Proceedings of Machine Learning Research vol 195:1–25, 2023 36th Annual Conference on Learning Theory

Accelerated and Sparse Algorithms for Approximate Personalized PageRank and Beyond

David Martínez-Rubio* MARTINEZ-RUBIO@ZIB.DE
Institute of Mathematics
Berlin Institute of Technology
Berlin, Germany

Elias Wirth* WIRTH@MATH.TU-BERLIN.DE
Institute of Mathematics
Berlin Institute of Technology
Berlin, Germany

Sebastian Pokutta POKUTTA@ZIB.DE
Institute of Mathematics
Berlin Institute of Technology
Berlin, Germany

Editors: Gergely Neu and Lorenzo Rosasco

Abstract

It has recently been shown that ISTA, an unaccelerated optimization method, presents sparse updates for the ℓ_1-regularized undirected personalized PageRank problem (Fountoulakis et al., 2019), leading to cheap iteration complexity and providing the same guarantees as the approximate personalized PageRank algorithm (APPR) (Andersen et al., 2006). In this work, we design an accelerated optimization algorithm for this problem that also performs sparse updates, providing an affirmative answer to the COLT 2022 open question of Fountoulakis and Yang (2022). Acceleration provides a reduced dependence on the condition number, while the dependence on the sparsity in our updates differs from the ISTA approach. Further, we design another algorithm by using conjugate directions to achieve an exact solution while exploiting sparsity. Both algorithms lead to faster convergence for certain parameter regimes. Our findings apply beyond PageRank and work for any quadratic objective whose Hessian is a positive-definite M-matrix.

1. Introduction

Graph clustering, the process of dividing a graph into subclusters that are internally similar or connected in some application-specific sense (Schaeffer, 2007), has been widely applied in various

. *Equal contribution.
. Most of the notations in this work have a link to their definitions. For example, if you click or tap on any instance of $\mathbf{x}^*$, you will jump to the place where it is defined as the minimizer of the function we consider in this work.

domains, including technical (Virtanen, 2003; Andersen et al., 2006), biological (Xu et al., 2002; Bader and Hogue, 2003; Boyer et al., 2005), and sociological (Newman, 2003; Traud et al., 2012) settings. With the advent of large-scale networks, traditional approaches that require access to the entire graph have become infeasible (Jeub et al., 2015; Leskovec et al., 2009; Fortunato and Hric, 2016). This trend has led to the development of *local graph clustering algorithms*, which only visit a small subset of vertices of the graph (Andersen et al., 2006; Andersen and Lang, 2008; Mahoney et al., 2012; Spielman and Teng, 2013; Kloster and Gleich, 2014; Orecchia and Zhu, 2014; Veldt et al., 2016; Wang et al., 2017; Yin et al., 2017; Fountoulakis et al., 2019).

At the heart of the study of these algorithms lies the *approximate personalized PageRank algorithm* (`APPR`) (Andersen et al., 2006), which approximates the solution of the PageRank linear system (Page et al., 1999) in an undirected graph and rounds the approximate solution to find local partitions of this graph. The `APPR` algorithm was introduced only from an algorithmic perspective, that is, its output is determined only algorithmically and not formulated as the solution to an optimization problem. Thus, quantifying the impact of heuristic modifications on the method is difficult, see, for example, (Gleich and Mahoney, 2014). Recently, Fountoulakis et al. (2019) proposed a variational formulation of the local graph clustering problem as an ℓ_1-regularized convex optimization problem, which they solved using the *iterative shrinkage-thresholding algorithm* (`ISTA`) (Parikh et al., 2014). In this problem, `ISTA` was shown to exhibit local behaviour, which leads to a running time that only depends on the nodes that are part of the solution and its neighbors, and is independent of the size of the graph. Fountoulakis and Yang (2022) raised the open question of whether accelerated versions of the `ISTA`-based approach or other acceleration techniques, for example, the *fast iterative shrinkage-thresholding algorithm* (`FISTA`) (Parikh et al., 2014), or *linear coupling* (Allen-Zhu and Orecchia, 2019), could lead to faster local graph clustering algorithms. In particular, `ISTA` enjoys low per-iteration complexity since its iterates are at least as sparse as the solution, and the question is whether we can attain acceleration and reduce the dependence on the condition number on the computational complexity, while keeping sparse per-iteration updates.

Sparse Algorithms and Acceleration. In this work, we answer the question in the affirmative. We first study the problem beyond acceleration and propose a method based on conjugate directions that optimizes exactly and is faster than `ISTA` and our accelerated algorithm in some parameter regimes. Then, we show that we can implement an approximate version of the previous method by means of acceleration while performing sparse updates, which leads to faster convergence for ill-conditioned problems, among others. See Table 1 for a summary of the complexities of our algorithms and of prior work, and see Appendix B for a discussion comparing these complexities. Our algorithms sequentially determine the coordinates in the support of the solution. The main differences between the two approaches are that the conjugate-directions-based approach solves the problem in increasing subspaces exactly and requires to incorporate new coordinates one by one, while the accelerated algorithm solves this approximately and can add any number of new coordinates at a time. Beyond the PageRank problem, our algorithms apply to the quadratic problem $\min_{x\in\mathbb{R}^n_{\geq 0}}\{g(\mathbf{x}) \stackrel{\text{def}}{=} \langle \mathbf{x}, Q\mathbf{x}\rangle - \langle \mathbf{b}, \mathbf{x}\rangle\}$, where Q is a symmetric positive-definite M-matrix.

Problem Structure. The rates achieved with our two methods exploit improved geometric understanding of the ℓ_1-regularized PageRank problem structure that we present. In particular, the ℓ_1-regularized problem can be posed as a problem constrained to the positive orthant $\mathbb{R}^n_{\geq 0}$. Based on this formulation, we characterize a region of points for which a negative gradient coordinate i

indicates i is in the support $\mathcal{S}^*$ of the optimal solution $\mathbf{x}^*$, we provide sufficient conditions for finding points in this region with negative gradient coordinates, and show coordinatewise monotonicity of minimizers restricted to some relevant increasing subspaces, among other things.

Table 1: Convergence rates of different algorithms exploiting sparsity for the ℓ_1-regularized PageRank problem and other more general quadratic optimization problems with Hessian Q, condition number L/α, $\mathcal{S}^* \stackrel{\text{def}}{=} \operatorname{supp}(\mathbf{x}^*)$, $\operatorname{vol}(\mathcal{S}^*) \stackrel{\text{def}}{=} \operatorname{nnz}(Q_{:,\mathcal{S}^*})$ and $\widetilde{\operatorname{vol}}(\mathcal{S}^*) \stackrel{\text{def}}{=} \operatorname{nnz}(Q_{\mathcal{S}^*,\mathcal{S}^*})$.

Method	**Time complexity**	**Space complexity**
`ISTA` (Fountoulakis et al., 2019)	$\widetilde{O}(\operatorname{vol}(\mathcal{S}^*)\frac{L}{\alpha})$	$O(\lvert\mathcal{S}^*\rvert)$
`CDPR` (Algorithm 2)	$O(\lvert\mathcal{S}^*\rvert^3 + \lvert\mathcal{S}^*\rvert\operatorname{vol}(\mathcal{S}^*))$	$O(\lvert\mathcal{S}^*\rvert^2)$
`ASPR` (Algorithm 4)	$\widetilde{O}(\lvert\mathcal{S}^*\rvert\widetilde{\operatorname{vol}}(\mathcal{S}^*)\sqrt{\frac{L}{\alpha}} + \lvert\mathcal{S}^*\rvert\operatorname{vol}(\mathcal{S}^*))$	$O(\lvert\mathcal{S}^*\rvert)$

1.1. Other Related Works

Our solutions make use of first-order methods: accelerated projected gradient descent (Nesterov, 1998) and the method of conjugate directions (Nocedal and Wright, 1999). First-order optimization methods are attractive in the high-dimensional regime, due to their fast per-iteration complexity in comparison to higher order methods. In the strongly convex and smooth case, accelerated gradient descent is an optimal first-order method (Nesterov, 1998) and it improves over gradient descent by reducing its dependence on the condition number. Because of this reason, accelerated gradient descent is especially useful for ill-conditioned problems. A method related to the conjugate directions method is the conjugate gradients algorithm (Nocedal and Wright, 1999). Both of these conjugate methods can work in affine subspaces (Gower, 2014), but to the best of our knowledge, it is not know how to provably use these algorithms with other kinds of constraints, see (Vollebregt, 2014) and references therein. For quadratic objectives, the conjugate gradient algorithm is also an accelerated method, and it belongs to the family of Krylov subspace methods, of which the generalized minimal residual method is an important example (Saad and Schultz, 1986). In fact, the conjugate gradient algorithm was the inspiration for the first nearly-accelerated method for smooth convex optimization by Arkadi Nemirovski (1981). Conjugate methods have been used to solve linear systems (Saad, 2003) and although these methods are known to exploit the sparsity of the matrix, to the best of our knowledge there are no analyses of conjugate methods that exploit the sparsity of the solution.

For the ℓ_1-regularized PageRank problem, Hu (2020) demonstrated through numerical experiments that the updates generated by `FISTA` do not exhibit the same level of sparsity as those produced by `ISTA` for this type of problem. To the best of our knowledge, no other works have studied the open question raised by Fountoulakis and Yang (2022).

1.2. Preliminaries

In this section, we introduce some definitions and notation to be used in the rest of this work.

Throughout, let $n \in \mathbb{N}$. We use $[n] = \{1, 2, \dots, n\}$. We use the big-$\mathcal{O}$ notation $\widetilde{\mathcal{O}}(\cdot)$ to omit logarithmic factors. Let $\mathbb{1}_n \in \mathbb{R}^n$ denote the all-ones vector. Denote the support of a vector $\mathbf{x} \in \mathbb{R}^n$ by $\operatorname{supp}(\mathbf{x}) = \{i \in [n] \mid x_i \neq 0\}$ and define the projection of $\mathbf{x} \in \mathbb{R}^n$ onto a convex subset $C \subseteq \mathbb{R}^n$ by $\operatorname{Proj}_C(\mathbf{x}) = \arg\min_{\mathbf{y} \in C} \|\mathbf{x} - \mathbf{y}\|_2$. For $i \in [n]$, we use $\mathbf{e}_i \in \mathbb{R}^n$ to denote the i-th unit vector and Δ^n to denote the n-dimensional simplex. For $S \subseteq [n]$, and a function $f : \mathbb{R}^n \to \mathbb{R}$, let $\nabla_S f(\mathbf{x})$ be the vector containing $(\nabla_i f(\mathbf{x}))_{i \in S}$ sorted by index. Throughout, $Q \in \mathcal{M}_{n \times n}(\mathbb{R})$ is always a symmetric positive-definite matrix with non-positive off-diagonal entries, that is, a symmetric M-matrix such that $Q \succ 0$. In this work, for one such matrix Q and a vector $\mathbf{b} \in \mathbb{R}^n$, we study the optimization of a quadratic of the form $g(\mathbf{x}) \stackrel{\text{def}}{=} \langle \mathbf{x}, Q\mathbf{x} \rangle - \langle \mathbf{b}, \mathbf{x} \rangle$ constrained to the positive orthant $\mathbb{R}^n_{\geq 0}$. By strong convexity, the solution is unique. In the sequel, we focus on optimization algorithms for this problem whose iterates always have support contained in the support of the optimal solution $\mathbf{x}^* \stackrel{\text{def}}{=} \arg\min_{\mathbf{x} \in \mathbb{R}^n_{\geq 0}} g(\mathbf{x})$. We define $\mathcal{S}^* \stackrel{\text{def}}{=} \operatorname{supp}(\mathbf{x}^*)$. We refer to coordinates $i \in [n]$ as good if $i \in \mathcal{S}^*$, and as bad otherwise. We denote by L and α upper and lower bounds on the eigenvalues of Q, that is, smoothness and strong convexity constants of g defined as above, respectively. In short, we have $0 \prec \alpha I \preccurlyeq \nabla^2 g(\mathbf{x}) \preccurlyeq LI$, for $\mathbf{x} \in \mathbb{R}^n$.

Throughout, $G = (V, E)$ is an undirected graph with vertex and edge sets V and E, respectively. We assume that $|V| = n$, that is, G consists of n vertices. Given two vertices $i, j \in [n]$, $i \backsim j$ denotes that they are neighbours. For $S \subseteq V$, $i \backsim S$ indicates that i is the neighbour of at least one node in S. As we describe in the next section, in PageRank problems, the matrix Q corresponds to a combination of the Laplacian of a graph and the identity matrix I. For a subset of vertices $S \subseteq V$, we formally define the volume of S as $\operatorname{vol}(S) = \sum_{i \in S} d_i + |S|$, that is, as the sum of the degrees of vertices in S, plus $|S|$, to account for the self loops induced by regularization, that presents a similar effect to lazyfying the walk given by the graph. Similarly, we formally define the internal volume of S as $\widetilde{\operatorname{vol}}(S) \stackrel{\text{def}}{=} |S| + \sum_{(i,j) \in E} \mathbf{1}_{\{i,j \in S\}}$, that is, as the sum of edges of the subgraph induced by S, plus $|S|$, to account for the regularization. This definition corresponds to $\operatorname{vol}(S) = \operatorname{nnz}(Q_{:,\mathcal{S}^*})$ and $\widetilde{\operatorname{vol}}(S) = \operatorname{nnz}(Q_{\mathcal{S}^*,\mathcal{S}^*})$, where $\operatorname{nnz}(\cdot)$ refers the number of non-zeros of a matrix, $Q_{:,\mathcal{S}^*}$ refers to the columns of Q indexed by $\mathcal{S}^*$ and $Q_{\mathcal{S}^*,\mathcal{S}^*}$ to the submatrix with entries $Q_{i,j}$ for $i, j \in \mathcal{S}^*$. This is the formal definition of $\operatorname{vol}(\cdot)$ and $\widetilde{\operatorname{vol}}(\cdot)$ that we use when working with a general M-matrix Q. The complexity of our results depends on $\operatorname{vol}(\mathcal{S}^*)$ and $\widetilde{\operatorname{vol}}(\mathcal{S}^*)$. Fountoulakis et al. (2019) showed that for the ℓ_1-regularized PageRank problem it is $\sum_{i \in \mathcal{S}^*} d_i \leq \frac{1}{\rho}$ and therefore $\operatorname{vol}(\mathcal{S}^*) \leq \frac{1}{\rho} + |\mathcal{S}^*|$, where ρ is the regularization parameter of the problem, see for example (1).

