AN INEXACT PROXIMAL PATH-FOLLOWING ALGORITHM FOR CONSTRAINED CONVEX MINIMIZATION

QUOC TRAN-DINH*, ANASTASIOS KYRILLIDIS*, AND VOLKAN CEVHER*

Abstract. Many scientific and engineering applications feature large-scale non-smooth convex minimization problems over convex sets. In this paper, we address an important instance of this broad class where we assume that the non-smooth objective is equipped with a tractable proximity operator and that the convex constraints afford a self-concordant barrier. We provide a new joint treatment of proximal and self-concordant barrier concepts and illustrate that such problems can be efficiently solved without lifting problem dimensions. We propose an inexact path-following algorithmic framework and theoretically characterize the worst case convergence as well as computational complexity of this framework, and also analyze its behavior when the proximal subproblems are solved inexactly. To illustrate our framework, we apply its instances to both synthetic and real-world applications and illustrate their accuracy and scalability in large-scale settings. As an added bonus, we describe how our framework can obtain points on the Pareto frontier of regularized problems with self-concordant objectives in a tuning free fashion.

Key words. Inexact path-following algorithm, self-concordant barrier, tractable proximity, proximal-Newton method, constrained convex optimization.

1. Problem statement and motivation. We consider the following constrained convex minimization problem, which has myriad applications in diverse disciplines, including machine learning, signal processing, statistics, and control [10, 17, 18, 32]:

$$g^* := \min_{\mathbf{x}\in\Omega} g(\mathbf{x}). \tag{1.1}$$

Here, $\Omega \subseteq \mathbb{R}^n$ is a nonempty, closed and convex set and g is a (possibly) non-smooth convex function from $\mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}$.

Several powerful methods have been developed to solve structural instances of (1.1). Perhaps, the most famous one is the class of interior point methods (IPM) that solves standard conic programming problems in polynomial time; an non-exhaustive list includes linear programming, quadratic programming, second order cone programming, and semidefinite programming [7, 25]. The key structure exploited in conventional IPMs is the existence of a *barrier function* for Ω (cf., Section 2), while g is a linear or convex quadratic function. In such cases, one considers the penalized family of parametric composite convex optimization problems:

$$\min_{\mathbf{x}\in\mathbf{R}^n} \left\{ F(\mathbf{x}; t) := f(\mathbf{x}) + \frac{1}{t} g(\mathbf{x}) \right\}, \tag{1.2}$$

where $t > 0$ is referred to as a penalty parameter. By solving (1.2) for a sequence of decreasing t values, i.e., $t \downarrow 0^+$, we can trace the central path of solutions $\mathbf{x}_t^*$ of (1.2) as it converges to the solution $\mathbf{x}^*$ of (1.1).

To this end, we transform the constrained problem (1.1) into a family of unconstrained minimization problems depending on t and hence potentially easier to solve these problems, i.e., (1.2), than (1.1). Unfortunately, assuming no further structure of f, the resulting path-following scheme is not guaranteed to converge, and solving (1.2) becomes harder as $t \downarrow 0^+$, see, e.g., [22]. Fortunately, Nesterov [23] introduced the

*Laboratory for Information and Inference Systems (LIONS), École Polytechnique Fédérale de Lausanne (EPFL), CH1015 - Lausanne, Switzerland.
E-mail: {quoc.trandinh, anastasios.kyrillidis, volkan.cevher}@epfl.ch.

self-concordance concept (cf., Section 2 for definitions), which characterizes a broad collection of penalty functions f and guarantees the polynomial-solvability of (1.2), by sequentially using Newton methods.

In addition to the constraints, the non-smooth objectives g also have a direct impact on the computational effort. Such problems do occur frequently in applications: Examples include but are not limited to sparse concentration matrix estimation with ℓ_1-norm (eq. (11) in [28]), data clustering with ℓ_1-norm (SDP reformulation in Section 4.1 of [19]), spectral line estimation with atomic norms (eq. (2.6) in [34] and eq. (3.4) in [9]), etc. Since off-the-shelf IPMs usually approximate g^* by solving a sequence of *smooth* problems [22, 23], g in (1.2) must allow a reformulation where standard smooth solvers can be applied (i.e., disciplined convex optimization (DCO) [14]).

Unfortunately, the DCO approach can inflate the problem dimensions, and suffers from the curse-of-dimensionality. For instance, when the problem (1.1) can be formulated into a semidefinite program, the scaling factors (e.g., the Nesterov-Todd scaling factor [26]) can create memory bottlenecks by destroying the sparsity of the underlying problem (e.g., by leading to dense KKT matrices in Newton systems).

As a concrete example, we consider max-norm clustering [19], where we seek a clustering matrix $\mathbf{K}$ that minimizes disagreement with a given affinity matrix $\mathbf{A}$

$$\begin{array}{ll} \min\limits_{\mathbf{L},\mathbf{R},\mathbf{K}\in\mathbb{R}^{p\times p}} & \|\mathrm{vec}\,(\mathbf{K}-\mathbf{A})\|_1 \\ \text{s.t.} & \begin{bmatrix} \mathbf{L} & \mathbf{K} \\ \mathbf{K}^T & \mathbf{R} \end{bmatrix} \succ 0,\ \mathbf{L}_{ii}\leq 1,\ \mathbf{R}_{ii}\leq 1,\ i=1,\ldots,p. \end{array} \tag{1.3}$$

This non-smooth formulation affords rigorous theoretical guarantees for its solution quality and can be formulated as a standard conic program. Unfortunately, we need to add $\mathcal{O}(p^2)$ slack variables to smooth the ℓ_1-norm term and process the linear constraints. Consequently, the efficiency of conventional IPM's significantly degrade.

1.1. Our approach. In general, when the penalty function f has a Lipschitz continuous gradient [23] and g has a computable proximity operator (cf., Section 2 for definitions), several well-characterized solutions to (1.2) exist [4, 5, 23, 24]. However, to the best of our knowledge, there is no unified framework for path-following schemes of (1.2) where f is a self-concordant barrier (hence, *non-globally Lipschitz continuous gradient*) and g is a non-smooth term with *proximal tractability*.

To this end, we address (1.1) with a new *proximal* path-following scheme, which solves (1.2) for a sequence of *adaptively selected* parameters t_k. Our scheme guarantees the following: If $\mathbf{x}^k$ is an approximate solution of $\mathbf{x}^*_{t_k}$ for $t \leftarrow t_k$ (i.e., within δ accuracy), then our method produces an approximate solution $\mathbf{x}^{k+1}$ of $\mathbf{x}^*_{t_{k+1}}$ for $t \leftarrow t_{k+1}$ within the same accuracy δ via applying *only one* proximal-Newton (PN) step:

$$\mathbf{x}^{k+1} :\approx \operatorname*{argmin}_{\mathbf{x}\in\mathrm{dom}(F)} \Big\{ \nabla f(\mathbf{x}^k)^T(\mathbf{x}-\mathbf{x}^k) + \frac{1}{2}(\mathbf{x}-\mathbf{x}^k)^T\nabla^2 f(\mathbf{x}^k)(\mathbf{x}-\mathbf{x}^k) + \frac{1}{t_{k+1}} g(\mathbf{x}) \Big\}, \tag{1.4}$$

where $\mathrm{dom}\,(F) := \mathrm{dom}\,(f) \cap \mathrm{dom}\,(g)$. This is the workhorse of our framework in a manner similar to the Newton schemes for standard path-following interior point methods [23, 25].

We now highlight the two salient features of our scheme that set us apart from existing approaches: First, solving the proximal-Newton problem (1.4) has been a major research area over the last decade, broadly known as composite optimization, where many accurate and scalable algorithms are customized for different g functions [5, 6, 24]. Our path following scheme leverages such "fast" algorithms as a black-box,

while approximating (1.1). These methods are theoretically as fast as the advanced "Hessian-free" IPM techniques, which use conjugate gradients, since $\nabla^2 f(\mathbf{x}) \succ 0$ for self concordant-barriers. In contrast, as we handle the non-smooth term g directly with proximity operators, we retain the original problem structure (i.e., we do not inflate problem dimensions or add additional constraints).

Second, adaptively updating the regularization parameter in composite optimization problems has itself attracted a great deal of interest; cf., the class of homotopy and continuation methods [16]. Many of these approaches lose their theoretical guarantees (if any) when the composite minimization problem has a self-concordant smooth term instead of a Lipschitz continuous gradient smooth term. In contrast, our scheme provides a rigorous way of updating regularizer weights and can be easily adapted for applications with self-concordant data terms [17, 32, 28], where none of these methods apply.

Our contributions: Our specific contributions in this paper are as follows:

(a) We extend the notion of path-following to handle composite forms in order to approximately track the solution trajectory of (1.2). As a consequence, we obtain an approximate solution of (1.1) by controlling the parameter t to 0^+.
(b) We provide an explicit formula to adaptively update the parameter t with convergence guarantees, without any manual tuning strategy.
(c) We provide a theoretical analysis of the worst-case complexity of our scheme to obtain a sequence of δ-accurate solutions, as t varies, while allows one to solve the subproblem (1.4) *inexactly* up to a given accuracy. The worst-case complexity of our method remains the same as in conventional path-following interior point methods [23].

Paper outline. Section 2 recalls the definitions of self-concordant functions and barriers and sets up optimization preliminaries. Section 3 deals with the inexact proximal-Newton iteration scheme for solving (1.2) at a fixed value of the parameter t. Section 4 presents the path-following framework with inexact proximal-Newton iterations and analyzes its convergence and worst-case compexity. Section 5 specifies our framework to solve constrained convex minimization problems of the form (1.1). Section 6 presents numerical experiments that highlight the strengths and weaknesses of our framework. Technical proofs are given in the appendix.

2. Preliminaries.

In this section, we set up the necessary notation and definitions revolving around self-concordance. We provide a fixed-point characterization of the optimality condition for (1.2) and then describe key technical results in deriving our framework.

2.1. Basic definitions.

Given $\mathbf{x}, \mathbf{y} \in \mathbb{R}^n$, we use either $\mathbf{x}^T\mathbf{y}$ or $\langle \mathbf{x}, \mathbf{y} \rangle$ to denote the inner product in $\mathbb{R}^n$. For a proper, lower semicontinuous convex function f, we denote its domain by $\operatorname{dom}(f)$ (i.e., $\operatorname{dom}(f) := \{\mathbf{x} \in \mathbb{R}^n \mid f(\mathbf{x}) < +\infty\}$ and its subdifferential at $\mathbf{x}$ by $\partial f(\mathbf{x}) := \{\mathbf{v} \in \mathbb{R}^n \mid f(\mathbf{y}) \geq f(\mathbf{x}) + \langle \mathbf{v}, \mathbf{y} - \mathbf{x} \rangle,\ \forall \mathbf{y} \in \operatorname{dom}(f)\}$ [29]. We also define $\operatorname{Dom}(f) := \operatorname{cl}(\operatorname{dom}(f))$ the closure of $\operatorname{dom}(f)$.

For a given twice differentiable function f such that $\nabla^2 f(\mathbf{x}) \succ 0$ at $\mathbf{x} \in \operatorname{dom}(f)$, we define the local norm $\|\mathbf{u}\|_{\mathbf{x}} := \langle \nabla^2 f(\mathbf{x})\mathbf{u},\ \mathbf{u} \rangle^{1/2}$ for any $\mathbf{u} \in \mathbb{R}^n$ while the dual norm is given by $\|\mathbf{v}\|_{\mathbf{x}}^* := \max_{\|\mathbf{u}\|_{\mathbf{x}} \leq 1} \mathbf{u}^T\mathbf{v} = \langle \left(\nabla^2 f(\mathbf{x})\right)^{-1} \mathbf{v},\ \mathbf{v} \rangle^{1/2},\ \forall \mathbf{v} \in \mathbb{R}^n$. It is clear that the Cauchy-Schwarz inequality holds, i.e., $\langle \mathbf{u}, \mathbf{v} \rangle \leq \|\mathbf{v}\|_{\mathbf{x}} \|\mathbf{v}\|_{\mathbf{x}}^*$. For our analysis, we also use two simple convex functions $\omega(t) := t - \ln(1+t)$ for $t \geq 0$ and $\omega_*(t) := -t - \ln(1-t)$ for $t \in [0, 1)$, which are strictly increasing in their domain.

An important concept in this paper is the self-concordance property [23, 25].

DEFINITION 2.1. *A convex function $\varphi : \mathbb{R} \to \mathbb{R}$ is called standard self-concordant if $|\varphi'''(\tau)| \leq 2\varphi''(\tau)^{3/2}$, $\forall \tau \in \mathbb{R}$. A function $f : \mathbb{R}^n \to \mathbb{R}$ is self-concordant if, for any $\tau \in \mathbb{R}$, the function $\varphi(\tau) := f(\mathbf{x} + \tau\mathbf{v})$ is self-concordant for all $\mathbf{x} \in \mathrm{dom}\,(f)$ and $\mathbf{v} \in \mathbb{R}^n$ such that $\mathbf{x} + \tau\mathbf{v} \in \mathrm{dom}\,(f)$.*

DEFINITION 2.2. *A standard self-concordant function f is a ν-self-concordant barrier for the set $\mathrm{Dom}(f)$ with parameter $\nu > 0$, if $\sup_{\mathbf{u}\in\mathbb{R}^n} \left\{2\nabla f(\mathbf{x})^T\mathbf{u} - \|\mathbf{u}\|_{\mathbf{x}}^2\right\} \leq \nu$ for all $\mathbf{x} \in \mathrm{dom}\,(f)$.*

We note that when $\nabla^2 f$ is non-degenerate (particularly, $\mathrm{dom}\,(f)$ contains no straight line [23, Theorem 4.1.3.]), a ν-self-concordant function f satisfies

$$\|\nabla f(\mathbf{x})\|_{\mathbf{x}}^* \leq \sqrt{\nu}. \tag{2.1}$$

In addition, for any sequence $\left\{\mathbf{x}^k\right\} \subset \mathrm{dom}\,(f)$, if $\mathbf{x}^k \to \bar{\mathbf{x}} \in \partial(\mathrm{dom}\,(f))$, where $\partial(\mathrm{dom}\,(f))$ is the boundary of $\mathrm{dom}\,(f)$, then $f(\mathbf{x}^k) \to +\infty$. For more details on self-concordant functions and self-concordant barriers, we refer the reader to Chapter 4 of [23]. Several simple sets are equipped with a self-concordant barrier. For instance, $f_{\mathbb{R}^n_+}(\mathbf{x}) := -\sum_{i=1}^n \log(x_i)$ is an n-self-concordant barrier of the orthogonal cone $\mathbb{R}^n_+$, $f(\mathbf{x}) = -\log(t^2 - \|\mathbf{x}\|_2^2)$ is a 2-self-concordant barrier of the Lorentz cone $\mathcal{L}_{n+1} := \{(\mathbf{x}, t) \in \mathbb{R}^n \times \mathbb{R}_+ \mid \|\mathbf{x}\|_2 \leq t\}$, and the semidefinite cone $\mathcal{S}^n_+$ is endowed with an n-self-concordant barrier $f_{\mathcal{S}^n_+}(\mathbf{X}) := -\log\det(\mathbf{X})$.

Given these definitions, we are now ready to state our main assumption used throughout this paper.

