MATHEMATICS OF OPERATIONS RESEARCH
Vol. 00, No. 0, Xxxxx 0000, pp. 000–000
ISSN 0364-765X | EISSN 1526-5471 | 00 | 0000 | 0001

INFORMS
DOI 10.1287/xxxx.0000.0000
© 0000 INFORMS

A single-phase, proximal path-following framework

Quoc Tran-Dinh
Department of Statistics and Operations Research, UNC, USA, quoctd@email.unc.edu

Anastasios Kyrillidis
University of Texas at Austin, USA, anastasios@utexas.edu

Volkan Cevher
Laboratory for Information and Inference Systems (LIONS), EPFL, Switzerland, volkan.cevher@epfl.ch

We propose a new proximal, path-following framework for a class of—possibly non-smooth—constrained convex problems. We consider settings where the non-smooth part is endowed with a proximity operator, and the constraint set is equipped with a self-concordant barrier. Our main contribution is a new re-parametrization of the optimality condition of the barrier problem, that allows us to process the objective function with its proximal operator within a new path following scheme. In particular, our approach relies on the following two main ideas. First, we re-parameterize the optimality condition as an auxiliary problem, such that a "good" initial point is available; by doing so, a family of alternative paths towards the optimum is generated. Second, we combine the proximal operator of the objective and path-following ideas to design a single phase, proximal, path-following algorithm.

Our method has several advantages. First, it allows handling non-smooth objectives via proximal operators; this avoids lifting the problem dimension via slack variables and additional constraints. Second, it consists of only a *single phase*: While the overall convergence rate of classical path-following schemes for smooth objectives does not suffer from the initialization phase, state-of-the-art proximal path-following schemes undergo slow convergence, in order to obtain a "good" starting point [47]. In this work, we show how to overcome this difficulty in the proximal setting and prove that our scheme has the same $\mathcal{O}(\sqrt{\nu}\log(1/\varepsilon))$ worst-case iteration-complexity with standard approaches [30, 33], but our method can handle nonsmooth objectives, where ν is the barrier parameter and ε is a desired accuracy. Finally, our framework allows errors in the calculation of proximal-Newton search directions, without sacrificing the worst-case iteration complexity. We demonstrate the merits of our algorithm via three numerical examples, where proximal operators play a key role to improve the performance over off-the-shelf interior-point solvers.

Key words: Proximal-Newton method, path-following schemes, non-smooth convex optimization.
MSC2000 subject classification: 90C06; 90C25; 90-08
OR/MS subject classification: Interior-point methods, non-smooth convex programming

1. Introduction.

This paper studies the following constrained convex optimization problem:

$$G^\star := \min_{x\in\mathbb{R}^p}\Big\{G(x) := \langle c, x\rangle + g(x) : x\in\mathcal{X}\Big\}, \tag{1}$$

where $c\in\mathbb{R}^p$, g is a possibly non-smooth, proper, closed and convex function from $\mathbb{R}^p$ to $\mathbb{R}\cup\{+\infty\}$ and $\mathcal{X}$ is a nonempty, closed and convex set in $\mathbb{R}^p$.[1]

For generic $\mathcal{X}$ and for G just linear or quadratic, *interior point methods* (IPMs) often constitute the method-of-choice for the numerical solutions of (1), with a well-characterized worst-case

[1] In our discussion, we separate the linear term $\langle c, x\rangle$ from g for our convenience in processing numerical examples in the last section. This linear term can be absorbed into g, and does not affect our analysis. The structure of $\mathcal{X}$ highly affects the efficiency of optimization schemes for (1). While simple constraints are suitable for projected optimization methods, complicated linear constraints can be handled by penalty or augmented Lagrangian techniques, combined with splitting or alternating direction methods [8, 9, 35]. In this work though, we even consider problem cases with more complicated structures, such as hyperbolic or nonstandard cones, where such approaches may not apply or may be no longer efficient.

complexity. A non-exhaustive list of special instances of (1) includes linear programs, quadratic programs, second order cone programs and, semi-definite programs [1, 6, 7, 16, 27, 28, 30, 34, 39, 42, 51, 52].

At the heart of IPMs lies the notion of *interior barriers*: these mimic the effect of the constraint set $\mathcal{X}$ in (1) by appropriately weighting the objective function with a barrier f over the set $\mathcal{X}$, as follows:

$$F_t^\star := \min_{x\in\operatorname{int}(\mathcal{X})} \Big\{ F_t(x) := G(x) + tf(x) \Big\}; \tag{2}$$

here, f models the structure of the feasible set $\mathcal{X}$ and $t > 0$ is a penalty parameter. For different values of t, the regularized problem generates a sequence of solutions $\{x^\star(t) : t > 0\}$, known as *central path*, converging to $x^\star$ of (1), as t goes to 0^+. *Path-following IPMs* operate along the central path: Starting from a decent initial point and, for a properly decreasing sequence of t values, they solve (2) only approximately, by performing a "few"[2] Newton iterations for each t value. For path-following schemes to work with attractive guarantees, the initial point must lie within the quadratic convergence region of the Newton sub-problems. Indeed, the standard path-following strategy guarantees that each approximate solution of (2) lies within Newton's method quadratic convergence region for the next value of t, and operates as warm-start for that new problem instance [7, 29, 33, 30]. In their seminal work [33], Nesterov and Nemirovski showed that such path-following schemes admit a polynomial worst-case complexity, as long as the underlying Newton method has a polynomial complexity.

Based on the above, standard path-following schemes [27, 30, 33] could be characterized by two phases: Phase I and Phase II. In Phase I and for an initial value of t, say t_0, one has to determine a good initial point for Phase II; this requires solving (2) up to sufficient accuracy, such that the Newton method for (2) admits fast convergence. In Phase II and using the output of Phase I as a warm-start, we path-follow with a provably polynomial time complexity.

Taking into account both phases, standard path-following schemes—where (1) is a smooth objective—are characterized by the following iteration complexity: The total number of iterations required to obtain an ε-solution is

$$\mathcal{O}(\sqrt{\nu}\log(1/\varepsilon)), \tag{3}$$

where ν is a barrier parameter (see Section 2 for details) and ε is the approximate parameter, according to the following definition:

Definition 1. Given a tolerance $\varepsilon > 0$, we say that $x_\varepsilon^\star$ is an ε-solution for (1) if

$$x_\varepsilon^\star \in \mathcal{X}, \quad \text{and} \quad G(x_\varepsilon^\star) - G^\star \leq \varepsilon.$$

1.1. Path-following schemes for non-smooth objectives.

For many applications in machine learning, optimization and signal processing [7, 37, 47], g is usually non-smooth in order to leverage the true underlying structure in $x^\star$. An exemplar of such g is the ℓ_1-norm regularization, *i.e.*, $g(x) = \|x\|_1$, with applications in high-dimensional statistics, compressive sensing, scientific and medical imaging [17, 21, 45, 38, 53, 26, 12, 19], among others. Other examples for g include the $\ell_{1,2}$-group norm [3, 22, 23] and the nuclear norm [11].

Unfortunately, non-smoothness of the objective reduces the optimization efficiency. In such settings, one can often reformulate (1) into a standard conic program, by introducing slack variables to model g. Such a technique is known as *disciplined convex programming* (DCP) [18] and has been incorporated in well-known software packages, such as CVX [18] and YALMIP [25]. Existing off-the-shelf solvers are then utilized to solve the resulting problem. However, DCP could potentially increase the problem dimension significantly; this, in sequence, reduces the efficiency of the

[2] Standard path-following schemes perform just *one* Newton iteration.

IPMs. For instance, in the example above where $g(x) = \|x\|_1$, DCP introduces p slack variables to reformulate g into p additional second-order cone constraints.

In this paper, we focus on cases where g is endowed with a low-cost proximity operator:

$$\mathrm{prox}_g(x) := \arg\min_{u \in \mathbb{R}^p} \left\{ g(u) + {}^1\!/\!{}_2 \cdot \|u - x\|^2 \right\}.$$

If g has a tractable proximity operator, then we can often preserve the optimization efficiency; *e.g.*, for simple objectives, using such proximity operators have been proven to be practical in real applications [5, 13, 30]. However, for generic $\mathcal{X}$ constraints in (1), the resulting interior barrier f in (2) does not have Lipschitz continuous gradients and, thus, prevents us from using such schemes. This fact necessitates the design of a new breed of path-following schemes, that can accommodate non-smooth terms in the objective.

[47] is one of the first works that treat jointly interior barrier path-following schemes and proximity operators, in order to design a new proximal path-following algorithm for problems as in (1). According to [47], the proposed algorithm follows a two-phase approach, with Phase II having the same worst-case iteration-complexity (3) (up to constants) with standard smooth path-following schemes [30, 33]. However, the initialization Phase I in the proposed scheme requires substantial computational effort, which usually dominates the overall computational time. In particular, using a damped-step proximal-Newton scheme to find a good initial point to be used in Phase II, the algorithm in [47] requires

$$\left\lfloor \frac{F_{t_0}(x^0) - F_{t_0}(x^\star_{t_0})}{\omega\left((1-\kappa)\beta\right)} \right\rfloor$$

Newton iterations in Phase I, for arbitrary selected $t_0 > 0$ and x^0, and $\kappa \in (0,1), \beta \in (0, 0.15]$, $\omega(q) = q - \log(1+q)$; see [47, Theorem 4.4] for more details. *I.e.*, in stark contrast to the global iteration complexity (3) of smooth path-following schemes, Phase I of [47] for non-smooth objectives undergoes a sublinear convergence rate.

1.2. Motivation.

From our discussion so far, it is clear that most existing works on path-following IPMs require two phases. In the case of smooth objectives in (1), Phase I is often implemented as a damped-step Newton scheme, which has a sublinear convergence rate, or an auxiliary path-following scheme, with a linear convergence rate that satisfies the global, worst-case complexity in (3) [30, 33]. In standard conic programming, one can unify a two-phase algorithm in a single-phase IP path-following scheme via homogeneous and self-dual embedded strategies; see, *e.g.*, [43, 50, 52]. Such strategies parameterize the KKT condition of the primal and dual conic program so that one can immediately have an initial point, without performing Phase I. Unfortunately, to the best of our knowledge, such single-phase approaches have not been yet studied for proximal, path-following IPMs, in order to handle non-smooth, nonlinear constrained convex problems.

1.3. Our contributions.

The goal of this paper is to develop a new single-phase, proximal path-following algorithm for (1). To do so, we first re-parameterize the optimality condition of the barrier problem associated with (1) as a *parametric monotone inclusion* (PMI). Then, we design an appropriate proximal path-following scheme to approximate the solution of such PMI, while controlling the penalty parameter. Finally, we show how to recover an approximate solution of (1), from the approximate solution of the PMI. Thus, with an appropriate choice of parameters, we show how we can eliminate Phase I, while we still maintain the global, polynomial time, worst-case iteration-complexity.

The main contributions of this paper can be summarized as follows:

(i) We introduce a new parameterization for the optimality condition of (2) to appropriately select the parameters such that much less computation for initialization is needed. Hence, we can eliminate Phase I in the traditional path-following scheme.

(*ii*) We design a single-phase, path-following algorithm to compute an ε-solution of (1). For each t value, the resulting algorithm only requires a *single proximal Newton iteration* of a strongly convex quadratic composite subproblem. The algorithm also allows inexact proximal Newton steps, with a verifiable stopping criterion (*cf.* eq. (20)).

In particular, we establish the following result:

Theorem 1. *The total number of proximal Newton iterations required by the proposed algorithm to reach an ε-solution of* (1) *is upper bounded by* $\mathcal{O}\left(\sqrt{\nu}\log\left(\frac{\nu}{\varepsilon}\right)\right)$.

A complete and formal description of the above theorem and its proof are provided in Section 4. Our *proximal* algorithm admits the same iteration-complexity, as standard path-following methods [30, 33] (up to a constant).

1.4. The structure of the paper.

This paper is organized as follows. Sections 2 and 3 contain basic definitions and notions, used in our analysis. There, we also introduce a new re-parameterization of the central path in order to obtain a *predefined* initial point. Section 4 presents a novel algorithm and its complexity theory for the non-smooth objective function. Section 5 provides three numerical examples that highlight the merits of our algorithm. Technical discussions and proofs are deferred to the appendix.

2. Preliminaries.

In this section, we provide the basic notation used in the rest of the paper, as well as two key concepts: proximity operators and self-concordant (barrier) functions.

2.1. Basic definitions.

Given $x, y \in \mathbb{R}^p$, we use $\langle x, y\rangle$ or $x^T y$ to denote the inner product in $\mathbb{R}^n$. For a proper, closed and convex function g, we denote by $\mathrm{dom}(g)$ its domain, (*i.e.*, $\mathrm{dom}(g) := \{x \in \mathbb{R}^n : g(x) < +\infty\}$), and by $\partial g(x) := \{v \in \mathbb{R}^n \ : \ g(y) \geq g(x) + \langle v, y - x\rangle, \ \forall y \in \mathrm{dom}(g)\}$ its subdifferential at x. We also denote by $\mathrm{Dom}(g) := \mathrm{cl}(\mathrm{dom}(g))$ the closure of $\mathrm{dom}(g)$ [40]. We use $\mathcal{C}^3(\mathcal{X})$ to denote the class of three times continuously differentiable functions from $\mathcal{X} \subseteq \mathbb{R}^p$ to $\mathbb{R}$.