2. Personalized PageRank with ℓ_1-Regularization

In this section, we introduce the PageRank problem that we study in this work, and we recall the variational formulation due to Fountoulakis et al. (2019). Let $G = (V, E)$ be a connected undirected graph with n vertices. We note that there are techniques to reduce an unconnected PageRank problem to a connected one, see for example Eiron et al. (2004). The PageRank problem is motivated by trying to find the stationary distribution $x \in \Delta^n$ of the uniform random walk $AD^{-1}x = x$, where $D \stackrel{\text{def}}{=} \operatorname{diag}(d_1, \dots, d_n)$ is the matrix with the degrees $\{d_i\}_{i=1}^n$ in its diagonal and A is the adjacency matrix of G, that is, $A_{i,j} = 1$ if $i \backsim j$ and 0 otherwise. Similarly for a weighted random walk. We want uniqueness of the stationary distribution, for convergence and robustness purposes, and this requires to essentially make the chain strongly connected, that is, irreducible. To that aim, given $\alpha \in (0, 1)$,

in the general directed Personalized PageRank problem we change the Markov chain to have the transition matrix $(1-\alpha)AD^{-1}+\alpha s\mathbb{1}_n{}^T$, where $\mathbf{s}\in\Delta$ is called the teleportation distribution, because it makes the random walk to teleport with probability α to a random node dictated by the distribution s. The original PageRank problem had $\mathbf{s}=\mathbb{1}_n/n$ and $\alpha\approx 0.15$, and in general they can be chosen arbitrarily as long as it yields irreducibility, such as in the fully supported case ($s_i>0$ for all $i\in[n]$). In our undirected PageRank problem, we already have uniqueness of the stationary distribution, by connectedness. However, the teleportation distribution is still used to bias the solution toward results similar to what you are interested in and it can be an arbitrary $\mathbf{s}\in\Delta$. For example, if we set $\mathbf{s}=e_i$, we compute a cluster around node i (Fountoulakis et al., 2019). It is not necessary, but it could be desirable to lazify the walk, for instance to $\frac{1}{2}(I+AD^{-1})$, in order to ensure aperiodicity of the chain which implies the chain converges. Note that a distribution is stationary for the Markov chain if and only if it is a stationary distribution for the lazified walk. We use this lazy version.

So finally, using $\mathbf{x}\in\Delta$, we are interested in approximately solving the system $(I+AD^{-1})\frac{1-\alpha}{2}\mathbf{x}+\alpha\mathbf{s}=\mathbf{x}$, which after multiplying by $D^{-1/2}$ on the right and rescaling $\mathbf{x}$ to $D^{1/2}\mathbf{x}$ results in the problem of finding a point $\mathbf{x}$ with approximately zero gradient for the quadratic objective $f(\mathbf{x})\stackrel{\text{def}}{=}\frac{1}{2}\langle\mathbf{x},Q\mathbf{x}\rangle-\alpha\langle D^{-1/2}\mathbf{s},\mathbf{x}\rangle$, where $Q=\alpha I+\frac{1-\alpha}{2}\mathcal{L}$, and where $\mathcal{L}\stackrel{\text{def}}{=}I-D^{-1/2}AD^{-1/2}$ is the symmetric normalized Laplacian matrix, which is known to satisfy $0\prec\mathcal{L}\preccurlyeq 2I$ (Butler et al., 2006). By construction, $0\prec\alpha I\preccurlyeq Q\preccurlyeq LI$, for $L=1$, so f is α-strongly convex and L-smooth. Note that $Q_{i,j}\le 0$ for $i\neq j$, so indeed Q is a positive definite M-matrix, which is what our algorithms require. This problem was tackled by the APPR algorithm (Andersen et al., 2006) from an algorithmic perspective and Fountoulakis et al. (2019) showed that for $\rho>0$, if we approximately optimize the ℓ_1-regularized problem

$$\min_{\mathbf{x}\in\mathbb{R}^n} f(\mathbf{x})+\alpha\rho\|D^{1/2}\mathbf{x}\|_1, \tag{1}$$

we obtain the same guarantees as APPR. This variational formulation allows to address the problem from an optimization perspective and to reason about guarantees on the sparsity of the solution. For instance, Fountoulakis et al. (2019) showed that $\sum_{i\in\mathcal{S}^*}d_i\lesssim 1/\rho$ and then, assuming $|\mathcal{S}^*|=O(\sum_{i\in\mathcal{S}^*}d_i)$ they conclude that ISTA finds an ε-minimizer after $\widetilde{O}(\mathrm{vol}(\mathcal{S}^*)\frac{L}{\alpha})=\widetilde{O}(\frac{L}{\rho\alpha})$ operations. Due to the strong convexity of the objective, (1) has a unique minimizer $\mathbf{x}^*$, and it is in $\mathbb{R}_{\ge 0}$, since the linear system above, that is $\arg\min_{x\in\mathbb{R}^n} f(\mathbf{x})$, was defined so its solution is a distribution. Letting

$$g(\mathbf{x})\stackrel{\text{def}}{=}f(\mathbf{x})+\alpha\langle\rho D^{1/2}\mathbb{1}_n,\mathbf{x}\rangle=\frac{1}{2}\langle\mathbf{x},Q\mathbf{x}\rangle+\alpha\langle-D^{-1/2}\mathbf{s}+\rho D^{1/2}\mathbb{1}_n,\mathbf{x}\rangle, \tag{2}$$

the optimality conditions for $\arg\min_{\mathbf{x}\in\mathbb{R}^n_{\ge 0}} g(\mathbf{x})$ are equivalent to the optimality conditions of Problem (1) and we have

$$\min_{\mathbf{x}\in\mathbb{R}^n} f(\mathbf{x})+\alpha\rho\|D^{1/2}\mathbf{x}\|_1=\min_{\mathbf{x}\in\mathbb{R}^n_{\ge 0}} g(\mathbf{x}). \tag{3}$$

The algorithms presented in this work apply in particular to the minimization of g defined in (2).

2.1. Projected Gradient Descent (PGD)

Fountoulakis et al. (2019) tackled Problem (1) by applying ISTA to it, initialized at $\mathbb{0}$, and they showed that each iterate $\mathbf{x}^{(t)}$ of the algorithm satisfies $\mathbf{x}^{(t)}\ge\mathbb{0}$. Given $\mathbf{x}^{(t-1)}$, the update rule of

Algorithm 1 Projected gradient descent (PGD)

Input: Closed and convex set $C \subseteq \mathbb{R}^n$, initial point $\mathbf{x}^{(0)} \in C$, $f\colon C \to \mathbb{R}$ an α-strongly convex and L-smooth function, and $T \in \mathbb{N}$.
Output: $\mathbf{x}^{(T)} \in C$.

1: **for** $t = 0, 1, \ldots, T-1$ **do**
2: $\quad \mathbf{x}^{(t+1)} \leftarrow \mathrm{Proj}_C\left(\mathbf{x}^{(t)} - \frac{1}{L}\nabla f(\mathbf{x}^{(t)})\right)$
3: **end for**

ISTA defines the next iterate as $\mathbf{x}^{(t)} \stackrel{\text{def}}{=} \arg\min_{\mathbf{x}\in\mathbb{R}^n} \rho\alpha\|D^{1/2}\mathbf{x}\|_1 + \frac{1}{2}\|\mathbf{x} - (\mathbf{x}^{(t-1)} - \nabla_i f(\mathbf{x}^{(t-1)}))\|_2^2 = \arg\min_{\mathbf{x}\in\mathbb{R}^n_{\geq 0}} \frac{1}{2}\|\mathbf{x} - (\mathbf{x}^{(t-1)} - \nabla g(\mathbf{x}^{(t-1)}))\|^2$, where the equality follows directly by checking each coordinate, since the problems are separable. We note that the right hand side is the optimization problem that defines PGD for g in $\mathbb{R}^n_{\geq 0}$. We present projected gradient descent (PGD) in Algorithm 1, which will be useful to our analysis. None of our algorithms for addressing (3) run PGD as a subroutine. The application of PGD to the set $C \subseteq \mathbb{R}^n$, initial point $\mathbf{x}^{(0)} \in C$, objective $f\colon C \to \mathbb{R}$, and number of iterations $T \in \mathbb{N}$ is denoted by $\mathbf{x}^{(T)} = \texttt{PGD}(C, \mathbf{x}^{(0)}, f, T)$.

Fact 1 (Convergence rate of PGD) *Let $C \subseteq \mathbb{R}^n$ be a closed convex set, $\mathbf{x}^{(0)} \in C$, and $f\colon C \to \mathbb{R}$ an α-strongly convex and L-smooth function with minimizer at $\mathbf{x}^*$. Then, for the iterates of Algorithm 1, it holds that $\|\mathbf{x}^{(t)} - \mathbf{x}^*\|_2^2 \leq \left(1 - \frac{1}{\kappa}\right)^t \|\mathbf{x}^{(0)} - \mathbf{x}^*\|_2^2$, where $\kappa \stackrel{\text{def}}{=} \frac{L}{\alpha}$. See Nesterov (1998, Theorem 2.2.8) for a proof.*

2.2. Geometrical Understanding of the Problem Setting

Fountoulakis et al. (2019) proved that for their method, the iterates $\mathbf{x}^{(t)}$ never decrease coordinate-wise, and they concluded $\mathbf{x}^* \in \mathbb{R}^n_{\geq 0}$ as a consequence of this fact and the convergence guarantees of ISTA: $\mathbb{0} \leq \mathbf{x}^{(1)} \leq \cdots \leq \mathbf{x}^{(t)} \leq \mathbf{x}^{(t+1)} \to \mathbf{x}^*$. We generalize this result proven for the iterates of ISTA to several geometric statements on the problem. This result holds in a more general setting, namely a quadratic with a positive-definite M-matrix as Hessian. The proof illustrates the geometry of the problem and we include it below. For any point $\mathbf{x}$ such that $x_i = 0$ if $i \notin \mathcal{S}^*$ and $\nabla_i g(\mathbf{x}) \leq 0$ if $i \in \mathcal{S}^*$, we have that $\mathbf{x} \leq \mathbf{x}^*$, among other things.

Proposition 2 *Let g be as in (2) and let $S \subseteq [n]$ be a set of indices such that we have a point $\mathbf{x}^{(0)} \in \mathbb{R}^n$ with $\mathrm{supp}(\mathbf{x}^{(0)}) \subseteq S$ and $\nabla_i g(\mathbf{x}^{(0)}) \leq 0$ for all $i \in S$. Let $C \stackrel{\text{def}}{=} \mathrm{span}(\{\mathbf{e}_i \mid i \in S\}) \cap \mathbb{R}^n_{\geq 0}$, $\mathbf{x}^{(*,C)} \stackrel{\text{def}}{=} \arg\min_{\mathbf{x}\in C} g(\mathbf{x})$ and $\mathbf{x}^* \stackrel{\text{def}}{=} \arg\min_{\mathbf{x}\in\mathbb{R}^n_{\geq 0}} g(\mathbf{x})$. Then:*

1. *It holds that $\mathbf{x}^{(0)} \leq \mathbf{x}^{(*,C)}$ and $\nabla_i g(\mathbf{x}^{(*,C)}) = 0$ for all $i \in S$.*
2. *If for $i \in S$, we have $x_i^{(0)} > 0$ or $\nabla_i g(\mathbf{x}^{(0)}) < 0$, then $x_i^{(*,C)} > 0$.*
3. *If $x_i^{(*,C)} > 0$ for all $i \in S$, we have $\mathbf{x}^{(*,C)} \leq \mathbf{x}^*$ and therefore $S \subseteq \mathcal{S}^*$.*

Proof First, by definition of C, for all $\mathbf{x} \in C$, we have $x_i = 0$ if $i \notin S$. Let $\{\mathbf{x}^{(t)}\}_{t=0}^{\infty}$ be the sequence of iterates created by $\texttt{PGD}(C, \mathbf{x}^{(0)}, g, \cdot)$ when the algorithm is run for infinitely many iterations. We

first prove that for all $t \geq 0$ and for all $i \in S$, we have $\nabla_i g(\mathbf{x}^{(t)}) \leq 0$. It holds for $t = 0$ by assumption. If we assume it holds for some $t \geq 0$, then we have

$$x_i^{(t+1)} = x_i^{(t)} - \frac{1}{L}\nabla_i g(\mathbf{x}^{(t)}) \geq x_i^{(t)} \tag{4}$$

for all $i \in S$, that is, the points do not decrease coordinatewise. Let the function $\bar{g}$ be g restricted to $\operatorname{span}(\{\mathbf{e}_i \mid i \in S\})$ and note $\nabla_i g(\mathbf{x}) = \nabla_i \bar{g}(\mathbf{x})$ for $i \in S$. The function $\bar{g}$ is a quadratic with Hessian $Q_{S,S}$, that is, it is formed by $Q_{i,j}$ for $i, j \in S$. Quadratics have affine gradients and so we have by (4) that $\nabla \bar{g}(\mathbf{x}^{(t+1)}) = \nabla \bar{g}(\mathbf{x}^{(t)}) - \frac{1}{L} Q_{S,S} \nabla \bar{g}(\mathbf{x}^{(t)}) \leq 0$, where the last inequality is due to the assumption $\nabla \bar{g}(\mathbf{x}^{(t)}) \leq 0$, and $(I - \frac{1}{L} Q_{S,S})_{i,j} \geq 0$ for all $i, j \in S$. The latter holds because for $i, j \in S$, $i \neq j$, we have $Q_{i,j} \leq 0$ and due to smoothness, it is $Q_{i,i} = e_i^\intercal Q e_i \leq L$. Thus, by induction, for all $t \in \mathbb{N}$ and $i \in S$, we have $\nabla_i g(\mathbf{x}^{(t)}) \leq 0$. This has two consequences. Firstly, $\mathbf{x}^{(0)} \leq \mathbf{x}^{(1)} \leq \ldots$, and so $\mathbf{x}^{(0)} \leq \mathbf{x}^{(*,C)}$ since the iterates of PGD converge to $\mathbf{x}^{(*,C)}$, by Fact 1. Secondly, using the limit and continuity of $\nabla g(\cdot)$, it is $\nabla_i g(\mathbf{x}^{(*,C)}) \leq 0$ for $i \in S$. This fact and the optimality of $\mathbf{x}^{(*,C)}$ imply $\nabla_i g(\mathbf{x}^{(*,C)}) = 0$ for all $i \in S$, proving the first statement.

For the second stament, fix $i \in S$. Note that by the assumption and the update rule $x_i^{(t+1)} = x_i^{(t)} - \frac{1}{L}\nabla_i g(\mathbf{x}^{(t)})$, it holds that $x_i^{(1)} > 0$, and thus, since $x_i^{(*,C)} \geq x_i^{(1)}$, we have $x_i^{(*,C)} > 0$.

