Noname manuscript No.
(will be inserted by the editor)

The Complexity of Large-scale Convex Programming under a Linear Optimization Oracle

Guanghui Lan

the date of receipt and acceptance should be inserted later

Abstract This paper considers a general class of iterative optimization algorithms, referred to as linear-optimization-based convex programming (LCP) methods, for solving large-scale convex programming (CP) problems. The LCP methods, covering the classic conditional gradient (CG) method (a.k.a., Frank-Wolfe method) as a special case, can only solve a linear optimization subproblem at each iteration. In this paper, we first establish a series of lower complexity bounds for the LCP methods to solve different classes of CP problems, including smooth, nonsmooth and certain saddle-point problems. We then formally establish the theoretical optimality or nearly optimality, in the large-scale case, for the CG method and its variants to solve different classes of CP problems. We also introduce several new optimal LCP methods, obtained by properly modifying Nesterov's accelerated gradient method, and demonstrate their possible advantages over the classic CG for solving certain classes of large-scale CP problems.

Keywords: convex programming, complexity, conditional gradient method, Frank-Wolfe method, Nesterov's method

AMS 2000 subject classification: 90C25, 90C06, 90C22, 49M37

1 Introduction

The last few years have seen an increasing instrest in the application of convex programming (CP) models for machine learning, image processing, and polynomial optimization, etc. The CP problems arising from these applications, however, are often of high dimension and hence challenging to solve. In particular, they are generally beyond the capability of second-order interior-point methods due to the highly demanding iteration costs of these optimization techniques. This has motivated the currently active research on first-order methods which possess cheaper iteration costs for large-scale CP, including Nesterov's optimal method [26–28] and several stochastic first-order algorithms in [24, 21]. These optimization algorithms are relatively simple, and suitable for the situation when low or moderate solution accuracy is sought-after.

In this paper, we study a different class of optimization algorithms, referred to as *linear-optimization-based convex programming (LCP)* methods, for large-scale CP. Specifically, consider the CP problem of

$$f^* := \min_{x \in X} f(x), \tag{1.1}$$

where $X \subseteq \mathbb{R}^n$ is a convex compact set and $f : X \to \mathbb{R}$ is a closed convex function. The LCP methods solve problem (1.1) by iteratively calling a *linear optimization* (LO) oracle, which, for a given input vector $p \in \mathbb{R}^n$, computes the

The author of this paper was partially supported by NSF grant CMMI-1000347, ONR grant N00014-13-1-0036 and NSF CAREER Award CMMI-1254446.

Department of Industrial and Systems Engineering, University of Florida, Gainesville, FL, 32611. (email: glan@ise.ufl.edu).

Address(es) of author(s) should be given

solution of subproblems given in the form of

$$\text{Argmin}_{x\in X}\langle p, x\rangle. \tag{1.2}$$

In particular, if p is computed based on first-order information, then we call these algorithms *first-order LCP* methods. Clearly, the difference between first-order LCP methods and the more general first-order methods exists in the restrictions on the format of subproblems. For example, in the well-known subgradient (mirror) descent method [25] and Nesterov's method [26,27], we solve the projection (or prox-mapping) subproblems given in the form of

$$\text{argmin}_{x\in X}\left\{\langle p, x\rangle + d(x)\right\}. \tag{1.3}$$

Here $d : X \to \mathbb{R}$ is a certain strongly convex function (e.g., $d(x) = \|x\|_2^2/2$).

The development of LCP methods dates back to the conditional gradient (CG) method (a.k.a., Frank-Wolfe algorithm) developed by Frank and Wolfe in 1956 [13]. This method has recently regained some interests from both optimization and machine learning community (see, e.g., [1–3,7,17–19,23,32,33]) mainly for the following reasons.

- *Low iteration cost.* In many cases, the solution of the linear subproblem (1.2) is much easier to solve than the nonlinear subproblem (1.3). For example, if X is a spectahedron given by $X = \{x \in \mathbb{R}^{n\times n} : \text{Tr}(x) = 1, x \succeq 0\}$, the solution of (1.2) can be much faster than that of (1.3).
- *Simplicity.* The CG method is simple to implement since it does not require the selection of the distance function $d(x)$ in (1.3) and the fine-tuning of stepsizes, which are required in most other first-order methods (with exceptions to some extent for a few level-type first-order methods, see [4,22]).
- *Structural properties for the generated solutions.* The output solutions of the CG method may have certain desirable structural properties, e.g., sparsity and low rank, as they can often be written as the convex combination of a small number of extreme points of X.

Numerical studies (e.g., [17]) indicate that the CG method can be competitive to the more involved gradient-type methods for solving certain classes of CP problems. It is also worth noting that the CG method, when applied to the linear feasibility problems, is closely related to the von Neumann algorithm studied by Dantzig [8,9], and later in Epelman and Freund [12].

This paper focuses on the complexity analysis of CP under an LO oracle, as well as the development of new LCP methods for large-scale CP. Although there exists rich complexity theory for the general first-order methods for large-scale CP in the literature, the study on the complexity of CP under an LO oracle is still limited. More specifically, in view of the classic CP complexity theory [25,27], if f is a general nonsmooth Lipschitz continuous convex function such that

$$|f(x) - f(y)| \le M\|x - y\|,\ \forall x, y \in X, \tag{1.4}$$

then the number of iterations required by any first-order methods to find an ϵ-solution of (1.1), i.e., a point $\bar{x} \in X$ s.t. $f(\bar{x}) - f^* \le \epsilon$, cannot be smaller than $\mathcal{O}(1/\epsilon^2)$ if n is sufficiently large. In addition, if f is a general smooth convex function satisfying

$$\|f'(x) - f'(y)\|_* \le L\|x - y\|, \forall x, y \in X, \tag{1.5}$$

then the number of iterations required by any first-order methods to find an ϵ-solution of (1.1) cannot be smaller than $\mathcal{O}(1/\sqrt{\epsilon})$ if n is large enough. These lower complexity bounds can be achieved, for example, by the aforementioned subgradient (mirror) descend method and Nesterov's method, respectively, for nonsmooth and smooth convex optimization. In addition, in a recent breakthrough paper [28], Nesterov studied an important class of saddle point problems with f is given by

$$f(x) = \max_{y\in Y}\left\{\langle Ax, y\rangle - \hat{f}(y)\right\}. \tag{1.6}$$

Here $Y \subseteq \mathbb{R}^m$ is a convex compact set, $A : \mathbb{R}^n \to \mathbb{R}^m$ a linear operator and $\hat{f} : Y \to \mathbb{R}$ is a simple convex function. Although f given by (1.6) is nonsmooth in general, Nesterov showed that it can be closely approximated by a smooth function. Accordingly, he devised a novel smoothing scheme that can achieve the $\mathcal{O}(1/\epsilon)$ for solving this class of saddle point problems. However, under the assumption that only linear subproblems given in the form of (1.2) (rather than (1.3)) are allowed, it is unclear whether the aforementioned complexity bounds for solving smooth, nonsmooth and saddle point problems are still achievable or not.

Our contribution in this paper lies on the following three aspects. Firstly, we show that to solve CP problems under an LO oracle is fundamentally more difficult than to solve CP problems without such restrictions, by establishing a

series of lower complexity bounds for solving different classes of CP problems under an LO oracle. More specifically, we show that for solving general smooth CP problems satisfying (1.5), the complexity (or number of calls to the LO oracle), in the worst case, cannot be smaller than

$$\mathcal{O}(1)\left\{n, \frac{LD_X^2}{\epsilon}\right\}, \tag{1.7}$$

where $\mathcal{O}(1)$ denotes an absolute constant, n is the dimension of the problem, and $D_X := \max_{x,y\in X} \|x-y\|$. Moreover, for solving the aforementioned saddle point problems with f given by (1.6), we show that the number of calls to the LO oracle cannot be smaller than

$$\mathcal{O}(1)\left\{n, \frac{\|A\|^2 D_X^2 D_Y^2}{\epsilon^2}\right\}. \tag{1.8}$$

We further show that the number of calls to the LO oracle for solving general nonsmooth CP problems cannot be smaller than

$$\mathcal{O}(1)\left\{n, \frac{M^2 D_X^2}{\epsilon^2}\right\}. \tag{1.9}$$

It should be pointed out that these lower complexity bounds are obtained not only for the aforementioned first-order LCP methods, but also for any other LCP methods including those based on higher-order information.

Secondly, we formally establish the (near) optimality of the CG method and its variants for solving different classes of CP problems under an LO oracle.

a) If f is a smooth convex function satisfying (1.5), it is well-known that the number of iterations required by the classic CG method to find an ϵ-solution of (1.1) will be bounded by $\mathcal{O}(1/\epsilon)$. Hence, in view of (1.7), the classic CG is an optimal LCP method if n is sufficiently large, i.e., $n \geq LD_X^2/\epsilon$. Moreover, it is also well-known that for general first-order methods, one can employ non-Euclidean norm $\|\cdot\|$ and the distance function $d(x)$ in (1.3) to accelerate the solutions for CP problems with certain types of feasible sets X. However, we demonstrate that the CG method is invariant to the selection of $\|\cdot\|$ and thus self-adaptive to the geometry of the feasible region X.
b) If f is a special nonsmooth function given by (1.6), we show that the CG method can achieve the lower complexity bound in (1.8) after properly smoothing the objective function. Note that, although a similar bound has been developed in [7], the optimality of this bound has not yet been established. In addition, the smoothing technique developed here is slightly different from those in [28, 7] as we do not require explicit knowledge of D_X, D_Y and the target accuracy ϵ given in advance.
c) If f is a general nonsmooth function satisfying (1.4), we show that the CG method can achieve a nearly optimal complexity bound in terms of its dependence on ϵ after properly incorporating the randomized smoothing technique (e.g., [11]). In particular, by applying this method to the bilinear saddle point problems with f given by (1.6), we obtain an first-order algorithm which only requires linear optimization in both primal and dual space to solve this class of problems. It appears to us that no such techniques have been presented before in the literature (see discussions in Section 1 of [29]).
d) We also discuss the possibility to improve the complexity of the CG method under strong convexity assumption about $f(\cdot)$ and with an enhanced LO oracle.

Thirdly, we present a few new LCP methods, namely the primal averaging CG (PA-CG) and primal-dual averaging CG (PDA-CG) algorithms, for solving large-scale CP problems under an LO oracle. These methods are obtained by replacing the projection subproblems with linear optimization subproblems in Nesterov's accelerated gradient methods. We demonstrate that these new LCP methods not only exhibit the aforementioned optimal (or nearly optimal) complexity bounds for solving different CP problems under an LO oracle, but also possess some unique convergence properties. In particular, we show that the rate of convergence of these new LCP methods depends on the summation of the distances among the solutions of (1.2). By exploiting this fact, we develop certain necessary conditions for the LO oracle under which the PA-CG and PDA-CG would exhibit an $\mathcal{O}(1/\sqrt{\epsilon})$ iteration complexity for solving smooth CP problems. This result thus helps to build up the connection between LCP methods and the general optimal first-order methods for CP. We also demonstrate through our preliminary numerical experiments that one of these new methods, namely PDA-CG, can significantly outperform the CG method for solving certain classes of CP problems, e.g., those with box-type constraints.

This paper is organized as follows. We first introduce a few lower complexity bounds for solving different classes of CP problems under an LO oracle in Section 2. In Section 3, we formally show the optimality of the classic CG

method for solving smooth CP problems, develop different variants of the CG method which are optimal or nearly optimal for solving different nonsmooth CP problems, and present possible improvement of the CG method to solve strongly convex CP problems. We then present a few new LCP methods, namely PA-CG and PDA-CG, establish their convergence properties and conduct numerical comparisons in Section 4. Some brief concluding remarks are made in Section 5.

1.1 Notation and terminology

Let $X \in \mathbb{R}^n$ and $Y \in \mathbb{R}^m$ be given convex compact sets. Also let $\|\cdot\|_X$ and $\|\cdot\|_Y$ be the norms (not necessarily associated with inner product) in $\mathbb{R}^n$ and $\mathbb{R}^m$, respectively. For the sake of simplicity, we often skip the subscripts in the norms $\|\cdot\|_X$ and $\|\cdot\|_Y$. We define the diameter of the sets X and Y, respectively, as

$$D_X \equiv D_{X,\|\cdot\|} := \max_{x,y\in X} \|x - y\| \tag{1.10}$$

and

$$D_Y \equiv D_{Y,\|\cdot\|} := \max_{x,y\in Y} \|x - y\|. \tag{1.11}$$

For a given norm $\|\cdot\|$, we denote its conjugate by $\|s\|_* = \max_{\|x\|\le 1}\langle s, x\rangle$. We use $\|\cdot\|_1$ and $\|\cdot\|_2$, respectively, to denote the regular l_1 and l_2 norms. Let $A : \mathbb{R}^n \to \mathbb{R}^m$ be a given linear operator, we use $\|A\|$ to denote its operator norm given by $\|A\| := \max_{\|x\|\le 1} \|Ax\|$. Let $f : X \to \mathbb{R}$ be a convex function, we denote its linear approximation at x by

$$l_f(x; y) := f(x) + \langle f'(x), y - x\rangle. \tag{1.12}$$

Clearly, if f satisfies (1.5), then

$$f(y) \le l_f(x; y) + \frac{L}{2}\|y - x\|^2, \quad \forall\, x, y \in X. \tag{1.13}$$

Notice that the constant L in (1.5) and (1.13) depends on $\|\cdot\|$.