For a given twice differentiable function f such that $\nabla^2 f(x) \succ 0$ at some $x \in \mathrm{dom}(f)$, we define the local norm, and its dual, as

$$\|u\|_x := \langle \nabla^2 f(x) u, u\rangle^{1/2}, \forall u \in \mathbb{R}^n, \quad \text{and} \quad \|v\|_x^* := \max_{\|u\|_x \leq 1} \langle u, v\rangle = \langle \nabla^2 f(x)^{-1} v, v\rangle^{1/2},$$

respectively, for $u, v \in \mathbb{R}^p$. Note that the Cauchy-Schwarz inequality holds, *i.e.*, $\langle u, v\rangle \leq \|u\|_x \|v\|_x^*$.

2.2. Proximity operators.

The proximity operator of a proper, closed and convex function g is defined as the following strongly convex program:

$$\mathrm{prox}_g(x) := \arg\min_{u \in \mathbb{R}^p} \left\{ g(u) + {}^1\!/_2 \cdot \|u - x\|^2 \right\}. \tag{4}$$

In general, computing prox_g is nearly as hard as minimizing g itself. However, there exist several structured smooth and non-smooth convex functions g that have a closed-form solution or a low-cost evaluation of the proximity operator. We capture this idea in the following definition.

Definition 2 (*Tractable proximity operator*). A proper, closed and convex function $g : \mathbb{R}^p \to \mathbb{R} \cup \{+\infty\}$ has a *tractable* proximity operator if (4) can be computed efficiently via a closed-form solution or via a polynomial time algorithm.

Examples of such functions include the ℓ_1-norm—where the proximity operator is the well-known soft-thresholding operator [13]—and the indicator functions of simple sets (*e.g.*, boxes, cones and simplexes)—where the proximity operator is simply the projection operator. Further examples can be found in [4, 13, 37].

2.3. Self-concordant functions and self-concordant barriers.

A concept used in our analysis is the self-concordance property, introduced by Nesterov and Nemirovskii [30, 33].

Definition 3. A univariate convex function $\varphi \in \mathcal{C}^3(\mathrm{dom}(\varphi))$ is called *standard self-concordant* if $|\varphi'''(\tau)| \le 2\varphi''(\tau)^{3/2}$ for all $\tau \in \mathrm{dom}(\varphi)$, where $\mathrm{dom}(\varphi)$ is an open set in $\mathbb{R}$. Moreover, a function $f : \mathrm{dom}(f) \subseteq \mathbb{R}^n \to \mathbb{R}$ is standard self-concordant if, for any $X \in \mathrm{dom}(f)$ and $v \in \mathbb{R}^n$, the univariate function φ where $\tau \mapsto \varphi(\tau) := f(x + \tau v)$ is standard self-concordant.

Definition 4. A standard self-concordant function $f : \mathrm{dom}(f) \subset \mathbb{R}^n \to \mathbb{R}$ is a *ν-self-concordant barrier* for the set $\mathrm{Dom}(f)$ with parameter $\nu > 0$, if

$$\sup_{u \in \mathbb{R}^n} \left\{ 2\langle \nabla f(x), u\rangle - \|u\|_x^2 \right\} \le \nu, \quad \forall x \in \mathrm{dom}(f).$$

In addition, $f(x) \to \infty$ as x tends to the boundary of $\mathrm{dom}(f)$.

We note that when $\nabla^2 f$ is non-degenerate (particularly, when $\mathrm{dom}(f)$ contains no straight line [30, Theorem 4.1.3.]), a ν-self-concordant function f satisfies

$$\|\nabla f(x)\|_x^* \le \sqrt{\nu}, \quad \forall x \in \mathrm{dom}(f). \tag{5}$$

Self-concordant functions have non-global Lipschitz gradients and can be used to analyze the complexity of Newton-methods [10, 30, 33], as well as first-order variants [15]. For more details on self-concordant functions and self-concordant barriers, we refer the reader to Chapter 4 of [30].

Several simple sets are equipped with a self-concordant barrier. For instance, $f_{\mathbb{R}^n_+}(x) := -\sum_{i=1}^n \log(x_i)$ is a n-self-concordant barrier of the orthant cone $\mathbb{R}^n_+$, $f(x,t) = -\log(t^2 - \|x\|_2^2)$ is a 2-self-concordant barrier of the Lorentz cone $\mathcal{L}_{n+1} := \{(x,t) \in \mathbb{R}^n \times \mathbb{R}_+ : \|x\|_2 \le t\}$, and the semidefinite cone $\mathbb{S}^n_+$ is endowed with the n-self-concordant barrier $f_{\mathbb{S}^n_+}(X) := -\log\det(X)$.

Finally, we define the analytical center $\bar{x}^\star_f$ of f as

$$\bar{x}^\star_f := \arg\min\left\{ f(x) : x \in \mathrm{int}(\mathcal{X}) \right\} \quad \Leftrightarrow \quad \nabla f(\bar{x}^\star_f) = 0. \tag{6}$$

If $\mathcal{X}$ is bounded, then $\bar{x}^\star_f$ exists and is unique [31]. Some properties of the analytical center, important for our scheme, are presented in Section 3. In this paper, we develop algorithms for (1) with a general self-concordant barrier f of $\mathcal{X}$ as defined by Definition 4.

2.4. Basic assumptions.

We make the following assumption, regarding problem (1).

Assumption 1. *The solution set $\mathcal{X}^\star$ of (1) is nonempty. The objective function g in (1) is proper, closed and convex, and $\mathcal{X} \subseteq \mathrm{dom}(g)$. The feasible set $\mathcal{X}$ is nonempty, closed and convex (with nonempty interior $\mathrm{int}(\mathcal{X})$) and is endowed with a ν-self-concordant barrier f such that $\mathrm{Dom}(f) := cl(\mathrm{dom}(f)) = \mathcal{X}$. The analytical center $\bar{x}^\star_f$ of f exists.*

Except for the last condition, Assumption 1 is common for interior-point methods. The last condition can be satisfied by adding an auxiliary constraint $\|x\|_2 \le R$ for sufficiently large R; this technique has been also used in [33] and it does not affect the solution of (1) when R is large.

3. Re-parameterizing the central path.

In this section, we introduce a new parameterization strategy, which will be used in our scheme for (1).

3.1. Barrier formulation and central path of (1).

Since $\mathcal{X}$ is endowed with a ν-self-concordant barrier f, according to Assumption A.1, the barrier formulation of (1) is given by

$$F_t^\star := \min_{x \in \mathrm{int}(\mathcal{X})} \Big\{ F_t(x) := G(x) + tf(x) \equiv \langle c, x\rangle + g(x) + tf(x) \Big\}, \tag{7}$$

where $t > 0$ is the penalty parameter. We denote by $\bar{x}^\star_t$ the solution of (7) at a given value $t > 0$. Define $r_t(x) := c + \partial g(x) + t\nabla f(x)$. The optimality condition of (7) is necessary and sufficient for $\bar{x}^\star_t$ to be an optimal solution of (7), and can be written as follows:

$$0 \in r_t(\bar{x}^\star_t) \equiv c + \partial g(\bar{x}^\star_t) + t\nabla f(\bar{x}^\star_t). \tag{8}$$

We also denote by $\bar{\mathcal{C}} := \{\bar{x}^\star_t : t > 0\}$ the set of solutions of (7), which generates a central path (or a solution trajectory) associated with (1). We refer to each solution $\bar{x}^\star_t$ as a central point.

3.2. Parameterization of the optimality condition. Let us fix $x^0 \in \mathrm{dom}(f)$; a specific selection of x^0 is provided later on. For given x^0, let $\xi_0 \in \partial g(x^0)$ be an arbitrary subgradient of g at x^0, and set $\zeta_0 := \nabla f(x^0) + t_0^{-1}(c + \xi_0)$. For a given parameter $\eta > 0$, define

$$h_\eta(x) := f(x) - \eta\langle \zeta_0, x\rangle \quad \text{and} \quad r_{t,\eta}(x) := c + \partial g(x) + t\nabla h_\eta(x). \tag{9}$$

with the gradient $\nabla h_\eta(x) := \nabla f(x) - \eta\zeta_0$. We further define a η-parameterized version of (7) as

$$H_t^\star := \min_x \left\{ H_t(x) := \langle c, x\rangle + g(x) + t h_\eta(x) \right\}. \tag{10}$$

Observe that, for a fixed value of $\eta > 0$, the optimality condition of (10) is given by

$$0 \in r_{t,\eta}(x_t^\star) \equiv c + \partial g(x_t^\star) + t\nabla h_\eta(x_t^\star). \tag{11}$$

Due to the convexity of f and g, both h_η and $r_{t,\eta}$ are monotone in the sense of nonlinear operator theory [41]. Since f is a barrier function, its domain is not the whole space and, hence, h_η and $r_{t,\eta}$ are not maximal. Clearly, when g is smooth, (11) reduces to a system of nonlinear equations.

We provide next some remarks regarding the η-parameterized problem in (10):

- Observe that, if we set $\eta = 0$, $h_\eta(x) \equiv f(x)$ and thus, (10) is equivalent to (7). Therefore, for any other value $\eta > 0$, the problem in (10) differs from the original formulation (7) by a factor $-t\eta\langle\zeta_0, x\rangle$.
- Fix parameters $\eta > 0, t > 0$ and let $x_t^\star$ be the solution of (10), which is different from the solution $\bar{x}_t^\star$ of (7), given the remark above. However, as $t \to 0$ in a path-following scheme, both $x_t^\star$ and $\bar{x}_t^\star$ converge to an optimum $x^\star$ of (1).
- Based on the above, for fixed $t > 0$ and different values of η, (10) leads to a family of paths towards $x^\star$ of (1).

Our aim in this paper is to properly combine the quantities t_0, x^0 and η, such that (i) solving iteratively (10) always has fast convergence (even at the initial point x^0) and, (ii) while (10) differs from (7), its solution trajectory is closely related to the solution trajectory of the original barrier formulation. The above are further discussed in the next subsections.

3.3. A functional connection between solutions of (7) and (10) and the key role of $\bar{x}_f^\star$. Given the definitions above, let us first study the relationship between *exact* solutions of (7) and (10), for fixed values $t > 0$ and $\eta > 0$.

Lemma 1. *Let $t > 0$ be fixed. Assume $\eta > 0$ and ζ_0 be chosen such that $\bar{m}_0 = \eta\|\zeta_0\|_{\bar{x}_t^\star}^* < 1$. Define $\bar{\Delta}_t := \|x_t^\star - \bar{x}_t^\star\|_{\bar{x}_t^\star}$ as the local distance between $\bar{x}_t^\star$ and $x_t^\star$, the solutions of (7) and (10), respectively. Then,*

$$\bar{\Delta}_t \leq \frac{\bar{m}_0}{1 - \bar{m}_0}.$$

The proof is provided in Appendix 7.1. The above lemma indicates that, while (7) and (10) define different central paths towards $x^\star$, there is an upper bound on the distance between $\bar{x}_t^\star$ and $x_t^\star$, which is controlled by the selection of η, t_0 and x^0. However, $\|\zeta_0\|_{\bar{x}_t^\star}^*$ cannot be evaluated a priori, since $\bar{x}_t^\star$ is unknown.

We can overcome this difficulty by using the *analytical center point* $\bar{x}_f^\star$ in (6). A key property of $\bar{x}_f^\star$ is the following [30, Corollary 4.2.1]: Define $n_\nu := \nu + 2\sqrt{\nu}$, where ν is the self-concordant barrier parameter. Then, $\|v\|_x^* \leq n_\nu\|v\|_{\bar{x}_f^\star}^*$ for any $x \in \mathrm{int}(\mathcal{X})$ and $v \in \mathbb{R}^p$. If f is a logarithmically homogeneous barrier function, then $n_\nu := 1$ and $\|v\|_x^* \leq \|v\|_{\bar{x}_f^\star}^*$. The observation leads to the following Corollary; the proof easily follows from that of Lemma 1 and the properties above.

COROLLARY 1. *Consider the configuration in Lemma 1 and define $m_0 = \eta n_\nu \|\zeta_0\|^*_{\bar{x}^\star_f} < 1$. Then,*

$$\bar{\Delta}_t \leq \frac{m_0}{1-m_0}. \tag{12}$$

Moreover, if we choose as the initial point $x^0 := \bar{x}^\star_f$, then $m_0 = n_\nu t_0^{-1}\eta\|c+\xi_0\|^*_{\bar{x}^\star_f}$.

In the corollary above, we bound the quantity m_0 using the local norm at the analytical center $\bar{x}^\star_f$. This allows us to estimate the theoretical worst-case bound in Theorem 3, described next. In practice—and if a "better" starting point is provided—one can alternatively redefine the quantities above and remove the dependency on $\bar{x}^\star_f$, without affecting the analysis followed in our scheme.

The above observations lead to the following lemma: given a point x, we bound $\|x-\bar{x}^\star\|_{\bar{x}^\star_t}$ by the distance $\|x-x^\star\|_{x^\star_t}$, using the bound (12); the proof is given in Appendix 7.2.

LEMMA 2. *Consider the configuration in Corollary 1, such that $m_0 < \frac{1}{2}$. Let $\lambda_t(x) := \|x-x^\star_t\|_{x^\star_t}$ and $\bar{\lambda}_t(x) := \|x-\bar{x}^\star_t\|_{\bar{x}^\star_t}$, for any $x\in \mathrm{int}(\mathcal{X})$. Then, the following connection between $\lambda_t(x)$ and $\bar{\lambda}_t(x)$ holds:*

$$\bar{\lambda}_t(x) \leq \frac{\lambda_t(x)}{1-\bar{\Delta}_t} + \bar{\Delta}_t \leq \frac{(1-m_0)\lambda_t(x)}{1-2m_0} + \frac{m_0}{1-m_0}. \tag{13}$$

The above lemma indicates that, given fixed $t>0$, any approximate solution $\widehat{x}_t$ to (10), that is "good" enough (*i.e.*, the metric $\lambda_t(\widehat{x}_t)$ is small), signifies that $\widehat{x}_t$ is also "close" to the optimal of (7) (*i.e.*, the metric $\bar{\lambda}_t(\widehat{x}_t)$ is bounded by $\lambda_t(\widehat{x}_t)$ and, thus, can be controlled). This fact allows the use of (10), instead of (7), and provides freedom to cleverly select initial parameters t_0 and η for faster convergence. The next section proposes such an initialization procedure.