For the third statement, we sequentially apply the first one to obtain optimizers in increasing subspaces, until we reach $\mathbf{x}^*$, while showing they do not decrease coordinatewise. Suppose that $x_i^{(*,C)} > 0$ for all $i \in S$. If $\mathbf{x}^{(*,C)} = \mathbf{x}^*$, the statement holds. Thus, we assume that $\mathbf{x}^{(*,C)} \neq \mathbf{x}^*$. In that case, for $k \in \mathbb{N}$, define the optimizer $\mathbf{y}^{(*,k)} \stackrel{\text{def}}{=} \arg\min_{\mathbf{y} \in B^{(k-1)}} g(\mathbf{y})$ with respect to the set $B^{(k-1)} \stackrel{\text{def}}{=} \operatorname{span}(\{\mathbf{e}_i \mid i \in R^{(k-1)}\}) \cap \mathbb{R}^n_{\geq 0}$, where $R^{(k-1)} \stackrel{\text{def}}{=} R^{(k-2)} \cup N^{(k-1)}$ for $k > 0$ and $R^{(-1)} \stackrel{\text{def}}{=} S$, and where $N^{(k-1)} \stackrel{\text{def}}{=} \{i \in [n] \mid y_i^{(*,k-1)} = 0, \nabla_i g(\mathbf{y}^{(*,k-1)}) < 0\}$. By Property 1, it holds that $\mathbf{x}^{(*,C)} = \mathbf{y}^{(*,0)} \leq \ldots \leq \mathbf{y}^{(*,k)}$ and $\nabla_i g(\mathbf{y}^{(*,k)}) = 0$ for all $i \in R^{(k-1)}$ and $k \in \mathbb{N}$. Let $K \in \mathbb{N}$ denote the first iteration for which $R^{(K)} = R^{(K-1)}$, or, equivalently $N^{(K)} = \emptyset$. The existence of such a K is guaranteed because otherwise $R^{(k)} \subset R^{(k+1)}$ for all $k \in \mathbb{N}$, but necessarily it is $|R^{(k)}| \leq n$. Thus, $\nabla_i g(\mathbf{y}^{(*,K)}) = 0$ for all $i \in R^{(K-1)}$ and $\nabla_i g(\mathbf{y}^{(*,K)}) \geq 0$ for all $i \notin R^{(K-1)}$. In summary, $\mathbf{y}^{(*,K)}$ satisfies the optimality conditions of the problem $\min_{\mathbf{x} \in \mathbb{R}^n_{\geq 0}} g(\mathbf{x})$, implying that $\mathbf{y}^{(*,K)} = \mathbf{x}^*$. Since $S = R^{(-1)} \subseteq R^{(K)}$, Property 3 holds. ∎

2.3. Algorithmic Intuition

In this section, we present the high-level idea of our algorithms for addressing (3). The core idea behind them is to start with the set of known good indices $S^{(-1)} = \emptyset$ and iteratively expand it, $S^{(-1)} \subsetneq S^{(0)} \subsetneq \ldots \subsetneq S^{(T)}$, until we have $S^{(T)} = \mathcal{S}^*$ or we find an ε-minimizer of (3). For $t \in \{0, 1, \ldots, T\}$, to determine elements $i \in \mathcal{S}^* \setminus S^{(t-1)}$, we let

$$\mathbf{x}^{(*,t)} = \arg\min_{\mathbf{x} \in C^{(t-1)}} g(\mathbf{x}), \tag{5}$$

where $C^{(t-1)} \stackrel{\text{def}}{=} \operatorname{span}(\{\mathbf{e}_i \mid i \in S^{(t-1)}\}) \cap \mathbb{R}^n_{\geq 0}$. By an argument following Proposition 2 that we will detail later, $\nabla_i g(\mathbf{x}^{(*,t)}) < 0$ for at least one $i \in \mathcal{S}^* \setminus S^{(t-1)}$ and $\nabla_j g(\mathbf{x}^{(*,t)}) \geq 0$ for all

$j \notin \mathcal{S}^*$. This observation motivates the following procedure: At iteration $t \in \{0, 1, \ldots, T-1\}$, construct $\mathbf{x}^{(*,t)}$, check if $N^{(t)} = \{i \in [n] \mid \nabla_i g(\mathbf{x}^{(*,t)}) < 0\}$ is not empty, and, in such a case, set $S^{(t)} \subseteq S^{(t-1)} \cup N^{(t)}$ and repeat the procedure. Should it ever happen that $N^{(t)} = \emptyset$, then we have $\mathbf{x}^{(*,t)} = \mathbf{x}^*$, that is, we found the optimal solution to (3) and the algorithm can be terminated. When using conjugate directions as the optimization algorithm for constructing (5), and when only incorporating good coordinates one by one, we obtain Algorithm 2 (CDPR), see Section 3. For our second algorithm, Algorithm 4 (ASPR), we use accelerated projected gradient descent to construct only an approximation of (5) and we show that this method still allows us to proceed. We discuss the subtleties arising from using an approximation algorithm in Section 4.

3. Conjugate Directions for PageRank

Algorithm 2 Conjugate directions PageRank algorithm (CDPR)

Input: Quadratic function $g : \mathbb{R}^n \to \mathbb{R}$ with Hessian $Q \succ 0$ being a symmetric M-matrix. The ℓ_1-regularized PageRank problem corresponds to choosing g as in (2).

Output: $\mathbf{x}^{(T)} = \arg\min_{\mathbf{x} \in \mathbb{R}^n_{\geq 0}} g(\mathbf{x})$, where $T \in \mathbb{N}$ is the first iteration for which $N^{(T)} = \emptyset$.

$t \leftarrow 0$
$\mathbf{x}^{(t)} \leftarrow \mathbb{0}$
$N^{(t)} \leftarrow \left\{i \in [n] \mid \nabla_i g(\mathbf{x}^{(t)}) < 0\right\}$
while $N^{(t)} \neq \emptyset$ **do**
  $i^{(t)} \in N^{(t)}$
  $\mathbf{u}^{(t)} \leftarrow \nabla_{i^{(t)}} g(\mathbf{x}^{(t)}) \cdot \mathbf{e}_{i^{(t)}}$
  $\beta_k^{(t)} \leftarrow \nabla_{i^{(t)}} g(\mathbf{x}^{(t)}) \langle Q_{:,i^{(t)}}, \bar{\mathbf{d}}^{(k)} \rangle$ for all $k = 0, \ldots, t-1$ ◇ equal to $-\frac{\langle \mathbf{u}^{(t)}, Q\mathbf{d}^{(k)} \rangle}{\langle \mathbf{d}^{(k)}, Q\mathbf{d}^{(k)} \rangle}$
  $\mathbf{d}^{(t)} \leftarrow \mathbf{u}^{(t)} + \sum_{k=0}^{t-1} \beta_k^{(t)} \mathbf{d}^{(k)}$ ◇ store this sparse vector
  $\bar{\mathbf{d}}^{(t)} \leftarrow \frac{\mathbf{d}^{(t)}}{\langle \mathbf{d}^{(t)}, Q\mathbf{d}^{(t)} \rangle}$ ◇ store $\langle \mathbf{d}^{(t)}, Q\mathbf{d}^{(t)} \rangle$
  $\eta^{(t)} \leftarrow -\langle \nabla g(\mathbf{x}^{(t)}), \bar{\mathbf{d}}^{(t)} \rangle$
  $\mathbf{x}^{(t+1)} \leftarrow \mathbf{x}^{(t)} + \eta^{(t)} \mathbf{d}^{(t)}$ ◇ equal to $\arg\min_{\mathbf{x} \in C^{(t)}} g(\mathbf{x}) = \arg\min_{\eta \in \mathbb{R}} g(\mathbf{x}^{(t)} + \eta \mathbf{d}^{(t)})$
  $N^{(t+1)} \leftarrow \left\{i \in [n] \mid \nabla_i g(\mathbf{x}^{(t+1)}) < 0\right\}$
  $t \leftarrow t + 1$
end while

With the geometric properties of the problem we established in Section 2, we are ready to introduce the conjugate directions PageRank algorithm (CDPR) Algorithm 2, a *conjugate-directions*-based approach for addressing (3), which outperforms the ISTA-solver due to Fountoulakis et al. (2019) in certain parameter regimes. CDPR is based on the algorithmic blueprint outlined in Section 2.3 and constructs $\mathbf{x}^{(*,T)}$ as in (5) using conjugate directions. As we will prove formally, it is $\mathbb{0} \leq \mathbf{x}^{(*,t)}$ for all $t \in \{0, 1, \ldots, T\}$, allowing us to solve the constrained problem (5) by dropping the non-negativity constraints and using the method of conjugate directions. This is an important point, since this method is designed for affine spaces only and, to the best of our knowledge, cannot deal with other constraints. Conjugate directions are an attractive mechanism for finding (5), as it allows to exploit the sparsity of the solution, is exact, and does not rely on the strong convexity of the objective, leading to a time complexity independent of α. Note that, even though we may learn about several

new good coordinates at the end of an iteration, in order to maintain the invariants required for our CDPR, we can add at most one new coordinate to $S^{(t)}$ at a time. This algorithm requires more memory than the ISTA-solver of Fountoulakis et al. (2019) and ASPR, which is due to storing an increasing Q-orthogonal basis that is required to perform exact optimization over $C^{(t)}$ by performing Gram-Schmidt with respect to Q.

Algorithm 2 works in the following way. Initialize with $S^{(-1)} \stackrel{\text{def}}{=} \emptyset$, and $\mathbf{x}^{(*,0)} \stackrel{\text{def}}{=} \mathbb{0}$. For $t \in \{0, 1, \ldots, T\}$, let the set of known good coordinates be $S^{(t)} \stackrel{\text{def}}{=} S^{(t-1)} \cup \{i^{(t)}\}$, and define $C^{(t)} \stackrel{\text{def}}{=} \operatorname{span}(\{\mathbf{e}_i \mid i \in S^{(t)}\}) \cap \mathbb{R}^n_{\geq 0}$, and $\mathbf{x}^{(*,t)} \stackrel{\text{def}}{=} \arg\min_{\mathbf{x} \in C^{(t-1)}} g(\mathbf{x})$. At each iteration $t \in \{0, 1, \ldots, T-1\}$, we start at $\mathbf{x}^{(*,t)} \geq 0$, for which it holds that $\nabla_i g(\mathbf{x}^{(*,t)}) = 0$ for $i \in S^{(t-1)}$ and there exists at least one $i^{(t)} \notin S^{(t-1)}$ such that $\nabla_{i^{(t)}} g(\mathbf{x}^{(*,t)}) < 0$ unless we are already at the optimal solution, that is, $\mathbf{x}^{(*,t-1)} = \mathbf{x}^*$. We arbitrarily select one such index, and then perform Gram-Schmidt with respect to Q in order to obtain $\mathbf{d}^{(t)}$ that is Q-orthogonal to all $\mathbf{d}^{(k)}$ for $k < t$. Next, one can see that the optimizer $\mathbf{x}^{(t+1)}$ along the line $\mathbf{x}^{(t)} + \eta^{(t)}\mathbf{d}^{(t)}$ results in the optimizer for the subspace $\operatorname{span}(\{\mathbf{e}_i \mid i \in S^{(t)}\})$, which is $\mathbf{x}^{(*,t+1)} \geq 0$. After $|\mathcal{S}^*|$ iterations, we obtain $\mathbf{x}^*$. We formalize and prove the claims of the overview below.

Theorem 3 [↓] *For all $t \in \{0, 1, \ldots, T\}$ and $k \in \{0, 1, \ldots, t-1\}$, the following properties are satisfied for Algorithm 2:*

1. *It holds that $\langle \mathbf{d}^{(t)}, Q\mathbf{d}^{(k)} \rangle = 0$.*
2. *We have that $\langle \nabla g(\mathbf{x}^{(t)}), \mathbf{d}^{(k)} \rangle = 0$ and $\nabla_i g(\mathbf{x}^{(t)}) = 0$ for all $i \in S^{(t-1)}$.*
3. *It is $\mathbf{x}^{(t)}_i > 0$ for all $i \in S^{(t-1)}$, and $\mathbb{0} = \mathbf{x}^{(0)} = \mathbf{x}^{(*,0)} \leq \mathbf{x}^{(1)} = \mathbf{x}^{(*,1)} \leq \ldots \leq \mathbf{x}^{(T)} = \mathbf{x}^{(*,T)}$.*
4. *It holds that $\mathbf{x}^{(T)} = \mathbf{x}^*$.*

Unlike our next algorithm, ASPR, the time complexity of Algorithm 2 does not depend on α, L, or ε, and we optimize exactly. We detail the computational complexities of our algorithm below.

Theorem 4 (Computational complexities) [↓] *The time complexity of Algorithm 2 is $O(|\mathcal{S}^*|^3 + |\mathcal{S}^*|\mathrm{vol}(\mathcal{S}^*))$ and its space complexity is $O(|\mathcal{S}^*|^2)$.*

4. Accelerated Sparse PageRank

In this section, we introduce the accelerated sparse PageRank algorithm (ASPR) in Algorithm 4, which is an approach based on accelerated projected gradient descent (APGD) for addressing (3). Let $\mathbf{x}^{(*,0)} = \mathbb{0}$, let $S^{(-1)} = \emptyset$, and for $t \in [T]$, let $\mathbf{x}^{(*,t)} = \arg\min_{\mathbf{x} \in C^{(t-1)}} g(\mathbf{x})$. We now explain the necessary modifications to the exact algorithm outlined in Section 2.3 such that an approximate solver of (5) can be incorporated. First, we recall the convergence of accelerated projected gradient descent (APGD) (Nesterov, 1998) in Algorithm 3, which is used as a subroutine in Algorithm 4. APGD applied to the set $C \subseteq \mathbb{R}^n$, initial point $\mathbf{x}^{(0)} \in C$, objective $f : C \to \mathbb{R}$, and number of iterations $T \in \mathbb{N}$ is denoted by $\mathbf{x} \leftarrow \texttt{APGD}(C, \mathbf{x}^{(0)}, f, T)$. For strongly convex objectives, APGD enjoys the following convergence rate.

Algorithm 3 Accelerated projected gradient descent (APGD)

Input: Closed and convex set $C \subseteq \mathbb{R}^n$, initial point $\mathbf{x}^{(0)} \in C$, $f : C \to \mathbb{R}$ an α-strongly convex and L-smooth function, condition number $\kappa \stackrel{\text{def}}{=} L/\alpha$, and $T \in \mathbb{N}$.

Output: $\mathbf{y}^{(T)} \in C$.

