which the variables are only "mostly" nondecreasing, the constraint can be relaxed via a nearly-isotonic penalty (Tibshirani et al., 2011) of the form $\sum_{i=1}^{p-1} \max\{\boldsymbol{x}_i - \boldsymbol{x}_{i+1}, 0\}$, in which only the non-increasing coefficients are penalized. This penalty can be split the same way as the isotonic constraints above.

**$\ell_1$ trend filtering.** This penalty is defined by the absolute value of the second order differences and promotes piecewise-linear coefficients (Kim et al., 2009). It is defined as $\|\boldsymbol{x}\|_{\text{TF}} \overset{\text{def}}{=} \sum_{i=1}^{p-2} |\boldsymbol{x}_i - 2\boldsymbol{x}_{i+1} + \boldsymbol{x}_{i+2}|$. We can split this sum into 3 proximal terms such that the resulting terms: the $j$-th term contains the factors for which $i$ is congruent to 3 modulo $j$.

**Constraints over doubly stochastic matrices.** Optimization problems with constraints on the set of doubly stochastic matrices appear in many convex relaxations of combinatorial problems such as seriation (Fogel et al., 2013), quadratic assignment (Lawler, 1963) and graph matching (Conte et al., 2004; Aflalo et al., 2015). The set of double stochastic matrices is composed of square matrices with nonnegative entries, each of whose rows and columns sum to 1, i.e., $\{\boldsymbol{X}^T\mathbf{1} = \mathbf{1}, \boldsymbol{X}\mathbf{1} = \mathbf{1}, \boldsymbol{X} \geq \mathbf{0}\}$. The indicator function over this set can be split as

$$\underbrace{\imath\{\boldsymbol{X}^T\mathbf{1} = \mathbf{1}, \boldsymbol{X}\mathbf{1} = \mathbf{1}\}}_{=g(\boldsymbol{X})} + \underbrace{\imath\{\boldsymbol{X} \geq \mathbf{0}\}}_{=h(\boldsymbol{X})} \quad , \tag{19}$$

and the projection onto both sets is available in closed form (Lu et al., 2016, §4.3).

**Dispersive sparsity.** In some applications it is desirable to encourage dispersion of the sparse coefficients. This happens for example in the modeling of neural spiking, as the spikes are assumed to be spaced across time (Hegde et al., 2009). El Halabi & Cevher (2015) showed that this behavior can be promoted by considering a penalty of the form $\|\boldsymbol{x}\|_1 + \imath\{\boldsymbol{B}|\boldsymbol{x}| \leq c\}$ for a matrix $\boldsymbol{B}$ and some predefined constant $c$, where $|\boldsymbol{x}|$ denotes the component-wise absolute value. This penalty can be split into three proximal terms by the introduction of a dummy variable $\boldsymbol{z}$, resulting in $\|\boldsymbol{x}\|_1 + \imath\{\boldsymbol{B}\boldsymbol{z} \leq \boldsymbol{c}\} + \imath\{\boldsymbol{z} = |\boldsymbol{x}|\}$.

**Combination by addition**. A popular method to promote the joint behavior of different penalties is by adding them. This has been used to successfully learn models with sparse and nonnegative coefficients (Yuan & Lin, 2007), sparse and low rank matrices (Richard et al., 2012), sparse and piecewise constant (Gramfort et al., 2013), to name a few.

### 4.2 Benchmarks

In this subsection we provide an empirical evaluation of the proposed method. Due to space constraints we only give here a high level overview, deferring details as well as an extended set of experiments to Appendix E. We consider the following methods:

- The proposed Adaptive TOS method (Algorithm 1), in its both variants.
- The TOS method of Davis & Yin (2015), with step-sizes $1/L_f$ and $1.99/L_f$ (the method is convergent for step-sizes $< 2/L_f$).
- The PDHG or Condat-Vũ algorithm (Condat, 2013b; Vũ, 2013), with step-sizes $\tau$ and $\beta/\tau$, where $\beta$ was chosen as the one giving the best overall performance over the grid $\beta = 0.9, 0.5, 0.1$ (giving it a slight advantage).
- The adaptive PDHG of Malitsky & Pock (2018), with step-size hyperparameter $\beta$ chosen by the same technique as for PDHG.
- The averaged operator line search method of Giselsson et al. (2016) combined with TOS, named TOS-AOLS.

