Journal of Global Optimization (2022) 83:273–299
https://doi.org/10.1007/s10898-021-01105-z

A Newton Frank–Wolfe method for constrained self-concordant minimization

Deyi Liu[1] · **Volkan Cevher**[2] · **Quoc Tran-Dinh**[1]

Received: 30 June 2020 / Accepted: 18 October 2021 / Published online: 20 November 2021
© The Author(s), under exclusive licence to Springer Science+Business Media, LLC, part of Springer Nature 2021

Abstract
We develop a new Newton Frank–Wolfe algorithm to solve a class of constrained self-concordant minimization problems using linear minimization oracles (LMO). Unlike L-smooth convex functions, where the Lipschitz continuity of the objective gradient holds globally, the class of self-concordant functions only has local bounds, making it difficult to estimate the number of linear minimization oracle (LMO) calls for the underlying optimization algorithm. Fortunately, we can still prove that the number of LMO calls of our method is nearly the same as that of the standard Frank-Wolfe method in the L-smooth case. Specifically, our method requires at most $\mathcal{O}\left(\varepsilon^{-(1+\nu)}\right)$ LMO's, where ε is the desired accuracy, and $\nu \in (0, 0.139)$ is a given constant depending on the chosen initial point of the proposed algorithm. Our intensive numerical experiments on three applications: portfolio design with the competitive ratio, D-optimal experimental design, and logistic regression with elastic-net regularizer, show that the proposed Newton Frank–Wolfe method outperforms different state-of-the-art competitors.

Keywords Frank–Wolfe method · Inexact projected Newton scheme · Self-concordant function · Constrained convex optimization · Oracle complexity

Mathematics Subject Classification 90C25 · 90-08

✉ Quoc Tran-Dinh
quoctd@email.unc.edu

Deyi Liu
deyi@live.unc.edu

Volkan Cevher
volkan.cevher@epfl.ch

[1] Department of Statistics and Operations Research, The University of North Carolina at Chapel Hill, 318 Hanes Hall, Chapel Hill, NC 27599-3260, USA

[2] Laboratory for Information and Inference Systems, EPFL, Lausanne, Switzerland

1 Introduction

In this paper, we consider the following constrained convex optimization problem:

$$f^\star := \min_{\mathbf{x}\in\mathcal{X}} f(\mathbf{x}). \tag{1}$$

Here we assume that $\mathcal{X}$ is a nonempty, closed, and convex subset in $\mathbb{R}^p$ and $f : \mathbb{R}^p \to \mathbb{R} \cup \{+\infty\}$ is a smooth [1] and convex function such that $\mathrm{dom}(f) \cap \mathcal{X} \neq \emptyset$. We emphasize that, in our setting, $\mathrm{dom}(f)$ does not necessarily contain $\mathcal{X}$. Among first-order methods, the Frank–Wolfe (FW) method [14] (or more generally, the conditional gradient method) has gained tremendous popularity lately due to its scalability and its theoretical guarantees when the objective is L-smooth (i.e., its gradient ∇f is Lipschitz continuous with a Lipschitz constant L) on $\mathcal{X}$. The scalability of FW is mainly due to its computational primitive, called the linear minimization oracle (LMO):

$$\mathcal{L}_{\mathcal{X}}(\mathbf{s}) := \operatorname*{argmin}_{\mathbf{u}\in\mathcal{X}} \langle \mathbf{s}, \mathbf{u}\rangle. \tag{2}$$

There are many applications, such as latent group LASSO and simplex optimization problems where computing the LMO is significantly cheaper as compared to projecting onto the constraint set $\mathcal{X}$. If $\mathcal{X}$ is polyhedral, then evaluating $\mathcal{L}_{\mathcal{X}}(\mathbf{s})$ requires to solve a linear program, which can be achieved in polynomial-time up to very high accuracy. In many cases, evaluating $\mathcal{L}_{\mathcal{X}}(\mathbf{s})$ can be done in a closed form or with a low-order polynomial-time algorithms such as using quick-sort, see, e.g., [26] and its subsequent references.

While existing Frank–Wolfe methods can handle a sufficiently large class of convex problems, there are many machine learning problems where the objective function involves logarithmic, ridge regularized exponential, and log-determinant functions. These problems so far cannot exploit the rate as well as the scalability of the FW algorithm or its key variants. Our work precisely bridges this gap by focusing on objective functions where $f : \mathbb{R}^p \to \mathbb{R} \cup \{+\infty\}$ is standard self-concordant (see Definition 1) and $\mathcal{X}$ is a nonempty, compact, and convex set in $\mathbb{R}^p$. We emphasize that the class of self-concordant functions intersects with the class of Lipschitz continuous gradient functions, but they are different. In particular, we assume:

Assumption 1 The solution set $\mathcal{X}^\star$ of (1) is nonempty. The function f in (1) is standard self-concordant (cf. Definition 1) and its Hessian $\nabla^2 f(\mathbf{x})$ is nondegenerate (i.e., $\nabla^2 f(\mathbf{x})$ is positive definite) for any $\mathbf{x} \in \mathrm{dom}(f)$. The constraint set $\mathcal{X}$ is closed and bounded, and its LMO defined by (2) can be computed efficiently and accurately.

Note that when $\mathcal{X} \nsubseteq \mathrm{dom}(f)$, we cannot guarantee that the Hessian $\nabla^2 f$ is bounded on $\mathcal{X}$. For instance, a univariate function $f(x) := -\log(x) - \log(1-x)$ is self-concordant with its domain $\mathrm{dom}(f) = (0, 1)$. If we consider $\mathcal{X} := [0, 1]$, then f is not L-smooth on $\mathcal{X}$.

Under Assumption 1, problem (1) covers various applications in statistics and machine learning such as D-optimal experimental design [23,31], minimum-volume enclosing ellipsoid [9], quantum tomography [21], logistic regression with elastic-net regularizer [42], portfolio optimization [37], and optimal transport [36].

Related work Motivated by the fact that, for many convex sets, including simplex, polytopes, and spectrahedron, computing a linear minimization oracle is much more efficient than evaluating their projection [25,26], various linear minimization oracle-based algorithms have

[1] The smoothness of f is only defined on $\mathrm{dom}(f)$, an open set.

been proposed, see, e.g., [14,25,26,29,30]. Recently, such approaches are extended to the primal-dual setting in [44,45].

The most classical one is the Frank–Wolfe algorithm proposed in [14] for minimizing a quadratic function over a polytope. It has been shown that the convergence rate of this method is $\mathcal{O}(1/k)$ and is tight under the L-smoothness assumption, where k is the iteration counter. After that, many works have attempted to improve the convergence rate of the Frank–Wolfe algorithm and its variants by imposing further assumptions or exploiting the underlying problem structures. For instance, [3] showed a linear convergence of the Frank–Wolfe method under the assumption that f is a quadratic function and the optimal solution $\mathbf{x}^\star$ is in the interior of $\mathcal{X}$. [22] firstly proposed a variant of the Frank–Wolfe method with away-step and proved its linear rate to the optimal value if f is strongly convex, $\mathcal{X}$ is a polytope, and the optimal solution $\mathbf{x}^\star$ is in the interior of $\mathcal{X}$.

Recently, [15,28] showed that the result of [22] still holds even when $\mathbf{x}^\star$ is on the boundary of $\mathcal{X}$. This can be viewed as the first general global linear convergence result of Frank–Wolfe algorithms. [16] showed that the convergence rate of the Frank–Wolfe algorithm can be accelerated up to $\mathcal{O}\left(1/k^2\right)$ if f is strongly convex and $\mathcal{X}$ is a "strongly convex set" (see their definition).

All the results mentioned above rely on the L-smooth assumption of the objective function f. Moreover, the primal-dual methods in [44,45] suffer in proving convergence rate since they can only handle the self-concordant function by splitting the objective and then relying on the proximal operator of the self-concordant function.

For the non-L-smooth case, the literature is minimal. Notably, [34] is the first work, to the best of our knowledge, that proved that the Frank–Wolfe method could converge with $\mathcal{O}(1/k)$ rate for the Poisson phase retrieval problem where f is a logarithmic objective. This result relies on a specific simplex structure of the feasible set $\mathcal{X}$ and proved that the objective function f is eventually L-smooth on $\mathcal{X}$. However, the worst-case bound is rather loosely estimated. In addition, [9] showed a linear convergence of the Frank–Wolfe method with away-step for the minimum-volume enclosing ellipsoid problem with a log-determinant objective. The algorithms and analyses in the respective papers exploit the cost function and the structure, but it is not clear how they can handle more general self-concordant objectives. Note that since both objective functions in the aforementioned works are self-concordant, they are covered by our framework in this paper. Another related work is [13], which was available several months later after our paper was online. However, the algorithm and its analysis in [13] are different from our work, and it relies on additional assumptions.

In terms of algorithm, there are also several papers exploiting combination between the Frank–Wolfe method and other schemes to solve different problems. For instance, [17,20] propose to combine the Frank–Wolfe method and a (quasi) Newton scheme to solve constrained nonlinear systems, where local and global convergence rates are established, respectively. [11,19] further generalize the Frank–Wolfe method in [17,20] to an inexact projection framework. Notice that these algorithms are fundamentally different from our method. In fact, they first solve the Newton system and then apply a Frank–Wolfe method to estimate the projection, while our method uses Frank–Wolfe scheme directly to solve the constrained quadratic subproblem (4). In a concurrent work [18], a Frank–Wolfe variant is proposed as a subsolver for the subproblem of the underlying quasi-Newton method, which is similar to ours. However, [18] does not establish an explicit convergence rate for the proposed method and uses a different set of assumptions.

Our goal and approach Our first goal is to tackle an important class of problems (1), where existing LMO-based methods do not have convergence guarantees. Our results have advantages when computing the LMO is cheaper than computing projections. Otherwise, the

first-order methods, e.g., from [41] can also be applied. For this purpose, we apply a projected Newton method to solve (1) and use the Frank–Wolfe method in the subproblems to approximate the projected Newton direction. This approach leads to a double-loop algorithm, where the outer loop performs an inexact projected Newton scheme, and the inner loop carries out an adaptive Frank–Wolfe scheme by automatically adjusting the inner accuracy.

Notice that our algorithm enjoys several additional computational advantages. When the feasible set $\mathcal{X}$ is a polytope, our subproblem becomes minimizing a quadratic function over a polytope. By the result of [28], we can use the Frank–Wolfe algorithm with away-steps to attain linear convergence without sacrificing the overall complexity. Since our objective function in the subproblem is quadratic, the optimal step-size at each iteration has a closed form expression, leading to structure exploiting variants (see Algorithm 2). Finally, our algorithm can enhance Frank–Wolfe-type approaches by using the inexact projected Newton direction.

Our contribution To this end, our contribution can be summarized as follows:

(a) We propose a double-loop algorithm to solve (1) when f is *self-concordant* (see Definition 1) and $\mathcal{X}$ is equipped with a *tractable* linear minimization oracle. The proposed algorithm is self-adaptive, i.e., it does not require tuning for the step-size and accuracy of the subproblem.

(b) We prove that the gradient and Hessian complexity of our method is $\mathcal{O}\left(\log\left(\frac{1}{\varepsilon}\right)\right)$, while the LMO complexity is $\mathcal{O}\left(\varepsilon^{-(1+\nu)}\right)$, where $\nu := \frac{\log(1-2\beta)}{\log(2\beta)}$ and $\beta > 0$ can be sufficiently small. When β approaches zero, the complexity bound also approaches $\mathcal{O}\left(\frac{1}{\varepsilon}\right)$ as in the Frank–Wolfe methods for the L-smooth case.

To the best of our knowledge, this work is the first one studying LMO-based methods for solving (1) with non-Lipschitz continuous gradient functions on a general convex set $\mathcal{X}$. It also covers the models in [9,34] as special cases, via a completely different approach.

Paper outline The rest of this paper is organized as follows. Section 2 recalls some basic notation and preliminaries of self-concordant functions. Section 3 presents the main algorithm. Section 4 proves the local linear convergence of the outer loop and gives a rigorous analysis of the total oracle complexity. Three numerical experiments are given in Sect. 5. Finally, we draw some conclusions in Sect. 6. For the sake of presentation, all the technical proofs are deferred to the "Appendix".

2 Theoretical background

Basic notation We work with Euclidean spaces, $\mathbb{R}^p$ and $\mathbb{R}^n$, equipped with standard inner product $\langle\cdot,\cdot\rangle$ and Euclidean norm $\|\cdot\|$. For a given proper, closed, and convex function $f : \mathbb{R}^p \to \mathbb{R}\cup\{+\infty\}$, $\mathrm{dom}(f) := \{\mathbf{x} \in \mathbb{R}^p \mid f(\mathbf{x}) < +\infty\}$ denotes the domain of f, ∂f denotes the subdifferential of f, and f^* is its Fenchel conjugate. For a symmetric matrix $\mathbf{A} \in \mathbb{R}^{n\times n}$, $\lambda_{\max}(\mathbf{A})$ denotes the largest eigenvalue of $\mathbf{A}$. We use $[k]$ to denote the set $\{1, \cdots, k\}$, and $\mathbf{e}$ to denote the vector whose elements are 1s. For a vector $\mathbf{u} \in \mathbb{R}^p$, $\mathrm{Diag}(\mathbf{u})$ is a $p \times p$ diagonal matrix formed by $\mathbf{u}$. We also define two nonnegative and monotonically increasing functions $\omega(\tau) := \tau - \log(1+\tau)$ for $\tau \in [0, \infty)$ and $\omega_*(\tau) := -\tau - \log(1-\tau)$ for $\tau \in [0, 1)$. We use $\mathbf{H} \succeq 0$ (resp., $\mathbf{H} \succ 0$) to denote a symmetric positive semidefinite (resp., definite) matrix $\mathbf{H}$.

2.1 Self-concordant functions

Our class of objective functions in (1) is self-concordant. Hence, we recall the definition of self-concordant functions introduced in [33] here.

Definition 1 A three times continuously differentiable [2] univariate function $\varphi : \mathbb{R} \to \mathbb{R} \cup \{+\infty\}$ is said to be self-concordant with a parameter $M_\varphi \geq 0$ if $|\varphi'''(\tau)| \leq M_\varphi \varphi''(\tau)^{3/2}$ for all $\tau \in \mathrm{dom}(\varphi)$. A three times continuously differentiable function $f : \mathbb{R}^p \to \mathbb{R} \cup \{+\infty\}$ is said to be self-concordant with a parameter $M_f \geq 0$ if $\varphi(\tau) := f(\mathbf{x} + \tau \mathbf{v})$ is self-concordant with the same parameter M_f for any $\mathbf{x} \in \mathrm{dom}(f)$ and $\mathbf{v} \in \mathbb{R}^p$. If $M_f = 2$, then we say that f is standard self-concordant.

Note that any self-concordant function f can be rescaled to the standard form as $\hat{f}(\cdot) := (M_f^2/4) f(\cdot)$. When $\mathrm{dom}(f)$ does not contain straight line, $\nabla^2 f(\mathbf{x})$ is nondegenerate (i.e., positive definite) [32, Theorem 4.1.3], and therefore we can define a local norm associated with f together with its dual norm as follows:

$$\|\mathbf{u}\|_{\mathbf{x}} := \left(\mathbf{u}^\top \nabla^2 f(\mathbf{x}) \mathbf{u}\right)^{1/2} \quad \text{and} \quad \|\mathbf{u}\|_{\mathbf{x}}^* := \left(\mathbf{u}^\top \nabla^2 f(\mathbf{x})^{-1} \mathbf{u}\right)^{1/2}.$$

These norms are weighted and satisfy the Cauchy-Schwarz inequality $\langle \mathbf{u}, \mathbf{v} \rangle \leq \|\mathbf{u}\|_{\mathbf{x}} \|\mathbf{v}\|_{\mathbf{x}}^*$ for $\mathbf{u}, \mathbf{v} \in \mathbb{R}^p$.

