Optimization of Graph Total Variation via Active-Set-based Combinatorial Reconditioning

Zhenzhang Ye
TU Munich
zhenzhang.ye@tum.de

Thomas Möllenhoff
TU Munich
thomas.moellenhoff@tum.de

Tao Wu
TU Munich
tao.wu@tum.de

Daniel Cremers
TU Munich
cremers@tum.de

Abstract

Structured convex optimization on weighted graphs finds numerous applications in machine learning and computer vision. In this work, we propose a novel adaptive preconditioning strategy for proximal algorithms on this problem class. Our preconditioner is driven by a sharp analysis of the local linear convergence rate depending on the "active set" at the current iterate. We show that nested-forest decomposition of the inactive edges yields a guaranteed local linear convergence rate. Further, we propose a practical greedy heuristic which realizes such nested decompositions and show in several numerical experiments that our reconditioning strategy, when applied to proximal gradient or primal-dual hybrid gradient algorithm, achieves competitive performances. Our results suggest that local convergence analysis can serve as a guideline for selecting variable metrics in proximal algorithms.

1 Introduction

Preconditioning, as a way of transforming a difficult linear system into an easier one to solve, enjoys a rich history. Recently, *proximal algorithms* (Combettes and Pesquet, 2011; Parikh and Boyd, 2013; Chambolle and Pock, 2016a) have received a surge of popularity in solving structured non-smooth convex optimization problems. Unlike in the case of linear systems, putting forward a satisfactory theory and implementation of preconditioning in the general non-smooth setting remains an unsolved challenge (Pock and Chambolle, 2011; Giselsson and Boyd, 2014a,b; Lee et al., 2014; Bredies and Sun, 2015; Fougner and Boyd, 2015; Giselsson and Boyd, 2015; Becker et al., 2018). This is mainly due to two obstacles:

(i) The non-linear dynamics of proximal algorithms, as well as the geometry of the non-smooth energy are more involved than in the quadratic case. A precise characterization of the convergence behavior, which could guide the proper choice of metric (preconditioner), is challenging.

(ii) In cases where the proper choice of metric is clear, non-diagonal preconditioners typically make the proximal operators in the algorithm much more expensive to evaluate. While reducing the number of outer iterations, each inner iteration could be even of similar complexity as the original problem (Lee et al., 2014).

In this vein, numerous efforts have been devoted to achieve a better understanding of the dynamics of proximal algorithms (see, e.g., Nishihara et al. (2015); Garrigos et al. (2017)), and exploring scenarios where non-diagonally scaled proximal mappings are still efficient to evaluate (Friedlander and Goh, 2017; Becker and Fadili, 2012; Becker et al., 2018).

In this paper we take a novel perspective, circumventing issue **(i)** by resorting to the local convergence analysis. This does not yield provable guarantees on the global iteration complexity. Nevertheless, we show empirically that the local analysis yield an improvement long before the local linear convergence regime is entered (see Fig. 3).To overcome difficulty **(ii)** we consider structured convex problems on weighted graphs, where metrics based on tree decompositions are amenable to efficient proximal evaluation by recent message-passing algorithms (Kolmogorov et al., 2016). Specifically, given an undirected weighted graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \omega)$, whose edges are weighted by a function $\omega : \mathcal{E} \to \mathbb{R}_{>0}$, we consider the structured convex optimization on $\mathcal{G}$:

$$\min_{u \in \mathbb{R}^{\mathcal{V}}} G(u) + \mathrm{TV}_{\mathcal{G}}(u), \tag{1}$$

where $\mathrm{TV}_{\mathcal{G}}$ is the graph total variation

$$\mathrm{TV}_{\mathcal{G}}(u) = \sum_{e=(i,j) \in \mathcal{E}} \omega_e \, |u_i - u_j|. \tag{2}$$

Proceedings of the 23$^{\text{rd}}$International Conference on Artificial Intelligence and Statistics (AISTATS) 2020, Palermo, Italy. PMLR: Volume 108. Copyright 2020 by the author(s).

The function $G : \mathbb{R}^{\mathcal{V}} \to \mathbb{R} \cup \{+\infty\}$ is assumed to be proper, lower semi-continuous and convex.

We define the vertex-to-edge map $K : \mathbb{R}^{\mathcal{V}} \to \mathbb{R}^{\mathcal{E}}$ by

$$K = \operatorname{diag}(\omega)\nabla,$$

where ∇ is the (transposed) incidence matrix of $\mathcal{G}$, i.e.,

$$(\nabla u)_e = u_i - u_j, \quad \forall e = (i, j) \in \mathcal{E},$$

with arbitrarily fixed orientation. With this notation we can succinctly write $\mathrm{TV}_{\mathcal{G}}(u) = \|Ku\|_1$.

Problems of form (1) are relevant in computer vision (Gilboa and Osher, 2008; Lou et al., 2010; Chambolle and Pock, 2011; Newcombe et al., 2011), unsupervised and transductive learning (Hein and Setzer, 2011; Hein et al., 2013; Bresson et al., 2013; Garcia-Cardona et al., 2014), collaborative filtering (Benzi et al., 2016) and clustering (Garcia-Cardona et al., 2014).

For separable convex $G(u) = \sum_{i \in \mathcal{V}} g_i(u_i)$, problem (1) can be efficiently solved in polynomial time by parametric max-flow methods (Chambolle and Darbon, 2009; Hochbaum, 2001). To handle non-separable but differentiable G, Xin et al. (2014) proposed a (primal) proximal gradient iteration, reducing (1) to a sequence of separable problems which are solved by parametric max-flow. For problems on regular grids, Condat (2013); Barbero and Sra (2014); Kolmogorov et al. (2016) proposed a splitting into *chains*, leading to 1D total variation subproblems which can be solved efficiently. Kumar and Bach (2017) proposed an active-set method for submodular minimization (which includes the graph total variation as a special case), which is different from the active-set strategy pursued here. Landrieu and Obozinski (2017); Raguet and Landrieu (2018) recently proposed a fast method for graph total variation by assuming that the solution is piecewise constant and refining that partition by solving a sequence of max-flow problems. Closely related to the present approach are projected Newton methods (Schmidt et al., 2012), which have also been applied to the total variation (Barbero and Sra, 2011).

In contrast, the main focus of this paper is to advance the understanding of preconditioning in proximal algorithms. We consider two types of algorithms:

(1) (Dual) proximal gradient (PG). Assume G^* is C^2 such that $l_{G^*} I \preceq \nabla^2 G^*(\cdot) \preceq L_{G^*} I$ for some constants $l_{G^*}, L_{G^*} > 0$. Based on the (Fenchel) dual formulation of (1), written

$$\min_{p \in \mathbb{R}^{\mathcal{E}}} G^*(-K^\top p) + \delta\{\|p\|_\infty \leq 1\}, \tag{3}$$

one can apply the proximal (or projected) gradient:

$$\begin{aligned} p^{k+1} = \arg\min_{p \in \mathbb{R}^{\mathcal{E}}} & - \left\langle K \nabla G^*(-K^\top p^k), p \right\rangle \\ & + \delta\{\|p\|_\infty \leq 1\} + \frac{t}{2}\|p - p^k\|_{T_k}^2. \end{aligned} \tag{4}$$

Here $T_k \in \mathbb{R}^{|\mathcal{E}| \times |\mathcal{E}|}$ is a symmetric positive definite matrix which induces a scaled norm $\|\cdot\|_{T_k}$ defined by $\|u\|_{T_k}^2 = \langle u, u \rangle_{T_k} = u^\top T_k u$.