2 Lower Complexity Bounds for CP under an LO oracle

Our goal in this section is to establish a few lower complexity bounds for solving different classes of CP problems under an LO oracle. More specifically, we first introduce a generic LCP algorithm in Subsection 2.1 and then present a few lower complexity bounds for these types of algorithms to solve different smooth and nonsmooth CP problems in Subsections 2.2 and 2.3, respectively.

2.1 A generic LCP algorithm

The LCP algorithms solve problem (1.1) iteratively. In particular, at the k-th iteration, these algorithms perform a call to the LO oracle in order to update the iterates by minimizing a given linear function $\langle p_k, x\rangle$ over the feasible region X. A generic framework for these types of algorithms is described as follows.

Algorithm 1 A generic LCP algorithm

Let $x_0 \in X$ be given.
for $k = 1, 2, \ldots,$ **do**
  Define the linear function $\langle p_k, \cdot\rangle$.
  Call the LO oracle to compute $x_k \in \mathrm{Argmin}_{x\in X}\langle p_k, x\rangle$.
  Output $y_k \in \mathrm{Conv}\{x_0, \ldots, x_k\}$.
end for

Observe the above LCP algorithm can be quite general. Firstly, there are no restrictions regarding the definition of the linear function $\langle p_k, \cdot\rangle$. For example, if f is a smooth function, then p_k can be defined as the gradient computed

at some feasible solution or a linear combination of some previously computed gradients. If f is nonsmooth, we can define p_k as the gradient computed for a certain approximation function of f. We can also consider the situation when some random noise or second-order information is incorporated into the definition of p_k. Secondly, the output solution y_k is written as a convex combination of $x_0, \ldots, x_k$, and thus can be different from any points in $\{x_k\}$. We will show in Sections 3 and 4 that Algorithm 1 covers, as certain special cases, the classic CG method and several new LCP methods to be studied in this paper.

It is interesting to observe the difference between the above LCP algorithm and the general first-order methods for CP. One one hand, the LCP algorithm can only solve linear, rather than nonlinear subproblems (e.g., projection or prox-mapping) to update iterates. On the other hand, the LCP algorithm allows more flexibility in the definitions of the search direction p_k and the output solution y_k.

2.2 Lower complexity bounds for smooth minimization

In this subsection, we consider a class of smooth CP problems, denoted by $\mathcal{F}^{1,1}_{L,\|\cdot\|}(X)$, which consist of any CP problems given in the form of (1.1) with f satisfying assumption (1.5). Our goal is to derive a lower bound on the number of iterations required by any LCP methods for solving this class of problems.

The complexity analysis has been an important topic in convex programming (see Nemirovski and Yudin [25], and Nesterov [27]). However, the study on the complexity for LCP methods is quite limited. Existing results focus on a specific algorithm, namely the classic CG method. More specifically, in 1968, Canon and Cullum [6] proved an asymptotic lower bound of $\Omega(1/k^{1+\mu})$, for any $\mu > 0$, on the rate of convergence for the CG method. Jaggi [18] revisited this algorithm and established a lower bound on the number of iteration performed by this algorithm for finding an approximate solution with certain sparse pattern.

Similarly to the classic complexity analysis for CP in [25,27], we assume that the LO oracle used in the LCP algorithm is *resisting*, implying that: i) the LCP algorithm does not know how the solution of (1.2) is computed; and ii) in the worst case, the LO oracle provides the least amount of information for the LCP algorithm to solve problem (1.1). Using this assumption, we will construct a class of worst-case instances in $\mathcal{F}^{1,1}_{L,\|\cdot\|}(X)$, inspired by [18], and establish a lower bound on the number of iterations required by any LCP algorithms to solve these instances.

Theorem 1 *Let $\epsilon > 0$ be a given target accuracy. The number of iterations required by any LCP methods to solve the problem class $\mathcal{F}^{1,1}_{L,\|\cdot\|}(X)$, in the worst case, cannot be smaller than*

$$\left\lceil \min\left\{ \frac{n}{2}, \frac{LD_X^2}{4\epsilon} \right\} \right\rceil - 1, \tag{2.1}$$

where D_X is given by (1.10).

Proof. Consider the CP problem of

$$f_0^* := \min_{x \in X_0} \left\{ f_0(x) := \frac{L}{2} \sum_{i=1}^{n} (x^{(i)})^2 \right\}, \tag{2.2}$$

where $X_0 := \left\{ x \in \mathbb{R}^n : \sum_{i=1}^n x^{(i)} = D, x^{(i)} \geq 0 \right\}$ for some $D > 0$. It can be easily seen that the optimal solution x^* and the optimal value f_0^* for problem (2.2) are given by

$$x^* = \left(\frac{D}{n}, \ldots, \frac{D}{n} \right) \quad \text{and} \quad f_0^* = \frac{LD^2}{n}. \tag{2.3}$$

Clearly, this class of problems belong to $\mathcal{F}^{1,1}_{L,\|\cdot\|}(X)$ with $\|\cdot\| = \|\cdot\|_2$.

Without loss of generality, we assume that the initial point is given by $x_0 = De_1$ where $e_1 = (1, 0, \ldots, 0)$ is the unit vector. Otherwise, for an arbitrary $x_0 \in X_0$, we can consider a similar problem given by

$$\begin{array}{ll} \min_x & \left(x^{(1)}\right)^2 + \sum_{i=2}^n \left(x^{(i)} - x_0^{(i)}\right)^2 \\ \text{s.t.} & x^{(1)} + \sum_{i=2}^n \left(x^{(i)} - x_0^{(i)}\right) = D \\ & x^{(1)} \geq 0 \\ & x^{(i)} - x_0^{(i)} \geq 0, i = 2, \ldots, n. \end{array}$$

and adapt our following argument to this problem without much modification.

Now suppose that problem (2.2) is to be solved by an LCP algorithm. At the k-th iteration, this algorithm will call the LO oracle to compute a new search point x_k based on the input vector p_k, $k = 1, \ldots$. We assume that the LO oracle is resisting in the sense that it always outputs an extreme point $x_k \in \{De_1, De_2, \ldots, De_n\}$ such that

$$x_k \in \mathrm{Argmin}_{x \in X_0} \langle p_k, x \rangle.$$

Here e_i, $i = 1, \ldots, n$, denotes the i-th unit vector in $\mathbb{R}^n$. In addition, whenever x_k is not uniquely defined, it breaks the tie arbitrarily. Let us denote $x_k = De_{p_k}$ for some $1 \le p_k \le n$. By definition, we have $y_k \in D\,\mathrm{Conv}\{x_0, x_1, \ldots, x_k\}$ and hence

$$y_k \in D\,\mathrm{Conv}\{e_1, e_{p_1}, e_{p_2}, \ldots, e_{p_k}\}. \tag{2.4}$$

Suppose that totally q unit vectors from the set $\{e_1, e_{p_1}, e_{p_2}, \ldots, e_{p_k}\}$ are linearly independent for some $1 \le q \le k+1 \le n$. Without loss of generality, assume that the vectors e_1, e_{p_1}, e_{p_2}, $\ldots$, $e_{p_{q-1}}$ are linearly independent. Therefore, we have

$$\begin{aligned} f_0(y_k) &\ge \min_x \{f_0(x) : x \in D\,\mathrm{Conv}\{e_1, e_{p_1}, e_{p_2}, \ldots, e_{p_k}\}\} \\ &= \min_x \left\{f_0(x) : x \in D\,\mathrm{Conv}\{e_1, e_{p_1}, e_{p_2}, \ldots, e_{p_{q-1}}\}\right\} \\ &= \frac{LD^2}{q} \ge \frac{LD^2}{k+1}, \end{aligned}$$

where the second identity follows from the definition of f_0 in (2.2). The above inequality together with (2.3) then imply that

$$f_0(y_k) - f_0^* \ge \frac{LD^2}{k+1} - \frac{LD^2}{n} \tag{2.5}$$

for any $k = 1, \ldots, n-1$. Let us denote

$$\bar{K} := \left\lceil \min\left\{\frac{n}{2}, \frac{LD_{X_0}^2}{4\epsilon}\right\}\right\rceil - 1.$$

By the definition of D_X and X_0, and the fact that $\|\cdot\| = \|\cdot\|_2$, we can easily see that $D_{X_0} = \sqrt{2}D$ and hence that

$$\bar{K} = \left\lceil \frac{1}{2}\min\left\{n, \frac{LD^2}{\epsilon}\right\}\right\rceil - 1.$$

Using (2.5) and the above identity, we conclude that, for any $1 \le k \le \bar{K}$,

$$\begin{aligned} f_0(y_k) - f_0^* &\ge \frac{LD^2}{\bar{K}+1} - \frac{LD^2}{n} \ge \frac{2LD^2}{\min\left\{n, \frac{LD^2}{\epsilon}\right\}} - \frac{LD^2}{n} \\ &= \frac{LD^2}{\min\left\{n, \frac{LD^2}{\epsilon}\right\}} + \left(\frac{LD^2}{\min\left\{n, \frac{LD^2}{\epsilon}\right\}} - \frac{LD^2}{n}\right) \ge \frac{LD^2}{\frac{LD^2}{\epsilon}} + \left(\frac{LD^2}{n} - \frac{LD^2}{n}\right) = \epsilon. \end{aligned}$$

Our result then immediately follows since (2.2) is a special class of problems in $\mathcal{F}_{L,\|\cdot\|}^{1,1}(X)$. ∎

We now add a few remarks about the results obtained in Theorem 1. First, it can be easily seen from (2.1) that, if $n \ge LD_X^2/(2\epsilon)$, then the number of iterations required by any LCP methods for solving $\mathcal{F}_{L,\|\cdot\|}^{1,1}(X)$, in the worst case, cannot be smaller than $\mathcal{O}(1)LD_X^2/\epsilon$. Second, it is worth noting that the objective function f_0 in (2.2) is actually strongly convex. Hence, the performance of the LCP methods cannot be improved by assuming strong convexity when n is sufficiently large (see Section 3.4 for more discussions). This is in sharp contrast to the general first-order methods whose complexity for solving strongly convex problems depends on $\log(1/\epsilon)$.

2.3 Lower complexity bounds for nonsmooth minimization

In this subsection, we consider two classes of nonsmooth CP problems. The first one is a general class of nonsmooth CP problems, denoted by $\mathcal{F}^0_{M,\|\cdot\|}(X)$, which consist of any CP problems given in the form of (1.1) with f satisfying (1.4). The second one is a special class of bilinear saddle-point problems, denoted by $\mathcal{F}^0_{\|A\|}(X,Y)$, composed of all CP problems (1.1) with f given by (1.6). Our goal in this subsection is to derive the lower complexity bounds for any LCP algorithms to solve these two classes of nonsmooth CP problems.

It can be seen that, if $f(\cdot)$ is given by (1.6), then

$$\|f'(x)\|_* \le \|A\| D_Y, \quad \forall x \in X,$$

where D_Y is given by (1.10). Hence, the saddle point problems $\mathcal{F}^0_{\|A\|}(X,Y)$ are a special class of nonsmooth CP problems.

Theorem 2 below provides a few lower complexity bounds for solving these two classes of nonsmooth CP problems by using LCP algorithms.

Theorem 2 *Let $\epsilon > 0$ be a given target accuracy. Then, the number of iterations required by any LCP methods to solve the problem classes $\mathcal{F}^0_{M,\|\cdot\|}(X)$ and $\mathcal{F}^0_{\|A\|}(X,Y)$, respectively, cannot be smaller than*

$$\frac{1}{4}\min\left\{n, \frac{M^2 D_X^2}{2\epsilon^2}\right\} - 1 \tag{2.6}$$

and

$$\frac{1}{4}\min\left\{n, \frac{\|A\|^2 D_X^2 D_Y^2}{2\epsilon^2}\right\} - 1, \tag{2.7}$$

where D_X and D_Y are defined in (1.10) and (1.11), respectively.