The class of self-concordant functions is sufficiently broad to cover many important applications. It is closed under nonnegative combination and affine transformation. Any linear and convex quadratic functions are self-concordant. The function $f_1(\mathbf{x}) := -\log(\mathbf{x})$ and $f_2(\mathbf{x}) := \mathbf{x}\log(\mathbf{x}) - \log(\mathbf{x})$ are self-concordant. For symmetric positive semidefinite matrices, $f_3(\mathbf{X}) := -\log\det(\mathbf{X})$ is also self-concordant, which is widely used in covariance estimation-type and experimental design problems. In statistical learning, the regularized logistic regression model with $f_4(\mathbf{x}) := \frac{1}{n}\sum_{i=1}^n \log(1 + \exp(-y_i \mathbf{a}_i^\top \mathbf{x})) + \frac{\mu_f}{2}\|\mathbf{x}\|^2$ and the regularized Poisson regression model with $f_5(\mathbf{x}) := \frac{1}{n}\sum_{i=1}^n \left(y_i \exp(\frac{-\mathbf{a}_i^\top \mathbf{x}}{2}) + \exp(\frac{\mathbf{a}_i^\top \mathbf{x}}{2}) \right) + \frac{\mu_f}{2}\|\mathbf{x}\|^2$ are both self-concordant. Note that all the functions introduced above are not globally L-smooth on their domain except for f_4. In addition, any three times continuously differentiable and strongly convex function with Lipschitz Hessian continuity is also self-concordant. We refer the reader to [35,40] for more examples and theoretical results.

2.2 Approximate solutions

Since $\nabla^2 f(\mathbf{x})$ is nondegenerate, (1) has only one optimal solution $\mathbf{x}^\star$. Moreover, $\nabla^2 f(\mathbf{x}^\star) \succ 0$. Our goal is to design an algorithm to approximate $\mathbf{x}^\star$ as follows:

Definition 2 Given a tolerance $\varepsilon > 0$, we say that $\mathbf{x}_\varepsilon^\star$ is an ε-solution of (1) if

$$\|\mathbf{x}_\varepsilon^\star - \mathbf{x}^\star\|_{\mathbf{x}^\star} \leq \varepsilon. \tag{3}$$

Different from existing Frank–Wolfe methods where an approximate solution $\mathbf{x}_\varepsilon^\star$ is defined by $f(\mathbf{x}_\varepsilon^\star) - f^\star \leq \varepsilon$, we define it via a local norm. However, we show in Theorem 4 that these two concepts are related to each other.

[2] The differentiability of φ is only defined on $\mathrm{dom}(\varphi)$, an open set.

3 The proposed Newton Frank–Wolfe algorithm

Since f in (1) is standard self-concordant, we first approximate it by a quadratic surrogate and apply a projected Newton method to solve (1). More precisely, given $\mathbf{x} \in \mathrm{dom}(f) \cap \mathcal{X}$, the projected Newton method computes a search direction at $\mathbf{x}$ by solving the following constrained convex quadratic program:

$$T(\mathbf{x}) := \operatorname*{argmin}_{\mathbf{u}\in\mathcal{X}} \left\{ Q_f(\mathbf{u};\mathbf{x}) := \langle \nabla f(\mathbf{x}), \mathbf{u}-\mathbf{x}\rangle + \frac{1}{2}(\mathbf{u}-\mathbf{x})^\top \nabla^2 f(\mathbf{x})(\mathbf{u}-\mathbf{x}) \right\}. \tag{4}$$

Since $\nabla^2 f(\mathbf{x})$ is positive definite by Assumption 1, $T(\mathbf{x})$ is the unique solution of (4). However, this problem often does not have a closed-form solution, and we need to approximate it up to a given accuracy. Since we aim at exploiting LMO of $\mathcal{X}$, we apply a Frank–Wolfe scheme to solve (4). The optimality condition of (4) becomes

$$\langle \nabla Q_f(T(\mathbf{x});\mathbf{x}), T(\mathbf{x}) - \mathbf{u}\rangle \le 0, \quad \forall \mathbf{u} \in \mathcal{X}, \tag{5}$$

where $\nabla Q_f(T(\mathbf{x});\mathbf{x}) = \nabla f(\mathbf{x}) + \nabla^2 f(\mathbf{x})(T(\mathbf{x}) - \mathbf{x})$. Using this optimality condition, we can define an inexact solution of (4) as follows:

Definition 3 Given a tolerance $\eta > 0$, we say that $T_\eta(\mathbf{x})$ is an η-solution of (4) if

$$\max_{\mathbf{u}\in\mathcal{X}} \langle \nabla Q_f(T_\eta(\mathbf{x});\mathbf{x}), T_\eta(\mathbf{x}) - \mathbf{u}\rangle \le \eta^2. \tag{6}$$

The following lemma shows that the distance between $T(\mathbf{x})$ and $T_\eta(\mathbf{x})$ can be bounded by η. Therefore, this justifies the well-definedness of Definition 3.

Lemma 1 *Let $T_\eta(\mathbf{x})$ be an η-solution defined by Definition 3 and $T(\mathbf{x})$ be the exact solution of (4). Then, it holds that $\|T_\eta(\mathbf{x}) - T(\mathbf{x})\|_{\mathbf{x}} \le \eta$.*

Proof From Definition 3, we have $\langle \nabla Q_f(T_\eta(\mathbf{x});\mathbf{x}), T_\eta(\mathbf{x}) - T(\mathbf{x})\rangle \le \eta^2$. Since $Q_f(\cdot;\mathbf{x})$ is a convex quadratic function, it is easy to show that

$$\begin{aligned}&\langle \nabla Q_f(T(\mathbf{x});\mathbf{x}) + \nabla^2 f(\mathbf{x})(T_\eta(\mathbf{x}) - T(\mathbf{x})), T_\eta(\mathbf{x}) - T(\mathbf{x})\rangle \\ &= \langle \nabla Q_f(T_\eta(\mathbf{x});\mathbf{x}), T_\eta(\mathbf{x}) - T(\mathbf{x})\rangle \le \eta^2.\end{aligned}$$

Substituting $T_\eta(\mathbf{x})$ for $\mathbf{u}$ in the optimality condition (5), we obtain

$$\langle \nabla Q_f(T(\mathbf{x});\mathbf{x}), T_\eta(\mathbf{x}) - T(\mathbf{x})\rangle \ge 0.$$

Combining the above two inequalities, we finally get

$$\langle \nabla^2 f(\mathbf{x})(T_\eta(\mathbf{x}) - T(\mathbf{x})), T_\eta(\mathbf{x}) - T(\mathbf{x})\rangle \le \eta^2,$$

which is equivalent to $\|T_\eta(\mathbf{x}) - T(\mathbf{x})\|_{\mathbf{x}} \le \eta$. □

Now, we combine our inexact projected Newton scheme and the well-known Frank–Wolfe algorithm to develop a new algorithm as presented in Algorithm 1.

Let us make a few remarks on Algorithm 1.

(a) *Discussion on structure* Algorithm 1 integrates both damped-step and full-step inexact projected Newton schemes. First, it performs the damped-step scheme to generate $\{\mathbf{x}^k\}$ starting from an initial point $\mathbf{x}^0$ that may be far from the optimal solution $\mathbf{x}^\star$. Then, once $\|\mathbf{x}^k - \mathbf{x}^\star\|_{\mathbf{x}^\star} \le \beta$ is satisfied, it switches to the full-step scheme. For the damped-step stage, we will show later that Algorithm 1 only performs a finite number of iterations.

Algorithm 1 (*Newton Frank–Wolfe Algorithm*)

Inputs: Input $\varepsilon > 0$ and $\mathbf{x}^0 \in \mathrm{dom}(f) \cap \mathcal{X}$.
- Choose $(\beta, \sigma, C) > 0$ such that (11) holds. Choose $C_1 \in (0, 0.5)$ and $\delta \in (0, 1)$.
- Set $\lambda_{-1} := \frac{\beta}{\sigma}$ and $\eta_0 := \min\{\frac{\beta}{C}, C_1 h^{-1}(\beta)\}$, where h is defined in (8).

for $k := 0, 1, \cdots$ **do**
  $\mathbf{z}^k :=$ **Adaptive_Frank_Wolfe_Subroutine**$(\nabla f(\mathbf{x}^k), \nabla^2 f(\mathbf{x}^k)[\cdot], \mathbf{x}^k, \eta_k^2)$.
  $\mathbf{d}^k := \mathbf{z}^k - \mathbf{x}^k$ and $\gamma_k := \|\mathbf{d}^k\|_{\mathbf{x}^k}$.
  if $\gamma_k + \eta_k \le h^{-1}(\beta)$ **or** $\lambda_{k-1} \le \beta$ **then**
    $\lambda_k := \sigma\lambda_{k-1}$ and $\eta_{k+1} := \sigma\eta_k$
    $\mathbf{x}^{k+1} := \mathbf{x}^k + \mathbf{d}^k$ (**full-step**)
  else
    $\lambda_k := \lambda_{k-1}$ and $\eta_{k+1} := \eta_k$.
    $\alpha_k := \delta(\gamma_k^2 - \eta_k^2)/(\gamma_k^3 + \gamma_k^2 - \eta_k^2\gamma_k)$.
    $\mathbf{x}^{k+1} := \mathbf{x}^k + \alpha_k\mathbf{d}^k$ (**damped-step**)
  end if
  if $\lambda_k \le \varepsilon$ **then**
    return $\mathbf{x}^{k+1}$
  end if
end for

Algorithm 2 (*Adaptive Frank–Wolfe Subroutine*)

Adaptive_Frank_Wolfe_Subroutine$(\mathbf{h}, \mathbf{H}[\cdot], \mathbf{u}^0, \eta)$
for $t := 0, 1, \cdots T$ **do**
  $\mathbf{g}^t := \mathbf{h} + \mathbf{H}(\mathbf{u}^t - \mathbf{u}^0)$.
  $\mathbf{v}^t := \arg\max_{\mathbf{s}\in\mathcal{X}} \langle \mathbf{g}^t, \mathbf{u}^t - \mathbf{s}\rangle$.
  $V_t := \langle \mathbf{g}^t, \mathbf{u}^t - \mathbf{v}^t\rangle$.
  if $V_t > \eta$ **then**
    $\delta_t := \|\mathbf{v}^t - \mathbf{u}^t\|_{\mathbf{H}}^2$ and $\tau_t := \min\{1, V_t/\delta_t\}$.
    $\mathbf{u}^{t+1} := (1 - \tau_t)\mathbf{u}^t + \tau_t\mathbf{v}^t$.
  else
    return $\mathbf{u}^t$.
  end if
end for

(b) *Discussion on the Newton decrement* λ_k. Due to the update rule of λ_k in Algorithm 1 we have $\lambda_k := \beta\sigma^k$. As proved in (12) of Theorem 2 below, one has $\|\mathbf{x}^k - \mathbf{x}^\star\|_{\mathbf{x}^\star} \le \lambda_k$ in the full-step stage. [3] Since λ_k is decreased geometrically by a factor $\sigma \in (0, 1)$ as $\lambda_k := \sigma\lambda_{k-1}$, $\|\mathbf{x}^k - \mathbf{x}^\star\|_{\mathbf{x}^\star}$ converges linearly to zero (see Theorem 2). Notice that in the damped-step stage, we keep λ_k unchanged. Therefore, λ_k does not upper bound $\|\mathbf{x}^k - \mathbf{x}^\star\|_{\mathbf{x}^\star}$ in this case.

(c) *Discussion on the inner accuracy* η_k. The quantity η_k is used to measure $\|T(\mathbf{x}^k) - \mathbf{z}^k\|_{\mathbf{x}^k}$ (see (4) for the definition of $T(\mathbf{x}^k)$). In Algorithm 1, $\mathbf{z}^k$ is calculated by Algorithm 2 as

$$\mathbf{z}^k := \textbf{Adaptive\_Frank\_Wolfe\_Subroutine}(\nabla f(\mathbf{x}^k), \nabla^2 f(\mathbf{x}^k)[\cdot], \mathbf{x}^k, \eta_k^2). \quad (7)$$

[3] Notice that Theorem 2 is proven under an assumption that x^0 is sufficiently close to $x^\star$ (the optimal solution of (1)) so that the damped step is never invoked.

From the stop criterion of Algorithm 2, $\mathbf{z}^k$ is an η_k-approximate solution by Definition 3 at $\mathbf{x} = \mathbf{x}^k$ and may not be in $\mathrm{dom}(f)$. According Lemma 1, we have $\|\mathbf{z}^k - T(\mathbf{x}^k)\|_{\mathbf{x}^k} \le \eta_k$. Therefore, η_k measures the accuracy for solving the subproblem. In the damped-step stage, we keep η_k as a constant. In the full-step one, η_k is decreased by a factor of $\sigma \in (0, 1)$ at each iteration to guarantee that we get a more accurate projected Newton direction when the algorithm approaches the optimal solution $\mathbf{x}^\star$. The following lemma shows that the choice of our step-size guarantees that $\mathbf{x}^k \in \mathrm{dom}(f) \cap \mathcal{X}$ regardless of the full-step or the damped-step.

Lemma 2 *Let* $\{\mathbf{x}^k\}$ *be generated by Algorithm 1. Then* $\{\mathbf{x}^k\} \subset \mathrm{dom}(f) \cap \mathcal{X}$.

Proof We prove this lemma by induction. Due to the initialization of Algorithm 1, we have $x^0 \in \mathrm{dom}(f) \cap \mathcal{X}$. Assume that $x^k \in \mathrm{dom}(f) \cap \mathcal{X}$ for $k \ge 0$. We now show that $x^{k+1} \in \mathrm{dom}(f) \cap \mathcal{X}$. Since $\mathcal{X}$ is convex, $\mathbf{x}^{k+1} = (1-\alpha_k)\mathbf{x}^k + \alpha_k \mathbf{z}^k$, $\mathbf{x}^k \in \mathcal{X}$, $\mathbf{z}^k \in \mathcal{X}$, and $\alpha_k \in (0, 1]$, it is obvious that $\mathbf{x}^{k+1} \in \mathcal{X}$. We only need to show that $\mathbf{x}^{k+1} \in \mathrm{dom}(f)$. If we update $\mathbf{x}^{k+1}$ by the damped-step, then by Algorithm 1, we have

$$\|\mathbf{x}^{k+1} - \mathbf{x}^k\|_{\mathbf{x}^k} = \alpha_k \|\mathbf{z}^k - \mathbf{x}^k\|_{\mathbf{x}^k} = \alpha_k \gamma_k = \frac{\delta(\gamma_k^2 - \eta_k^2)}{\gamma_k^2 - \eta_k^2 + \gamma_k} < 1.$$

Alternatively, if we update $\mathbf{x}^{k+1}$ by the full-step, then by (12) of Theorem 2 below, we have $\|\mathbf{x}^{k+1} - \mathbf{x}^k\|_{\mathbf{x}^k} = 2\beta\sigma^k < 1$. In both cases, we have $\|\mathbf{x}^{k+1} - \mathbf{x}^k\|_{\mathbf{x}^k} < 1$. Hence, by using [32, Theorem 4.1.5], we conclude that $x^{k+1} \in \mathrm{dom}(f)$. Consequently, $x^{k+1} \in \mathrm{dom}(f) \cap \mathcal{X}$. By induction, we have $\{x^k\} \subset \mathrm{dom}(f) \cap \mathcal{X}$. □

(d) *Discussion on the switching condition* $\gamma_k + \eta_k \le h^{-1}(\beta)$. When $\gamma_k + \eta_k > h^{-1}(\beta)$, we use a damped-step scheme with the step-size

$$\alpha_k := \frac{\delta(\gamma_k^2 - \eta_k^2)}{\gamma_k^3 + \gamma_k^2 - \eta_k^2 \gamma_k}.$$

This step-size is derived from Lemma 3 in the "Appendix", and is in $(0, 1)$. Once $\gamma_k + \eta_k \le h^{-1}(\beta)$ is satisfied, we move to the full-step stage and no longer use the damped-step one. In addition, from Lemma 4 in the "Appendix", we can see that if $\gamma_k + \eta_k \le h^{-1}(\beta)$, then we have $\|\mathbf{x}^k - \mathbf{x}^\star\|_{\mathbf{x}^\star} \le \beta$, which means that we already find a good initial point for the full-step stage.