(2) Primal-dual hybrid gradient (PDHG). Another equivalent formulation of (1) is the following convex-concave saddle-point problem:

$$\min_{u \in \mathbb{R}^{\mathcal{V}}} \max_{p \in \mathbb{R}^{\mathcal{E}}} \ \langle Ku, p \rangle + G(u) - \delta\{\|p\|_\infty \leq 1\}, \tag{5}$$

to which one can apply the primal-dual hybrid gradient (PDHG) algorithm:

$$u^{k+1} = \arg\min_{u \in \mathbb{R}^{\mathcal{V}}} \ G(u) + \langle p^k, Ku \rangle + \frac{s}{2}\|u - u^k\|^2, \tag{6}$$

$$\begin{aligned} p^{k+1} = \arg\min_{p \in \mathbb{R}^{\mathcal{E}}} & \ - \left\langle K(2u^{k+1} - u^k), p \right\rangle \\ & + \delta\{\|p\|_\infty \leq 1\} + \frac{t}{2}\|p - p^k\|_{T_k}^2. \end{aligned} \tag{7}$$

1.1 Related work on preconditioning

The (vanilla) PG and PDHG (typically with $T_k \equiv I$), as special instances of proximal algorithms, are widely applied in convex optimization – we refer to Combettes and Pesquet (2011); Parikh and Boyd (2013); Chambolle and Pock (2016a) for the surveys containing relevant historical accounts and interconnection of algorithms. Acceleration of these algorithms is of significant research and practical interests. To this end, momentum-based acceleration techniques, which are traced back to the seminal works by Nesterov (1983) and Polyak (1964), were recently developed for PG (Beck and Teboulle, 2009; Ochs et al., 2014) and PDHG (Chambolle and Pock, 2016b) and achieved impressive performances (Chambolle and Pock, 2016a).

In contrast to momentum methods, preconditioning techniques for proximal algorithms are less developed and understood, as previously discussed in the introduction. To clarify further, in the context of proximal methods there are roughly two separate streams of ideas referred to as preconditioning.

In the first one, the aim is to make the individual update steps in the algorithm easier while retaining a convergent method (Bredies and Sun, 2015; Chambolle and Pock, 2011). While making each iteration faster, the effect on the overall complexity is unclear.

The second line of works, aims at improving the theoretical convergence rate and thereby reducing the number of outer iterations, see Giselsson and Boyd (2014a,b, 2015). However, these works make very restrictive assumptions on the problem class and do not apply to our setting. A consensus among these works is to minimize the *(finite) condition number* $\kappa(T^{-1/2}K)$, which

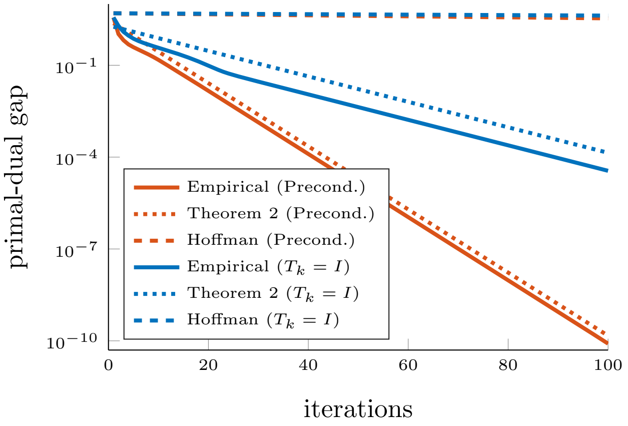

Figure 1: Local vs global analysis of the linear convergence of the PG iteration (4). The local linear rate sharply matches the observed convergence behaviour, while the global rate based on Hoffman's bound is not informative. We guide the construction of our preconditioner based on the local convergence theory.

is defined by

$$\kappa(\cdot) = \frac{\sigma_{\max}(\cdot)}{\sigma_{\min>0}(\cdot)}, \tag{8}$$

as a reasonable heuristic in practice. This was pursued for general problems in Pock and Chambolle (2011); Fougner and Boyd (2015); Diamond and Boyd (2017). Recently, Möllenhoff et al. (2018) proposed forest-structured preconditioners for $K = \text{diag}(\omega)\nabla$ which are provably optimal in terms of $\kappa(T^{-1/2}K)$.

2 Local convergence analysis

While the condition number $\kappa(T^{-1/2}K)$ has proven to be reasonable heuristic in practice, a more quantified connection between the convergence rate and the preconditioner T would be desirable.

For problems of form (1), global linear convergence of PG (4) can be established using Hoffman's bound (Hoffman, 1952; Klatte and Thiere, 1995; Necoara et al., 2015; Karimi et al., 2016). However, the linear rate obtained from that bound is mainly of theoretical interest, as it does not really inform us about the practical performance of the method but rather gives a (weak) upper bound. Secondly, Hoffman's bound is an inherently combinatorial expression that is very challenging to compute even for small problem instances.

Instead, we aim to choose the preconditioner to improve the local convergence behaviour of the method. It turns out that for a wide range of *partly smooth* functions the local dynamics of the PG, PDHG and accelerated variants thereof are well understood, see Liang et al. (2014, 2017, 2018). This will be a basis for our theory. In Fig. 1 we show the linear rate predicted by Hoffman's bound to the local rate on a small 4×3 grid graph for which Hoffman's bound is still tractable to compute. As discussed above, the global rate by Hoffman's bound is not informative. The local analysis we present in Theorem 2 below (which proceeds similar to Liang et al. (2014)) is sharp, matches the empirical performance and will be the guide of our preconditioners.

Lemma 1. *Let h be C^2 with $l_h I \preceq \nabla^2 h(\cdot) \preceq L_h I$ for some constants $l_h, L_h > 0$. Then the gradient descent on $\min_x \; h(Ax + b)$ with step size $1/t = 2/(L_h \sigma_{max}(A)^2 + l_h \sigma_{min>0}(A)^2)$ satisfies*

$$\|x^{k+1} - x^*\| \leq \frac{\varphi - 1}{\varphi + 1} \|x^k - x^*\|, \tag{9}$$

with $\varphi = \kappa(A)^2 \cdot \kappa(h)$, $\kappa(h) := L_h / l_h$.

Proof. See the supplementary material. □

The analysis in Theorem 2 below hinges on finite identification of the *active set* define as

$$\mathcal{A}(p) = \{e \in \mathcal{E} \; : \; |p_e| = 1\}\,. \tag{10}$$

The associated projection matrix is defined as

$$(P_{\mathcal{A}} p)_e = \begin{cases} p_e & \text{if } e \in \mathcal{A}, \\ 0 & \text{if } e \notin \mathcal{A}. \end{cases} \tag{11}$$

Correspondingly, let $\mathcal{I}(p) := \mathcal{E} \backslash \mathcal{A}(p)$ be the *inactive set* and $P_{\mathcal{I}} := I - P_{\mathcal{A}}$.