$\mathbf{z}^{(0)} \leftarrow \mathbf{y}^{(0)} \leftarrow \mathbf{x}^{(0)}$; $A_0 \leftarrow 0$; $a_0 \leftarrow 1$

for $t = 0, 1, \ldots, T-1$ **do**

  $A_{t+1} \leftarrow A_t + a_t$ ⋄ equal to $A_t(\frac{2\kappa}{2\kappa+1-\sqrt{1+4\kappa}}) \geq A_t(1-\frac{1}{2\sqrt{\kappa}})^{-1}$ if $t \geq 1$

  $\mathbf{x}^{(t+1)} \leftarrow \frac{A_t}{A_{t+1}}\mathbf{y}^{(t)} + \frac{a_t}{A_{t+1}}\mathbf{z}^{(t)}$

  $\mathbf{z}^{(t+1)} \leftarrow \text{Proj}_C\left(\frac{\kappa-1+A_t}{\kappa-1+A_{t+1}}\mathbf{z}^{(t)} + \frac{a_t}{\kappa-1+A_{t+1}}\left(\mathbf{x}^{(t+1)} - \frac{1}{\alpha}\nabla f(\mathbf{x}^{(t+1)})\right)\right)$

  $\mathbf{y}^{(t+1)} \leftarrow \frac{A_t}{A_{t+1}}\mathbf{y}^{(t)} + \frac{a_t}{A_{t+1}}\mathbf{z}^{(t+1)}$

  $a_{t+1} \leftarrow A_{t+1}(\frac{2\kappa}{2\kappa+1-\sqrt{1+4\kappa}} - 1)$

end for

Proposition 5 (Convergence rate of APGD) [↓] *Let $C \subseteq \mathbb{R}^n$ be a closed convex set, $\mathbf{x}^{(0)} \in C$, and $f : C \to \mathbb{R}$ an α-strongly convex and L-smooth function with minimizer $\mathbf{x}^*$. Then, for the iterates of Algorithm 3, it holds that $f(\mathbf{y}^{(t)}) - f(\mathbf{x}^*) \leq (1-\frac{1}{2\sqrt{\kappa}})^{t-1}\frac{(L-\alpha)\|\mathbf{x}^{(0)}-\mathbf{x}^*\|^2}{2}$, for $\kappa \stackrel{\text{def}}{=} \frac{L}{\alpha}$. We thus obtain an ε-minimizer in $T = 1 + \lceil 2\sqrt{\kappa}\log(\frac{(L-\alpha)\|\mathbf{x}^{(0)}-\mathbf{x}^*\|^2}{2\varepsilon})\rceil \leq 1 + \lceil 2\sqrt{\kappa}\log(\frac{(L-\alpha)\|\nabla f(\mathbf{x}^{(0)})\|_2^2}{2\varepsilon\alpha^2})\rceil$ iterations.*

As in the Algorithm 2 in the previous section, our Algorithm 4 constructs a sequence of subsets $S^{(t)}$ of the support of $\mathbf{x}^*$. In contrast to CDPR, Algorithm 4 does not compute $\mathbf{x}^{(*,t+1)}$ for $t \in \{0, 1, \ldots, T-1\}$ exactly, but instead employs APGD as a subroutine to construct a point $\bar{\mathbf{x}}^{(t+1)}$ that is close enough to $\mathbf{x}^{(*,t+1)}$, and then reduces all positive entries of $\bar{\mathbf{x}}^{(t+1)}$ slightly, obtaining $\mathbf{x}^{(t+1)} \leq \mathbf{x}^{(*,t+1)}$. The following Lemma 6 establishes that if a coordinate of a point is decreased, the gradient of g at all other coordinates does not decrease, implying that for all points $\mathbf{x} \in \mathbb{R}^n_{\geq 0}$ satisfying $\mathbf{x} \leq \mathbf{x}^{(*,t+1)}$, no bad coordinate has a negative gradient.

Lemma 6 [↓] *Let $\mathbf{x} \in \mathbb{R}^n$, and let $\mathbf{y} = \mathbf{x} - \varepsilon\mathbf{e}_i$, for some $\varepsilon > 0$, $i \in [n]$. Then, for all $j \in [n] \setminus \{i\}$, it holds that $\nabla_j g(\mathbf{y}) \geq \nabla_j g(\mathbf{x})$. If instead $\varepsilon < 0$, then $\nabla_j g(\mathbf{y}) \leq \nabla_j g(\mathbf{x})$.*

The second part of Lemma 6 implies that we only have $\nabla_i g(\mathbf{x}^{(t+1)}) < 0$ for coordinates i for which $\nabla_i g(\mathbf{x}^{(*,t+1)}) < 0$, but it suggests that there could be none satisfying the former. To address this issue, APGD is run to sufficient accuracy to guarantee that $g(\mathbf{x}^{(t+1)}) - g(\mathbf{x}^{(*,t+1)}) \leq \frac{\varepsilon\alpha}{L}$. Then, we show that either $g(\mathbf{x}^{(t+1)}) - g(\mathbf{x}^*) \leq \varepsilon$ or one step of PGD from $\mathbf{x}^{(t+1)}$ would make more progress than what we can do in the current space $C^{(t)}$, of which $\mathbf{x}^{(*,t+1)}$ is minimizer, and so the gradient contains a negative entry. All such entries are good coordinates $i \in S^* \setminus S^{(t)}$, similarly to what we had at $\mathbf{x}^{(*,t+1)}$ in CDPR. We note that unlike for CDPR, this time we can incorporate all of these coordinates at once to the algorithm. In Theorem 7 below, we address all these challenges associated with computing $\mathbf{x}^{(t+1)}$ in Algorithm 4 in lieu of $\mathbf{x}^{(*,t+1)}$, and we prove that indeed Algorithm 4 finds an ε-minimizer of g, while all the iterates are sparse, if the solution $\mathbf{x}^*$ is sparse.

Algorithm 4 Accelerated sparse PageRank algorithm (ASPR)

Input: Quadratic function $g : \mathbb{R}^n \to \mathbb{R}$ with Hessian $Q \succ 0$ being a symmetric M-matrix, accuracy $\varepsilon > 0$. The ℓ_1-regularized PageRank problem corresponds to choosing g as in (2).

Output: $\mathbf{x}^{(T)}$, where $T \in \mathbb{N}$ is the first iteration for which $S^{(T)} = S^{(T-1)}$.

$t \leftarrow 0$

$\mathbf{x}^{(0)} \leftarrow \mathbb{0}$

$S^{(t)} \leftarrow \{i \in [n] \mid \nabla_i g(\mathbf{x}^{(t)}) < 0\}$

while $S^{(t)} \neq S^{(t-1)}$ **do**

  $\delta_t \leftarrow \sqrt{\frac{\varepsilon\alpha}{(1+|S^{(t)}|)L^2}}$

  $\hat{\varepsilon}_t \leftarrow \frac{\delta_t^2 \alpha}{2} = \frac{\varepsilon\alpha^2}{2(1+|S^{(t)}|)L^2}$

  $C^{(t)} \leftarrow \operatorname{span}(\{\mathbf{e}_i \mid i \in S^{(t)}\}) \cap \mathbb{R}^n_{\geq 0}$

  $\bar{\mathbf{x}}^{(t+1)} \leftarrow \texttt{APGD}\left(C^{(t)}, \mathbf{x}^{(t)}, g, 1 + \left\lceil 2\sqrt{\kappa} \log\left(\frac{(L-\alpha)\|\nabla_{S^{(t)}} g(\mathbf{x}^{(t)})\|_2^2}{2\hat{\varepsilon}_t \alpha^2}\right)\right\rceil\right)$

  $\mathbf{x}^{(t+1)} \leftarrow \max\{\mathbb{0}, \bar{\mathbf{x}}^{(t+1)} - \delta_t \mathbb{1}_n\}$ ⋄ coordinatewise max, only needed for $i \in S^{(t)}$

  $S^{(t+1)} \leftarrow S^{(t)} \cup \{i \in [n] \mid \nabla_i g(\mathbf{x}^{(t+1)}) < 0\}$

  $t \leftarrow t + 1$

end while

Theorem 7 [↓] *Let $S^{(-1)} \stackrel{\text{def}}{=} \emptyset$, $\mathbf{x}^{(*,-1)} \stackrel{\text{def}}{=} \mathbb{0}$, $\mathbf{x}^{(*,0)} \stackrel{\text{def}}{=} \mathbb{0}$, and define $\mathbf{x}^{(*,t)} \stackrel{\text{def}}{=} \arg\min_{\mathbf{x} \in C^{(t-1)}} g(\mathbf{x})$ for $t \in [T]$, where $C^{(t-1)}$ is defined in Algorithm 4. For all $t \in \{0, 1, \ldots, T\}$, the following properties are satisfied for Algorithm 4:*

1. *It holds $x_i^{(*,t)} > 0$ if and only if $i \in S^{(t-1)}$. We also have $\nabla_i g(\mathbf{x}^{(*,t)}) = 0$ if $i \in S^{(t-1)}$.*
2. *It is $\mathbf{x}^{(t)} \leq \mathbf{x}^{(*,t)} \leq \mathbf{x}^*$ and $\mathbf{x}^{(*,t-1)} \leq \mathbf{x}^{(*,t)}$.*
3. *Our set of known good indices expands $S^{(t-1)} \subsetneq S^{(t)} \stackrel{\text{def}}{=} S^{(t-1)} \cup \left\{i \in [n] \mid \nabla_i g(\mathbf{x}^{(t)}) < 0\right\} \subseteq \mathcal{S}^*$, or $\mathbf{x}^{(t)}$ is an ε-minimizer of g. In particular, $g(\mathbf{x}^{(T)}) - g(\mathbf{x}^*) \leq \varepsilon$.*

Note that by the previous theorem, we have the chain $\mathbb{0} = \mathbf{x}^{(*,0)} \leq \mathbf{x}^{(*,1)} \leq \ldots \leq \mathbf{x}^{(*,T)} \leq \mathbf{x}^*$ and $S^{(-1)} \subsetneq S^{(0)} \subsetneq \ldots \subsetneq S^{(T-1)} = S^{(T)} \subseteq \mathcal{S}^*$. This implies that every iterate of Algorithm 4 only updates coordinates in $\mathcal{S}^*$. Thus, the final computational complexity of this accelerated method, specified below, depends on the sparsity of the solution and related quantities, answering the question posed by (Fountoulakis and Yang, 2022) in the affirmative.

Theorem 8 (Computational complexities) [↓] *The time complexity of Algorithm 4 is*

$$\widetilde{O}\left(|\mathcal{S}^*|\widetilde{\operatorname{vol}}(\mathcal{S}^*)\sqrt{\frac{L}{\alpha}} + |\mathcal{S}^*|\operatorname{vol}(\mathcal{S}^*)\right),$$

and its space complexity is $O(|\mathcal{S}^*|)$.

The question of Fountoulakis and Yang (2022) suggested that one has to possibly trade off lower dependence on the condition number for greater dependence on the sparsity. Surprisingly, the term

$|\mathcal{S}^*|\widetilde{\mathrm{vol}}(\mathcal{S}^*)$ multiplying the condition-number term can be smaller than the corresponding term $\mathrm{vol}(\mathcal{S}^*)$ of ISTA, so in such a case the accelerated method also improves on the dependence on the sparsity, and it enjoys an overall lower running time if $|\mathcal{S}^*| < L/\alpha$, see Appendix B.

4.1. Variants of Algorithm 4

An attractive property of Algorithm 4 is that by performing minor modifications to it, we can exploit the geometry to stop the APGD subroutine earlier, or we can naturally incorporate new lower bounds on the coordinates of $\mathbf{x}^*$ into the algorithm.

4.1.1. EARLY TERMINATION OF APGD IN ASPR

We first present another lemma about the geometry of Problem (3).

Lemma 9 [↓] *Let S be a set of indices such that $\mathbf{x}^{(*,C)} \stackrel{\text{def}}{=} \arg\min_{\mathbf{x}\in C} g(\mathbf{x})$ satisfies $x_j^{(*,C)} > 0$ if and only if $j \in S$, where $C \stackrel{\text{def}}{=} \mathrm{span}(\{\mathbf{e}_j \mid j \in S\}) \cap \mathbb{R}^n_{\geq 0}$. Let $\mathbf{x} \in \mathbb{R}^n_{\geq 0}$ be such that $\mathrm{supp}(\mathbf{x}) \subseteq S$, and $\nabla_j g(\mathbf{x}) \leq 0$ for all $j \in S$. Then, for any coordinate $i \notin S$ such that $\nabla_i g(\mathbf{x}) < 0$, we have $i \in \mathcal{S}^*$.*

By Statement 1 in Theorem 7, we can apply Lemma 9 with $S^{(t)}$ in Algorithm 4, for any $t \in \{0, 1, \ldots, T-1\}$. This motivates the following modification to Algorithm 4: In ASPR, if we compute the full gradient at each iteration (or every few iterations) in the APGD subroutine, then, for an iterate $\mathbf{x}$ in APGD, if we have $\nabla_i g(\mathbf{x}) \leq 0$ for all $i \in S^{(t)}$ and we observe some $j \notin S^{(t)}$ such that $\nabla_j g(\mathbf{x}) < 0$, then we can stop the APGD subroutine, incorporate all such coordinates to $S^{(t+1)}$ and continue with the next iteration of Algorithm 4. This modification does not come without drawbacks, since we need to compute full gradients in order to discover new coordinates early, instead of just gradients restricted to $S^{(t)}$. Interestingly, we can show that if we were to compute one full gradient for each iteration of the APGD subroutine, then the complexity of the conjugate directions method is no worse than the upper bound on the complexity for this variant of Algorithm 4, in the regime in which we prefer to use these algorithms over the ISTA-approach in Fountoulakis et al. (2019). Indeed, the complexity of this variant is $\widetilde{O}(|\mathcal{S}^*|\mathrm{vol}(\mathcal{S}^*)\sqrt{L/\alpha})$, and the complexity of Algorithm 2, which is $O(|\mathcal{S}^*|^3 + |\mathcal{S}^*|\mathrm{vol}(\mathcal{S}^*))$, can be upper bounded by $O(|\mathcal{S}^*|^2\mathrm{vol}(\mathcal{S}^*))$. If the complexity of the variant is better, up to constants and log factors, then we can exchange another $|\mathcal{S}^*|$ term by $\sqrt{L/\alpha}$ to conclude that this complexity is no better than the complexity of the ISTA approach $\widetilde{O}(\mathrm{vol}(\mathcal{S}^*)\frac{L}{\alpha})$, up to constants and log factors. Nonetheless, one can always compute the full gradient only sporadically to discover new good coordinates earlier, and we expect the empirical performance of Algorithm 4 to improve by implementing this modification. In future work, we will extensively test our algorithms with this and other variants to assess their practical performance.