3.4. The choice of initial parameters.

Here, we describe how we initialize t_0 and η. For ease of presentation and based on the discussion above, we choose $x^0 = \bar{x}^\star_f$. Observe that, for such x^0, we have $\nabla f(x^0) = \nabla f(\bar{x}^\star_f) = 0$.

Lemma 2 suggests that, for some $\beta\in(0,1)$, if we can bound $\lambda_t(\cdot)$ as $\lambda_t(\cdot)\leq\beta$, then $\bar{\lambda}_t(\cdot)$ is bounded as $\bar{\lambda}_t(\cdot) \leq \frac{(1-m_0)\beta}{1-2m_0} + \frac{m_0}{1-m_0}$. This observation leads to the following lemma; the proof is provided in Appendix 7.3.

LEMMA 3. *Let $\lambda_{t_0}(x^0) := \|x^0 - x^\star_{t_0}\|_{x^\star_{t_0}}$, where $x^\star_{t_0}$ is the solution of (10) at $t := t_0$. Let $\xi_0 \in \partial g(x^0)$ and, from (9), $r_{t_0,\eta}(x^0) := c+\xi_0+t_0\nabla h_\eta(x^0)$. Then, we have*

$$\lambda_{t_0}(x^0) \leq \frac{1-\gamma_{t_0}-\sqrt{1-6r_{t_0,\eta}+\gamma_{t_0}^2}}{2}, \tag{14}$$

provided that $\gamma_{t_0} := \|r_{t_0,\eta}(x^0)\|^*_{x^0} \equiv |1-\eta|\,\|\zeta_0\|^*_{x^0} < 3-2\sqrt{2}$.

In plain words, Lemma 3 provides a recipe for initial selection of parameters: Our goal is to choose an initial point x^0 such that $\lambda_{t_0}(x^0)\leq\beta$, for a predefined constant $\beta\in(0,1)$. Using (14), we observe that in order to satisfy $\lambda_{t_0}(x^0)\leq\beta$, it is sufficient to require

$$1-\gamma_{t_0}-\sqrt{1-6\gamma_{t_0}+\gamma_{t_0}^2}\leq 2\beta \quad\Rightarrow\quad \gamma_{t_0}\leq\frac{\beta(1-\beta)}{1+\beta}.$$

Since

$$\gamma_{t_0} = |1-\eta|\,\|\zeta_0\|^*_{x^0},$$

the inequality $\gamma_{t_0}\leq\frac{\beta(1-\beta)}{1+\beta}$ further implies

$$|1-\eta| \leq \frac{\beta(1-\beta)}{(1+\beta)|\,\|\zeta_0\|^*_{x^0}}.$$

Hence, we obtain

$$\eta \in \left[1 - \frac{\beta(1-\beta)}{(1+\beta)|\,\|\zeta_0\|_{x^0}^*}, \; 1 + \frac{\beta(1-\beta)}{(1+\beta)|\,\|\zeta_0\|_{x^0}^*}\right]. \tag{15}$$

As we describe next, by our theory, it holds $\bar{\lambda}_t(\cdot) \le \frac{(1-m_0)\beta}{1-2m_0} + \frac{m_0}{1-m_0} < 1$. Since $\frac{m_0}{1-m_0} \le \frac{m_0}{1-2m_0}$, we have

$$\frac{(1-m_0)\beta}{1-2m_0} + \frac{m_0}{1-m_0} \le \frac{(1-m_0)\beta + m_0}{1-2m_0} < 1.$$

This further leads to $m_0 < \frac{1-\beta}{3+\beta}$ and, by definition of m_0 and $x^0 \equiv \bar{x}_f^\star$, we have

$$\eta n_\nu \|\zeta_0\|_{\bar{x}_f^\star}^* = \frac{\eta n_\nu}{t_0}\|c + \xi_0\|_{\bar{x}_f^\star}^* < \frac{1-\beta}{3+\beta}.$$

If we take $\eta = 1$, which is satisfies (15), then we can choose t_0 such that

$$t_0 > \frac{(1-\beta)}{(3+\beta)n_\nu\|c+\xi_0\|_{\bar{x}_f^\star}^*}. \tag{16}$$

This condition provides a rule to select t_0. In general, t_0 can be chosen based on the value of η selected from (15) as $t_0 > \frac{(1-\beta)}{\eta(3+\beta)n_\nu\|c+\xi_0\|_{\bar{x}_f^\star}^*}$.

4. The single phase proximal path-following algorithm.

In this section, we present the main ideas of our algorithm. According to the previous section, to solve (1), one can parameterize the path-following scheme (7) into (10) and, given proper initialization, solve iteratively (10)—i.e., in a path-following fashion, for decreasing values of t.

In Subsections 4.1 and 4.2, we describe schemes to solve (10) up to some accuracy and how the errors, due to approximation, propagate into our theory. Based on these ideas, Subsection 4.3 describes the main recursion of our algorithm, along with the update rule for t parameter. Subsection 4.4 provides a practical stopping criterion procedure, such that an ε-solution is achieved. Subsection 4.5 provides an overview of the algorithm and its theoretical guarantees.

4.1. An exact proximal Newton scheme.

In our discussions so far, $x_t^\star$ denotes the *exact* solution to (10), for a given value of paramter t. Since optimizing the objective in its original form is a difficult task, it is common practice to iteratively solve (10) via first- or second-order Taylor approximations of the smooth part.

In this work, we focus on Newton-type solutions. Let $Q(\cdot; y)$ be the second-order Taylor approximation of $h_\eta(\cdot)$ around y, *i.e.:*

$$\begin{aligned} Q(x;y) &:= \langle \nabla h_\eta(y), x - y\rangle + \frac{1}{2}\langle \nabla^2 h_\eta(y)(x-y), x-y\rangle \\ &= \langle \nabla f(y) - \eta\zeta_0, x-y\rangle + \frac{1}{2}\langle \nabla^2 f(y)(x-y), x-y\rangle. \end{aligned}$$

Then, $x_t^\star$ can be obtained by iteratively solving

$$x_t^{k+1} \longleftarrow \mathop{\arg\min}_{x\in\mathrm{int}(\mathcal{X})}\left\{\hat{F}_t(x; x_t^k) := tQ(x; x_t^k) + G(x)\right\}, \tag{17}$$

with *perfect* accuracy, and x_t^k is in a convergence region of such a proximal Newton method. Then,

$$x_t^\infty \equiv x_t^\star.$$

We note that, for given point x_t^k, we can write the optimality condition of (17) as follows:

$$0 \in t\left[\nabla h_\eta(x_t^k) + \nabla^2 h_\eta(x_t^k)(x_t^\star - x_t^k)\right] + c + \partial g(x_t^\star). \tag{18}$$

To solve (17), one can use convex quadratic composite minimization solvers; see, *e.g.*, [5, 9, 32].

4.2. Inexact proximal Newton scheme. In practice, we can not solve (17) exactly due to the nonsmooth part g, but only hope for an approximate solution, up to a given accuracy $\delta > 0$ [24]. The next definition characterizes such inexact solutions.

Definition 5. Fix $t > 0$ and let w be an anchor point (in (17), $w = x_t^k$). Moreover, let $x_t^\star$ be the exact solution, obtained by solving (17) perfectly. We say that a point $z \in \mathrm{int}\,(\mathcal{X})$ is a δ-solution to (17) if

$$\|z - x_t^\star\|_w \leq \delta, \tag{19}$$

for a given tolerance $\delta \geq 0$. We denote this notion by $z :\approx x_t^\star$.

Unfortunately, $x_t^\star$ is unknown and, thus, we can not check the condition (19). This condition however holds indirectly, when the following holds [47]:

$$\hat{F}_t(z; w) - \hat{F}_t(x_t^\star; w) \leq \frac{t\delta^2}{2}, \tag{20}$$

where $\hat{F}_t(\cdot;\cdot)$ is defined in (17). This last condition can be evaluated via several convex optimization algorithms, including first-order methods, *e.g.*, [5, 32].

We will use these ideas next to define our inexact proximal-Newton path-following scheme.

4.3. A new, inexact proximal-Newton path-following scheme. Here, we design a new, path-following scheme that operates over the re-parameterized central path in (10). This new algorithm chooses an initial point, as described in Section 3, and selects values for parameter t via a new update rule, that differs from that of [47].

At the heart of our approach lies the following recursion:

$$\begin{cases} t_{k+1} & := t_k + d_k, \\ x_{t_{k+1}}^{k+1} & :\approx \underset{x \in \mathrm{int}(\mathcal{X})}{\arg\min} \left\{ \hat{F}_{t_{k+1}}(x; x_{t_k}^k) := t_{k+1} Q(x; x_{t_k}^k) + G(x) \right\}. \end{cases} \tag{21}$$

That is, starting from initial points t_0 and $x^0 \equiv x_{t_0}^0$, we update the penalty parameter t from t_k to t_{k+1} via the rule $t_{k+1} := t_k + d_k$, at the k-th iteration; see next for details. Then, we perform *a single* proximal-Newton iteration, in order to approximate the solution to the minimization problem in (21). Observe that, while such a step roughly approximates the minimizer of (21) satisfying (19), in our analysis we can still guarantee convergence close to $x^\star$ of (1), using ideas in Subsection 4.2.

Update rule for parameter t. With $x_{t_k}^k$ being the inexact solution of (21) and $x_{t_k}^\star$ the exact solution of (11) at $t = t_k$, we define the local distances

$$\lambda_t(x) := \|x - x_t^\star\|_{x_t^\star}, \quad \text{and} \quad \Delta_k := \|x_{t_{k+1}}^\star - x_{t_k}^\star\|_{x_{t_{k+1}}^\star}, \tag{22}$$

for any $t > 0$ and $x \in \mathrm{int}\,(\mathcal{X})$. Before we provide a closed-form solution for d_k in the update rule $t_{k+1} := t_k + d_k$, we require the following lemma, which shows the relation between $\lambda_{t_{k+1}}(x_{t_{k+1}}^{k+1})$ and $\lambda_{t_{k+1}}(x_{t_k}^k)$, as well as the relation between $\lambda_{t_k}(x_{t_k}^k)$ and Δ_k. The proof of this result can be found in Appendix 7.4.

Lemma 4. *Given $x_{t_k}^k$ and t_{k+1}, let $x_{t_{k+1}}^{k+1}$ be an approximation of $x_{t_{k+1}}^\star$ computed by the inexact proximal-Newton scheme (21). Let $\lambda_{t_{k+1}}(x_{t_k}^k) = \|x_{t_k}^k - x_{t_{k+1}}^\star\|_{x_{t_{k+1}}^\star}$ and $\lambda_{t_{k+1}}(x_{t_{k+1}}^{k+1}) = \|x_{t_{k+1}}^{k+1} - x_{t_{k+1}}^\star\|_{x_{t_{k+1}}^\star}$ be defined by (22). If $\lambda_{t_{k+1}}(x_{t_k}^k) \in [0, 0.118975]$, then we have*

$$\lambda_{t_{k+1}}(x_{t_{k+1}}^{k+1}) \leq 1.135042\delta_{k+1} + 5\lambda_{t_{k+1}}(x_{t_k}^k)^2. \tag{23}$$

Moreover, the right-hand side of (23) is nondecreasing w.r.t. $\lambda_{t_{k+1}}(x_{t_k}^k)$ and $\delta_{k+1} \geq 0$.

Let Δ_k be defined by (22). Then, we have the following estimate:

$$\lambda_{t_{k+1}}(x_{t_k}^k) \leq \frac{\lambda_{t_k}(x_{t_k}^k)}{1 - \Delta_k} + \Delta_k, \tag{24}$$

provided that $\Delta_k < 1$.

In words, (23) reveals that the quadratic convergence rate of consecutive inexact proximal-Newton steps in (21) (measured by $\lambda_{t_k}(x^k_{t_k})$) is preserved per iteration. Moreover, (23) describes how the approximation parameter δ_{k+1} accumulates over iterations (*i.e.*, it is an additive term).

Now, we need to bound the distance Δ_k. The next lemma shows how we can bound Δ_k based on the update rule $t_{k+1} = t_k + d_k$ for $d_k \neq 0$; the proof is provided in Appendix 7.5. This lemma also provides a rule for d_k selection.

Lemma 5. *Given constant $c_\beta > 0$, let $\sigma_\beta := \frac{c_\beta}{(1+c_\beta)\sqrt{\nu}}$. Then, Δ_k defined by (22) satisfies*

$$\frac{\Delta_k}{1+\Delta_k} \leq \frac{|d_k|}{t_k}\|\nabla f(x^*_{t_{k+1}})\|^*_{x^*_{t_{k+1}}} \leq \frac{|d_k|\sqrt{\nu}}{t_k}. \tag{25}$$

Moreover, if we choose $d_k := -\sigma_\beta t_k$, the Δ_k is bounded by $\Delta_k \leq c_\beta$.