Theorem 2. *Suppose that* (4) *generates a sequence $\{p^k\}$ which converges to a minimizer $p^* \in \mathbb{R}^{\mathcal{E}}$ of* (3). *Under the assumptions that*

(A1) For each $e \in \mathcal{E}$, $\left(K \nabla G^*(-K^\top p^*)\right)_e = 0 \Rightarrow |p^*_e| < 1$;

(A2) For each $k \in \mathbb{N}$, $\underline{t} I \preceq T_k \preceq \bar{t} I$ with fixed $\underline{t}$, $\bar{t} > 0$;

(A3) T_k depends on p^k only through $\mathcal{A}(p^k)$;

there exists $\bar{k} \in \mathbb{N}$ such that for all $k \geq \bar{k}$:

(i) Finite identification, i.e.,

$$\mathcal{A}(p^k) = \mathcal{A}(p^*) \equiv \mathcal{A}^*, \; T_k \equiv T. \tag{12}$$

(ii) Local linear convergence, i.e.,

$$\|p^k - p^*\|_T \leq \left(\frac{\varphi - 1}{\varphi + 1}\right)^{k - \bar{k}} \|p^{\bar{k}} - p^*\|_T, \tag{13}$$

with

$$\varphi = \kappa(\Pi_{U(\mathcal{A}^*)} T^{-1/2} K)^2 \cdot \kappa(G^*), \tag{14}$$

and $\Pi_{U(\mathcal{A}^*)}$ the orthogonal projection onto the subspace $U(\mathcal{A}^*) := \ker(P_{\mathcal{A}^*} T^{-1/2})$.

Proof. (i) Finite identification of the active set follows by invoking Burke and Moré (1988, Corollary 3.6). The strict complementary condition at p^* required by that corollary is (A1). Further, the corollary requires

$$\operatorname{dist}(0, \nabla J(p^k) + N(p^k)) \to 0, \tag{15}$$

where $J = G^* \circ (-K^\top)$ and

$$N(\bar{p}) = \Big\{ p \in \mathbb{R}^{\mathcal{E}} : p_e = 0 \ \text{ if } e \notin \mathcal{A}(\bar{p}), \\ \operatorname{sgn}(\bar{p}_e) \cdot p_e \geq 0 \ \text{ if } e \in \mathcal{A}(\bar{p}) \Big\}, \tag{16}$$

denotes the normal cone at $\bar{p}$. From the optimality conditions of (4) it follows

$$tT_k(p^k - p^{k+1}) - \big(\nabla J(p^k) - \nabla J(p^{k+1})\big) \\ \in \nabla J(p^{k+1}) + N(p^{k+1}). \tag{17}$$

Then we have

$$\begin{aligned} &\operatorname{dist}(0, \nabla J(p^{k+1}) + N(p^{k+1})) \\ &\leq \|tT_k(p^k - p^{k+1}) - (\nabla J(p^k) - \nabla J(p^{k+1}))\| \\ &\leq \big(t\|T_k\| + L_{G^*}\lambda_{\max}(K^\top K)\big)\, \|p^k - p^{k+1}\|. \end{aligned} \tag{18}$$

Convergence of $\{p^k\}$ to p^* implies $\|p^k - p^{k+1}\| \to 0$ and (15) follows by (A2). Since the active set is constant for $k \geq \bar{k}$ we have by (A3) that $T_k \equiv T$.

(ii). Assume in the following that $k \geq \bar{k}$. Since $T_k = T$ due to (i), (4) is equivalent the projected gradient descent applied to

$$\min_{q \in \mathbb{R}^{\mathcal{E}}} \ \widetilde{J}(q) \quad \text{s.t. } \|T^{-1/2}q\|_\infty \leq 1, \tag{19}$$

under the change of variable $p = T^{-1/2}q$, $\widetilde{J} = J \circ T^{-1/2}$. The iteration in q is given by

$$q^{k+1} = \operatorname*{arg\,min}_{\|T^{-1/2}q\|_\infty \leq 1} \ \langle \nabla \widetilde{J}(q^k), q \rangle + \frac{t}{2}\|q - q^k\|^2, \tag{20}$$

whose optimality condition reads

$$t(q^k - q^{k+1}) \in T^{-1/2}N(T^{-1/2}q^{k+1}) + \nabla \widetilde{J}(q^k). \tag{21}$$

From (i) we know that $P_{\mathcal{A}^*}p^{k+1} = P_{\mathcal{A}^*}p^k$, which yields

$$\begin{aligned} &P_{\mathcal{A}^*}T^{-1/2}q^{k+1} = P_{\mathcal{A}^*}T^{-1/2}q^k, \\ &\Rightarrow q^{k+1} - q^k \in \ker(P_{\mathcal{A}^*}T^{-1/2}) = U(\mathcal{A}^*). \end{aligned} \tag{22}$$

In addition, in view of (16) we have

$$T^{-1/2}N(T^{-1/2}q^k) \subset U(\mathcal{A}^*)^\perp. \tag{23}$$

Thus, applying $\Pi_{U(\mathcal{A}^*)}$ on both sides of (21) yields an *equivalent* characterization:

$$0 = \Pi_{U(\mathcal{A}^*)}\nabla \widetilde{J}(q^k) + t(q^{k+1} - q^k). \tag{24}$$

Indeed, this is the gradient descent on $\widetilde{J}$ restricted to $U(\mathcal{A}^*)$, which we rewrite as

$$\begin{aligned} q^{k+1} &= q^k + t^{-1}\Pi_{U(\mathcal{A}^*)}T^{-1/2}K^\top \nabla G^*(-K^\top T^{-1/2}q^k) \\ &= q^k + t^{-1}\Pi_{U(\mathcal{A}^*)}T^{-1/2}K^\top \nabla G^*(-K^\top T^{-1/2} \\ &\qquad (\Pi_{U(\mathcal{A}^*)}q^k + \Pi_{U(\mathcal{A}^*)^\perp}q^{\bar{k}})). \end{aligned} \tag{25}$$

Hence (20) is equivalent to gradient descent on the function $G^* \circ (A \cdot + b)$ with $A = -K^\top T^{-1/2}\Pi_{U(\mathcal{A}^*)}$, $b = -K^\top T^{-1/2}\Pi_{U(\mathcal{A}^*)^\perp}p^{\bar{k}}$. Using Lemma 1 yields the linear convergence in $\{q^k\}$. As $\|q^k\| = \|T^{1/2}p^k\| = \|p^k\|_T$, we achieve the linear convergence, with respect to the T-norm, of the original sequence $\{p^k\}$. □

Corollary 3. *Let φ be given as in* (14)*. Locally (i.e., for $k \geq \bar{k}$), with fixed $T \equiv T_{\bar{k}}$ we have $\|p^k - p^*\| \leq \varepsilon$ whenever*

$$k \geq \bar{k} + \frac{\varphi + 1}{2}\log\left(\frac{\|p^{\bar{k}} - p^*\|\sqrt{\kappa(T)}}{\varepsilon}\right). \tag{26}$$

We remark that there are bounds in literature on $\bar{k}$, see Liang et al. (2017, Prop. 3.6) or the recent works (Nutini et al., 2017b,a). Analyzing which choice of variable metric T_k lead to fast identification of $\mathcal{A}^*$ is beyond the scope of this work.