We compared these methods on 4 different problems and show the results in Figure 1. In the first row we show the benchmarks on a logistic regression problem with overlapping group lasso penalty that we apply to two text datasets (`RCV1` and `real-sim`). Subfigures A and C were run with the regularization parameter chosen to give 50% of sparsity, while B, E are run with higher levels of sparsity, chosen to give 5% of sparsity.

In the second and third row we considered a battery of inverse problems with different penalties on synthetic datasets. These consists of a least squares (G, H, I, J) or logistic regression (rest) smooth term and 4 different penalties specified in the title of each plot (overlapping group lasso, total variation, trace norm $\ell_1$ and nearly isotonic, see Appendix E for a precise formulation). For each problem, we show 2 different benchmarks, corresponding to the low and high regularization regimes (denoted low reg and high reg). We comment on a few trends from Fig. 1:

- **Best performing method**. On 10 out of 12 experiments, the adaptive TOS algorithm (Variant 2) is the best performing method, and in the other cases (E, H) its performance is roughly the same as that of the best performing

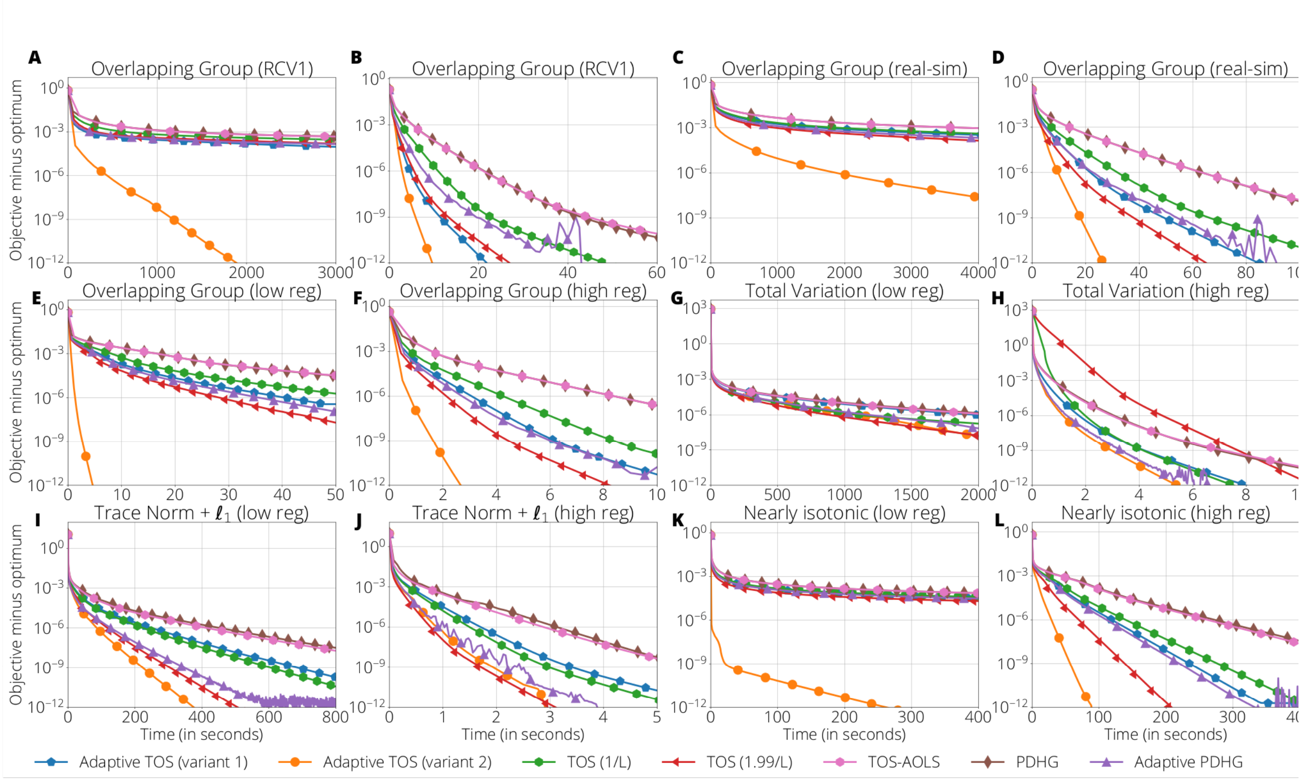


Figure 1: **Comparison of different proximal splitting methods**. The *top row* gives result for two real datasets with an overlapping group lasso penalty. The *second and third row* show results on synthetic datasets for 4 different penalties: overlapping group lasso (E, F), 2-dimensional total variation (G, H), trace norm + $\ell_1$ (I, J) and nearly isotonic (K, L). The Adaptive TOS (Variant 2, i.e., with growing step-sizes) is the best performing method on 10 out of 12 experiments, and roughly equivalent to the best performing method in the other 2 cases (G, J).

method. In contrast, on 3 instances (A, I, K) it is an order of magnitude faster than the next method.

- **Low vs high regularization regime.** The advantage of the adaptive method is highly correlated with the amount of regularization: in the low regularization regime, on 3 out of 6 the adaptive TOS is an order of magnitude faster then the fixed step-size method, while in the high regularization regime the difference shrinks and in the same problems is never more than a factor 2.6. Note also that there is in general a large computing time difference between the low and high regularization regime.