(e) *Discussion on the Frank–Wolfe subroutine* The subroutine (7) is an adaptive Frank–Wolfe variant, which is customized to solve the following constrained convex quadratic program:

$$\min_{\mathbf{x} \in \mathcal{X}} \left\{ \psi(\mathbf{x}) := \langle \mathbf{h}, \mathbf{x} - \mathbf{u}^0 \rangle + \tfrac{1}{2} \langle \mathbf{H}(\mathbf{x} - \mathbf{u}^0), \mathbf{x} - \mathbf{u}^0 \rangle \right\}.$$

The step size τ_t in (7) is computed via the following exact linesearch condition (see [29] for further details):

$$\tau_t := \arg\min_{\alpha \in [0,1]} \left\{ \psi(\mathbf{u}^t + \alpha(\mathbf{v}^t - \mathbf{u}^t)) \right\}.$$

(f) *Discussion on the Hessian evaluation* $\nabla^2 f(\cdot)$. In practice, we do not need to evaluate the full Hessian $\nabla^2 f(\mathbf{x}^k)$ at each iteration k. We only need to evaluate the matrix-vector operator $\nabla^2 f(\mathbf{x}^k)\mathbf{v}$ for a given direction $\mathbf{v}$. Similarly, the computation of γ_k does not incur significant cost. Indeed, since we have already computed $\nabla^2 f(\mathbf{x}^k)\mathbf{d}^k$ in (7), computing γ_k requires only one additional vector inner product $\langle \nabla^2 f(\mathbf{x}^k)\mathbf{d}^k, \mathbf{d}^k \rangle$.

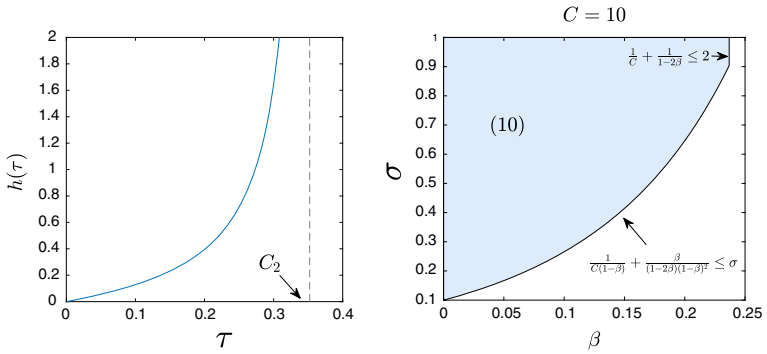

Fig. 1 The shape of h (left) and the feasible region of (β, σ) for (11) when $C = 10$ (right)

(g) *Faster Frank–Wolfe variants* Since $\mathcal{X}$ in(1) is a general convex set, our analysis below is based on the standard Frank–Wolfe variant [26]. However, when it is possible (e.g., $\mathcal{X}$ is a polytope or a strongly convex set [16]), we can replace this standard Frank–Wolfe subroutine by a faster variant. For instance, if $\mathcal{X}$ is a polytope, then we can use an away-step variant, which often has a linear convergence rate [28]. If $\mathcal{X}$ is strongly convex [16], then we can apply an accelerated variant, which can achieve up to $\mathcal{O}\left(1/T^2\right)$ convergence rate. In both cases, the LMO complexity stated in Theorems 3 and 4 still holds (up to a constant factor), or can even be improved.

4 Convergence and complexity analysis

Our analysis closely follows the outline below:

- Given $\beta \in (0, 1)$, we show that we only need a finite number of damped-steps to reach $\mathbf{x}^k$ such that $\|\mathbf{x}^k - \mathbf{x}^\star\|_{\mathbf{x}^\star} \leq \beta$. We call it the damped-step stage.
- Once $\|\mathbf{x}^k - \mathbf{x}^\star\|_{\mathbf{x}^\star} \leq \beta$ is satisfied, we prove a linear convergence of the full-step projected Newton scheme. We call this the full-step stage.
- We finally estimate the overall LMO complexity of Algorithm 1.

4.1 Finite complexity of damped-step stage

Before we present the main theorem of this section, let us first define a univariate function $h : \mathbb{R}_+ \to \mathbb{R}_+$, whose shape is shown in Fig. 1, as

$$h(\tau) := \frac{\tau(1 - 2\tau + 2\tau^2)}{(1 - 2\tau)(1 - \tau)^2 - \tau^2}. \tag{8}$$

From Fig. 1, h is nonnegative and monotonically increasing on $[0, C_2)$ for the constant $C_2 \in (0.3, 0.4)$ such that $(1 - 2C_2)(1 - C_2)^2 - C_2^2 = 0$.

The following theorem states that Algorithm 1 only needs a finite number of LMO calls $\mathcal{T}_1$ to achieve $\mathbf{x}^k$ such that $\|\mathbf{x}^k - \mathbf{x}^\star\|_{\mathbf{x}^\star} \leq \beta$. Although $\mathcal{T}_1$ is independent of tolerance ε, it depends on the pre-defined constants β and C_1 in the algorithm and the structure of f and $\mathcal{X}$.

Theorem 1 *Let $\omega(\tau) := \tau - \log(1+\tau)$. If we choose the parameters as in Algorithm 1, then after at most*

$$K := \frac{f(\mathbf{x}^0) - f(\mathbf{x}^\star)}{\delta\omega\left(\frac{1-2C_1}{1-C_1}h^{-1}(\beta)\right)} \tag{9}$$

outer iterations of the damped-step scheme, we can guarantee that $\gamma_k + \eta_k \le h^{-1}(\beta)$ for some $k \in [K]$, which implies that $\|\mathbf{x}^k - \mathbf{x}^\star\|_{\mathbf{x}^\star} \le \beta$. Moreover, the total number of LMO calls is at most

$$\mathcal{T}_1 := \frac{6D_{\mathcal{X}}^2\lambda_{\max}(\nabla^2 f(\mathbf{x}^0))}{(C_1 h^{-1}(\beta))^2}\frac{1-(1-\delta)^{2K+1}}{\delta(1-\delta)^{2K}}, \tag{10}$$

where $D_{\mathcal{X}} := \max_{\mathbf{x},\mathbf{y}\in\mathcal{X}} \|\mathbf{x}-\mathbf{y}\|$. The number of gradient $\nabla f(\mathbf{x}^k)$ and Hessian $\nabla^2 f(\mathbf{x}^k)$ evaluations is also at most K.

Proof Notice that in Algorithm 1, we always have $\eta_k \le C_1 h^{-1}(\beta)$ in the damped-step stage, where $C_1 \in (0, 0.5)$. Clearly, if $\gamma_k+\eta_k > h^{-1}(\beta)$, then $\gamma_k > h^{-1}(\beta)-\eta_k \ge (1-C_1)h^{-1}(\beta)$. Therefore, we can show that

$$\frac{\gamma_k^2-\eta_k^2}{\gamma_k} \ge \frac{((1-C_1)h^{-1}(\beta))^2 - (C_1 h^{-1}(\beta))^2}{(1-C_1)h^{-1}(\beta)} = \frac{1-2C_1}{1-C_1}h^{-1}(\beta).$$

Using Lemma 3 in the "Appendix" and the monotonicity of ω we also have

$$f(\mathbf{x}^{k+1}) \overset{(26)}{\le} f(\mathbf{x}^k) - \delta\omega\left(\frac{\gamma_k^2-\eta_k^2}{\gamma_k}\right) \le f(\mathbf{x}^k) - \delta\omega\left(\frac{1-2C_1}{1-C_1}h^{-1}(\beta)\right).$$

Consequently, we need at most $K := \frac{f(\mathbf{x}^0)-f(\mathbf{x}^\star)}{\delta\omega\left(\frac{1-2C_1}{1-C_1}h^{-1}(\beta)\right)}$ outer iterations to get $\gamma_k+\eta_k \le h^{-1}(\beta)$ as stated in (9).

From Lemma 6 in the "Appendix", we can show that the number of LMO calls needed at the k-th outer iteration is $T_k := \frac{6\lambda_{\max}(\nabla^2 f(\mathbf{x}^k))D_{\mathcal{X}}^2}{\eta_k^2}$. Since f is self-concordant, we have

$$\nabla^2 f(\mathbf{x}^{k+1}) \preceq \frac{\nabla^2 f(\mathbf{x}^k)}{(1-\|\mathbf{x}^{k+1}-\mathbf{x}^k\|_{\mathbf{x}^k})^2} = \frac{\nabla^2 f(\mathbf{x}^k)}{(1-\alpha_k\gamma_k)^2} \preceq \frac{\nabla^2 f(\mathbf{x}^k)}{(1-\delta)^2},$$

which implies that $\nabla^2 f(\mathbf{x}^k) \le \frac{\nabla^2 f(\mathbf{x}^0)}{(1-\delta)^{2k}}$. Hence, the total number of LMO calls can be computed by

$$\begin{aligned}
\mathcal{T}_1 &:= \textstyle\sum_{k=0}^K T_k = 6D_{\mathcal{X}}^2 \sum_{k=0}^K \frac{\lambda_{\max}(\nabla^2 f(\mathbf{x}^k))}{\eta_k^2} \\
&\le \frac{6D_{\mathcal{X}}^2}{(C_1 h^{-1}(\beta))^2} \textstyle\sum_{k=0}^K \frac{\lambda_{\max}(\nabla^2 f(\mathbf{x}^0))}{(1-\delta)^{2k}} \\
&\le \frac{6D_{\mathcal{X}}^2}{(C_1 h^{-1}(\beta))^2} \textstyle\sum_{k=0}^{2K} \frac{\lambda_{\max}(\nabla^2 f(\mathbf{x}^0))}{(1-\delta)^{k}} \\
&= \frac{6D_{\mathcal{X}}^2\lambda_{\max}(\nabla^2 f(\mathbf{x}^0))}{(C_1 h^{-1}(\beta))^2}\frac{1-(1-\delta)^{2K+1}}{\delta(1-\delta)^{2K}}.
\end{aligned}$$

Finally, if $\gamma_k+\eta_k \le h^{-1}(\beta)$, then we have $\bar{\lambda}_k := \|\mathbf{x}^k-\mathbf{x}^\star\|_{\mathbf{x}^\star} \overset{(30)}{\le} h(\gamma_k+\eta_k) \le h(h^{-1}(\beta)) = \beta$. □

4.2 Linear convergence of full-step stage

Theorem 1 shows that we only need a finite number of damped-steps to obtain $\mathbf{x}^k$ such that $\|\mathbf{x}^k-\mathbf{x}^\star\|_{\mathbf{x}^\star} \le \beta$. Therefore, without loss of generality, we always assume that $\|\mathbf{x}^0-\mathbf{x}^\star\|_{\mathbf{x}^\star} \le \beta$ in the rest of this paper. Using this assumption, we analyze convergence rate of $\{\mathbf{x}^k\}$ to the unique optimal solution $\mathbf{x}^\star$ of (1). In this case, Algorithm 1 always choose full-steps, i.e., $\mathbf{x}^{k+1} := \mathbf{x}^k + \mathbf{d}^k = \mathbf{z}^k$.

The following theorem states the linear convergence of $\|\mathbf{x}^k - \mathbf{x}^\star\|_{\mathbf{x}^\star}$ and $\|\mathbf{x}^{k+1} - \mathbf{x}^k\|_{\mathbf{x}^k}$. The convergence of $\|\mathbf{x}^{k+1} - \mathbf{x}^k\|_{\mathbf{x}^k}$ will be used in Theorem 3 to bound $\{\nabla^2 f(\mathbf{x}^k)\}$ which is key to our LMO complexity analysis.

Theorem 2 *Suppose that* $\|\mathbf{x}^0 - \mathbf{x}^\star\|_{\mathbf{x}^\star} \le \beta$ *and the triple* (σ, β, C) *satisfies:*

$$\begin{cases} \sigma \in (0,1), \quad \beta \in (0, 0.5), \quad C > 1, \\ \frac{1}{C(1-\beta)} + \frac{\beta}{(1-2\beta)(1-\beta)^2} \le \sigma, \\ \frac{1}{C} + \frac{1}{(1-2\beta)} \le 2. \end{cases} \tag{11}$$

Let $\eta_k := \frac{\beta\sigma^k}{C}$ *and* $\{\mathbf{x}^k\}$ *be updated by the full-step stage in Algorithm 1. Then, for* $k \ge 0$*, the following bounds hold:*

$$\|\mathbf{x}^k - \mathbf{x}^\star\|_{\mathbf{x}^\star} \le \beta\sigma^k \ \text{ and } \ \|\mathbf{x}^{k+1} - \mathbf{x}^k\|_{\mathbf{x}^k} \le 2\beta\sigma^k. \tag{12}$$

Proof We prove this theorem by induction. Firstly, we have $\|\mathbf{x}^0 - \mathbf{x}^\star\|_{\mathbf{x}^\star} \le \beta\sigma^0 = \beta < 1$ by assumption. Next, suppose that $\bar{\lambda}_k := \|\mathbf{x}^k - \mathbf{x}^\star\|_{\mathbf{x}^\star} \le \beta\sigma^k$ for $k \ge 0$. By induction, we can derive

$$\begin{aligned} \bar{\lambda}_{k+1} &:= \|\mathbf{x}^{k+1} - \mathbf{x}^\star\|_{\mathbf{x}^\star} \\ &\overset{(35)}{\le} \frac{\eta_k}{1-\bar{\lambda}_k} + \frac{\bar{\lambda}_k^2}{(1-\bar{\lambda}_k)^2(1-2\bar{\lambda}_k)} \\ &= \frac{\beta\sigma^k}{C(1-\bar{\lambda}_k)} + \frac{\bar{\lambda}_k^2}{(1-\bar{\lambda}_k)^2(1-2\bar{\lambda}_k)} \\ &\le \left(\frac{1}{C(1-\bar{\lambda}_k)} + \frac{\bar{\lambda}_k}{(1-\bar{\lambda}_k)^2(1-2\bar{\lambda}_k)}\right)\beta\sigma^k \qquad \text{(by induction)} \\ &\le \left(\frac{1}{C(1-\beta)} + \frac{\beta}{(1-\beta)^2(1-2\beta)}\right)\beta\sigma^k \qquad \text{(by induction)} \\ &\overset{(11)}{\le} \beta\sigma^{k+1}, \end{aligned}$$

which proves the first estimate of (12).

Similarly, we also have

$$\begin{aligned} \|\mathbf{x}^{k+1} - \mathbf{x}^k\|_{\mathbf{x}^k} &\overset{(36)}{\le} \eta_k + \frac{\bar{\lambda}_k^2}{(1-2\bar{\lambda}_k)(1-\bar{\lambda}_k)} + \frac{\bar{\lambda}_k}{1-\bar{\lambda}_k} \\ &= \frac{\beta\sigma^k}{C} + \frac{\bar{\lambda}_k^2}{(1-2\bar{\lambda}_k)(1-\bar{\lambda}_k)} + \frac{\bar{\lambda}_k}{1-\bar{\lambda}_k} \\ &\le \left(\frac{1}{C} + \frac{\bar{\lambda}_k}{(1-\bar{\lambda}_k)(1-2\bar{\lambda}_k)} + \frac{1}{1-\bar{\lambda}_k}\right)\beta\sigma^k \qquad \text{(by induction)} \\ &\le \left(\frac{1}{C} + \frac{\beta}{(1-\beta)(1-2\beta)} + \frac{1}{1-\beta}\right)\beta\sigma^k \qquad \text{(by induction)} \\ &= \left(\frac{1}{C} + \frac{1}{(1-2\beta)}\right)\beta\sigma^k \\ &\overset{(11)}{\le} 2\beta\sigma^k, \end{aligned}$$

which proves the second estimate of (12). □

Theorem 2 shows that $\{\mathbf{x}^k\}$ linearly converges to $\mathbf{x}^\star$ with a contraction factor $\sigma \in (0,1)$ chosen from (11). Figure 1 shows the feasible region of (β, σ) for (11) when $C = 10$. From this figure, we can see that (11) will always hold once β is sufficiently small. Therefore, theoretically, we can let β arbitrarily close to 0.