3 Combinatorial preconditioner

Suggested by the local convergence analysis and Corollary 3 from the previous section, an ideal preconditioner T ought to minimize the condition number $\kappa(\Pi_{U(\mathcal{A}^*)}T^{-1/2}K)$ once the active set $\mathcal{A}^*$ is identified. In practice, however, computationally amenable choices of T are rather constrained due to a generic *trade-off* between convergence speed of (outer) iterations and per-iteration cost, i.e., the T-scaled proximal evaluation in (4) or (7). A dense matrix T, in general, will render inner iterations expensive, as in the case of proximal Newton method (Lee et al., 2014). For this reason, Pock and Chambolle (2011); Giselsson and Boyd (2014a,b); Becker and Fadili (2012); Becker et al. (2018) consider diagonal or low-rank preconditioners to keep the inner iterations fast and tractable.

Towards yet better balance of this trade-off, Möllenhoff et al. (2018) made use of fast TV solver on trees (Condat, 2013; Kolmogorov et al., 2016) and proposed a class of block diagonal preconditioners via graph partitioning (aiming at optimizing $\kappa(T^{-1/2}K)$ heuristically, however). The optimal condition number $\kappa(T^{-1/2}K)$ is achieved by matroid partitioning. As a remark, combinatorial preconditioners for solving linear systems involving graph Laplacians date back to Vaidya (1991); refer to Spielman (2010) for a more detailed survey.

In this section, we construct combinatorial preconditioners which are more faithful, compared to the ones from Möllenhoff et al. (2018), to the (local) convergence analysis. In a nutshell, given the current active/inactive sets of edges, we partition the graph into *inactively nested forests* in the sense of (37), so that the resulting preconditioner yields a guaranteed (local) convergence rate, which is made precise in Theorem 5.

To construct our preconditioner, let the edge set $\mathcal{E}$ be partitioned into L mutually disjoint subsets, i.e., $\mathcal{E} = \bigsqcup_{l=1}^{L} \mathcal{E}_l$, such that each subgraph $\mathcal{G}_l = (\mathcal{V}, \mathcal{E}_l, \omega|_{\mathcal{E}_l})$ is a *forest*. Correspondingly, we define P_l as the canonical projection from $\mathbb{R}^{\mathcal{E}}$ to $\mathbb{R}^{\mathcal{E}_l}$, i.e., $P_l p = p|_{\mathcal{E}_l}$ for any $p \in \mathbb{R}^{\mathcal{E}}$. Thus, the matrix K can be decomposed into submatrices $\{K_l\}_{l=1}^{L}$ where each $K_l = P_l K \in \mathbb{R}^{|\mathcal{E}_l| \times |\mathcal{V}|}$. Analogously, let $\nabla_l = P_l \nabla$. Note that each $\nabla_l^\top$ (or $K_l^\top$) has full column rank, and hence

$$T_l := K_l K_l^\top, \quad \forall l \in \{1, ..., L\}, \tag{27}$$

is symmetric positive definite.

We then define our preconditioner as

$$T := \sum_{l=1}^{L} P_l^\top T_l P_l. \tag{28}$$

In view of Theorem 2, we analyze in the following the condition number of the following matrix:

$$\begin{aligned}\Pi_{\mathcal{I}} &:= K^\top T^{-1/2} \Pi_{U(\mathcal{A})} T^{-1/2} K \\ &= K^\top T^{-1/2} (I - T^{-1/2} P_{\mathcal{A}} (T^{-1/2} P_{\mathcal{A}})^\dagger) T^{-1/2} K.\end{aligned} \tag{29}$$

As a preparatory result, the following lemma decomposes $\Pi_{\mathcal{I}}$ into orthogonal projections onto subspaces.

Lemma 4. *Given $\mathcal{E} = \mathcal{A} \sqcup \mathcal{I}$, let $\mathcal{G}$ be partitioned into L nonempty forests $\{\mathcal{G}_l\}_{l=1}^{L}$. Then the matrix defined in (29) can be characterized as*

$$\Pi_{\mathcal{I}} = \sum_{l=1}^{L} \Pi_{\mathcal{I},l}, \tag{30}$$

where each $\Pi_{\mathcal{I},l}$ is the orthogonal projection onto the linear subspace $\mathcal{S}_{\mathcal{I},l}$ defined by

$$\mathcal{S}_{\mathcal{I},l} := \operatorname{span}\{\nabla_e^\top : e \in \mathcal{I} \cap \mathcal{E}_l\}. \tag{31}$$

Proof. (i) We show the identity (30) with $I_l := P_l I P_l^\top$, $P_{\mathcal{A},l} := P_l P_{\mathcal{A}} P_l^\top$, $P_{\mathcal{I},l} := P_l P_{\mathcal{I}} P_l^\top$, and

$$\begin{aligned}\Pi_{\mathcal{I},l} := &K_l^\top T_l^{-1/2} (I_l - (T_l^{-1/2} P_{\mathcal{A},l})(T_l^{-1/2} P_{\mathcal{A},l})^\dagger) \\ &T_l^{-1/2} K_l.\end{aligned} \tag{32}$$

Note that

$$\begin{aligned}K^\top T^{-1/2} &= \sum_{l=1}^{L} K^\top P_l^\top T_l^{-1/2} P_l \\ &= \sum_{l=1}^{L} K_l^\top T_l^{-1/2} P_l,\end{aligned} \tag{33}$$

$$\begin{aligned}T^{-1/2} P_{\mathcal{A}} &= \left(\sum_{l=1}^{L} P_l^\top T_l^{-1/2} P_l\right) \left(\sum_{l'=1}^{L} P_{l'}^\top P_{\mathcal{A},l'} P_{l'}\right) \\ &= \sum_{l=1}^{L} P_l^\top T_l^{-1/2} P_{\mathcal{A},l} P_l,\end{aligned} \tag{34}$$

$$(T^{-1/2} P_{\mathcal{A}})^\dagger = \sum_{l=1}^{L} P_l^\top (T_l^{-1/2} P_{\mathcal{A},l})^\dagger P_l. \tag{35}$$

By plugging (33)–(35) into (29), we accomplish (i).

(ii) We show each $\Pi_{\mathcal{I},l}$ is the orthogonal projection onto $\mathcal{S}_{\mathcal{I},l}$. First, it is easy to see $\Pi_{\mathcal{I},l}$ is symmetric and $\Pi_{\mathcal{I},l}^2 = \Pi_{\mathcal{I},l}$, and hence an orthogonal projection. Secondly, note that $\operatorname{rank} \Pi_{\mathcal{I},l} = |\mathcal{I} \cap \mathcal{E}_l| = \operatorname{rank} K_l^\top P_{\mathcal{I},l}$. Furthermore, we have the following equation:

$$\begin{aligned}\Pi_{\mathcal{I},l} K_l^\top P_{\mathcal{I},l} = &K_l^\top P_{\mathcal{I},l} - K_l^\top T_l^{-1/2} \\ &(T_l^{-1/2} P_{\mathcal{A},l})(T_l^{-1/2} P_{\mathcal{A},l})^\dagger T_l^{1/2} P_{\mathcal{I},l} \\ = &K_l^\top P_{\mathcal{I},l},\end{aligned} \tag{36}$$

which completes step (ii). □

Theorem 5. *Given $\mathcal{E} = \mathcal{A} \sqcup \mathcal{I}$, let $\mathcal{G}$ be partitioned into L nonempty, inactively nested forests $\{\mathcal{G}_l\}_{l=1}^{L}$ in the sense that*

$$\mathcal{S}_{\mathcal{I},1} = ... = \mathcal{S}_{\mathcal{I},\hat{l}} \supsetneq \mathcal{S}_{\mathcal{I},\hat{l}+1} \supseteq ... \supseteq \mathcal{S}_{\mathcal{I},L} \supsetneq \{0\}, \tag{37}$$

with the subspaces defined in (31). Then we have $\lambda_{\min>0}(\Pi_{\mathcal{I}}) = \hat{l}$ and the (local) convergence rate in Theorem 2 is $\varphi = (L/\hat{l}) \cdot \kappa(G^*)$.