Proof. We first show the bound in (2.6). Consider the CP problem of

$$\hat{f}_0^* := \min_{x \in X_0}\left\{\hat{f}(x) := M\left(\sum_{i=1}^n x_i^2\right)^{\frac{1}{2}}\right\}, \tag{2.8}$$

where $X_0 := \left\{x \in \mathbb{R}^n : \sum_{i=1}^n x^{(i)} = D, x^{(i)} \ge 0\right\}$ for some $D > 0$. It can be easily seen that the optimal solution x^* and the optimal value f_0^* for problem (2.8) are given by

$$x^* = \left(\frac{D}{n}, \ldots, \frac{D}{n}\right) \quad \text{and} \quad \hat{f}_0^* = \frac{MD}{\sqrt{n}}. \tag{2.9}$$

Clearly, this class of problems belong to $\mathcal{F}^0_{M,\|\cdot\|}(X)$ with $\|\cdot\| = \|\cdot\|_2$. Now suppose that problem (2.2) is to be solved by an arbitrary LCP method. Without loss of generality, we assume that the initial point is given by $x_0 = De_1$ where $e_1 = (1, 0, \ldots, 0)$ is the unit vector. Assume that the LO oracle is resisting in the sense that it always outputs an extreme point solution. By using an argument similar to the one used in the proof of (2.4), we can show that

$$y_k \in D\,\mathrm{Conv}\{e_1, e_{p_1}, e_{p_2}, \ldots, e_{p_k}\}$$

where e_{p_i}, $i = 1, \ldots, k$, are the unit vectors in $\mathbb{R}^n$. Suppose that totally q unit vectors in the set $\{e_1, e_{p_1}, e_{p_2}, \ldots, e_{p_k}\}$ are linearly independent for some $1 \le q \le k+1 \le n$. We have

$$\hat{f}_0(y_k) \ge \min_x\left\{\hat{f}_0(x) : x \in D\,\mathrm{Conv}\{e_1, e_{p_1}, e_{p_2}, \ldots, e_{p_k}\}\right\} = \frac{MD}{\sqrt{q}} \ge \frac{MD}{\sqrt{k+1}},$$

where the identity follows from the definition of $\hat{f}_0$ in (2.8). The above inequality together with (2.9) then imply that

$$\hat{f}_0(y_k) - \hat{f}_0^* \ge \frac{MD}{\sqrt{k+1}} - \frac{LD^2}{\sqrt{n}} \tag{2.10}$$

for any $k = 1, \ldots, n-1$. Let us denote

$$\bar{K} := \frac{1}{4}\left\lceil \min\left\{n, \frac{M^2 D_{X_0}^2}{2\epsilon^2}\right\}\right\rceil - 1.$$

Using the above definition, (2.10) and the fact that $D_{X_0} = \sqrt{2}D$, we conclude that

$$\hat{f}_0(y_k) - \hat{f}_0^* \geq \frac{MD}{\sqrt{\bar{K}+1}} - \frac{MD}{n} \geq \frac{2MD}{\min\left\{\sqrt{n}, \frac{MD}{\epsilon}\right\}} - \frac{MD}{\sqrt{n}} \geq \epsilon$$

for any $1 \leq k \leq \bar{K}$. Our result in (2.6) then immediately follows since (2.8) is a special class of problems in $\mathcal{F}^0_{M,\|\cdot\|}(X)$.

In order to prove the lower complexity bound in (2.7), we consider a class of saddle point problems given in the form of

$$\min_{x \in X_0} \max_{\|y\|_2 \leq \tilde{D}} M\langle x, y\rangle. \tag{2.11}$$

Clearly, these problems belong to $\mathcal{S}_{\|A\|}(X, Y)$ with $A = MI$. Noting that problem (2.11) is equivalent to

$$\min_{x \in X_0} M\tilde{D}\left(\sum_{i=1}^n x_i^2\right)^{\frac{1}{2}},$$

we can show the lower complexity bound in (2.7) by using an argument similar to the one used in the proof of bound (2.6). ∎

Observe that while the lower complexity bound in (2.6) is in the same order of magnitude as the one established in [25,27] for the general first-order methods to solve $\mathcal{F}^0_{M,\|\cdot\|}(X)$. However, the bound in (2.6) holds not only for first-order LCP methods, but also for any other LCP methods, including those based on higher-order information to solve $\mathcal{F}^0_{M,\|\cdot\|}(X)$.

3 The Optimality of CG Methods for CP under an LO oracle

Our goal in this section is to establish the optimality or near optimality of the classic CG method and its variants for solving different classes of CP problems under an LO oracle. More specifically, we discuss the classic CG method for solving smooth CP problems $\mathcal{F}^{1,1}_{L,\|\cdot\|}(X)$ in Subsection 3.1, and then present different variants of the CG method to solve nonsmooth CP problems $\mathcal{F}^0_{\|A\|}(X, Y)$ and $\mathcal{F}^0_{M,\|\cdot\|}(X)$, respectively, in Subsections 3.2 and 3.3. Some discussions about strongly convex problems are included in Subsection 3.4.

3.1 Optimal CG methods for $\mathcal{F}^{1,1}_{L,\|\cdot\|}(X)$ under an LO oracle

The classic CG method [13,10] is one of the earliest iterative algorithms to solve problem (1.1). The basic scheme of this algorithm is stated as follows.

Algorithm 2 The Classic Conditional Gradient (CG) Method

Let $x_0 \in X$ be given. Set $y_0 = x_0$.
for $k = 1, \ldots$ **do**
  Call the LO oracle to compute $x_k \in \mathrm{Argmin}_{x \in X}\langle f'(y_{k-1}), x\rangle$.
  Set $y_k = (1 - \alpha_k)y_{k-1} + \alpha_k x_k$ for some $\alpha_k \in [0, 1]$.
end for

We now add a few remarks about the classic CG method. Firstly, it can be easily seen that the classic CG method is a special case of the LCP algorithm discussed in Subsection 2.1. More specifically, the search direction p_k appearing

in the generic LCP algorithm is simply set to the gradient $f'(y_{k-1})$ in Algorithm 3, and the output y_k is taken as a convex combination of y_{k-1} and x_k. Secondly, in order to guarantee the convergence of the classic CG method, we need to properly specify the stepsizes α_k used in the definition of y_k. There are two popular options for selecting α_k: one is to set

$$\alpha_k = \frac{2}{k+1}, \quad k = 1, 2, \ldots, \tag{3.1}$$

and the other is to compute α_k by solving a one-dimensional minimization problem:

$$\alpha_k = \text{argmin}_{\alpha \in [0,1]} f((1-\alpha)y_{k-1} + \alpha x_k), \quad k = 1, 2, \ldots. \tag{3.2}$$

It is well-known that if f satisfies (1.5) and α_k is set to either (3.1) or (3.2), then the classic CG method will exhibit an $\mathcal{O}(1/k)$ rate of convergence for solving problem (1.1).

We now formally describe the convergence properties of the above classic CG method. Observe that, in contrast with existing analysis of the classic CG method, we state explicitly in Theorem 3 how the rate of convergence associated with this algorithm depends on distance between the previous iterate y_{k-1} and the output of the LO oracle, i.e., $\|x_k - y_{k-1}\|$. In addition, our analysis for the classic CG method is slightly different than the standard ones, and some of the techniques developed here will be used later for the analysis of some new LCP methods in Section 4.

We first state a simple technical result.

Lemma 1 *Let $\gamma_k \in (0, 1]$, $k = 1, 2, \ldots$, be given. If the sequence $\{\Delta_k\}_{k \geq 0}$ satisfies*

$$\Delta_k \leq (1 - \gamma_k)\Delta_{k-1} + B_k, \quad k = 1, 2, \ldots, \tag{3.3}$$

then

$$\Delta_k \leq \Gamma_k(1 - \gamma_1)\Delta_0 + \Gamma_k \sum_{i=1}^{k} \frac{B_i}{\Gamma_i}, \tag{3.4}$$

where

$$\Gamma_k := \begin{cases} 1, & k = 1, \\ (1 - \gamma_k)\,\Gamma_{k-1}, & k \geq 2. \end{cases} \tag{3.5}$$

Proof. Dividing both sides of (3.3) by Γ_k, we obtain

$$\frac{\Delta_1}{\Gamma_1} \leq \frac{(1-\gamma_1)\Delta_0}{\Gamma_1} + \frac{B_1}{\Gamma_1}$$

and

$$\frac{\Delta_k}{\Gamma_k} \leq \frac{\Delta_{k-1}}{\Gamma_{k-1}} + \frac{B_k}{\Gamma_k}, \quad \forall k \geq 2.$$

Summing up these inequalities, we obtain (3.4). ∎

We are now ready to describe the main convergence properties of the CG method.

Theorem 3 *Let $\{x_k\}$ be the sequence generated by the classic CG method applied to problem (1.1) with the stepsize policy in (3.1) or (3.2). If $f(\cdot)$ satisfies (1.5), then for any $k = 1, 2, \ldots$,*

$$f(y_k) - f^* \leq \frac{2L}{k(k+1)} \sum_{i=1}^{k} \|x_i - y_{i-1}\|^2. \tag{3.6}$$

Proof. Let Γ_k be defined in (3.5) with

$$\gamma_k := \frac{2}{k+1}. \tag{3.7}$$

It is easy to check that

$$\Gamma_k = \frac{2}{k(k+1)} \quad \text{and} \quad \frac{\gamma_k^2}{\Gamma_k} \leq 2, \quad k = 1, 2, \ldots. \tag{3.8}$$

Denoting $\tilde{y}_k = (1-\gamma_k)y_{k-1}+\gamma_k x_k$, we conclude from from (3.1) (or (3.2)) and the definition of y_k in Algorithm 3 that $f(y_k) \le f(\tilde{y}_k)$. It also follows from the definition of $\tilde{y}_k$ that $\tilde{y}_k - y_{k-1} = \gamma_k(x_k - y_{k-1})$. Letting $l_f(x;y)$ be defined in (1.12) and using these two observations, (1.13), the definition of x_k and the convexity of $f(\cdot)$, we have

$$\begin{aligned} f(y_k) &\le f(\tilde{y}_k) \le l_f(y_{k-1};\tilde{y}_k) + \frac{L}{2}\|y_k - y_{k-1}\|^2 \\ &= (1-\gamma_k)f(y_{k-1}) + \gamma_k l_f(y_{k-1};x_k) + \frac{L}{2}\gamma_k^2\|x_k - y_{k-1}\|^2 \\ &\le (1-\gamma_k)f(y_{k-1}) + \gamma_k l_f(y_{k-1};x) + \frac{L}{2}\gamma_k^2\|x_k - y_{k-1}\|^2, \\ &\le (1-\gamma_k)f(y_{k-1}) + \gamma_k f(x) + \frac{L}{2}\gamma_k^2\|x_k - y_{k-1}\|^2, \ \forall x \in X. \end{aligned} \tag{3.9}$$

Subtracting $f(x)$ from both sides of the above inequality, we obtain

$$f(y_k) - f(x) \le (1-\gamma_k)[f(y_{k-1}) - f(x)] + \frac{L}{2}\gamma_k^2\|x_k - y_{k-1}\|^2, \tag{3.10}$$

which, in view of Lemma 1, then implies that

$$\begin{aligned} f(y_k) - f(x) &\le \Gamma_k(1-\gamma_1)[f(y_0) - f(x)] + \frac{\Gamma_k L}{2}\sum_{i=1}^{k}\frac{\gamma_i^2}{\Gamma_i}\|x_i - y_{i-1}\|^2 \\ &\le \frac{2L}{k(k+1)}\sum_{i=1}^{k}\|x_i - y_{i-1}\|^2, \quad k = 1,2,\ldots, \end{aligned} \tag{3.11}$$

where the last inequality follows from the fact that $\gamma_1 = 1$ and (3.8). ∎

We now add a few remarks about the results obtained in Theorem 3. Firstly, note that by (3.6) and the definition of D_X in (1.10), we have, for any $k = 1,\ldots,$

$$f(y_k) - f^* \le \frac{2L}{k+1}D_X^2.$$

Hence, the number of iterations required by the classic CG method to find an ϵ-solution of problem (1.1) is bounded by

$$\mathcal{O}(1)\frac{LD_X^2}{\epsilon}. \tag{3.12}$$

Comparing the above bound with (2.1), we conclude that the classic CG algorithm is an optimal LCP method for solving $\mathcal{F}^{1,1}_{L,\|\cdot\|}(X)$ if n is sufficiently large.

Secondly, although the CG method does not require the selection of the norm $\|\cdot\|$, the iteration complexity of this algorithm, as stated in (3.12), does depend on $\|\cdot\|$ as the two constants, i.e., $L \equiv L_{\|\cdot\|}$ and $D_X \equiv D_{X,\|\cdot\|}$, depend on $\|\cdot\|$. However, since the result in (3.12) holds for an arbitrary $\|\cdot\|$, the iteration complexity of the classic CG method to solve problem (1.1) can actually be bounded by

$$\mathcal{O}(1)\inf_{\|\cdot\|}\left\{\frac{L_{\|\cdot\|}D^2_{X,\|\cdot\|}}{\epsilon}\right\}. \tag{3.13}$$

For example, if X is a simplex, a widely-accepted strategy to accelerate gradient type methods is to set $\|\cdot\| = \|\cdot\|_1$ and $d(x) = \sum_{i=1}^n x_i \log x_i$ in (1.3), in order to obtain (nearly) dimension-independent complexity results (see [25,28, 21]). On the other hand, the classic CG method can automatically adjust to the geometry of the feasible set X in order to obtain such scalability to high-dimensional problems (see Lemma 7 in [18] for some related discussions).

Thirdly, observe that the rate of convergence in (3.6) depends on $\|x_k - y_{k-1}\|$ which usually does not vanish as k increases. For example, suppose $\{y_k\} \to x^*$ (this is true if x^* is a unique optimal solution of (1.1)), the distance $\{\|x_k - y_{k-1}\|\}$ does not necessarily converge to zero unless x^* is an extreme point of X. In these cases, the summation $\sum_{i=1}^k \|x_i - y_{i-1}\|^2$ increases linearly with respect k. We will discuss some techniques in Section 4 that might help to improve this situation.

3.2 Optimal CG methods for $\mathcal{F}^0_{\|A\|}(X,Y)$ under an LO oracle

In this subsection, we show that the CG method, after incorporating some proper modification, can achieve the optimal complexity for solving the saddle point problems $\mathcal{F}^0_{\|A\|}(X,Y)$ under an LO oracle.