4.3 Overall LMO complexity analysis

This subsection focuses on the analysis of LMO complexity of Algorithm 1. We first show that Algorithm 1 needs $\mathcal{O}\big(\varepsilon^{-2(1+\nu)}\big)$ LMO calls to reach an ε-solution defined by (3) where $\nu := \frac{\log(1-2\beta)}{\log(\sigma)}$. Consequently, we can show that it needs $\mathcal{O}\big(\varepsilon^{-(1+\nu)}\big)$-LMO calls to find an ε-solution $\mathbf{x}_\varepsilon^\star$ such that $f(\mathbf{x}_\varepsilon^\star) - f^\star \le \varepsilon$.

Theorem 3 *Suppose that $\|\mathbf{x}^0 - \mathbf{x}^\star\|_{\mathbf{x}^\star} \le \beta$. If we choose the parameters β, σ, C, and $\{\eta_k\}$ as in Theorem 2 and update $\{\mathbf{x}^k\}$ by the full-step stage, then to obtain an ε-solution $\mathbf{x}_\varepsilon^\star := \mathbf{x}^k$ defined by (3), it requires*

$$\begin{cases} \mathcal{O}\left(\log(\varepsilon^{-1})\right) & \text{gradient evaluations } \nabla f(\mathbf{x}^k), \\ \mathcal{O}\left(\log(\varepsilon^{-1})\right) & \text{Hessian evaluations} \nabla^2 f(\mathbf{x}^k), \text{ and} \\ \mathcal{O}\left(\varepsilon^{-2(1+\nu)}\right) & \text{LMO calls, with } \nu := \frac{\log(1-2\beta)}{\log(\sigma)}. \end{cases}$$

Proof By self-concordance of f, using [32, Theorem 4.1.6], it holds that

$$\nabla^2 f(\mathbf{x}^{k+1}) \preceq \frac{\nabla^2 f(\mathbf{x}^k)}{(1 - \|\mathbf{x}^{k+1} - \mathbf{x}^k\|_{\mathbf{x}^k})^2} \overset{(12)}{\preceq} \frac{\nabla^2 f(\mathbf{x}^k)}{(1-2\beta\sigma^k)^2} \preceq \frac{\nabla^2 f(\mathbf{x}^k)}{(1-2\beta)^2}.$$

By induction, we have

$$\nabla^2 f(\mathbf{x}^k) \preceq \left(\frac{1}{1-2\beta}\right)^{2k} \nabla^2 f(\mathbf{x}^0).$$

Therefore, we can bound the maximum eigenvalue $\lambda_{\max}(\nabla^2 f(\mathbf{x}^k))$ of $\nabla^2 f(\mathbf{x}^k)$ as

$$\lambda_{\max}(\nabla^2 f(\mathbf{x}^k)) \le \left(\frac{1}{1-2\beta}\right)^{2k} \lambda_{\max}(\nabla^2 f(\mathbf{x}^0)). \tag{13}$$

Let us denote by $\hat{\lambda}_0 := \lambda_{\max}(\nabla^2 f(\mathbf{x}^0))$. Then, from Lemma 6 in the "Appendix", we can see that the number of LMO calls at the k-th outer iteration is at most

$$\mathcal{O}_k := \frac{6\lambda_{\max}(\nabla^2 f(\mathbf{x}^k)) D_{\mathcal{X}}^2}{\eta_k^2} \overset{(13)}{\le} \frac{6\hat{\lambda}_0 D_{\mathcal{X}}^2}{(1-2\beta)^{2k}\eta_k^2} = \frac{6C^2\hat{\lambda}_0 D_{\mathcal{X}}^2}{\beta^2((1-2\beta)\sigma)^{2k}}, \tag{14}$$

where the last equality holds because we set $\eta_k := \beta\sigma^k/C$ in Theorem 2.

To obtain an ε-solution $\mathbf{x}^k$ defined by (3), we need to impose $\beta\sigma^k \le \varepsilon$ (recall that $\|\mathbf{x}^k - \mathbf{x}^\star\|_{\mathbf{x}^\star} \le \beta\sigma^k$ by Theorem 2), which is equivalent to $k \ge \frac{\log(\beta/\varepsilon)}{\log(1/\sigma)}$. Since $\beta \in (0,1)$, the outer iteration number is at most $\frac{\log(1/\varepsilon)}{\log(1/\sigma)} = \frac{\log(\varepsilon)}{\log(\sigma)}$. This number is also the total number of gradient and Hessian evaluations.

Finally, by (14), the total number of LO calls of Algorithm 1 is estimated as

$$\begin{aligned}\sum_{k=0}^{\frac{\log(\varepsilon)}{\log(\sigma)}} \frac{6C^2\hat{\lambda}_0 D_{\mathcal{X}}^2}{\beta^2((1-2\beta)\sigma)^{2k}} &= \frac{6C^2\hat{\lambda}_0 D_{\mathcal{X}}^2}{\beta^2} \sum_{k=0}^{\frac{\log(\varepsilon)}{\log(\sigma)}} \left(\frac{1}{(1-2\beta)\sigma}\right)^{2k} \\ &\leq \frac{3C^2\hat{\lambda}_0 D_{\mathcal{X}}^2}{\beta^3} \left(\frac{1}{(1-2\beta)\sigma}\right)^{\frac{2\log(\varepsilon)}{\log(\sigma)}} \\ &= \frac{3C^2\hat{\lambda}_0 D_{\mathcal{X}}^2}{\beta^3} \left(\frac{1}{\varepsilon}\right)^{2\left(1+\frac{\log(1-2\beta)}{\log(\sigma)}\right)},\end{aligned}$$

where the last equality holds since $\tau^{\alpha\log(s)} = s^{\alpha\log(\tau)}$. □

From Theorem 3, we can observe that a small value of β gives a better oracle complexity bound, but increases the number of oracle calls in the damped-step stage. Hence, we need to trade-off between the damped-step stage and the full-step stage. In practice, we do not recommend to choose an extremely small β but some value in the range of $[0.01, 0.1]$.

Finally, the following theorem states the LMO complexity of Algorithm 1 on the objective residuals.

Theorem 4 *Suppose that $\|\mathbf{x}^0 - \mathbf{x}^\star\|_{\mathbf{x}^\star} \leq \beta$. If we choose σ, β, C, and $\{\eta_k\}$ as in Theorem 2 and update $\{\mathbf{x}^k\}$ by the full-step stage, then we have*

$$f(\mathbf{x}^{k+1}) - f(\mathbf{x}^\star) \leq \left(\frac{12\beta^3}{1-2\beta} + \frac{\beta^2}{C^2} + \beta^2\right)\sigma^{2k}.$$

Consequently, the total LMO complexity of Algorithm 1 to achieve an ε-solution $\mathbf{x}_\varepsilon^\star := \mathbf{x}^k$ such that $f(\mathbf{x}_\varepsilon^\star) - f^\star \leq \varepsilon$ is $\mathcal{O}\left(\varepsilon^{-(1+\nu)}\right)$, where $\nu := \frac{\log(1-2\beta)}{\log(\sigma)}$.

Proof It is easy to check that $\omega_*(\tau) \leq \tau^2$ for $0 < \tau < 0.5$. Therefore, $\omega_*(\beta\sigma^k) \leq (\beta\sigma^k)^2$ for $k \geq 0$. Since $\eta_k := \frac{\beta\sigma^k}{C}$, $\gamma_k \leq 2\beta\sigma^k$, and $\bar{\lambda}_k := \|\mathbf{x}^k - \mathbf{x}^\star\|_{\mathbf{x}^\star} \leq \beta\sigma^k$ in Theorem 2, for $k \geq k_0$, we have

$$\begin{aligned}f(\mathbf{x}^{k+1}) - f(\mathbf{x}^\star) &\overset{(44)}{\leq} \frac{\gamma_k^2(\gamma_k+\bar{\lambda}_k)}{1-\gamma_k} + \eta_k^2 + \omega_*(\bar{\lambda}_{k+1}) \\ &\overset{(12)}{\leq} \frac{12\beta^3\sigma^{3k}}{1-2\beta\sigma^k} + \frac{\beta^2\sigma^{2k}}{C^2} + \omega_*(\beta\sigma^{k+1}) \\ &\leq \left(\frac{12\beta^3\sigma^k}{1-2\beta\sigma^k} + \frac{\beta^2}{C^2} + \beta^2\sigma^2\right)\sigma^{2k} \\ &\leq \left(\frac{12\beta^3}{1-2\beta} + \frac{\beta^2}{C^2} + \beta^2\right)\sigma^{2k}.\end{aligned} \tag{15}$$

Let $C_1 > \frac{12\beta^2}{1-2\beta} + \frac{\beta^2}{C^2} + \beta^2$ be a constant. To guarantee $f(\mathbf{x}^{k+1}) - f(\mathbf{x}^\star) \leq \varepsilon$, we impose $C_1\sigma^{2k} \leq \varepsilon$ i.e. $k \geq \frac{\log(\varepsilon/C_1)}{2\log(\sigma)}$. Therefore, the outer iteration number is at most $\frac{\log(\varepsilon/C_1)}{2\log(\sigma)} + 1$. Using (14), the total number of LMO calls will be

$$\begin{aligned}\mathcal{T}_2 := \sum_{k=0}^{\frac{\log(\varepsilon/C_1)}{2\log(\sigma)}+1} \frac{6C^2\lambda_{\max}(\nabla^2 f(\mathbf{x}^0))D_{\mathcal{X}}^2}{\beta^2((1-2\beta)\sigma)^{2k}} &= \mathcal{O}\left(\sum_{k=0}^{\frac{\log(\varepsilon/C_1)}{2\log(\sigma)}+1} \left(\frac{1}{(1-2\beta)\sigma}\right)^{2k}\right) \\ = \mathcal{O}\left(\left(\frac{1}{(1-2\beta)\sigma}\right)^{\frac{\log(\varepsilon/C_1)}{\log(\sigma)}}\right) &= \mathcal{O}\left(\left(\frac{1}{\varepsilon}\right)^{\frac{\log((1-2\beta)\sigma)}{\log(\sigma)}}\right),\end{aligned} \tag{16}$$

where the last equality follows from the fact that $\tau^{\alpha\log(s)} = s^{\alpha\log(\tau)}$. □

4.4 Trade-off between the damped-step and full-step stages

Fix $\beta \in [0, 0.1]$, let us choose

$$C = \frac{1}{\sigma}\frac{2(1-2\beta)(1-\beta)}{2(1-2\beta)(1-\beta)^2 - 1} > 0, \quad \text{and} \quad \sigma = 2\beta.$$

It is easy to verify that (11) still holds. The overall complexity in Theorem 4 becomes $\mathcal{O}\left(\varepsilon^{-(1+\nu)}\right) = \mathcal{O}\left(\varepsilon^{-(1+\frac{\log(1-2\beta)}{\log(2\beta)})}\right)$. Here, since $\beta \in (0, 0.1]$, we have $\nu := \frac{\log(1-2\beta)}{\log(2\beta)} \leq 0.139$. As a concrete example, if we choose $\beta := 0.05$, then the conditions (11) of Theorem 2 hold if we choose $(C, \sigma) = (27.3814, 0.1)$. In this case, $\nu := \frac{\log(1-2\beta)}{\log(\sigma)} = 0.0458$ which is very close to zero.

Now we show that the LMO complexity of the full-step stage: $\mathcal{T}_2$ in (16) dominates the LMO complexity of the damped-step stage: $\mathcal{T}_1$ in (10). Let us choose $\delta := \varepsilon$. Then, the step-size of the damped-step stage becomes $\alpha_k = \frac{\varepsilon(\gamma_k^2 - \eta_k^2)}{\gamma_k^3 + \gamma_k^2 - \eta_k^2\gamma_k}$, which is proportional to ε. In this case, the number of iterations K of the damped-step stage in Theorem 1 is

$$K = \frac{R}{\varepsilon} = \mathcal{O}\left(\frac{1}{\varepsilon}\right), \quad \text{where } R := \frac{f(\mathbf{x}^0) - f(\mathbf{x}^\star)}{\omega\left(\frac{1-2C_1}{1-C_1}h^{-1}(\beta)\right)} \text{ is a fixed constant.}$$

Moreover, for a sufficiently small ε, we have $(1-\delta)^{2K} = (1-\varepsilon)^{\frac{2R}{\varepsilon}} = \Theta\left(\frac{1}{e^{2R}}\right)$. Hence, by Theorem 1, the total LMO calls of the damped-step stage can be bounded by

$$\mathcal{T}_1 := \mathcal{O}\left(\frac{1}{\delta(1-\delta)^{2K}}\right) = \mathcal{O}\left(\frac{e^{2R}}{\varepsilon}\right) = \mathcal{O}\left(\frac{1}{\varepsilon}\right).$$

Therefore, the LMO complexity $\mathcal{T}_2 := \mathcal{O}\left(\varepsilon^{-(1+\nu)}\right)$ in the full-step stage dominates the one $\mathcal{T}_1 = \mathcal{O}\left(\varepsilon^{-1}\right)$ in the damped-step stage. Overall, the total complexity of Algorithm 1 is $\mathcal{O}\left(\varepsilon^{-(1+\nu)}\right)$, as stated in Theorem 4.

5 Numerical experiments

We provide three numerical examples to illustrate the performance of Algorithm 1. We emphasize that the objective function f of the first two examples is not globally L-smooth. Hence, existing Frank–Wolfe and projected gradient-based methods may not have theoretical guarantees. In the following experiments, we implement Algorithms 1 and its competitors in Matlab running on a Linux desktop with 3.6GHz Intel Core i7-7700 and 16Gb memory. Our code is available at https://github.com/unc-optimization/FWPN.

5.1 Portfolio optimization

Consider the following portfolio optimization model studied in [40, Section 6.4]:

$$\begin{cases} \min\limits_{\mathbf{x}\in\mathbb{R}^p}\left\{f(\mathbf{x}) := -\sum_{i=1}^n \log(\mathbf{a}_i^\top\mathbf{x})\right\} \\ \text{s.t.} \ \ \sum_{i=1}^p \mathbf{x}_i = 1, \ \mathbf{x} \geq 0, \end{cases} \tag{17}$$

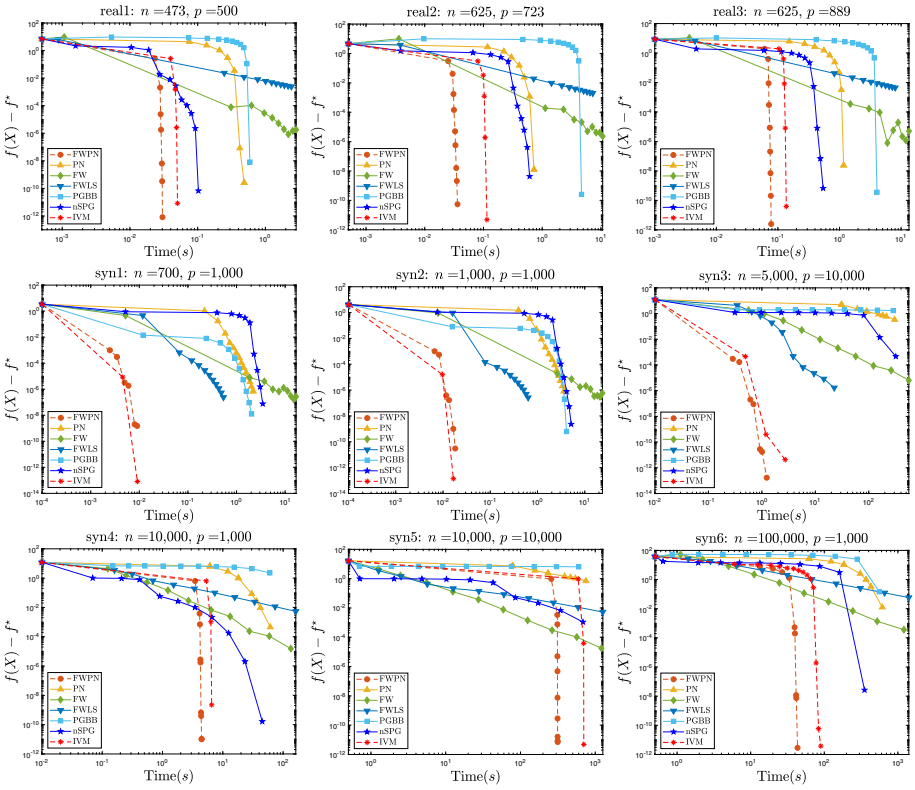

Fig. 2 A comparison between 7 methods for solving problem (17) on 9 datasets

where $\mathbf{a}_i \in \mathbb{R}^p$ for $i = 1, \cdots, n$. Let $\mathbf{A} := [\mathbf{a}_1, \cdots, \mathbf{a}_n]^\top \in \mathbb{R}^{n \times p}$. In the portfolio optimization model, $\mathbf{A}_{ij}$ represents the return of stock j in scenario i and $\log(\cdot)$ is the utility function. Our goal is to allocate assets to different stock companies to maximize the expected return.