- **Uniform curvature**. The problems (G, H, I, J) in Fig. 1 use as smooth term a quadratic loss (i.e., constant Hessian), while the other methods use a logistic loss (non-constant Hessian). This suggests that the use of the adaptive step-size strategy (and in particular Variant 2 with its growing step-size) is more beneficial for smooth terms with non-uniform curvature.

## 5 Conclusion and Future Work

We have presented and analyzed a novel adaptive step-size method to solve optimization problems consisting in a sum of a smooth term accessed through its gradient and two or more potentially non-smooth terms accessed through their proximal operator. The method does not rely on any step-size hyperparameter (except for an initial estimate) and extensive empirical evaluation has showed computational gains on a variety of problems. We mention two possible extensions of this work.

First, existing convergence results fail to fully explain their surprisingly good empirical convergence. To the best of our knowledge, no work so far has derived linear convergence rates in absence of strong convexity and smoothness of one of the proximal terms for these methods (as is however empirically observed, see e.g. Figure 1).

Second, it is an open question whether this or other adaptive step-size methods can be accelerated, as is the case of proximal gradient descent, which admits the adaptive FISTA variant (Beck & Teboulle, 2009).

## Acknowledgements

FP would like to thank Wotao Yin and Walaa Moursi for fruitful discussions, Michael Eickenberg for feedback on the manuscript and Patrick R. Johnstone for reporting a typo in the paper.

FP is funded through the European Union's Horizon 2020 research and innovation programme under the Marie Sklodorowska-Curie grant agreement 748900. Computing time on was donated by Amazon through the program "AWS Cloud Credits for Research".

GG is funded by the Canada Excellence Research Chair in Data Science for Realtime Decision-making and by the NSERC Discovery Grant RGPIN-2017-06936.

## References

Aflalo, Y., Bronstein, A., and Kimmel, R. On convex relaxation of graph isomorphism. *Proceedings of the National Academy of Sciences*, 2015.