Since the objective function f given by (1.6) is nonsmooth in general, we cannot directly apply the CG method to $\mathcal{F}^0_{\|A\|}(X,Y)$. However, as shown by Nesterov [28], the function $f(\cdot)$ in (1.6) can be closely approximated by a class of smooth convex functions. More specifically, for a given strongly convex function $v: Y \to \mathbb{R}$ such that

$$v(y) \geq v(x) + \langle v'(x), y - x\rangle + \frac{\sigma_v}{2}\|y-x\|^2, \forall x, y \in Y, \tag{3.14}$$

let us denote $c_v := \operatorname{argmin}_{y\in Y} v(y)$, $V(y) := v(y) - v(c_v) - \langle \nabla v(c_v), y - c_v\rangle$ and

$$\mathcal{D}^2_{Y,V} := \max_{y\in Y} V(y). \tag{3.15}$$

Then the function $f(\cdot)$ in (1.6) can be closely approximated by

$$f_\eta(x) := \max_y \left\{ \langle Ax, y\rangle - \hat{f}(y) - \eta\left[V(y) - \mathcal{D}^2_{Y,V}\right] :\ y \in Y \right\}. \tag{3.16}$$

Indeed, by definition we have $0 \leq V(y) \leq \mathcal{D}^2_{Y,V}$ and hence, for any $\eta \geq 0$,

$$f(x) \leq f_\eta(x) \leq f(x) + \eta\,\mathcal{D}^2_{Y,V}, \quad \forall x \in X. \tag{3.17}$$

Moreover, Nesterov [28] shows that $f_\eta(\cdot)$ is differentiable and its gradients are Lipschitz continuous with the Lipschitz constant given by

$$\mathcal{L}_\eta := \frac{\|A\|^2}{\eta\sigma_v}. \tag{3.18}$$

In view of this result, we modify the CG method to solve $\mathcal{F}^0_{\|A\|}(X,Y)$ by replacing the gradient $f'(y_k)$ in Algorithm 3 with the gradient $f'_{\eta_k}(y_k)$ for some $\eta_k > 0$. Observe that in the original Nesterov smoothing scheme [28], we first need to define the smooth approximation function f_η in (3.16) by specifying in advance the smoothing parameter η and then apply a smooth optimization method to solve the approximation problem. The specification of η usually requires explicit knowledge of D_X, $\mathcal{D}^2_{Y,V}$ and the target accuracy ϵ given a priori. However, by using a novel analysis, we show that one can use variable smoothing parameters η_k and thus does not need to know the target accuracy ϵ in advance. In addition, wrong estimation on D_X and $\mathcal{D}^2_{Y,V}$ only affects the rate of convergence of the modified CG method by a constant factor. Our analysis relies on a slightly different construction of $f_\eta(\cdot)$ in (3.16) (i.e., the constant term $\eta\mathcal{D}^2_{Y,V}$ in (3.16) does not appear in [28]) and the following simple observation.

Lemma 2 *Let $f_\eta(\cdot)$ be defined in (3.16) and $\eta_1 \geq \eta_2 \geq 0$ be given. Then, we have $f_{\eta_1}(x) \geq f_{\eta_2}(x)$ for any $x \in X$.*

Proof. The result directly follows from the definition of $f_\eta(\cdot)$ in (3.16) and the fact that $V(y) - \mathcal{D}^2_{Y,V} \leq 0$. ∎

We are now ready to describe the main convergence properties of this modified CG method to solve $\mathcal{F}^0_{\|A\|}(X,Y)$.

Theorem 4 *Let $\{x_k\}$ and $\{y_k\}$ be the two sequences generated by the CG method with $f'(y_k)$ replaced by $f_{\eta_k}(y_k)$, where $f_\eta(\cdot)$ is defined in (1.6). If the stepsizes α_k, $k = 1, 2, \ldots$, are set to (3.1) or (3.2), and $\{\eta_k\}$ satisfies*

$$\eta_1 \geq \eta_2 \geq \ldots, \tag{3.19}$$

then we have, for any $k = 1, 2, \ldots$,

$$f(y_k) - f^* \leq \frac{2}{k(k+1)}\left[\sum_{i=1}^k \left(i\eta_i \mathcal{D}^2_{Y,V} + \frac{\|A\|^2}{\sigma_v \eta_i}\|x_i - y_{i-1}\|^2\right)\right]. \tag{3.20}$$

In particular, if

$$\eta_k = \frac{\|A\| D_X}{\mathcal{D}_{Y,V}\sqrt{\sigma_v k}}, \tag{3.21}$$

then we have, for any $k = 1, 2, \ldots$,

$$f(y_k) - f^* \leq \frac{2\sqrt{2}\|A\| D_X \, \mathcal{D}_{Y,V}}{\sqrt{\sigma_v k}}, \tag{3.22}$$

where D_X *and* $\mathcal{D}_{Y,V}$ *are defined in (1.10) and (3.15), respectively.*

Proof. Let Γ_k and γ_k be defined in (3.5) and (3.7), respectively. Similarly to (3.10), we have, for any $x \in X$,

$$\begin{aligned} f_{\eta_k}(y_k) &\leq (1-\gamma_k)[f_{\eta_k}(y_{k-1})] + \gamma_k f_{\eta_k}(x) + \frac{L_{\eta_k}}{2}\gamma_k^2 \|x_k - y_{k-1}\|^2 \\ &\leq (1-\gamma_k)[f_{\eta_{k-1}}(y_{k-1})] + \gamma_k f_{\eta_k}(x) + \frac{L_{\eta_k}}{2}\gamma_k^2 \|x_k - y_{k-1}\|^2 \\ &\leq (1-\gamma_k)[f_{\eta_{k-1}}(y_{k-1})] + \gamma_k [f(x) + \eta_k \mathcal{D}_{Y,V}^2] + \frac{L_{\eta_k}}{2}\gamma_k^2 \|x_k - y_{k-1}\|^2, \end{aligned}$$

where the second inequality follows from (3.19) and Lemma 2, and the third inequality follows from (3.17). Now subtracting $f(x)$ from both sides of the above inequality, we obtain, $\forall x \in X$,

$$\begin{aligned} f_{\eta_k}(y_k) - f(x) &\leq (1-\gamma_k)[f_{\eta_{k-1}}(y_{k-1}) - f(x)] + \gamma_k \eta_k \mathcal{D}_{Y,V}^2 + \frac{L_{\eta_k}}{2}\gamma_k^2 \|x_k - y_{k-1}\|^2 \\ &\leq (1-\gamma_k)[f_{\eta_{k-1}}(y_{k-1}) - f(x)] + \gamma_k \eta_k \mathcal{D}_{Y,V}^2 + \frac{\|A\|^2 \gamma_k^2}{2\sigma_v \eta_k} \|x_k - y_{k-1}\|^2, \end{aligned}$$

which, in view of Lemma 1, (3.7) and (3.8), then implies that, $\forall x \in X$,

$$f_{\eta_k}(y_k) - f(x) \leq \frac{2}{k(k+1)} \left[\sum_{i=1}^{k} \left(i\eta_i \mathcal{D}_{Y,V}^2 + \frac{\|A\|^2}{\sigma_v \eta_i} \|x_i - y_{i-1}\|^2 \right) \right], \quad \forall k \geq 1.$$

Our result in (3.20) then immediately follows from (3.17) and the above inequality. Now it is easy to see that the selection of η_k in (3.21) satisfies (3.19). By (3.20) and (3.21), we have

$$\begin{aligned} f(y_k) - f^* &\leq \frac{2}{k(k+1)} \left[\sum_{i=1}^{k} \left(i\eta_i \mathcal{D}_{Y,V}^2 + \frac{\|A\|^2}{\sigma_v \eta_i} D_X^2 \right) \right] \\ &= \frac{4\|A\| D_X \mathcal{D}_{v,Y}}{k(k+1)\sqrt{\sigma_v}} \sum_{i=1}^{k} \sqrt{i} \leq \frac{8\sqrt{2}\|A\| D_X \mathcal{D}_{Y,V}}{3\sqrt{\sigma_v k}}, \end{aligned}$$

where the last inequality follows from the fact that

$$\sum_{i=1}^{k} \sqrt{i} \leq \int_0^{k+1} t dt \leq \frac{2}{3}(k+1)^{\frac{3}{2}} \leq \frac{2\sqrt{2}}{3}(k+1)\sqrt{k}. \tag{3.23}$$

■

A few remarks about the results obtained in Theorem 4 are in order. First, observe that the specification of η_k in (3.21) requires the estimation of a few problem parameters, including $\|A\|$, D_X, $\mathcal{D}_{Y,V}$ and σ_v. However, wrong estimation on these parameters will only result in the increase on the rate of convergence of the modified CG method by a constant factor. For example, if $\eta_k = 1/\sqrt{k}$ for any $k \geq 1$, then (3.20) reduces to

$$f(y_k) - f^* \leq \frac{8\sqrt{2}}{3\sqrt{k}} \left(\mathcal{D}_{Y,V}^2 + \frac{\|A\|^2 D_X^2}{\sigma_v} \right).$$

It is worth noting that similar adaptive smoothing schemes can also be used when one applies Nesterov's accelerated gradient method to solve $\mathcal{F}^0_{\|A\|}(X, Y)$. Second, suppose that the norm $\|\cdot\|$ in the dual space associated with Y is an inner product norm and $v(y) = \|y\|^2/2$. In this case, by the definitions of D_Y and $\mathcal{D}_{Y,V}$ in (1.11) and (3.15), we have $\mathcal{D}_{Y,V} \leq D_Y$. Using this observation and (3.22), we conclude that the number of iterations required by the modified CG method to solve $\mathcal{F}^0_{\|A\|}(X, Y)$ can be bounded by

$$\mathcal{O}(1) \left(\frac{\|A\| D_X D_Y}{\epsilon} \right)^2,$$

which, in view of Theorem 2, is optimal when n is sufficiently large.

3.3 Nearly optimal CG methods for $\mathcal{F}^0_{M,\|\cdot\|}(X)$ under an LO oracle

In this subsection, we present a randomized CG method and demonstrate that it can achieve a nearly optimal rate of convergence for solving general nonsmooth CP problems $\mathcal{F}^0_{M,\|\cdot\|}(X)$ under an LO oracle. To the best of our knowledge, no such CG methods have not been presented before for solving general nonsmooth CP problems in the literature.

The basic idea is to approximate the general nonsmooth CP problems $\mathcal{F}^0_{M,\|\cdot\|}(X)$ by using the convolution-based smoothing. The intuition underlying such a approach is that convolving two functions yields a new function that is at least as smooth as the smoother one of the original two functions. In particular, let μ denote the density of a random variable with respect to Lebesgue measure and consider the function f_μ given by

$$f_\mu(x) := (f * \mu)(x) = \int_{\mathbb{R}^n} f(y)\mu(x-y)d(y) = \mathbb{E}_\mu[x+Z],$$

where Z is a random variable with density μ. Since μ is a density with respect to Lebesgue measure, f_μ is differentiable [5]. The above convolution-based smoothing technique has been extensively studied in stochastic optimization, e.g., [5,11,20,30,31]. For the sake of simplicity, we assume throughout this subsection that $\|\cdot\| = \|\cdot\|_2$ and Z is uniformly distributed over a certain Euclidean ball. The following result is known in the literature (see, e.g., [11]).

Lemma 3 *Let ξ be uniformly distributed over the l_2-ball $\mathcal{B}_2(0,1) := \{x \in \mathbb{R}^n : \|x\|_2 \le 1\}$ and $u > 0$ is given. Suppose that (1.4) holds for any $x, y \in X + u\,\mathcal{B}_2(0,1)$. Then, the following statements hold for the function $f_u(\cdot)$ given by*

$$f_u(x) := \mathbb{E}_\mu[f(x+u\xi)]. \tag{3.24}$$

a) $f(x) \le f_u(x) \le f(x) + Mu$;
b) $f_u(x)$ has $M\sqrt{n}/u$-Lipschitz continuous gradient with respect to $\|\cdot\|_2$;
c) $\mathbb{E}[f'(x+u\xi)] = f'_u(x)$ and $\mathbb{E}[\|f'(x+u\xi) - f'_u(x)\|^2] \le M^2$;
d) If $u_1 \ge u_2 \ge 0$, then $f_{u_1}(x) \ge f_{u_2}(x)$ for any $x \in X$.

In view of the above result, we can apply the CG method directly to $\min_{x\in X} f_u(x)$ for a properly chosen μ in order to solve the original problem (1.1). The only problem is that we cannot compute the gradient of $f_u(\cdot)$ exactly. To address this issue, we will generate an i.i.d. random sample $(\xi_1, \ldots, \xi_T)$ for some $T > 0$ and approximate the gradient $f'_\mu(x)$ by $\tilde{f}'_u(x) := \frac{1}{T}\sum_{t=1}^T f'(x, u\xi_t)$. After incorporating the aforementioned randomized smoothing scheme, the CG method applied to $\mathcal{F}^0_{M,\|\cdot\|}(X)$ exhibits the following convergence properties.