4.1.2. UPDATING CONSTRAINTS

In Algorithm 4, every time we observe $\nabla_i g(\mathbf{x}) \leq 0$ for all $i \in S^{(t)}$, whether for the iterates of the APGD subroutine or for $\mathbf{x}^{(t+1)}$, we have by Statement 1 of Proposition 2 and Statement 2 of

Theorem 7 that $\mathbf{x} \leq \mathbf{x}^{(*,t+1)} \leq \mathbf{x}^*$. Using this new lower bound on the coordinates of $\mathbf{x}^*$, we can update our constraints. If we initialize the constraints to $\bar{C} \leftarrow \mathbb{R}^n_{\geq 0}$, we can update them to $\bar{C} \leftarrow \bar{C} \cap \{\mathbf{y} \in \mathbb{R}^n_{\geq 0} \mid \mathbf{y} \geq \mathbf{x}\}$ every time we find one such point $\mathbf{x}$. This can help avoiding the momentum of APGD taking us far unnecessarily. We note that these constraints are isomorphic to the positive orthant, and only require storing up to $|\mathcal{S}^*|$ numbers.

5. Conclusion

We successfully integrated acceleration of optimization techniques into the field of graph clustering, thereby answering the open question raised in Fountoulakis et al. (2019). Our results provide evidence of the efficacy of this approach, demonstrating that optimization-based algorithms can be effectively employed to address graph-based learning tasks with great efficiency at scale. This work holds the potential to inspire the development of new algorithms that leverage the power of advanced optimization techniques to tackle other graph-based challenges in a scalable manner.

Acknowledgments

This research was partially funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under Germany´s Excellence Strategy – The Berlin Mathematics Research Center MATH$^+$ (EXC-2046/1, project ID 390685689, BMS Stipend).

References

Zeyuan Allen-Zhu and Lorenzo Orecchia. Nearly linear-time packing and covering LP solvers - achieving width-independence and -convergence. *Math. Program.*, 175(1-2):307–353, 2019. doi: 10.1007/s10107-018-1244-x. URL *https://doi.org/10.1007/s10107-018-1244-x*.

Reid Andersen and Kevin J Lang. An algorithm for improving graph partitions. In *Proceedings of the Annual Symposium on Discrete Algorithms*, volume 8, pages 651–660, 2008.

Reid Andersen, Fan Chung, and Kevin Lang. Local graph partitioning using pagerank vectors. In *Proceedings of the Annual Symposium on Foundations of Computer Science*, pages 475–486. IEEE, 2006.

Arkadi Nemirovski, 1981. [Online; accessed 06-February-2023], *https://blogs.princeton.edu/imabandit/wp-content/uploads/sites/122/2019/06/Nemirovski81_Russian.pdf*.

Gary D Bader and Christopher WV Hogue. An automated method for finding molecular complexes in large protein interaction networks. *Bioinformatics*, 4(1):1–27, 2003.

Frédéric Boyer, Anne Morgat, Laurent Labarre, Joël Pothier, and Alain Viari. Syntons, metabolons and interactons: an exact graph-theoretical approach for exploring neighbourhood between genomic and functional data. *Bioinformatics*, 21(23):4209–4215, 2005.

Steve Butler, Fan Chung, et al. Spectral graph theory. *Handbook of linear algebra*, page 47, 2006.

Francisco Criado, David Martínez-Rubio, and Sebastian Pokutta. Fast algorithms for packing proportional fairness and its dual. *arXiv preprint arXiv:2109.03678*, 2021.

Jelena Diakonikolas and Lorenzo Orecchia. The approximate duality gap technique: A unified theory of first-order methods. *SIAM J. Optim.*, 29(1):660–689, 2019. doi: 10.1137/18M1172314. URL *https://doi.org/10.1137/18M1172314*.

Nadav Eiron, Kevin S McCurley, and John A Tomlin. Ranking the web frontier. In *Proceedings of the 13th international conference on World Wide Web*, pages 309–318, 2004.

Santo Fortunato and Darko Hric. Community detection in networks: A user guide. *Physics Reports*, 659:1–44, 2016.

Kimon Fountoulakis and Shenghao Yang. Open problem: Running time complexity of accelerated ℓ_1-regularized pagerank. In *Proceedings of the Conference on Learning Theory*, pages 5630–5632. PMLR, 2022.

Kimon Fountoulakis, Farbod Roosta-Khorasani, Julian Shun, Xiang Cheng, and Michael W Mahoney. Variational perspective on local graph clustering. *Mathematical Programming*, 174(1):553–573, 2019.

David Gleich and Michael Mahoney. Anti-differentiating approximation algorithms: A case study with min-cuts, spectral, and flow. In *Proceedings of the International Conference on Machine Learning*, pages 1018–1025. PMLR, 2014.

Robert M Gower. Conjugate gradients: The short and painful explanation with oblique projections. Technical report, tech. report, University of Edinburgh, Maxwell Institute for Mathematical . . . , 2014.

Chufeng Hu. Local graph clustering using l1-regularized pagerank algorithms. Master's thesis, University of Waterloo, 2020.

Lucas GS Jeub, Prakash Balachandran, Mason A Porter, Peter J Mucha, and Michael W Mahoney. Think locally, act locally: Detection of small, medium-sized, and large communities in large networks. *Physical Review E*, 91(1):012821, 2015.

Kyle Kloster and David F Gleich. Heat kernel based community detection. In *Proceedings of the International Conference on Knowledge Discovery and Data Mining*, pages 1386–1395, 2014.

Jure Leskovec, Kevin J Lang, Anirban Dasgupta, and Michael W Mahoney. Community structure in large networks: Natural cluster sizes and the absence of large well-defined clusters. *Internet Mathematics*, 6(1):29–123, 2009.

Michael W Mahoney, Lorenzo Orecchia, and Nisheeth K Vishnoi. A local spectral method for graphs: With applications to improving graph partitions and exploring data graphs locally. *Journal of Machine Learning Research*, 13(1):2339–2365, 2012.

Yurii Nesterov. Introductory lectures on convex programming volume I: Basic course. *Lecture notes*, 3(4), 1998.

Mark EJ Newman. Properties of highly clustered networks. *Physical Review E*, 68(2):026121, 2003.

Jorge Nocedal and Stephen J. Wright. *Numerical Optimization*. Springer, 1999. ISBN 978-0-387-98793-4. doi: 10.1007/b98874. URL *https://doi.org/10.1007/b98874*.

Lorenzo Orecchia and Zeyuan Allen Zhu. Flow-based algorithms for local graph clustering. In *Proceedings of the Annual Symposium on Discrete Algorithms*, pages 1267–1286. SIAM, 2014.

Lawrence Page, Sergey Brin, Rajeev Motwani, and Terry Winograd. The pagerank citation ranking: Bringing order to the web. Technical Report 1999-66, Stanford InfoLab, November 1999.

Neal Parikh, Stephen Boyd, et al. Proximal algorithms. *Foundations and trends in Optimization*, 1 (3):127–239, 2014.

Youcef Saad and Martin H Schultz. Gmres: A generalized minimal residual algorithm for solving nonsymmetric linear systems. *SIAM Journal on scientific and statistical computing*, 7(3):856–869, 1986.

Yousef Saad. *Iterative methods for sparse linear systems*. SIAM, 2003. ISBN 978-0-89871-534-7. doi: 10.1137/1.9780898718003. URL *https://doi.org/10.1137/1.9780898718003*.

Satu Elisa Schaeffer. Graph clustering. *Computer Science Review*, 1(1):27–64, 2007.

Daniel A Spielman and Shang-Hua Teng. A local clustering algorithm for massive graphs and its application to nearly linear time graph partitioning. *Journal on Computing*, 42(1):1–26, 2013.

Amanda L Traud, Peter J Mucha, and Mason A Porter. Social structure of facebook networks. *Physica A: Statistical Mechanics and its Applications*, 391(16):4165–4180, 2012.

Nate Veldt, David Gleich, and Michael Mahoney. A simple and strongly-local flow-based method for cut improvement. In *Proceedings of the International Conference on Machine Learning*, pages 1938–1947. PMLR, 2016.

Satu Virtanen. Clustering the chilean web. In *Proceedings of the International Conference on Numerical Simulation of Semiconductor Optoelectronic Devices (IEEE Cat. No. 03EX726)*, pages 229–231. IEEE, 2003.

Edwin A. H. Vollebregt. The bound-constrained conjugate gradient method for non-negative matrices. *J. Optim. Theory Appl.*, 162(3):931–953, 2014. doi: 10.1007/s10957-013-0499-x. URL *https://doi.org/10.1007/s10957-013-0499-x*.

Di Wang, Kimon Fountoulakis, Monika Henzinger, Michael W Mahoney, and Satish Rao. Capacity releasing diffusion for speed and locality. In *Proceedings of the International Conference on Machine Learning*, pages 3598–3607. PMLR, 2017.

Ying Xu, Victor Olman, and Dong Xu. Clustering gene expression data using a graph-theoretic approach: an application of minimum spanning trees. *Bioinformatics*, 18(4):536–545, 2002.

Hao Yin, Austin R Benson, Jure Leskovec, and David F Gleich. Local higher-order graph clustering. In *Proceedings of the International Conference on Knowledge Discovery and Data Mining*, pages 555–564. PMLR, 2017.

Appendix A. Missing Proofs

Remark 10 *We recall that, as we pointed out in Section 2.1,* ISTA *on f is equivalent to projected gradient descent in $\mathbb{R}^n_{\geq 0}$ on g. Proposition 2 allows to quite simply recover the result in (Fountoulakis et al., 2019) about* ISTA *initialized at $\mathbb{0}$ having iterates with support in $\mathcal{S}^*$. Moreover, our argument below applies to the optimization of the more general problem where Q is an arbitrary symmetric positive-definite M-matrix.*

Indeed, we satisfy the assumptions of Proposition 2 for initial point $\mathbf{x}^{(0)}$ and set of indices $S \leftarrow \{i \mid \nabla_i g(\mathbf{x}^{(0)}) < 0\}$, which is non-empty unless $\mathbf{x}^{(0)}$ is the solution $\mathbf{x}^*$. Let the corresponding feasible set be $C \stackrel{\text{def}}{=} \operatorname{span}(\{\mathbf{e}_i \mid i \in S\}) \cap \mathbb{R}^n_{\geq 0}$. Now, while the iterates of PGD$(\mathbb{R}_{\geq 0}, \mathbf{x}^{(0)}, g, \cdot)$ *remain in C, that is, for t such that $\mathbf{x}^{(t)} \in C$, they behave exactly like* PGD$(C, \mathbf{x}^{(0)}, g, \cdot)$ *and so, we have the following invariant by the proof of the Proposition 2: $\mathbf{x}^{(t)} \leq \mathbf{x}^{(*,C)}$ and $\nabla_S g(\mathbf{x}^{(t)}) \leq 0$ and $\mathbf{x}^{(t)} \leq \mathbf{x}^{(t+1)}$. If this algorithm leaves C at step t, that is $\mathbf{x}^{(t)} \in C$ and $\mathbf{x}^{(t+1)} \notin C$, we have $\nabla_{\operatorname{supp}(\mathbf{x}^{(t+1)})} g(\mathbf{x}^{(t)}) \leq 0$, since the invariant guarantees $\nabla_S g(\mathbf{x}^{(t)}) \leq 0$ and if $i \in \operatorname{supp}(\mathbf{x}^{(t+1)} \setminus S$, it must be $\nabla_i g(\mathbf{x}^{(t)}) < 0$ by definition of the* PGD *update rule. In particular, we can apply Proposition 2 again with initial point $\mathbf{x}^{(t)}$ and the larger set of indices $\operatorname{supp}(\mathbf{x}^{(t+1)})$, and so on, proving that the invariant $\nabla_{\operatorname{supp}(\mathbf{x}^{(t)})} g(\mathbf{x}^{(t)}) \leq 0$ and $\mathbf{x}^{(t)} \leq \mathbf{x}^{(t+1)}$ holds for all $t \geq 0$. By the global convergence of* PGD$(\mathbb{R}_{\geq 0}, \mathbf{x}^{(0)}, g, \cdot)$*, we have $\mathbf{x}^{(t)} \leq \mathbf{x}^*$ for all $t \geq 0$, so it is always $\operatorname{supp}(\mathbf{x}^{(t)}) \subseteq \mathcal{S}^*$.*

In the rest of this section, we present proofs not found in the main text.

Proof of Theorem 3. We prove the properties in order:

1. For $t = 0$, the statement is trivial. Let $t \in \{0, 1, \ldots, T-1\}$ and assume that $\langle \mathbf{d}^{(j)}, Q\mathbf{d}^{(k)} \rangle = 0$ for all $j, k \in \{0, 1, \ldots, t\}$ such that $j > k$. Then,

$$\langle \mathbf{d}^{(t+1)}, Q\mathbf{d}^{(k)} \rangle = \langle \mathbf{u}^{(t+1)} + \sum_{k=0}^{t} \beta_k^{(t+1)} \mathbf{d}^{(k)}, Q\mathbf{d}^{(k)} \rangle = \langle \mathbf{u}^{(t+1)}, Q\mathbf{d}^{(k)} \rangle + \beta_k^{(t+1)} \langle \mathbf{d}^{(k)}, Q\mathbf{d}^{(k)} \rangle = 0,$$

   where the second and third equalities follow from the induction hypothesis and the definition of $\beta_k^{(t+1)}$, respectively.

2. By induction. For $t = 0$, there is nothing to prove. For some $t \in \{0, 1, \ldots, T-1\}$ suppose that for all $k \in \{0, 1, \ldots, t\}$, it holds that $\langle \nabla g(\mathbf{x}^{(t)}), \mathbf{d}^{(k)} \rangle = 0$. Then, since $\mathbf{x}^{(t+1)} = \mathbf{x}^{(t)} + \eta^{(t)} \mathbf{d}^{(t)}$, we have $\nabla g(\mathbf{x}^{(t+1)}) = \nabla g(\mathbf{x}^{(t)}) + \eta^{(t)} Q\mathbf{d}^{(t)}$. Thus, by Property 1 and the induction hypothesis, we have

$$\langle \nabla g(\mathbf{x}^{(t+1)}), \mathbf{d}^{(k)} \rangle = \langle \nabla g(\mathbf{x}^{(t)}), \mathbf{d}^{(k)} \rangle + \eta^{(t)} \langle Q\mathbf{d}^{(t)}, \mathbf{d}^{(k)} \rangle = 0$$

   for all $k < t$. By the definition of $\eta^{(t)}$, we also have $\langle \nabla g(\mathbf{x}^{(t+1)}), \mathbf{d}^{(t)} \rangle = 0$. Thus,

$$\langle \nabla g(\mathbf{x}^{(t+1)}), \mathbf{d}^{(k)} \rangle = 0 \qquad \text{for all } t \in \{0, 1, \ldots, T-1\} \text{ and } k \in \{0, 1, \ldots, t\}. \tag{6}$$

   Thus, since for $t \in \{0, 1, \ldots, T\}$, $\operatorname{span}(\{\mathbf{d}^{(0)}, \mathbf{d}^{(1)}, \ldots, \mathbf{d}^{(t-1)}\}) = \operatorname{span}(\{\mathbf{e}_i \mid \mathbf{e}_i \in S^{(t-1)}\})$, it holds that $\nabla_i g(\mathbf{x}^{(t)}) = 0$ for all $i \in S^{(t-1)}$ by (6).