Based on the above, we describe next the main result of this section: Assume that the point $x^k_{t_k}$ is in the quadratic convergence region of the inexact proximal-Newton method (21), *i.e.*, $\lambda_{t_k}(x^k_{t_k}) \leq \beta$ for given $\beta \in (0, 0.118975]$. The following theorem describes a condition on Δ_k such that $\lambda_{t_{k+1}}(x^{k+1}_{t_{k+1}}) \leq \beta$. This in sequence determines the update rule of t values. The following theorem summarizes this requirement.

Theorem 2. *Let $\{\lambda_{t_k}(x^k_{t_k})\}$ be the sequence generated by the inexact proximal-Newton scheme (21). For any $\beta \in (0, 0.118975]$, if we choose δ_k and Δ_k such that*

$$\delta_{k+1} \leq 0.066517\beta \quad \text{and} \quad \Delta_k \leq \frac{1}{2}\left[1 + 0.43\sqrt{\beta} - \sqrt{(1-0.43\sqrt{\beta})^2 + 4\beta}\right], \tag{26}$$

then the condition $\lambda_{t_k}(x^k_{t_k}) \leq \beta$ implies $\lambda_{t_{k+1}}(x^{k+1}_{t_{k+1}}) \leq \beta$. Consequently, the penalty parameter t_k is updated by

$$t_{k+1} := (1-\sigma_\beta)t_k = \left(1 - \frac{c_\beta}{(1+c_\beta)\sqrt{\nu}}\right)t_k, \tag{27}$$

which guarantees that Δ_k satisfies the condition (26), where

$$c_\beta := \frac{1}{2}\left[1 + 0.43\sqrt{\beta} - \sqrt{(1-0.43\sqrt{\beta})^2 + 4\beta}\right] \in (0, 0.044183].$$

In addition, $c^{\max}_\beta := \max\{c_\beta : \beta \in (0, 0.118975]\} = 0.044183$ when $\beta = 0.042231$.

Proof. Under the assumption $\lambda_{t_k}(x^k_{t_k}) \leq \beta$ and the first condition (26) with $\delta_{k+1} \leq 0.066517\beta$, we can obtain from (23) and (24) that $\lambda_{t_{k+1}}(x^{k+1}_{t_{k+1}}) \leq 0.0755\beta + 5\left(\frac{\beta}{1-\Delta_k} + \Delta_k\right)^2$. To guarantee $\lambda_{t_{k+1}}(x^{k+1}_{t_{k+1}}) \leq \beta$, we have

$$0.0755\beta + 5\left(\frac{\beta}{1-\Delta_k} + \Delta_k\right)^2 \leq \beta \quad \Rightarrow \quad \frac{\beta}{1-\Delta_k} + \Delta_k \leq 0.43\sqrt{\beta}.$$

The last condition implies

$$\Delta_k \leq \frac{1}{2}\left[1 + 0.43\sqrt{\beta} - \sqrt{(1-0.43\sqrt{\beta})^2 + 4\beta}\right].$$

This is the second condition of (26), provided that $\beta \in (0, 0.118975]$. The second statement of this theorem follows from (26) and Lemma 5, while the last statement is computed numerically. □

4.4. Stopping criterion.

We require a stopping criterion that guarantees an ε-solution for (1) according to Definition 1. To achieve this, we present the following Lemma; the proof is provided in Appendix 7.6.

LEMMA 6. *Let $\{x^k_{t_k}\}$ be the sequence generated by (21). Then, it holds that*

$$0 \le G(x^{k+1}_{t_{k+1}}) - G^\star \le t_{k+1} \cdot \psi\Big(\nu, \zeta_0, \bar{\lambda}_{t_{k+1}}(x^{k+1}_{t_{k+1}}), \bar{\lambda}_{t_{k+1}}(x^k_{t_k}), \delta_{k+1}\Big), \tag{28}$$

where ψ is defined as

$$\psi(\nu, m_0, \lambda, \lambda_+, \delta) := \nu + \sqrt{\nu}\frac{\lambda_+}{1-\lambda} + \frac{\lambda}{(1-\lambda)^2}(\lambda + \lambda_+ + \delta) + \frac{\delta^2}{2} + m_0\lambda_+, \tag{29}$$

and $m_0 := \eta n_\nu \|\zeta_0\|^*_{\bar{x}^\star_f} = \eta n_\nu \|\nabla f(x^0) + t_0^{-1}(c + \xi_0)\|^*_{\bar{x}^\star_f}$.

Since $\lambda_{t_{k+1}}(x^{k+1}_{t_{k+1}}) \le \beta$ and $\lambda_{t_{k+1}}(x^k_{t_k}) \le 0.43\sqrt{\beta}$, $\delta_{k+1} \le \bar{\delta}$ and $x^0 = x^\star_f$, we can show that

$$\bar{\lambda}_{t_{k+1}}(x^{k+1}_{t_{k+1}}) \le \frac{(1-m_0)\beta}{1-2m_0} + \frac{m_0}{1-m_0} := \gamma_0, \quad \text{and} \quad \bar{\lambda}_{t_{k+1}}(x^k_{t_k}) \le \frac{0.43\sqrt{\beta}(1-m_0)}{1-2m_0} + \frac{m_0}{1-m_0} := \hat{\gamma}_0.$$

By using Lemma 6, we can see that

$$0 \le G(x^{k+1}_{t_{k+1}}) - G^\star \le t_{k+1} \cdot \psi_\beta(\nu), \tag{30}$$

where $\psi_\beta(\nu) := \psi(\nu, m_0, \gamma_0, \hat{\gamma}_0, \bar{\delta})$. Then, if $t_k \cdot \psi_\beta(\nu) \le \varepsilon$, we are guaranteed that $x^{k+1}_{t_{k+1}}$ is a ε-solution and we can terminate our algorithm.

4.5. Overview of our scheme.

We summarize the proposed scheme in Algorithm 1.

Algorithm 1 Single-phase, proximal path-following scheme

```
Input: Tolerance $\varepsilon > 0$.
Initialization:
  1. Compute $x^\star_f$ and set $x^0 := x^\star_f$. Compute a subgradient $\xi_0 \in \partial g(x^0)$.
  2. Compute $c_0 := \|c + \xi_0\|^*_{x^\star_f}$. Choose $\beta \in (0, 0.118975]$ and set $\bar{\delta} := 0.066517\beta$.
  3. Choose $t_0 > \frac{(1-\beta)}{(3+\beta)n_\nu c_0}, \eta := 1$ and compute $m_0$ from Lemma 1.
  4. Set $\gamma_0 := \frac{(1-m_0)\beta}{1-2m_0} + \frac{m_0}{1-m_0}$ and $\hat{\gamma}_0 := \frac{0.43\sqrt{\beta}(1-m_0)}{1-2m_0} + \frac{m_0}{1-m_0}$.
  5. Set $\psi_\beta(\nu) := \psi(\nu, m_0, \gamma_0, \hat{\gamma}_0, \bar{\delta})$.
  6. Set $c_\beta := \frac{1}{2}\left[1 + 0.43\sqrt{\beta} - \sqrt{(1 - 0.43\sqrt{\beta})^2 + 4\beta}\right]$ and $\sigma_\beta := \frac{c_\beta}{(1+c_\beta)\sqrt{\nu}}$.
for $k := 0$ to $k_{\max}$ do
  7. If $t_k\psi_\beta(\nu) \le \varepsilon$, then terminate.
  8. Update $t_{k+1} := (1 - \sigma_\beta)t_k$.
  9. Perform the inexact full-step proximal-Newton iteration by solving
         $x^{k+1}_{t_{k+1}} :\approx \mathop{\mathrm{argmin}}_{x \in \mathrm{int}(\mathcal{X})} \left\{\hat{F}_{t_{k+1}}(x; x^k_{t_k}) := t_{k+1}Q(x; x^k_{t_k}) + G(x)\right\}$
     up to a given accuracy $\delta_{k+1} \le \bar{\delta}$.
end for
```

It is clear that the computational bottleneck of Algorithm 1 lies in Step 9, where we need to approximately solve a strongly convex quadratic composite subproblem. We comment on this step and its solution in Section 5.

The following theorem summarizes the worst-case iteration-complexity of Algorithm 1.

Theorem 3. *Let $\{(x^k,t_k)\}$ be the sequence generated by Algorithm 1. Then, the total number of iterations required to reach an ε-solution x^k of (1) does not exceed*

$$k_{\max} := \left\lfloor \frac{\log\left(\frac{\psi_\beta(\nu)}{t_0\varepsilon}\right)}{-\log(1-\sigma_\beta)} \right\rfloor + 1. \tag{31}$$

Thus, the worst-case iteration-complexity of Algorithm 1 is $\mathcal{O}\left(\sqrt{\nu}\log\left(\frac{\nu}{t_0\varepsilon}\right)\right)$.

Proof. From (27), we can see that $t_k = (1-\sigma_\beta)^k t_0$. Hence, to obtain $0 \le G(x^{k+1}_{t_{k+1}}) - G^\star \le \varepsilon$, using (8), we require $t_k \ge \frac{\psi_\beta(\nu)}{\varepsilon}$, or $k \ge \frac{\log\left(\frac{\psi_\beta(\nu)}{t_0\varepsilon}\right)}{-\log(1-\sigma_\beta)}$. By rounding up this estimate, we obtain $k_{\max}$ as in (31). We note that $-\log(1-\sigma_\beta) = \mathcal{O}(1/\sqrt{\nu})$. In addition, by (29), we have $\psi_\beta(\nu) = \mathcal{O}\left(\nu + t_0^{-1}n_\nu\|c+\xi_0\|^*_{x_f^\star}\right) = \mathcal{O}(\nu)$ due to (16). Hence, the worst-case iteration-complexity of Algorithm 1 is $\mathcal{O}\left(\sqrt{\nu}\log\left(\frac{\nu}{t_0\varepsilon}\right)\right)$. □

We note that the worst-case iteration-complexity stated in Theorem 3 is a global worst-case complexity, which is different from the one in [47]. As already mentioned in the Introduction, in the latter case we require

$$\left\lfloor \frac{F_{t_0}(x^0) - F_{t_0}(x^\star_{t_0})}{\omega\left((1-\kappa)\beta\right)} \right\rfloor$$

iterations in Phase I, for arbitrary selected t_0 and x^0, and $\kappa \in (0,1), \beta \in (0,0.15]$, $\omega(q) = q - \log(1+q)$, *i.e.*, Phase I has a sublinear convergence rate to the initial point $x^0_{t_0}$.

We illustrate the basic idea of our single-phase scheme compared to the two-phase scheme in [47] in Figure 1. Our method follows different central path generated by the solution trajectory of the re-parameterized barrier problem, where an initial point x^0 is immediately available.

4.6. The exact variant. We consider a special case of Algorithm 1, where the subproblem (17) at Step 9 can be solved exactly to obtain $x^{k+1}_{t_{k+1}}$ such that $x^{k+1}_{t_{k+1}} = \bar{x}^{k+1}_{t_{k+1}}$. In this case, we can enlarge the constant c_β in (27) to obtain a better factor σ_β. More precisely, we can use

$$\bar{c}_\beta := \frac{1}{2}\left[1 + 0.45\sqrt{\beta} - \sqrt{(1-0.45\sqrt{\beta})^2 + 4\beta}\right] > c_\beta, \tag{32}$$

where $\beta \in (0, 0.116764]$. Hence, we obtain a faster convergence (up to a constant factor) in this case. For instance, we can numerically check that $\bar{c}^{\max}_\beta := \max\{\bar{c}_\beta : \beta \in (0,0.116764]\} = 0.048186 > c^{\max}_\beta = 0.044183$ with respect to $\beta = 0.045864$.

5. Numerical experiments.

In this section, we first discuss some implementation aspects of Algorithm 1: (*i*) how one can solve efficiently the subproblem in step 9 of Algorithm 1 and, (*ii*) how we can compute the analytical center $\bar{x}^\star_f$. In sequence, we illustrate the merits of our approach via three numerical examples, where we compare with state-of-the-art interior-point algorithms.

Inexact proximal-Newton step. The key step of Algorithm 1 is the proximal Newton direction. This corresponds to solving the following strongly convex quadratic composite problem:

$$\min_{d\in\mathbb{R}^p}\left\{q(d) := \langle h_k, d\rangle + {}^1\!/\!_2\cdot\langle H_k d, d\rangle + g(x^k+d)\right\}, \tag{33}$$

where $x^k, h_k \in \mathbb{R}^p$, and H_k is a symmetric positive definite matrix.