We implement Algorithm 1, abbreviated by FWPN, to solve (17). We also implement the standard projected Newton method which uses accelerated projected gradient method to compute the search direction, the Frank–Wolfe algorithm [14] and its linesearch variant [26], the projected gradient method using Barzilai-Borwein's step-size [1,38], the nonmonotone spectral projected gradient method [6], and the inexact variable metric method [18] to solve this problem. We name these algorithms by PN, FW, FW-LS, PG-BB, nSPG, and IVM, respectively. For PN and PG-BB, we use the algorithm in [7] to compute the projection onto the simplex set.

We test these algorithms both on synthetic and real data. For the real data, we download three US stock datasets from http://www.excelclout.com/historical-stock-prices-in-excel/. We name these datasets by `real1`, `real2`, and `real3`. We generate synthetic datasets as follows. We generate a matrix $\mathbf{A}$ as $\mathbf{A} := \text{ones}(n, p) + \mathcal{N}(0, 0.1)$ which allows each stock to vary about 10% among scenarios. We test with six examples, where$(n, p) = (7 \times 10^2, 10^3)$, $(10^3, 10^3)$, $(5 \times 10^3, 10^4)$, $(10^4, 10^3)$, $(10^4, 10^4)$, and $(10^5, 10^3)$, respectively. We call these six datasets `syn1`, `syn2`, `syn3`, `syn4`, `syn5`, and `syn6`, respectively. The results and the performance of these six algorithms are shown in Fig. 2.

From Fig. 2, one can observe that our algorithm, FWPN, clearly outperforms the other competitors on both real and synthetic datasets. In our algorithm, we use a Frank–Wolfe method with away-step to solve the simplex constrained quadratic subproblem which has a linear convergence rate as proved in [28]. As we can see from Fig. 2, PGBB, nSPG, and PN work relatively well compared to other candidates on the real datasets. nSPG works quite well if the data is well-conditioned (see the plots of the `syn4` and `syn6` datasets) but will perform poorly if the condition number is large (see the plots of `syn3` and `syn5` datasets). Also notice that the IVM method is slightly worse than our FWPN method in most cases. In fact, both methods have the same subproblem, and we also apply the same subsolver to both methods. However, due to different stepsize strategies, their performance is not identical. As expected, the standard FW and its linesearch variant cannot reach a highly accurate solution.

5.2 *D*-optimal experimental design

Our second example in this section is the following convex optimization model in D-optimal experimental design:

$$\begin{cases} \min\limits_{\mathbf{x}\in\mathbb{R}^p} \left\{ f(\mathbf{x}) := -\log\det(\mathbf{A}\mathbf{X}\mathbf{A}^\top) \right\} \\ \text{s.t.} \quad \sum_{j=1}^p \mathbf{x}_j = 1, \ \mathbf{x} \geq 0, \end{cases} \tag{18}$$

where $\mathbf{A} := [\mathbf{a}_1, \cdots, \mathbf{a}_p] \in \mathbb{R}^{n\times p}$, $\mathbf{X} := \mathrm{Diag}(\mathbf{x})$, and $\mathbf{a}_i \in \mathbb{R}^n$ for $i = 1, \cdots, p$. It is well-known that the dual problem of (18) is the minimum-volume enclosing ellipsoid (MVEE) problem:

$$\begin{cases} \min\limits_{\mathbf{H}\succeq 0} \left\{ g(\mathbf{H}) := -\log\det(\mathbf{H}) \right\} \\ \text{s.t.} \quad \mathbf{a}_i^\top \mathbf{H}\mathbf{a}_i \leq n, \quad i = 1, \cdots, p. \end{cases} \tag{19}$$

The objective of this problem is to find the minimum ellipsoid that covers the points $\mathbf{a}_1, \cdots, \mathbf{a}_p \in \mathbb{R}^n$. The datasets $\{\mathbf{a}_i\}_{i=1}^p$ are generated using independent multinomial Gaussian distribution $\mathcal{N}(\mathbf{0}, \mathbf{\Sigma})$ as in [9]. To solve (18), one state-of-the-art solver is the Frank–Wolfe algorithm with away-step [28]. We observe that the linesearch problem for computing the optimal step-size τ:

$$\min_{\tau\in[0,1]} f((1-\tau)\mathbf{x} + \tau\mathbf{e}_j)$$

has a closed-form solution (see [27] for more details). Therefore, we do not have to carry out a linesearch at each iteration of the Frank–Wolfe algorithm.

Recently, [9] showed that the Frank–Wolfe algorithm with away-step has a linear convergence rate for this specific problem. Figure 3 reveals the performance of our algorithm (FWPN), Frank–Wolfe algorithm with away-step, the nonmonotone spectral projected gradient method (nSPG) [6], and the inexact variable metric method (IVM) [18] on three datasets, where the dimension n varies from 100 to 5, 000. Note that existing literature only tested for problems with $n \leq 500$. As far as we are aware of, this is the first attempt to solve problem (18) with n up to 5, 000.

From Fig. 3, our method outperforms the other three competitors on both large and small datasets, including IVM. Figure 3 also shows that when the size of the problem is small, our algorithm is slightly better than the Frank–Wolfe method with away-step. However, when the size of the problem becomes large, our algorithm highly outperforms the Frank–Wolfe

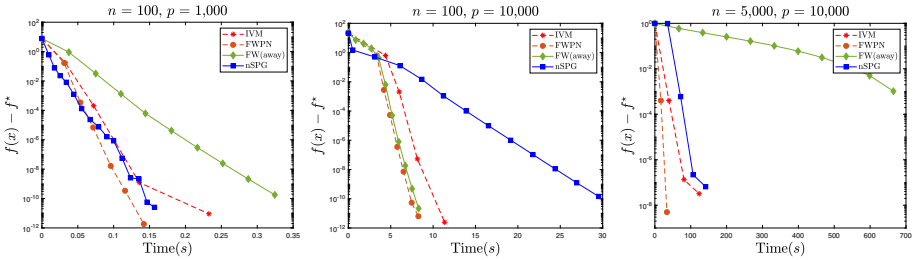

Fig. 3 A comparison between 4 algorithms for solving (18) on three datasets

method in terms of computational time. This happens due to a small number of projected Newton steps while each inner iteration requires significantly small computational time.

5.3 Logistic Regression with Elastic-net Regularizer

Finally, let us consider the following logistic regression with elastic-net regularizer:

$$\min_{\mathbf{x}\in\mathbb{R}^p} \left\{ \frac{1}{n}\mathbf{e}^\top \log(\mathbf{e} + \exp(\mathbf{A}^\top \mathbf{x})) + \frac{\mu}{2}\|\mathbf{x}\|^2 + \rho\|\mathbf{x}\|_1 \right\}, \tag{20}$$

where $\mathbf{e} := (1, 1, \cdots, 1)^\top \in \mathbb{R}^n$, $\mathbf{A} := [-y_1\mathbf{a}_1, \cdots, -y_n\mathbf{a}_n] \in \mathbb{R}^{p\times n}$, and $(\mathbf{a}_i, y_i) \in \mathbb{R}^p \times \{-1, 1\}$ for $i = 1, \cdots, n$.

It is well-known that (20) is the Lagrangian formulation of the following constrained problem with a suitable penalty parameter $\rho > 0$ [24, Section 3.4.2]. Although [24] only consider the standard linear regression problem, it is trivial to extend it to logistic regression of the form:

$$\begin{cases} \min\limits_{\mathbf{x}\in\mathbb{R}^p} \left\{ f(\mathbf{x}) := \frac{1}{n}\mathbf{e}^\top \log(\mathbf{e} + \exp(\mathbf{A}^\top \mathbf{x})) + \frac{\mu}{2}\|\mathbf{x}\|^2 \right\} \\ \text{s.t.} \quad \|\mathbf{x}\|_1 \le \rho_1. \end{cases} \tag{21}$$

It is has been shown in [40] that $f(\mathbf{x}) := \frac{1}{n}\mathbf{e}^\top \log(\mathbf{e}+\exp(\mathbf{A}^\top \mathbf{x})) + \frac{\mu}{2}\|\mathbf{x}\|^2$ is self-concordant. Therefore, (21) fits into our template (1) with $\mathcal{X} := \{\mathbf{x} \in \mathbb{R}^p \mid \|\mathbf{x}\|_1 \le \rho_1\}$.

For this example, the objective function f is also L-smooth and strongly convex. Hence, we can compare Algorithm 1 (FWPN) with the standard proximal-gradient method [4], the accelerated proximal-gradient method with linesearch and restart [5,39], the nonmonotone spectral projected gradient method [6], and the inexact variable metric method [18]. These methods are abbreviated by PG, APG-LSRS, nSPG, and IVM, respectively. We use binary classification datasets: **a1a**, **a9a**, **w1a**, **w8a**, **covtype**, **news20**, **real-sim** from [8] and generate the datasets **mnist17** and **mnist38** from the **mnist** dataset where digits are chosen from {1, 7} and {3, 8}, respectively. We set $\mu := \frac{1}{n}$ as in [10], and ρ_1 is set to be 10, which guarantees that the sparsity of the solution is maintained between 1% and 10%.

Since we need to evaluate the projection on an ℓ_1-norm ball at each iteration of PG and APG-LSRS, we use the algorithm provided by [12] which only need $\mathcal{O}(p)$ time. For our algorithm, since the ℓ_1-norm ball is still a polytope, we can linearly solve the subproblem by using the Frank–Wolfe algorithm with away-step from [28]. The performance and results of three algorithms on the above datasets are presented in Fig. 4 in terms of objective residuals against CPU time.

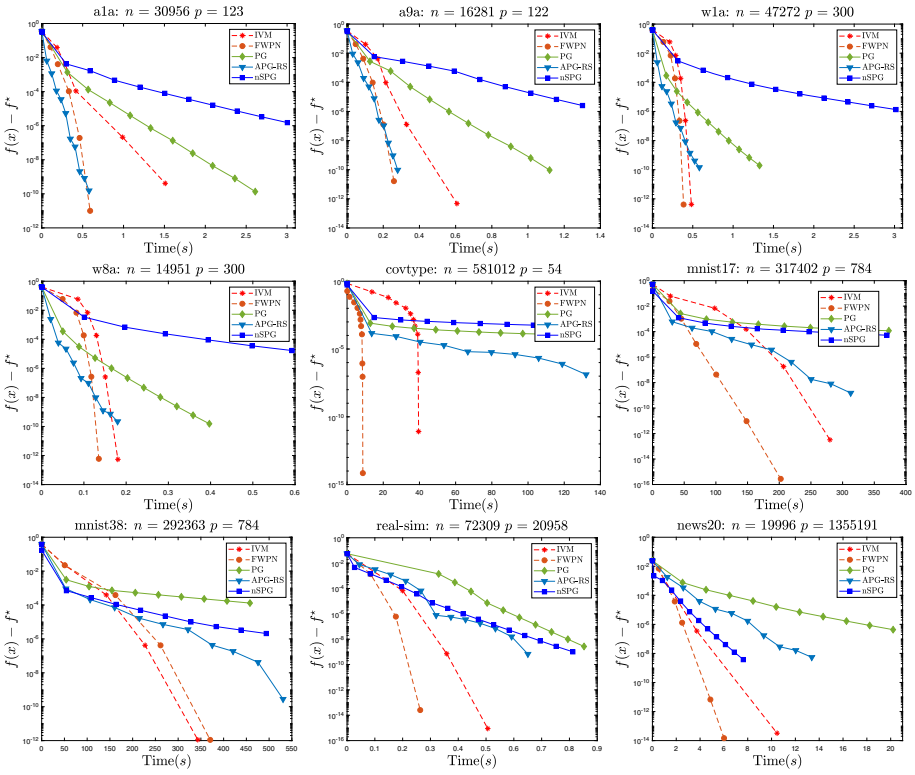

Fig. 4 A comparison between 5 methods for solving (21) on 9 different datasets

From Fig. 4, one can observe that our algorithm outperforms PG, APG-LSRS, and nSPG on all datasets. This happens thanks to the low computational cost of the linear minimization oracle and the linear convergence of the FW method with away-step. We also notice that our algorithm is still better than IVM on most datasets except for **mnist38**. It is interesting that although our algorithm is a hybrid method between second-order and first-order methods, we can still solve high-dimensional problems (e.g., when $p = 1,355,191$ in **news20** dataset) as often seen in first-order methods. We gain this efficiency due to the use of Hessian-vector products instead of full Hessian evaluations.

6 Conclusions

In this paper, we have combined the well-known Frank–Wolfe scheme (known as a variant of the conditional gradient method) and an inexact projected Newton (second-order) method to develop a novel hybrid algorithm for solving a class of constrained convex optimization problems with self-concordant objective function. Our approach is different from existing methods that heavily rely on the L-smooth assumption. Under this new setting, we have derived the first rigorous convergence and complexity analysis for the proposed method. Surprisingly, the LMO complexity of our algorithm is still comparable with the Frank–Wolfe algorithms for a different class of problems. In addition, our algorithm enjoys several

computation advantages on some specific problems, which are also supported by the three numerical examples in Sect. 5. Moreover, the last example has shown that our algorithm still outperforms first-order methods on large-scale instances. Our finding suggests that sometimes it is worth carefully combining first-order and second-order methods for solving large-scale problems in non-standard settings.

Acknowledgements Q. Tran-Dinh was partly supported by the National Science Foundation (NSF), Grant No. DMS-1619884 and the Office of Naval Research (ONR), Grant No. N00014-20-1-2088. V. Cevher was partly supported by the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation program (Grant agreement n 725594 - time-data) and by 2019 Google Faculty Research Award.

Appendix: The proof of technical results

Let us recall the following key properties of standard self-concordant functions. Let f be standard self-concordant and $\mathbf{x}, \mathbf{y} \in \mathrm{dom}(f)$ such that $\|\mathbf{y}-\mathbf{x}\|_{\mathbf{x}} < 1$. Then

$$\big(\|\mathbf{u}\|_{\mathbf{y}}\big)^2 := \mathbf{u}^\top\nabla^2 f(\mathbf{y})\mathbf{u} \le \mathbf{u}^\top\frac{\nabla^2 f(\mathbf{x})}{(1-\|\mathbf{y}-\mathbf{x}\|_{\mathbf{x}})^2}\mathbf{u} = \Big(\frac{\|\mathbf{u}\|_{\mathbf{x}}}{1-\|\mathbf{y}-\mathbf{x}\|_{\mathbf{x}}}\Big)^2, \quad \forall \mathbf{u}\in\mathbb{R}^p. \tag{22}$$

Similarly, if $\|\mathbf{y}-\mathbf{x}\|_{\mathbf{y}} < 1$, then

$$\big(\|\mathbf{u}\|_{\mathbf{y}}\big)^2 := \mathbf{u}^\top\nabla^2 f(\mathbf{y})\mathbf{u} \le \mathbf{u}^\top\frac{\nabla^2 f(\mathbf{x})}{(1-\|\mathbf{y}-\mathbf{x}\|_{\mathbf{y}})^2}\mathbf{u} = \Big(\frac{\|\mathbf{u}\|_{\mathbf{x}}}{1-\|\mathbf{y}-\mathbf{x}\|_{\mathbf{y}}}\Big)^2, \quad \forall \mathbf{u}\in\mathbb{R}^p. \tag{23}$$

These inequalities can be found in [32, Theorem 4.1.6]. In addition, from [43, equation (72)], we have

$$\|\nabla f(\mathbf{y}) - \nabla f(\mathbf{x}) - \nabla^2 f(\mathbf{x})(\mathbf{y}-\mathbf{x})\|^*_{\mathbf{x}} \le \frac{\|\mathbf{y}-\mathbf{x}\|^2_{\mathbf{x}}}{1-\|\mathbf{y}-\mathbf{x}\|_{\mathbf{x}}}, \tag{24}$$

if $\|\mathbf{y}-\mathbf{x}\|_{\mathbf{x}} < 1$. [4] These inequalities will be repeatedly used in our proofs below.