Proof. By Lemma 4, we have $\lambda_{\max}(\Pi_{\mathcal{I}}) \leq \sum_{l=1}^{L} \lambda_{\max}(\Pi_{\mathcal{I},l}) \leq L$. In fact, the equality holds since $\Pi_{\mathcal{I}} v = Lv$ for some nonzero $v \in \mathcal{S}_{\mathcal{I},L}$. On the other hand, for any $v \in \operatorname{ran} \Pi_{\mathcal{I}}$, we have $\langle v, \Pi_{\mathcal{I}} v\rangle \geq \sum_{l=1}^{\hat{l}} \langle v, \Pi_{\mathcal{I},l} v\rangle = \hat{l}\|v\|^2$. The equality holds for some nonzero $v \in \operatorname{ran} \Pi_{\mathcal{I}} \cap (\mathcal{S}_{\mathcal{I},\hat{l}+1})^\perp$. This yields $\lambda_{\min>0}(\Pi_{\mathcal{I}}) = \hat{l}$. □

4 Implementation

In this section, we specify how to construct our preconditioner and apply active-set-based reconditioning to both PG and PDHG algorithms. We only trigger reconditioning every n iterations. When reconditioning is performed at iteration k, a greedy heuristic is used for constructing the preconditioner T_k; see Section 4.1. Then, using the separability of $\|\cdot\|_\infty$, we perform the updates across the subgraphs $\{\mathcal{G}_l\}_{l=1}^L$:

$$p^{k+1}|_{\mathcal{E}_l} = \arg\min_{p\in\mathbb{R}^{\mathcal{E}_l}} -\left\langle K\bar{u}^k|_{\mathcal{E}_l}, p\right\rangle + \delta\{\|p\|_\infty \leq 1\} + \frac{t}{2}\|p - p^k\|^2_{T_{k,l}}, \quad (38)$$

where $\bar{u}^k$ is defined as:

$$\bar{u}^k = \begin{cases} \nabla G^*(-K^\top p^k), & \text{for PG,} \\ 2u^{k+1} - u^k, & \text{for PDHG.} \end{cases} \quad (39)$$

The proximal evaluation required by (38) is detailed in Section 4.2, which invokes the message-passing algorithm on trees. The overall complexity of the reconditioned algorithm is discussed in Section 4.3.

4.1 Constructing preconditioner

Following Theorem 5 we aim to find a preconditioner T_k which minimizes the condition number $\varphi = (L/\widehat{l}) \cdot \kappa(G^*)$, and hence the local linear convergence rate. Theoretically, optimal T_k can be found in polynomial time by Matroid partitioning as in Möllenhoff et al. (2018). The computation time is prohibitively large for the graphs in practice, however. Here we present a greedy heuristic to find inactively nested forests.

Given an input graph $\mathcal{G}$ we partition the graph based on the active set at the current dual variable p^k. We assign to each edge $e \in \mathcal{E}$ an additional weight $\rho_e = 1 - \left|1 - \left|p_e^k\right|\right|$. Then, a minimum spanning forest according to that weight is generated using Kruskal's algorithm (Kruskal, 1956). This spanning forest is then subtracted from current graph and is added to the set $\{\mathcal{G}_l\}_{l=1}^L$. We perform this generation and subtraction iteratively until no edges remain in the original graph.

The partitioning weight is introduced for two reasons: Firstly, we found it unstable to determine the active set $\mathcal{A}(p^k)$ numerically according to a threshold; Secondly, computing the preconditioner T_k is quite expensive for large graphs. This strategy could extend the suitable duration of current preconditioner since a potential active edge often has a larger partitioning weight.

4.2 Backward solver

The introduction of the proposed preconditioner T_k makes the backward update (38) more expensive. Here we describe how to solve it efficiently, following Möllenhoff et al. (2018). Combining the linear and the quadratic term, (38) can be re-written as:

$$p^{k+1}|_{\mathcal{E}_l} = \arg\min_{\|p\|_\infty\leq 1} \frac{1}{2}\|K_l^\top p + f_l\|^2, \quad (40)$$

where $f_l = -K_l^\top p^k|_{\mathcal{E}_l} - \bar{u}^k/t$. The (Fenchel) dual problem of (40) is given by

$$v_l = \arg\min_{u\in\mathbb{R}^{\mathcal{V}}} \frac{1}{2}\|u - f_l\|^2 + \|K_l u\|_1, \quad (41)$$

which is simply a weighted total variation problem on the individual trees in the forest $\mathcal{G}_l$. We solve the problem (41) using the message-passing algorithm introduced in Kolmogorov et al. (2016). To retrieve $p^{k+1}|_{\mathcal{E}_l}$ from v_l one can use the optimality condition:

$$K_l^\top p^{k+1}|_{\mathcal{E}_l} = v_l - f_l. \quad (42)$$

4.3 Discussion on complexity

For non-preconditioned proximal gradient, the complexity of each iteration is $\mathcal{O}(|\mathcal{E}|)$. The preconditioned variant has complexity $\mathcal{O}(\sum_{t=1}^{\mathcal{T}} |\mathcal{E}_t| \log(|\mathcal{E}_t|))$ where $\mathcal{T}$ is the number of trees using the message-passing algorithm (Kolmogorov et al., 2016). The preconditioned update can still be parallelized, as the message-passing can run for each tree in parallel. Construction of the preconditioner T_k based on the greedy inactively nested forest strategy with Prim's or Kruskal's algorithm is $\mathcal{O}(|\mathcal{E}|^2 \log(|\mathcal{E}|)/|\mathcal{V}|)$ (Cheriton and Tarjan, 1976).

After entering the local linear convergence phase, the overall iteration complexity is $\mathcal{O}(\varphi \log(1/\varepsilon))$ to find an ε-accurate solution (see Corollary 3). While each iteration of this algorithm is slighty more costly (by roughly a factor of $\log(|\mathcal{E}|)$), the condition number φ is drastically reduced. For regular grids we have that $\varphi \in \mathcal{O}(|\mathcal{V}|)$ (cf. Möllenhoff et al. (2018, Theorem 4)) in the non-preconditioned case. The proposed preconditioner improves this to a *constant* $\varphi \in \mathcal{O}(1)$, independent of problem size at the expense of a slightly more expensive dual update step (up to a logarithmic factor).

5 Applications

In the following experiments we compare four strategies: non-preconditioned $T_k = I$, diagonally scaled $T_k = \text{diag}(KK^\top)$, nested (linear) forest from Möllenhoff et al. (2018) and the proposed "inactively nested forest" for both PG and PDHG. In PG, the step size is $t = 1/(L \cdot L_{G^*})$, for PDHG we set $s = 0.11$, $t = 10 \cdot L$.