ASSUMPTION A. 1. *The function f in (1.2) is a ν-self-concordant barrier with $\nu > 0$. The function $g : \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}$ is proper, lower semi-continuous, convex and possibly nonsmooth.*

2.2. Optimality condition of (1.2). Given $t > 0$, we assume that problem (1.2) has a solution $\mathbf{x}_t^*$. Since f is strictly convex, this solution is also unique. The optimality condition of (1.2) can be written as

$$\mathbf{0} \in \nabla f(\mathbf{x}_t^*) + \frac{1}{t}\partial g(\mathbf{x}_t^*). \tag{2.2}$$

The formula (2.2) expresses a monotone *generalized equation*, which has widely been studied in convex optimization; e.g., see [12]. If g is smooth, (2.2) reduces to $\nabla f(\mathbf{x}_t^*) + \frac{1}{t}\nabla g(\mathbf{x}_t^*) = \mathbf{0}$, a system of nonlinear equations. Any $\mathbf{x}_t^*$ satisfying (2.2) is called a stationary point of (1.2), which is also the global optimum of (1.2), for given t.

DEFINITION 2.3. *Let $\mathbf{x} \in \mathrm{dom}\,(F)$ such that $\nabla^2 f(\mathbf{x}) \succ 0$ and let $\mathbf{s} \in \mathbb{R}^n$ be an arbitrary given point. We define the operator $P_{\mathbf{x}}^g(\cdot; t)$ with an input $\mathbf{s}$ and a parameter $t > 0$ as follows:*

$$P_{\mathbf{x}}^g(\mathbf{s}; t) = \operatorname*{argmin}_{\mathbf{y}\in\mathbb{R}^n} \left\{\frac{1}{t}g(\mathbf{y}) + \frac{1}{2}\mathbf{y}^T\nabla^2 f(\mathbf{x})\mathbf{y} - \mathbf{s}^T\mathbf{y}\right\}. \tag{2.3}$$

Since $\nabla^2 f(\mathbf{x}) \succ 0$, we can write (2.3) as

$$P_{\mathbf{x}}^g(\mathbf{s}; t) = \operatorname*{argmin}_{\mathbf{y}\in\mathbb{R}^n} \left\{g(\mathbf{y}) + \frac{t}{2}\left\|\mathbf{y} - \nabla^2 f(\mathbf{x})^{-1}\mathbf{s}\right\|_{\nabla^2 f(\mathbf{x})}^2\right\},$$

which is the standard proximal operator of $\nabla^2 f(\mathbf{x})^{-1}\mathbf{s}$ with respect to the weighted norm $\|\cdot\|_{\nabla^2 f(\mathbf{x})}$. Given $\mathbf{x}$ and $\mathbf{s}$ as defined above, we define the following mapping:

$$S_{\mathbf{x}}(\mathbf{s}; t) := \nabla^2 f(\mathbf{x})\mathbf{s} - \nabla f(\mathbf{s}). \tag{2.4}$$

The optimality condition in (2.2) implies the following fixed-point characterization of the mapping $P^g_{\mathbf{x}}(\cdot;t)$. The proof can be found in [36].

LEMMA 2.4. *Let $t > 0$ be fixed. Then, the mapping $P^g_{\mathbf{x}}(\cdot;t)$ defined in* (2.3) *is co-coercive and therefore nonexpansive w.r.t. the local norms, i.e.:*

$$\text{[co-coercive]}: \quad \langle P^g_{\mathbf{x}}(\mathbf{u};t) - P^g_{\mathbf{x}}(\mathbf{v};t), \mathbf{u}-\mathbf{v}\rangle \geq \|P^g_{\mathbf{x}}(\mathbf{u};t) - P^g_{\mathbf{x}}(\mathbf{v};t)\|^2_{\mathbf{x}}, \tag{2.5}$$

$$\text{[nonexpansive]}: \quad \|P^g_{\mathbf{x}}(\mathbf{u};t) - P^g_{\mathbf{x}}(\mathbf{v};t)\|_{\mathbf{x}} \leq \|\mathbf{u}-\mathbf{v}\|^*_{\mathbf{x}}, \ \forall \mathbf{u},\mathbf{v}\in\mathbb{R}^n. \tag{2.6}$$

Furthermore, the following fixed-point characterization holds:

$$\mathbf{x}^*_t = P^g_{\mathbf{x}^*_t}\left(S_{\mathbf{x}^*_t}(\mathbf{x}^*_t;t);t\right), \tag{2.7}$$

where $\mathbf{x}^*_t \in \operatorname{dom}(F)$ is the minimizer of (1.2)*, i.e., $\mathbf{x}^*_t = \operatorname{argmin}_{\mathbf{x}\in\mathbb{R}^n} F(\mathbf{x};t)$.*

For our convergence analysis, we also need the following result.

LEMMA 2.5. *For fixed $t > 0$, let $\mathbf{x}^*_t$ be the unique solution of* (1.2)*. Then, for any $\mathbf{x} \in \operatorname{dom}(F)$, the following estimate holds:*

$$\omega\left(\|\mathbf{x}-\mathbf{x}^*_t\|_{\mathbf{x}^*_t}\right) \leq F(\mathbf{x};t) - F(\mathbf{x}^*_t;t). \tag{2.8}$$

Proof. By the self-concordance property of f for any $\mathbf{x} \in \operatorname{dom}(F)$, the convexity of g and (2.2) it follows that

$$\begin{aligned} F(\mathbf{x}) - F(\mathbf{x}^*_t) &= f(\mathbf{x}) - f(\mathbf{x}^*_t) + \frac{1}{t}\left(g(\mathbf{x}) - g(\mathbf{x}^*_t)\right) \\ &\geq \left\langle \nabla f(\mathbf{x}^*_t) + \frac{1}{t}\mathbf{v}^*_t, \mathbf{x}-\mathbf{x}^*_t\right\rangle + \omega\left(\|\mathbf{x}-\mathbf{x}^*_t\|_{\mathbf{x}^*_t}\right), \quad \forall \mathbf{v}^*_t \in \partial g(\mathbf{x}^*_t), \\ &\overset{(2.2)}{=} \omega\left(\|\mathbf{x}-\mathbf{x}^*_t\|_{\mathbf{x}^*_t}\right), \end{aligned}$$

which is indeed (2.8). □

3. Proximal-Newton iterations for fixed t. Let us consider the unconstrained problem (1.2) for a given fixed parameter value $t > 0$. Since f is self-concordant, we can approximate it around $\mathbf{x}^k \in \operatorname{dom}(F)$ via the second order Taylor series expansion:

$$Q(\mathbf{x};\mathbf{x}^k) := f(\mathbf{x}^k) + \nabla f(\mathbf{x}^k)^T(\mathbf{x}-\mathbf{x}^k) + \frac{1}{2}(\mathbf{x}-\mathbf{x}^k)^T\nabla^2 f(\mathbf{x}^k)(\mathbf{x}-\mathbf{x}^k). \tag{3.1}$$

Given this quadratic surrogate of f, we can approximate $F(\mathbf{x};t)$ around $\mathbf{x}^k$ as:

$$F^k(\mathbf{x};t) := Q(\mathbf{x};\mathbf{x}^k) + \frac{1}{t}g(\mathbf{x}). \tag{3.2}$$

Starting from an arbitrary initial point $\mathbf{x}^0 \in \operatorname{dom}(F)$ and given a fixed value $t > 0$, the *inexact* full-step proximal-Newton method for solving (1.2) generates a sequence $\{\mathbf{x}^k\}_{k\geq 0}$, by approximately minimizing the composite quadratic model (3.2) as

$$\mathbf{x}^{k+1} \approx \operatorname*{argmin}_{\mathbf{x}\in\operatorname{dom}(F)} F^k(\mathbf{x};t). \tag{3.3}$$

Here, the "approximation" sense ($\approx$) highlights the inability of numerical methods to iteratively solve (3.3) with *exact accuracy* in all cases and will be made precise in Definition 3.1 below.

However, since $\nabla^2 f(\mathbf{x}^k) \succ 0$, $\arg\min_{\mathbf{x}\in\mathrm{dom}(F)} F^k(\mathbf{x};t)$ is a strongly convex program and it has the unique *exact* solution $\bar{\mathbf{x}}^{k+1}$. Moreover, the following optimality condition holds

$$\mathbf{0} \in \nabla f(\mathbf{x}^k) + \nabla^2 f(\mathbf{x}^k)(\bar{\mathbf{x}}^{k+1} - \mathbf{x}^k) + \frac{1}{t}\partial g(\bar{\mathbf{x}}^{k+1}). \tag{3.4}$$

Due to (2.2), it is obvious to show that if $\bar{\mathbf{x}}^{k+1} \equiv \mathbf{x}^k$, then $\mathbf{x}^k$ is the optimal solution of (1.2) for fixed t.

3.1. Inexact solutions of (3.3). In practice, solving (3.3) *exactly* is *infeasible*. Thus, we can only solve (3.3) up to a given accuracy $\delta \geq 0$, using algorithmic solutions such as fast proximal-gradient methods [5, 23, 24].

DEFINITION 3.1. *Given $t > 0$ and a tolerance $\delta \geq 0$, a point $\mathbf{x}^{k+1} \in \mathrm{dom}\,(F)$ is called a δ-solution to (3.3) if*

$$F^k(\mathbf{x}^{k+1};t) - F^k(\bar{\mathbf{x}}^{k+1};t) \leq \frac{\delta^2}{2}, \tag{3.5}$$

where $\bar{\mathbf{x}}^{k+1} := \arg\min_{\mathbf{x}\in\mathrm{dom}(F)} F^k(\mathbf{x};t)$ is the exact solution of (3.3).

A useful inequality for our subsequent developments is given in the next lemma.

LEMMA 3.2. *Given fixed $t > 0$, let $\mathbf{x}^k$ be the inexact solution of (3.3) at the k-iteration and $\bar{\mathbf{x}}^{k+1}$ be the exact solution of (3.3) at the $(k+1)$-th iteration. Then, $\forall \mathbf{x} \in \mathrm{dom}\,(F)$, the following inequality holds:*

$$\frac{1}{2}\left\|\mathbf{x} - \bar{\mathbf{x}}^{k+1}\right\|^2_{\mathbf{x}^k} \leq F^k_t(\mathbf{x}) - F^k_t(\bar{\mathbf{x}}^{k+1}), \quad \forall \mathbf{x} \in \mathrm{dom}\,(F)\,. \tag{3.6}$$

Proof. Since $\nabla^2 f(\mathbf{x}^k) \succ 0$, by definition of $F(\mathbf{x}^k;t)$, the proof follows similar motions with the proof of Lemma 2.5, based on the optimality condition (3.4) and the convexity of the g term. □

This lemma, in combination with Definition 3.1, indicates that, if we can find a δ-solution, then

$$\left\|\mathbf{x}^{k+1} - \bar{\mathbf{x}}^{k+1}\right\|_{\mathbf{x}^k} \leq \delta. \tag{3.7}$$

3.2. Contraction property of inexact proximal-Newton iterations. In this subsection, we provide a theoretical characterization of the per-iteration behavior of the inexact full-step proximal-Newton scheme (3.3) for fixed $t > 0$. Let $\mathbf{x}^k \in \mathrm{dom}\,(F)$ be the δ-solution of (3.3) and let $\mathbf{x}^*_t$ be the exact solution of (1.2). We define

$$\lambda_k := \left\|\mathbf{x}^k - \mathbf{x}^*_t\right\|_{\mathbf{x}^*_t}, \tag{3.8}$$

as the *weighted* distance between $\mathbf{x}^k$ and $\mathbf{x}^*_t$. The following theorem characterizes the contraction properties of λ_k; the proof can be found in the appendix.

THEOREM 3.3. *Given $\mathbf{x}^k \in \mathrm{dom}\,(F)$, let $\mathbf{x}^{k+1}$ be a δ-solution of (3.3) for a given $\delta \geq 0$. Then, if $\lambda_k \in [0, 1 - \frac{\sqrt{2}}{2})$, we have*

$$\lambda_{k+1} \leq \frac{\delta}{1-\lambda_k} + \left(\frac{3 - 2\lambda_k}{1 - 4\lambda_k + 2\lambda_k^2}\right)\lambda_k^2. \tag{3.9}$$

Moreover, the right-hand side of (3.9) is nondecreasing w.r.t. λ_k and $\delta \geq 0$.

To illustrate the contraction properties of λ_k, we assume that the accuracy δ can be chosen such that $\delta := \xi\lambda_k$ for a given $\xi \in (0,1)$. Furthermore, let us define $\varphi(\lambda,\xi) := \frac{\xi}{1-\lambda} + \frac{3\lambda-2\lambda^2}{1-4\lambda+2\lambda^2}$ on $[0, 1-\frac{\sqrt{2}}{2})$. Then, (3.9) can be rewritten as

$$\lambda_{k+1} \leq \varphi(\lambda_k,\xi)\lambda_k. \tag{3.10}$$

From (3.10), we observe that, if $\varphi < 1$, the distance of $\mathbf{x}^{k+1}$ from $\mathbf{x}_t^*$ becomes smaller than that of $\mathbf{x}^k$, i.e., it ensures the convergence of the proximal-Newton scheme (3.3). To this end, we need to find a range of λ_k values, $\forall k$, (say Λ), such that $\varphi < 1$. Varying ξ, we can choose this range Λ: Since φ is non-decreasing, the larger ξ is, the smaller the range of Λ becomes. This observation is illustrated in Figure 3.1. For

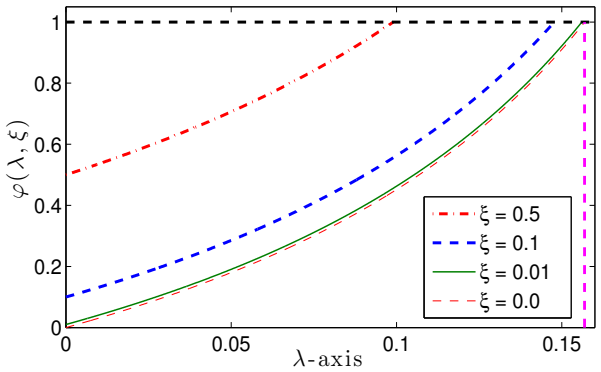


FIG. 3.1. *The behavior of the contraction factor function $\varphi(\lambda,\xi)$ at different values of ξ*

$\xi \in [0, 0.5]$, the interval where $\varphi < 1$ varies from $[0, 0.1]$ to $[0, 0.15]$. Moreover, when $\xi = 0.001$, the value of φ is very close to the case of $\xi = 0$. In practice, this suggests that if we set the accuracy $\delta < 10^{-3}$, the inexact scheme performs closely to the ideal case (i.e., $\delta = 0$).

Theoretically, if we assume that the subproblem (3.3) is solved exactly, then the estimate (3.9) reduces to $\bar{\lambda}_{k+1} \leq \left(\frac{3-2\bar{\lambda}_k}{1-4\bar{\lambda}_k+2\bar{\lambda}_k^2}\right)\bar{\lambda}_k^2$, where $\bar{\lambda}_k := \left\|\bar{\mathbf{x}}^k - \mathbf{x}_t^*\right\|_{\mathbf{x}_t^*}$. The algorithms and the convergence theory corresponding to this case are studied in [36].

A important consequence of Theorem 3.3 is the following corollary.

COROLLARY 3.4. *For a fixed $t > 0$ and a given constant $c > 0$, let $\{\mathbf{x}^k\}_{k\geq 0}$ be a sequence of δ-solutions, generated by solving (3.3) approximately.*

(a) If we choose δ and $\mathbf{x}^0$ such that $\delta \leq 0.15\lambda_k$, $\forall k \geq 0$, and $\lambda_0 \leq 0.1427$, then $\{\mathbf{x}^k\}_{k\geq 0}$ converges to $\mathbf{x}_t^*$ at a linear rate.