Two key lemmas for proving theorem 1

We need the following two lemmas to prove Theorem 1. The first lemma describes the decreasing of the objective value when applying damped-step iterations.

Lemma 3 *Let $\gamma_k := \|\mathbf{z}^k - \mathbf{x}^k\|_{\mathbf{x}^k}$ be the local distance between $\mathbf{z}^k$ to $\mathbf{x}^k$, where $\mathbf{z}^k$ is the output of Algorithm 2 at x^k with $\eta = \eta_k^2$. Recall that $\|\mathbf{z}^k - T(\mathbf{x}^k)\|_{\mathbf{x}^k} \le \eta_k$. If we choose $\alpha \in (0,1)$ such that $\alpha\gamma_k < 1$ and update $\mathbf{x}^{k+1} := \mathbf{x}^k + \alpha(\mathbf{z}^k - \mathbf{x}^k)$, then we have*

$$f(\mathbf{x}^{k+1}) \le f(\mathbf{x}^k) - \big[\alpha(\gamma_k^2 - \eta_k^2) - \omega_*(\alpha\gamma_k)\big]. \tag{25}$$

Assume $\gamma_k > \eta_k$. If $\delta \in (0,1)$ and the step size is $\alpha_k := \frac{\delta(\gamma_k^2-\eta_k^2)}{\gamma_k(\gamma_k^2+\gamma_k-\eta_k^2)}$ then we have $\alpha_k\gamma_k < \delta < 1$. Moreover, it also holds that

$$f(\mathbf{x}^{k+1}) \le f(\mathbf{x}^k) - \delta\omega\Big(\frac{\gamma_k^2-\eta_k^2}{\gamma_k}\Big), \tag{26}$$

where $\omega(\tau) := \tau - \log(1+\tau)$ and $\omega_*(\tau) := -\tau - \log(1-\tau)$ are two nonnegative and convex functions.

[4] One can see from the proof leading to [43, equation (72)] that the relation holds more generally when the $\mathbf{z}_+$ and $\mathbf{z}$ are replaced by any two vectors satisfying $\|\mathbf{y}-\mathbf{x}\|_{\mathbf{x}} \le 1$.

Proof From (7) and the stop criterion of Algorithm 2, $\mathbf{z}^k$ is an η_k-solution of (4) at $\mathbf{x} = \mathbf{x}^k$. It is clear that $\mathbf{z}^k$ satisfies

$$\langle \nabla f(\mathbf{x}^k) + \nabla^2 f(\mathbf{x}^k)(\mathbf{z}^k - \mathbf{x}^k), \mathbf{z}^k - \mathbf{x}^k \rangle \leq \eta_k^2.$$

This inequality leads to

$$\langle \nabla f(\mathbf{x}^k), \mathbf{z}^k - \mathbf{x}^k \rangle \leq \eta_k^2 - \|\mathbf{z}^k - \mathbf{x}^k\|_{\mathbf{x}^k}^2. \tag{27}$$

Therefore, using the self-concordance of f [32, Theorem 4.1.8], we can derive

$$\begin{array}{rcl} f(\mathbf{x}^{k+1}) & \leq & f(\mathbf{x}^k) + \left\langle \nabla f(\mathbf{x}^k), \mathbf{x}^{k+1} - \mathbf{x}^k \right\rangle + \omega_*(\|\mathbf{x}^{k+1} - \mathbf{x}^k\|_{\mathbf{x}^k}) \\ & = & f(\mathbf{x}^k) + \alpha \left\langle \nabla f(\mathbf{x}^k), \mathbf{z}^k - \mathbf{x}^k \right\rangle + \omega_*(\alpha \|\mathbf{z}^k - \mathbf{x}^k\|_{\mathbf{x}^k}) \\ & \overset{(27)}{\leq} & f(\mathbf{x}^k) + \alpha \left(\eta_k^2 - \|\mathbf{z}^k - \mathbf{x}^k\|_{\mathbf{x}^k}^2 \right) + \omega_*(\alpha \|\mathbf{z}^k - \mathbf{x}^k\|_{\mathbf{x}^k}) \\ & = & f(\mathbf{x}^k) - [\alpha(\gamma_k^2 - \eta_k^2) - \omega_*(\alpha\gamma_k)]. \end{array} \tag{28}$$

This is exactly (25).

Assume that $\gamma_k^2 > \eta_k^2$. Define $\psi(\alpha) := \alpha(\gamma_k^2 - \eta_k^2) - \omega_*(\alpha\gamma_k)$ and plug $\alpha_k = \frac{\delta(\gamma_k^2 - \eta_k^2)}{\gamma_k(\gamma_k^2 + \gamma_k - \eta_k^2)}$ into $\psi(\alpha)$, we arrive at

$$\begin{array}{rl} \psi(\alpha_k) & = \alpha_k(\gamma_k^2 - \eta_k^2) - \omega_*(\alpha_k\gamma_k) \\ & = \alpha_k(\gamma_k^2 - \eta_k^2 + \gamma_k) + \log(1 - \alpha_k\gamma_k) \\ & = \frac{\delta(\gamma_k^2 - \eta_k^2)}{\gamma_k} + \log\left(1 - \frac{\delta(\gamma_k^2 - \eta_k^2)}{\gamma_k^2 - \eta_k^2 + \gamma_k}\right) \\ & \geq \frac{\delta(\gamma_k^2 - \eta_k^2)}{\gamma_k} + \delta\log\left(1 - \frac{(\gamma_k^2 - \eta_k^2)}{\gamma_k^2 - \eta_k^2 + \gamma_k}\right) \\ & = \delta\omega(\frac{\gamma_k^2 - \eta_k^2}{\gamma_k}), \end{array} \tag{29}$$

where we use $\log(1 - \delta s) \geq \delta \log(1 - s)$ in $s \in (0, 1)$ for the inequality. Using (28) and (29) we proves (26). □

The following lemma shows that the residual $\|\mathbf{x}^k - \mathbf{x}^\star\|_{\mathbf{x}^\star}$ can be bounded by the projected Newton decrement $\bar{\gamma}_k := \|T(\mathbf{x}^k) - \mathbf{x}^k\|_{\mathbf{x}^k}$.

Lemma 4 *Let $\bar{\lambda}_k := \|\mathbf{x}^k - \mathbf{x}^\star\|_{\mathbf{x}^\star}$, $\bar{\gamma}_k := \|T(\mathbf{x}^k) - \mathbf{x}^k\|_{\mathbf{x}^k}$, $\gamma_k := \|\mathbf{z}^k - \mathbf{x}^k\|_{\mathbf{x}^k}$, and h be defined by (8). Recall that $\|\mathbf{z}^k - T(\mathbf{x}^k)\|_{\mathbf{x}^k} \leq \eta_k$. If $\gamma_k + \eta_k \in (0, C_2)$, then we have*

$$\bar{\lambda}_k \leq h(\bar{\gamma}_k) \leq h(\gamma_k + \eta_k). \tag{30}$$

Proof Firstly, we can write down the optimality condition of (4) and (1), respectively as follows:

$$\begin{cases} \left\langle \nabla f(\mathbf{x}^k) + \nabla^2 f(\mathbf{x}^k)[T(\mathbf{x}^k) - \mathbf{x}^k], \mathbf{x} - T(\mathbf{x}^k) \right\rangle \geq 0, \quad \forall \mathbf{x} \in \mathcal{X}, \\ \langle \nabla f(\mathbf{x}^\star), \mathbf{x} - \mathbf{x}^\star \rangle \geq 0, \quad \forall \mathbf{x} \in \mathcal{X}. \end{cases}$$

Substituting $\mathbf{x}^\star$ for $\mathbf{x}$ into the first inequality and $T(\mathbf{x}^k)$ for x into the second inequality, respectively we get

$$\begin{cases} \left\langle \nabla f(\mathbf{x}^k) + \nabla^2 f(\mathbf{x}^k)[T(\mathbf{x}^k) - \mathbf{x}^k], \mathbf{x}^\star - T(\mathbf{x}^k) \right\rangle \geq 0, \\ \left\langle \nabla f(\mathbf{x}^\star), T(\mathbf{x}^k) - \mathbf{x}^\star \right\rangle \geq 0. \end{cases}$$

Adding up both inequalities yields

$$\langle \nabla f(\mathbf{x}^\star) - \nabla f(\mathbf{x}^k) - \nabla^2 f(\mathbf{x}^k)[T(\mathbf{x}^k) - \mathbf{x}^k], T(\mathbf{x}^k) - \mathbf{x}^\star \rangle \geq 0,$$

which is equivalent to

$$\begin{aligned}&\langle \nabla f(T(\mathbf{x}^k)) - \nabla f(\mathbf{x}^k) - \nabla^2 f(\mathbf{x}^k)[T(\mathbf{x}^k) - \mathbf{x}^k], T(\mathbf{x}^k) - \mathbf{x}^\star\rangle \\ &\geq \langle \nabla f(T(\mathbf{x}^k)) - \nabla f(\mathbf{x}^\star), T(\mathbf{x}^k) - \mathbf{x}^\star\rangle.\end{aligned}$$

Since f is self-concordant, by [32, Theorem 4.1.7], we have

$$\left\langle \nabla f(T(\mathbf{x}^k)) - \nabla f(\mathbf{x}^\star), T(\mathbf{x}^k) - \mathbf{x}^\star \right\rangle \geq \frac{\|T(\mathbf{x}^k) - \mathbf{x}^\star\|^2_{T(\mathbf{x}^k)}}{1 + \|T(\mathbf{x}^k) - \mathbf{x}^\star\|_{T(\mathbf{x}^k)}}.$$

By the Cauchy-Schwarz inequality, this estimate leads to

$$\frac{\|T(\mathbf{x}^k) - \mathbf{x}^\star\|_{T(\mathbf{x}^k)}}{1 + \|T(\mathbf{x}^k) - \mathbf{x}^\star\|_{T(\mathbf{x}^k)}} \leq \|\nabla f(T(\mathbf{x}^k)) - \nabla f(\mathbf{x}^k) - \nabla^2 f(\mathbf{x}^k)[T(\mathbf{x}^k) - \mathbf{x}^k]\|^*_{T(\mathbf{x}^k)}. \quad (31)$$

Now, we can bound the right-hand side of the above inequality as

$$\begin{aligned}\mathcal{R} &:= \|\nabla f(T(\mathbf{x}^k)) - \nabla f(\mathbf{x}^k) - \nabla^2 f(\mathbf{x}^k)[T(\mathbf{x}^k) - \mathbf{x}^k]\|^*_{T(\mathbf{x}^k)} \\ &\leq \frac{\|\nabla f(T(\mathbf{x}^k)) - \nabla f(\mathbf{x}^k) - \nabla^2 f(\mathbf{x}^k)[T(\mathbf{x}^k) - \mathbf{x}^k]\|^*_{\mathbf{x}^k}}{1 - \|T(\mathbf{x}^k) - \mathbf{x}^k\|_{\mathbf{x}^k}} \\ &\overset{(24)}{\leq} \left(\frac{\|T(\mathbf{x}^k) - \mathbf{x}^k\|_{\mathbf{x}^k}}{1 - \|T(\mathbf{x}^k) - \mathbf{x}^k\|_{\mathbf{x}^k}}\right)^2,\end{aligned} \quad (32)$$

where the first inequality comes from the dual form of (23), i.e., $\frac{\|\mathbf{u}\|^*_{\mathbf{y}}}{1-\|\mathbf{y}-\mathbf{x}\|_{\mathbf{y}}} \geq \|\mathbf{u}\|^*_{\mathbf{x}}$ for $\mathbf{u} \in \mathbb{R}^p$, [5] and the last term holds since $\|T(\mathbf{x}^k) - \mathbf{x}^k\|_{\mathbf{x}^k} \leq \gamma_k + \eta_k \leq C_2 < 0.5$.

From (31) and (32), we have

$$\frac{\|T(\mathbf{x}^k) - \mathbf{x}^\star\|_{T(\mathbf{x}^k)}}{1 + \|T(\mathbf{x}^k) - \mathbf{x}^\star\|_{T(\mathbf{x}^k)}} \leq \left(\frac{\|T(\mathbf{x}^k) - \mathbf{x}^k\|_{\mathbf{x}^k}}{1 - \|T(\mathbf{x}^k) - \mathbf{x}^k\|_{\mathbf{x}^k}}\right)^2,$$

which can be reformulated as

$$\|T(\mathbf{x}^k) - \mathbf{x}^\star\|_{T(\mathbf{x}^k)} \leq \frac{\|T(\mathbf{x}^k) - \mathbf{x}^k\|^2_{\mathbf{x}^k}}{1 - 2\|T(\mathbf{x}^k) - \mathbf{x}^k\|_{\mathbf{x}^k}}. \quad (33)$$

Next, since we want to use $\|T(\mathbf{x}^k) - \mathbf{x}^k\|_{\mathbf{x}^k}$ to bound $\|\mathbf{x}^k - \mathbf{x}^\star\|_{\mathbf{x}^k}$, we can derive

$$\begin{aligned}\|\mathbf{x}^k - \mathbf{x}^\star\|_{\mathbf{x}^k} &\leq \|\mathbf{x}^k - T(\mathbf{x}^k)\|_{\mathbf{x}^k} + \|T(\mathbf{x}^k) - \mathbf{x}^\star\|_{\mathbf{x}^k} \\ &\overset{(23)}{\leq} \|\mathbf{x}^k - T(\mathbf{x}^k)\|_{\mathbf{x}^k} + \frac{\|T(\mathbf{x}^k) - \mathbf{x}^\star\|_{T(\mathbf{x}^k)}}{1 - \|\mathbf{x}^k - T(\mathbf{x}^k)\|_{\mathbf{x}^k}} \\ &\overset{(33)}{\leq} \|\mathbf{x}^k - T(\mathbf{x}^k)\|_{\mathbf{x}^k} + \frac{\|\mathbf{x}^k - T(\mathbf{x}^k)\|^2_{\mathbf{x}^k}}{\left(1 - 2\|\mathbf{x}^k - T(\mathbf{x}^k)\|_{\mathbf{x}^k}\right)\left(1 - \|\mathbf{x}^k - T(\mathbf{x}^k)\|_{\mathbf{x}^k}\right)} \\ &= \bar{\gamma}_k + \frac{\bar{\gamma}_k^2}{(1 - 2\bar{\gamma}_k)(1 - \bar{\gamma}_k)}.\end{aligned} \quad (34)$$

Notice that (23) of the above inequality holds because of $\|\mathbf{x}^k - T(\mathbf{x}^k)\|_{\mathbf{x}^k} \leq \gamma_k + \eta_k \leq C_2 < 1$, where C_2 is a constant defined right after (8). Since h is monotonically increasing and $\bar{\gamma}_k \leq \gamma_k + \eta_k$, we finally get

$$\|\mathbf{x}^k - \mathbf{x}^\star\|_{\mathbf{x}^\star} \overset{(22)}{\leq} \frac{\|\mathbf{x}^k - \mathbf{x}^\star\|_{\mathbf{x}^k}}{1 - \|\mathbf{x}^k - \mathbf{x}^\star\|_{\mathbf{x}^k}} \overset{(34)}{\leq} \frac{\bar{\gamma}_k(1 - 2\bar{\gamma}_k + 2\bar{\gamma}_k^2)}{(1 - 2\bar{\gamma}_k)(1 - \bar{\gamma}_k)^2 - \bar{\gamma}_k^2} = h(\bar{\gamma}_k) \leq h(\gamma_k + \eta_k),$$

[5] In fact, by (23), we have $\nabla^2 f(\mathbf{y}) \preceq \frac{\nabla^2 f(\mathbf{x})}{(1-\|\mathbf{y}-\mathbf{x}\|_{\mathbf{y}})^2}$, which is equivalent to $\nabla^2 f(\mathbf{x})^{-1} \preceq \frac{\nabla^2 f(\mathbf{y})^{-1}}{(1-\|\mathbf{y}-\mathbf{x}\|_{\mathbf{y}})^2}$. Therefore, we have $\frac{\|\mathbf{u}\|^*_{\mathbf{y}}}{1-\|\mathbf{y}-\mathbf{x}\|_{\mathbf{y}}} \geq \|\mathbf{u}\|^*_{\mathbf{x}}$ for $\mathbf{u} \in \mathbb{R}^p$.

which proves (30). Notice that we can also prove $\|\mathbf{x}^k - \mathbf{x}^\star\|_{\mathbf{x}^k} < 1$ to justify (22) of the above inequality, by using (34) and $\bar{\gamma}_k \le C_2$. □

Key bounds for proving theorem 2

The following lemma shows that $\|\mathbf{x}^{k+1} - \mathbf{x}^\star\|_{\mathbf{x}^\star}$ and $\|\mathbf{x}^{k+1} - \mathbf{x}^k\|_{\mathbf{x}^k}$ can both be bounded by $\|\mathbf{x}^k - \mathbf{x}^\star\|_{\mathbf{x}^\star}$ when $\|\mathbf{x}^k - \mathbf{x}^\star\|_{\mathbf{x}^\star}$ is sufficiently small.