5.1 Numerical validation on synthetic data

As a first numerical example, we consider the fused Lasso (Tibshirani et al., 2005) (also called ROF model in imaging (Rudin et al., 1992))

$$\min_{u\in\mathbb{R}^{\mathcal{V}}} \frac{1}{2}\|u - f\|^2 + \|Ku\|_1. \quad (43)$$

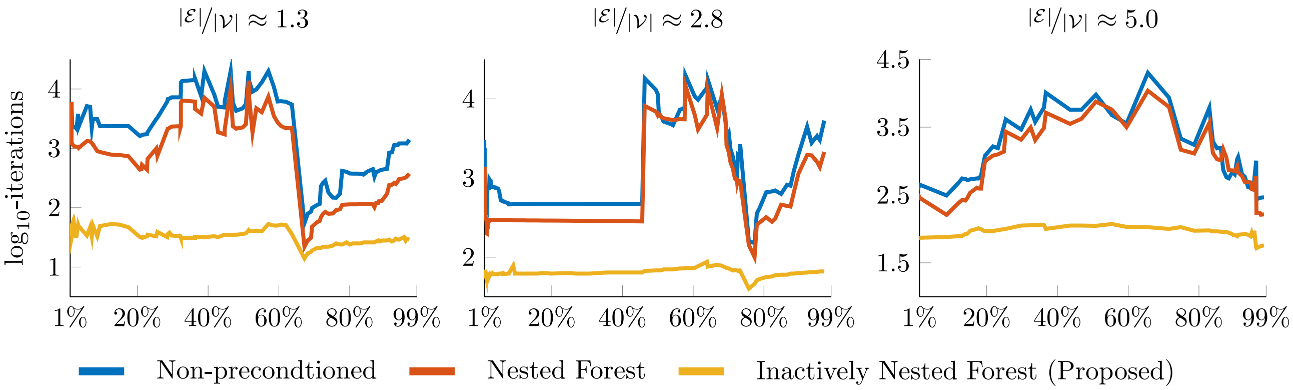

Figure 2: We show $\log_{10}$-iterations required by PG to reach a primal-dual gap smaller than 10^{-10} over percentage of active edges at the optimal solution for random graphs with varying edge-to-vertex ratio. The reconditioning strategy requires several orders of magnitude less iterations than no preconditioner and the nested-forest preconditioner (Möllenhoff et al., 2018).

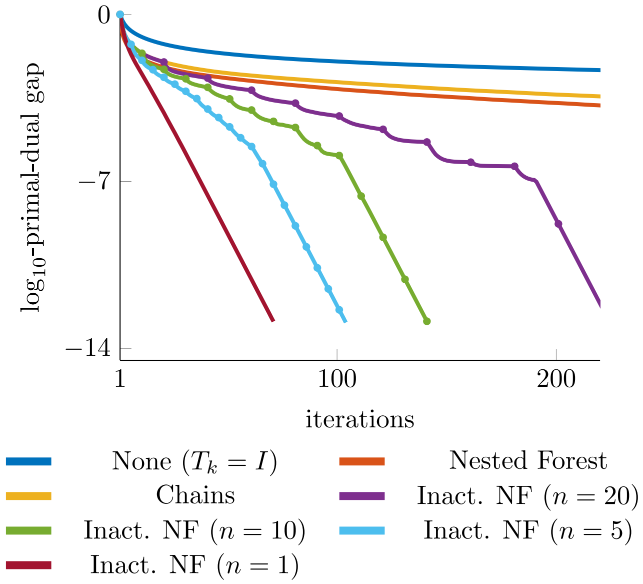

Figure 3: We show $\log_{10}$-primal dual gap vs iterations for PG (4) with various choices of T_k. The non-preconditioned choice $T_k = I$ performs the worst, followed by the preconditioners proposed in Möllenhoff et al. (2018). We indicate reconditioning by a dot and carry it out every n iterations for $n \in \{20, 10, 5, 1\}$. Smaller n leads to an improved performance.

We solve (43) on random graphs with fixed $|\mathcal{V}| = 512$ using proximal gradient (PG) with f chosen uniformly random in $[0, 1]$. Here, we consider two factors: edge-to-vertex ratio and percentage of active edges at the optimal solution. For the proposed reconditioning we set the frequency to $n = 1$.

The results are shown in Fig. 2. For reasonable amounts of active edges at the solution (30% – 80%) the proposed preconditioning strategy requires orders of magnitude less iterations to reach a primal-dual gap under 10^{-10}. Moreover, it is shown that we require the fewest iterations across all scenarios.

In Fig. 3 we show $\log_{10}$-primal-dual gap over iterations for PG applied to (43) on a 100×100 grid graph with different choices of T_k and moderate regularization strength (30% of active edges at the optimal solution). The proposed preconditioner outperforms vanilla PG ($T_k = I$) and the recent preconditioners proposed in Möllenhoff et al. (2018). Reconditioning more often leads to faster convergence, but as recomputing the preconditioner is expensive there is a trade-off between reducing the number of iterations and fast updates. In practice, a choice of the reconditioning frequency n between 5 and 30 leads to the best performance.

5.2 Fused Lasso on real-world graphs

To consider a more realistic scenario, we solve the model (43) on real-world graphs from a popular graph-cut benchmark considered in Goldberg et al. (2011). Furthermore, instead of using standard PG we used the accelerated FISTA variant (Chambolle and Dossal, 2015; Attouch et al., 2015; Liang et al., 2017) with overrelaxation parameter $\beta_k = (k-1)/(k+2)$. Reconditioning takes place at every 30 iterations. We discard the momentum for one iteration after reconditioning, which improved the stability. In Table 1, we show the running time and number of iterations of FISTA with non-preconditioned, diagonal preconditioner, nested forest (Möllenhoff et al., 2018), linear forest (Möllenhoff et al., 2018) and the proposed inactively nested forest.

Our preconditioner outperforms the other methods in all cases on number of iterations, despite a rather large choice of $n = 30$. However, the linear forest from Möllenhoff et al. (2018) performs better with

Instance		None		Diagonal		Nest. Forest		Lin. Forest		Inact. NF	
name	$\frac{\lvert\mathcal{A}_*\rvert}{\lvert\mathcal{E}\rvert}$	it$[10^3]$	time[s]	it$[10^3]$	time[s]	it$[10^3]$	time[s]	it$[10^3]$	time[s]	it$[10^3]$	time[s]
rmf-long	0.02	–	–	19	473	12	2539	18	**233.3**	**1.9**	474.9
rmf-wide	0.19	–	–	62	665	27	2274	43	213.1	**0.19**	**18.54**
horse	0.02	–	–	–	–	2.9	340.8	37	355.6	**0.73**	**155.3**
alue	0.03	–	–	–	–	4.5	117.2	100	270.9	**0.71**	**155.3**
lux	0.01	–	–	–	–	13	1254	–	–	**0.40**	**54.34**
punch	0.01	488	968	–	–	14.9	1445	203	872.1	**0.34**	**62.66**
BVZ*	0.35	27	74.0	22	730	1.12	434.8	0.57	**6.59**	**0.49**	261
manga*	0.05	–	–	–	–	38	48591	5.3	**230**	**1.41**	4609
KZ2	0.5	419	3042	1.6	159	0.43	614.1	0.6	**56.0**	**0.42**	965.9
ferro	0.09	9.25	186	5.83	639.6	0.36	430.3	0.93	**86.23**	**0.27**	609.3

Table 1: We show the number of iterations and running time to reach a relative primal dual gap less than 10^{-10} on (43) on real-world graphs. FISTA with various choices of T_k is used to solve these problems. "–" means the algorithm failed to reach the tolerance within 5×10^5 iterations. "*" means that graph has a grid structure.