3. By induction. For $t = 0$, it holds that $\mathbf{x}_i^{(0)} > 0$ for all $i \in S^{(-1)} = \emptyset$ and $\mathbb{0} = \mathbf{x}^{(0)} = \mathbf{x}^{(*,0)}$. Suppose that the statement holds for some $t \in \{0, 1, \ldots, T-1\}$, that is, $\mathbf{x}_i^{(t)} > 0$ for all $i \in S^{(t-1)}$ and $\mathbb{0} = \mathbf{x}^{(0)} = \mathbf{x}^{(*,0)} \leq \mathbf{x}^{(1)} = \mathbf{x}^{(*,1)} \leq \ldots \leq \mathbf{x}^{(t)} = \mathbf{x}^{(*,t)}$. By Proposition 2 applied to g, $S^{(t)}$, and $\mathbf{x}^{(t)} = \mathbf{x}^{(*,t)}$, we have that $x_i^{(*,t+1)} > 0$ for all $i \in S^{(t)}$ and $\nabla_i g(\mathbf{x}^{(*,t+1)}) = 0$ for all $i \in S^{(t)}$, that is, $\mathbf{x}^{(*,t+1)} = \arg\min_{\mathbf{x} \in \operatorname{span}(\{\mathbf{e}_i \mid i \in S^{(t)}\})} g(\mathbf{x})$. By Property 2, $\nabla_i g(\mathbf{x}^{(t+1)}) = 0$ for all $i \in S^{(t)}$, that is, $\mathbf{x}^{(t+1)} = \arg\min_{\mathbf{x} \in \operatorname{span}(\{\mathbf{e}_i \mid i \in S^{(t)}\})} g(\mathbf{x})$. By strong convexity of g restricted to $\operatorname{span}(\{\mathbf{e}_i \mid i \in S^{(t)}\})$, $\mathbf{x}^{(*,t+1)} = \mathbf{x}^{(t+1)}$.

4. By Property 3, $\mathbb{0} \leq \mathbf{x}^{(T)}$, that is, $\mathbf{x}^{(T)}$ is a feasible solution to the optimization problem (3). By Property 2, $\nabla_i g(\mathbf{x}^{(T)}) = 0$ for all $i \in S^{(T-1)}$. Since $T \in \mathbb{N}$ is the first iteration for which $N^{(T)} = \left\{i \in [n] \mid \nabla_i g(\mathbf{x}^{(T)}) < 0\right\} = \emptyset$, $\mathbf{x}^{(T)}$ satisfies the optimality conditions of (3) and $\mathbf{x}^{(T)} = \mathbf{x}^*$.

■

Proof of Theorem 4. We run Algorithm 2 for $|\mathcal{S}^*|$ iterations. We summarize the costs of operations performed during one iteration $t \in \{0, 1, \ldots, T\}$. The cost of computing $N^{(t+1)}$ is $O(\operatorname{vol}(\mathcal{S}^*))$. Note we do not need to store the gradient, at most we would store $\nabla_{S^{(t+1)}} g(\mathbf{x}^{(t+1)})$, and this is not necessary. Note that the vectors $\mathbf{x}^{(t+1)}$ and $\mathbf{d}^{(t)}$ and $\bar{\mathbf{d}}^{(t)}$ are sparse, their support is in $\mathcal{S}^*$. Thus, computing $\bar{\mathbf{d}}^{(t)}$ takes $O(\widetilde{\operatorname{vol}}(\mathcal{S}^*))$ and computing $\eta^{(t)}$ and $\mathbf{x}^{(t+1)}$ takes $O(|\mathcal{S}^*|)$. Finally, we discuss the complexity of computing $\beta_k^{(t)}$ for $k < t$. In order to compute these values efficiently throughout the algorithm's execution, we stored our normalized Q-orthogonal partial basis consisting of the vectors $\bar{\mathbf{d}}^{(k)}$, for all $k \in \{0, 1, \ldots, T\}$. Since $\operatorname{supp}(\mathbf{u}^{(t)}) = 1$, the cost of computing one $\beta_k^{(t)}$ is only $O(|\mathcal{S}^*|)$ and thus computing all them for $k < t$ and computing $\mathbf{d}^{(t)}$ takes $O(|\mathcal{S}^*|^2)$ operations. In summary, the time complexity of Algorithm 2 is $O(|\mathcal{S}^*|^3 + |\mathcal{S}^*|\operatorname{vol}(\mathcal{S}^*))$. The space complexity of Algorithm 2 is dominated by the cost of storing $\mathbf{d}^{(k)}$ for $k \in \{0, 1, \ldots, t-1\}$, which is $O(|\mathcal{S}^*|^2)$. ■

Proof of Proposition 5. Accelerated gradient descent is a classical algorithm, see Diakonikolas and Orecchia (2019, Theorem 4.10) for instance for an analysis of a strongly-convex version with projections. We have not found the analysis of the form of our Algorithm 3 in the literature, namely the strongly-convex non-lazy version supporting projections. Hence for completeness, we present its analysis here. We do not claim any novelty on this algorithm or analysis, it is a combination of known accelerated techniques.

If we denote the indicator of a closed convex set C by $I_C(\mathbf{x})$ whose value is 0 if $\mathbf{x} \in C$ and $+\infty$ otherwise, we can define $\psi(\mathbf{x}) \stackrel{\text{def}}{=} I_C(\mathbf{x}) + \frac{\sigma}{2}\|\mathbf{x} - \mathbf{x}^{(0)}\|_2^2$ to be a regularizer that encodes the constraints and is σ-strongly convex in the feasible set. Algorithm 3 uses $\sigma = L - \mu$ and $C = \mathbb{R}^n_{\geq 0}$, but any $\sigma > 0$ works and the running time does not depend on it. The method uses the following

updates, for initial points $\mathbf{x}^{(0)} = \mathbf{y}^{(0)} = \mathbf{z}^{(0)} \in C$:

$$\begin{aligned}
\mathbf{x}^{(k+1)} &\leftarrow \frac{a_{k+1}}{A_{k+1}}\mathbf{z}^{(k)} + \frac{A_k}{A_{k+1}}\mathbf{y}^{(k)} \\
\mathbf{z}^{(k+1)} &\leftarrow \arg\min_{u\in C}\left\{\frac{\sigma + A_k\mu}{2}\|\mathbf{z}^{(k)} - u\|_2^2 + \frac{a_{k+1}\mu}{2}\|u - (\mathbf{x}^{(k+1)} - \frac{1}{\mu}\nabla f(\mathbf{x}^{(k+1)}))\|_2^2\right\} \\
&= \Pi_C\left(\frac{\sigma/\mu + A_k}{\sigma/\mu + A_{k+1}}\mathbf{z}^{(k)} + \frac{a_{k+1}}{\sigma/\mu + A_{k+1}}(\mathbf{x}^{(k+1)} - \frac{1}{\mu}\nabla f(\mathbf{x}^{(k+1)}))\right) \\
\mathbf{y}^{(k+1)} &\leftarrow \frac{a_{k+1}}{A_{k+1}}\mathbf{z}^{(k+1)} + \frac{A_k}{A_{k+1}}\mathbf{y}^{(k)}
\end{aligned} \tag{7}$$

where $A_t \stackrel{\text{def}}{=} A_{t-1} + a_t = \sum_{i=0}^{t} a_i$, and $a_i > 0$ are positive parameters to be determined later, except for $a_0 = 0$. For $V_0 = 0$, recursively define

$$V_k \stackrel{\text{def}}{=} \min_{\mathbf{u}\in C}\left\{V_{k-1} + \frac{\sigma + A_{k-1}\mu}{2}\|\mathbf{z}^{(k-1)} - \mathbf{u}\|_2^2 + a_k\left(\langle\nabla f(\mathbf{x}^{(k)}), \mathbf{u} - \mathbf{x}^{(k)}\rangle + \frac{\mu}{2}\|\mathbf{u} - \mathbf{x}^{(k)}\|_2^2\right)\right\}$$

so that we have the following lower bound estimation of the optimum

$$\begin{aligned}
A_t f(\mathbf{x}^*) &\stackrel{①}{\geq} \sum_{k=1}^{t} a_k f(\mathbf{x}^{(k)}) + \min_{\mathbf{u}\in C}\left\{\sum_{k=1}^{t} a_k\left(\langle\nabla f(\mathbf{x}^{(k)}), \mathbf{u} - \mathbf{x}^{(k)}\rangle + \frac{\mu}{2}\|\mathbf{u} - \mathbf{x}^{(k)}\|_2^2\right) + \frac{\sigma}{2}\|\mathbf{u} - \mathbf{x}^{(0)}\|_2^2\right\} \\
&\quad - \frac{\sigma}{2}\|\mathbf{x}^* - \mathbf{x}^{(0)}\|_2^2 \\
&\stackrel{②}{\geq} \sum_{k=1}^{t} a_k f(\mathbf{x}^{(k)}) + V_t + \min_{\mathbf{u}\in C}\left\{\frac{\sigma + \mu A_t}{2}\|\mathbf{u} - \mathbf{z}^{(t)}\|_2^2\right\} - \frac{\sigma}{2}\|\mathbf{x}^* - \mathbf{x}^{(0)}\|_2^2 \\
&\stackrel{③}{=} \sum_{k=1}^{t} a_k f(\mathbf{x}^{(k)}) + V_t - \frac{\sigma}{2}\|\mathbf{x}^* - \mathbf{x}^{(0)}\|_2^2 \stackrel{\text{def}}{=} A_t L_t,
\end{aligned}$$

where ① uses convexity, adds and subtracts $\psi(\mathbf{x})$ and takes a minimum. Then, ② uses $\mathbf{x}^{(0)} = \mathbf{z}^{(0)}$ and then it uses Lemma 11 sequentially t times (for $k = 1, \ldots, t$ and taking into account that $V_0 = A_0 = 0$) and ③ holds because the value of the minimum is 0, since $\mathbf{z}^{(t)} \in C$. We define the lower bound L_t on $f(x^*)$ as the one satisfying the equality above.

Now let $U_t \stackrel{\text{def}}{=} f(\mathbf{y}^{(t)})$ be an upper bound on the function value of our current point. If we show $A_k(U_k - L_k) \leq A_{k-1}(U_{k-1} - L_{k-1})$ for all $k \in \{1, \ldots, t\}$, then we can bound the duality gap as $f(\mathbf{y}^{(t)}) - f(\mathbf{x}^*) \leq U_t - L_t \leq \frac{A_1}{A_t}(U_1 - L_1)$. We proceed to show this inequality for $k \geq 1$ and we

also bound $A_1(U_1 - L_1)$, which allows us to conclude. We have, for all $k \geq 1$:

$$
\begin{aligned}
&A_k(U_k - L_k) - A_{k-1}(U_{k-1} - L_{k-1}) - \mathbf{1}_{\{k=1\}}\frac{\sigma}{2}\|\mathbf{x}^* - \mathbf{x}^{(0)}\|_2^2 \\
&\overset{①}{=} A_{k-1}(f(\mathbf{x}^{(k)}) - f(\mathbf{y}^{(k-1)})) + a_k f(\mathbf{x}^{(k)}) + A_k(f(\mathbf{y}^{(k)}) - f(\mathbf{x}^{(k)})) \\
&\quad - \sum_{i=1}^{k} a_i f(\mathbf{x}^{(i)}) - V_k + \sum_{i=1}^{k-1} a_i f(\mathbf{x}^{(i)}) + V_{k-1} \\
&\overset{②}{\leq} A_{k-1}\langle \nabla f(\mathbf{x}^{(k)}), \mathbf{x}^{(k)} - \mathbf{y}^{(k-1)}\rangle - V_k + V_{k-1} + A_k(f(\mathbf{y}^{(k)}) - f(\mathbf{x}^{(k)})) \\
&\overset{③}{=} \langle \nabla f(\mathbf{x}^{(k)}), A_{k-1}(\mathbf{x}^{(k)} - \mathbf{y}^{(k-1)}) - a_k(\mathbf{z}^{(k)} - \mathbf{x}^{(k)})\rangle - \frac{\sigma + A_{k-1}\mu}{2}\|\mathbf{z}^{(k)} - \mathbf{z}^{(k-1)}\|_2^2 \\
&\quad - \mathbf{1}_{\{k=1\}}\frac{a_1\mu}{2}\|\mathbf{z}^{(k)} - \mathbf{x}^{(k)}\|_2^2 + A_k(f(\mathbf{y}^{(k)}) - f(\mathbf{x}^{(k)})) \\
&\overset{④}{=} a_k\langle \nabla f(\mathbf{x}^{(k)}), \mathbf{z}^{(k-1)} - \mathbf{z}^{(k)}\rangle - \frac{\sigma + (A_{k-1} + \mathbf{1}_{\{k=1\}}a_k)\mu}{2}\|\mathbf{z}^{(k)} - \mathbf{z}^{(k-1)}\|_2^2 + A_k(f(\mathbf{y}^{(k)}) - f(\mathbf{x}^{(k)})) \\
&\overset{⑤}{\leq} \left(\frac{La_k^2}{2A_k} - \frac{\sigma + (A_{k-1} + \mathbf{1}_{\{k=1\}}a_k)\mu}{2}\right)\|\mathbf{z}^{(k)} - \mathbf{z}^{(k-1)}\|_2^2 \overset{⑥}{\leq} 0.
\end{aligned}
\tag{8}
$$