Theorem 5 *Let $\{x_k\}$ and $\{y_k\}$ be the two sequences generated by the classic CG method with $f'(y_{k-1})$ replaced by*

$$\tilde{f}'_{u_k}(y_{k-1}) := \frac{1}{T_k}\sum_{t=1}^{T_k} f'(y_{k-1}, u_k\xi_t). \tag{3.25}$$

where f_u is defined in (3.24) and $\{\xi_1, \ldots, \xi_{T_k}\}$ is an i.i.d. sample of ξ. If the stepsizes α_k, $k = 1, 2, \ldots$, are set to (3.1) or (3.2), and $\{u_k\}$ satisfies

$$u_1 \ge u_2 \ge \ldots, \tag{3.26}$$

then we have

$$\mathbb{E}[f(y_k)] - f(x) \le \frac{2M}{k(k+1)}\left[\sum_{i=1}^k \left(\frac{i}{\sqrt{T_i}}D_X + iu_i + \frac{\sqrt{n}}{u_i}D_X^2\right)\right], \tag{3.27}$$

where M is given by (1.4). In particular, if

$$T_k = k \quad \text{and} \quad u_k = \frac{n^{\frac{1}{4}}D_X}{\sqrt{k}}, \tag{3.28}$$

then

$$\mathbb{E}[f(y_k)] - f(x) \le \frac{4(1+2n^{\frac{1}{4}})MD_X}{3\sqrt{k}}, k = 1, 2, \ldots. \tag{3.29}$$

Proof. Let γ_k be defined in (3.7), similarly to (3.9), we have

$$\begin{aligned} f_{u_k}(y_k) &\le (1-\gamma_k) f_{u_k}(y_{k-1}) + \gamma_k l_{f_{u_k}}(x_k; y_{k-1}) + \frac{M\sqrt{n}}{2u_k}\gamma_k^2 \|x_k - y_{k-1}\|^2 \\ &\le (1-\gamma_k) f_{u_{k-1}}(y_{k-1}) + \gamma_k l_{f_{u_k}}(x_k; y_{k-1}) + \frac{M\sqrt{n}}{2u_k}\gamma_k^2 \|x_k - y_{k-1}\|^2, \end{aligned} \tag{3.30}$$

where the last inequality follows from the fact that $f_{u_{k-1}}(y_{k-1}) \ge f_{u_k}(y_{k-1})$ due to Lemma 3.d). Let us denote $\delta_k := f'_{u_k}(y_{k-1}) - \tilde{f}'_{u_k}(y_{k-1})$. Noting that by definition of x_k and the convexity of $f_{u_k}(\cdot)$,

$$\begin{aligned} l_{f_{u_k}}(x_k; y_{k-1}) &= f_{u_k}(y_{k-1}) + \langle f'_{u_k}(y_{k-1}, x_k - y_{k-1}\rangle \\ &= f_{u_k}(y_{k-1}) + \langle \tilde{f}'_{u_k}(y_{k-1}), x_k - y_{k-1}\rangle + \langle \delta_k, x_k - y_{k-1}\rangle \\ &\le f_{u_k}(y_{k-1}) + \langle \tilde{f}'_{u_k}(y_{k-1}), x - y_{k-1}\rangle + \langle \delta_k, x_k - y_{k-1}\rangle \\ &= f_{u_k}(y_{k-1}) + \langle f'_{u_k}(y_{k-1}), x - y_{k-1}\rangle + \langle \delta_k, x_k - x\rangle \\ &\le f_{u_k}(x) + \|\delta_k\| D_X \le f(x) + \|\delta_k\| D_X + M u_k, \quad \forall x \in X, \end{aligned}$$

where the last inequality follows from Lemma 3.a), we conclude from (3.30) that, $\forall x \in X$,

$$f_{u_k}(y_k) \le (1-\gamma_k) f_{u_{k-1}}(y_{k-1}) + \gamma_k \left[f(x) + \|\delta_k\| D_X + M u_k\right] + \frac{M\sqrt{n}}{2u_k}\gamma_k^2 \|x_k - y_{k-1}\|^2,$$

which implies that

$$f_{u_k}(y_k) - f(x) \le (1-\gamma_k)[f_{u_{k-1}}(y_{k-1}) - f(x)] + \gamma_k \left[\|\delta_k\| D_X + M u_k\right] + \frac{M\sqrt{n}}{2u_k}\gamma_k^2 D_X^2,$$

Noting that by Jensen's inequality and Lemma 3.c),

$$\{\mathbb{E}[\|\delta_k\|]\}^2 \le \mathbb{E}[\|\delta_k\|^2] = \frac{1}{T_k^2}\sum_{t=1}^{T_k} \mathbb{E}[\|f'(y_{k-1} + u_k\xi_k) - f'_{u_k}(y_{k-1})\|^2] \le \frac{M^2}{T_k}, \tag{3.31}$$

we conclude from the previous inequality that

$$\mathbb{E}[f_{u_k}(y_k) - f(x)] \le (1-\gamma_k)\mathbb{E}[f_{u_{k-1}}(y_{k-1}) - f(x)] + \frac{\gamma_k}{\sqrt{T_k}} M D_X + M\gamma_k u_k + \frac{M\sqrt{n}}{2u_k}\gamma_k^2 D_X^2,$$

which, in view of Lemma 1, (3.7) and (3.8), then implies that, $\forall x \in X$,

$$\mathbb{E}[f_{u_k}(y_k) - f(x)] \le \frac{2}{k(k+1)}\left[\sum_{i=1}^{k}\left(\frac{i}{\sqrt{T_i}} M D_X + M i u_i + \frac{M\sqrt{n}}{u_i} D_X^2\right)\right]$$

The result in (3.27) follows directly from Lemma 3.a) and the above inequality. Using (3.23), (3.27) and (3.28), we can easily verify that the bound in (3.29) holds. ∎

We now add a few remarks about the results obtained in Theorem 5. Firstly, note that in order to obtain the result in (3.29), we need to set $T_k = k$. This implies that at the k-th iteration of the randomized CG method in Theorem 5, we need to take an i.i.d. sample $\{\xi_1, \ldots, \xi_k\}$ of ξ and compute the corresponding gradients $\{f'(y_{k-1}, \xi_1), \ldots, f'(y_{k-1}, \xi_k)\}$. Also note that from the proof of the above result, we can recycle the generated samples $\{\xi_1, \ldots, \xi_k\}$ for usage in subsequent iterations.

Secondly, since $\mathcal{F}^0_{\|A\|}(X, Y) \subset \mathcal{F}^0_{M,\|\cdot\|}(X)$, we can apply the randomized CG method to solve the saddle point problems $\mathcal{F}^0_{\|A\|}(X, Y)$. In comparison with the smoothing CG method in Subsection 3.2, we do not need to solve the subproblems given in the form of (3.16), but to solve the subproblems

$$\max_y \left\{\langle A(x + \xi_i), y\rangle - \hat{f}(y) : y \in Y\right\},$$

in order to compute $f'(y_{k-1}, \xi_i)$, $i = 1, \ldots, k$, at the k-th iteration. In particular, if $\hat{f}(y) = 0$, then we only need to solve linear optimization subproblems over the set Y. To the best of our knowledge, this is the first time that optimization algorithms of this type has been proposed in the literature (see discussions in Section 1 of [29]).

Thirdly, in view of (3.29), the number of iterations (calls to the LO oracle) required by the randomized CG method to find a solution $\bar{x}$ such that $\mathbb{E}[f(\bar{x}) - f^*] \leq \epsilon$ can be bounded by

$$\mathcal{O}(1)\frac{\sqrt{n}M^2D_X^2}{\epsilon^2}.$$

According to the lower complexity bound in (2.6), we conclude that the above complexity bound is nearly optimal for the following reasons: i) the above result is in the the same order of magnitude as (2.6) with an additional factor of $\sqrt{n}$; and ii) the termination criterion is in terms of expectation. Note that while it is possible to show that the relation (3.29) holds with overwhelming probability by developing certain large deviation results associated with (3.29), such a result has been skipped in this paper for the sake of simplicity, see, e.g., [16] for some similar developments.

3.4 CG methods for strongly convex problems under an enhanced LO oracle

In this subsection, we assume that the objective function $f(\cdot)$ in (1.1) is smooth and strongly convex, i.e., in addition to (1.5), it also satisfies

$$f(y) - f(x) - \langle f'(x), y - x\rangle \geq \frac{\mu}{2}\|y - x\|^2, \ \forall x, y \in X. \tag{3.32}$$

These problems have been extensively studied in the literature. For example, it has been shown in [26,27] that the optimal complexity for the general first-order methods to solve this class of problems is given by by

$$\mathcal{O}(1)\sqrt{\frac{L}{\mu}}\max\left(\log\frac{\mu D_X}{\epsilon}, 1\right).$$

On the other hand, as noted in Subsection 2.2, the number of calls to the LO oracle for the LCP methods to solve these problems cannot be smaller than $\mathcal{O}(LD_X^2/\epsilon)$.

Our goal in this subsection is to show that, under certain stronger assumptions on the LO oracle, we can somehow "improve" the complexity of the CG method for solving these strongly convex problems. More specifically, we assume throughout this subsection that we have access to an enhanced LO oracle, which can solve optimization problems given in the form of

$$\min\{\langle p, x\rangle : x \in X, \|x\| \leq R\}. \tag{3.33}$$

For example, we can assume that the norm $\|\cdot\|$ is chosen such that problem (3.33) is relatively easy to solve. In particular, if X is a polytope, we can set $\|\cdot\| = \|\cdot\|_\infty$ or $\|\cdot\| = \|\cdot\|_1$ and then the complexity to solve (3.33) will be comparable to the one to solve (1.2). Note however, that such a selection of $\|\cdot\|$ will possibly increase the value of the condition number given by L/μ. Motivated by [15], we present a shrinking CG method under the above assumption on the enhanced LO oracle[1].

Algorithm 3 The Shrinking Conditional Gradient (CG) Method

Let $p_0 \in X$ be given. Set $R_0 = D_X$.
for $t = 1, \ldots$ **do**
  Set $y_0 = p_{t-1}$.
  for $k = 1, \ldots, 8L/\mu$ **do**
    Call the enhanced LO oracle to compute $x_k \in \mathrm{Argmin}_{x \in X_{t-1}}\langle f'(y_{k-1}), x\rangle$,
    where $X_{t-1} := \{x \in X : \|x - p_{t-1}\| \leq R_{t-1}\}$.
    Set $y_k = (1 - \alpha_k)y_{k-1} + \alpha_k x_k$ for some $\alpha_k \in [0, 1]$.
  end for
  Set $p_t = y_k$ and $R_t = R_{t-1}/\sqrt{2}$;
end for

[1] We recently notice that Garber and Hanzan [14] have made some interesting development for CG methods applied to strongly convex problems It should be noted, however, that the algorithm and analysis given here seem to be different than those in [14]

Note that an outer (resp., inner) iteration of the above shrinking CG method occurs whenever t (resp., k) increases by 1. Observe also that the feasible set X_t will be reduced at every outer iteration t. The following result summarizes the convergence properties for this algorithm.

Theorem 6 *Suppose that conditions (1.5) and (3.32) hold. If the stepsizes $\{\alpha_k\}$ in the shrinking CG method are set to (3.1) or (3.2), then the number of calls to the enhanced* LO *oracle performed by this algorithm to find an ϵ-solution of problem (1.1) can be bounded by*

$$\frac{8L}{\mu}\left\lceil \max\left(\log\frac{\mu R_0}{\epsilon}, 1\right)\right\rceil. \tag{3.34}$$

Proof. Denote $K \equiv 8L/\mu$. We first claim that $x^* \in X_t$ for any $t \geq 0$. This relation is obviously true for $t = 0$ since $\|y_0 - x^*\| \leq R_0 = D_X$. Now suppose that $x^* \in X_{t-1}$ for some $t \geq 1$. Under this assumption, relation (3.11) holds with $x = x^*$ for inner iterations $k = 1, \ldots, K$ performed at the t-th outer iteration. Hence, we have

$$f(y_k) - f(x^*) \leq \frac{2L}{k(k+1)}\sum_{i=1}^{k}\|x_i - y_{i-1}\|^2 \leq \frac{2L}{k+1}R_{t-1}^2, \quad k = 1, \ldots, K. \tag{3.35}$$

Letting $k = K$ in the above relation, and using the facts that $p_t = y_K$ and $f(y_K) - f^* \geq \mu\|y_K - x^*\|^2/2$, we conclude that

$$\|p_t - x^*\|^2 \leq \frac{2}{\mu}[f(p_t) - f^*] = \frac{2}{\mu}[f(y_K) - f^*] \leq \frac{4L}{\mu(K+1)}R_{t-1}^2 \leq \frac{1}{2}R_{t-1}^2 = R_t^2, \tag{3.36}$$

which implies that $x^* \in X_t$. We now provide a bound on the total number of calls to the LO oracle (i.e., the total number of inner iterations) performed by the shrinking CG method. It follows from (3.36) and the definition of R_t that

$$f(p_t) - f^* \leq \frac{\mu}{2}R_t^2 = \frac{\mu}{2}\frac{R_0}{2^{t-1}}, \quad t = 1, 2, \ldots.$$

Hence the total number of outer iterations performed by the shrinking CG method for finding an ϵ-solution of (1.1) is bounded by $\lceil \max(\log \mu R_0/\epsilon, 1)\rceil$. This observation, in view of the fact that K inner iterations are performed at each outer iteration t, then implies that the total number of inner iterations is bounded by (3.34). ∎

4 Acceleration schemes for LCP methods

Our goal in this section is to present a few new LCP methods for CP, obtained by replacing the projection (prox-mapping) subproblems with linear optimization subproblems in Nesterov's accelerated gradient method. Throughout this section, we focus on smooth CP problems $\mathcal{F}^{1,1}_{L,\|\cdot\|}(X)$. However, the developed algorithms can be easily modified to solve saddle point problems, general nonsmooth CP problems and strongly convex problems, by using similar ideas to those described in Section 3.

4.1 Primal averaging CG method

In this subsection, we present a new LCP method, obtained by incorporating a primal averaging step into the CG method. This algorithm is formally described as follows.