(b) If we choose δ and $\mathbf{x}^0$ such that $\delta \leq c\lambda_k^2$, $\forall k \geq 0$, and

$$\lambda_0 \in \left[0, \min\left\{0.15, \frac{1}{1.177c + 6.068}\right\}\right],$$

then $\{\mathbf{x}^k\}_{k\geq 0}$ converges to $\mathbf{x}_t^*$ at a quadratic rate.

Proof. (a). For $\delta := 0.15\lambda_k$, we consider the function $\hat{\varphi}(\lambda) := \frac{0.15}{1-\lambda} + \frac{3\lambda-2\lambda^2}{1-4\lambda+2\lambda^2}$. This function is increasing for $\lambda \in \Lambda := [0, 0.1427]$ and $\hat{\varphi}(\lambda) < 1, \forall\lambda \in \Lambda$. Therefore, it follows from (3.9) that $\lambda_{k+1} \leq \max_{\lambda\in\Lambda}\{\hat{\varphi}(\lambda)\}\cdot\lambda_k$ for $k \geq 0$, which implies $\{\lambda_k\}_{k\geq 0}$ converges to zero at a linear rate.

(b). For $\lambda_k \in [0, 0.15]$, we can see that the weight factor of the second term in the right hand side of (3.9), $\frac{3-2\lambda}{1-4\lambda+2\lambda^2}$, is increasing and moreover, $\frac{3-2\lambda}{1-4\lambda+2\lambda^2} \leq 6.068$. Thus, for $\delta \leq c\lambda_k^2$, we have

$$\lambda_{k+1} \leq \left(\frac{c}{1-\lambda_k} + 6.068 \right) \lambda_k^2 \leq (1.177c + 6.068)\, \lambda_k^2.$$

From this inequality, we can easily check that, if $\lambda_0 \leq \min\left\{0.15, \frac{1}{1.177c+6.068}\right\}$ then $\lambda_{k+1} \leq \lambda_k$. Moreover, $\{\lambda_k\}_{k\geq 0}$ converges to zero at a quadratic rate. ∎

4. A proximal path following framework. Our discussion so far focuses on the case of minimizing (3.3) for a fixed $t > 0$. Nevertheless, in order to solve the initial problem (1.1), one requires to trace the sequence of solutions $\{\mathbf{x}^k\}_{k\geq 0}$, as $t \downarrow 0^+$. For smooth self-concordant barrier function minimization problems, Nesterov in [23] presented a path following strategy where a *single Newton step* per iteration is used, for each well-chosen penalty parameter t_k. Here, we adopt a similar strategy to handle composite self-concordant barrier problems of the form (1.2) with a possibly nonsmooth convex function g, *mutatis mutandis*.

Our contribution lies at the adaptive selection of t: given an approximate solution of the proximal Newton step (1.4), we derive an update rule for the regularization parameter t. In stark contrast, classical path-following (homotopy or continuation) methods [13, 15] usually discretize the parameter t *a priori* and then solve (1.2) over this grid.

Our proximal path following scheme goes through the following motions: Starting from an initial value $t \leftarrow t_0$, we solve (1.2) to obtain δ-solution $\mathbf{x}^0$ to $\mathbf{x}^*_{t_0}$; the selection of t_0 is generally problem dependent. Section 4.3 describes a procedure on how to compute a starting point $\mathbf{x}^0$. Then, at each k-th iteration, the scheme *adaptively* updates t_k and uses this new penalty parameter to perform a *single* proximal-Newton (PN) iteration to approximately compute $\mathbf{x}^{k+1}$ which is provably close to $\mathbf{x}^*_{t_{k+1}}$. This strategy is illustrated in Figure 4.1.

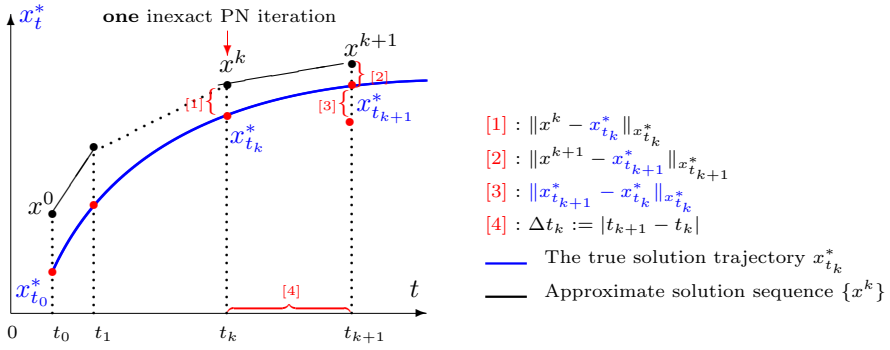


FIG. 4.1. *The approximate sequence* $\{x^k\}_{k\geq 0}$ *along the solution trajectory* x^*_t.

4.1. Quadratic convergence region. For our developments, we define $\tilde{\lambda}_k := \|\mathbf{x}^k - \mathbf{x}^*_{t_{k+1}}\|_{\mathbf{x}^*_{t_{k+1}}}$; we bring to the attention of the reader the differences with $\lambda_{k+1} := \|\mathbf{x}^{k+1} - \mathbf{x}^*_{t_{k+1}}\|_{\mathbf{x}^*_{t_{k+1}}}$. Given these definitions, for $t \equiv t_{k+1}$, (3.9) becomes

$$\lambda_{k+1} \leq \frac{\delta}{1-\tilde{\lambda}_k} + \left(\frac{3-2\tilde{\lambda}_k}{1-4\tilde{\lambda}_k+2\tilde{\lambda}_k^2} \right) \tilde{\lambda}_k^2, \tag{4.1}$$

provided that $0 \leq \tilde{\lambda}_k < 1 - \sqrt{2}/2$. Let us define the weighted distance between two solutions $\mathbf{x}^*_{t_{k+1}}$ and $\mathbf{x}^*_{t_k}$ w.r.t. two different values t_{k+1} and t_k of t as

$$\Delta_k := \|\mathbf{x}^*_{t_{k+1}} - \mathbf{x}^*_{t_k}\|_{\mathbf{x}^*_{t_{k+1}}}. \tag{4.2}$$

The following theorem shows that, for a range of values for Δ_k and δ, if $\lambda_k \leq \beta$, then at the $(k+1)$-th iteration, we maintain the property $\lambda_{k+1} \leq \beta$ for a given $\beta > 0$. The proof can be found in the appendix.

THEOREM 4.1. *Let $\beta \in (0, 0.15]$ be fixed. Assume that δ and Δ_k satisfy $\delta \leq 0.075\beta$ and $\Delta_k \leq \frac{\sqrt{\beta}-2.581\beta}{2.581+\sqrt{\beta}}$. Then, if $\lambda_k \leq \beta$, then our scheme guarantees that $\tilde{\lambda}_k \leq \frac{1}{2.581}\sqrt{\beta}$ and $\lambda_{k+1} \leq \beta$.*

Let us define $\mathcal{Q}_\beta^{t_k} := \left\{\mathbf{x}^k \in \mathrm{dom}\,(F) \mid \lambda_k \leq \beta\right\}$. We refer to $\mathcal{Q}_\beta^{t_k}$ as the *quadratic convergence region* of the inexact proximal-Newton iterations (3.3) for solving (1.2). For fixed $t_k > 0$, from Corollary 3.4, we can see that if the starting point $\mathbf{x}^0$ is chosen such that $\lambda_0 \in \mathcal{Q}_\beta^{t_k}$, then the whole sequence $\left\{\mathbf{x}^k\right\}$ generated by the proximal-Newton scheme belongs to $\mathcal{Q}_\beta^{t_k}$ and converges to $\mathbf{x}^*_{t_k}$, the solution of (1.2), at a quadratic rate. In plain words, Theorem 4.1 indicates that if δ-solution $\mathbf{x}^k$ is in the quadratic convergence region $\mathcal{Q}_\beta^{t_k}$ at $\mathbf{x}^*_{t_k}$ then, we can configure the proposed scheme such that the next δ-solution $\mathbf{x}^{k+1}$ remains in the quadratic convergence region $\mathcal{Q}_\beta^{t_{k+1}}$ at $\mathbf{x}^*_{t_{k+1}}$.

4.2. An adaptive update rule for t. Next, we show how we can update the penalty parameter t in our path-following scheme to ensure the condition on Δ_k in Theorem 4.1. The penalty parameter t is updated as

$$t_{k+1} := t_k + d_k, \tag{4.3}$$

where d_k is a decrement or an increment over the current penalty parameter t_k. The following lemma shows how we can choose d_k; the proof is provided in the Appendix.

LEMMA 4.2. *Let Δ_k be defined by (4.2) such that $\Delta_k < 1$ and the penalty parameter for the $(k+1)$-th iteration be updated by (4.3). Then, we have*

$$\frac{\Delta_k}{1+\Delta_k} \leq \frac{|d_k|}{t_k}\|\nabla^2 f(\mathbf{x}^*_{t_{k+1}})\|^*_{\mathbf{x}^*_{t_{k+1}}} \leq \frac{|d_k|}{t_k}\sqrt{\nu}. \tag{4.4}$$

Consequently, if we choose d_k such that $|d_k| \leq \frac{t_k}{\sqrt{\nu}}$, then $\Delta_k \leq \frac{|d_k|\sqrt{\nu}}{t_k - |d_k|\sqrt{\nu}}$.

Now, we combine Lemma 4.2 and Theorem 4.1 to establish an update rule for t_k. The condition $\Delta_k \leq \frac{\sqrt{\beta}-2.581\beta}{2.581+\sqrt{\beta}} =: C(\beta)$ in Theorem 4.1 holds if we force

$$\frac{|d_k|\sqrt{\nu}}{t_k - |d_k|\sqrt{\nu}} \leq C(\beta),$$

which leads to $|d_k| \leq \sigma_\beta \cdot t_k$, where $\sigma_\beta := \frac{C(\beta)}{(1+C(\beta))\sqrt{\nu}} \in (0,1)$. Then, based on Lemma 4.2, we can update t_k as

$$t_{k+1} := (1 \pm \sigma_\beta)t_k, \tag{4.5}$$

i.e., we can either increase t_k or decrease t_k by a factor $1 \pm \sigma_\beta$ at each iteration while preserving the properties of Lemma 4.2. For example, for $\beta = 0.05$, we have $C(\beta) \approx 0.033715$ and $\sigma_\beta \approx \frac{0.033715}{\sqrt{\nu}}$.

4.3. Finding a starting point. In order to initialize the algorithm, we need to find a point $\mathbf{x}^0_{t_0} \in \mathrm{dom}\,(F)$ such that $\lambda_0 := \|\mathbf{x}^0_{t_0} - \mathbf{x}^*_{t_0}\|_{\mathbf{x}^*_{t_0}} \leq \beta$ for given $\beta \in (0, 0.15]$ as indicated in Theorem 4.1. To achieve this goal, we apply the *inexact damped* proximal-Newton method: Given $t_0 > 0$ and an initial point $\mathbf{x}^0 \in \mathrm{dom}\,(F)$, we generate a sequence $\{\mathbf{x}^j\}_{j\geq 0}$, starting from $\mathbf{x}^0$, by computing

$$\mathbf{x}^{j+1} := \mathbf{x}^j + \alpha_j \mathbf{d}^j, \text{ with } \mathbf{d}^j := \mathbf{s}^j - \mathbf{x}^j, \tag{4.6}$$

where $\alpha_j \in (0, 1]$ is a given step size which will be defined later, $\mathbf{d}^j$ is the approximate proximal-Newton search direction, and $\mathbf{s}^j$ is a trial point obtained by approximately solving the following convex subproblem:

$$\mathbf{s}^j \approx \bar{\mathbf{s}}^j := \arg\min_{\mathbf{s}\in\mathrm{dom}(F)} F^j(\mathbf{s}; t_0), \tag{4.7}$$

Again, we denote with $\bar{\mathbf{s}}^j$ the exact solution of (4.7) and the approximation "$\approx$" is defined as in Definition 3.1 with the accuracy $\delta \geq 0$.

It follows from (3.5) that

$$F^j(\bar{\mathbf{s}}^j; t_0) \leq F^j(\mathbf{s}^j; t_0) \leq F^j(\bar{\mathbf{s}}^j; t_0) + \frac{\delta^2}{2}. \tag{4.8}$$

Given the inexact proximal-Newton search direction $\mathbf{d}^j$, we define $\zeta_j := \|\mathbf{d}^j\|_{\mathbf{x}^j}$ as the inexact proximal-Newton decrement [37]. The following lemma shows how to choose the step size α_j; the proof is given in the appendix.

LEMMA 4.3. *Let $\{\mathbf{x}^j\}_{j\geq 0}$ be a sequence generated by the inexact damped proximal-Newton scheme (4.6). If we choose the accuracy δ_j such that $\delta \leq \zeta_j$ then, with $\alpha_j := \frac{\zeta_j - \delta}{(1+\zeta_j-\delta)\zeta_j} \in [0, 1]$ we have*

$$F(\mathbf{x}^{j+1}; t_0) - F(\mathbf{x}^*_{t_0}; t_0) \leq F(\mathbf{x}^j; t_0) - F(\mathbf{x}^*_{t_0}; t_0) - \omega(\zeta_j - \delta), \quad \forall j \geq 0. \tag{4.9}$$

Moreover, for fixed $\delta \leq \zeta_j$, the above step size α_j is optimal.

At each iteration, assume $\delta := \kappa\zeta_j$ where $\kappa \in (0, 1)$ (see Section 6). For a given $\beta \in (0, 0.15]$, from Lemma 2.5 we deduce that $\lambda_0 := \left\|\mathbf{x}^0_{t_0} - \mathbf{x}^*_{t_0}\right\|_{\mathbf{x}^*_{t_0}} \leq \beta$ if $F(\mathbf{x}^0_{t_0}; t_0) - F(\mathbf{x}^*_{t_0}; t_0) \leq \omega(\beta)$. To achieve such bound, assume that we can estimate an upper bound of the quantity $\gamma_0 \geq F(\mathbf{x}^0; t_0) - F(\mathbf{x}^*_{t_0}; t_0) \geq 0$. By using the estimate (4.9), we deduce

$$\begin{aligned} F(\mathbf{x}^{j+1}; t_0) - F(\mathbf{x}^*_{t_0}; t_0) &\leq F(\mathbf{x}^0; t_0) - F(\mathbf{x}^*_{t_0}; t_0) - \sum_{l=0}^{j} \omega\left((1-\kappa)\zeta_l\right) \\ &\leq \gamma_0 - \sum_{l=0}^{j} \omega\left((1-\kappa)\zeta_l\right). \end{aligned} \tag{4.10}$$

We now define

$$\Gamma_{j_{\max}} := \sum_{l=0}^{j_{\max}} \omega\left((1-\kappa)\zeta_l\right) + \omega(\beta). \tag{4.11}$$

Then, we observe that, if $\Gamma_{j_{\max}} \geq \gamma_0$, then one can guarantee $\lambda_0 \leq \beta$, where $\mathbf{x}^0_{t_0} := \mathbf{x}^{j_{\max}+1}$. We note that for $j \leq j_{\max}$, we have $\zeta_j \geq \beta$. Therefore, if we choose $\delta := \kappa\beta$ for some $\kappa \in (0, 1)$, we satisfy the assumption $\delta < \zeta_j$.