Agarwal, A., Negahban, S., and Wainwright, M. J. Fast global convergence rates of gradient methods for high-dimensional statistical recovery. In *Advances in Neural Information Processing Systems*, 2010.

Barbero, Á. and Sra, S. Modular proximal optimization for multidimensional total-variation regularization. *preprint arXiv:1411.0589*, 2014.

Bauschke, H. H. and Combettes, P. L. *Convex analysis and monotone operator theory in Hilbert spaces*. Springer Science & Business Media, 2017.

Bauschke, H. H., Boţ, R. I., Hare, W. L., and Moursi, W. M. Attouch–Théra duality revisited: paramonotonicity and operator splitting. *Journal of Approximation Theory*, 2012.

Beck, A. and Teboulle, M. Gradient-based algorithms with applications to signal recovery. *Convex Optimization in Signal Processing and Communications*, 2009.

Bertsekas, D. P. *Convex optimization algorithms*. Athena Scientific Belmont, 2015.

Best, M. J. and Chakravarti, N. Active set algorithms for isotonic regression; A unifying framework. *Mathematical Programming*, 1990.

Boyd, S., Parikh, N., Chu, E., Peleato, B., and Eckstein, J. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends in Machine Learning*, 2011.

Cai, J.-F., Candès, E. J., and Shen, Z. A singular value thresholding algorithm for matrix completion. *SIAM Journal on Optimization*, 2010.

Candès, E. J., Romberg, J., and Tao, T. Robust uncertainty principles: exact signal reconstruction from highly incomplete frequency information. *IEEE Transactions on information theory*, 2006.

Chambolle, A. and Pock, T. On the ergodic convergence rates of a first-order primal–dual algorithm. *Mathematical Programming*, 2015.

Chambolle, A. and Pock, T. An introduction to continuous optimization for imaging. *Acta Numerica*, 2016.

Combettes, P. L. and Pesquet, J.-C. Proximal splitting methods in signal processing. In *Fixed-point algorithms for inverse problems in science and engineering*. Springer, 2011.

Condat, L. A direct algorithm for 1D total variation denoising. *IEEE Signal Processing Letters*, 2013a.

Condat, L. A primal–dual splitting method for convex optimization involving Lipschitzian, proximable and linear composite terms. *Journal of Optimization Theory and Applications*, 2013b.

Conte, D., Foggia, P., Sansone, C., and Vento, M. Thirty years of graph matching in pattern recognition. *International journal of pattern recognition and artificial intelligence*, 2004.

Davis, D. and Yin, W. A three-operator splitting scheme and its optimization applications. *preprint arXiv:1504.01032v1*, 2015.

Davis, D. and Yin, W. A three-operator splitting scheme and its optimization applications. *Set-Valued and Variational Analysis*, 2017.

El Halabi, M. and Cevher, V. A totally unimodular view of structured sparsity. In *Proceedings of the 18th International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2015.

Fogel, F., Jenatton, R., Bach, F., and d'Aspremont, A. Convex relaxations for permutation problems. In *Advances in Neural Information Processing Systems*, 2013.

Gabay, D. and Mercier, B. A dual algorithm for the solution of nonlinear variational problems via finite element approximation. *Computers & Mathematics with Applications*, 1976.

Gidel, G., Jebara, T., and Lacoste-Julien, S. Frank-Wolfe Algorithms for Saddle Point Problems. In *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*, 2017.

Giselsson, P., Fält, M., and Boyd, S. Line search for averaged operator iteration. In *IEEE 55th Conference on Decision and Control (CDC)*. IEEE, 2016.

Glowinski, R. and Marroco, A. Sur l'approximation, par éléments finis d'ordre un, et la résolution, par pénalisation-dualité d'une classe de problèmes de Dirichlet non linéaires. *Revue française d'automatique, informatique, recherche opérationnelle*, 1975.