There exist many efficient first-order and proximal quasi-Newton methods to solve (33), see, *e.g.*, [5, 6] for concrete instances of proximal methods, as well as [48, 49] for primal and dual approaches

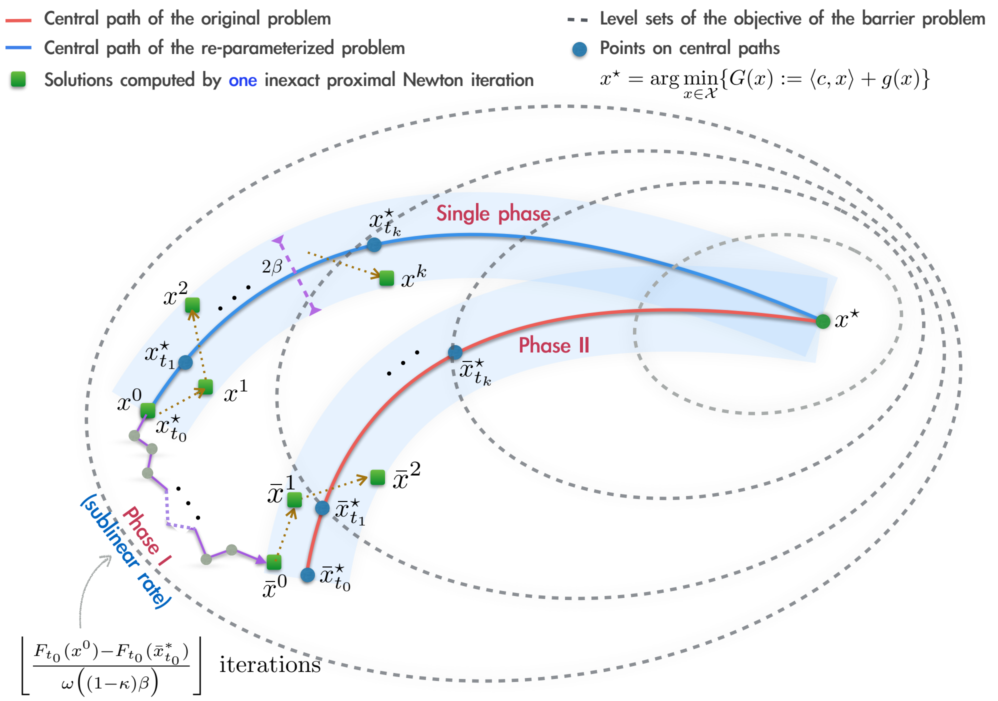
Central path of the original problem
Central path of the re-parameterized problem
Solutions computed by one inexact proximal Newton iteration
Level sets of the objective of the barrier problem
Points on central paths
$x^\star = \arg\min_{x\in\mathcal{X}}\{G(x) := \langle c, x\rangle + g(x)\}$
Single phase
Phase II
Phase I
(sublinear rate)
$x^\star_{t_k}$
2β
x^k
x^2
$x^\star_{t_1}$
x^1
x^0
$x^\star_{t_0}$
$x^\star$
$\bar{x}^\star_{t_k}$
$\bar{x}^1$
$\bar{x}^2$
$\bar{x}^\star_{t_1}$
$\bar{x}^0$
$\bar{x}^\star_{t_0}$
$\left\lfloor \frac{F_{t_0}(x^0)-F_{t_0}(\bar{x}^\star_{t_0})}{\omega\left((1-\kappa)\beta\right)} \right\rfloor$ iterations

FIGURE 1. Illustration of differences between path-following trajectories followed by single-phase Algorithm 1 and two-phase algorithm in [47]. In the latter case and given an initial point, say $x^0 \equiv \bar{x}^\star_f$, [47] first performs $\left\lfloor \frac{F_{t_0}(x^0)-F_{t_0}(x^\star_{t_0})}{\omega((1-\kappa)\beta)} \right\rfloor$ iterations for PHASE I to obtain an initial point $\bar{x}^0$, within the quadratic convergence region of Newton method. Then, the fast convergent PHASE II follows the central path (in red color) towards $x^\star$. Our algorithm avoids the sublinearly convergent PHASE I by properly selecting t_0, η and x^0, and follows a different central path generated by the solution trajectory of the re-parameterized barrier problem (blue curve).

on that matter. The efficiency of such algorithms strongly depends on the computation of prox_g. In addition, since (33) is strongly convex, restart strategies, as in [36, 44] for first order methods, can achieve fast convergence rate. When g is absent, (33) reduces to a positive definite linear system $H_k d = -h_k$, which can be efficiently solved by conjugate gradient scheme or Cholesky methods.

Concluding, we state that problem (33) has special features, which can be exploited in practice:

(i) Often, both matrices H_k and its inverse H_k^{-1} are available, which allow us to estimate both the Lipschitz constant of ∇q and the strong convexity parameter of q. Hence, one can design accelerated gradient methods that have linear convergence rate.

(ii) By following a "warm-start" strategy, *i.e.*, each iteration is initialized with the previously computed estimate, (33) quickly reaches a high accuracy solution in a few iterations.

The analytical center point. To obtain the theoretical complexity bound of Theorem 3, we require the computation of the analytical center $\bar{x}^\star_f$ of the barrier function f. While computing $\bar{x}^\star_f$ might be challenging for some problem instances, there are several practical cases where can be computed analytically. For example, if $f(x) := -\sum_{i=1}^p \log(1-x_i^2)$, *i.e.*, f is the barrier of the box set $\mathcal{X} := \{x \in \mathbb{R}^p : -1 \le x_i \le 1,\ i = 1, \cdots, p\}$, then $\bar{x}^\star_f = \mathbf{0} \in \mathbb{R}^p$. In the case of $f(X) := -\log\det(X) - \log\det(U - X)$ for the set $\mathcal{X} := \{X \in \mathbb{S}^p_+ : 0 \preceq X \preceq U\}$, where $\mathbb{S}^p_+$ denotes the set of positive semi-definite matrices in $p \times p$ dimensions, we have $\bar{X}^\star_f = 0.5U$, where $\bar{X}^\star_f$ denotes the analytical center in a matrix form. In general cases, $\bar{x}^\star_f$ can be computed after a few Newton iterations. More details on computation of $\bar{x}^\star_f$ can be found in [30, 33].

Next, we study three numerical examples. We first compare with the two-phase algorithm in [47]; then, we compare Algorithm 1 with some off-the-shelf interior-point solvers such as SDPT3 [46], SeDuMi [43] and Mosek [2].

5.1. The Max-Cut problem. In this example, we consider the SDP relaxation of the well-known Max-Cut problem as a test case. In particular, consider the following problem:

$$\max_X \left\{ (1/4)\langle L, X\rangle : X \succeq 0,\ \mathrm{diag}(X) = \boldsymbol{e} \right\}, \tag{34}$$

where $X \in \mathbb{S}^p_+$ is the positive semi-definite optimization variable, L is the Laplacian matrix of the corresponding underlying graph of the problem, $\mathrm{diag}(X)$ is the diagonal of X and $\boldsymbol{e} := (1, 1, \cdots, 1)^\top \in \mathbb{R}^p$. The purpose of this section is to compare Algorithm 1 with the two-phase algorithm in [47]. We note that in the latter case, the algorithm is also an inexact proximal interior point method, that follows a two-phase procedure.

If we define $c := -(1/4)L$, $g(X) := \delta_{\mathcal{X}}(X)$, the indicator of the feasible set $\mathcal{X} := \{X \in \mathbb{S}^p_+ : \mathrm{diag}(X) = \boldsymbol{e}\}$, then (34) can be reformulated into (1). In this case, the proximal operator of g is just the projection onto the affine subspace $\mathcal{X}$, which can be computed in a closed form. Moreover, (17) can be solved in a closed from: it requires only one Cholesky decomposition and two matrix-matrix multiplications.

Table 1. Summary of results on the small-sized Max-Cut problems. Here, Error $:= \|X^k - X^\star_{\mathrm{SDPT3}}\|_F / \|X^\star_{\mathrm{SDPT3}}\|_F$ and $f^\star_{\mathrm{SDPT3}}$ denotes the objective value obtained by using IPM solver SDPT3 [46] with high accuracy. For the case of [47], the two quantities in Iters column denote the number of iterations required for Phase I and Phase II, respectively. `g05_n.0` is for unweighted graphs with edge probability 0.5; `pm1s_100.0` is for a weighted graph with edge weights chosen uniformly from $\{-1, 0, 1\}$ and density 0.1; `wd09_100.0` is for a 0.1 density ten graph with integer edge weights chosen from $[-10, 10]$; `t2g20_5555` is for each dimension three two-dimensional toroidal grid graphs with gaussian distributed weights and dimension 20×20; `t3g7_5555` is for each dimension three three-dimensional toroidal grid graphs with gaussian distributed weights and dimension $7 \times 7 \times 7$. In these two last problems, the adjacency matrix A is normalized by $\sqrt{\max |A_{ij}|}$.

			[47]				Algorithm 1			
Name	p	$f^\star_{\mathrm{SDPT3}}$	$f(X)$	Error	Iters	Time[s]	$f(X)$	Error	Iters	Time[s]
g05_60.0	60	-59.00	-58.94	4.35e-03	160/680	0.40	-58.94	4.35e-03	704	0.32
g05_80.0	80	-80.00	-79.92	4.38e-03	292/772	0.63	-79.92	4.39e-03	799	0.48
g05_100.0	100	-100.00	-99.90	4.41e-03	351/877	0.94	-99.90	4.38e-03	910	0.75
pm1s_100.0	100	-52.58	-52.52	3.76e-03	233/1015	1.40	-52.52	3.77e-03	1042	0.85
w09_100.0	100	-80.75	-80.67	4.20e-03	729/968	1.30	-80.67	4.21e-03	996	0.87
t3g7_5555	343	-20620.30	-20616.76	4.45e-03	107/32	2.23	-20599.78	1.99e-03	89	1.30
t2g20_5555	400	-31163.19	-31153.93	1.33e-02	159/99	3.41	-31154.04	1.24e-02	157	2.21

We test both algorithms on 7 small-sized Max-Cut problems generated by `Rudy`[3]. We also consider 4 medium-sized problems from the Gset data set[4], which were also generated from `Rudy`. Both algorithms are tested in Matlab R2015a environment, running on a MacBook Pro. Laptop 2.6GHz Intel Core i7 with 16GB memory. The initial value of t_0 is set at $t_0 := 0.025$ for both cases. We terminate the execution if $|f(X^k) - f^\star_{\mathrm{SDPT3}}| / |f^\star_{\mathrm{SDPT3}}| \leq 10^{-3}$, where $f(X) := -\mathrm{trace}(LX)$.

The results are provided in Tables 1-2. Algorithm 1 outperforms [47] in terms of total computational time, while achieving the same, if not better, solution w.r.t. objective value. We observe the following trade-off w.r.t. the algorithm in [47]: if we increase the initial value of t_0 in [47], then the number of iterations in Phase I is deceasing, but the number of iterations in Phase II is increasing. We emphasize that both algorithms use the worst-case update rule without any line-search on the step-size as in off-the-shelf solvers.

[3] `http://biqmac.uni-sklu.ac.at/biqmaclib`.

[4] `http://www.cise.ufl.edu/research/sparse/matrices/Gset/index.html`.

Table 2. Summary of results on the medium-sized Max-Cut problems. Here, Error $:= \|X^k - X^\star_{\text{SDPT3}}\|_F / \|X^\star_{\text{SDPT3}}\|_F$ and $f^\star_{\text{SDPT3}}$ denotes the objective value obtained by using IPM solver SDPT3 [46] with high accuracy. For the case of [47], the two quantities in Iters column denote the number of iterations required for Phase I and Phase II, respectively. Each problem Gxx is sparse with %1 to %3 upper triangle nonzero, binary entries.

			[47]				Algorithm 1			
Name (Gxx)	p	$f^\star_{\text{SDPT3}}$	$f(X)$	Error	Iters	Time[s]	$f(X)$	Error	Iters	Time[s]
G01	800	-12080.12	-12080.12	1.46e-02	149/805	104.48	-12080.13	1.46e-02	569	62.43
G43	1000	-7029.29	-7029.30	2.03e-02	153/1031	208.82	-7029.30	2.03e-02	712	143.00
G22	2000	-14116.01	-14116.03	3.57e-02	215/623	805.99	-14116.06	3.57e-02	561	741.90
G48	3000	-5998.57	-5998.57	1.38e-02	225/2893	8487.08	-5998.59	1.40e-02	1978	7978.35

5.2. The Max-k-Cut problem.

Here, we consider the SDP relaxation of the Max-k-Cut problem, proposed in [14, eq. (3)]:

$$\max_X \left\{ \frac{k-1}{2k} \langle L, X \rangle : X \succeq 0,\ \operatorname{diag}(X) = \boldsymbol{e},\ X \geq -\frac{1}{k-1} E_p \right\}, \tag{35}$$

where L is the Laplacian matrix of the corresponding graph, $\boldsymbol{e} := (1, 1, \cdots, 1)^T$, and E_p is the $p \times p$ all-ones matrix. Observe that $X \geq Y$, for two matrices X, Y, correspond to entry-wise inequality. Similarly to (34), if we define $c := -\frac{(k-1)}{2k} L$ and $g(X) := \delta_{\mathcal{X}}(X)$ with $\mathcal{X} := \left\{ X \in \mathbb{S}^p_+ : \operatorname{diag}(X) = \boldsymbol{e},\ X \geq -\frac{1}{k-1} E_p \right\}$, (35) is a special instance of the class of problems described by (1).

We compare Algorithm 1 with three well-established, off-the-shelf interior-point solvers: SDPT3 [46], SeDuMi [43][5], and Mosek [2][6]. We consider synthetically generated p-node graphs, where each edge is generated from a Bern($1/4, 3/4$) probability distribution; we also set $k = 4$. The parameters of Algorithm 1 are set as in the previous example, and all algorithms are terminated if $|f(X^k) - f^\star| / |f^\star| \leq 10^{-5}$, where $f^\star$ is the best optimal value produced by three off-the-shelf solvers. We solve (17) with a fast projected gradient method, with adaptive restart and a warm-start strategy [44]: Such configuration requires few iterations to achieve our desired accuracy $\delta = 2.8 \times 10^{-2}$ or higher.

Table 3. Comparison results on the Max-k-Cut problem. Here, `Iters` is the number of iterations; Time[s] is the computational time in second; $f(X) = -\text{trace}(LX)$; `svars` is the number of slack variables; and `cnstr` is the number of linear constraints. In addition, we have $p(p+1)/2$ variables in X and one SDP constraint.