Algorithm 1: Inexact path following proximal Newton algorithm

Input: Choose $t_0 > 0$, $\beta \in (0, 0.15]$, $\kappa \in (0,1)$ and $\mathbf{x}^0 \in \mathrm{dom}\,(F)$. Compute an upper bound $\gamma_0 > 0$ for $F(\mathbf{x}^0; t_0) - F(\mathbf{x}^*_{t_0}; t_0)$.

Initialize: $\Gamma_{-1} := \omega(\beta)$, $C(\beta) := \frac{\sqrt{\beta} - 2.581\beta}{2.581 + \sqrt{\beta}}$, $\sigma_\beta := \frac{C(\beta)}{(1 + C(\beta))\sqrt{\nu}}$.

Phase I: Computing an initial point

for $j = 0, \cdots, j_{\max}$

1: Compute $\mathbf{d}^j$ via (4.6) by solving (4.7) approximately up to $\delta := \kappa\beta$.

2: Compute $\zeta_j := \|\mathbf{d}^j\|_{\mathbf{x}^j}$.

3: $\Gamma_j := \Gamma_{j-1} + \omega((1-\kappa)\zeta_j)$.

if $\Gamma_j \geq \gamma_0$ **then**

4: $\mathbf{x}^0_{t_0} := \mathbf{x}^j$.

5: **break**

end if

6: $\mathbf{x}^{j+1} := \mathbf{x}^j + \alpha_j \mathbf{d}^j$ where $\alpha_j := (1-\kappa)\left[1 + (1-\kappa)\zeta_j\right]^{-1}$.

end for

Phase II: Path following iteration

for $k = 0, \ldots, k_{\max}$ or **while** stopping criterion is not met

7: $t_{k+1} := (1 \pm \sigma_\beta)\, t_k$.

8: Given $\mathbf{x}^k_{t_k}$, solve (1.4) approximately up to $\delta_k \leq 0.075\beta$ to obtain $\mathbf{x}^{k+1}_{t_{k+1}}$.

end

4.4. Our prototype scheme. The proposed algorithm is given in Algorithm 1. The main steps are Step 1 and Step 7, where we need to solve two convex subproblems of the form (3.3)-(4.7). For certain regularizers g such as the ℓ_1-norm, the nuclear norm or the indicator of a simple convex set, there exist several efficient algorithms for this kind of optimization problems [5, 6, 23, 24]. The update rule for t_k at Step 6 of Phase II is based on the worst-case estimate (4.4). In practice, we can adaptively update t_k as discussed later in Section 6.

4.5. Convergence analysis. In this subsection, we provide the full complexity analysis for Phase I and Phase II of Algorithm 1 separately. Since we consider the case $t \downarrow 0^+$, we assume $t_{k+1} = (1 - \sigma_\beta)t_k$ and $t_0 \gg 0^+$. The worst-case complexity estimate of Algorithm 1 is given in the following theorem.

THEOREM 4.4. *The number of iterations required in Phase I to find $\mathbf{x}^0_{t_0} \in \mathrm{dom}\,(F)$ such that $\lambda_0 \leq \beta$ is at most*

$$j_{\max} := \left\lfloor \frac{F(\mathbf{x}^0; t_0) - F(\mathbf{x}^*_{t_0}; t_0)}{\omega((1-\kappa)\beta)} \right\rfloor + 1. \tag{4.12}$$

The number of iterations required in Phase II to reach the approximate solution $\mathbf{x}^{k_{\max}}$ of $\mathbf{x}^*_{t_f}$, where t_f is a user-defined value, close to 0^+ and $\lambda_{k_{\max}} \leq \beta$, is at most

$$k_{\max} := \left\lfloor \frac{\ln(t_0/t_f)}{-\ln(1-\sigma_\beta)} \right\rfloor + 1, \tag{4.13}$$

where σ_β is given by (4.5). *The worst-case complexity of Phase II is $\mathcal{O}\left(\sqrt{\nu}\ln\left(\frac{t_0}{t_f}\right)\right)$.*

Proof. From Lemma 4.3 and the choice of δ we have

$$F(\mathbf{x}^{j+1};t_0) - F(\mathbf{x}^*_{t_0};t_0) \le F(\mathbf{x}^j;t_0) - F(\mathbf{x}^*_{t_0};t_0) - \omega\left((1-\kappa)\zeta_j\right), \quad \forall j \ge 0.$$

Moreover,

$$\begin{aligned} 0 \le F(\mathbf{x}^j;t_0) - F(\mathbf{x}^*_{t_0};t_0) &\le F(\mathbf{x}^0;t_0) - F(\mathbf{x}^*_{t_0};t_0) - \sum_{l=0}^{j-1} \omega\left((1-\kappa)\zeta_l\right) \\ &\le F(\mathbf{x}^j;t_0) - F(\mathbf{x}^*_{t_0};t_0) - j\omega((1-\kappa)\beta). \end{aligned}$$

This implies

$$j \le \frac{F(\mathbf{x}^0;t_0) - F(\mathbf{x}^*_{t_0};t_0)}{\omega((1-\kappa)\beta)},$$

which shows that the number of iterations to obtain $\lambda_0 \le \beta$ is at most $j_{\max}$.

For Phase II, by induction, we have $t_k = t_0(1-\sigma_\beta)^k$. Since we desire $t_k \le t_f$, which leads to $k \ge \frac{\ln(t_0/t_f)}{-\ln(1-\sigma_\beta)}$.

Finally, note that $\ln(1-\sigma_\beta) \approx \sigma_\beta$. By the definition of $\sigma_\beta = \frac{C(\beta)}{(C(\beta)+1)\sqrt{\nu}}$, we obtain that the worst-case complexity of Phase II, which is $\mathcal{O}\left(\sqrt{\nu}\ln\left(\frac{t_0}{t_f}\right)\right)$. ∎

5. Application to constrained convex optimization. We now specify Algorithm 1 to solve the constrained convex programming problem of the form (1.1). We assume that f is the ν - self-concordant barrier associated with Ω such that $\mathrm{Dom}(f) \equiv \Omega$. First, we show the relation between the solution of the constrained problem (1.1) and the parametric problem (1.2) in the following lemma, whose proof can be found in the appendix.

LEMMA 5.1. *Let $\mathbf{x}^*$ be a solution of* (1.1) *and $\mathbf{x}^*_t$ be the solution of* (1.2) *at a given $t > 0$, i.e., $\mathbf{x}^*_t \in \mathrm{int}(\Omega)$. Then, for any $t > 0$, $\mathbf{x}^*_t$ is strictly feasible to* (1.2) *and*

$$0 \le g(\mathbf{x}^*_t) - g(\mathbf{x}^*) \le t\nu. \tag{5.1}$$

Let $\mathbf{x}^{k+1}$ be the point generated by Algorithm 1 *at the iteration $k+1$ and $\mathbf{x}^*_{t_{k+1}}$ be the solution of* (1.2) *at $t = t_{k+1}$. Then*

$$-\nu t_{k+1} \le g(\mathbf{x}^{k+1}) - g(\mathbf{x}^*_{t_{k+1}}) \le t_{k+1}\left(\sqrt{\nu}\frac{\lambda_{k+1}}{1-\tilde{\lambda}_k} + \frac{\tilde{\lambda}_k}{(1-\tilde{\lambda}_k)^2}\left(\lambda_{k+1}+\tilde{\lambda}_k+\delta\right) + \frac{\delta^2}{2}\right). \tag{5.2}$$

provided that $\tilde{\lambda}_k < 1$. Consequently, it holds that

$$-\nu t_{k+1} \le g(\mathbf{x}^{k+1}) - g(\mathbf{x}^*) \le t_{k+1}\psi(\nu,\tilde{\lambda}_k,\lambda_{k+1},\delta), \tag{5.3}$$

where $\psi(\nu,\tilde{\lambda}_k,\lambda_{k+1},\delta) := \nu + \sqrt{\nu}\frac{\lambda_{k+1}}{1-\tilde{\lambda}_k} + \frac{\tilde{\lambda}_k}{(1-\tilde{\lambda}_k)^2}\left(\lambda_{k+1}+\tilde{\lambda}_k+\delta\right) + \frac{\delta^2}{2}$ and $\tilde{\lambda}_k < 1$.

The estimate (5.1) in Lemma 5.1 shows that for sufficiently small $t > 0$, the solution $\mathbf{x}^*_t$ of (1.2) approximates the solution $\mathbf{x}^*$ of (1.1), i.e. $g(\mathbf{x}^*_t) \to g(\mathbf{x}^*)$ as $t \downarrow 0^+$. The estimate (5.3) in Lemma 5.1 suggests that if a sequence $\left\{(\mathbf{x}^k,t_k)\right\}_{k\ge0}$ is generated by Algorithm 1 for $t_f \le \varepsilon$ then $\left\{\mathbf{x}^k\right\}_{k\ge0}$ converges to $\mathbf{x}^*$ provided that the parameter t_k is updated as $t_{k+1} := (1-\sigma_\beta)t_k$ and $\delta \le \bar{\delta}$ (See Theorem 4.1).

If we apply Algorithm 1 to solve the constrained optimization problem (1.1), then we need to change the stopping criterion as $t_f \leq \varepsilon$ for a given accuracy $\varepsilon > 0$. Then the convergence of Algorithm 1 for solving (1.1) is given in the following theorem.

THEOREM 5.2. *Let* $\left\{(\mathbf{x}^k, t_k)\right\}_{k\geq 0}$ *be a sequence generated by Algorithm 1 for solving* (1.1)*. Then, after* $k_{\max}$ *iterations in Phase II, we have the following bound*

$$\left|g(\mathbf{x}^{k_{\max}}) - g(\mathbf{x}^*)\right| \leq \psi(\beta,\nu) t_{k_{\max}}, \tag{5.4}$$

where $\psi(\beta,\nu) := \nu + \sqrt{\nu}\frac{\beta}{1-0.4\sqrt{\beta}} + \frac{0.4\sqrt{\beta}}{(1-0.4\sqrt{\beta})^2}\left(1.075\beta + 0.4\sqrt{\beta}\right) + 0.003\beta^2$ *is a constant.*

Consequently, the worst-case analytical complexity of Phase II in Algorithm 1 to achieve an ε*-optimal solution, i.e.,* $\left|g(\mathbf{x}^{k_{\max}}) - g(\mathbf{x}^*)\right| \leq \varepsilon$*, is* $\mathcal{O}\left(\sqrt{\nu}\log(\frac{t_0\psi(\beta,\nu)}{\varepsilon})\right)$.

Proof. By the definition of ψ in Lemma 5.1 we can easily show that $\psi(\beta,\nu) \geq \nu$. On one hand, using this relation and (5.3), we have $\left|g(\mathbf{x}^{k+1}) - g(\mathbf{x}^*)\right| \leq \psi(\beta,\nu)t_{k+1}$. On the other hand, by induction, we have $t_k = (1-\sigma_\beta)^k t_0$ after k iterations. Therefore, if $(1-\sigma_\beta)^k t_0\psi(\beta,\nu) \leq \varepsilon$, we can conclude that $\left|g(\mathbf{x}^k) - g(\mathbf{x}^*)\right| \leq \varepsilon$. The last condition leads to $k \geq \frac{\log\left(\frac{t_0\psi(\beta,\nu)}{\varepsilon}\right)}{-\log(1-\sigma_\beta)}$. Since $\log(1-\sigma_\beta) \approx -\sigma_\beta$, we conclude that the worst-case complexity of Phase II in Algorithm 1 is $\mathcal{O}\left(\sqrt{\nu}\log(\frac{t_0\psi(\beta,\nu)}{\varepsilon})\right)$. ∎

6. Numerical experiments. In this section, we first discuss the implementation aspects of Algorithm 1. Next, we show how to customize this algorithm to solve a standard convex programming problem. Then, we provide three numerical examples: The first example is a synthetic low-rank approximation problem with additional constraints to highlight the inefficiency of off-the-self solvers. The second one is an application to clustering using max-norm as a concrete example for constrained convex optimization. The third example is an application to graph learning where we track the approximate solution of this problem along the regularization parameter horizon.

6.1. Implementation issues. Some fundamental implementation issues in Algorithm 1 are the following:

Methods for subproblems (3.3) *and* (4.7) *and warm-start.* The main ingredient in Algorithm 1 is the solution of (3.3) and (4.7). The more efficiently this problem is solved, the faster Algorithm 1 becomes. For certain classes of g, e.g., ℓ_1-norm, nuclear norm, atomic norm or simple projections, this problem is well-studied.

Subproblems (3.3) and (4.7) have the same structure over the iterations. This observation can be exploited a priori by using the similarity between $\nabla f(\mathbf{x}^{k-1})$, $\nabla^2 f(\mathbf{x}^{k-1})$ and $\nabla f(\mathbf{x}^k)$, $\nabla^2 f(\mathbf{x}^k)$, for each k. Since evaluating ∇f and $\nabla^2 f$ is the most costly part in the subsolvers, exploiting properly the problem structure for computing these quantities can accelerate the algorithm (see the examples below).

Second, (3.3) and (4.7) is strongly convex. Several first order methods can be applied and yield a linear convergence [5, 23, 24]. When g is the indicator of a polytope or a convex quadratic set (e.g., Euclidian balls), it turns out to be a quadratic program or a quadratically constrained quadratic program. Efficiency of solving this problem is well-understood.

Finally, warm-start strategies is key for efficiently solving (3.3) and (4.7). Given that the information from the previous iteration is available, the distance bewteen $\mathbf{x}^k$ and $\mathbf{x}^{k+1}$ is usually small. This observation suggests us to initialize the subsolvers with the solution provided by the previous iteration. Note that warm-start is very important in active-set methods [27], which can be used as a workhorse for (3.3) and (4.7).

Adaptive parameter update. Since the update rule $t_{k+1} := (1 \pm \sigma_\beta)t_k$ is based on the worst-case estimate of σ_β, it is better to replace it by an adaptive factor σ_k for acceleration. In fact, from the proof of Lemma 4.2 we can derive

$$t_{k+1}(1+\Delta_k)^{-1}\Delta_k \leq |d_k| \left\|\nabla f(\mathbf{x}^*_{t_k})\right\|^*_{\mathbf{x}^*_{t_{k+1}}}. \tag{6.1}$$

First, one can show that $\|\nabla f(\mathbf{x}^*_{t_k})\|^*_{\mathbf{x}^*_{t_{k+1}}} \leq (1-\tilde{\lambda}_k)^{-1}\|\nabla f(\mathbf{x}^*_{t_k})\|^*_{\mathbf{x}^k}$. Second, by the triangle inequality, we have $\|\nabla f(\mathbf{x}^*_{t_k})\|^*_{\mathbf{x}^k} \leq \|\nabla f(\mathbf{x}^*_{t_k}) - \nabla f(\mathbf{x}^k)\|^*_{\mathbf{x}^k} + \|\nabla f(\mathbf{x}^k)\|^*_{\mathbf{x}^k}$. However, since $\|\nabla f(\mathbf{x}^*_{t_k}) - \nabla f(\mathbf{x}^k)\|^*_{\mathbf{x}^k} \leq (1-\lambda_k)^{-1}\lambda_k$, the last inequality leads to $\|\nabla f(\mathbf{x}^*_{t_k})\|^*_{\mathbf{x}^k} \leq (1-\lambda_k)^{-1}\lambda_k + \|\nabla f(\mathbf{x}^k)\|^*_{\mathbf{x}^k}$. Combining all these derivations, we eventually get

$$\|\nabla f(\mathbf{x}^*_{t_k})\|^*_{\mathbf{x}^*_{t_{k+1}}} \leq (1-\tilde{\lambda}_k)^{-1}\left((1-\lambda_k)^{-1}\lambda_k + \|\nabla f(\mathbf{x}^k)\|^*_{\mathbf{x}^k}\right). \tag{6.2}$$

From Theorem 4.1, we have $\lambda_k \leq \beta$ and $\tilde{\lambda}_k \leq 0.3874\sqrt{\beta}$. Then, if we define

$$\begin{aligned} R_k(\beta) &:= (1-0.3874\sqrt{\beta})^{-1}\left((1-\beta)^{-1}\beta + \left\|\nabla f(\mathbf{x}^k)\right\|^*_{\mathbf{x}^k}\right) \\ &\leq (1-0.3874\sqrt{\beta})^{-1}\left((1-\beta)^{-1}\beta + \sqrt{\nu}\right), \end{aligned} \tag{6.3}$$

then, we can derive the update rule for t_k as $t_{k+1} = (1 \pm \sigma_k)t_k$, where σ_k is given as

$$\sigma_k := \max\left\{\frac{C(\beta)}{C(\beta) + (1-C(\beta))R_k(\beta)}, \sigma_\beta\right\} \in (0,1), \tag{6.4}$$

and $C(\beta)$ and σ_β are given in the previous section. A similar strategy for updating t in the case $f(\mathbf{x})$ is replaced by $f(\mathbf{x}) + \mathbf{c}^T\mathbf{x}$ can be derived by using the same trick, as used in the third numerical example below.