Algorithm 4 The Primal Averaging Conditional Gradient (PA-CG) Method

Let $x_0 \in X$ be given. Set $y_0 = x_0$.
for $k = 1, \ldots$ **do**
  Set $z_{k-1} = \frac{k-1}{k+1}y_{k-1} + \frac{2}{k+1}x_{k-1}$ and $p_k = f'(z_{k-1})$.
  Call the LO oracle to compute $x_k \in \mathrm{Argmin}_{x\in X}\langle p_k, x\rangle$.
  Set $y_k = (1-\alpha_k)y_{k-1} + \alpha_k x_k$ for some $\alpha_k \in [0, 1]$.
end for

It can be easily see that the PA-CG method stated above is a special case of the LCP method in Algorithm 1. It differs from the classic CG method in the way that the search direction p_k is defined. In particular, while p_k is set to $f'(x_{k-1})$ in the classic CG algorithm, the search direction p_k in PA-CG is given by $f'(z_{k-1})$ for some $z_{k-1} \in \mathrm{Conv}\{x_0, x_1, \ldots, x_{k-1}\}$. In other words, we will need to "average" the primal sequence $\{x_k\}$ before calling the LO oracle to update the iterates. It is worth noting that the PA-CG method can be viewed as a variant of Nesterov's method in [27,21], obtained by replacing the projection (or prox-mapping) subproblem with a simpler linear optimization subproblem.

By properly choosing the stepsize parameter α_k, we have the following convergence results for the PA-CG method described above.

Theorem 7 *Let $\{x_k\}$ and $\{y_k\}$ be the sequences generated by the PA-CG method applied to problem (1.1) with the stepsize policy in (3.1) or (3.2). Then we have*

$$f(y_k) - f^* \le \frac{2L}{k(k+1)} \sum_{i=1}^{k} \|x_i - x_{i-1}\|^2, \quad k = 1, 2, \ldots, \tag{4.1}$$

where L is given by (1.13).

Proof. Let γ_k and Γ_k be defined in (3.5) and (3.7), respectively. Denote $\tilde{y}_k = (1-\gamma_k)y_{k-1} + \gamma_k x_k$. It can be easily seen from (3.1) (or (3.2)) and the definition of y_k in Algorithm 4 that $f(y_k) \le f(\tilde{y}_k)$. Also by definition, we have $z_{k-1} = (1-\gamma_k)y_{k-1} + \gamma_k x_{k-1}$ and hence

$$\tilde{y}_k - z_{k-1} = \gamma_k(x_k - x_{k-1}).$$

Letting $l_f(\cdot,\cdot)$ be defined in (1.12), and using the previous two observations, (1.13), the definition of x_k in Algorithm 4, and the convexity of $f(\cdot)$, we obtain

$$\begin{aligned} f(y_k) \le f(\tilde{y}_k) &\le l_f(z_{k-1}; \tilde{y}_k) + \frac{L}{2}\|\tilde{y}_k - z_{k-1}\|^2 \\ &= (1-\gamma_k) l_f(z_{k-1}; y_{k-1}) + \gamma_k l_f(z_{k-1}; x_k) + \frac{L}{2}\gamma_k^2 \|x_k - x_{k-1}\|^2 \\ &\le (1-\gamma_k) f(y_{k-1}) + \gamma_k l_f(z_{k-1}; x) + \frac{L}{2}\gamma_k^2 \|x_k - x_{k-1}\|^2 \\ &\le (1-\gamma_k) f(y_{k-1}) + \gamma_k f(x) + \frac{L}{2}\gamma_k^2 \|x_k - x_{k-1}\|^2. \end{aligned} \tag{4.2}$$

Subtracting $f(x)$ from both sides of the above inequality, we have

$$f(y_k) - f(x) \le (1-\gamma_k)[f(y_{k-1}) - f(x)] + \frac{L}{2}\gamma_k^2 \|x_k - x_{k-1}\|^2,$$

which, in view of Lemma 1, (3.8) and the fact that $\gamma_1 = 1$, then implies that, $\forall x \in X$,

$$\begin{aligned} f(y_k) - f(x) &\le \Gamma_k(1-\gamma_1)[f(y_0) - f(x)] + \frac{\Gamma_k L}{2} \sum_{i=1}^{k} \frac{\gamma_i^2}{\Gamma_i} \|x_i - x_{i-1}\|^2 \\ &\le \frac{2L}{k(k+1)} \sum_{i=1}^{k} \|x_i - x_{i-1}\|^2, \quad k = 1, 2, \ldots. \end{aligned}$$

■

We now add a few remarks about the results obtained in Theorem 7. Firstly, similarly to (3.12), we can easily see that the number of iterations required by the PA-CG method to find an ϵ-solution of problem (1.1) is bounded by $\mathcal{O}(1)LD_X^2/\epsilon$. Therefore, the PA-CG method is an optimal LCP method for solving $\mathcal{F}^{1,1}_{L,\|\cdot\|}(X)$ when n is sufficiently large. In addition, since the selection of $\|\cdot\|$ is arbitrary, the iteration complexity of this method can also be bounded by (3.13).

Secondly, while the rate of convergence for the CG method (cf. (3.6)) depends on $\|x_k - y_{k-1}\|$, the one for the PA-CG method depends on $\|x_k - x_{k-1}\|$, i.e., the distance between the output of the LO oracle in two consecutive iterations. Clearly, the distance $\|x_k - x_{k-1}\|$ will depend on the geometry of X and the difference between p_k and p_{k-1}. Let γ_k be defined in (3.7) and suppose that α_k is set to (3.1) (i.e., $\alpha_k = \gamma_k$). Observe that by definitions of z_k and y_k in Algorithm 4, we have

$$\begin{aligned} z_k - z_{k-1} &= (y_k - y_{k-1}) + \gamma_{k+1}(x_k - y_k) - \gamma_k(x_{k-1} - y_{k-1}) \\ &= \alpha_k(x_k - y_{k-1}) + \gamma_{k+1}(x_k - y_k) - \gamma_k(x_{k-1} - y_{k-1}) \\ &= \gamma_k(x_k - y_{k-1}) + \gamma_{k+1}(x_k - y_k) - \gamma_k(x_{k-1} - y_{k-1}), \end{aligned}$$

which implies that $\|z_k - z_{k-1}\| \leq 3\gamma_k D_X$. Using this observation, (1.5) and the definition of p_k, we have

$$\|p_k - p_{k-1}\|_* = \|f'(z_{k-1}) - f'(z_{k-2})\|_* \leq 3\gamma_{k-1} L D_X. \tag{4.3}$$

Hence, the difference between p_k and p_{k-1} vanishes as k increases. By exploiting this fact, we establish in Corollary 1 certain necessary conditions about the LO oracle, under which the rate of convergence of the PA-CG algorithm can be improved. It should be noted, however, that this result is more of theoretical interest only, since these assumptions on the LO oracle are quite strong and hard to be satisfied over a global scope.

Corollary 1 *Let $\{y_k\}$ be the sequence generated by the PA-CG method applied to problem (1.1) with the stepsize policy in (3.1). Suppose that the* LO *oracle satisfies*

$$\|x_k - x_{k-1}\| \leq Q\|p_k - p_{k-1}\|_*^\rho, \quad k \geq 2, \tag{4.4}$$

for some $\rho \in (0, 1]$ and $Q > 0$. Then we have, for any $k \geq 1$,

$$f(y_k) - f^* \leq \mathcal{O}(1) \begin{cases} Q^2 L^{2\rho+1} D_X^{2\rho} / \left[(1-2\rho)\, k^{2\rho+1}\right], & \rho \in (0, 0.5), \\ Q^2 L^2 D_X \log(k+1)/k^2, & \rho = 0.5, \\ Q^2 L^{2\rho+1} D_X^{2\rho} / \left[(2\rho-1)\, k^2\right], & \rho \in (0.5, 1]. \end{cases} \tag{4.5}$$

Proof. Let γ_k be defined in (3.7). By (4.3) and (4.4), we have

$$\|x_k - x_{k-1}\| \leq Q\|p_k - p_{k-1}\|_*^\rho \leq Q(3\gamma_k L D_X)^\rho$$

for any $k \geq 2$. The result follows by plugging the above bound into (4.1) and noting that

$$\sum_{i=1}^{k} (i+1)^{-2\rho} \leq \begin{cases} \frac{(k+1)^{-2\rho+1}}{1-2\rho}, & \rho \in (0, 0.5), \\ \log(k+1), & \rho = 0.5, \\ \frac{1}{2\rho-1}, & \rho \in (0.5, 1]. \end{cases}$$

■

The bound obtained in (4.5) provides some interesting insights on the relation between first-order LCP methods and the general optimal first-order methods for CP. More specifically, if the LO oracle satisfies the Holder's continuity condition (4.4) for some $\rho \in (0.5, 1]$, then we can obtain an $\mathcal{O}(1/k^2)$ rate of convergence for the PA-CG method for solving $\mathcal{F}_{L,\|\cdot\|}^{1,1}(X)$.

4.2 Primal-dual averaging CG methods

Our goal in this subsection is to present another new LCP method, namely the primal-dual averaging CG method, obtained by introducing a different acceleration scheme into the CG method. This algorithm is formally described as follows.

Algorithm 5 The Primal-Dual Averaging Conditional Gradient (PDA-CG) Method

Let $x_0 \in X$ be given and set $y_0 = x_0$.
for $k = 1, \ldots$ **do**
  Set $z_{k-1} = \frac{k-1}{k+1} y_{k-1} + \frac{2}{k+1} x_{k-1}$.
  Set $p_k = \Theta_k^{-1} \sum_{i=1}^{k} [\theta_i f'(z_{i-1})]$, where $\theta_i \geq 0$ are given and $\Theta_k = \sum_{i=1}^{k} \theta_i$.
  Call the LO oracle to compute $x_k \in \mathrm{Argmin}_{x \in X} \langle p_k, x \rangle$.
  Set $y_k = (1 - \alpha_k) y_{k-1} + \alpha_k x_k$ for some $\alpha_k \in [0, 1]$.
end for

Clearly, the above PDA-CG method is also a special LCP algorithm. While the input vector p_k to the LO oracle is set to $f'(z_{k-1})$ in the PA-CG method in the previous subsection, the vector p_k in the PDA-CG method is defined as a weighted average of $f'(z_{i-1})$, $i = 1, \ldots, k$, for some properly chosen weights θ_i, $i = 1, \ldots, k$. This algorithm can also be viewed as the projection-free version of an ∞-memory variant of Nesterov's accelerated gradient method as stated in [28,34].

Note that by convexity of f, the function $\Psi_k(x)$ given by

$$\Psi_k(x) := \begin{cases} 0, & k = 0, \\ \Theta_k^{-1} \sum_{i=1}^{k} \theta_i l_f(z_{i-1}; x), & k \geq 1, \end{cases} \tag{4.6}$$

underestimates $f(x)$ for any $x \in X$. In particular, by the definition of x_k in Algorithm 5, we have

$$\Psi_k(x_k) \leq \Psi_k(x) \leq f(x), \quad \forall\, x \in X, \tag{4.7}$$

and hence $\Psi_k(x_k)$ provides a lower bound on the optimal value f^* of problem (1.1). In order to establish the convergence of the PDA-CG method, we first need to show a simple technical result about $\Psi_k(x_k)$.

Lemma 4 *Let $\{x_k\}$ and $\{z_k\}$ be the two sequences computed by the PDA-CG method. We have*

$$\theta_k \, l_f(z_{k-1}; x_k) \leq \Theta_k \Psi_k(x_k) - \Theta_{k-1} \Psi_{k-1}(x_{k-1}), \quad k = 1, 2, \ldots, \tag{4.8}$$

where $l_f(\cdot\,;\cdot)$ and $\Psi_k(\cdot)$ are defined in (1.12) and (4.6), respectively.

Proof. It can be easily seen from (4.6) and the definition of x_k in Algorithm 5 that $x_k \in \mathrm{Argmin}_{x \in X} \Psi_k(x)$ and hence that $\Psi_{k-1}(x_{k-1}) \leq \Psi_{k-1}(x_k)$. Using the previous observation and (4.6), we obtain

$$\begin{aligned} \Theta_k \Psi_k(x_k) &= \sum_{i=1}^{k} \theta_i l_f(z_{i-1}; x_i) = \theta_k \, l_f(z_{k-1}; x_k) + \sum_{i=1}^{k-1} \theta_i l_f(z_{i-1}; x_i) \\ &= \theta_k l_f(z_{k-1}; x_k) + \Theta_{k-1} \Psi_{k-1}(x_k) \\ &\geq \theta_k l_f(z_{k-1}; x_k) + \Theta_{k-1} \Psi_{k-1}(x_{k-1}). \end{aligned}$$

■

We are now ready to establish the main convergence properties of the PDA-CG method.

Theorem 8 *Let $\{x_k\}$ and $\{y_k\}$ be the two sequences generated by the PDA-CG method applied to problem (1.1) with the stepsize policy in (3.1) or (3.2). Also let $\{\gamma_k\}$ be defined in (3.7). If the parameters θ_k are chosen such that*

$$\theta_k \Theta_k^{-1} = \gamma_k, \quad k = 1, \ldots, \tag{4.9}$$

Then, we have

$$f(y_k) - f^* \leq f(y_k) - \Psi_k(x_k) \leq \frac{2L}{k(k+1)} \sum_{i=1}^{k} \|x_i - x_{i-1}\|^2 \tag{4.10}$$

for any $k = 1, 2, \ldots$, where L is given by (1.13).