Size	Lifting		Algorithm 1		SeDuMi		SDPT3		Mosek	
p	`svars`	`cnstr`	$f(X)$	Time[s]	$f(X)$	Time[s]	$f(X)$	Time[s]	$f(X)$	Time[s]
50	1,225	1,275	-87.733	7.32	-86.174	4.76	-86.160	**2.02**	-86.138	3.84
75	2,775	2,850	-166.237	9.80	-166.236	55.41	-166.214	10.76	-166.214	**8.91**
100	4,950	5,050	-316.741	**18.37**	-316.746	732.16	-316.709	48.67	-316.653	26.63
150	11,175	11,325	-654.703	**73.63**	-654.684	5,121.34	-654.539	484.46	-654.673	366.36
200	19,900	20,100	-1185.784	**169.08**	-1185.783	25,521.39	-1185.760	2,122.91	-1185.647	2,048.95

Table 3 contains some experimental results. Observe that, if p is small, all algorithms perform well with the off-the-shelf solvers returning faster a good solution. However, when p increases, their computational time significantly increases, as compared to Algorithm 1. One reason that this happens is that standard SDP solvers require $p(p-1)/2$ slack variables and $p(p-1)/2$ additional linear

[5] Both implementations include Matlab and optimized C-coded parts.

[6] Available for academic use at `https://mosek.com`.

constraints, in order to process the component-wise inequality constraints. Such reformulation of the problem significantly increases variable and constraint size and, hence, lead slower execution. In stark contrast, Algorithm 1 handles both linear and inequality constraints by a simple projection, which requires only $p(p+1)/2$ basic operations. Figure 2 graphically illustrates the scalability of the four algorithms under comparison, based on the results contained in Table 3.

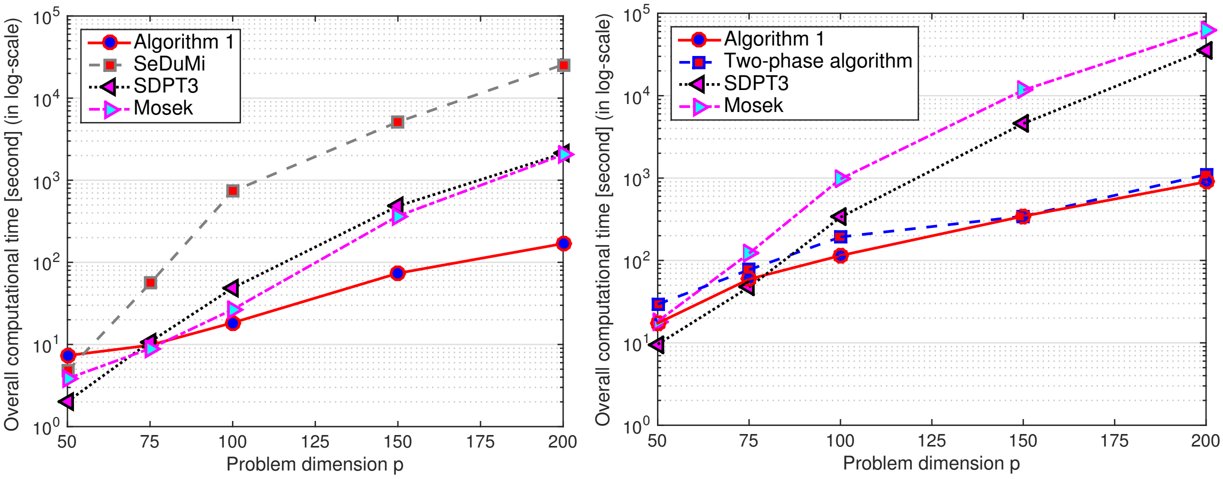

FIGURE 2. Overall execution time, as a function of problem dimension. **Left panel**: MAX-k-CUT problem (35); **Right panel**: Clustering problem (36).

5.3. Max-norm clustering. In this last problem case, we consider the max-norm clustering task [20], where we seek a clustering matrix K that minimizes the disagreement with a given affinity matrix A:

$$\begin{array}{ll} \displaystyle\min_{x:=[L,R,K]\in\mathbb{R}^{p\times p}} & \|\mathrm{vec}(K-A)\|_1 \\ \text{s.t.} & \mathcal{Q}(x) := \begin{bmatrix} L & K \\ K^T & R \end{bmatrix} \succeq 0,\ L_{ii}\leq 1,\ R_{ii}\leq 1,\ i=1,\cdots,p. \end{array} \tag{36}$$

Here, vec is the vectorization operator of a matrix (*i.e.*, $\mathrm{vec}(X) := (X_1^T,\cdots,X_n^T)^T$, where X_i is the i-th column of X). Note that (36) is an SDP convex relaxation to the *correlation clustering* problem; see [20] for details. While (36) comes with rigorous theoretical guarantees and can be formulated as a standard conic program, we need to add $O(p^2)$ slack variables to process the ℓ_1-norm term and the linear constraints. Moreover, the scaling factors (*e.g.*, the Nesterov-Todd scaling factor regarding the semidefinite cone [34]) can create memory bottlenecks in practice, by destroying the sparsity of the underlying problem (*e.g.*, by leading to dense KKT matrices in the Newton systems).

Here, we solve (36) using our path-following scheme. In particular, by defining $x := \mathrm{vec}([K,L,R]$, $f(x) := -\log\det(\mathcal{Q}(x))$ and $g(x) := \|\mathrm{vec}(K-A)\|_1 + \delta_{\mathcal{C}}(x)$, we can transform (36) into (1), where $\delta_{\mathcal{C}}$ is the indicator function of $\mathcal{C} := \{x : L_{ii}\leq 1,\ R_{ii}\leq 1,\ i=1,\cdots,p\}$.

We compare the following solvers: Algorithm 1, the two-phase algorithm in [47], SDPT3 and Mosek. The initial penalty parameter t_0 is set to $t_0 := 0.25$ and the relative tolerance is fixed at 10^{-4} for all algorithms. The data is generated as suggested in [20, 47]. The results of 5 test problem instances are shown in Table 4 sizes p ranging from 50 to 200[7].

[7] Since SDPT3 and Mosek cannot run for bigger problems in our personal computer, we restrict to problem sizes up to $p = 200$.

Table 4. The performance of Algorithm 1, as compared to three methods on the clustering problem (36). Here, Time[s] is the computational time in second; $f(X) = \|\mathrm{vec}(K - A)\|_1$, and $s\%$ is the sparsity of $K - A$.

Size	Algorithm 1			[47]			SDPT3		Mosek	
p	$f(X)$	Time[s]	s%	$f(X)$	Time[s]	s%	$f(X)$	Time[s]	$f(X)$	Time[s]
50	563.90	17.30	49%	563.90	29.38	49.5%	563.86	9.60	563.92	18.27
75	1,308.19	59.30	43.8%	1,308.18	77.05	43.9%	1,308.15	47.40	1,308.32	121.74
100	2,228.62	114.59	35.8%	2,228.61	192.79	35.9%	2,228.59	334.76	2,228.78	975.10
150	5,328.12	344.29	42.4%	5,327.99	344.32	42.5%	5,327.84	4,584.03	5,328.14	11,665.52
200	9,883.92	899.10	45.8%	9,883.81	1,102.97	47.9%	9,883.68	35,974.60	9,884.21	62,835.42

Both SDPT3 and Mosek are approximately 40 and 60 times slower than Algorithm 1 and [47], especially when $p > 100$. We note that such solvers require $p^2 + 2p$ slack variables, p^2 additional second order cone constraints and $2p$ additional linear constraints to reformulate (36) into a standard SDP problem. Hence, the size of the resulting SDP problem is much larger than of the original one in (36). We also see that Algorithm 1 is faster than the two-phase algorithm, in terms of total execution time.

We note that SDPT3 gives a slightly better objective value than Algorithm 1. However, its solution K is fully dense, in contrast to those of Algorithm 1 and [47], reducing its interpretation in applications. Figure 2 (right) also reveals the scalability of these four algorithms for solving (36).

6. Conclusions.

In this work, we propose a new path-following framework for a, possibly non-smooth, constrained convex minimization template, which includes linear programming as a special case. For our framework, we assume that the constraints in the optimization template are endowed with a self-concordant barrier and, the non-smooth term has a tractable proximity operator. Our workhorse is a new re-parameterization of the optimality condition of the convex optimization problem, which allows us to select a different central path towards $x^\star$, without relying on the sublinear convergent Phase I of proximal path-following approaches, as in [47].

We illustrate that the new scheme retains the same global, worst-case, iteration-complexity with standard approaches [30, 33]. Moreover, we theoretically show that inexact solutions to subproblems do not sacrifice the worst-case complexity, when controlled appropriately. Finally, we numerically illustrate the effectiveness of our framework on Max-Cut and clustering problems, where the proximal operator play a key role in space efficient optimization.

Acknowledgments: We would like to thank Yurii Nesterov for useful discussions on the initialization technique used in this work. We also thank Cong Bang Vu for his overall suggestions on the final version of this manuscript. This work was supported in part by ERC Future Proof, SNF 200021-146750 and SNF CRSII2-147633.

7. Appendix: proofs of main results.

This section contains proofs of technical results, presented in the main text.

7.1. Proof of Lemma 1.

Let $x^\star_t$ be the solution of (10) and $\bar{x}^\star_t$ be the solution of (7). By the optimality conditions in (8) and (11), we have $-t\nabla f(\bar{x}^\star_t) \in \partial G(\bar{x}^\star_t)$ and $-t\nabla h_\eta(x^\star_t) \in \partial G(x^\star_t)$. Moreover, by the convexity of G, we have $\langle \nabla f(\bar{x}^\star_t) - \nabla h_\eta(x^\star_t), x^\star_t - \bar{x}^\star_t\rangle \geq 0$. Using the definition $\nabla h_\eta(x) := t\nabla f(x) - \eta\zeta_0$, the last inequality leads to

$$\langle \nabla f(x^\star_t) - \nabla f(\bar{x}^\star_t), x^\star_t - \bar{x}^\star_t\rangle \leq \eta\langle \zeta_0, x^\star_t - \bar{x}^\star_t\rangle.$$

Further, by [30, Theorem 4.1.5] and the Cauchy-Schwarz inequality, this inequality implies

$$\frac{\|x^\star_t - \bar{x}^\star_t\|_{\bar{x}^\star_t}}{1 + \|x^\star_t - \bar{x}^\star_t\|_{\bar{x}^\star_t}} \leq \eta\|\zeta_0\|^*_{\bar{x}^\star_t} \Longrightarrow \bar{\Delta}_t \leq \frac{\bar{m}_0}{1 - \bar{m}_0}, \tag{37}$$

which completes the proof of this Lemma.

For the Corollary 1, one observes [30, Corollary 4.2.1] that $\|\zeta_0\|^*_{\bar{x}^\star_t} \leq n_\nu \|\zeta_0\|^*_{\bar{x}^\star_f}$, where $\bar{x}^\star_f$ is the analytical center of f. Following the same motions, one can easily obtain (12). □

7.2. Proof of Lemma 2. By definition of the local norm $\bar{\lambda}_t(x)$, we have:

$$\begin{aligned}\bar{\lambda}_t(x) &= \langle \nabla^2 f(\bar{x}^\star_t)(x - \bar{x}^\star_t), x - \bar{x}^\star_t\rangle^{1/2}\\ &\leq \langle \nabla^2 f(\bar{x}^\star_t)(x^\star_t - \bar{x}^\star_t), x^\star_t - \bar{x}^\star_t\rangle^{1/2} + \langle \nabla^2 f(\bar{x}^\star_t)(x - x^\star_t), x - x^\star_t\rangle^{1/2}\\ &\leq \bar{\Delta}_t + \left(1 - \|x^\star_t - \bar{x}^\star_t\|_{\bar{x}^\star_t}\right)^{-1} \langle \nabla^2 f(x^\star_t)(x - x^\star_t), x - x^\star_t\rangle^{1/2}\\ &= \bar{\Delta}_t + \frac{\lambda_t(x)}{1 - \bar{\Delta}_t}.\end{aligned}$$

Here, in the first inequality, we use the triangle inequality for the weighted norm $\|\cdot\|_{\nabla^2 f(x^\star_{t_{k+1}})}$, while in the second inequality we apply [30, Theorem 4.1.6]. The proof is completed when we use (12) to upper bound the above inequality. □

7.3. Proof of Lemma 3. Since $x^\star_{t_0}$ is the solution of (10) at $t = t_0$, there exists $\xi^\star_{t_0} \in \partial g(x^\star_{t_0})$ such that: $t_0 h_\eta(x^\star_{t_0}) + c + \xi^\star_{t_0} = 0$. Hence,

$$(\xi_0 - \xi^\star_{t_0}) = r_{t_0,\eta}(x^0) - t_0[\nabla f(x^0) - \nabla f(x^\star_{t_0})].$$

By convexity of g, we have

$$0 \leq \langle \xi_0 - \xi^\star_{t_0}, x^0 - x^\star_{t_0}\rangle = \langle r_{t_0,\eta}(x^0) - t_0[\nabla f(x^0) - \nabla f(x^\star_{t_0})], x^0 - x^\star_{t_0}\rangle$$

This inequality leads to $t_0\langle \nabla f(x^0) - \nabla f(x^\star_{t_0}), x^0 - x^\star_{t_0}\rangle \leq \langle r_{t_0,\eta}(x^0), x^0 - x^\star_{t_0}\rangle$. Using the self-concordance of f in [30, Theorem 4.1.7] and the Cauchy-Schwarz inequality, we can derive

$$\begin{aligned}\frac{t_0\lambda_{t_0}(x^0)^2}{1 + \lambda_{t_0}(x^0)} &\leq t_0\langle \nabla f(x^0) - \nabla f(x^\star_{t_0}), x^0 - x^\star_{t_0}\rangle\\ &\leq \langle r_{t_0,\eta}(x^0), x^0 - x^\star_{t_0}\rangle\\ &\leq \|r_{t_0,\eta}(x^0)\|^*_{x^\star_{t_0}} \lambda_{t_0}(x^0).\end{aligned}$$