Lemma 5 *Suppose that $\bar{\lambda}_k := \|\mathbf{x}^k - \mathbf{x}^\star\|_{\mathbf{x}^\star} \le \beta$, where $\beta \in (0, 0.5)$ is chosen by Algorithm 1. Then, we have*

$$\bar{\lambda}_{k+1} \le \frac{\eta_k}{1-\bar{\lambda}_k} + \frac{\bar{\lambda}_k^2}{(1-\bar{\lambda}_k)^2(1-2\bar{\lambda}_k)}. \tag{35}$$

In addition, we can also bound $\|\mathbf{x}^{k+1} - \mathbf{x}^k\|_{\mathbf{x}^k}$ as follows:

$$\|\mathbf{x}^{k+1} - \mathbf{x}^k\|_{\mathbf{x}^k} \le \eta_k + \frac{\bar{\lambda}_k^2}{(1-2\bar{\lambda}_k)(1-\bar{\lambda}_k)} + \frac{\bar{\lambda}_k}{1-\bar{\lambda}_k}. \tag{36}$$

Proof Since we always choose full-step $\alpha_k = 1$, we have $\mathbf{x}^{k+1} = \mathbf{z}^k$. Therefore, $\|\mathbf{x}^{k+1} - T(\mathbf{x}^k)\|_{\mathbf{x}^k} = \|\mathbf{z}^k - T(\mathbf{x}^k)\|_{\mathbf{x}^k} \le \eta_k$, which leads to

$$\begin{array}{rcl} \bar{\lambda}_{k+1} & = & \|\mathbf{x}^{k+1} - \mathbf{x}^\star\|_{\mathbf{x}^\star} \le \|\mathbf{x}^{k+1} - T(\mathbf{x}^k)\|_{\mathbf{x}^\star} + \|T(\mathbf{x}^k) - \mathbf{x}^\star\|_{\mathbf{x}^\star} \\ & \overset{(23)}{\le} & \frac{\|\mathbf{x}^{k+1} - T(\mathbf{x}^k)\|_{\mathbf{x}^k}}{1-\|\mathbf{x}^k - \mathbf{x}^\star\|_{\mathbf{x}^\star}} + \frac{\|T(\mathbf{x}^k) - \mathbf{x}^\star\|_{\mathbf{x}^k}}{1-\|\mathbf{x}^k - \mathbf{x}^\star\|_{\mathbf{x}^\star}} \\ & \le & \frac{\eta_k}{1-\bar{\lambda}_k} + \frac{\|T(\mathbf{x}^k) - \mathbf{x}^\star\|_{\mathbf{x}^k}}{1-\bar{\lambda}_k}. \end{array} \tag{37}$$

Now, we bound $\|T(\mathbf{x}^k) - \mathbf{x}^\star\|_{\mathbf{x}^k}$ as follows. Firstly, the optimality conditions of (4) and (1) can be written as

$$\begin{cases} \langle \nabla f(\mathbf{x}^k) + \nabla^2 f(\mathbf{x}^k)(T(\mathbf{x}^k) - \mathbf{x}^k), \mathbf{x} - T(\mathbf{x}^k)\rangle \ge 0, \quad \forall \mathbf{x} \in \mathcal{X}, \\ \langle \nabla f(\mathbf{x}^\star), \mathbf{x} - \mathbf{x}^\star\rangle \ge 0, \quad \forall \mathbf{x} \in \mathcal{X}. \end{cases}$$

This can be rewritten equivalently to

$$\begin{cases} \langle \nabla^2 f(\mathbf{x}^k)[T(\mathbf{x}^k) - (\mathbf{x}^k - \nabla^2 f(\mathbf{x}^k)^{-1}\nabla f(\mathbf{x}^k))], \mathbf{x} - T(\mathbf{x}^k)\rangle \ge 0, \quad \forall \mathbf{x} \in \mathcal{X}, \\ \langle \nabla^2 f(\mathbf{x}^k)[\mathbf{x}^\star - (\mathbf{x}^\star - \nabla^2 f(\mathbf{x}^k)^{-1}\nabla f(\mathbf{x}^\star))], \mathbf{x} - \mathbf{x}^\star\rangle \ge 0, \quad \forall \mathbf{x} \in \mathcal{X}. \end{cases} \tag{38}$$

Similar to the proof of [2, Theorem 3.14], we can show that (38) is equivalent to

$$\begin{cases} T(\mathbf{x}^k) = \mathrm{proj}_{\mathcal{X}}^{\nabla^2 f(\mathbf{x}^k)}\left(\mathbf{x}^k - \nabla^2 f(\mathbf{x}^k)^{-1}\nabla f(\mathbf{x}^k)\right), \\ \mathbf{x}^\star \quad = \mathrm{proj}_{\mathcal{X}}^{\nabla^2 f(\mathbf{x}^k)}\left(\mathbf{x}^\star - \nabla^2 f(\mathbf{x}^k)^{-1}\nabla f(\mathbf{x}^\star)\right). \end{cases} \tag{39}$$

Using the nonexpansiveness of the projection operator [2, Chapter 4] we can derive

$$\begin{aligned}
\|T(\mathbf{x}^k)-\mathbf{x}^\star\|_{\mathbf{x}^k} &\overset{(39)}{=} \left\|\mathrm{proj}_{\mathcal{X}}^{\nabla^2 f(\mathbf{x}^k)}\left(\mathbf{x}^k-\nabla^2 f(\mathbf{x}^k)^{-1}\nabla f(\mathbf{x}^k)\right)\right. \\
&\quad \left. -\mathrm{proj}_{\mathcal{X}}^{\nabla^2 f(\mathbf{x}^k)}\left(\mathbf{x}^\star-\nabla^2 f(\mathbf{x}^k)^{-1}\nabla f(\mathbf{x}^\star)\right)\right\|_{\mathbf{x}^k} \\
&\leq \|\mathbf{x}^k-\mathbf{x}^\star-\nabla^2 f(\mathbf{x}^k)^{-1}(\nabla f(\mathbf{x}^k)-\nabla f(\mathbf{x}^\star))\|_{\mathbf{x}^k} \\
&= \|\nabla f(\mathbf{x}^\star)-\nabla f(\mathbf{x}^k)-\nabla^2 f(\mathbf{x}^k)(\mathbf{x}^\star-\mathbf{x}^k)\|_{\mathbf{x}^k}^* \\
&\overset{(24)}{\leq} \frac{\|\mathbf{x}^\star-\mathbf{x}^k\|_{\mathbf{x}^k}^2}{1-\|\mathbf{x}^\star-\mathbf{x}^k\|_{\mathbf{x}^k}} \\
&\overset{(22)}{\leq} \frac{\|\mathbf{x}^\star-\mathbf{x}^k\|_{\mathbf{x}^\star}^2}{(1-2\|\mathbf{x}^\star-\mathbf{x}^k\|_{\mathbf{x}^\star})(1-\|\mathbf{x}^\star-\mathbf{x}^k\|_{\mathbf{x}^\star})} \\
&= \frac{\bar{\lambda}_k^2}{(1-2\bar{\lambda}_k)(1-\bar{\lambda}_k)}.
\end{aligned} \tag{40}$$

We make the following two explanation for (40):

- In the second inequality of (40), $1-\|\mathbf{x}^\star-\mathbf{x}^k\|_{\mathbf{x}^k}$ in the denominator can be justified by $\|\mathbf{x}^\star-\mathbf{x}^k\|_{\mathbf{x}^k}<1$, which follows directly from (24) and our assumption that $\|\mathbf{x}^\star-\mathbf{x}^k\|_{\mathbf{x}^\star}\leq\beta<0.5$ stated at the beginning of this lemma.
- For the last inequality of (40), we first have $0<\|\mathbf{x}^\star-\mathbf{x}^k\|_{\mathbf{x}^k}\leq\frac{\|\mathbf{x}^\star-\mathbf{x}^k\|_{\mathbf{x}^\star}}{1-\|\mathbf{x}^\star-\mathbf{x}^k\|_{\mathbf{x}^\star}}<1$ by (22) and our assumption that $\|\mathbf{x}^\star-\mathbf{x}^k\|_{\mathbf{x}^\star}<0.5$. Since $\frac{t^2}{1-t}$ is increasing for $t\in(0,1)$, we can replace $\|\mathbf{x}^\star-\mathbf{x}^k\|_{\mathbf{x}^k}$ by $\frac{\|\mathbf{x}^\star-\mathbf{x}^k\|_{\mathbf{x}^\star}}{1-\|\mathbf{x}^\star-\mathbf{x}^k\|_{\mathbf{x}^\star}}$ to get the last inequality of (40).

Plugging (40) into (37), we get (35).

Finally, we note that

$$\begin{aligned}
\|\mathbf{x}^{k+1}-\mathbf{x}^k\|_{\mathbf{x}^k} &\leq \|\mathbf{x}^{k+1}-T(\mathbf{x}^k)\|_{\mathbf{x}^k}+\|\mathbf{x}^\star-T(\mathbf{x}^k)\|_{\mathbf{x}^k}+\|\mathbf{x}^k-\mathbf{x}^\star\|_{\mathbf{x}^k} \\
&\overset{(40)}{\leq} \|\mathbf{x}^{k+1}-T(\mathbf{x}^k)\|_{\mathbf{x}^k}+\frac{\bar{\lambda}_k^2}{(1-2\bar{\lambda}_k)(1-\bar{\lambda}_k)}+\|\mathbf{x}^k-\mathbf{x}^\star\|_{\mathbf{x}^k} \\
&\overset{(22)}{\leq} \eta_k+\frac{\bar{\lambda}_k^2}{(1-2\bar{\lambda}_k)(1-\bar{\lambda}_k)}+\frac{\|\mathbf{x}^k-\mathbf{x}^\star\|_{\mathbf{x}^\star}}{1-\|\mathbf{x}^k-\mathbf{x}^\star\|_{\mathbf{x}^\star}} \\
&= \eta_k+\frac{\bar{\lambda}_k^2}{(1-2\bar{\lambda}_k)(1-\bar{\lambda}_k)}+\frac{\bar{\lambda}_k}{1-\bar{\lambda}_k},
\end{aligned}$$

which proves (36). □

An intermediate lemma for proving theorem 3

Firstly, the following lemma establishes the sublinear convergence rate of the Frank–Wolfe gap in each outer iteration.

Lemma 6 *At the k-th outer iteration of Algorithm 1, if we run the Frank–Wolfe subroutine (7) to update $\mathbf{u}^t$, then, after T_k iterations, we have*

$$\min_{t=1,\cdots,T_k} V_k(\mathbf{u}^t)\leq\frac{6\lambda_{\max}(\nabla^2 f(\mathbf{x}^k))D_{\mathcal{X}}^2}{T_k+1}, \tag{41}$$

where $V_k(\mathbf{u}^t):=\max_{\mathbf{u}\in\mathcal{X}}\langle\nabla f(\mathbf{x}^k)+\nabla^2 f(\mathbf{x}^k)(\mathbf{u}^t-\mathbf{x}^k),\mathbf{u}^t-\mathbf{u}\rangle$. As a result, the number of LMO calls at the k-th outer iteration of Algorithm 1 is at most $O_k:=\frac{6\lambda_{\max}(\nabla^2 f(\mathbf{x}^k))D_{\mathcal{X}}^2}{\eta_k^2}$.

Proof Let $\phi_k(\mathbf{u}) = \langle \nabla f(\mathbf{x}^k), \mathbf{u} - \mathbf{x}^k \rangle + 1/2 \langle \nabla^2 f(\mathbf{x}^k)(\mathbf{u} - \mathbf{x}^k), \mathbf{u} - \mathbf{x}^k \rangle$ and $\{\mathbf{u}^t\}$ be generated by the Frank–Wolfe subroutine (7). Then, it is well-known that (see [26, Theorem 1]):

$$\phi_k(\mathbf{u}^t) - \phi_k^\star \leq \frac{2\lambda_{\max}(\nabla^2 f(\mathbf{x}^k)) D_{\mathcal{X}}^2}{t+1}. \tag{42}$$

Let $\mathbf{v}^t := \arg\min_{\mathbf{u} \in \mathcal{X}} \{\langle \nabla \phi_k(\mathbf{u}^t), \mathbf{u} \rangle\}$. Notice that

$$\begin{aligned}
\phi_k(\mathbf{u}^{t+1}) &= \min_{\tau \in [0,1]} \{\phi_k((1-\tau)\mathbf{u}^t + \tau \mathbf{v}^t)\} \leq \phi_k\left(\left(1 - \tfrac{2}{t+1}\right)\mathbf{u}^t + \tfrac{2}{t+1}\mathbf{v}^t\right) \\
&\leq \phi_k(\mathbf{u}^t) + \tfrac{2}{t+1}\langle \nabla \phi_k(\mathbf{u}^t), (\mathbf{v}^t - \mathbf{u}^t)\rangle + \tfrac{\lambda_{\max}(\nabla^2 f(\mathbf{x}^k))}{2}\left(\tfrac{2}{t+1}\right)^2 \|\mathbf{v}^t - \mathbf{u}^t\|^2 \\
&\leq \phi_k(\mathbf{u}^t) - \tfrac{2}{t+1} V_k(\mathbf{u}^t) + \tfrac{2\lambda_{\max}(\nabla^2 f(\mathbf{x}^k))}{(t+1)^2} D_{\mathcal{X}}^2.
\end{aligned}$$

This is equivalent to

$$t V_k(\mathbf{u}^t) \leq \frac{t(t+1)}{2}\left(\phi_k(\mathbf{u}^t) - \phi_k(\mathbf{u}^{t+1})\right) + \frac{t\lambda_{\max}(\nabla^2 f(\mathbf{x}^k))}{t+1} D_{\mathcal{X}}^2. \tag{43}$$

Summing up this inequality from $t = 1$ to T_k, we get

$$\begin{aligned}
\frac{T_k(T_k+1)}{2} \min_{t=1,\cdots,T_k} \left\{V_k(\mathbf{u}^t)\right\} &\leq \sum_{t=1}^{T_k} t V_k(\mathbf{u}^t) \\
&\overset{(43)}{\leq} \sum_{t=1}^{T_k} t\phi_k(\mathbf{u}^t) \\
&\quad - \tfrac{T_k(T_k+1)}{2}\phi_k(\mathbf{u}^{T_k+1}) + T_k\lambda_{\max}(\nabla^2 f(\mathbf{x}^k)) D_{\mathcal{X}}^2 \\
&\leq \sum_{t=1}^{T_k} t(\phi_k(\mathbf{u}^t) - \phi_k^\star) + T_k\lambda_{\max}(\nabla^2 f(\mathbf{x}^k)) D_{\mathcal{X}}^2 \\
&\overset{(42)}{\leq} 3T_k\lambda_{\max}(\nabla^2 f(\mathbf{x}^k)) D_{\mathcal{X}}^2,
\end{aligned}$$

which implies (41). □

An intermediate lemma for proving theorem 4

The following lemma states that we can bound $f(\mathbf{x}^k) - f^\star$ by $\|\mathbf{x}^{k+1} - \mathbf{x}^k\|_{\mathbf{x}^k}$ and $\|\mathbf{x}^k - \mathbf{x}^\star\|_{\mathbf{x}^\star}$. Therefore, from the convergence rate of $\|\mathbf{x}^{k+1} - \mathbf{x}^k\|_{\mathbf{x}^k}$ and $\|\mathbf{x}^k - \mathbf{x}^\star\|_{\mathbf{x}^\star}$ in Theorem 2, we can obtain a convergence rate of $\{f(\mathbf{x}^k) - f^\star\}$.