In ① we write out the definition of the left hand side, but we add and subtract $A_k f(\mathbf{x}^{(k)})$. The point $\mathbf{y}^{(k)}$ is defined to reduce the upper bound on the objective with respect to what we would obtain at $\mathbf{x}^{(k)}$, where we compute the gradient, which is why we compare to $A_k f(\mathbf{x}^{(k)})$ to $A_k U_k$. We also canceled the terms $\pm\frac{\sigma}{2}\|\mathbf{x}^* - \mathbf{x}^{(0)}\|_2^2$ coming from the lower bounds, except for $k = 1$, since $A_0 = 0$. In ②, we use convexity on the first summand and cancel several terms. In ③ we write the definition of V_k with its argmin $\mathbf{z}^{(k)}$, group terms, and drop $-a_k\frac{\mu}{2}\|\mathbf{x}^{(k)} - \mathbf{z}^{(k)}\|_2^2$, except for $k = 1$, for which we have $\mathbf{x}^{(1)} = \mathbf{z}^{(0)}$ so we will be able to group it with the other terms in ④. In ④, we use equality $(A_{k-1} + a_k)\mathbf{x}^{(k)} = A_k\mathbf{x}^{(k)} = A_{k-1}\mathbf{y}^{(k-1)} + a_k\mathbf{z}^{(k-1)}$, which holds by definition of $\mathbf{x}^{(k)}$ and A_k. The point $\mathbf{y}^{(k)}$ was defined to improve the upper bound with respect to using $f(\mathbf{x}^{(k)})$ as upper bound and it was chosen so that the vector $\mathbf{y}^{(k)} - \mathbf{x}^{(k)}$ is exactly $\frac{a_k}{A_k}(\mathbf{z}^{(k)} - \mathbf{z}^{(k-1)})$ so we can use smoothness to show ⑤:

$$
\begin{aligned}
A_k(f(\mathbf{y}^{(k)}) - f(\mathbf{x}^{(k)})) &\leq A_k\left(\langle \nabla f(\mathbf{x}^{(k)}), \mathbf{y}^{(k)} - \mathbf{x}^{(k)}\rangle + \frac{L}{2}\|\mathbf{y}^{(k)} - \mathbf{x}^{(k)}\|_2^2\right) \\
&= a_k\langle \nabla f(\mathbf{x}^{(k)}), \mathbf{z}^{(k)} - \mathbf{z}^{(k-1)}\rangle + \frac{La_k^2}{2A_k}\|\mathbf{z}^{(k)} - \mathbf{z}^{(k-1)}\|_2^2.
\end{aligned}
$$

Finally, we choose the step-sizes a_k so that ⑥ holds. For $k \geq 2$, it is enough if we drop σ and we have $a_k^2 \leq \frac{A_k A_{k-1}}{\kappa}$, where $\kappa \overset{\text{def}}{=} L/\mu$. Using $A_k = A_{k-1} + a_k$ we just need to solve the equation $a_k^2 - a_k\frac{A_{k-1}}{\kappa} - \frac{A_{k-1}^2}{\kappa} = 0$ in order to obtain the maximum a_k satisfying the inequality. This yields $a_k = A_{k-1}(\frac{1+\sqrt{1+4\kappa}}{2\kappa})$, and so $A_k = A_{k-1} + a_k = A_{k-1}(\frac{2\kappa+1+\sqrt{1+4\kappa}}{2\kappa})$. For $k = 1$, we need $a_1 L \leq \sigma + a_1\mu$, since we choose $A_0 = 0$. We can choose, as in Algorithm 3, $a_1 = 1 = A_1$ and $\sigma = L - \mu$, but in general we can also choose an arbitrary $\sigma \geq 0$ and $a_1 = A_1 = \frac{\sigma}{L-\mu}$.

We conclude with:

$$f(\mathbf{y}^{(t)}) - f(\mathbf{x}^*) \le U_t - L_t \overset{①}{\le} \frac{A_1(U_1 - L_1)}{A_t} \overset{②}{\le} \frac{\sigma}{2A_t}\|\mathbf{x}^* - \mathbf{x}^{(0)}\|_2^2$$
$$\overset{③}{=} \frac{(L-\mu)\|\mathbf{x}^* - \mathbf{x}^{(0)}\|_2^2}{2}\left(\frac{2\kappa}{2\kappa + 1 - \sqrt{1+4\kappa}}\right)^{-(t-1)} \le \frac{(L-\mu)\|\mathbf{x}^* - \mathbf{x}^{(0)}\|_2^2}{2}\left(1 - \frac{1}{2\sqrt{\kappa}}\right)^{t-1},$$

where ① and ② are due to (8), ③ holds by the choice of A_t, and the last inequality is a bound to make the expression simple. The second part of Proposition 5 is a straightforward corollary of the first part. For any $T \ge 1 + \lceil 2\sqrt{\kappa}\log(\frac{(L-\alpha)\|\mathbf{x}^{(0)} - \mathbf{x}^*\|_2^2}{2\varepsilon})\rceil$ we have

$$\begin{aligned} f(\mathbf{y}^{(T)}) - f(\mathbf{x}^*) &\le \left(1 - \frac{1}{2\sqrt{\kappa}}\right)^{T-1} \frac{(L-\alpha)\|\mathbf{x}^{(0)} - \mathbf{x}^*\|_2^2}{2} && \triangleright \text{ by the first part of Proposition 5} \\ &\le \exp\left(-\frac{1}{2\sqrt{\kappa}}(T-1)\right)\frac{(L-\alpha)\|\mathbf{x}^{(0)} - \mathbf{x}^*\|_2^2}{2} && \triangleright \text{ since } (1+x) \le e^x \text{ for all } x \in \mathbb{R} \\ &\le \varepsilon \end{aligned}$$

In particular, by α-strong convexity of f, we have $\|\mathbf{x}^{(0)} - \mathbf{x}^*\|_2^2 \le \frac{\|\nabla f(\mathbf{x}^{(0)})\|_2^2}{\alpha^2}$ so we obtain an ε-minimizer after $1 + \lceil 2\sqrt{\kappa}\log(\frac{(L-\alpha)\|\nabla f(\mathbf{x}^{(0)})\|_2^2}{2\varepsilon\alpha^2})\rceil$ iterations. ■

Lemma 11 (Mirror lemma) *For all $u \in C$ and $k \ge 1$ we have*

$$V_{k-1} + \frac{\sigma + \mu A_{k-1}}{2}\|\mathbf{u} - \mathbf{z}^{(k-1)}\|_2^2 + a_k\left(\langle \nabla f(\mathbf{x}^{(k)}), u - \mathbf{x}^{(k)}\rangle + \frac{\mu}{2}\|\mathbf{u} - \mathbf{x}^{(k)}\|_2^2\right) \ge V_k + \frac{\sigma + \mu A_k}{2}\|\mathbf{u} - \mathbf{z}^{(k)}\|_2^2. \tag{9}$$

Proof The minimum value of the left hand side of (9) over $u \in C$ is V_k by definition. The left hand side is a quadratic with leading term $\frac{\sigma+\mu A_{k-1}}{2} + \frac{\mu a_k}{2} = \frac{\sigma+\mu A_k}{2}$ and whose minimizer is $\hat{\mathbf{z}}^{(k)} \overset{\text{def}}{=} \frac{\sigma + A_{k-1}}{\sigma + A_k}\mathbf{z}^{(k-1)} + \frac{a_k}{\sigma + A_k}(\mathbf{x}^{(k)} - \frac{1}{\mu}\nabla f(\mathbf{x}^{(k)}))$ since this point satisfies the first order optimality condition:

$$(\sigma + \mu A_{k-1})(\hat{\mathbf{z}}^{(k)} - \mathbf{z}^{(k-1)}) + a_k \nabla f(\mathbf{x}^{(k)}) + a_k \mu(\hat{\mathbf{z}}^{(k)} - \mathbf{x}^{(k)}) = 0.$$

Thus, we have that for all $u \in C$ the left hand side of (9) is equal to $V_k + \frac{\sigma+\mu A_k}{2}\|u - \hat{\mathbf{z}}^{(k)}\|_2^2$ which is $\ge V_k + \frac{\sigma+\mu A_k}{2}\|u - \mathbf{z}^{(k)}\|_2^2$ by the definition of $\mathbf{z}^{(k)}$ as the projection of $\hat{\mathbf{z}}^{(k)}$ onto C. ■

Proof of Lemma 6. Let $i, j \in [n]$ such that $i \ne j$. Geometrically, since the gradient of g is an affine function, the set of points $\mathbf{y}$ for which $\nabla_j g(\mathbf{y}) \ge c$ for some value c, forms a halfspace. Fixing x_i, any point otherwise coordinatewise smaller than $\mathbf{x}$ does not increase in gradient, since the off-diagonal entries of Q are non-positive. That is, the corresponding $(n-1)$-dimensional halfspace is defined by a packing constraint (Allen-Zhu and Orecchia, 2019; Criado et al., 2021). Formally, we have

$$\nabla_j g(\mathbf{y}) - \nabla_j g(\mathbf{x}) = (Q\mathbf{y})_j - (Q\mathbf{x})_j = -\varepsilon(Q\mathbf{e}_i)_j = -\varepsilon Q_{j,i} \ge 0,$$

where the last inequality uses $Q_{i,j} = -\frac{(1-\alpha)A_{i,j}}{2d_i d_j} \leq 0$. The second statement is analogous. ■

Proof of Theorem 7. Because we have $S^{(-1)} = \emptyset$, $\mathbf{x}^{(*,-1)} = \mathbf{x}^{(*,0)} = \mathbf{x}^{(0)} = \mathbb{0}$, and by definition it is $\mathbf{x}^* \in \mathbb{R}^n_{\geq 0}$, we have that the first two properties hold trivially for $t = 0$. Property 3 also holds for $t = 0$. Indeed, if the set of known good indices does not expand, we have $\nabla g(\mathbf{x}^{(0)}) \geq 0$ and so we have $\mathbf{x}^{(0)} = \mathbb{0} = \mathbf{x}^*$, and thus $\mathbf{x}^{(0)}$ is an ε-minimizer of g for any $\varepsilon > 0$.

We now prove the three properties inductively. Fix $t \in \{0, 1, \ldots, T-1\}$ and assume Properties 1 to 3 hold for this choice of $k \in \{0, \ldots, t\}$. We will prove they hold for $t+1$.

The value of the accuracy $\hat{\varepsilon}_t$ in Algorithm 4 was chosen to compute $\bar{\mathbf{x}}^{(t+1)}$ close enough to $\mathbf{x}^{(*,t+1)}$. In particular, we have

$$\|\bar{\mathbf{x}}^{(t+1)} - \mathbf{x}^{(*,t+1)}\|^2 \overset{①}{\leq} \frac{2}{\alpha}(g(\mathbf{x}^{(t+1)}) - g(\mathbf{x}^{(*,t+1)})) \overset{②}{\leq} \frac{2\hat{\varepsilon}_t}{\alpha} \overset{③}{=} \delta_t^2, \tag{10}$$

where we used α-strong convexity of g for ①, the convergence guarantee of APGD on $\bar{\mathbf{x}}^{(t+1)}$ for ②, and for ③ we used the definition of $\hat{\varepsilon}_t$. The above allows to show that $\mathbf{x}^{(t+1)} \overset{\text{def}}{=} \max\{\mathbb{0}, \bar{\mathbf{x}}^{(t+1)} - \delta_t \mathbb{1}_n\}$, where the max is taken coordinatewise, satisfies

$$\mathbf{x}^{(t+1)} \leq \mathbf{x}^{(*,t+1)}. \tag{11}$$

Suppose this property does not hold and that for some i we have $x_i^{(t+1)} > x_i^{(*,t+1)} \geq 0$. Then, we would have that $x_i^{(t+1)} = \bar{x}_i^{(t+1)} - \delta_t$ and

$$\|\bar{\mathbf{x}}^{(t+1)} - \mathbf{x}^{(*,t+1)}\| \geq |\bar{x}_i^{(t+1)} - x_i^{(*,t+1)}| \geq \bar{x}_i^{(t+1)} - x_i^{(*,t+1)} = x_i^{(t+1)} + \delta_t - x_i^{(*,t+1)} > \delta_t,$$

which is a contradiction. Note that we have

$$\nabla_j g(\mathbf{x}^{(*,t+1)}) \overset{①}{\leq} \nabla_j g(\mathbf{x}^{(t+1)}) \overset{②}{<} 0 \qquad \text{for all } j \in S^{(t+1)} \setminus S^{(t)}, \tag{12}$$

since ② holds by definition of $S^{(t+1)}$ and ① is due to Lemma 6 and the fact that we can write $\mathbf{x}^{(t+1)} = \mathbf{x}^{(*,t+1)} - \sum_{i \in S^{(t)}} \omega_i \mathbf{e}_i$ for some $\omega_i \in \mathbb{R}_{\geq 0}$, since we just proved $\mathbf{x}^{(t+1)} \leq \mathbf{x}^{(*,t+1)}$ in (11), and by construction $\operatorname{supp}(\mathbf{x}^{(t+1)}) \subseteq S^{(t)}$ and $\operatorname{supp}(\mathbf{x}^{(*,t+1)}) \subseteq S^{(t)}$.

We now show that

$$\mathbf{x}^{(*,t)} \leq \mathbf{x}^{(*,t+1)}. \tag{13}$$

This fact holds by Item 1 of Proposition 2 with starting point $\mathbf{x}^{(*,t)}$ and $S \leftarrow S^{(t)}$, which makes it $\mathbf{x}^{(*,C)} \leftarrow \mathbf{x}^{(*,t+1)}$. The assumptions of Proposition 2 hold since $\mathbf{x}^{(*,t)} = 0$ if $i \in [n] \setminus S^{(t)} \subseteq [n] \setminus S^{(t-1)}$ by construction and we have $\nabla_i g(\mathbf{x}^{(*,t)}) = 0$ for $i \in S^{(t-1)}$ by induction hypothesis of Property 1 and $\nabla_i g(\mathbf{x}^{(*,t)}) < 0$ for $i \in S^{(t)} \setminus S^{(t-1)}$ by the same argument we provided to show (12). In the same context, we also use Item 2 of Proposition 2, using the fact that $\nabla_i g(\mathbf{x}^{(*,t)}) < 0$ for $i \in S^{(t)} \setminus S^{(t-1)}$ and that by Property 1 for t, we have $x_i^{(*,t)} > 0$ for all $i \in S^{(t-1)}$. Therefore, we conclude $x_i^{(*,t+1)} > 0$ for all $i \in S^{(t)}$, which means we proved Property 1 for $t+1$.

Moreover, now using Item 3 of Proposition 2 in this context, we conclude

$$\mathbf{x}^{(*,t+1)} \leq \mathbf{x}^*. \tag{14}$$

Thus, Property 2 holds for $t+1$ since we proved (11), (13) and (14).

Now we prove Property 3 for $t+1$. We note that the value of δ_t was chosen so that the retracted point $\mathbf{x}^{(t+1)}$ still enjoys a small enough gap:

$$\begin{aligned} g(\mathbf{x}^{(t+1)}) - g(\mathbf{x}^{(*,t+1)}) &\overset{①}{\leq} \frac{L}{2}\|\mathbf{x}^{(t+1)} - \mathbf{x}^{(*,t+1)}\|_2^2 \\ &\overset{②}{\leq} L(\|\bar{\mathbf{x}}^{(t+1)} - \mathbf{x}^{(*,t+1)}\|_2^2 + |S^{(t)}|\delta_t^2) \\ &\overset{③}{\leq} L(1+|S^{(t)}|)\delta_t^2 \\ &\overset{④}{\leq} \frac{\varepsilon\alpha}{L}. \end{aligned} \tag{15}$$

Above, ① uses the optimality of $\mathbf{x}^{(*,t+1)}$ and L-smoothness, while ② holds because by construction of $\mathbf{x}^{(t+1)}$, we have $\sum_{i\in S^{(t)}} |x_i^{(t+1)} - x_i^*|^2 \leq \sum_{i\in S^{(t)}} (|\bar{x}_i^{(t+1)} - x_i^*| + \delta_t)^2 \leq 2\sum_{i\in S^{(t)}} (\bar{x}_i^{(t+1)} - x_i^*)^2 + 2|S^{(t)}|\delta_t^2$. We have ③ by (10) and ④ holds by the definition of δ_t, which was made to satisfy this inequality.