Hence, $\frac{t_0\lambda_{t_0}(x^0)}{1+\lambda_{t_0}(x^0)} \leq \|r_{t_0,\eta}(x^0)\|^*_{x^\star_{t_0}}$. Moreover, by [30, Theorem 4.1.6], we have $\|r_{t_0,\eta}(x^0)\|^*_{x^\star_{t_0}} \leq \frac{\|r_{t_0,\eta}(x^0)\|^*_{x^0}}{1-\lambda_{t_0}(x^0)}$. Combining these two inequalities, we obtain

$$\frac{\lambda_{t_0}(x^0)(1 - \lambda_{t_0}(x^0))}{1 + \lambda_{t_0}(x^0)} \leq t_0^{-1}\|r_{t_0,\eta}(x^0)\|^*_{x^0}$$

After few elementary calculations, we can easily show that if $\|r_{t_0,\eta}(x^0)\|^*_{x^0} < t_0(3 - 2\sqrt{2})$, then we obtain (14), which also guarantees its right-hand side to be positive. □

7.4. Proof of Lemma 4. Let $\lambda_{t_{k+1}}(x^{k+1}_{t_{k+1}})$ and $\lambda_{t_{k+1}}(x^k_{t_k})$ be defined by (22). It was proved in [47, Theorem 3.3] that

$$\lambda_{t_{k+1}}(x^{k+1}_{t_{k+1}}) \leq \frac{\delta_k}{1 - \lambda_{t_{k+1}}(x^k_{t_k})} + \left(\frac{3 - 2\lambda_{t_{k+1}}(x^k_{t_k})}{1 - 4\lambda_{t_{k+1}}(x^k_{t_k}) + 2\lambda_{t_{k+1}}(x^k_{t_k})^2}\right)\lambda_{t_{k+1}}(x^k_{t_k})^2, \tag{38}$$

where $\lambda_{t_{k+1}}(x^k_{t_k}) < 1 - 1/\sqrt{2}$. Now, we consider the function $m(t) := \frac{3-2t}{1-4t+2t^2}$ for $t \in [0, 1 - 1/\sqrt{2})$. We can numerically check that if $t \in [0, 0.118975]$ then $m(t) \leq 5$. In this case, we also have $\frac{1}{1-t} \leq$

1.135042. Using these upper bounds into (38), we obtain $\lambda_{t_{k+1}}(x^{k+1}_{t_{k+1}}) \le 1.135042\delta_k + 5\lambda_{t_{k+1}}(x^k_{t_k})^2$, which is exactly (23), whenever $\lambda_{t_{k+1}}(x^k_{t_k}) \in [0, 0.118975]$.

The proof of the estimate (23) can be found in [47]. We only prove (24). We note that

$$\begin{aligned}\lambda_{t_{k+1}}(x^k_{t_k}) &= \langle \nabla^2 f(x^\star_{t_{k+1}})(x^k_{t_k} - x^\star_{t_{k+1}}), x^k_{t_k} - x^\star_{t_{k+1}}\rangle^{1/2}\\ &\le \langle \nabla^2 f(x^\star_{t_{k+1}})(x^\star_{t_k} - x^\star_{t_{k+1}}), x^\star_{t_k} - x^\star_{t_{k+1}}\rangle^{1/2} + \langle \nabla^2 f(x^\star_{t_{k+1}})(x^k_{t_k} - x^\star_{t_k}), x^k_{t_k} - x^\star_{t_k}\rangle^{1/2}\\ &\le \Delta_k + \left(1 - \|x^\star_{t_k} - x^\star_{t_{k+1}}\|_{x^\star_{t_{k+1}}}\right)^{-1}\langle \nabla^2 f(x^\star_{t_k})(x^k_{t_k} - x^\star_{t_k}), x^k_{t_k} - x^\star_{t_k}\rangle^{1/2}\\ &= \Delta_k + \frac{\lambda_{t_k}(x^k_{t_k})}{1 - \Delta_k},\end{aligned}$$

where is indeed (24). Here, in the first inequality, we use the triangle inequality for the weighted norm $\|\cdot\|_{\nabla^2 f(x^\star_{t_{k+1}})}$, while in the second inequality we apply [30, Theorem 4.1.6]. □

7.5. Proof of Lemma 5. Since $x^\star_{t_k}$ and $x^\star_{t_{k+1}}$ are the solutions of (11) at $t = t_k$ and t_{k+1}, respectively, we have

$$0 \in t_k \nabla h_\eta(x^\star_{t_k}) + \partial G(x^\star_{t_k}) \quad \text{and} \quad 0 \in t_{k+1}\nabla h_\eta(x^\star_{t_{k+1}}) + \partial G(x^\star_{t_{k+1}}).$$

Hence, there exist $v^\star_{t_k} \in \partial G(x^\star_{t_k})$ and $v^\star_{t_{k+1}} \in \partial G(x^\star_{t_{k+1}})$ such that $v^\star_{t_k} = -t_k\nabla h_\eta(x^\star_{t_k})$ and $v^\star_{t_{k+1}} = -t_{k+1}\nabla h_\eta(x^\star_{t_{k+1}})$. Using the convexity of G, we have

$$\langle t_{k+1}\nabla h_\eta(x^\star_{t_{k+1}}) - t_k\nabla h_\eta(x^\star_{t_k}), x^\star_{t_{k+1}} - x^\star_{t_k}\rangle = -\langle v^\star_{t_{k+1}} - v^\star_{t_k}, x^\star_{t_{k+1}} - x^\star_{t_k}\rangle \le 0.$$

Hence, we can show that

$$\langle t_{k+1}\nabla h_\eta(x^\star_{t_{k+1}}) - t_k\nabla h_\eta(x^\star_{t_k}), x^\star_{t_{k+1}} - x^\star_{t_k}\rangle \le 0. \tag{39}$$

By the definition ∇h_η and the update rule (21) of t_k, we have

$$t_{k+1}\nabla h_\eta(x^\star_{t_{k+1}}) - t_k\nabla h_\eta(x^\star_{t_k}) = t_k[\nabla f(x^\star_{t_{k+1}}) - \nabla f(x^\star_{t_k})] + d_k\nabla f(x^\star_{t_{k+1}}). \tag{40}$$

Combining (39) and (40), then using [30, Theorem 4.1.7], the Cauchy-Schwarz inequality, and the definition of Δ_k in (22) we obtain

$$\begin{aligned}\frac{t_k\Delta_k^2}{1+\Delta_k} &\le t_k\langle \nabla f(x^\star_{t_{k+1}}) - \nabla f(x^\star_{t_k}), x^\star_{t_{k+1}} - x^\star_{t_k}\rangle\\ &\le -d_k\langle \nabla f(x^\star_{t_{k+1}}), x^\star_{t_{k+1}} - x^\star_{t_k}\rangle\\ &\le |d_k|\,\|\nabla f(x^\star_{t_{k+1}})\|^*_{x^\star_{t_{k+1}}}\,\Delta_k,\end{aligned} \tag{41}$$

which implies the first inequality of (25). The second inequality of (25) follows from the fact that $\|\nabla f(x^\star_{t_{k+1}})\|^*_{x^\star_{t_{k+1}}} \le \sqrt{\nu}$ due to [30, formula 2.4.2]. The last statement of this lemma is a direct consequence of (25). □

7.6. Proof of Lemma 6. By (20) and given $\hat{F}_{t_{k+1}}(x^{k+1}_{t_{k+1}}; x^k_{t_k}) - \hat{F}_{t_{k+1}}(\bar{x}^{k+1}_{t_{k+1}}; x^k_{t_k}) \le \frac{t_{k+1}\delta^2_{k+1}}{2}$, we have

$$\begin{aligned}G(x^{k+1}_{t_{k+1}}) &\le G(\bar{x}^{k+1}_{t_{k+1}}) + t_{k+1}Q_k(\bar{x}^{k+1}_{t_{k+1}}; x^k_{t_k}) - t_{k+1}Q_k(x^{k+1}_{t_{k+1}}; x^k_{t_k}) + \frac{t_{k+1}\delta^2_{k+1}}{2}\\ &= G(\bar{x}^{k+1}_{t_{k+1}}) + t_{k+1}\langle \nabla f(x^k_{t_k}), \bar{x}^{k+1}_{t_{k+1}} - x^{k+1}_{t_{k+1}}\rangle - t_{k+1}\eta\langle \zeta_0, \bar{x}^{k+1}_{t_{k+1}} - x^{k+1}_{t_{k+1}}\rangle\\ &\quad + \frac{t_{k+1}}{2}\left(\|\bar{x}^{k+1}_{t_{k+1}} - x^k_{t_k}\|^2_{x^k_{t_k}} - \|x^{k+1}_{t_{k+1}} - x^k_{t_k}\|^2_{x^k_{t_k}}\right) + \frac{t_{k+1}\delta^2_{k+1}}{2}.\end{aligned} \tag{42}$$

Now, since $\bar{x}^{k+1}_{t_{k+1}}$ is the exact solution of (17), there exists $\bar{v}^{k+1} \in \partial G(\bar{x}^{k+1}_{t_{k+1}})$ such that

$$\bar{v}^{k+1} = -t_{k+1}\big(\nabla f(x^k_{t_k}) - \eta\zeta_0\big) - t_{k+1}\nabla^2 f(x^k_{t_k})(\bar{x}^{k+1}_{t_{k+1}} - x^k_{t_k}). \tag{43}$$

Next, using the convexity of G, with $\bar{v}^{k+1} \in \partial G(\bar{x}^{k+1}_{t_{k+1}})$, we have

$$\begin{aligned}
G(\bar{x}^{\star}_{t_{k+1}}) - G(\bar{x}^{k+1}_{t_{k+1}}) &\geq \langle \bar{v}^{k+1}, \bar{x}^{\star}_{t_{k+1}} - \bar{x}^{k+1}_{t_{k+1}}\rangle \\
&= -t_{k+1}\langle \nabla f(x^k_{t_k}), \bar{x}^{\star}_{t_{k+1}} - \bar{x}^{k+1}_{t_{k+1}}\rangle + t_{k+1}\eta_0\langle \zeta_0, \bar{x}^{\star}_{t_{k+1}} - \bar{x}^{k+1}_{t_{k+1}}\rangle \\
&\quad - t_{k+1}\langle \nabla^2 f(x^k_{t_k})(\bar{x}^{k+1}_{t_{k+1}} - x^k_{t_k}), \bar{x}^{\star}_{t_{k+1}} - \bar{x}^{k+1}_{t_{k+1}}\rangle
\end{aligned} \tag{44}$$

Summing up (42) and (44), and rearranging the result, we can derive

$$\begin{aligned}
G(\bar{x}^{\star}_{t_{k+1}}) - G(x^{k+1}_{t_{k+1}}) &\geq -t_{k+1}\langle \nabla f(x^k_{t_k}), \bar{x}^{\star}_{t_{k+1}} - \bar{x}^{k+1}_{t_{k+1}}\rangle + t_{k+1}\eta\langle \zeta_0, x^{\star}_{t_{k+1}} - \bar{x}^{k+1}_{t_{k+1}}\rangle \\
&\quad - t_{k+1}\langle \nabla^2 f(x^k_{t_k})(\bar{x}^{k+1}_{t_{k+1}} - x^k_{t_k}), \bar{x}^{\star}_{t_{k+1}} - \bar{x}^{k+1}_{t_{k+1}}\rangle \\
&\quad - \frac{t_{k+1}}{2}\left(\|\bar{x}^{k+1}_{t_{k+1}} - x^k_{t_k}\|^2_{x^k_{t_k}} - \|x^{k+1}_{t_{k+1}} - x^k_{t_k}\|^2_{x^k_{t_k}}\right) - \frac{t_{k+1}\delta^2_{k+1}}{2} \\
&\quad - t_{k+1}\langle \nabla f(x^k_{t_k}), \bar{x}^{k+1}_{t_{k+1}} - x^{k+1}_{t_{k+1}}\rangle + t_{k+1}\eta\langle \zeta_0, \bar{x}^{k+1}_{t_{k+1}} - x^{k+1}_{t_{k+1}}\rangle \\
&= -t_{k+1}\langle \nabla f(x^k_{t_k}), \bar{x}^{\star}_{t_{k+1}} - x^{k+1}_{t_{k+1}}\rangle + t_{k+1}\eta\langle \zeta_0, \bar{x}^{\star}_{t_{k+1}} - x^{k+1}_{t_{k+1}}\rangle \\
&\quad - t_{k+1}\langle \nabla^2 f(x^k_{t_k})(\bar{x}^{k+1}_{t_{k+1}} - x^k_{t_k}), \bar{x}^{\star}_{t_{k+1}} - \bar{x}^{k+1}_{t_{k+1}}\rangle \\
&\quad - \frac{t_{k+1}}{2}\left(\|\bar{x}^{k+1}_{t_{k+1}} - x^k_{t_k}\|^2_{x^k_{t_k}} - \|x^{k+1}_{t_{k+1}} - x^k_{t_k}\|^2_{x^k_{t_k}}\right) - \frac{t_{k+1}\delta^2_{k+1}}{2}.
\end{aligned} \tag{45}$$