6.2. Instances of Algorithm 1. Algorithm 1 can be customized to solve a broad class of constrained convex problems of the form:

$$\begin{array}{ll} \min\limits_{\mathbf{x}\in\mathbf{R}^n} & h(\mathbf{x}) \\ \text{s.t} & \mathbf{x} \in \mathcal{C} \cap \Omega, \end{array} \tag{6.5}$$

where h is a proper, lower semicontinuous and convex function, $\mathcal{C}$ is a nonempty, closed and convex set, Ω is also a nonempty, closed and convex endowed with a ν-self-concordant barrier f. Let $g(\mathbf{x}) := h(\mathbf{x}) + \delta_{\mathcal{C}}(\mathbf{x})$, where $\delta_{\mathcal{C}}$ is the indicator function of $\mathcal{C}$. Then, problem (6.5) can equivalently be converted into (1.1).

As a concrete example, we show that Algorithm 1 can be customized to solve the constrained problems of the form (1.1) with additional linear equality constraints $\mathbf{Ax} = \mathbf{b}$. For simplicity of discussion, let us consider the following standard quadratic conic programming problem:

$$\begin{array}{ll} \min\limits_{\mathbf{x}\in\mathcal{K}} & \frac{1}{2}\mathbf{x}^T\mathbf{Q}\mathbf{x} + \mathbf{q}^T\mathbf{x} \\ \text{s.t.} & \mathbf{Ax} = \mathbf{b}, \end{array} \tag{6.6}$$

where $\mathbf{Q}$ is a symmetric positive semidefinite and $\mathcal{K}$ is a proper, closed, self-dual cone in $\mathbf{R}^n$ (including positive semidefinite cone), which is endowed with a ν-self-concordant barrier f. It is also possible to include inequality constraints $\mathbf{Bx} \leq \mathbf{c}$.

In order to customize Algorithm 1 for solving (6.6), we define $g(\mathbf{x}) := \frac{1}{2}\mathbf{x}^T\mathbf{Q}\mathbf{x} + \mathbf{q}^T\mathbf{x} + \delta_{\mathcal{C}}(\mathbf{x})$, where $\delta_{\mathcal{C}}$ is the indicator function of $\mathcal{C} := \{\mathbf{x} \in \mathbf{R}^n \mid \mathbf{A}\mathbf{x} = \mathbf{b}\}$. Then, problem (6.6) can be cast into (1.1). In principle, we can apply Algorithm 1 to solve the resulting problem. Now, let us consider the corresponding convex subproblem (3.3) associated with (6.6) as follows

$$\min_{\mathbf{x}\in\mathrm{int}(\mathcal{K})}\Big\{\frac{1}{2}\mathbf{x}^T\left(t\nabla^2 f(\mathbf{x}^k) + \mathbf{Q}\right)\mathbf{x} + \left(\mathbf{q}+t\nabla f(\mathbf{x}^k)-t\nabla^2 f(\mathbf{x}^k)\mathbf{x}^k\right)^T\mathbf{x} + \delta_{\mathcal{C}}(\mathbf{x})\Big\}. \tag{6.7}$$

The optimality condition for this problem becomes

$$\begin{cases}\left(\mathbf{Q} + t\nabla^2 f(\mathbf{x}^k)\right)\mathbf{x} + \mathbf{q} + t\nabla f(\mathbf{x}^k) - t\nabla^2 f(\mathbf{x}^k)\mathbf{x}^k + \mathbf{A}^T\mathbf{y} & = 0,\\ \mathbf{A}\mathbf{x} - \mathbf{b} & = 0.\end{cases} \tag{6.8}$$

Here, $\mathbf{y}$ is the Lagrange multiplier associated with the equality constraints $\mathbf{A}\mathbf{x}-\mathbf{b} = 0$. Let us define $\mathbf{d} := \mathbf{x} - \mathbf{x}^k$, then we can write (6.8) as follows

$$\begin{pmatrix}\mathbf{Q} + t\nabla^2 f(\mathbf{x}^k) & \mathbf{A}^T\\ \mathbf{A} & \mathbf{0}\end{pmatrix}\begin{pmatrix}\mathbf{d}\\ \mathbf{y}\end{pmatrix} = \begin{pmatrix}-\mathbf{q} - \mathbf{Q}\mathbf{x}^k - t\nabla f(\mathbf{x}^k)\\ \mathbf{b} - \mathbf{A}\mathbf{x}^k\end{pmatrix}. \tag{6.9}$$

Solving this linear system provides us a Newton search direction for Algorithm 1. In fact, this linear system (6.9) coincides with the system of computing Newton direction in standard primal interior-point methods for solving (6.6) directly, see, e.g., [7, 25, 30, 39].

6.3. Low-rank SDP matrix approximation. To illustrate the scalability and accuracy of the proposed path-following scheme, we consider the following matrix approximation problem:

$$\begin{array}{ll}\min\limits_{\mathbf{X}} & \rho\left\|\mathrm{vec}\left(\mathbf{X}-\mathbf{M}\right)\right\|_1 + (1-\rho)\mathrm{tr}\left(\mathbf{X}\right)\\ \text{s.t.} & \mathbf{X}\succeq 0,\ \mathbf{L}_{ij}\le\mathbf{X}_{ij}\le\mathbf{U}_{ij},\ i,j=1,\ldots,n.\end{array} \tag{6.10}$$

Here $\mathbf{M}\in\mathbb{R}^{n\times n}$ is a given matrix (not necessarily positive definite); $\rho\in[0,1]$ is a given regularization parameter and $\mathbf{L}$ and $\mathbf{U}$ are the element-wise lower and upper bound of $\mathbf{M}$. Problem (6.10) is a convex relaxation of the problem of approximating $\mathbf{M}$ by a low-rank and positive semidefinite matrix $\mathbf{X}$. Here, the trace-norm is used to approximate the rank of $\mathbf{X}$ and $\|\cdot\|_1$ is used to measure the distance from $\mathbf{X}$ to $\mathbf{M}$.

Let $\Omega := \mathcal{S}^n_{++}$ the cone of symmetric positive semidefinite matrices, and $g(\mathbf{X}) := \rho\left\|\mathrm{vec}\left(\mathbf{X}-\mathbf{M}\right)\right\|_1 + (1-\rho)\mathrm{tr}\left(\mathbf{X}\right) + \delta_{[\mathbf{L},\mathbf{U}]}(\mathbf{X})$, where $\delta_{[\mathbf{L},\mathbf{U}]}$ is the indicator function of the interval

$$[\mathbf{L},\mathbf{U}] := \left\{\mathbf{X}\in\mathcal{S}^n \mid \mathbf{L}_{ij}\le\mathbf{X}_{ij}\le\mathbf{U}_{ij},\ i,j=1,\cdots,n\right\}.$$

Since $f(\mathbf{X}) := -\log\det(\mathbf{X})$ is the standard barrier function of Ω, we can reformulate (6.10) in the form of (1.1).

In this example, we test Algorithm 1 and compare it with two standard interior point solvers, called `SDPT3` [38] and `SeDuMi` [33]. The parameters are configured as follows. We choose $t_0 := 10^{-2}$ and terminate the algorithm if $t_k \le 10^{-7}$. The starting point $\mathbf{X}^0$ is set to $\mathbf{X}^0 := 0.1\mathbb{I}$, where $\mathbb{I}$ is the identity matrix. We tackle (1.4) and (4.7) by applying the FISTA algorithm [5], where the accuracy is controlled at each iteration.

The data is generated as follows. First, we generate a sparse Gaussian random matrix $\mathbf{R}\sim\mathcal{N}(0,1)$ of the size $n\times k$, where $k = \lfloor 0.25n\rfloor$ is the rank of $\mathbf{R}$, and the

sparsity is 25%. Then, we generate matrix $\mathbf{M} := \mathbf{R}^T\mathbf{R} + 10^{-4}\mathbf{E}$, where $\mathbf{E} \sim \mathcal{N}(0, \mathbb{I})$. The lower bound $\mathbf{L}$ and the upper bound $\mathbf{U}$ are given as $\mathbf{L} := (m_l - 0.1\,|m_l|)\mathbb{I}$ and $\mathbf{U} := (m_u + 0.1\,|m_u|)\mathbb{I}$, where $m_l := \min_{i,j} \mathbf{M}_{ij}$ and $m_u := \max_{i,j} \mathbf{M}_{ij}$.

We test three algorithms on five problems of size $n \in \{80, 100, \ldots, 160\}$ w.r.t. $\rho = 0.2$. Table 6.1 reports the results and the performance of these three algorithms. Our platform is MATLAB 2011b on a PC Intel Xeon X5690 at 3.47GHz per core with 94Gb RAM.

TABLE 6.1
Comparison of Algorithm 1, `SDPT3` and `SeDuMi`

	Solver\n	80	100	120	140	160
Size	$[n_v; n_c]$	[16,200; 9,720]	[25,250; 15,150]	[36,300; 21,780]	[49,350; 29,610]	[64,400; 38,640]
Time (sec)	PFPN	15.738	24.046	24.817	25.326	36.531
	SDPT3	156.340	508.418	881.398	1742.502	2948.441
	SeDuMi	231.530	970.390	3820.828	9258.429	17096.580
$g(\mathbf{X}^*)$	PFPN	306.9159	497.6706	635.4304	842.4626	1096.6516
	SDPT3	306.9153	497.6754	635.4306	842.4644	1096.6540
	SeDuMi	306.9176	497.6821	635.4384	842.4776	1096.6695
[rank, sparsity]	PFPN	[20, 30.53%]	[26, 27.37%]	[30, 25.27%]	[35, 23.64%]	[40, 21.54%]
	SDPT3	[20, 41.02%]	[25, 36.99%]	[30, 51.61%]	[35, 45.03%]	[40, 49.07%]
	SeDuMi	[20, 45.23%]	[25, 64.20%]	[30, 54.83%]	[35, 60.87%]	[40, 59.24%]

From Table 6.1 we can see that if we reformulate problem (6.10) into a standard SDP problem where `SDPT3` and `SeDuMi` can solved, then the number of variables n_v and the number of constraints n_c increase rapidly (highlighted with red color). Consequently, the computational time in `SDPT3` and `SeDuMi` also increase significantly compared to Algorithm 1. Moreover, `SeDuMi` is much slower than `SDPT3` in this particular example. Since Algorithm 1 does not require to transform problem (6.10) into a standard SDP problem, we can clearly see the computational advantage of this algorithm to standard interior-point solvers, e.g., `SDPT3` and `SeDuMi`, for solving problem (6.10). We note that the implementation of the proposed scheme is still a prototype, coded in MATLAB without any preconditioning strategy.

6.4. Max-norm and ℓ_1-norm optimization in clustering. In this example, we show an application of Algorithm 1 to solve a constrained SDP problem arising from the correlation clustering [3], where the number of clusters is unknown. Briefly, the problem statement is as follows: Given a graph with p vertices, let $\mathbf{A}$ be its affinity matrix (cf., [3] for the definition). The clustering goal here is to partition the set of vertices such that the total disagreement with the edge labels is minimized in $\mathbf{A}$, which is an explicitly combinatorial problem. The work in [19] proposes a tight convex relaxation (1.3), poses significant difficulties to the IPM methods in large-scale. The approach is called max-norm constrained clustering, and if solved correctly, has rigorous theoretical guarantees of correctness for its solution.

In this example, we demonstrate that Algorithm 1 can obtain medium accuracy solutions in a scalable fashion as compared to a state-of-the-art IPM. Here, we use the adaptive update rule (4.5). The algorithm terminates if $t_k \leq 10^{-3}$ and $\lambda_k \leq 10^{-8}$. We also solve (1.4) and (4.7) by applying FISTA.

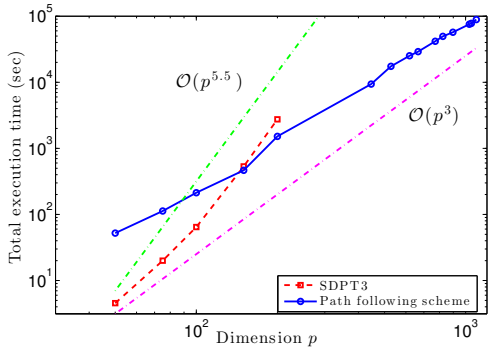
We compare our algorithm with the off-the-self, IPM implementation `SDPT3` [38], both in terms of time- and memory-complexity. Since the curse-of-dimensionality renders the execution of `SDPT3` impossible in large dimensions, we use the low precision

TABLE 6.2

Average values over 10 Monte Carlo iterations for each dimension p. The variable $\mathbf{K}^*$ refers to the respective solution at convergence as returned by the algorithms under comparison.

	p	50	75	100	150	200
Time (sec)	PF	62.450	109.426	202.600	416.044	1573.881
	SDPT3	4.396	21.282	64.939	522.021	2588.721
	[19]	102.217	236.366	354.444	778.904	1420.844
$g(\mathbf{K}^*)$	PF	549.1567	1293.6727	2232.5897	5396.0485	9809.6066
	SDPT3	549.1860	1293.7890	2233.0747	5396.7305	9809.6934
	[19]	597.8825	1387.1379	2496.6535	5583.8605	9958.0974

mode in `SDPT3` (i.e., $\varepsilon \approx 1.5 \times 10^{-8}$) in order to execute larger problems within a reasonable time frame. We compare these two schemes based on synthetic data, generated as described in [19].



	SDPT3		PF scheme
p	variables	constraints	variables
50	15.1	2.6	10
75	33.9	5.8	22.5
100	60.2	10.2	40
150	135.3	22.8	90
200	240.4	40.4	160

FIG. 6.1. ***(Left)*** *Execution times.* ***(Right)*** *Number of variables and equality constraints in thousands.*

In terms of solution accuracy, our scheme with the aforementioned parameter settings is comparable to the low-precision mode of `SDPT3`, and can often obtain accurate solutions (cf., Table 6.2). However, Figure 6.1(Left) illustrates that our path following scheme has a rather dramatic scaling advantage as compared to `SDPT3`: $\mathcal{O}(p^3)$ for ours vs. $\mathcal{O}(p^{5.5})$ for `SDPT3`. Because of this scaling, `SDPT3` cannot handle problems instances where $p > 200$ in our computer.