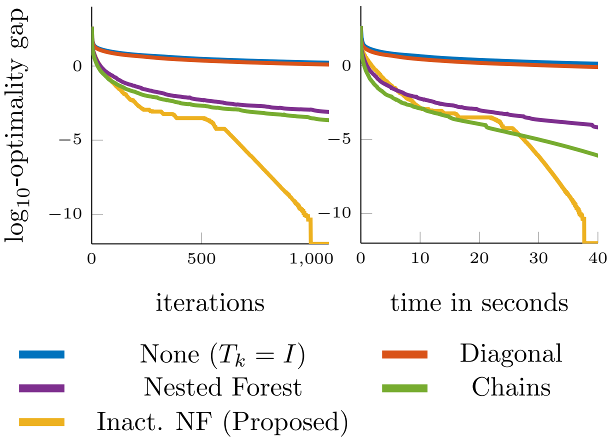

Figure 4: $\log_{10}$-optimality gap over iterations (left) and time (right) for PDHG with various preconditioners applied to a TV deconvolution problem.

respect to the running time on 5 out of 10 datasets. The two datasets with grid structures leads to chain partition on which message-passing is much faster than on trees. The sizes of last two graphs are huge ($|\mathcal{V}| \approx 250.000$, $|\mathcal{E}| \approx 600.000$) and therefore partitioning is quite expensive. To summarize, the proposed preconditioning strategy improves the number of iterations, but to ensure a shorter overall running time, an efficient implementation or improved strategy on reconstructing the tree decomposition might be required.

5.3 Linear inverse problems

In this image processing experiment we consider a TV deconvolution problem on a regular 2D grid of size 116×87. The data term is given by $G(u) = \frac{1}{2}\|Au - f\|^2$, where the forward model A is a convolution with motion blur kernel with radius 3. We construct f by applying the forward model and adding Gaussian noise. The overall problem is solved using PDHG. The primal update is a quadratic problem and we use a few iterations of (warm started) conjugate gradient. Considering the size of the problem, we set the reconditioning frequency to $n = 5$ for the proposed approach.

In Fig. 4 we show the $\log_{10}$-optimality gap over iterations and time for various choices of preconditioners. The diagonal preconditioner is the one from Pock and Chambolle (2011) with $\alpha = 1$. The forest preconditioners perform comparably when the accuracy is lower. Once the local convergence regime is entered, the proposed algorithm achieves linear convergence rate. Especially for high accuracies, the proposed reconditioning strategy outperforms the other approaches with respect to overall running time and iterations.

6 Discussion and conclusion

We presented an efficient reconditioning strategy for proximal algorithms on graphs. By relying on a sharp analysis of the local linear convergence rate we proposed an edge partitioning of the graph into forests which provably boosts the linear convergence rate. The scaled dual updates are still efficiently computable thanks to a message-passing algorithm on trees.

While one is tempted to commit to a super-linearly convergent solver once the optimal active set is identified (as e.g., mentioned in Liang et al. (2014, 2017, 2018); Nutini et al. (2017b)), it is unfortunately difficult to verify in practice whether the current active set is the optimal. Furthermore, as observed in the numerical experiments, the adaptive preconditioning strategy practically improves the convergence to some extent also *before* the local linear convergence regime is entered. The result suggests that local convergence analysis can serve as a practical guideline for constructing preconditioners for proximal algorithms.

References

Attouch, H., Peypouquet, J., and Redont, P. (2015). Fast convergence of an inertial gradient-like system with vanishing viscosity. *arXiv:1507.04782*.

Barbero, A. and Sra, S. (2011). Fast Newton-type methods for total variation regularization. In *Proceedings of the 32nd International Conference on Machine Learning, ICML*.

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

Bauschke, H. H. and Combettes, P. L. (2011). *Convex Analysis and Monotone Operator Theory in Hilbert Spaces*. Springer Science & Business Media.

Beck, A. and Teboulle, M. (2009). A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM J. Imaging Sci.*, 2:183–202.

Becker, S., Fadili, J., and Ochs, P. (2018). On quasi-Newton forward–backward splitting: Proximal calculus and convergence. *arXiv:1801.08691*.

Becker, S. and Fadili, M. J. (2012). A quasi-Newton proximal splitting method. In *Proceedings of the 26th International Conference on Neural Information Processing Systems, NIPS*.

Benzi, K., Kalofolias, V., Bresson, X., and Vandergheynst, P. (2016). Song recommendation with non-negative matrix factorization and graph total variation. In *Proceedings of the 42nd IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP*.

Bredies, K. and Sun, H. (2015). Preconditioned Douglas–Rachford splitting methods for convex-concave saddle-point problems. *SIAM J. Numer. Anal.*, 53:421–444.

Bresson, X., Laurent, T., Uminsky, D., and Von Brecht, J. (2013). Multiclass total variation clustering. In *Proceedings of the 27th International Conference on Neural Information Processing Systems, NIPS*.

Burke, J. V. and Moré, J. J. (1988). On the identification of active constraints. *SIAM J. Numer. Anal.*, 25(5):1197–1211.

Chambolle, A. and Darbon, J. (2009). On total variation minimization and surface evolution using parametric maximum flows. *Int. J. Comput. Vis.*, 84:288–307.

Chambolle, A. and Dossal, C. (2015). On the convergence of the iterates of the "fast iterative shrinkage/thresholding algorithm". *J. Optim. Theory Appl.*, 166:968–982.

Chambolle, A. and Pock, T. (2011). A first-order primal-dual algorithm for convex problems with applications to imaging. *J. Math. Imaging Vis.*, 40:120–145.

Chambolle, A. and Pock, T. (2016a). An introduction to continuous optimization for imaging. *Acta Numer.*, 25:161–319.

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

Cheriton, D. and Tarjan, R. E. (1976). Finding minimum spanning trees. *SIAM J. Comput.*, 5(4):724–742.

Combettes, P. L. and Pesquet, J.-C. (2011). Proximal splitting methods in signal processing. In *Fixed-Point Algorithms for Inverse Problems in Science and Engineering*, pages 185–212. Springer.

Condat, L. (2013). A direct algorithm for 1D total variation denoising. *IEEE Signal Process. Lett.*, 20:1054–1057.

Diamond, S. and Boyd, S. (2017). Stochastic matrix-free equilibration. *J. Optim. Theory Appl.*, 172:436–454.

Fougner, C. and Boyd, S. (2015). Parameter selection and pre-conditioning for a graph form solver. *arXiv:1503.08366*.

Friedlander, M. P. and Goh, G. (2017). Efficient evaluation of scaled proximal operators. *Electron. Trans. Numer. Anal.*, 46:1–22.

Garcia-Cardona, C., Merkurjev, E., Bertozzi, A. L., Flenner, A., and Percus, A. G. (2014). Multiclass data segmentation using diffuse interface methods on graphs. *IEEE Trans. Pattern Anal. Mach. Intell.*, 36(8):1600–1613.