Proof. Denote $\tilde{y}_k = (1-\gamma_k)y_{k-1}+\gamma_k x_k$. It follows from (3.1) (or (3.2)) and the definition of y_k that $f(y_k) \le f(\tilde{y}_k)$. Also noting that, by definition, we have $z_{k-1} = (1-\gamma_k)y_{k-1} + \gamma_k x_{k-1}$ and hence

$$\tilde{y}_k - z_{k-1} = \gamma_k(x_k - x_{k-1}).$$

Using these two observations, (1.13), the definitions of x_k in Algorithm 5, the convexity of f and (4.8), we obtain

$$\begin{aligned} f(y_k) &\le f(\tilde{y}_k) \le l_f(z_{k-1};\tilde{y}_k) + \frac{L}{2}\|\tilde{y}_k - z_{k-1}\|^2 \\ &= (1-\gamma_k) l_f(z_{k-1};y_{k-1}) + \ \gamma_k l_f(z_{k-1};x_k) + \frac{L}{2}\gamma_k^2\|x_k - x_{k-1}\|^2 \\ &= (1-\gamma_k) f(y_{k-1}) + \gamma_k l_f(x_k;z_{k-1}) + \frac{L}{2}\gamma_k^2\|x_k - x_{k-1}\|^2 \\ &\le (1-\gamma_k) f(y_{k-1}) + \gamma_k\theta_k^{-1}[\Theta_k\Psi_k(x_k) - \Theta_{k-1}\Psi_{k-1}(x_{k-1})] + \frac{L}{2}\gamma_k^2\|x_k - x_{k-1}\|^2. \end{aligned} \tag{4.11}$$

Also, using (4.9) and the fact that $\Theta_{k-1} = \Theta_k - \theta_k$, we have

$$\begin{aligned} \gamma_k\theta_k^{-1}[\Theta_k\Psi_k(x_k) - \Theta_{k-1}\Psi_{k-1}(x_{k-1})] &= \Psi_k(x_k) - \Theta_{k-1}\Theta_k^{-1}\Psi_{k-1}(x_{k-1}) \\ &= \Psi_k(x_k) - \left(1 - \theta_k\Theta_k^{-1}\right)\Psi_{k-1}(x_{k-1}) \\ &= \Psi_k(x_k) - (1-\gamma_k)\Psi_{k-1}(x_{k-1}). \end{aligned}$$

Combining the above two relations and re-arranging the terms, we obtain

$$f(y_k) - \Psi_k(x_k) \le (1-\gamma_k)\left[f(y_{k-1}) - \Psi_{k-1}(x_{k-1})\right] + \frac{L}{2}\gamma_k^2\|x_k - x_{k-1}\|^2,$$

which, in view of Lemma 1, (3.7) and (3.8), then implies that

$$f(y_k) - \Psi_k(x_k) \le \frac{2L}{k(k+1)}\sum_{i=1}^{k}\|x_i - x_{i-1}\|^2.$$

Our result then immediately follows from (4.7) and the above inequality. ∎

We now add a few remarks about the results obtained in Theorem 8. Firstly, observe that we can simply set $\theta_k = k$, $k = 1, 2, \ldots$ in order to satisfy (4.9). Secondly, in view of the discussion after Theorem 7, the PDA-CG method is also an optimal LCP method for $\mathcal{F}^{1,1}_{L,\|\cdot\|}(X)$ when n is sufficiently large, since its the rate of convergence is exactly the same as the one for the PD-CG method. In addition, its rate of convergence is invariant of the selection of the norm $\|\cdot\|$ (see (3.13)). Thirdly, according to (4.10), we can compute an online lower bound $\Psi_k(x_k)$ on the optimal value f^*, and terminate the PDA-CG method based on the optimality gap $f(y_k) - \Psi_k(x_k)$.

Similar to the PA-CG method, the rate of convergence of the PDA-CG method depends on $x_k - x_{k-1}$, which in turn depends on the geometry of X and the input vectors p_k and p_{k-1} to the LO oracle. One can easily check the closeness between p_k and p_{k-1}. Indeed, by the definition of p_k, we have $p_k = \Theta_k^{-1}[(1-\theta_k)p_{k-1} + \theta_k f_k'(z_{k-1})$ and hence

$$p_k - p_{k-1} = \Theta_k^{-1}\theta_k[p_{k-1} + f_k'(z_{k-1})] = \gamma_k[p_{k-1} + f_k'(z_{k-1})], \tag{4.12}$$

where the last inequality follows from (4.9). Noting that by (1.13), we have $\|f'(x)\|_* \le \|f'(x^*)\|_* + LD_X$ for any $x \in X$ and hence that $\|p_k\|_* \le \|f'(x^*)\|_* + LD_X$ due to the definition of p_k. Using these observations, we obtain

$$\|p_k - p_{k-1}\|_* \le 2\gamma_k[\|f'(x_*)\|_* + LD_X], \quad k \ge 1. \tag{4.13}$$

Hence, under certain continuity assumptions on the LO oracle, we can obtain a result similar to Corollary 1. Note that both stepsize policies in (3.1) and (3.2) can be used in this result.

Corollary 2 *Let $\{y_k\}$ be the sequences generated by the PDA-CG method applied to problem (1.1) with the stepsize policy in (3.1) or (3.2). Assume that (4.9) holds. Also suppose that the* LO *oracle satisfies (4.4) for some $\rho \in (0,1]$ and $Q > 0$. Then we have, for any $k \geq 1$,*

$$f(y_k) - f^* \leq \mathcal{O}(1) \begin{cases} LQ^2 \left[\|f'(x_*)\|_* + L\ D_X\right]^{2\rho} / \left[(1-2\rho)\, k^{2\rho+1}\right], & \rho \in (0, 0.5), \\ LQ^2 \left[\|f'(x_*)\|_* + L\ D_X\right] \log(k+1)/k^2, & \rho = 0.5, \\ LQ^2 \left[\|f'(x_*)\|_* + L\ D_X\right]^{2\rho} / \left[(2\rho-1)\, k^2\right], & \rho \in (0.5, 1]. \end{cases} \tag{4.14}$$

Similar to Corollary 1, Corollary 2 also helps to build some connections between LCP methods and the more general optimal first-order method. However, these results are more of theoretical interest only, since the LO oracle does not necessarily satisfy (4.4) for any $\rho > 0$, but only for $\rho = 0$ and $Q = D_X$.

4.3 Numerical Illustration

Our goal in this subsection is to compare through some preliminary numerical experiments the three LCP methods for solving $\mathcal{F}^{1,1}_{L,\|\cdot\|}(X)$, i.e., CG, PA-CG and PDA-CG, all of which share similar worst-case complexity bounds. More specifically, we conduct three sets of experiments for solving quadratic programming (QP) problems over a few different types of feasible sets. In our first set of experiments, we consider the QP problems over a standard simplex or spectrahedron. In particular, let $A \in \mathbb{R}^{m\times n}$, $\mathcal{A} : \mathbb{R}^{n\times n} \to \mathbb{R}^m$ and $b \in \mathbb{R}^m$ be given, the QP over a standard simplex and over a standard spectrahedron, respectively, are defined as $\min_{x\in\Delta_n} \|Ax - b\|_2^2$ and $\min_{x\in S_n} \|\mathcal{A}x - b\|_2^2$, where

$$\Delta_n := \left\{ x \in \mathbb{R}^n : \sum_{i=1}^n x_i = 1, x_i \geq 0, i = 1, \ldots, n\} \right\} \tag{4.15}$$

and

$$S_n := \left\{ x \in \mathbb{R}^{n\times n} : \mathrm{Tr}(x) = 1, x \succeq 0 \right\}. \tag{4.16}$$

We can easily see that $\Delta_n \subset S_n$ by setting x to be diagonal in (4.16). In our second set of experiments, we consider the QP problems over a hypercube, i.e., $\min_{x\in B_n} \|Ax - b\|_2^2$, where

$$B_n := \left\{ x \in \mathbb{R}^n : x_i \in [0,1], i = 1, \ldots, n \right\}. \tag{4.17}$$

It is well-known that to solve these problems becomes more and more difficult as n increases. In our last set of experiments, we consider the QP problems over a hypercube intersected with a simplex, i.e., $\min_{x\in H_n(r)} \|Ax - b\|_2^2$, where

$$H_n(r) := \left\{ x \in \mathbb{R}^n : \sum_{i=1}^n x_i \leq rn, x_i \in [0,1] \right\} \tag{4.18}$$

for some $r \in (0,1]$. These problems arise from certain important applications, including compressed sensing and portfolio optimization.

Our experiments have been carried out on a set of instances that are randomly generated as follows. Firstly, we randomly generate a feasible solution s_0 in Δ_n, B_n, S_n and $H_n(r)$, and a linear operator $A : \mathbb{R}^n \to \mathbb{R}^m$ (or $\mathcal{A} : \mathbb{R}^{n\times n} \to \mathbb{R}^m$) with sparse entries uniformly distributed over $[0,1]$. Then we compute $b = As_0$ (or $b = \mathcal{A}s_0$). Clearly in this way, the optimal values of these instances are given by 0. Also note that a sparsity parameter d has been used when generating the linear operator A (or $\mathcal{A}$). Totally 36 instances have been generated, see Table 1 for more details.

The CG, PA-CG and PDA-CG algorithms are implemented in Matlab R2011b. Observe that we have discussed two stepsize policies for these algorithms: the stepsize policy (3.1) is the one that we have used in our experiments for its simplicity, while one can also use the stepsize policy (3.2) with more expensive iteration costs. The parameter θ_k in PDA-CG is simply set to $\theta_k = k$. The initial point y_0 is randomly generated and remains the same for different algorithms. We report the results in Tables 2, 3 and 4, respectively, for minimization over simplex/spectrahedron, hypercube, and hypercube intersected with simplex. For each problem instance, we compute the objective values at the search points y_0, y_{100} and y_{1000}, and the total CPU time (in seconds, Intel Core i7-2600 3.4 GHz) required for performing $1,000$ iterations of these algorithms.

Table 1 Randomly generated instances

Inst.	Domain	n	m	d	Inst.	Domain	n	m	d
SIM11	Δ_n	2,000	500	1.0	SIM12	Δ_n	2,000	1,000	1.0
SIM21	Δ_n	4,000	1,000	0.8	SIM22	Δ_n	4,000	2,000	0.8
SIM31	Δ_n	8,000	2,000	0.6	SIM32	Δ_n	8,000	4,000	0.6
SPE41	S_n	100	500	0.6	SPE42	S_n	100	1,000	0.6
SPE51	S_n	200	500	0.4	SPE52	S_n	200	1,000	0.4
SPE61	S_n	400	500	0.2	SPE62	S_n	400	1,000	0.2
CUB11	C_n	500	100	1.0	CUB12	C_n	500	200	1.0
CUB21	C_n	1,000	250	1.0	CUB22	C_n	1,000	5,00	1.0
CUB31	C_n	2,000	500	1.0	CUB32	C_n	2,000	1,000	1.0
CUB41	C_n	4,000	1,000	0.8	CUB42	C_n	4,000	2,000	0.8
CUB51	C_n	8,000	2,000	0.6	CUB52	C_n	8,000	4,000	0.6
CUB61	C_n	16,000	4,000	0.4	CUB62	C_n	16,000	8,000	0.4
HYB11	$H_n(0.25)$	4,000	1,000	0.8	HYB12	$H_n(0.25)$	4,000	2,000	0.8
HYB21	$H_n(0.5)$	4,000	1,000	0.8	HYB22	$H_n(0.5)$	4,000	2,000	0.8
HYB31	$H_n(0.25)$	8,000	2,000	0.6	HYB32	$H_n(0.25)$	8,000	4,000	0.6
HYB41	$H_n(0.5)$	8,000	2,000	0.6	HYB42	$H_n(0.5)$	8,000	4,000	0.6
HYB51	$H_n(0.25)$	16,000	4,000	0.4	HYB52	$H_n(0.25)$	16,000	8,000	0.4
HYB61	$H_n(0.5)$	16,000	4,000	0.4	HYB62	$H_n(0.5)$	16,000	8,000	0.4

Table 2 Comparison of CG methods for minimization over simplex/spectrahedron

		CG			PA-CG			PDA-CG		
Inst	$f(y_0)$	$f(y_{100})$	$f(y_{1000})$	Time	$f(y_{100})$	$f(y_{1000})$	Time	$f(y_{100})$	$f(y_{1000})$	Time
SIM11	1.27e-1	1.09e-1	2.81e-3	1.84	1.14e-1	2.69e-3	3.54	1.27e-1	5.02e-3	3.32
SIM12	2.49e-1	2.49e-1	8.20e-3	3.47	2.91e-1	9.22e-3	6.75	2.49e-1	1.30e-2	6.85
SIM21	1.25e-1	1.25e-1	7.94e-3	6.15	1.25e-1	7.87e-3	12.01	1.25e-1	1.49e-2	12.33
SIM22	2.53e-1	2.53e-1	2.55e-2	11.99	2.53e-1	2.54e-2	23.17	2.53e-1	3.74e-2	23.19
SIM31	1.17e-1	1.17e-1	2.32e-2	18.75	1.17e-1	2.36e-2	37.24	1.17e-1	4.13e-2	37.18
SIM32	2.25e-1	2.25e-1	7.02e-2	38.78	2.25e-1	6.80e-2	75.70	2.25e-1	9.87e-2	75.87
SPE41	5.51e+1	1.65e-1	1.66e-3	14.00	2.80e-1	2.91e-3	19.79	4.47e-1	5.72e-3	21.21
SPE42	1.01e+2	5.24e-1	6.33e-3	18.90	8.74e-1	1.13e-2	30.79	9.70e-1	1.77e-2	33.77
SPE51	1.52e+1	8.85e-2	8.40e-4	30.90	1.90e-1	1.65e-3	46.65	1.87e-1	1.94e-3	48.33
SPE52	3.65e+1	1.97e-1	1.99e-3	45.89	3.22e-1	3.55e-3	78.00	8.73e-1	1.37e-2	79.86
SPE61	3.01e+0	3.90e-2	3.84e-4	73.80	1.00e-1	9.60e-4	104.97	5.23e-2	5.64e-4	112.80
SPE62	5.92e+0	8.02e-2	8.00e-4	109.34	1.58e-1	1.44e-3	177.80	1.82e-1	2.02e-3	181.01