Gramfort, A., Thirion, B., and Varoquaux, G. Identifying predictive regions from fMRI with TV-L1 prior. In *International Workshop on Pattern Recognition in Neuroimaging*. IEEE, 2013.

Hegde, C., Duarte, M. F., and Cevher, V. Compressive sensing recovery of spike trains using a structured sparsity model. In *SPARS'09-Signal Processing with Adaptive Sparse Structured Representations*, 2009.

Hiriart-Urruty, J.-B. and Lemaréchal, C. *Convex analysis and minimization algorithms I: Fundamentals*. Springer science & business media, 1993.

Iusem, A. N. On Some Properties of Generalized Proximal Point Methods for Variational Inequalities. *Journal of Optimization Theory and Applications*, 1998.

Jacob, L., Obozinski, G., and Vert, J.-P. Group lasso with overlap and graph lasso. In *Proceedings of the 26th annual international conference on machine learning*. ACM, 2009.

Johnson, N. A dynamic programming algorithm for the fused lasso and $L_0$-segmentation. *Journal of Computational and Graphical Statistics*, 2013.

Kim, S.-J., Koh, K., Boyd, S., and Gorinevsky, D. $\ell_1$ trend filtering. *SIAM review*, 2009.

Lawler, E. L. The quadratic assignment problem. *Management science*, 1963.

Lewis, D. D., Yang, Y., Rose, T. G., and Li, F. RCV1: A new benchmark collection for text categorization research. *Journal of machine learning research*, 2004.

Lions, P.-L. and Mercier, B. Splitting algorithms for the sum of two nonlinear operators. *SIAM Journal on Numerical Analysis*, 1979.

Lu, Y., Huang, K., and Liu, C.-L. A fast projected fixed-point algorithm for large graph matching. *Pattern Recognition*, 2016.

Malitsky, Y. and Pock, T. A first-order primal-dual algorithm with linesearch. *SIAM Journal on Optimization*, 2018.

Nesterov, Y. et al. Gradient methods for minimizing composite objective function. *Mathematical Programming*, 2013.

Nocedal, J. and Wright, S. *Numerical optimization*. Springer Science & Business Media, 2006.

Parikh, N. and Boyd, S. Proximal algorithms. *Foundations and Trends in optimization*, 2013.

Pedregosa, F. C-OPT: composite optimization in Python. 2018. doi: 10.5281/zenodo.1283339. URL `http://copt.bianp.net`.

Raguet, H., Fadili, J., and Peyré, G. A generalized forward-backward splitting. *SIAM Journal on Imaging Sciences*, 2013.

Richard, E., Savalle, P.-a., and Vayatis, N. Estimation of Simultaneously Sparse and Low Rank Matrices. In *Proceedings of the 29th International Conference on Machine Learning*, 2012.

Richtárik, P. and Takáč, M. Iteration complexity of randomized block-coordinate descent methods for minimizing a composite function. *Mathematical Programming*, 2014.

Rockafellar, R. T. Convex analysis, 1997.

Rockafellar, R. T. and Wets, R. J.-B. *Variational analysis*. Springer Science & Business Media, 1998.

Rudin, L. I., Osher, S., and Fatemi, E. Nonlinear total variation based noise removal algorithms. *Physica D: Nonlinear Phenomena*, 1992.

Tibshirani, R. J., Hoefling, H., and Tibshirani, R. Nearly-isotonic regression. *Technometrics*, 2011.

Vũ, B. C. A splitting algorithm for dual monotone inclusions involving cocoercive operators. *Advances in Computational Mathematics*, 2013.

Yan, M. A New Primal–Dual Algorithm for Minimizing the Sum of Three Functions with a Linear Operator. *Journal of Scientific Computing*, 2018.

Yuan, M. and Lin, Y. Model selection and estimation in regression with grouped variables. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 2006.

Yuan, M. and Lin, Y. On the non-negative garrotte estimator. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 2007.