Now, by using the Cauchy-Schwarz inequality, we can further estimate (45) as

$$\begin{aligned}
G(\bar{x}^{\star}_{t_{k+1}}) - G(x^{k+1}_{t_{k+1}}) &\geq -t_{k+1}\|\nabla f(x^k_{t_k})\|^{*}_{\bar{x}^{\star}_{t_{k+1}}}\|\bar{x}^{\star}_{t_{k+1}} - x^{k+1}_{t_{k+1}}\|_{\bar{x}^{\star}_{t_{k+1}}} \\
&\quad - t_{k+1}\,|\eta|\,\|\zeta_0\|^{*}_{\bar{x}^{\star}_{t_{k+1}}}\|\bar{x}^{\star}_{t_{k+1}} - x^{k+1}_{t_{k+1}}\|_{\bar{x}^{\star}_{t_{k+1}}} \\
&\quad - t_{k+1}\langle \nabla^2 f(x^k_{t_k})(\bar{x}^{k+1}_{t_{k+1}} - x^k_{t_k}), \bar{x}^{\star}_{t_{k+1}} - \bar{x}^{k+1}_{t_{k+1}}\rangle \\
&\quad - \frac{t_{k+1}}{2}\left(\|\bar{x}^{k+1}_{t_{k+1}} - x^k_{t_k}\|^2_{x^k_{t_k}} - \|x^{k+1}_{t_{k+1}} - x^k_{t_k}\|^2_{x^k_{t_k}}\right) - \frac{t_{k+1}\delta^2_{k+1}}{2}.
\end{aligned} \tag{46}$$

We consider the term

$$\mathcal{T}_{[1]} := \|\bar{x}^{k+1}_{t_{k+1}} - x^k_{t_k}\|^2_{x^k_{t_k}} - \|x^{k+1}_{t_{k+1}} - x^k_{t_k}\|^2_{x^k_{t_k}} + 2\langle \nabla^2 f(x^k_{t_k})(\bar{x}^{k+1}_{t_{k+1}} - x^k_{t_k}), \bar{x}^{\star}_{t_{k+1}} - \bar{x}^{k+1}_{t_{k+1}}\rangle.$$

Similarly to the proof of [47, Lemma 5.1], we can show that

$$\mathcal{T}_{[1]} \leq \frac{2\bar{\lambda}_{t_{k+1}}(x^k_{t_k})}{(1 - \bar{\lambda}_{t_{k+1}}(x^k_{t_k}))^2}\left(\bar{\lambda}_{t_{k+1}}(x^k_{t_k}) + \bar{\lambda}_{t_{k+1}}(x^{k+1}_{t_{k+1}}) + \delta_{k+1}\right). \tag{47}$$

Next, by using the self-concordance of f and the definition of $\lambda_t(x)$, we have

$$\big(1 - \bar{\lambda}_{t_{k+1}}(x^k_{t_k})\big)^2\nabla^2 f(\bar{x}^{\star}_{t_{k+1}}) \preceq \nabla^2 f(x^k_{t_k}) \preceq \big(1 - \bar{\lambda}_{t_{k+1}}(x^k_{t_k})\big)^{-2}\nabla^2 f(\bar{x}^{\star}_{t_{k+1}}). \tag{48}$$

On the one hand, using (48) and $\|\nabla f(x^k_{t_k})\|^{*}_{x^k_{t_k}} \leq \sqrt{\nu}$, we easily get

$$\|\nabla f(x^k_{t_k})\|^{*}_{\bar{x}^{\star}_{t_{k+1}}} \leq (1 - \bar{\lambda}_{t_{k+1}}(x^k_{t_k}))^{-1}\|\nabla f(x^k_{t_k})\|^{*}_{x^k_{t_k}} \leq (1 - \bar{\lambda}_{t_{k+1}}(x^k_{t_k}))^{-1}\sqrt{\nu}. \tag{49}$$

On the other hand, by [30, Corollary 4.1.7], we can show that

$$\|\zeta_0\|^*_{\bar{x}^\star_{t_{k+1}}} \leq n_\nu \|\zeta_0\|^*_{x^\star_f} := n_\nu p_0. \tag{50}$$

Substituting (48), (49) and (50) into (46), we finally obtain

$$\begin{aligned} G(\bar{x}^\star_{t_{k+1}}) - G(x^{k+1}_{t_{k+1}}) \geq -t_{k+1}\Bigg(& \frac{\sqrt{\nu}\bar{\lambda}_{t_{k+1}}(x^{k+1}_{t_{k+1}})}{(1-\bar{\lambda}_{t_{k+1}}(x^k_{t_k}))^2} + \eta n_\nu p_0 \bar{\lambda}_{t_{k+1}}(x^{k+1}_{t_{k+1}}) \\ & + \frac{\bar{\lambda}_{t_{k+1}}(x^k_{t_k})}{(1-\bar{\lambda}_{t_{k+1}}(x^k_{t_k}))^2}\Big(\bar{\lambda}_{t_{k+1}}(x^k_{t_k}) + \bar{\lambda}_{t_{k+1}}(x^{k+1}_{t_{k+1}}) + \delta_{k+1}\Big) + \frac{\delta^2_{k+1}}{2}\Bigg), \end{aligned}$$

which is (28) by combing with $G^\star - G(\bar{x}^\star_{t_{k+1}}) \geq -\nu t_{k+1}$. The remaining statements of this lemma are proved as [47, Lemma 5.1]. □

References

[1] M. S. Andersen, J. Dahl, Z. Liu, and L. Vandenberghe. Interior-point methods for large-scale cone programming. *In: S. Sra, S. Nowozin, S. J. Wright (ed.) Optimization for Machine Learning*, MIT Press:55–83, 2011.

[2] MOSEK ApS. *The MOSEK optimization toolbox for MATLAB manual. Version 7.1 (Revision 28).*, 2015.

[3] Luca Baldassarre, Nirav Bhan, Volkan Cevher, Anastasios Kyrillidis, and Siddhartha Satpathi. Group-sparse model selection: Hardness and relaxations. *arXiv preprint arXiv:1303.3207*, 2013.

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

[5] A. Beck and M. Teboulle. A fast iterative shrinkage-thresholding agorithm for linear inverse problems. *SIAM J. Imaging Sci.*, 2(1):183–202, 2009.

[6] S. Becker, E. J. Candès, and M. Grant. Templates for convex cone problems with applications to sparse signal recovery. *Math. Program. Compt.*, 3(3):165–218, 2011.

[7] A. Ben-Tal and A. Nemirovski. *Lectures on modern convex optimization: Analysis, algorithms, and engineering applications*, volume 3 of *MPS/SIAM Series on Optimization*. SIAM, 2001.

[8] Dimitri P. Bertsekas. *Constrained Optimization and Lagrange Multiplier Methods.* Athena Scientific, 1996.

[9] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends in Machine Learning*, 3(1):1–122, 2011.

[10] S. Boyd and L. Vandenberghe. *Convex Optimization.* University Press, Cambridge, 2004.

[11] Jian-Feng Cai, Emmanuel J Candès, and Zuowei Shen. A singular value thresholding algorithm for matrix completion. *SIAM Journal on Optimization*, 20(4):1956–1982, 2010.

[12] E. Candes, Y. Eldar, T. Strohmer, and V. Voroninski. Phase retrieval via matrix completion. *SIAM Journal on Imaging Sciences*, 6(1):199–225, 2013.

[13] P. Combettes and Pesquet J.-C. Signal recovery by proximal forward-backward splitting. In *Fixed-Point Algorithms for Inverse Problems in Science and Engineering*, pages 185–212. Springer-Verlag, 2011.

[14] Alan Frieze and Mark Jerrum. Improved approximation algorithms for maxk-cut and max bisection. *Algorithmica*, 18(1):67–81, 1997.

[15] Roy Frostig, Rong Ge, Sham M Kakade, and Aaron Sidford. Competing with the empirical risk minimizer in a single pass. *arXiv preprint arXiv:1412.6606*, 2014.

[16] J. Gondzio. Interior point methods 25 years later. *European Journal of Operational Research*, 218(3):587–601, 2012.

[17] A. Gramfort and M. Kowalski. Improving M/EEG source localizationwith an inter-condition sparse prior. In *IEEE International Symposium on Biomedical Imaging*, 2009.

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

[19] K. Jaganathan, S. Oymak, and B. Hassibi. Sparse phase retrieval: Uniqueness guarantees and recovery algorithms. *arXiv preprint arXiv:1311.2745*, 2013.

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

[21] R. Jenatton, A. Gramfort, V. Michel, G. Obozinski, F. Bach, and B. Thirion. Multi-scale mining of fmri data with hierarchical structured sparsity. In *Pattern Recognition in NeuroImaging (PRNI)*, 2011.

[22] Rodolphe Jenatton, Jean-Yves Audibert, and Francis Bach. Structured variable selection with sparsity-inducing norms. *The Journal of Machine Learning Research*, 12:2777–2824, 2011.

[23] Anastasios Kyrillidis, Luca Baldassarre, Marwa El Halabi, Quoc Tran-Dinh, and Volkan Cevher. Structured sparsity: Discrete and convex approaches. In *Compressed Sensing and its Applications*, pages 341–387. Springer, 2015.

[24] Anastasios Kyrillidis, Rabeeh Karimi Mahabadi, Quoc Tran Dinh, and Volkan Cevher. Scalable sparse covariance estimation via self-concordance. In *Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence*, 2014.

[25] J. Löefberg. YALMIP : A Toolbox for Modeling and Optimization in MATLAB. In *Proceedings of the CACSD Conference*, Taipei, Taiwan, 2004.

[26] RP Millane. Phase retrieval in crystallography and optics. *JOSA A*, 7(3):394–411, 1990.

[27] A. Nemirovski and A. Shapiro. Convex approximations of chance constrained programs. *SIAM Journal on Optimization*, 17(4):969–996, 2006.

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

[29] Arkadi Nemirovski. Lectures on modern convex optimization. In *Society for Industrial and Applied Mathematics (SIAM)*. Citeseer, 2001.

[30] Y. Nesterov. *Introductory lectures on convex optimization: A basic course*, volume 87 of *Applied Optimization*. Kluwer Academic Publishers, 2004.

[31] Y. Nesterov. Barrier subgradient method. *Math. Program., Ser. B*, 127:31–56, 2011.

[32] Y. Nesterov. Gradient methods for minimizing composite objective function. *Math. Program.*, 140(1):125–161, 2013.

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

[34] Y. Nesterov and M.J. Todd. Self-scaled barriers and interior-point methods for convex programming. *Math. Oper. Research*, 22(1):1–42, 1997.

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

[36] B. O'Donoghue and E. Candes. Adaptive Restart for Accelerated Gradient Schemes. *Found. Comput. Math.*, 15:715–732, April 2015.

[37] N. Parikh and S. Boyd. Proximal algorithms. *Foundations and Trends in Optimization*, 1(3):123–231, 2013.

[38] F. Rapaport, E. Barillot, and J.P. Vert. Classification of arrayCGH data using fused SVM. *Bioinformatics*, 24(13):i375–i382, 2008.

[39] J. Renegar. *A Mathematical View of Interior-Point Methods in Convex Optimization*, volume 2 of *MPS/SIAM Series on Optimization*. SIAM, 2001.

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

[41] R.T. Rockafellar and R. J-B. Wets. *Variational Analysis.* Springer-Verlag, 1997.

[42] C. Roos, T. Terlaky, and J.-Ph. Vial. *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).

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

[44] Weijie Su, Stephen Boyd, and Emmanuel Candes. A differential equation for modeling nesterovs accelerated gradient method: Theory and insights. In *Advances in Neural Information Processing Systems*, pages 2510–2518, 2014.

[45] A. Subramanian, P. Tamayo, V.K. Mootha, S. Mukherjee, B.L. Ebert, M.A. Gillette, A. Paulovich, S.L. Pomeroy, T.R. Golub, and E.S. Lander. Gene set enrichment analysis: a knowledge-based approach for interpreting genome-wide expression profiles. *Proceedings of the National Academy of Sciences of the United States of America*, 102(43):15545–15550, 2005.

[46] K.-Ch. Toh, M.J. Todd, and R.H. T́utüncü. On the implementation and usage of SDPT3 – a Matlab software package for semidefinite-quadratic-linear programming, version 4.0. Tech. report, NUS Singapore, 2010.

[47] Q. Tran-Dinh, A. Kyrillidis, and V. Cevher. An inexact proximal path-following algorithm for constrained convex minimization. *SIAM J. Optim.*, 24(4):1718–1745, 2014.

[48] Q. Tran-Dinh, A. Kyrillidis, and V. Cevher. Composite self-concordant minimization. *J. Mach. Learn. Res.*, 15:374–416, 2015.

[49] Quoc Tran Dinh, Anastasios Kyrillidis, and Volkan Cevher. A proximal Newton framework for composite minimization: Graph learning without Cholesky decompositions and matrix inversions. In *Proceedings of The 30th International Conference on Machine Learning*, pages 271–279, 2013.

[50] R.H. Tütünkü, K.C. Toh, and M.J. Todd. Solving semidefinite-quadratic-linear programs using SDPT3. *Math. Program.*, 95:189–217, 2003.

[51] Robert Vanderbei, Han Liu, Lie Wang, and Kevin Lin. Optimization for compressed sensing: the simplex method and kronecker sparsification. *arXiv preprint arXiv:1312.4426*, 2013.

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

[53] H. Zhou, M.E. Sehl, J.S. Sinsheimer, and K. Lange. Association screening of common and rare genetic variants by penalized regression. *Bioinformatics*, 26(19):2375, 2010.