Reasons for our scalability are twofold. First, our path following scheme avoids "lifting" the problem into higher dimensions. Hence, as the problem dimensions grow (cf., Fig. 6.1(Right); numbers are in thousands), our memory requirement scales in a better fashion. Moreover, we do not have to handle additional (in)equality constraints. Second, the subproblem solver has linear convergence rate due to its construction (i.e., $\nabla^2 f \succ 0$). Hence, our fast solver (FISTA) obtains medium accuracy solutions quickly since the proximal operator is efficient and has a closed form.

We also compare the proposed scheme with the scalable Factorization Method (FM), presented in [19]: a state-of-the-art, non-convex implementation of (1.3), based on splitting techniques. The code is publicly available at `http://www.ali-jalali.com/`. We modified this code to include a stopping criterion at a tolerance of

$$\left\|\mathbf{K}^{k+1} - \mathbf{K}^k\right\|_F \leq 10^{-8} \max\left\{\left\|\mathbf{K}^k\right\|_F, 1\right\}.$$

In Table 6.2, we report the average results of 10 Monte-Carlo realizations for different p's. While the non-convex approach exhibits lower computational complexity empirically,[1] its solution quality suffers as compared to the convex solution, which has theoretical guarantees. It is clear that the non-convex approach is rather susceptible to local minima.

6.5. Sparse Pareto frontier in sparse graph learning. Many machine learning and signal processing problems naturally feature composite minimization problems where f is directly self-concordant, such as sparse regression with unknown noise variance [32], Poisson imaging [17], one-bit compressive sensing, and graph learning [28, 20]. Here, we consider the graph learning problem: Let $\mathbf{\Sigma}$ be the covariance matrix of a Gaussian Markov random field (GMRF) and let $\mathbf{X} = \mathbf{\Sigma}^{-1}$. To satisfy the conditional dependencies with respect to the GMRF, $\mathbf{X}$ must have zero in $\mathbf{X}_{ij}$ corresponding to the absence of an edge between node i and node j [11]. Hence, given the empirical covariance $\widehat{\mathbf{\Sigma}} \succeq 0$, which is possibly rank deficient, we would like to learn the underlying GMRF.

It turns out that we can still learn GMRF's with theoretical consistency guarantees from a number of data samples as few as $m = \mathcal{O}(d^2 \log p)$ [28], where d is the graph node degree, via

$$\min_{\mathbf{X}\in\mathbb{R}^{p\times p}:\ \mathbf{X}\succ 0} \left\{ -\log\det(\mathbf{X}) + \operatorname{tr}\left(\widehat{\mathbf{\Sigma}}\mathbf{X}\right) + \rho \left\|\operatorname{vec}(\mathbf{X})\right\|_1 \right\}, \tag{6.11}$$

where $\rho > 0$ is a regularization parameter. We easily observe that (6.11) satisfies the $\mathcal{P}(t)$ formulation for $t = 1/\rho$. Unfortunately, the theoretical results only indicate the existence of a regularization parameter for consistent estimates and we have to tune to obtain the best ρ^* in practice. We note that the function $f(\mathbf{X}) := -\log\det(\mathbf{X})$ is a self-concordant barrier of $\mathcal{S}_+^p$. As discussed in Subsection 6.1, we can modify the update rule for ρ_k, we can still apply Algorithm 1 to track the Pareto frontier of problem (6.11) for the case $f(\cdot) + \langle \mathbf{c}, \cdot \rangle$.

To the best of our knowledge, the selection of ρ^* with respect to a general-purpose objective, such as $\mathcal{P}(\rho)$, still remains widely open. For GMRF learning, a homotopy approach is proposed in [31, 21], where ρ is updated by a non-adaptive multiplicative factor such that $\rho_{k+1} = c\rho_k$ for $0 < c < 1$. This approach is usually time consuming in practice, and may skip solutions with sparsity close to the desired sparsity level. Traditionally, (6.11) is addressed by IPM's. Other than [37] exploited here, we do not know any scalable method that has rigorous global convergence guarantees for (6.11) as it has a globally non-Lipschitz continuous gradient. The authors in [1] use a probabilistic heuristic to select ρ: as the number of samples go to infinity, this heuristic leads to the maximum likelihood (unregularized) estimator. In practice though, the proposed ρ values are quite large and do not consistently lead to good solutions.

To this end, our scheme provides an adaptive strategy on how to update the regularization parameter. For instance, we can pick a range $\rho \in [\rho_{\min}, \rho_0]$ and apply our path-following scheme, starting from ρ_0 until we either achieve the desired solution sparsity or we reach the lower bound $\rho_{\min}$. To illustrate the approach, we choose two real data examples from `http://ima.umn.edu/~maxxa007/send_SICS/`: `Lymph` and `Leukemia`, where the GMRF sizes are $p = 587$ and 1255, respectively. Figure 6.2 shows the solution sparsity vs. the penalty parameter curve (not to be confused with f vs. g curve, which is a convex Pareto curve) as obtained in a tuning-free fashion by our scheme.

[1] Theoretically, FM's computational cost is proportional to the cost of $p \times p$ matrix multiplications.

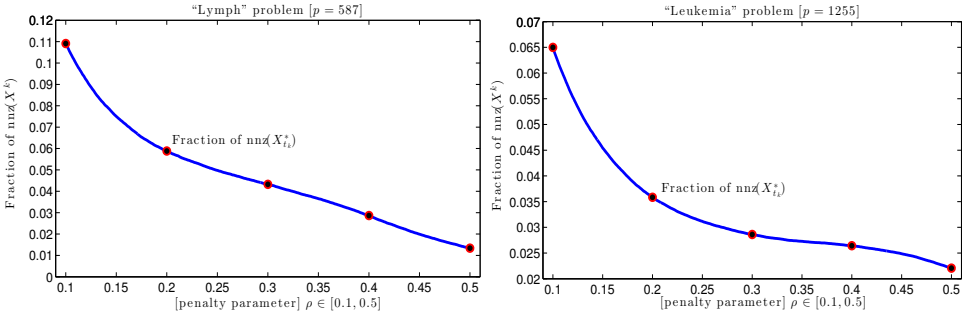


FIG. 6.2. *Impact of the regularization parameter to the solution sparsity.*

TABLE 6.3
The relative error and the number of nonzero elements of two approximate solutions

ρ	0.1	0.2	0.3	0.4	0.5
Lymph ($n = 587$)					
Relative error e_k	0.0011	0.0013	0.0018	0.0018	7.5342×10^{-6}
`n.n.z.` ($\widetilde{\mathbf{X}}(\rho_k)/\mathbf{X}^k$)	37587/37561	20275/20269	14901/14875	9869/9871	4615/4615
Leukemia ($n = 1255$)					
Relative error e_k	6.1643×10^{-4}	5.5701×10^{-4}	6.2124×10^{-4}	5.6060×10^{-4}	3.6497×10^{-6}
`n.n.z.` ($\widetilde{\mathbf{X}}(\rho_k)/\mathbf{X}^k$)	102313/102253	56451/56421	45051/45055	41613/41609	34761/34761

In order to verify the obtained Pareto curve $\{\mathbf{X}^k\}$ well approximates the true solution trajectory $\mathbf{X}^*(\rho)$ of the problem (6.11), we apply the proximal-Newton algorithm in [36] to compute the approximate solution $\widetilde{\mathbf{X}}(\rho_k)$ to $\mathbf{X}^*(\rho_k)$ at five different points of ρ. The relative errors $e_k := \left\|\mathbf{X}^k - \widetilde{\mathbf{X}}(\rho_k)\right\|_F / \max\left\{\widetilde{\mathbf{X}}(\rho_k)\right\}$ as well as the number of nonzero elements `n.n.z.` are shown in Table 6.3. We can see from this table that both solutions are relatively close to each other both in terms of relative error and the sparsity.

7. Concluding remarks. We have proposed a new inexact path-following framework for minimizing (possibly) non-smooth and non-Lipschitz gradient objectives under constraints that admit a self-concordant barrier. We have shown how to solve such problems scalably without inflating problem dimensions or introducing additional slack variables and constraints. Our method is quite modular: custom implementations only require the corresponding custom solver for the composite subproblem (1.4) with a strongly convex quadratic smooth term and a tractable proximity of the second term g. We have provided a rigorous analysis that establish the worse complexity of our approach via a new joint treatment of proximal methods and self-concordant optimization schemes. While our scheme maintains the original problem structure, its worst-case complexity remains the same as in standard path-following interior point methods [23]. We have also shown how the new scheme can obtain points on the Pareto frontier of regularized problems (with globally non-Lipschitz gradient of the smooth part). We have numerically illustrated our method on three examples involving the nonsmooth constrained convex programming problems of matrix variables. Numerical results have shown that the new path-following scheme is superior

to some off-the-self solvers that require to transform the problem into standard conic programs.

8. Acknowledgment. This work was supported in part by the European Commission under Grant MIRG-268398, ERC Future Proof and SNF 200021-132548, SNF 200021-146750 and SNF CRSII2-147633.

Appendix A. Technical proofs. We provide in this appendix the full proofs of two theorems: Theorem 3.3 and Theorem 4.1, and three technical lemmas: Lemma 4.2, Lemma 4.3 and Lemma 5.1.

A.1. The proof of Theorem 3.3. We define the restricted approximate gap between $\nabla^2 f(\mathbf{x}_t^*)$ and $\nabla^2 f(\mathbf{x}^k)$ along the direction $\bar{\mathbf{x}}^{k+1} - \mathbf{x}^k$ as $\bar{\mathbf{r}}^k := (\nabla^2 f(\mathbf{x}_t^*) - \nabla^2 f(\mathbf{x}^k))(\bar{\mathbf{x}}^{k+1} - \mathbf{x}^k)$. Then, by using the definition (2.3) of $P_{\mathbf{x}}^g$ and (2.4) of $S_{\mathbf{x}}$, we can write (3.4) equivalently to

$$\bar{\mathbf{x}}^{k+1} = P_{\mathbf{x}_t^*}^g \left(S_{\mathbf{x}_t^*}(\mathbf{x}^k) + \bar{\mathbf{r}}^k \right). \tag{A.1}$$

Now, we can estimate $\bar{\lambda}_{k+1} := \left\|\bar{\mathbf{x}}^{k+1} - \mathbf{x}_t^*\right\|_{\mathbf{x}_t^*}$ as follows

$$\begin{aligned}
\bar{\lambda}_{k+1} &:= \left\|\bar{\mathbf{x}}^{k+1} - \mathbf{x}_t^*\right\|_{\mathbf{x}_t^*} \\
&\overset{(A.1)+(2.7)}{=} \left\| P_{\mathbf{x}_t^*}^g \left(S_{\mathbf{x}_t^*}(\mathbf{x}^k) + \bar{\mathbf{r}}^k \right) - P_{\mathbf{x}_t^*}^g \left(S_{\mathbf{x}_t^*}(\mathbf{x}_t^*) \right) \right\|_{\mathbf{x}_t^*} \\
&\overset{(2.6)}{\leq} \left\| S_{\mathbf{x}_t^*}(\mathbf{x}^k) - S_{\mathbf{x}_t^*}(\mathbf{x}_t^*) + \bar{\mathbf{r}}^k \right\|_{\mathbf{x}_t^*}^* \\
&\leq \left\| S_{\mathbf{x}_t^*}(\mathbf{x}^k) - S_{\mathbf{x}_t^*}(\mathbf{x}_t^*) \right\|_{\mathbf{x}_t^*}^* + \left\|\bar{\mathbf{r}}^k\right\|_{\mathbf{x}_t^*}^*. \qquad \text{(A.2)}
\end{aligned}$$

Similarly to the proof of [36, Theorem 5], we show that

$$\left\| S_{\mathbf{x}_t^*}(\mathbf{x}^k) - S_{\mathbf{x}_t^*}(\mathbf{x}_t^*) \right\|_{\mathbf{x}_t^*}^* \leq \frac{\lambda_k^2}{1 - \lambda_k}, \tag{A.3}$$

provided that $\lambda_k < 1$.

Next, we estimate $\left\|\bar{\mathbf{r}}^k\right\|_{\mathbf{x}_t^*}^*$. We have

$$\begin{aligned}
\left\|\bar{\mathbf{r}}^k\right\|_{\mathbf{x}_t^*}^* &= \left\| \left(\nabla^2 f(\mathbf{x}_t^*) - \nabla^2 f(\mathbf{x}^k)\right)(\bar{\mathbf{x}}^{k+1} - \mathbf{x}^k) \right\|_{\mathbf{x}_t^*}^* \\
&\leq \left\| \nabla^2 f(\mathbf{x}_t^*)^{-1/2} \left(\nabla^2 f(\mathbf{x}_t^*) - \nabla^2 f(\mathbf{x}^k)\right) \nabla^2 f(\mathbf{x}_t^*)^{-1/2} \right\|_{2\to 2} \left\|\bar{\mathbf{x}}^{k+1} - \mathbf{x}^k\right\|_{\mathbf{x}_t^*} \\
&= \left\| \mathbb{I} - \nabla^2 f(\mathbf{x}_t^*)^{-1/2} \nabla^2 f(\mathbf{x}^k) \nabla^2 f(\mathbf{x}_t^*)^{-1/2} \right\|_{2\to 2} \left\|\bar{\mathbf{x}}^{k+1} - \mathbf{x}^k\right\|_{\mathbf{x}_t^*}. \qquad \text{(A.4)}
\end{aligned}$$

By applying [23, Theorem 4.1.6], we can show that

$$\begin{aligned}
\left\| \mathbb{I} - \nabla^2 f(\mathbf{x}_t^*)^{-1/2} \nabla^2 f(\mathbf{x}^k) \nabla^2 f(\mathbf{x}_t^*)^{-1/2} \right\|_{2\to 2} &\leq \max\left\{1 - (1-\lambda_k)^2, (1-\lambda_k)^{-2} - 1\right\} \\
&= \frac{2\lambda_k - \lambda_k^2}{(1-\lambda_k)^2}.
\end{aligned}$$

Substituting this estimate into (A.4) and then using the triangle inequality, we obtain

$$\left\|\bar{\mathbf{r}}^k\right\|_{\mathbf{x}_t^*} \leq \left(\frac{2\lambda_k - \lambda_k^2}{(1-\lambda_k)^2}\right) \left\|\bar{\mathbf{x}}^{k+1} - \mathbf{x}^k\right\|_{\mathbf{x}_t^*} \leq \left(\frac{2\lambda_k - \lambda_k^2}{(1-\lambda_k)^2}\right) \left(\bar{\lambda}_{k+1} + \lambda_k\right), \tag{A.5}$$

provided that $\lambda_k < 1$.

Substituting (A.3) and (A.5) into (A.2) and then rearranging the result, we deduce

$$\bar{\lambda}_{k+1} \leq \left(\frac{3-2\lambda_k}{1-4\lambda_k+2\lambda_k^2}\right)\lambda_k^2, \tag{A.6}$$

provided that $1-4\lambda_k+2\lambda_k^2 > 0$. We can easily show that the condition $1-4\lambda_k+2\lambda_k^2 > 0$ holds if $\lambda_k \in [0, 1-\frac{\sqrt{2}}{2})$.