Table 3 Comparison of CG methods for minimization over hypercube

		CG			PA-CG			PDA-CG		
Inst	$f(y_0)$	$f(y_{100})$	$f(y_{1000})$	Time	$f(y_{100})$	$f(y_{1000})$	Time	$f(y_{100})$	$f(y_{1000})$	Time
CUB11	1.98e+5	3.52e+1	3.50e-1	0.13	2.04e+1	1.41e+0	0.23	2.94e+0	3.17e-2	0.24
CUB12	4.02e+4	9.96e+1	3.64e+0	0.18	1.16e+2	1.08e+1	0.36	4.81e+0	1.65e-2	0.34
CUB21	2.18e+6	5.23e+2	1.53e+0	0.40	4.33e+2	4.46e+1	0.70	1.51e+1	3.24e-1	0.72
CUB22	4.49e+6	9.61e+2	7.60e+1	0.84	8.90e+2	1.86e+2	1.45	6.67e+1	1.67e-1	1.47
CUB31	1.73e+7	2.43e+3	2.13e+2	1.82	2.23e+3	4.68e+2	3.33	2.60e+2	1.67e+0	3.33
CUB32	3.25e+7	5.35e+3	6.74e+2	3.43	5.85e+3	1.56e+3	6.57	7.84e+2	1.41e+0	6.60
CUB41	1.03e+8	1.03e+4	1.38e+3	6.02	9.50e+3	2.46e+3	11.95	1.58e+3	1.23e+1	11.80
CUB42	1.95e+8	2.24e+4	4.64e+3	11.82	2.00e+4	8.88e+3	23.05	4.95e+3	1.04e+1	23.74
CUB51	5.43e+8	4.65e+4	9.83e+3	18.69	4.70e+4	1.22e+4	37.22	8.70e+3	6.63e+1	37.44
CUB52	1.09e+9	7.38e+4	2.74e+4	38.62	7.60e+4	3.48e+4	75.88	2.21e+4	5.53e+1	76.18
CUB61	2.32e+9	1.39e+5	4.56e+4	59.04	1.13e+5	4.94e+4	117.40	2.96e+4	3.60e+2	116.52
CUB62	4.60e+9	2.26e+5	1.25e+5	115.62	2.71e+5	1.39e+5	228.64	9.79e+4	2.35e+2	226.36

Table 4 Comparison of CG methods for minimization over hypercube intersected with simplex

		CG			PA-CG			PDA-CG		
Inst	$f(y_0)$	$f(y_{100})$	$f(y_{1000})$	Time	$f(y_{100})$	$f(y_{1000})$	Time	$f(y_{100})$	$f(y_{1000})$	Time
HYB11	1.58e+7	1.12e+3	7.80e+1	6.60	1.14e+3	8.27e+1	12.50	4.88e+1	2.72e-1	12.33
HYB12	3.11e+7	3.56e+3	1.11e+3	12.39	3.45e+3	1.06e+3	23.91	1.12e+3	8.16e+0	24.27
HYB21	1.00e+8	2.48e+3	6.18e+2	6.58	2.10e+3	7.50e+2	12.23	4.66e+2	1.05e+1	12.32
HYB22	2.00e+8	7.73e+3	3.39e+3	12.32	6.61e+3	3.82e+3	23.97	1.53e+3	7.67e+0	23.83
HYB31	8.45e+7	5.75e+3	3.67e+2	20.06	4.73e+3	3.85e+3	39.09	3.65e+2	1.77e+0	38.84
HYB32	1.67e+8	1.58e+4	4.29e+3	40.27	1.45e+4	4.41e+3	78.10	5.45e+3	3.94e+1	78.26
HYB41	5.47e+8	1.02e+4	3.04e+3	20.01	1.00e+4	3.44e+3	38.15	3.95e+3	5.25e+1	38.43
HYB42	1.06e+9	3.09e+4	1.56e+4	39.90	3.15e+4	1.73e+4	79.91	1.04e+4	4.35e+1	79.07
HYB51	3.57e+8	1.82e+4	1.82e+3	60.14	1.74e+4	1.81e+3	117.99	1.55e+3	7.00e+0	117.76
HYB52	7.10e+8	5.58e+4	1.62e+4	117.23	5.52e+4	1.71e+4	231.05	1.66e+4	1.34e+2	232.59
HYB61	2.33e+9	3.88e+4	1.26e+4	60.64	3.76e+4	1.42e+4	119.09	1.84e+4	1.98e+2	118.71
HYB62	4.69e+9	1.08e+5	5.34e+4	117.80	1.00e+5	5.93e+4	233.31	6.85e+4	2.02e+2	232.12

We make a few observations about the results obtained in Tables 2, 3 and 4. Firstly, for solving the QP problems over a standard simplex/spectrahedron, all these three algorithms are about the same, with the CG method slightly outperforming the other two. Secondly, for solving the QP problems over a hypercube, PDA-CG can significantly outperform both CG and PA-CG by orders of magnitude. More specifically, as it can be seen from Table 3, although the CPU times for PDA-CG are about as twice as the ones for CG, the function values computed at the 100 iterations of PDA-CG are already comparable to those computed at the $1,000$ iterations for both CG and PA-CG. Moreover, the objective values at the $1,000$ iterations of the PDA-CG are better than those for CG and PA-CG by $1-3$ accuracy digits, and the difference seems to become larger as n increases. Thirdly, it can be seen from Table 4 that PDA-CG also outperforms both CG and PA-CG by orders of magnitude for solving the QP problems over a hypercube intersected with simplex. Therefore, we conclude that the PDA-CG method, although sharing similar worst-case complexity bounds with both CG and PA-CG, might significantly outperform the latter two algorithms for solving certain classes of CP problems, e.g., those with box-type constraints.

5 Concluding remarks

In this paper, we study a new class of optimization algorithms, namely the LCP methods, which covers the classic CG method as a special case. We establish a few lower complexity bounds for these algorithms to solve different classes of CP problems. We formally show that the classic CG method is an optimal LCP method for solving smooth CP problems and present new variants of this algorithm that are optimal or nearly optimal for solving certain saddle point and general nonsmooth problems under an LO oracle. Finally, we develop a few new LCP methods, namely PA-CG and PDA-CG, by properly modifying Nesterov's accelerated gradient method, and show that they also exhibit the optimal rate of convergence for solving smooth CP problems under an LO oracle. In addition, we demonstrate through our preliminary numerical experiments that the PDA-CG method can significantly outperform the classic CG for solving certain classes of large-scale CP problems.

References

1. S.D. Ahipasaoglu and M.J. Todd. A modified frank-wolfe algorithm for computing minimum-area enclosing ellipsoidal cylinders: Theory and algorithms. *Computational Geometry*, 46:494–519, 2013.
2. F. Bach, S. Lacoste-Julien, and G. Obozinski. On the equivalence between herding and conditional gradient algorithms. In *the 29th International Conference on Machine Learning*, 2012.
3. A. Beck and M. Teboulle. A conditional gradient method with linear rate of convergence for solving convex linear systems. *Math. Methods Oper. Res.*, 59:235–247, 2004.
4. A. Ben-Tal and A. S. Nemirovski. Non-Euclidean restricted memory level method for large-scale convex optimization. *Mathematical Programming*, 102:407–456, 2005.
5. D.P. Bertsekas. Stochastic optimization problems with nondifferentiable cost functionals. *Journal of Optimization Theory and Applications*, 12:218–231, 1973.
6. M.D. Canon and C.D. Cullum. A tight upper bound on the rate of convergence of frank-wolfe algorithm. *SIAM Journal on Control*, 6(4):509–516, 1968.

7. B. Cox, A. Juditsky, and A. S. Nemirovski. Dual subgradient algorithms for large-scale nonsmooth learning problems. Manuscript, School of ISyE, Georgia Tech, Atlanta, GA, 30332, USA, 2013. submitted to *Mathematical Programming*, Series B.
8. G.B. Dantzig. Converting a converging algorithm into a polynomially bounded algorithm. Technical report sol 91-5, Stanford University, 1991.
9. G.B. Dantzig. An ϵ-precise feasible solution to a linear program with a convexity constraint in $1/\epsilon^2$ iterations independent of problem size. Technical report sol 91-5, Stanford University, 1992.
10. V. Demyanov and A. Rubinov. *Approximate Methods in Optimization Problems*. American Elsevier, 1970.
11. J. C. Duchi, P. L. Bartlett, and M. J. Wainwright. Randomized smoothing for stochastic optimization. *SIAM Journal on Optimization*, 22:674–701, 2012.
12. M. Epelman and R. M. Freund. Condition number complexity of an elementary algorithm for computing a reliable solution of a conic linear system. *Mathematical Programming*, 88:451485, 2000.
13. M. Frank and P. Wolfe. An algorithm for quadratic programming. *Naval Research Logistics Quarterly*, 3:95–110, 1956.
14. D. Garber and E. Hazan. A linearly convergent conditional gradient algorithm with applications to online and stochastic optimization. Manuscript, January 2013. arXiv:1301.4666.
15. S. Ghadimi and G. Lan. Optimal stochastic approximation algorithms for strongly convex stochastic composite optimization, II: shrinking procedures and optimal algorithms. Technical report, 2010. *SIAM Journal on Optimization* (to appear).
16. S. Ghadimi and G. Lan. Stochastic first- and zeroth-order methods for nonconvex stochastic programming. Technical report, Department of Industrial and Systems Engineering, University of Florida, Gainesville, FL 32611, USA, June 2012. *SIAM Journal on Optimization* (under second-round review).
17. Z. Harchaoui, A. Juditsky, and A. S. Nemirovski. Conditional gradient algorithms for machine learning. NIPS OPT workshop, 2012.
18. M. Jaggi. Revisiting frank-wolfe: Projection-free sparse convex optimization. In *the 30th International Conference on Machine Learning*, 2013.
19. M. Jaggi and M. Sulovský. A simple algorithm for nuclear norm regularized problems. In *the 27th International Conference on Machine Learning*, 2010.
20. V. Katkovnik and Y. Kulchitsky. Convergence of a class of random search algorithms. *Automation and Remote Control*, 33:1321–1326, 1972.
21. G. Lan. An optimal method for stochastic composite optimization. *Mathematical Programming*, 133(1):365–397, 2012.
22. G. Lan. Bundle-level type methods uniformly optimal for smooth and non-smooth convex optimization. Manuscript, Department of Industrial and Systems Engineering, University of Florida, Gainesville, FL 32611, USA, Janurary 2013. submitted to Mathematical Programming.
23. R. Luss and M. Teboulle. Conditional gradient algorithms for rank one matrix approximations with a sparsity constraint. *SIAM Review*, 55:65–98, 2013.
24. A. S. Nemirovski, A. Juditsky, G. Lan, and A. Shapiro. Robust stochastic approximation approach to stochastic programming. *SIAM Journal on Optimization*, 19:1574–1609, 2009.
25. A. S. Nemirovski and D. Yudin. *Problem complexity and method efficiency in optimization*. Wiley-Interscience Series in Discrete Mathematics. John Wiley, XV, 1983.
26. Y. E. Nesterov. A method for unconstrained convex minimization problem with the rate of convergence $O(1/k^2)$. *Doklady AN SSSR*, 269:543–547, 1983.
27. Y. E. Nesterov. *Introductory Lectures on Convex Optimization: a basic course*. Kluwer Academic Publishers, Massachusetts, 2004.
28. Y. E. Nesterov. Smooth minimization of nonsmooth functions. *Mathematical Programming*, 103:127–152, 2005.
29. Y. E. Nesterov. Barrier subgradient method. Technical report, Center for Operations Research and Econometrics (CORE), Catholic University of Louvain, October 2008.
30. Y. E. Nesterov. Random gradient-free minimization of convex functions. Technical report, Center for Operations Research and Econometrics (CORE), Catholic University of Louvain, January 2010.
31. R. Y. Rubinstein. *Simulation and the Monte Carlo Method*. Wiley, New York, 1981.
32. A. Gonen S. Shalev-Shwartz and O. Shamir. Large-scale convex minimization with a low rank constraint. In *the 28th International Conference on Machine Learning*, 2011.
33. C. Shen, J. Kim, L. Wang, and A. van den Hengel. Positive semidefinite metric learning using boosting-like algorithms. *Journal of Machine Learning Research*, 13:1007–1036, 2012.
34. P. Tseng. On accelerated proximal gradient methods for convex-concave optimization. Manuscript, University of Washington, Seattle, May 2008.