Garrigos, G., Rosasco, L., and Villa, S. (2017). Convergence of the forward-backward algorithm: Beyond the worst case with the help of geometry. *arXiv:1703.09477*.

Gilboa, G. and Osher, S. (2008). Nonlocal operators with applications to image processing. *Multiscale Modeling & Simulation*, 7(3):1005–1028.

Giselsson, P. and Boyd, S. (2014a). Diagonal scaling in Douglas-Rachford splitting and ADMM. In *Proceedings of the 53rd IEEE Conference on Decision and Control, CDC*.

Giselsson, P. and Boyd, S. (2014b). Preconditioning in fast dual gradient methods. In *Proceedings of the 53rd IEEE Conference on Decision and Control, CDC*.

Giselsson, P. and Boyd, S. (2015). Metric selection in fast dual forward–backward splitting. *Automatica*, 62:1–10.

Goldberg, A., Hed, S., Kaplan, H., Tarjan, R., and Werneck, R. (2011). Maximum flows by incremental breadth-first search. *European Symposium on Algorithms, ALGO ESA*.

Hein, M. and Setzer, S. (2011). Beyond spectral clustering – tight relaxations of balanced graph cuts. In *Proceedings of the 25th International Conference on Neural Information Processing Systems, NIPS*.

Hein, M., Setzer, S., Jost, L., and Rangapuram, S. S. (2013). The total variation on hypergraphs – learning on hypergraphs revisited. In *Proceedings of the 27th International Conference on Neural Information Processing Systems, NIPS*.

Hochbaum, D. S. (2001). An efficient algorithm for image segmentation, Markov random fields and related problems. *Journal of the ACM (JACM)*, 48(4):686–701.

Hoffman, A. J. (1952). On approximate solutions of systems of linear inequalities. *J. Res. Natl. Bur. Standards*, 49:263–265.

Karimi, H., Nutini, J., and Schmidt, M. (2016). Linear convergence of gradient and proximal-gradient methods under the Polyak-Łojasiewicz condition. In *Proceedings of the Joint European Conference on Machine Learning and Knowledge Discovery in Databases ECML PKDD*.

Klatte, D. and Thiere, G. (1995). Error bounds for solutions of linear equations and inequalities. *Zeitschrift für Operations Research*, 41(2):191–214.

Kolmogorov, V., Pock, T., and Rolinek, M. (2016). Total variation on a tree. *SIAM J. Imaging Sci.*, 9:605–636.

Kruskal, J. B. (1956). On the shortest spanning subtree of a graph and the traveling salesman problem. *Proceedings of the American Mathematical society*.

Kumar, K. and Bach, F. (2017). Active-set methods for submodular minimization problems. *J. Mach. Learn. Res.*, 18(132):1–31.

Landrieu, L. and Obozinski, G. (2017). Cut pursuit: Fast algorithms to learn piecewise constant functions on general weighted graphs. *SIAM J. on Imaging Sci.*, 10(4):1724–1766.

Lee, J. D., Sun, Y., and Saunders, M. A. (2014). Proximal Newton-type methods for minimizing composite functions. *SIAM J. Optim.*, 24:1420–1443.

Liang, J., Fadili, J., and Peyré, G. (2014). Local linear convergence of forward–backward under partial smoothness. In *Proceedings of the 28th International Conference on Neural Information Processing Systems, NIPS*.

Liang, J., Fadili, J., and Peyré, G. (2017). Activity identification and local linear convergence of forward–backward-type methods. *SIAM J. Optim.*, 27:408–437.

Liang, J., Fadili, J., and Peyré, G. (2018). Local linear convergence analysis of primal–dual splitting methods. *Optimization*, 67(6):821–853.

Lou, Y., Zhang, X., Osher, S., and Bertozzi, A. (2010). Image recovery via nonlocal operators. *Journal of Scientific Computing*, 42(2):185–197.

Möllenhoff, T., Ye, Z., Wu, T., and Cremers, D. (2018). Combinatorial preconditioners for proximal algorithms on graphs. In *Proceedings of the 21st International Conference on Artificial Intelligence and Statistics, AISTATS*.

Necoara, I., Nesterov, Y., and Glineur, F. (2015). Linear convergence of first order methods for non-strongly convex optimization. *arXiv:1504.06298*.

Nesterov, Y. (1983). A method for solving the convex programming problem with convergence rate $\mathcal{O}(1/k^2)$. *Soviet Mathematics Doklady*, 269:543–547.

Newcombe, R. A., Lovegrove, S. J., and Davison, A. J. (2011). DTAM: Dense tracking and mapping in real-time. In *Proceedings of the 13th International Conference on Computer Vision, ICCV*.

Nishihara, R., Lessard, L., Recht, B., Packard, A., and Jordan, M. (2015). A general analysis of the convergence of ADMM. In *Proceedings of the 32nd International Conference on Machine Learning, ICML*.

Nutini, J., Laradji, I., and Schmidt, M. (2017a). Let's make block coordinate descent go fast: Faster greedy rules, message-passing, active-set complexity, and superlinear convergence. *arXiv:1712.08859*.

Nutini, J., Schmidt, M., and Hare, W. (2017b). "Active-set complexity" of proximal gradient: How long does it take to find the sparsity pattern? *arXiv:1712.03577*.

Ochs, P., Chen, Y., Brox, T., and Pock, T. (2014). iPiano: Inertial proximal algorithm for nonconvex optimization. *SIAM J. Imaging Sci.*, 7:1388–1419.

Parikh, N. and Boyd, S. (2013). Proximal algorithms. *Foundations and Trends in Optimization*, 1:123–231.

Pock, T. and Chambolle, A. (2011). Diagonal preconditioning for first order primal-dual algorithms in convex optimization. In *Proceedings of the 13th International Conference on Computer Vision, ICCV*.

Polyak, B. T. (1964). Some methods of speeding up the convergence of iteration methods. *USSR Computational Mathematics and Mathematical Physics*, 4:1–17.

Raguet, H. and Landrieu, L. (2018). Cut-pursuit algorithm for regularizing nonsmooth functionals with graph total variation. In *Proceedings of the 35th International Conference on Machine Learning, ICML*.

Rudin, L. I., Osher, S., and Fatemi, E. (1992). Nonlinear total variation based noise removal algorithms. *Physica D: Nonlinear Phenomena*, 60(1):259–268.

Schmidt, M., Kim, D., and Sra, S. (2012). Projected Newton-type methods in machine learning. *Optimization for Machine Learning*.

Spielman, D. A. (2010). Algorithms, graph theory, and linear equations in Laplacian matrices. In *Proceedings of the International Congress of Mathematicians, ICM*.

Tibshirani, R., Saunders, M., Rosset, S., Zhu, J., and Knight, K. (2005). Sparsity and smoothness via the fused lasso. *J. R. Stat. Soc.*, 67(1):91–108.

Vaidya, P. M. (1991). Solving linear equations with symmetric diagonally dominant matrices by constructing good preconditioners. (A talk based on the manuscript was presented at the IMA Workshop on Graph Theory and Sparse Matrix Computation).

Xin, B., Kawahara, Y., Wang, Y., and Gao, W. (2014). Efficient generalized fused Lasso and its application to the diagnosis of Alzheimer's disease. In *Proceedings of the 28th AAAI Conference on Artificial Intelligence*.