Note that $0 \preceq \nabla^2 f(\mathbf{x}_t^*) \preceq (1-\lambda_k)^{-2}\nabla^2 f(\mathbf{x}^k)$ due to [23, Theorem 4.1.6]. For any $\mathbf{u}$, we have $\|\mathbf{u}\|_{\mathbf{x}_t^*} \leq (1-\lambda_k)^{-1}\|\mathbf{u}\|_{\mathbf{x}^k}$. By using this inequality, (3.7) and the triangle inequality, it is easy to show that

$$\begin{aligned}\lambda_{k+1} = \left\|\mathbf{x}^{k+1}-\mathbf{x}_t^*\right\|_{\mathbf{x}_t^*} &\leq \left\|\mathbf{x}^{k+1}-\bar{\mathbf{x}}^{k+1}\right\|_{\mathbf{x}_t^*} + \left\|\bar{\mathbf{x}}^{k+1}-\mathbf{x}_t^*\right\|_{\mathbf{x}_t^*}\\ &\overset{(3.7)}{\leq} (1-\lambda_k)^{-1}\delta + \bar{\lambda}_{k+1}.\end{aligned}$$

By substituting (A.6) into this inequality, we obtain

$$\lambda_{k+1} \leq \frac{\delta}{1-\lambda_k} + \left(\frac{3-2\lambda_k}{1-4\lambda_k+2\lambda_k^2}\right)\lambda_k^2. \tag{A.7}$$

Since $\lambda_k \in [0, 1-\frac{\sqrt{2}}{2})$, the right-hand side of (A.7) is well-defined. Moreover, it is obvious to check that the right-hand side of (A.7) is increasing w.r.t. $\delta \geq 0$ and $\lambda_k \in [0, 1-\frac{\sqrt{2}}{2})$. □

A.2. The proof of Theorem 4.1. We define the function $\psi(\lambda) := \frac{3-2\lambda}{1-4\lambda+2\lambda^2}$. It is easy to check that ψ is increasing in $[0, 1-\sqrt{2}/2)$. Let us limit the range of $\lambda \in [0, 0.15]$. Then, one can show that $\max\{\psi(\lambda) \mid \lambda \in [0, 0.15]\} \leq 6.5$. Hence, we can upper estimate (4.1) as

$$\lambda_{k+1} \leq 1.18\delta + 6.07\tilde{\lambda}_k^2. \tag{A.8}$$

Now, we recall the following estimate from [35, Lemma A.1.(c)] as

$$\tilde{\lambda}_k \leq \frac{\lambda_k + \Delta_k}{1-\Delta_k}, \tag{A.9}$$

provided that $\Delta_k < 1$.

Let us fix some $\beta \in (0, 1.5]$. By the assumption $\lambda_k \leq \beta$, it follows from (A.9) that

$$\tilde{\lambda}_k \leq \frac{\lambda_k + \Delta_k}{1-\Delta_k} \leq \frac{\beta + \Delta_k}{1-\Delta_k}. \tag{A.10}$$

Substituting (A.10) into (A.8) we obtain

$$\lambda_{k+1} \leq 1.18\delta + 6.07\left(\frac{\beta+\Delta_k}{1-\Delta_k}\right)^2. \tag{A.11}$$

Since we desire $\lambda_{k+1} \leq \beta$, by using (A.11), we require $\left(\frac{\beta+\Delta_k}{1-\Delta_k}\right)^2 \leq \frac{\beta-1.18\delta}{6.07}$ provided that $\delta < \beta/1.18$. Since $\delta \leq 0.075\beta$, the last condition leads to

$$0 \leq \Delta_k \leq \frac{\sqrt{\beta} - 2.581\beta}{2.581 + \sqrt{\beta}} < 1, \tag{A.12}$$

for any $\beta \in (0, 0.15]$. Finally, we can easy check that $\tilde{\lambda}_k \leq \frac{1}{2.581}\sqrt{\beta}$ due to (A.8). □

A.3. The proof of Lemma 4.2. Since $\mathbf{x}^*_{t_k}$ and $\mathbf{x}^*_{t_{k+1}}$ are the solutions of (1.2) at $t = t_k$ and $t = t_{k+1}$, respectively, they satisfy the following optimality conditions:

$$\begin{aligned}\mathbf{0} &\in t_k\nabla f(\mathbf{x}^*_{t_k}) + \partial g(\mathbf{x}^*_{t_k}),\\ \mathbf{0} &\in t_{k+1}\nabla f(\mathbf{x}^*_{t_{k+1}}) + \partial g(\mathbf{x}^*_{t_{k+1}}).\end{aligned}$$

Hence, there exist $\mathbf{v}_k \in \partial g(\mathbf{x}^*_{t_k})$ and $\mathbf{v}_{k+1} \in \partial g(\mathbf{x}^*_{t_{k+1}})$ such that $\mathbf{v}_k = -t_k\nabla f(\mathbf{x}^*_{t_k})$ and $\mathbf{v}_{k+1} = -t_{k+1}\nabla f(\mathbf{x}^*_{t_{k+1}})$. Then, we have

$$\begin{aligned}\mathbf{v}_{k+1} - \mathbf{v}_k &= t_k\nabla f(\mathbf{x}^*_{t_k}) - t_{k+1}\nabla f(\mathbf{x}^*_{t_{k+1}})\\ &\overset{(4.3)}{=} t_k\left(\nabla f(\mathbf{x}^*_{t_k}) - \nabla f(\mathbf{x}^*_{t_{k+1}})\right) - d_k\nabla f(\mathbf{x}^*_{t_{k+1}}).\end{aligned}$$

By using the convexity of g, the last expression implies

$$\begin{aligned}0 &\le (\mathbf{v}_{k+1} - \mathbf{v}_k)^T(\mathbf{x}^*_{t_{k+1}} - \mathbf{x}^*_{t_k})\\ &= t_k\left(\nabla f(\mathbf{x}^*_{t_k}) - \nabla f(\mathbf{x}^*_{t_{k+1}})\right)^T(\mathbf{x}^*_{t_{k+1}} - \mathbf{x}^*_{t_k}) - d_k\nabla f(\mathbf{x}^*_{t_{k+1}})^T(\mathbf{x}^*_{t_{k+1}} - \mathbf{x}^*_{t_k})\\ &\le t_k\left(\nabla f(\mathbf{x}^*_{t_k}) - \nabla f(\mathbf{x}^*_{t_{k+1}})\right)^T(\mathbf{x}^*_{t_{k+1}} - \mathbf{x}^*_{t_k}) + |d_k|\,\|\nabla f(\mathbf{x}^*_{t_{k+1}})\|^*_{\mathbf{x}^*_{t_{k+1}}}\|\mathbf{x}^*_{t_{k+1}} - \mathbf{x}^*_{t_k}\|_{\mathbf{x}^*_{t_{k+1}}},\end{aligned}$$

where the last inequality is due to the generalized Cauchy-Schwatz inequality. Since $t_k > 0$, we can deduce from the last inequality as

$$\left(\nabla f(\mathbf{x}^*_{t_{k+1}}) - \nabla f(\mathbf{x}^*_{t_k})\right)^T(\mathbf{x}^*_{t_{k+1}} - \mathbf{x}^*_{t_k}) \le \frac{|d_k|}{t_k}\|\nabla f(\mathbf{x}^*_{t_{k+1}})\|^*_{\mathbf{x}^*_{t_{k+1}}}\|\mathbf{x}^*_{t_{k+1}} - \mathbf{x}^*_{t_k}\|_{\mathbf{x}^*_{t_{k+1}}}. \tag{A.13}$$

However, since f is standard self-concordant, by applying [23, Theorem 4.1.7], we have

$$\left(\nabla f(\mathbf{x}^*_{t_{k+1}}) - \nabla f(\mathbf{x}^*_{t_k})\right)^T(\mathbf{x}^*_{t_{k+1}} - \mathbf{x}^*_{t_k}) \ge \frac{\|\mathbf{x}^*_{t_{k+1}} - \mathbf{x}^*_{t_k}\|^2_{\mathbf{x}^*_{t_{k+1}}}}{1 + \|\mathbf{x}^*_{t_{k+1}} - \mathbf{x}^*_{t_k}\|_{\mathbf{x}^*_{t_{k+1}}}}.$$

Using this inequality together with (A.13) we obtain

$$\frac{\|\mathbf{x}^*_{t_{k+1}} - \mathbf{x}^*_{t_k}\|_{\mathbf{x}^*_{t_{k+1}}}}{1 + \|\mathbf{x}^*_{t_{k+1}} - \mathbf{x}^*_{t_k}\|_{\mathbf{x}^*_{t_{k+1}}}} \le \frac{|d_k|}{t_k}\|\nabla f(\mathbf{x}^*_{t_{k+1}})\|^*_{\mathbf{x}^*_{t_{k+1}}} \overset{(2.1)}{\le} \frac{|d_k|}{t_k}\sqrt{\nu}.$$

where by the defintion of Δ_k, this completes the proof of (4.4). The last statement in Lemma 4.2 is a direct consequence of (4.4). □

A.4. The proof of Lemma 4.3. Let $g_0(\cdot) := t_0^{-1}g(\cdot)$. Similar to the proof of [37, Lemma 3.3], we can estimate

$$F(\mathbf{x}^{j+1}; t_0) - F(\mathbf{x}^j; t_0) \le -\alpha_j\nabla f(\mathbf{x}^j)^T\mathbf{d}^j + \omega_*(\alpha_j\zeta_j) + \alpha_j\left(g_0(\mathbf{s}^j) - g_0(\mathbf{x}^j)\right), \tag{A.14}$$

where $\alpha_j\zeta_j < 1$ and $\omega_*(\tau) := -\tau - \ln(1-\tau)$. From the definition (3.2) of $F^j_{t_0}$ and (4.8) we have

$$\begin{aligned}g_0(\mathbf{s}^j) - g_0(\mathbf{x}^j) &\le g_0(\bar{\mathbf{s}}^j) - g_0(\mathbf{x}^j) + \frac{\delta^2}{2} + \nabla f(\mathbf{x}^j)^T(\bar{\mathbf{s}}^j - \mathbf{s}^j)\\ &\quad + \frac{1}{2}\left(\left\|\bar{\mathbf{s}}^j - \mathbf{x}^j\right\|^2_{\mathbf{x}^j} - \left\|\mathbf{s}^j - \mathbf{x}^j\right\|^2_{\mathbf{x}^j}\right).\end{aligned} \tag{A.15}$$

Since $\bar{\mathbf{s}}^j$ is the exact solution of (4.7), using the optimality condition (3.4) of this problem, we have

$$\bar{\mathbf{v}}^j = -\nabla f(\mathbf{x}^j) - \nabla^2 f(\mathbf{x}^j)(\bar{\mathbf{s}}^j - \mathbf{x}^j),\ \bar{\mathbf{v}}^j \in \partial g_0(\bar{\mathbf{s}}^j). \tag{A.16}$$

By the convexity of g_0, (A.16) implies

$$g_0(\bar{\mathbf{s}}^j) - g_0(\mathbf{x}^j) \le -\nabla f(\mathbf{x}^j)^T(\bar{\mathbf{s}}^j - \mathbf{x}^j) - \left\|\bar{\mathbf{s}}^j - \mathbf{x}^j\right\|^2_{\mathbf{x}^j}.$$

Substituting this inequality into (A.15) and rearranging the result by using $\zeta_j = \left\|\mathbf{d}^j\right\|_{\mathbf{x}^j} = \left\|\mathbf{s}^j - \mathbf{x}^j\right\|_{\mathbf{x}^j}$, we obtain

$$g_0(\mathbf{s}^j) - g_0(\mathbf{x}^j) \le \frac{\delta^2}{2} - \nabla f(\mathbf{x}^j)^T(\mathbf{s}^j - \mathbf{x}^j) - \frac{1}{2}\left(\left\|\bar{\mathbf{s}}^j - \mathbf{x}^j\right\|^2_{\mathbf{x}^j} + \zeta_j^2\right). \tag{A.17}$$

By using the triangle inequality and (3.6) we deduce

$$\left\|\bar{\mathbf{s}}^j - \mathbf{x}^j\right\|_{\mathbf{x}^j} \ge \left\|\mathbf{s}^j - \mathbf{x}^j\right\|_{\mathbf{x}^j} - \left\|\mathbf{s}^j - \bar{\mathbf{s}}^j\right\|_{\mathbf{x}^j} \ge \zeta_j - \delta.$$

Hence, with $\delta \le \zeta_j$, this inequality implies

$$\left\|\bar{\mathbf{s}}^j - \mathbf{x}^j\right\|^2_{\mathbf{x}^j} \ge \zeta_j^2 + \delta^2 - 2\zeta_j\delta. \tag{A.18}$$

Combining (A.14), (A.17) and (A.18), we finally get

$$F(\mathbf{x}^{j+1}; t_0) - F(\mathbf{x}^j; t_0) \le \omega_*(\alpha_j\zeta_j) - \zeta_j(\zeta_j - \delta)\alpha_j, \tag{A.19}$$

provided that $\alpha_j\zeta_j < 1$ and $\delta \le \zeta_j$.

Now we consider the function $\varphi(\alpha) := \zeta_j(\zeta_j - \delta)\alpha - \omega_*(\zeta_j\alpha)$. This function is concave, it attains the maximum at $\alpha_j := \frac{\zeta_j - \delta}{\zeta_j(1+\zeta_j-\delta)}$ provided that $\delta \le \zeta_j$. In this case, we also have $\alpha_j\zeta_j = \frac{\zeta_j-\delta}{1+\zeta_j-\delta} < 1$ and $\varphi(\alpha_j) = \omega(\zeta_j - \delta)$. Substituting this value into (A.19) and then subtracting the result to $F(\mathbf{x}^*_{t_0}; t_0)$ we obtain (4.9). □

A.5. The proof of Lemma 5.1. Since f is the barrier function of Ω and $\mathbf{x}^*_t$ is the solution of (1.2), it is obvious that $\mathbf{x}^*_t \in \mathrm{int}(\Omega)$ and $g(\mathbf{x}^*) \le g(\mathbf{x}^*_t)$. We first prove (5.1). From [23, Theorem 4.2.4] we have

$$\nabla f(\mathbf{x})^T(\mathbf{y} - \mathbf{x}) < \nu, \quad \forall \mathbf{x}, \mathbf{y} \in \mathrm{dom}\,(f). \tag{A.20}$$

By using the convexity of g, the optimality condition (2.2) and the property (A.20) of the barrier function f, for any $\mathbf{x} \in \mathrm{dom}\,(F) \equiv \mathrm{dom}\,(f) \cap \mathrm{dom}\,(g)$, we have

$$\begin{aligned} g(\mathbf{x}) - g(\mathbf{x}^*_t) &\ge (\xi^*_t)^T(\mathbf{x} - \mathbf{x}^*_t),\ \forall (\xi^*_t) \in \partial g(\mathbf{x}^*_t) \\ &\overset{(2.2)}{\ge} -t\nabla f(\mathbf{x}^*_t)^T(\mathbf{x} - \mathbf{x}^*_t) \\ &\overset{(A.20)}{\ge} -t\nu. \end{aligned} \tag{A.21}$$

By substituting $\mathbf{x} = \mathbf{x}^*$ in (A.21) we obtain (5.1). Similarly, by letting $t = t_k$ and $\mathbf{x} = \mathbf{x}^k$ in (A.21) we obtain the right-hand side of (5.2).