Lemma 7 *Let* $\gamma_k := \|\mathbf{x}^{k+1} - \mathbf{x}^k\|_{\mathbf{x}^k} = \|\mathbf{z}^k - \mathbf{x}^k\|_{\mathbf{x}^k}$ *and* $\bar{\lambda}_k := \|\mathbf{x}^k - \mathbf{x}^\star\|_{\mathbf{x}^\star}$. *Suppose that* $\mathbf{x}^0 \in \mathrm{dom}(f) \cap \mathcal{X}$. *If* $0 < \gamma_k, \bar{\lambda}_k, \bar{\lambda}_{k+1} < 1$, *then we have*

$$f(\mathbf{x}^{k+1}) \leq f(\mathbf{x}^\star) + \frac{\gamma_k^2(\gamma_k + \bar{\lambda}_k)}{1 - \gamma_k} + \eta_k^2 + \omega_*(\bar{\lambda}_{k+1}), \tag{44}$$

where $\omega_*(\tau) := -\tau - \log(1-\tau)$.

Proof Firstly, from [32, Theorem 4.1.8], we have

$$f(\mathbf{x}^{k+1}) \leq f(\mathbf{x}^\star) + \left\langle \nabla f(\mathbf{x}^\star), \mathbf{x}^{k+1} - \mathbf{x}^\star \right\rangle + \omega_*(\|\mathbf{x}^{k+1} - \mathbf{x}^\star\|_{\mathbf{x}^\star}),$$

provided that $\|\mathbf{x}^{k+1}-\mathbf{x}^\star\|_{\mathbf{x}^\star} < 1$. Next, using $\left\langle \nabla f(\mathbf{x}^\star) - \nabla f(\mathbf{x}^{k+1}), \mathbf{x}^{k+1}-\mathbf{x}^\star \right\rangle \le 0$, we can further derive

$$f(\mathbf{x}^{k+1}) \le f(\mathbf{x}^\star) + \left\langle \nabla f(\mathbf{x}^{k+1}), \mathbf{x}^{k+1}-\mathbf{x}^\star \right\rangle + \omega_*(\|\mathbf{x}^{k+1}-\mathbf{x}^\star\|_{\mathbf{x}^\star}). \tag{45}$$

Now, we bound $\left\langle \nabla f(\mathbf{x}^{k+1}), \mathbf{x}^{k+1}-\mathbf{x}^\star \right\rangle$ as follows. We first notice that this term can be decomposed as

$$\begin{aligned}\left\langle \nabla f(\mathbf{x}^{k+1}), \mathbf{x}^{k+1}-\mathbf{x}^\star \right\rangle = &\underbrace{\langle \nabla f(\mathbf{x}^k) + \nabla^2 f(\mathbf{x}^k)(\mathbf{x}^{k+1}-\mathbf{x}^k), \mathbf{x}^{k+1}-\mathbf{x}^\star\rangle}_{\mathcal{T}_1} \\ &+ \underbrace{\langle \nabla f(\mathbf{x}^{k+1}) - \nabla f(\mathbf{x}^k) - \nabla^2 f(\mathbf{x}^k)(\mathbf{x}^{k+1}-\mathbf{x}^k), \mathbf{x}^{k+1}-\mathbf{x}^\star\rangle}_{\mathcal{T}_2}.\end{aligned}$$

Since $\mathbf{x}^{k+1}$ is an η^k-solution of (4) at $\mathbf{x}=\mathbf{x}^k$, we have

$$\mathcal{T}_1 = \langle \nabla f(\mathbf{x}^k) + \nabla^2 f(\mathbf{x}^k)(\mathbf{x}^{k+1}-\mathbf{x}^k), \mathbf{x}^{k+1}-\mathbf{x}^\star\rangle \le \eta_k^2. \tag{46}$$

Using the Cauchy-Schwarz inequality and the triangle inequality, $\mathcal{T}_2$ can also be bounded as

$$\begin{aligned}\mathcal{T}_2 &= \left\langle \nabla f(\mathbf{x}^{k+1}) - \nabla f(\mathbf{x}^k) - \nabla^2 f(\mathbf{x}^k)(\mathbf{x}^{k+1}-\mathbf{x}^k), \mathbf{x}^{k+1}-\mathbf{x}^\star \right\rangle \\ &\le \|\nabla f(\mathbf{x}^{k+1}) - \nabla f(\mathbf{x}^k) - \nabla^2 f(\mathbf{x}^k)(\mathbf{x}^{k+1}-\mathbf{x}^k)\|_{\mathbf{x}^k}^* \|\mathbf{x}^{k+1}-\mathbf{x}^\star\|_{\mathbf{x}^k} \\ &\overset{(24)}{\le} \frac{\|\mathbf{x}^{k+1}-\mathbf{x}^k\|_{\mathbf{x}^k}^2}{1-\|\mathbf{x}^{k+1}-\mathbf{x}^k\|_{\mathbf{x}^k}} \|\mathbf{x}^{k+1}-\mathbf{x}^\star\|_{\mathbf{x}^k} \\ &\le \frac{\|\mathbf{x}^{k+1}-\mathbf{x}^k\|_{\mathbf{x}^k}^2}{1-\|\mathbf{x}^{k+1}-\mathbf{x}^k\|_{\mathbf{x}^k}} \left[\|\mathbf{x}^k-\mathbf{x}^\star\|_{\mathbf{x}^k} + \|\mathbf{x}^{k+1}-\mathbf{x}^k\|_{\mathbf{x}^k}\right] \\ &= \frac{\gamma_k^2(\gamma_k+\bar{\lambda}_k)}{1-\gamma_k}.\end{aligned} \tag{47}$$

Finally, we can bound $f(\mathbf{x}^{k+1}) - f^\star$ as

$$\begin{aligned}f(\mathbf{x}^{k+1}) - f(\mathbf{x}^\star) &\overset{(45)}{\le} \left\langle \nabla f(\mathbf{x}^{k+1}), \mathbf{x}^{k+1}-\mathbf{x}^\star \right\rangle + \omega_*(\bar{\lambda}_{k+1}) \\ &= \mathcal{T}_1 + \mathcal{T}_2 + \omega_*(\bar{\lambda}_{k+1}) \\ &\overset{(46)(47)}{\le} \eta_k^2 + \frac{\gamma_k^2(\gamma_k+\bar{\lambda}_k)}{1-\gamma_k} + \omega_*(\bar{\lambda}_{k+1}),\end{aligned}$$

which proves (44). □

References

1. Barzilai, J., Borwein, J.M.: Two-point step size gradient methods. IMA J. Numer. Anal. **8**(1), 141–148 (1988)
2. Bauschke, H.H., Combettes, P.: Convex Analysis and Monotone Operators Theory in Hilbert Spaces. Springer-Verlag, 2nd edn. (2017)
3. Beck, A., Teboulle, M.: A conditional gradient method with linear rate of convergence for solving convex linear systems. Math. Methods Oper. Res. **59**(2), 235–247 (2004)
4. Beck, A., Teboulle, M.: A fast iterative shrinkage-thresholding algorithm for linear inverse problems. SIAM J. Imaging Sci. **2**(1), 183–202 (2009)
5. Becker, S., Candès, E.J., Grant, M.: Templates for convex cone problems with applications to sparse signal recovery. Math. Program. Compt. **3**(3), 165–218 (2011)
6. Birgin, E.G., Martínez, J.M., Raydan, M.: Nonmonotone spectral projected gradient methods on convex sets. SIAM J. Optim. **10**(4), 1196–1211 (2000)
7. Chen, Y., Ye, X.: Projection onto a simplex. Preprint arXiv:1101.6081 (2011)
8. Chang, C.-C., Lin, C.-J.: LIBSVM, A library for support vector machines. ACM Trans. Intell. Syst. Technol. **2**(3), 1–27 (2011)

9. Damla, S.A., Sun, P., Todd, M.J.: Linear convergence of a modified Frank–Wolfe algorithm for computing minimum-volume enclosing ellipsoids. Optim. Methods Softw. **23**(1), 5–19 (2008)
10. Defazio, A., Bach, F., Lacoste-Julien, S.: SAGA: A fast incremental gradient method with support for non-strongly convex composite objectives. In Advances in Neural Information Processing Systems (NIPS), pp. 1646–1654 (2014)
11. de Oliveira, F.R., Ferreira, O.P., Silva, G.N.: Newton's method with feasible inexact projections for solving constrained generalized equations. Comput. Optim. Appl. **72**(1), 159–177 (2019)
12. Duchi, J., Shalev-Shwartz, S., Singer, Y., Chandra, T.: Efficient projections onto the ℓ_1-ball for learning in high dimensions. In Proceedings of the 25th International Conference on Machine Learning, ICML '08, pp. 272–279, New York, NY, USA, ACM (2008)
13. Dvurechensky, P., Ostroukhov, P., Safin, K., Shtern, S., Staudigl, M.: Self-concordant analysis of Frank-Wolfe algorithms. In International Conference on Machine Learning, pp. 2814–2824. PMLR, (2020)
14. Frank, M., Wolfe, P.: An algorithm for quadratic programming. Naval Res. Logist. Q. **3**, 95–110 (1956)
15. Garber, D., Hazan, E.: A linearly convergent conditional gradient algorithm with applications to online and stochastic optimization. Preprint arXiv:1301.4666 (2013)
16. Garber, D., Hazan, E.: Faster rates for the Frank–Wolfe method over strongly-convex sets. In Proceedings of the 32nd International Conference on Machine Learning 951, pp. 541–549 (2015)
17. Gonçalves, M.L.N., Melo, J.G.: A newton conditional gradient method for constrained nonlinear systems. J. Comput. Appl. Math. **311**, 473–483 (2017)
18. Gonçalves, D. S., Gonçalves, M. L. N., Menezes, T. C.: Inexact variable metric method for convex-constrained optimization problems. Optimization, 1–19, (online first) (2021)
19. Gonçalves, D. S., Gonçalves, M. L. N., Oliveira, F. R.: Levenberg-marquardt methods with inexact projections for constrained nonlinear systems. Preprint arXiv:1908.06118 (2019)
20. Gonçalves, M.L.N., Oliveira, F.R.: On the global convergence of an inexact quasi-Newton conditional gradient method for constrained nonlinear systems. Numer. Algorithm **84**(2), 606–631 (2020)
21. Gross, D., Liu, Y.-K., Flammia, S., Becker, S., Eisert, J.: Quantum state tomography via compressed sensing. Phys. Rev. Lett. **105**(15), 150401 (2010)
22. Guelat, J., Marcotte, P.: Some comments on Wolfe's away step. Math. Program. **35**(1), 110–119 (1986)
23. Harman, R., Trnovská, M.: Approximate D-optimal designs of experiments on the convex hull of a finite set of information matrices. Math. Slov. **59**(6), 693–704 (2009)
24. Hastie, T., Tibshirani, R., Friedman, J.: The Elements of Statistical Learning: Data Mining, Inference, and Prediction. Springer Science & Business Media (2009)
25. Hazan, E.: Sparse approximate solutions to semidefinite programs. In: Latin American Symposium on Theoretical Informatics, pp. 306–316. Springer (2008)
26. Jaggi, M.: Revisiting Frank–Wolfe: projection-free sparse convex optimization. JMLR W&CP **28**(1), 427–435 (2013)
27. Khachiyan, L.G.: Rounding of polytopes in the real number model of computation. Math. Oper. Res. **21**(2), 307–320 (1996)
28. Lacoste-Julien, S., Jaggi, M.: On the global linear convergence of Frank–Wolfe optimization variants. In Advances in Neural Information Processing Systems (NIPS), pp. 496–504 (2015)
29. Lan, G., Zhou, Y.: Conditional gradient sliding for convex optimization. SIAM J. Optim. **26**(2), 1379–1409 (2016)
30. Lan, G., Ouyang, Y.: Accelerated gradient sliding for structured convex optimization. Preprint arXiv:1609.04905 (2016)
31. Lu, Z., Pong, T.K.: Computing optimal experimental designs via interior point method. SIAM J. Matrix Anal. Appl. **34**(4), 1556–1580 (2013)
32. Nesterov, Y.: Introductory Lectures on Convex Optimization: A Basic Course volume 87 of Applied Optimization. Kluwer Academic Publishers (2004)
33. Nesterov, Y., Nemirovski, A.: Interior-point polynomial algorithms in convex programming. Soc. Ind. Math. (1994)
34. Odor, G., Li, Y.-H., Yurtsever, A., Hsieh, Y.-P., Tran-Dinh, Q., El-Halabi, M., Cevher, V.: Frank-Wolfe works for non-lipschitz continuous gradient objectives: Scalable poisson phase retrieval. In 2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 6230–6234. IEEE (2016)
35. Ostrovskii, D.M., Bach, F.: Finite-sample analysis of M-estimators using self-concordance. Electron. J. Stat. **15**(1), 326–391 (2021)
36. Peyré, G., Cuturi, M.: Computational optimal transport. Found. Trends Mach. Learn. **11**(5–6), 355–607 (2019)
37. Ryu, E. K., Boyd, S.: Stochastic proximal iteration: a non-asymptotic improvement upon stochastic gradient descent. Author website, early draft (2014)

38. Raydan, M.: On the Barzilai and Borwein choice of steplength for the gradient method. IMA J. Numer. Anal. **13**(3), 321–326 (1993)
39. Su, W., Boyd, S., Candes, E.: A differential equation for modeling Nesterov's accelerated gradient method: theory and insights. In Advances in Neural Information Processing Systems (NIPS), pp. 2510–2518 (2014)
40. Sun, T., Tran-Dinh, Q.: Generalized self-concordant functions: a recipe for Newton-type methods. Math. Program. **178**, 145–213 (2019)
41. Tran-Dinh, Q., Kyrillidis, A., Cevher, V.: Composite self-concordant minimization. J. Mach. Learn. Res. **15**, 374–416 (2015)
42. Tran-Dinh, Q., Ling, L., Toh, K.-C.: A new homotopy proximal variable-metric framework for composite convex minimization. Math. Oper. Res., 1–28, (online first) (2021)
43. Tran-Dinh, Q., Sun, T., Lu, S.: Self-concordant inclusions: a unified framework for path-following generalized Newton-type algorithms. Math. Program. **177**(1–2), 173–223 (2019)
44. Yurtsever, A., Fercoq, O., Cevher, V.: A conditional-gradient-based augmented lagrangian framework. In International Conference on Machine Learning (ICML), pp. 7272–7281 (2019)
45. Yurtsever, A., Tran-Dinh, Q., Cevher, V.: A universal primal-dual convex optimization framework. Advances in Neural Information Processing Systems (NIPS), pp. 1–9 (2015)

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.