We now show that if $\mathbf{x}^{(t+1)}$ is not an ε-minimizer of g in $\mathbb{R}^n_{\geq 0}$, then one step of PGD makes more progress than what can be made in $C^{(t+1)}$, by (15), and so PGD explores a new coordinate, that is, we have a coordinate i with $\nabla_i g(\mathbf{x}^{(t+1)}) < 0$ and we can extend the set of good coordinates $S^{(t)}$. Indeed, suppose that $g(\mathbf{x}^{(t+1)}) - g(\mathbf{x}^*) > \varepsilon$. Let $\mathbf{y}^{(t+1)} = \mathrm{Proj}_{\mathbb{R}^n_{\geq 0}}\left(\mathbf{x}^{(t+1)} - \nabla g(\mathbf{x}^{(t+1)})\right)$. We use the following property in (Fountoulakis et al., 2019, Equation below (23)) from the guarantees of ISTA on the problem, or equivalently on PGD $\left(C^{(t+1)}, \mathbf{x}^{(t)} t, g, 1\right)$, see Section 2.1. We have

$$g(\mathbf{y}^{(t+1)}) - g(\mathbf{x}^*) \leq \left(1 - \frac{\alpha}{L}\right)(g(\mathbf{x}^{(t+1)}) - g(\mathbf{x}^*)). \tag{16}$$

Consequently, we obtain

$$g(\mathbf{x}^{(t+1)}) - g(\mathbf{x}^{(*,t+1)}) \overset{①}{\leq} \frac{\varepsilon\alpha}{L} \overset{②}{<} \frac{\alpha}{L}(g(\mathbf{x}^{(t+1)}) - g(\mathbf{x}^*)) \overset{③}{\leq} g(\mathbf{x}^{(t+1)}) - g(\mathbf{y}^{(t+1)}),$$

where ① holds by (15), ② holds by our earlier assumption $g(\mathbf{x}^{(t+1)}) - g(\mathbf{x}^*) > \varepsilon$, and ③ is obtained by (16) after adding $g(\mathbf{x}^{(t+1)}) - g(\mathbf{y}^{(t+1)})$ to both sides, and reorganizing. Hence, $g(\mathbf{x}^{(*,t+1)}) > g(\mathbf{y}^{(t+1)})$. Since $\mathbf{x}^{(*,t+1)}$ is the minimizer of g in $C^{(t)}$, it holds that $\mathbf{y}^{(t+1)} \notin C^{(t)}$ and so $\nabla_i g(\mathbf{x}^{(t+1)}) < 0$ for at least one $i \notin S^{(t)}$, and $S^{(t)} \subsetneq S^{(t+1)}$.

It remains to prove that $S^{(t+1)} \subseteq \mathcal{S}^*$. For $t+1 = T$ it is $S^{(T-1)} = S^{(T)}$ and the property holds by induction hypothesis. For the case $t+1 \neq T$, suppose the property does not hold and so there exists $j \notin S^{(t)}$ such that $j \notin \mathcal{S}^*$ and $\nabla_j g(\mathbf{x}^{(t+1)}) < 0$. In that case, we have by (12) that $\nabla_j g(\mathbf{x}^{(*,t+1)}) < 0$. On the other hand, it is $\nabla_i g(\mathbf{x}^{(*,t+1)}) = 0$ and $x_i^{(*,t+1)} > 0$ for $i \in S^{(t)}$ by Property 1 and so we can apply Proposition 2 with $S \leftarrow S^{(t)} \cup \{j\}$ and initial point $\mathbf{x}^{(*,t+1)}$ to conclude a contradiction, since by Item 2 it is $x_j^{(*,C)} > 0$ but by Item 3 we have $x_j^{(*,C)} \leq x_j^* = 0$.

Finally, by Property 3 for $t = T$, since $S^{(T-1)}$ does not expand, that is, $S^{(T-1)} = S^{(T)}$, it must be $g(\mathbf{x}^{(T)}) - g(\mathbf{x}^*) \leq \varepsilon$.

■

Proof of Theorem 8. For each iteration, the time complexity of Algorithm 4 is the cost of the APGD subroutine plus the full gradient computation in Line 10. By Theorem 7, APGD is called at most $T = |\mathcal{S}^*|$ times and it runs for $O\left(\sqrt{\frac{L}{\alpha}}\log\left(\frac{(L-\alpha)\|\nabla_{S^{(t)}}g(\mathbf{x}^{(t)})\|_2^2}{\hat{\varepsilon}_t\alpha^2}\right)\right)$ iterations at each stage t. One iteration of APGD involves the computation of the gradient restricted to the current subspace of good coordinates, and involves the update of the iterates, costing $O(\widetilde{\mathrm{vol}}(\mathcal{S}^*))$. The computation of the full gradient takes $O(\mathrm{vol}(\mathcal{S}^*))$ operations. Thus, the total running time of Algorithm 4 is

$$
\begin{aligned}
&O\left(|\mathcal{S}^*|\widetilde{\mathrm{vol}}(\mathcal{S}^*)\sqrt{\frac{L}{\alpha}}\log\left(\frac{(L-\alpha)\max_{t\in\{0,1,\dots,T-1\}}\|\nabla_{S^{(t)}}g(\mathbf{x}^{(t)})\|_2^2}{\hat{\varepsilon}_t\alpha^2}\right)+|\mathcal{S}^*|\mathrm{vol}(\mathcal{S}^*)\right)\\
&\overset{①}{=} O\left(|\mathcal{S}^*|\widetilde{\mathrm{vol}}(\mathcal{S}^*)\sqrt{\frac{L}{\alpha}}\log\left(\frac{L^2(L-\alpha)\|\mathbb{0}-\mathbf{x}^*\|^2}{\hat{\varepsilon}_t\alpha^2}\right)+|\mathcal{S}^*|\mathrm{vol}(\mathcal{S}^*)\right)\\
&=\widetilde{O}\left(|\mathcal{S}^*|\widetilde{\mathrm{vol}}(\mathcal{S}^*)\sqrt{\frac{L}{\alpha}}+|\mathcal{S}^*|\mathrm{vol}(\mathcal{S}^*)\right),
\end{aligned}
$$

where ① holds since by L-smoothness of g restricted to $\mathrm{span}(\{\mathbf{e}_i \mid i\in S^{(t)}\})$ and by $\mathbb{0}\le\mathbf{x}^{(t)}\le\mathbf{x}^{(*,t)}\le\mathbf{x}^*$ for all $t\in[T]$, we have $\|\nabla_{S^{(t)}}g(\mathbf{x}^{(t)})\|_2^2\le L\|\mathbf{x}^{(t)}-\mathbf{x}^{(*,t)}\|_2^2\le L\|\mathbb{0}-\mathbf{x}^*\|_2^2$. To further interpret the bound in the ℓ_1-regularized PageRank problem, we can further bound

$$
\begin{aligned}
\|\mathbb{0}-\mathbf{x}^*\|_2^2 &\le \frac{1}{\alpha^2}\|\nabla_{\mathcal{S}^*}g(\mathbb{0})-\nabla_{\mathcal{S}^*}g(\mathbf{x}^*)\|_2^2 && \triangleright\ \text{by } \alpha\text{-str. convexity of } g \text{ in } \mathrm{span}(\{\mathbf{e}_i\mid i\in\mathcal{S}^*\})\\
&\le \frac{1}{\alpha^2}\|\nabla_{\mathcal{S}^*}g(\mathbb{0})\|_2^2 && \triangleright\ \text{by optimality of } \mathbf{x}^*\\
&\le \frac{1}{\alpha^2}\|(-\alpha D^{-1/2}\mathbf{s}+\alpha\rho D^{1/2}\mathbb{1}_n)_{\mathcal{S}^*}\|_2^2 && \triangleright\ \text{by the gradient definition}\\
&\le \frac{1}{\alpha^2}\|(-D^{-1/2}\mathbb{1}_n+D^{1/2}\mathbb{1}_n)_{\mathcal{S}^*}\|_2^2 && \triangleright\ \alpha, s_i, \rho\le 1\\
&\le \frac{1}{\alpha^2}(1+\sqrt{\mathrm{vol}(\mathcal{S}^*)})^2|\mathcal{S}^*| && \triangleright\ \text{maximum } d_i \text{ for } i\in\mathcal{S}^* \text{ is } \le|\mathrm{vol}(\mathcal{S}^*)|\\
&= O\left(\frac{1}{\alpha^2}|\mathcal{S}^*|\mathrm{vol}(\mathcal{S}^*)\right).
\end{aligned}
$$

Then, by the definition of $\hat{\varepsilon}_t$, the time complexity of Algorithm 4 of the ℓ_1-regularized PageRank problem is

$$
O\left(|\mathcal{S}^*|\widetilde{\mathrm{vol}}(\mathcal{S}^*)\sqrt{\frac{L}{\alpha}}\log\left(\frac{2L^4(1+|S^{(t)}|)(L-\alpha)|\mathcal{S}^*|\mathrm{vol}(\mathcal{S}^*)}{\alpha^6\varepsilon}\right)+|\mathcal{S}^*|\mathrm{vol}(\mathcal{S}^*)\right).
$$

The space complexity of Algorithm 4 is dominated by the cost of storing the gradient $\nabla_{S^{(t)}}g(\mathbf{x}^{(t)})$, which is $O(|\mathcal{S}^*|)$, since $S^{(t)}\subseteq\mathcal{S}^*$. Note that we require to compute the full gradient when updating $S^{(t+1)}$, but we only store the new indices. ■

Proof of Lemma 9. Fix $j \notin S$ such that $\nabla_j g(\mathbf{x}) < 0$. By the assumption $x_i = 0$ if $i \notin S$ and $\nabla_i g(\mathbf{x}) \leq 0$ if $i \in S$, and therefore we can use Proposition 2 with S and $\mathbf{x}$ to conclude $\mathbb{0} \leq \mathbf{x} \leq \mathbf{x}^{(*,C)}$ and $\nabla_j g(\mathbf{x}^{(*,C)}) = 0$. We can thus write $\mathbf{x} = \mathbf{x}^{(*,C)} - \sum_{i \in S} \omega_i \mathbf{e}_i$, for $\omega_i \in \mathbb{R}_{\geq 0}$ for all $i \in S$. By Lemma 6, it holds that $\nabla_j g(\mathbf{x}^{(*,C)}) \leq \nabla_j g(\mathbf{x}) < 0$. We now use Properties 2 and 3 in Proposition 2 with set of indices $\bar{S} \stackrel{\text{def}}{=} S \cup \{j\}$ and starting point $\mathbf{x}^{(*,C)}$. By Property 2 we have for $\mathbf{x}^{(*,\bar{C})} = \arg\min_{\mathbf{x} \in \bar{C}} g(\mathbf{x})$ that $x_j^{(*,\bar{C})} > 0$, where $\bar{C} \stackrel{\text{def}}{=} \operatorname{span}(\{\mathbf{e}_i \mid i \in \bar{S}\})$. By Property 1, we have $x_i^{(*,\bar{C})} \geq x_i^{(*,C)} > 0$ for $i \in S$. Thus, $x_i^{(*,\bar{C})} > 0$ for all $i \in \bar{S}$ and by Property 3, it holds $\bar{S} \subseteq \mathcal{S}^*$ and in particular $j \in \mathcal{S}^*$. ■

Appendix B. Algorithmic Comparisons

CDPR has worse space complexity, $O(|\mathcal{S}^*|^2)$, than ASPR and ISTA, both $O(|\mathcal{S}^*|)$. However, since CDPR finds the exact solution, CDPR outperforms the other methods in running time for small enough ε. Note that the time complexities of ISTA and ASPR depend on $\frac{1}{\varepsilon}$ only logarithmicly. We perform the remaining comparison for $\log(1/\varepsilon)$ treated as a constant, since it is so in practice. If

$$\frac{L}{\alpha} > \max\left\{\frac{|\mathcal{S}^*|^3}{\operatorname{vol}(\mathcal{S}^*)}, |\mathcal{S}^*|\right\},$$

then CDPR performs better than ISTA, up to constants. Since $\operatorname{vol}(\mathcal{S}^*) \geq |\mathcal{S}^*|$, this is, for example, satisfied when $\frac{L}{\alpha} > |\mathcal{S}^*|^2$. If

$$\frac{L}{\alpha} > \max\left\{\left(\frac{|\mathcal{S}^*|\widetilde{\operatorname{vol}}(\mathcal{S}^*)}{\operatorname{vol}(\mathcal{S}^*)}\right)^2, |\mathcal{S}^*|\right\},$$

then ASPR performs better than ISTA, up to constants and log factors. This is, for example, satisfied when $\frac{L}{\alpha} > |\mathcal{S}^*|^2$ or when $\frac{L}{\alpha} > |\mathcal{S}^*|$ and $\operatorname{vol}(\mathcal{S}^*) > |\mathcal{S}^*|^{5/2}$ since $\widetilde{\operatorname{vol}}(\mathcal{S}^*) \leq |\mathcal{S}^*|^2$. We note that for any graph,

If the convergence rates of CDPR and ASPR are dominated by $O(|\mathcal{S}^*|\operatorname{vol}(\mathcal{S}^*))$, then the algorithms perform similarly. However, if the time complexities of CDPR and ASPR are of orders $O(|\mathcal{S}^*|^3)$ and $O(|\mathcal{S}^*|\operatorname{vol}(\mathcal{S}^*)\sqrt{\frac{L}{\alpha}})$, respectively, then CDPR performs better than ASPR for

$$\frac{L}{\alpha} > \left(\frac{|\mathcal{S}^*|^2}{\widetilde{\operatorname{vol}}(\mathcal{S}^*)}\right)^2,$$

up to constants and log factors. We note that although Fountoulakis et al. (2019) describe their method as using $O(\operatorname{vol}(\mathcal{S}^*))$ memory, their ISTA solver actually only requires $O(|\mathcal{S}^*|)$ space, as it is enough to store the entries of the iterates and gradients corresponding to the good coordinates, whereas the gradient entries for bad coordinates can be discarded immediately after computation.