Next, we prove the left-hand side of (5.2). By using (3.5) in Definition 3.1 we can estimate

$$\begin{aligned} g(\mathbf{x}^{k+1}) &\le g(\bar{\mathbf{x}}^{k+1}) + t_{k+1}\left[Q(\bar{\mathbf{x}}^{k+1};\mathbf{x}^k) - Q(\mathbf{x}^{k+1};\mathbf{x}^k)\right] + t_{k+1}\frac{\delta^2}{2} \\ &\le g(\bar{\mathbf{x}}^{k+1}) + t_{k+1}\nabla f(\mathbf{x}^k)^T(\bar{\mathbf{x}}^{k+1} - \mathbf{x}^{k+1}) + t_{k+1}\frac{\delta^2}{2} \\ &\quad + \frac{t_{k+1}}{2}\left[\left\|\bar{\mathbf{x}}^{k+1} - \mathbf{x}^k\right\|^2_{\mathbf{x}^k} - \left\|\mathbf{x}^{k+1} - \mathbf{x}^k\right\|^2_{\mathbf{x}^k}\right], \end{aligned} \tag{A.22}$$

where $\bar{\mathbf{x}}^{k+1}$ is the exact solution of (3.3) at $t = t_{k+1}$. Moreover, from the optimality condition (3.4), there exists $\bar{\mathbf{v}}_{k+1} \in \partial g(\bar{\mathbf{x}}^{k+1})$ such that

$$\bar{\mathbf{v}}_{k+1} = -t_{k+1}\nabla f(\mathbf{x}^k) - t_k\nabla^2 f(\mathbf{x}^k)(\bar{\mathbf{x}}^{k+1} - \mathbf{x}^k). \tag{A.23}$$

By using the convexity of g we can estimate $g(\mathbf{x}^*_{t_k}) - g(\mathbf{x}^{k+1})$ as

$$\begin{aligned} g(\mathbf{x}^*_{t_{k+1}}) - g(\bar{\mathbf{x}}^{k+1}) &\ge \bar{\mathbf{v}}^T_{k+1}(\mathbf{x}^*_{t_{k+1}} - \bar{\mathbf{x}}^{k+1}) \\ &\overset{(A.23)}{=} -t_{k+1}\nabla f(\mathbf{x}^k)^T(\mathbf{x}^*_{t_{k+1}} - \bar{\mathbf{x}}^{k+1}) \\ &\quad - t_{k+1}(\bar{\mathbf{x}}^{k+1} - \mathbf{x}^k)^T\nabla^2 f(\mathbf{x}^k)(\mathbf{x}^*_{t_{k+1}} - \bar{\mathbf{x}}^{k+1}). \end{aligned} \tag{A.24}$$

Now we sum up (A.22) and (A.24) and then rearrange the result by using the Cauchy-Schwarz inequality to get

$$\begin{aligned} g(\mathbf{x}^*_{t_{k+1}}) - g(\mathbf{x}^{k+1}) &\ge -t_{k+1}\nabla f(\mathbf{x}^k)^T(\mathbf{x}^*_{t_{k+1}} - \mathbf{x}^{k+1}) - \frac{t_{k+1}}{2}\delta^2 \\ &\quad - \frac{t_{k+1}}{2}\Big[\|\bar{\mathbf{x}}^{k+1} - \mathbf{x}^k\|^2_{\mathbf{x}^k} - \|\mathbf{x}^{k+1} - \mathbf{x}^k\|^2_{\mathbf{x}^k} \\ &\quad + 2(\bar{\mathbf{x}}^{k+1} - \mathbf{x}^k)^T\nabla^2 f(\mathbf{x}^k)(\mathbf{x}^*_{t_{k+1}} - \bar{\mathbf{x}}^{k+1})\Big]_{[1]}. \end{aligned} \tag{A.25}$$

From [23, Theorem 4.1.6] we have

$$(1-\tilde{\lambda}_k)^2\nabla^2 f(\mathbf{x}^*_{t_{k+1}}) \preceq \nabla^2 f(\mathbf{x}^k) \preceq (1-\tilde{\lambda}_k)^{-2}\nabla^2 f(\mathbf{x}^*_{t_{k+1}}), \tag{A.26}$$

where $\tilde{\lambda}_k := \|\mathbf{x}^k - \mathbf{x}^*_{t_{k+1}}\|_{\mathbf{x}^*_{t_{k+1}}}$ defined as before. We can easily show that

$$\left\|\nabla f(\mathbf{x}^k)\right\|^*_{\mathbf{x}^*_{t_{k+1}}} \le (1-\tilde{\lambda}_k)^{-1}\left\|\nabla f(\mathbf{x}^k)\right\|^*_{\mathbf{x}^k} \overset{(2.1)}{\le} (1-\tilde{\lambda}_k)^{-1}\sqrt{\nu}.$$

Using this inequality together with the Cauchy-Shwarz inequality, we can prove that

$$\begin{aligned} \nabla f(\mathbf{x}^k)^T(\mathbf{x}^*_{t_{k+1}} - \mathbf{x}^{k+1}) &\le \left\|\nabla f(\mathbf{x}^k)\right\|^*_{\mathbf{x}^*_{t_{k+1}}}\left\|\mathbf{x}^{k+1} - \mathbf{x}^*_{t_{k+1}}\right\| \\ &= \sqrt{\nu}(1-\tilde{\lambda}_k)^{-1}\lambda_{k+1}. \end{aligned} \tag{A.27}$$

Next, we estimate the last term $[\cdots]_{[1]}$ of (A.25) as follows

$$\begin{aligned} [\cdots]_{[1]} &:= \|\bar{\mathbf{x}}^{k+1} - \mathbf{x}^k\|^2_{\mathbf{x}^k} - \|\mathbf{x}^{k+1} - \mathbf{x}^k\|^2_{\mathbf{x}^k} + 2(\bar{\mathbf{x}}^{k+1} - \mathbf{x}^k)^T\nabla^2 f(\mathbf{x}^k)(\mathbf{x}^*_{t_{k+1}} - \bar{\mathbf{x}}^{k+1}) \\ &= -\left\|\mathbf{x}^{k+1} - \mathbf{x}^k\right\|^2_{\mathbf{x}^k} - \left\|\bar{\mathbf{x}}^{k+1} - \mathbf{x}^k\right\|^2_{\mathbf{x}^k} + 2(\bar{\mathbf{x}}^{k+1} - \mathbf{x}^k)^T\nabla^2 f(\mathbf{x}^k)(\mathbf{x}^*_{t_{k+1}} - \mathbf{x}^k) \\ &\le -\frac{1}{2}\left\|\bar{\mathbf{x}}^{k+1} - \mathbf{x}^{k+1}\right\|^2_{\mathbf{x}^k} + 2(\bar{\mathbf{x}}^{k+1} - \mathbf{x}^k)^T\nabla^2 f(\mathbf{x}^k)(\mathbf{x}^*_{t_{k+1}} - \mathbf{x}^k) \\ &\overset{(A.26)}{\le} 2(1-\tilde{\lambda}_k)^{-2}\left(\left\|\bar{\mathbf{x}}^{k+1} - \mathbf{x}^{k+1}\right\|_{\mathbf{x}^k} + \lambda_{k+1} + \tilde{\lambda}_k\right)\tilde{\lambda}_k \\ &\le 2(1-\tilde{\lambda}_k)^{-2}\left(\delta + \lambda_{k+1} + \tilde{\lambda}_k\right)\tilde{\lambda}_k. \end{aligned} \tag{A.28}$$

Here, the two last inequalities are obtained by using the triangle inequality, the definition of λ_{k+1}, $\tilde{\lambda}_k$ and (3.7). Now, we combine (A.24), (A.25) and (A.28) to derive

$$g(\mathbf{x}^*_{t_{k+1}}) - g(\mathbf{x}^{k+1}) \geq -t_{k+1}\left[\sqrt{\nu}\frac{\lambda_{k+1}}{1-\tilde{\lambda}_k} + (1-\tilde{\lambda}_k)^{-2}\tilde{\lambda}_k\left(\lambda_{k+1} + \tilde{\lambda}_k + \delta\right) + \frac{\delta^2}{2}\right],$$

which is the left-hand side of (5.2) provided that $\tilde{\lambda}_k < 1$. Finally, the estimate (5.3) follows directly by summing up (5.1) and (5.2). □

REFERENCES

[1] Banerjee, O., El Ghaoui, L., and d'Aspremont, A. Model selection through sparse maximum likelihood estimation for multivariate gaussian or binary data. *Journal of Machine Learning Research*, 9, (2008), 485–516.

[2] Bank, B., Guddat, J., Klatte, D., Kummer, B., and Tammer, K. *Non-Linear Parametric Optimization*. Birkhauser Verlag, 1983.

[3] Bansal, N., Blum, A., and Chawla, S. Correlation Clustering. *Machine Learning*, 56, (2004), 89–113.

[4] Bauschke, H., and Combettes, P. *Convex analysis and monotone operators theory in Hilbert spaces*. Springer-Verlag, 2011.

[5] Beck, A., and Teboulle, M. A Fast Iterative Shrinkage-Thresholding Algorithm for Linear Inverse Problems. *SIAM J. Imaging Sciences*, 2(1), (2009), 183–202.

[6] Becker, S., Candès, E., and Grant, M. Templates for convex cone problems with applications to sparse signal recovery. *Mathematical Programming Computation*, 3, (2011), 165–218.

[7] Ben-Tal, A., and Nemirovski, A. *Lectures on Modern Convex Optimization: Analysis, Algorithms, and Engineering Applications*. SIAM, 2001.

[8] Berg, E. V. D., and Friedlander, M. P. Probing the Pareto frontier for basic pursuit solutions. *SIAM J. Sci. Comput.*, 31(2), (2008), 890–912.

[9] Bhaskar, B. N., Tang, G., and Recht, B. Atomic norm denoising with applications to line spectral estimation. *arXiv preprint arXiv:1204.0562*, (2012).

[10] Boyd, S., Ghaoui, L., Feron, E., and Balakrishnan, V. *Linear matrix inequalities in system and control theory*, vol. 14. SIAM, 1994.

[11] Dempster, A. P. Covariance selection. *Biometrics*, 28, (1972), 157–175.

[12] Facchinei, F., and Pang, J.-S. *Finite-dimensional variational inequalities and complementarity problems*, vol. 1-2. Springer-Verlag, 2003.

[13] Fiacco, A. *Introduction to sensitivity and stability analysis in nonlinear programming*. Academic Press, New York, 1983.

[14] Grant, M., Boyd, S., and Ye, Y. Disciplined convex programming. In *Global Optimization: From Theory to Implementation*, L. Liberti and N. Maculan, Eds., Nonconvex Optimization and its Applications. Springer, (2006), 155–210.

[15] Guddat, J., Vasquez, F. G., and Jongen, H. *Parametric Optimization: Singularities, Path-following and Jumps*. Teubner, Stuttgart, 1990.

[16] Hale, E., Yin, W., and Zhang, Y. Fixed-point continuation for ℓ_1-minimization: methodology and convergence. *SIAM J. Optim.*, 19(3), (2008), 1107–1130.

[17] Harmany, Z., Marcia, R., and Willett, R. M. This is SPIRAL-TAP: Sparse poisson intensity reconstruction algorithms - theory and practice. *IEEE Transactions on Image Processing (under review)*, (2012), 1–13.

[18] Hassibi, A., How, J., and Boyd, S. A path following method for solving bmi problems in control. In *Proceedings of American Control Conference*, 2, (1999), 1385–1389.

[19] Jalali, A., and Srebro, N. Clustering using max-norm constrained optimization. In *Proc. of International Conference on Machine Learning (ICML2012)*, (2012), 1–17.

[20] Kyrillidis, A., and Cevher, V. Fast proximal algorithms for self-concordant function minimization with application to sparse graph selection. *Proc. of the 38th International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, (2013), 1–5.

[21] Lu, Z. Adaptive first-order methods for general sparse inverse covariance selection. *SIAM J. Matrix Anal. & Appl.* 31(4), (2010), 2000–2016.

[22] Nemirovski, A. S., and Todd, M. J. Interior-point methods for optimization. *Acta Numerica*, 17(1), (2008), 191–234.

[23] Nesterov, Y. *Introductory lectures on convex optimization: a basic course*, vol. 87 of *Applied Optimization*. Kluwer Academic Publishers, 2004.

[24] Nesterov, Y. Gradient methods for minimizing composite objective function. *CORE Discussion paper*, 76, (2007).

[25] Nesterov, Y., and Nemirovski, A. *Interior-point Polynomial Algorithms in Convex Programming*. Society for Industrial Mathematics, 1994.

[26] Nesterov, Y. E., and Todd, M. J. Self-scaled barriers and interior-point methods for convex programming. *Mathematics of Operations Research*, 22(1), (1997), 1–42.

[27] Nocedal, J., and Wright, S. *Numerical Optimization*, 2 ed. Springer Series in Operations Research and Financial Engineering. Springer, 2006.

[28] Ravikumar, P., Wainwright, M. J., Raskutti, G., and Yu, B. High-dimensional covariance estimation by minimizing ℓ_1-penalized log-determinant divergence. *Electron. J. Statist.*, 5, (2011), 935–988.

[29] Rockafellar, R. T. *Convex Analysis*, vol. 28 of *Princeton Mathematics Series*. Princeton University Press, 1970.

[30] Roos, C., Terlaky, T., and Vial, J.-P. *Interior Point Methods for Linear Optimization*. Springer Science, Heidelberg/Boston, 2006. (Note: This book is a significantly revised new edition of Interior Point Approach to Linear Optimization: Theory and Algorithms).

[31] Scheinberg, K., and Rish, I. SINCO- a greedy coordinate ascent method for sparse inverse covariance selection problem. *Optimization-online (http://www.optimization-online.org/DB_FILE/2009/07/2359.pdf)*, (2009).

[32] Städler, N., Bülmann, P., and de Geer, S. V. l_1-penalization for mixture regression models. *Tech. Report.*, (2012), 1–35.

[33] Sturm, F. Using SeDuMi 1.02: A Matlab toolbox for optimization over symmetric cones. *Optim. Methods Software*, 11-12, (1999), 625–653.

[34] Tang, G., Bhaskar, B. N., Shah, P., and Recht, B. Compressed sensing off the grid. *arXiv preprint arXiv:1207.6053*, (2012).

[35] Tran-Dinh, Q., Necoara, I., Savorgnan, C., and Diehl, M. An Inexact Perturbed Path-Following Method for Lagrangian Decomposition in Large-Scale Separable Convex Optimization. *SIAM J. Optim.*, 23(1), (2013), 95–125.

[36] Tran-Dinh, Q., Kyrillidis, A., and Cevher, V. Composite self-concordant minimization. Tech. report, Lab. for Information and Inference Systems (LIONS), EPFL, Switzerland, CH-1015 Lausanne, Switzerland, (2013), 1–42.

[37] Tran-Dinh, Q., Kyrillidis, A., and Cevher, V. A proximal newton framework for composite minimization: Graph learning without Cholesky decompositions and matrix inversions. *JMLR W& CP*, 28(2), (2013), 271–279.

[38] Tütünkü, R., Toh, K., and Todd, M. Solving semidefinite-quadratic-linear programs using SDPT3. *Math. Program.*, 95, (2003), 189–217.

[39] Wright, S. *Primal-Dual Interior-Point Methods*. SIAM Publications, Philadelphia, 1997.